
domonic

Release 0.9.11

byteface

Jun 28, 2022

CONTENTS:

1	Generate HTML with Python 3	3
1.1	HTML TEMPLATING	3
1.2	install	4
1.3	CLI	4
1.4	The User Guide	4
1.5	Join-In	352
1.6	EXAMPLE PROJECTS	352
1.7	Disclaimer	352
	Python Module Index	353
	Index	355



GENERATE HTML WITH PYTHON 3

Domonic Not only a Python library for generating HTML

Domonic contains several evolving packages:

- `html` : Generate html with python 3
- `dom` : DOM API in python 3
- `javascript` : js API in python 3
- `terminal || cmd` : call terminal commands with python3 (*see at the end*)
- `JSON` : utils for loading / decorating / transformin
- `SVG` : Generate svg using python
- `aframe || x3d tags` : auto generate 3d worlds with aframe. (see examples folder)
- `dQuery` - Utils for querying domonic. (alt + 0 for the ° symbol)
- `geom` - `vec2`, `vec3` with `_dunders_` as well as shape classes

Take a look at the source code and contribute!

1.1 HTML TEMPLATING

```
from domonic.html import *

mydom = html(body(h1('Hello, World!')))
print(f"{mydom}")
```

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

To pretty print use an f-string. Which also adds the doctype.

1.2 install

```
python3 -m pip install domonic
```

or if you had it before upgrade:

```
python3 -m pip install domonic --upgrade
```

1.3 CLI

There's a few args you can pass to domonic on the command line to help you out.

To view the online the docs:

```
domonic -h
```

To see the version:

```
domonic -v
```

To quickly create a domonic project for prototyping:

```
domonic -p
```

To use xpath on a website (new):

```
domonic -x https://google.com //a
```

1.4 The User Guide

Here you can find some instructions for getting the most out of Domonic.

1.4.1 html

with domonic you can create beautiful, clean <html> straight out of the box.

```
mydom = html(body(h1('Hello, World!')))
print(f"{mydom}")
```

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

rendering

you can cast str() on any element to render it.

```
el_string = str(div())
print(el_string)
```

there's also a render method that takes 2 parameters, some pyml and an optional output file.

```
from domonic.html import *
page = div(span('Hello World'))
render(page, 'index.html')
```

templating

```
from domonic.html import *

output = render(
    html(
        head(
            style(),
            script(),
        ),
        body(
            div("hello world"),
            a("this is a link", _href="http://www.somesite.com", _style="font-
↪size:10px;"),
            ol(''.join([f'{li()}' for thing in range(5)])),
            h1("test", _class="test"),
        )
    )
)
```

```
<html><head><style></style><script></script></head><body><div>hello world</div><a href=
↪"http://www.somesite.com" style="font-size:10px;">this is a link</a><ol><li></li><li></li><li></li><li></li><li></li></ol><h1 class="test">test</h1></body></html>
```

Take a look in tests/test_html.py at the bootstrap5 alpha examples. All tests passed on several templates.

usage

```
print(html(body(h1('Hello, World!'))))
```

```
<html><body><h1>Hello, World!</h1></body></html>
```

attributes

prepend attributes with an underscore (avoids clashing with python keywords)

```
test = label(_class='classname', _for="someinput")
print(test)
```

```
<label class="classname" for="someinput"></label>
```

lists

just do list comprehension and join it to strip the square brackets

```
ul(''.join([f'{li()}' for thing in range(5)])),
```

```
<ul><li></li><li></li><li></li><li></li><li></li></ul>
```

data-tags

python doesn't allow hyphens in parameter names. so use variable keyword argument syntax for custom data-tags

```
div("test", **{"_data-test": "test"} )
```

DONT FORGET TO PREPEND THE UNDERSCORE.

script tags

load from a source...

```
script(_src="/docs/5.0/dist/js/bootstrap.bundle.min.js", _integrity="sha384-1234", _
↪crossorigin="anonymous"),
```

or do inline js...

```
script("""
let itbe = ""
"""),
```

style tags

load from a source...

```
link(_href="/docs/5.0/dist/css/bootstrap.min.css", _rel="stylesheet", __integrity=
↳ "sha384-12345", __crossorigin="anonymous"),
```

or do inline css...

```
style("""
.bd-placeholder-img {
    font-size: 1.125rem;
    text-anchor: middle;
    -webkit-user-select: none;
    -moz-user-select: none;
    -ms-user-select: none;
    user-select: none;
}
@media (min-width: 768px) {
    .bd-placeholder-img-lg {
        font-size: 3.5rem;
    }
}
"""),
```

Create Elements

to create your own custom elements you can use `create_element`

```
from domonic.html import *
create_element('custom_el', div('some content'), _id="test")
```

or you could use the DOM API...

```
from domonic.dom import *
from domonic.html import *

site = html()
el = document.createElement('myelement')
site.appendChild(el)
print(site)
```

For more info about the DOM API navigate to that section...

Decorators

You can use decorators to wrap elements around function results

```
from domonic.decorators import el

@el(html)
@el(body)
@el(div)
def test():
    return 'hi!'

print(test())
# <html><body><div>hi!</div></body></html>
```

Magic methods

Multiply

You can quickly clone nodes with a multiplier which will return a list...

```
from domonic.html import *
mydivs = div()*100
```

but you will have to render them yourself by iterating and calling string...

```
print(''.join([str(c) for c in mydivs]))
```

Divide

A divisor also creates more but will instead call render and give a list of strings...

```
from domonic.html import *
print(div()/100)
```

but this means they are now rendered and can't be edited.

Although you could convert them back by calling parser then domonify. i.e.

```
mylist = li()/10
myobj = domonic.domonify(domonic.parse(mylist))
print(myobj)
```

OR

If other is anything, it is returned. Otherwise it returns self

```
from domonic.html import *
print(div() | False)
print(div() | True)
```

Another way is to use ternary i.e.

```
mything = div() if True else span(class='warning')
```

In place add/minus

You can add to or remove from the children of a Node with the in-place operators...

```

myorderedlist = ol()
myorderedlist += str(li() / 10)
print(myorderedlist)

```

This also works for text nodes but be aware they will be irreversibly flattened if you render...

```

a1 = button()
a1 += "hi"
a1 += "how"
a1 += ["are", "you", "today"]
print(a1)
a1 -= "hi"
print(a1)

```

Pass a dictionary to the right shift operator to add/update an attribute... (don't forget underscore or it will error)

```

a1 = img()
a1 >> {'_src': "http://www.someurl.com"}
print(a1)

```

Access an elements children as if it were a list...

```

mylist = ul(li(1), li(2), li(3))
print(mylist[1])

```

unpack children...

```

mylist = ul(li(), li(), li())
print(*mylist)
a1, b1, c1 = ul(li(1), li(2), li(3))
print(a1)
a1, b1, c1, d1, e1 = button() * 5
print(a1, b1, c1, d1, e1)

```

f-strings

To pretty print a domonic dom you can use a f-string...

```

print(f"{mydom}")

```

```

<!DOCTYPE html>
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>

```

which basically calls the format dunder on the tag. (if look at the code)

This is useful as it means use different ways to get output from domonic.

```
print(f"{mydom}")
print(f"{mydom!s}")
print(f"{mydom!r}")
print(f"{mydom!a}")
print(str(mydom))
print(mydom.__format__(''))
```

If the built in formatter is not up to your needs. You can also use other libraries that leverage beautifulsoup i.e.

```
output = render(html(body(h1('Hello, World!'))))
from html5print import HTMLBeautifier
print(HTMLBeautifier.beautify(output, 4))
```

For outputting pyml to a string there is a method in production called `__pyml__()` which may become `repr.` (but i'd been saving `repr` for logging)

You can also use this vscode plugin on .pyml and it does a nice job. (inspiration for the 'dentage' method)

<https://marketplace.visualstudio.com/items?itemName=mgbert.indent-nested-dictionary>

Quotes around attributes

The quotes around attributes can be finely controlled using the `DOMConfig.ATTRIBUTE_QUOTES` flag

By default everything is double quoted on render.

However a flag can be set to `None` which will not render quotes if the passed value is not a string.

Alternatively it can be set to use a single quotation mark or even `False` to use `None` at all and control it yourself.

Examples provided below.

```
>>> from domonic.html import *
>>> from domonic.dom import DOMConfig
>>> print(body(test="123"))
# <body test="123"></body>
>>> print(body(test=123))
# <body test="123"></body>
>>> DOMConfig.ATTRIBUTE_QUOTES = None
>>> print(body(test=123))
# <body test=123></body>
>>> print(body(test="123"))
# <body test="123"></body>
>>> DOMConfig.ATTRIBUTE_QUOTES = "'"
>>> print(body(test="123"))
# <body test='123'></body>
>>> DOMConfig.ATTRIBUTE_QUOTES = True
>>> print(body(test="123"))
# <body test="123"></body>
>>> print(body(test=123))
```

(continues on next page)

(continued from previous page)

```
# <body test="123"></body>
>>> DOMConfig.ATTRIBUTE_QUOTES = False
>>> print(body(test="123"))
# <body test=123></body>
>>> print(body(test="TEXT"))
# <body test=TEXT></body>
```

loading .pyml templates

```
div("Hello World")
#<div>Hello tabs</div>
```

‘loads’ imports a pyml file and turns it into a program

this example loads a template and passing params for rendering

```
from domonic import loads
from domonic.html import *

# create some vars. you will see these referenced in the template file
brand = "MyBrand"
links = ['one', 'two', 'three']

# load a template and pass it some data
webpage = domonic.loads('templates/webpage.com.pyml', links=links, brand=brand)

render(webpage, 'webpage.html')
```

‘load’ is different to ‘loads’, it takes html strings and converts to a program

```
from domonic.dQuery import °

webpage = domonic.load('<html><head></head><body id="test"></body></html>')
°(webpage)
°('#test').append(div("Hello World"))
render(webpage, 'webpage2.html')
```

- warning loads also is very basic and can only convert simple html as the parser is still in development

Notes on templating

while you can create a div with content like :

```
div("some content")
```

python doesn’t allow named params before unnamed ones. So you can’t do this:

```
div(_class="container", p("Some content"))
```

or it will complain the params are in the wrong order. You have to instead put content before attributes:

```
div( p("Some content"), _class="container")
```

which is annoying when a div gets long.

You can get around this by using 'html' which is available on every Element:

```
div( _class="container" ).html("Some content")
```

This is NOT like jQuery html func that returns just the inner content. use innerHTML for that.

It is used specifically for rendering.

Common Errors

If a templates syntax is incorrect it will not work.

There's a small learning curve in getting .pyml templates correct. Usually...

- (1) a missing comma between tags,
- (2) an underscore missing on an attribute or
- (3) params in the wrong order.

Use this reference when starting out to help when you get an error.

```
IndexError: list index out of range
# You most likely didn't put a underscore on an attribute.

SyntaxError: invalid syntax
# You are Missing a comma between attributes

SyntaxError: positional argument follows keyword argument
# You have to pass attributes LAST. and strings and objects first. *see notes on
↳ templating above*

TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'dict'
# You are Missing a comma between attributes. before the **{}
```

parsing

<https://github.com/byteface/domonic/issues/28>

Basic usage...

An examples of using the parser...

```
import requests
import html5lib
from domonic.ext.html5lib_ import getTreeBuilder

r = requests.get("https://google.com")
parser = html5lib.HTMLParser(tree=getTreeBuilder())
page = parser.parse(r.content.decode("utf-8"))
```

(continues on next page)

(continued from previous page)

```
# print the page with formatting
# print(f'{page}')

'''
links = page.getElementsByTagName('a')
for l in links:
    try:
        print(l.href)
    except Exception as e:
        # no href on this tag
        pass
'''

# turn the downloaded site into .pym1 ;)
print(page.__pym1__())
```

For a quick parse try the window module...

```
from domonic.window import *
window.location = "http://www.google.com"
print(window.document.title)
```

domonic.html

Generate HTML using python.

```
exception domonic.html.TemplateError(error, message='TemplateError: ')

class domonic.html.a(*args, **kwargs)

class domonic.html.abbr(*args, **kwargs)

class domonic.html.address(*args, **kwargs)

class domonic.html.applet(*args, **kwargs)

class domonic.html.area(*args, **kwargs)

class domonic.html.article(*args, **kwargs)

class domonic.html.aside(*args, **kwargs)

class domonic.html.audio(*args, autoplay: Optional[bool] = None, controls=None, loop=None, muted=None,
                        preload=None, src=None, **kwargs)

class domonic.html.b(*args, **kwargs)

class domonic.html.base(*args, href=None, target=None, **kwargs)

class domonic.html.basefont(*args, **kwargs)

class domonic.html.bdi(*args, **kwargs)

class domonic.html.bdo(*args, **kwargs)
```

```
class domonic.html.blockquote(*args, **kwargs)

class domonic.html.body(*args, aLink=None, background=None, bgColor=None, link=None, onload=None,
                        onunload=None, text=None, vLink=None, **kwargs)

class domonic.html.br(*args, **kwargs)

class domonic.html.button(*args, disabled: Optional[bool] = None, form=None, formaction: Optional[str] =
                        None, formenctype=None, formmethod=None, formnovalidate=None,
                        formtarget=None, name=None, type=None, value=None, **kwargs)

class domonic.html.canvas(*args, width: Optional[int] = None, height: Optional[int] = None, **kwargs)

class domonic.html.caption(*args, **kwargs)

class domonic.html.center(*args, **kwargs)

class domonic.html.cite(*args, **kwargs)

class domonic.html.closed_tag(*args, **kwargs)

class domonic.html.code(*args, **kwargs)

class domonic.html.col(*args, **kwargs)

class domonic.html.colgroup(*args, **kwargs)

class domonic.html.command(*args, **kwargs)

class domonic.html.comment(data)

domonic.html.create_element(name='custom_tag', *args, **kwargs)
    A method for creating custom tags
    tag name needs to be set due to custom tags with hyphens can't be classnames. i.e. hyphenated tags <some-
    custom-tag></some-custom-tag>

class domonic.html.data(*args, **kwargs)

class domonic.html.datalist(*args, **kwargs)

class domonic.html.dd(*args, **kwargs)

class domonic.html.details(*args, **kwargs)

class domonic.html.dfn(*args, **kwargs)

class domonic.html.dialog(*args, open=None, **kwargs)

class domonic.html.div(*args, **kwargs)

class domonic.html.dl(*args, **kwargs)

class domonic.html.doctype(name: str = 'html', publicId: str = "", systemId: str = "")

class domonic.html.dt(*args, **kwargs)

class domonic.html.em(*args, **kwargs)

class domonic.html.embed(*args, **kwargs)
```

```

class domonic.html.fieldset(*args, **kwargs)
class domonic.html.figcaption(*args, **kwargs)
class domonic.html.figure(*args, **kwargs)
class domonic.html.font(*args, **kwargs)
class domonic.html.footer(*args, **kwargs)
class domonic.html.form(*args, **kwargs)
class domonic.html.h1(*args, **kwargs)
class domonic.html.h2(*args, **kwargs)
class domonic.html.h3(*args, **kwargs)
class domonic.html.h4(*args, **kwargs)
class domonic.html.h5(*args, **kwargs)
class domonic.html.h6(*args, **kwargs)
class domonic.html.head(*args, **kwargs)
class domonic.html.header(*args, **kwargs)
class domonic.html.hgroup(*args, **kwargs)
class domonic.html.hr(*args, **kwargs)
class domonic.html.html(*args, **kwargs)
class domonic.html.i(*args, **kwargs)
class domonic.html.iframe(*args, **kwargs)
class domonic.html.img(*args, alt=None, src=None, crossorigin=None, height=None, ismap=None,
                      longdesc=None, sizes=None, srcset=None, usemap=None, width=None, **kwargs)
class domonic.html.input(*args, accept=None, alt=None, autocomplete=None, autofocus=None,
                        checked=None, dirname=None, disabled=None, form=None, formaction=None,
                        formenctype=None, formmethod=None, formnovalidate=None, formtarget=None,
                        height=None, _list=None, _max=None, maxlength=None, _min=None,
                        multiple=None, name=None, pattern=None, placeholder=None, readonly=None,
                        required=None, size=None, src=None, step=None, type=None, value=None,
                        width=None, **kwargs)
class domonic.html.ins(*args, **kwargs)
class domonic.html.isindex(*args, **kwargs)
class domonic.html.kbd(*args, **kwargs)
class domonic.html.keygen(*args, **kwargs)
class domonic.html.label(*args, **kwargs)
class domonic.html.legend(*args, **kwargs)

```

```
class domonic.html.li(*args, **kwargs)
class domonic.html.link(*args, **kwargs)
class domonic.html.listing(*args, **kwargs)
class domonic.html.main(*args, **kwargs)
class domonic.html.mark(*args, **kwargs)
class domonic.html.menu(*args, **kwargs)
class domonic.html.menuitem(*args, **kwargs)
class domonic.html.meta(*args, charset=None, content=None, http_equiv=None, name=None, **kwargs)
class domonic.html.meter(*args, value=None, _min=None, _max=None, low=None, high=None,
                        optimum=None, **kwargs)
class domonic.html.nav(*args, **kwargs)
class domonic.html.noscript(*args, **kwargs)
class domonic.html.ol(*args, **kwargs)
class domonic.html.optgroup(*args, **kwargs)
class domonic.html.option(*args, disabled=None, label=None, selected=None, value=None, **kwargs)
class domonic.html.output(*args, **kwargs)
class domonic.html.p(*args, **kwargs)
class domonic.html.param(*args, **kwargs)
class domonic.html.picture(*args, **kwargs)
class domonic.html.plaintext(*args, **kwargs)
class domonic.html.portal(*args, **kwargs)
class domonic.html.pre(*args, **kwargs)
class domonic.html.progress(*args, **kwargs)
class domonic.html.q(*args, **kwargs)
domonic.html.render(inp, outp="", to=None)
    write the input to string or to a file.
```

Parameters

- **inp** (*obj*) – A domonic tag. For example `div()`
- **outp** (*str*) – An optional output filename
- **to** (*str*) – An optional output type. if ‘pym1’ is specified then pym1 is returned instead of html.

Returns

A HTML rendered string

Return type`str`

```

class domonic.html.rp(*args, **kwargs)
class domonic.html.rt(*args, **kwargs)
class domonic.html.ruby(*args, **kwargs)
class domonic.html.s(*args, **kwargs)
class domonic.html.samp(*args, **kwargs)
class domonic.html.script(*args, **kwargs)
class domonic.html.section(*args, **kwargs)
class domonic.html.select(*args, autofocus: Optional[bool] = None, disabled: Optional[bool] = None,
                           multiple: Optional[bool] = None, name: Optional[str] = None, required:
                           Optional[bool] = None, size: Optional[int] = None, **kwargs)
class domonic.html.small(*args, **kwargs)
class domonic.html.source(*args, **kwargs)
class domonic.html.span(*args, **kwargs)
class domonic.html.strike(*args, **kwargs)
class domonic.html.strong(*args, **kwargs)
class domonic.html.style(*args, **kwargs)
class domonic.html.sub(*args, **kwargs)
class domonic.html.submit(*args, **kwargs)
class domonic.html.summary(*args, **kwargs)
class domonic.html.sup(*args, **kwargs)
class domonic.html.table(*args, align: Optional[str] = None, bgcolor=None, border=None,
                           cellpadding=None, cellspacing=None, frame=None, rules=None, summary=None,
                           width=None, **kwargs)
class domonic.html.tbody(*args, **kwargs)
class domonic.html.td(*args, **kwargs)
class domonic.html.template(*args, **kwargs)
class domonic.html.textarea(*args, autofocus=None, cols=None, disabled=None, form=None,
                             maxlength=None, name=None, placeholder=None, readonly=None,
                             required=None, rows=None, wrap=None, **kwargs)
class domonic.html.tfoot(*args, **kwargs)
class domonic.html.th(*args, **kwargs)
class domonic.html.thead(*args, **kwargs)

```

```
class domonic.html.title(*args, **kwargs)

class domonic.html.tr(*args, **kwargs)

class domonic.html.track(*args, **kwargs)

class domonic.html.u(*args, **kwargs)

class domonic.html.ul(*args, **kwargs)

class domonic.html.var(*args, **kwargs)

class domonic.html.video(*args, autoplay=None, controls=None, height=None, loop=None, muted=None,
                        poster=None, preload=None, src=None, width=None, **kwargs)

class domonic.html.wbr(*args, **kwargs)

class domonic.html.xmp(*args, **kwargs)
```

1.4.2 dom

The DOM is gradually evolving...

To use the DOM either reference your root 'html' node or import the dom modules global 'document'

```
# access the document via the html tag
mydom = html()
# mydom.getElementbyID...

# or by importing the document global
from domonic.dom import document
# document.createElement...
print(document)
```

The last 'html()' created will always be the 'document'. You can set it manually but it needs to be a Document instance. Before a 'html' class is created it is just an empty document so that static methods can be available.

Remember python globals are only to that module (unlike other langs). so you will have to import document explicitly when you need it (per method call after setting it). i.e

```
print(document)
d = html(body("Hello"))
print(document) # no change
print('body1', d.doctype)
print('body2', domonic.dom.document.doctype)
print('body3', document.doctype)
from domonic.dom import document # re-import to get the updated document
print('body4', document.doctype)
```

notice how before it was imported it was still just a class not an instance.

So in most cases just use your own root node and not document as it will be the only one that will be updated unless you import after any html node creation. The global is for when you need to access the document from a different module(file)

createElement

Here's an example of creating your own elements using the DOM API

```
from domonic.dom import *
from domonic.dom import document

site = html()
el = document.createElement('myelement')
site.appendChild(el)
print(site)
```

querySelectorAll

querySelectorAll and querySelector are useful for finding elements in the DOM.

```
mysite.querySelectorAll('button')
mysite.querySelectorAll('.fa-twitter')
mysite.querySelectorAll("a[rel=nofollow]")
mysite.querySelectorAll("a[href='#services']")
mysite.querySelectorAll("a[href$='technology']")

somalinks = mysite.querySelectorAll("a[href*='twitter']")
for l in somalinks:
    print(l.href)
```

To use the DOM either reference your root 'html' node or import the dom modules global 'document'

```
# access the document via the html tag
mydom = html()
# mydom.getElementbyID...

# or by importing the document global
from domonic.dom import document
# document.createElement...
print(document)
```

The last 'html()' you created will always be the 'document'. You can also set it manually but it needs to be a Document instance. Before a 'html' class is created there is an empty document so that static methods can be available.

a note on globals Remember python globals are only to that module (unlike other langs). So you will have to import document explicitly when you need it (per method call)

See the examples folder for other uses of a python virtual DOM.

DOMConfig

DOMConfig is for setting some render options on the dom.

i.e. here's we set all the current flags to be opposite their default...

```
from domonic.html import *
from domonic.dom import DOMConfig
DOMConfig.GLOBAL_AUTOESCAPE = True
DOMConfig.HTMX_ENABLED = True
DOMConfig.RENDER_OPTIONAL_CLOSING_TAGS = False
print(html(head(),body(div(h1('heading'),div(button('hi & hack',_get='/get_hi'))))))
# <html><head></head><body><div><h1>heading</h1><div><button data-hx-get="/get_hi">hi &
↪hack</button></div></div></body></html>
```

The full list of available DOM methods are listed below...

domonic.dom

The DOM represents a document with a logical tree. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

class domonic.dom.Attr(*name: str, value="", *args, **kwargs*)

getNamedItem(*name: str*)

Returns a specified attribute node from a NamedNodeMap

removeNamedItem(*name: str*) → bool

Removes a specified attribute node

setNamedItem(*name: str, value*) → bool

Sets the specified attribute node (by name)

class domonic.dom.CDATASection(*data*)

class domonic.dom.CharacterData(**args, **kwargs*)

The CharacterData abstract interface represents a Node object that contains characters. This is an abstract interface, meaning there aren't any objects of type CharacterData: it is implemented by other interfaces like Text, Comment, or ProcessingInstruction, which aren't abstract.

after(*newChild*)

Inserts a newChild node immediately after this ChildNode.

appendData(*data*)

Appends the given DOMString to the CharacterData.data string; when this method returns, data contains the concatenated DOMString.

before(*newChild*)

Inserts a newChild node immediately before this ChildNode.

deleteData(*offset: int, count: int*)

Removes the specified amount of characters, starting at the specified offset, from the CharacterData.data string; when this method returns, data contains the shortened DOMString.

insertData(*offset: int, data*)

Inserts the specified characters, at the specified offset, in the CharacterData.data string; when this method returns, data contains the modified DOMString.

property nextElementSibling

Returns the next element at the same node tree level

property previousElementSibling

returns the Element immediately prior to the specified one in its parent's children list, or None if the specified element is the first one in the list.

remove()

Removes this ChildNode from the children list of its parent.

replaceData(*offset: int, count: int, data*)

Replaces the specified amount of characters, starting at the specified offset, with the specified DOMString; when this method returns, data contains the modified DOMString.

replaceWith(*newChild*)

Replaces this ChildNode with a new one.

substringData(*offset: int, length: int*)

Returns a DOMString containing the part of CharacterData.data of the specified length and starting at the specified offset.

class domonic.dom.**ChildNode**(**args, **kwargs*)

not tested yet

after(*newChild*)

Inserts a newChild node immediately after this ChildNode.

before(*newChild*)

Inserts a newChild node immediately before this ChildNode.

remove()

Removes this ChildNode from the children list of its parent.

replaceWith(*newChild*)

Replaces this ChildNode with a new one.

class domonic.dom.**Comment**(*data*)

class domonic.dom.**DOMConfig**

DOMConfig: Not to be confused with the obsolete DOMConfiguration.

This class is used to set some global rules for our dom.

GLOBAL_AUTOESCAPE - If this is set to True, then all text nodes will be automatically escaped.
HTML5_MODE - doesn't render close tags on certain elements HTMX_ENABLED - includes htmx attributes into domonic for quicker notation

exception domonic.dom.**DOMException**(*code, message: Optional[str] = None*)

The DOMException interface represents an anormal event related to the DOM.

class domonic.dom.**DOMPoint**(*x: float, y: float, z: float = 0, w: float = 1*)

The DOMPoint interface represents a point specified by x and y coordinates.

class domonic.dom.**DOMPointReadOnly**(*x: float, y: float, z: float = 0, w: float = 1*)

The DOMPointReadOnly interface represents a point specified by x and y coordinates.

class domonic.dom.DOMQuad(*p1, p2, p3, p4*)

The DOMQuad interface represents a quadrilateral on the plane with its four corners represented as Cartesian coordinates.

class domonic.dom.DOMStringMap(**args, **kwargs*)

TODO - not tested yet TODO - make this like a dict

delete(*name: str*) → bool

Deletes the item with the specified name

get(*name: str*)

Returns the value of the item with the specified name

set(*name: str, value*)

Sets the value of the item with the specified name

class domonic.dom.DOMTimeStamp(*value*)

The DOMTimeStamp interface represents a numeric value which represents the number of milliseconds since the epoch.

class domonic.dom.DOMTokenList(*element: Node*)

DOMTokenList represents a set of space-separated tokens.

add(**args*)

Adds the given tokens to the list

contains(*token*) → bool

Returns true if the token is in the list, and false otherwise

item(*index: int*)

Returns the token at the specified index

remove(**args*)

Removes the given tokens from the list

toString() → str

Returns a string containing all tokens in the list, with spaces separating each token

toggle(*token, force=None*)

If force is not given, removes token from list if present, otherwise adds token to list. If force is true, adds token to list, and if force is false, removes token from list if present.

class domonic.dom.Document(**args, **kwargs*)

The Document interface represents the entire HTML or XML document.

property anchors

[get the anchors in the document]

property applets

Returns a collection of all <applet> elements in the document

property body

Returns the <body> element in the document

property characterSet

Returns the character encoding for the document

property charset

Use characterSet instead.

Type

Returns the character encoding for the document. Deprecated

close()

Closes the output stream previously opened with document.open()

static createAttribute(*name*)

Creates an attribute node

static createCDATASection(*data*)

Creates a CDATASection node with the specified data

static createComment(*message*)

Creates a Comment node with the specified text

static createDocumentFragment(args*)**

Creates an empty DocumentFragment node if not content passed. I added args as optional to pass content

static createElement(*_type: str, *args, **kwargs*)

Creates an Element node

static createElementNS(*namespaceURI, qualifiedName, options=None*)

Creates an element with the specified namespace URI and qualified name.

static createEntityReference(*name*)

Creates an EntityReference node with the specified name

static createEvent(*event_type=None*)

[Creates a new event]

Parameters

event_type (*[type]*, *optional*) – [description]. Defaults to None.

Returns

[a new event]

Return type

[*type*]

static createExpression(*xpath, nsResolver*)

Creates an XPathExpression object for the given XPath string.

static createNodeIterator(*root, whatToShow=None, filter=None*)

Creates a NodeIterator that can be used to traverse the document tree or subtree under root.

static createProcessingInstruction(*target, data*)

Creates a ProcessingInstruction node with the specified target and data

static createRange()

Creates a Range

static createTextNode(*text*)

[Creates a Text node with the specified text.

Parameters

text (*[str]*) – [the text to be inserted]

Returns

[a new Text node]

Return type

[*type*]

static createTreeWalker(*root*, *whatToShow=None*, *filter=None*, *entityReferenceExpansion=None*)

[creates a TreeWalker object]

Parameters

- **root** ([*type*]) – [the root node at which to begin traversal]
- **whatToShow** ([*type*], *optional*) – [what types of nodes to show]. Defaults to None.
- **filter** ([*type*], *optional*) – [a NodeFilter or a function to be called for each node]. Defaults to None.

Returns

[a new TreeWalker object]

Return type

[*type*]

property doctype

Returns the Document Type Declaration associated with the document

domConfig()

Returns the DOMConfig which has settings for how html content is rendered

domain()

Returns the domain name of the server that loaded the document

elementFromPoint(*x*, *y*)

Returns the topmost element at the specified coordinates.

elementsFromPoint(*x*, *y*)

Returns an array of all elements at the specified coordinates.

property embeds

[Returns a collection of all <embed> elements the document]

Returns

[a collection of all <embed> elements the document]

Return type

[*type*]

evaluate(*xpathExpression: str*, *contextNode: Optional[Node] = None*, *namespaceResolver=None*, *resultType=7*, *result=None*)

Evaluates an XPath expression and returns the result.

property forms

Returns a collection of all <form> elements in the document

fullscreenElement()

Returns the current element that is displayed in fullscreen mode

fullscreenEnabled()

Returns a Boolean value indicating whether the document can be viewed in fullscreen mode

getElementById(*_id*)

[Returns the element that has the ID attribute with the specified value]

Parameters

_id (*[str]*) – [the value of the ID attribute]

Returns

[the element that has the ID attribute with the specified value]

Return type

[*type*]

getElementsByName(*name: str*)

[Returns a NodeList containing all elements with a specified name]

Parameters

name (*[str]*) – [the name to search for]

Returns

[the matching elements]

Return type

[*type*]

property head: *HTMLHeadElement*

Returns the <head> element of the document

property images

Returns a collection of all elements in the document

property implementation

Returns the DOMImplementation object that handles this document

importNode(*node, deep=False*)

Imports a node from another document to this document.

property links

Returns a collection of all <a> and <area> elements in the document that have a href attribute

normalizeDocument()

Removes empty Text nodes, and joins adjacent nodes

open(*index='index.html'*)

Opens an HTML output stream to collect output from document.write()

property pictureInPictureEnabled

Returns whether Picture-in-Picture mode is enabled.

renameNode(*node, namespaceURI: str, nodename: str*)

[Renames the specified node, and returns the renamed node.]

Parameters

- **node** (*[type]*) – [the node to rename]
- **namespaceURI** (*[type]*) – [a namespace URI]
- **nodename** (*[type]*) – [a node name]

Returns

[description]

Return type`[type]`**property scripts**`[Returns a collection of <script> elements in the document]`**Returns**`[a collection of <script> elements in the document]`**Return type**`[type]`**strictErrorChecking()**`Returns a Boolean value indicating whether to stop on the first error`**property title: `str`**`[gets the title of the document]`**Returns**`The title of the document`**Return type**`[str]`**property visibilityState**`Returns the visibility state of the document`**write(html: `str` = "")**`[writes HTML text to a document]`**Parameters**`html (str, optional) – [the content to write to the document]`**writeln(html: `str` = "")**`[writes HTML text to a document, followed by a line break]`**Parameters**`html (str, optional) – [the content to write to the document]`**property xmlversion**`Returns the version of XML used for the document`**class** `domonic.dom.DocumentFragment(*args)`**property attributes: `NamedNodeMap`**`Returns a NamedNodeMap of an element's attributes`**getElementById(_id)**`[Returns the element that has the ID attribute with the specified value]`**Parameters**`_id ([str]) – [the value of the ID attribute]`**Returns**`[the element that has the ID attribute with the specified value]`**Return type**`[type]`

getElementsByTagName(*tagName: str*) → *HTMLCollection*

[Returns a collection of all child elements with the specified tag name]

Parameters

tagName (*str*) – [a DOMString representing the tag name to match]

Returns

[method returns a live HTMLCollection of elements with the given tag name.]

Return type

[*type*]

querySelector(*query: str*)

[Returns the first child element that matches a specified CSS selector(s) of an element]

Parameters

query (*str*) – [a CSS selector string]

Returns

[an Element object]

Return type

[*type*]

querySelectorAll(*query: str*)

[Returns all child elements that matches a specified CSS selector(s) of an element]

Parameters

query (*str*) – [a CSS selector string]

Returns

[a list of Element objects]

Return type

[*type*]

replaceChildren(*newChildren*) → *None*

Replaces the childNodes of the DocumentFragment object.

class domonic.dom.**DocumentType**(*name: str = 'html', publicId: str = '', systemId: str = ''*)

internalSubset()

A DOMString of the internal subset, or None. Eg “<!ELEMENT foo (bar)>”.

notations() → *NamedNodeMap*

A NamedNodeMap with notations declared in the DTD.

class domonic.dom.**Element**(**args, **kwargs*)

Baseclass for all html tags

append(**args*)

Inserts a set of Node objects or DOMString objects after the last child of the Element.

property attributes: *NamedNodeMap*

Returns a NamedNodeMap of an element’s attributes

blur()

Removes focus from an element

property classList

Returns the value of the classList attribute of an element

property className

Sets or returns the value of the className attribute of an element

click()

Simulates a mouse-click on an element

property clientHeight

Returns the height of an element, including padding

property clientLeft

Returns the width of the left border of an element

property clientTop

Returns the width of the top border of an element

property clientWidth

Returns the width of an element, including padding

property contentEditable: `bool`

Sets or returns whether an element is editable

property dataset

Returns the value of the dataset attribute of an element

property dir

returns the value of the dir attribute of an element

exitFullscreen()

Cancels an element in fullscreen mode

firstElementChild()

Returns the first child element of an element

focus()

Sets focus on an element

getAttribute(*attribute: str*) → `str`

Returns the specified attribute value of an element node

getAttributeNS(*namespaceURI, localName*)

Returns the value of the specified attribute

getAttributeNode(*attribute: str*) → `str`

Returns the specified attribute node

getAttributeNodeNS(*attr*)

Sets the attribute node of an element

getBoundingClientRect()

Returns the size of an element and its position relative to the viewport

getElementsByClassName(*className: str*) → `HTMLCollection`

[Returns a collection of all child elements with the specified class name]

Parameters

className (*str*) – [a DOMString representing the class name to match]

Returns

[a NodeList of all child elements with the specified class name]

Return type`[type]`**getElementsBySelector**(*all_selectors*, *document*)

Get DOM elements based on the given CSS Selector <https://simonwillison.net/2003/Mar/25/getElementsBySelector/> < original author http://www.openjs.com/scripts/dom/css_selector/ < ported to support ‘,’ <https://bin-co.com/python/scripts/getelementsbyselector-html-css-query.php> < ported to py2 (broken/bugs) *BSD LICENSED*

note - always needs a tag in the query i.e. (‘a.classname’) will work. but just (‘.classname’) wont fixed and ported to py3 here. quite cool means other peoples code works on my dom # TODO - needs to work in conjunctions with `_matchElement` so `querySelector` works a bit better and `dQuery` picks it up # TOOD - `*` = node content

Parameters

- **all_selectors** (`[type]`) – [description]
- **document** (`[type]`) – [description]

Returns`[description]`**Return type**`[type]`**getElementsByTagName**(*tagName*: *str*) → *HTMLCollection*

[Returns a collection of all child elements with the specified tag name]

Parameters**tagName** (*str*) – [a DOMString representing the tag name to match]**Returns**

[method returns a live HTMLCollection of elements with the given tag name.]

Return type`[type]`**hasAttribute**(*attribute*: *str*) → *bool*

Returns True if an element has the specified attribute, otherwise False

Parameters**attribute** (*str*) – [the attribute to test for]**Returns**

[True if an element has the specified attribute, otherwise False]

Return type*bool***hasAttributes**() → *bool*

Returns true if an element has any attributes, otherwise false

property id

Sets or returns the value of the id attribute of an element

property innerHTML

Sets or returns the content of an element

insertAdjacentElement(*position: str, element*)

Inserts an element adjacent to the current element

insertAdjacentHTML(*position: str, html: str*)

Inserts raw HTML adjacent to the current element

insertAdjacentText(*position: str, text: str*)

Inserts text adjacent to the current element

isContentEditable() → *bool*

Returns true if the content of an element is editable, otherwise false

lastElementChild()

[Returns the last child element of an element]

Returns

[the last child element of an element]

Return type

[*type*]

matches(*s: str*) → *bool*

[checks to see if the Element would be selected by the provided selectorString]

<https://developer.mozilla.org/en-US/docs/Web/API/Element/matches>

Parameters

s (*str*) – [css selector]

Returns

[True if selector matches Element otherwise False]

Return type

[*bool*]

namespaceURI()

Returns the namespace URI of an element

property nextElementSibling

Returns the next element at the same node tree level

property nextSibling

Returns the next node at the same node tree level

normalize()

Joins adjacent text nodes and removes empty text nodes in an element

offsetHeight()

Returns the height of an element, including padding, border and scrollbar

offsetLeft()

Returns the horizontal offset position of an element

offsetParent()

Returns the offset container of an element

offsetTop()

Returns the vertical offset position of an element

offsetWidth()

Returns the width of an element, including padding, border and scrollbar

property parentElement

Returns the parent element node of an element

prepend(*args)

Prepends a node to the current element

property previousElementSibling

returns the Element immediately prior to the specified one in its parent's children list, or None if the specified element is the first one in the list.

querySelector(query: str)

[Returns the first child element that matches a specified CSS selector(s) of an element]

Parameters

query (str) – [a CSS selector string]

Returns

[an Element object]

Return type

[type]

querySelectorAll(query: str)

[Returns all child elements that matches a specified CSS selector(s) of an element]

Parameters

query (str) – [a CSS selector string]

Returns

[a list of Element objects]

Return type

[type]

remove()

Removes the element from the DOM

removeAttribute(attribute: str)

Removes a specified attribute from an element

removeAttributeNS(namespaceURI, localName)

Removes an attribute from an element

removeAttributeNode(attribute)

Removes a specified attribute node, and returns the removed node

requestFullscreen()

Shows an element in fullscreen mode

scrollHeight()

Returns the entire height of an element, including padding

scrollIntoView()

Scrolls the specified element into the visible area of the browser window

scrollLeft()

Sets or returns the number of pixels an element's content is scrolled horizontally

scrollTop()

Sets or returns the number of pixels an element's content is scrolled vertically

scrollWidth()

Returns the entire width of an element, including padding

setAttribute(*attribute*, *value*)

Sets or changes the specified attribute, to the specified value

setAttributeNS(*namespaceURI*, *localName*, *value*)

Sets an attribute in the given namespace

setAttributeNode(*attr*)

[Sets or changes the specified attribute node]

Parameters

attr ([*type*]) – [an Attr object]

setAttributeNodeNS(*attr*)

Sets the attribute node of an element

property style

returns the value of the style attribute of an element

property title

returns the value of the title attribute of an element

toString() → str

Converts an element to a string

class domonic.dom.Entity(*args)**static fromChar(*char*: str) → str**

Returns the character corresponding to the given entity name.

static fromName(*entityName*: str) → str

Returns the entity name corresponding to the given character.

class domonic.dom.EntityReference(*args)

The EntityReference interface represents a reference to an entity, either parsed or unparsed, in an Entity Node. Note that this is not a CharacterData node, and does not have any child nodes.

static fromOrdinal(*ordinal*: int)

Returns the entity name corresponding to the given character.

static ordinal(*entityName*: str)

Returns the character corresponding to the given entity name.

class domonic.dom.HTMLAnchorElement(*args, **kwargs)**class domonic.dom.HTMLAreaElement(*args, **kwargs)****class domonic.dom.HTMLAudioElement(*args, autoplay: Optional[bool] = None, controls=None, loop=None, muted=None, preload=None, src=None, **kwargs)****class domonic.dom.HTMLBRElement(*args, **kwargs)****class domonic.dom.HTMLBaseElement(*args, href=None, target=None, **kwargs)**

```

class domonic.dom.HTMLBaseFontElement(*args, **kwargs)

class domonic.dom.HTMLBodyElement(*args, aLink=None, background=None, bgColor=None, link=None,
                                   onload=None, onunload=None, text=None, vLink=None, **kwargs)

class domonic.dom.HTMLButtonElement(*args, disabled: Optional[bool] = None, form=None, formaction:
                                     Optional[str] = None, formenctype=None, formmethod=None,
                                     formnovalidate=None, formtarget=None, name=None, type=None,
                                     value=None, **kwargs)

class domonic.dom.HTMLCanvasElement(*args, width: Optional[int] = None, height: Optional[int] = None,
                                     **kwargs)

class domonic.dom.HTMLCollection(iterable=(), /)

    item(index: int)
        [gets the indexth item in the collection. If index is greater than or equal to the number of nodes in the list,
         this returns null.]

        Parameters
            index ([type]) – [the index of the item to return.]

        Returns
            [the node at the indexth position, or None]

        Return type
            [type]

    namedItem(name: str)
        Returns the specific node whose ID or, as a fallback, name matches the string specified by name.

class domonic.dom.HTMLContentElement(*args, **kwargs)

class domonic.dom.HTMLDListElement(*args, **kwargs)

class domonic.dom.HTMLDataElement(*args, **kwargs)

class domonic.dom.HTMLDataListElement(*args, **kwargs)

class domonic.dom.HTMLDialogElement(*args, open=None, **kwargs)

class domonic.dom.HTMLDivElement(*args, **kwargs)

class domonic.dom.HTMLDocument(*args, **kwargs)

class domonic.dom.HTMLElement(*args, **kwargs)

class domonic.dom.HTMLEmbedElement(*args, **kwargs)

class domonic.dom.HTMLFieldSetElement(*args, **kwargs)

class domonic.dom.HTMLFormControlsCollection(*args, **kwargs)

class domonic.dom.HTMLFormElement(*args, action: Optional[str] = None, autocomplete=None, enctype:
                                   Optional[str] = None, method: Optional[str] = None, name:
                                   Optional[str] = None, novalidate: Optional[bool] = None, target=None,
                                   **kwargs)

class domonic.dom.HTMLFrameSetElement(*args, **kwargs)

```

```
class domonic.dom.HTMLHRElement(*args, **kwargs)

class domonic.dom.HTMLHeadElement(*args, **kwargs)

class domonic.dom.HTMLHeadingElement(*args, **kwargs)

class domonic.dom.HTMLIFrameElement(*args, src=None, name=None, sandbox=None,
                                     allowfullscreen=None, **kwargs)

class domonic.dom.HTMLImageElement(*args, alt=None, src=None, crossorigin=None, height=None,
                                    ismap=None, longdesc=None, sizes=None, srcset=None,
                                    usemap=None, width=None, **kwargs)

class domonic.dom.HTMLInputElement(*args, accept=None, alt=None, autocomplete=None, autofocus=None,
                                    checked=None, dirname=None, disabled=None, form=None,
                                    formaction=None, formenctype=None, formmethod=None,
                                    formnovalidate=None, formtarget=None, height=None, _list=None,
                                    _max=None, maxlength=None, _min=None, multiple=None,
                                    name=None, pattern=None, placeholder=None, readonly=None,
                                    required=None, size=None, src=None, step=None, type=None,
                                    value=None, width=None, **kwargs)

class domonic.dom.HTMLIsIndexElement(*args, **kwargs)

class domonic.dom.HTMLKeygenElement(*args, **kwargs)

class domonic.dom.HTMLLIElement(*args, **kwargs)

class domonic.dom.HTMLLabelElement(*args, **kwargs)

class domonic.dom.HTMLLegendElement(*args, **kwargs)

class domonic.dom.HTMLLinkElement(*args, **kwargs)

class domonic.dom.HTMLMapElement(*args, **kwargs)

class domonic.dom.HTMLMediaElement(*args, **kwargs)

class domonic.dom.HTMLMetaElement(*args, charset=None, content=None, http_equiv=None, name=None,
                                   **kwargs)

class domonic.dom.HTMLMeterElement(*args, value=None, _min=None, _max=None, low=None, high=None,
                                    optimum=None, **kwargs)

class domonic.dom.HTMLModElement(*args, **kwargs)

class domonic.dom.HTMLOLListElement(*args, **kwargs)

class domonic.dom.HTMLObjectElement(*args, **kwargs)

class domonic.dom.HTMLOptGroupElement(*args, **kwargs)

class domonic.dom.HTMLOptionElement(*args, disabled=None, label=None, selected=None, value=None,
                                    **kwargs)

class domonic.dom.HTMLOutputElement(*args, **kwargs)

class domonic.dom.HTMLParagraphElement(*args, **kwargs)
```

```

class domonic.dom.HTMLParamElement(*args, **kwargs)
class domonic.dom.HTMLPictureElement(*args, **kwargs)
class domonic.dom.HTMLPortalElement(*args, **kwargs)
class domonic.dom.HTMLPreElement(*args, **kwargs)
class domonic.dom.HTMLProgressElement(*args, **kwargs)
class domonic.dom.HTMLQuoteElement(*args, **kwargs)
class domonic.dom.HTMLScriptElement(*args, **kwargs)
class domonic.dom.HTMLSelectElement(*args, autofocus: Optional[bool] = None, disabled: Optional[bool] = None, multiple: Optional[bool] = None, name: Optional[str] = None, required: Optional[bool] = None, size: Optional[int] = None, **kwargs)

class domonic.dom.HTMLShadowElement(*args, **kwargs)
class domonic.dom.HTMLSourceElement(*args, **kwargs)
class domonic.dom.HTMLSpanElement(*args, **kwargs)
class domonic.dom.HTMLStyleElement(*args, **kwargs)
class domonic.dom.HTMLTableCaptionElement(*args, **kwargs)
class domonic.dom.HTMLTableCellElement(*args, **kwargs)
class domonic.dom.HTMLTableColElement(*args, **kwargs)
class domonic.dom.HTMLTableDataCellElement(*args, **kwargs)
class domonic.dom.HTMLTableElement(*args, align: Optional[str] = None, bgcolor=None, border=None, cellpadding=None, cellspacing=None, frame=None, rules=None, summary=None, width=None, **kwargs)

class domonic.dom.HTMLTableHeaderCellElement(*args, **kwargs)
class domonic.dom.HTMLTableRowElement(*args, **kwargs)
class domonic.dom.HTMLTableSectionElement(*args, **kwargs)
class domonic.dom.HTMLTemplateElement(*args, **kwargs)
class domonic.dom.HTMLTextAreaElement(*args, autofocus=None, cols=None, disabled=None, form=None, maxlength=None, name=None, placeholder=None, readonly=None, required=None, rows=None, wrap=None, **kwargs)

class domonic.dom.HTMLTimeElement(*args, **kwargs)
class domonic.dom.HTMLTitleElement(*args, **kwargs)
class domonic.dom.HTMLTrackElement(*args, **kwargs)
class domonic.dom.HTMLULListElement(*args, **kwargs)

```

```
class domonic.dom.HTMLUnknownElement(*args, **kwargs)
```

```
class domonic.dom.HTMLVideoElement(*args, autoplay=None, controls=None, height=None, loop=None,
                                     muted=None, poster=None, preload=None, src=None, width=None,
                                     **kwargs)
```

```
class domonic.dom.Node(*args, **kwargs)
```

An abstract base class upon which many other DOM API objects are based

appendChild(*aChild*: [Node](#)) → [Node](#)

Adds a child to the current element. If item is a DocumentFragment, all its children are added.

Parameters

item ([Node](#)) – The Node to add.

property attrib

Returns the attributes of the current node as a dict not a NamedNodeMap

property childElementCount: [int](#)

Returns the number of child elements an element has

property childNodes: [NodeList](#)

Returns a live NodeList containing all the children of this node

property children

Returns a collection of an element's child element (excluding text and comment nodes)

cloneNode(*deep*: [bool](#) = [True](#))

Returns a copy.

compareDocumentPosition(*otherElement*) → [int](#)

An integer value representing otherNode's position relative to node as a bitmask combining the following constant properties of Node:

<https://stackoverflow.com/questions/8334286/cross-browser-compare-document-position>

contains(*node*)

Check whether a node is a descendant of a given node

property firstChild

Returns the first child node of an element

hasChildNodes() → [bool](#)

Returns true if an element has any child nodes, otherwise false

insertBefore(*new_node*, *reference_node*=None)

inserts a node before a reference node as a child of a specified parent node. this will remove the node from its previous parent node, if any.

TODO - can throw value error if wrong ordered params. may be helpful to catch to say so.

isDefaultNamespace(*ns*)

Checks if a namespace is the default namespace

isEqualNode(*node*)

Checks if two elements are equal

isSameNode(*node*)

Checks if two elements are the same node

iter(*tag=None*)

Creates a tree iterator with the current element as the root. The iterator iterates over this element and all elements below it, in document (depth first) order. If tag is not None or '*', only elements whose tag equals tag are returned from the iterator. If the tree structure is modified during iteration, the result is undefined.

property lastChild

Returns the last child node of an element

lookupNamespaceURI(*ns: str*)

Returns the namespace URI for a given prefix

Parameters

ns – prefix - i.e 'xml', 'xlink', 'svg', etc

lookupPrefix(*ns*)

Returns the prefix for a given namespace URI

property nextSibling

[returns the next sibling of the current node.]

property nodeName

Returns the name of a node

property nodeValue

Sets or returns the value of a node

normalize()

Normalize a node's value

property ownerDocument

Returns the root element (document object) for an element

property previousSibling

[returns the previous sibling of the current node.]

removeChild(*node*)

removes a child node from the DOM and returns the removed node.

replaceChild(*newChild, oldChild*)

[Replaces a child node within the given (parent) node.]

Parameters

- **newChild** (*[type]*) – [a Node object]
- **oldChild** (*[type]*) – [a Node object]

Returns

[the old child node]

Return type

[type]

property rootNode

[read-only property returns a Node object representing the topmost node in the tree, or the current node if it's the topmost node in the tree]

Returns

[the topmost Node in the tree]

Return type*[Node]***property tag**

Returns the tag name of the current node

property tail

Returns the text content of the current node

property text

Returns the text content of the current node

property textContent

Returns the text content of a node and its descendants

```
class domonic.dom.NodeIterator(root, whatToShow=4294967295, filter=None,
                                entityReferenceExpansion=False)
```

*[NodeIterator is an iterator object that iterates over the descendants of a node, in tree order.]***nextNode()**

Returns the next Node in the document, or null if there are none.

pointerBeforeReferenceNode() → bool

Returns a boolean flag that indicates whether the NodeIterator is anchored before, the flag being true, or after, the flag being false, the anchor node.

previousNode()

Returns the previous Node in the document, or null if there are none.

referenceNode() → Node

Returns the Node that is being iterated over.

```
class domonic.dom.NodeList(iterable=(), /)
```

NodeList objects are collections of nodes

entries() → Iterable[Tuple[int, Node]]

Returns an iterator, allowing code to go through all key/value pairs contained in the collection. (In this case, the keys are numbers starting from 0 and the values are nodes.)

forEach(func, thisArg=None) → None

Calls a function for each item in the NodeList.

item(index) → Node

Returns an item in the list by its index, or null if the index is out-of-bounds.

keys() → Iterable[int]

Returns an iterator, allowing code to go through all the keys of the key/value pairs contained in the collection. (In this case, the keys are numbers starting from 0.)

values() → Iterable[Node]

Returns an iterator allowing code to go through all values (nodes) of the key/value pairs contained in the collection.

```
class domonic.dom.ParentNode(*args, **kwargs)
```

not tested yet

property children: NodeList

Return list of child nodes.

property firstElementChild

First Element child node.

property lastElementChild

Last Element child node.

class domonic.dom.**ProcessingInstruction**(*target*, *data*)

class domonic.dom.**RadioNodeList**(*name*: *str*)

property value

Returns the value of the first element in the collection, or null if there are no elements in the collection.

class domonic.dom.**ShadowRoot**(*host*, *mode*='open')

property on element that has hidden DOM

caretPositionFromPoint()

Returns a CaretPosition object containing the DOM node containing the caret, and caret's character offset within that node.

elementFromPoint(*x*, *y*)

Returns the topmost element at the specified coordinates.

elementsFromPoint(*x*, *y*)

Returns an array of all elements at the specified coordinates.

getSelection()

Returns a Selection object representing the range of text selected by the user, or the current position of the caret.

class domonic.dom.**Text**(**args*, ***kwargs*)

Text Node

property assignedSlot

Returns the slot whose assignedNodes contains this node.

property childNodes

Returns a live NodeList containing all the children of this node

property firstChild

Returns the first child node of an element

property nodeName

Returns the name of a node

splitText(*offset*: *int*)

Splits the Text node into two Text nodes at the specified offset, keeping both in the tree as siblings. The first node is returned, while the second node is discarded and exists outside the tree.

property wholeText

Returns a DOMString containing all the text content of the node and its descendants.

class domonic.dom.**TreeWalker**(*node*, *whatToShow*=4294967295, *_filter*=None, *expandEntityReferences*=False)

The TreeWalker object represents the nodes of a document subtree and a position within them.

firstChild()

Moves the current Node to the first visible child of the current node, and returns the found child. It also moves the current node to this child. If no such child exists, returns null and the current node is not changed.

lastChild()

Moves the current Node to the last visible child of the current node, and returns the found child. It also moves the current node to this child. If no such child exists, null is returned and the current node is not changed.

nextNode()

Moves the current Node to the next visible node in the document order, and returns the found node. It also moves the current node to this one. If no such node exists, returns None and the current node is not changed. can be used in a while loop to iterate over all the nodes in the document order.

nextSibling()

Moves the current Node to its next sibling, if any, and returns the found sibling. If there is no such node, null is returned and the current node is not changed.

parentNode()

Moves the current Node to the first visible ancestor node in the document order, and returns the found node. It also moves the current node to this one. If no such node exists, or if it is before that the root node defined at the object construction, returns null and the current node is not changed.

previousNode()

Moves the current Node to the previous visible node in the document order, and returns the found node. It also moves the current node to this one. If no such node exists, or if it is before that the root node defined at the object construction, returns null and the current node is not changed.

previousSibling()

Moves the current Node to its previous sibling, if any, and returns the found sibling. If there is no such node, return null and the current node is not changed.

property root

Returns a Node representing the root node as specified when the TreeWalker was created.

tree

Is a boolean value indicating, when discarding an entity reference its whole sub-tree must be discarded at the same time.

whatToShow(*options*)

Returns an unsigned long being a bitmask made of constants describing the types of Node that must be presented. Non-matching nodes are skipped, but their children may be included, if relevant. The possible values are:

`domonic.dom.console`

```
# self.screen = type('screen', (DOM,), {'name':'screen'})
```

domonic.events

<https://developer.mozilla.org/en-US/docs/Web/Events>

```
class domonic.events.AnimationEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ANIMATIONEND: str = 'animationend'

    ANIMATIONITERATION: str = 'animationiteration'

    ANIMATIONSTART: str = 'animationstart'

    animationName
        Returns the name of the animation

    elapsedTime
        Returns the number of seconds an animation has been running

    pseudoElement
        Returns the name of the pseudo-element of the animation

class domonic.events.BeforeUnloadEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    BEFOREUNLOAD: str = 'beforeunload'

    BeforeUnloadEvent

class domonic.events.ClipboardEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    COPY: str = 'copy'

    CUT: str = 'cut'

    PASTE: str = 'paste'

    clipboardData
        Returns an object containing the data affected by the clipboard operation

class domonic.events.CompositionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

class domonic.events.CustomEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

class domonic.events.DOMContentLoadedEvent(_type: str, options: Optional[dict] = None, *args,
                                             **kwargs)

    DOMCONTENTLOADED: str = 'DOMContentLoaded'

    document
        Returns the document that was loaded

class domonic.events.DragEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    DRAG: str = 'drag'

    DROP: str = 'drop'

    END: str = 'dragend'

    ENTER: str = 'dragenter'
```

EXIT: `str` = 'dragexit'

LEAVE: `str` = 'dragleave'

OVER: `str` = 'dragover'

START: `str` = 'dragstart'

`dataTransfer`

Returns the data that is dragged/dropped

```
class domonic.events.ErrorEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

ERROR: `str` = 'error'

```
class domonic.events.Event(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

event

ABORT: `str` = 'abort'

AFTERPRINT: `str` = 'afterprint'

BEFOREPRINT: `str` = 'beforeprint'

BEFOREUNLOAD: `str` = 'beforeunload'

CANPLAY: `str` = 'canplay'

CANPLAYTHROUGH: `str` = 'canplaythrough'

CHANGE: `str` = 'change'

DURATIONCHANGE: `str` = 'durationchange'

EMPTIED: `str` = 'emptied'

ENDED: `str` = 'ended'

ERROR: `str` = 'error'

FULLSCREENCHANGE: `str` = 'fullscreenchange'

FULLSCREENERROR: `str` = 'fullscreenerror'

INPUT: `str` = 'input'

INVALID: `str` = 'invalid'

LOAD: `str` = 'load'

LOADEDDATA: `str` = 'loadeddata'

LOADEDMETADATA: `str` = 'loadedmetadata'

MESSAGE: `str` = 'message'

OFFLINE: `str` = 'offline'

ONLINE: `str` = 'online'

OPEN: `str` = 'open'

```

PAUSE: str = 'pause'
PLAY: str = 'play'
PLAYING: str = 'playing'
PROGRESS: str = 'progress'
RATECHANGE: str = 'ratechange'
READYSTATECHANGE: str = 'readystatechange'
RESET: str = 'reset'
RESIZE: str = 'resize'
SCROLL: str = 'scroll'
SEARCH: str = 'search'
SEEKED: str = 'seeked'
SEEKING: str = 'seeking'
SELECT: str = 'select'
SHOW: str = 'show'
STALLED: str = 'stalled'
SUBMIT: str = 'submit'
SUSPEND: str = 'suspend'
TOGGLE: str = 'toggle'
UNLOAD: str = 'unload'
VOLUMECHANGE: str = 'volumechange'
WAITING: str = 'waiting'

```

```
stopPropagation()
```

[prevents further propagation of the current event in the capturing and bubbling phases]

`domonic.events.EventDispatcher`

legacy alias

```
class domonic.events.EventTarget(*args, **kwargs)
```

EventTarget is a class you can extend to give your obj event dispatching abilities

```
class domonic.events.ExtendableEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    extendable
```

Returns whether the event is extendable or not

```
    timeStamp: float
```

Returns the time stamp of the event

```
class domonic.events.FetchEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

FETCH: `str` = 'fetch'

clientId

Returns the client ID of the fetch request

property isReload

Returns whether the request is a reload or not

request

Returns the request object

respondWith(*response*)

Returns a promise that resolves to the response object

waitUntil(*promise*)

Returns a promise that resolves when the response is available

class domonic.events.**FocusEvent**(*_type: str, options: Optional[dict] = None, *args, **kwargs*)

BLUR: `str` = 'blur'

FOCUS: `str` = 'focus'

FOCUSIN: `str` = 'focusin'

FOCUSOUT: `str` = 'focusout'

class domonic.events.**GamePadEvent**(*_type: str, options: Optional[dict] = None, *args, **kwargs*)

START: `str` = 'gamepadconnected'

STOP: `str` = 'gamepaddisconnected'

class domonic.events.**HashChangeEvent**(*_type: str, options: Optional[dict] = None, *args, **kwargs*)

CHANGE: `str` = 'hashchange'

class domonic.events.**InputEvent**(*_type: str, options: Optional[dict] = None, *args, **kwargs*)

CHANGE: `str` = 'change'

INPUT: `str` = 'input'

SELECT: `str` = 'select'

data

Returns the inserted characters

dataTransfer

Returns an object containing information about the inserted/deleted data

getTargetRanges()

Returns an array containing target ranges that will be affected by the insertion/deletion

inputType

Returns the type of the change (i.e “inserting” or “deleting”)

isComposing

Returns whether the state of the event is composing or not


```

class domonic.events.KeyboardEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
    keyboard events

    DOM_KEY_LOCATION_JOYSTICK: int = 5

    DOM_KEY_LOCATION_LEFT: int = 0

    DOM_KEY_LOCATION_MOBILE: int = 4

    DOM_KEY_LOCATION_NUMPAD: int = 3

    DOM_KEY_LOCATION_RIGHT: int = 2

    DOM_KEY_LOCATION_STANDARD: int = 1

    KEYDOWN: str = 'keydown'

    KEYPRESS: str = 'keypress'

    KEYUP: str = 'keyup'

class domonic.events.MessageEvent(_type, options: Optional[dict] = None, *args, **kwargs)

    CONNECT: str = 'connect'

    DISCONNECT: str = 'disconnect'

    MESSAGE: str = 'message'

    data
        Returns the data of the message

    lastEventId
        Returns the last event id of the message

    origin
        Returns the origin of the message

    ports
        Returns the ports of the message

    source
        Returns the source of the message

class domonic.events.MouseEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
    mouse events

    CLICK: str = 'click'

    CONTEXTMENU: str = 'contextmenu'

    DBLCLICK: str = 'dblclick'

    MOUSEDOWN: str = 'mousedown'

    MOUSEENTER: str = 'mouseenter'

    MOUSELEAVE: str = 'mouseleave'

    MOUSEMOVE: str = 'mousemove'

```

MOUSEOUT: `str` = 'mouseout'

MOUSEOVER: `str` = 'mouseover'

MOUSEUP: `str` = 'mouseup'

`getModifierState()`

Returns an array containing target ranges that will be affected by the insertion/deletion

`class domonic.events.PageTransitionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

PAGEHIDE: `str` = 'pagehide'

PAGESHOW: `str` = 'pageshow'

`persisted`

Returns whether the webpage was cached by the browser

`class domonic.events.PointerEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

POINTER: `str` = 'pointer'

POINTERCANCEL: `str` = 'pointercancel'

POINTERDOWN: `str` = 'pointerdown'

POINTERENTER: `str` = 'pointerenter'

POINTERLEAVE: `str` = 'pointerleave'

POINTERMOVE: `str` = 'pointermove'

POINTEROUT: `str` = 'pointerout'

POINTEROVER: `str` = 'pointerover'

POINTERUP: `str` = 'pointerup'

`class domonic.events.PopStateEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

POPSTATE: `str` = 'popstate'

`state`

Returns an object containing a copy of the history entries

`class domonic.events.ProgressEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

ABORT: `str` = 'abort'

ERROR: `str` = 'error'

LOAD: `str` = 'load'

LOADED: `str` = 'loaded'

LOADEND: `str` = 'loadend'

LOADSTART: `str` = 'loadstart'

PROGRESS: `str` = 'progress'

```

    TIMEOUT: str = 'timeout'

class domonic.events.PromiseRejectionEvent(_type, options=None, *args, **kwargs)

    HANDLED: str = 'rejectionhandled'

    UNHANDLED: str = 'unhandledrejection'

    isRejected
        Returns whether the promise was rejected or not

    promise
        Returns the promise that was rejected

    reason
        Returns the reason of the rejection

class domonic.events.SVGEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ABORT: str = 'abort'

    ERROR: str = 'error'

    LOAD: str = 'load'

    LOADEDDEDATA: str = 'loadeddata'

    LOADEDMETADATA: str = 'loadedmetadata'

    LOADSTART: str = 'loadstart'

    PROGRESS: str = 'progress'

    SCROLL: str = 'scroll'

    UNLOAD: str = 'unload'

class domonic.events.SecurityPolicyViolationEvent(_type: str, options: Optional[dict] = None, *args,
                                                    **kwargs)

    SECURITY_POLICY_VIOLATION: str = 'securitypolicyviolation'

    blockedURI
        Returns the blocked URI

    frame
        Returns the frame that violated the policy

    isFrameAncestor
        Returns whether the frame is an ancestor of the frame that violated the policy

    isMainFrame
        Returns whether the frame is the main frame

    originalPolicy
        Returns the original policy

    timeStamp: float
        Returns the time stamp of the event

```

violatedDirective

Returns the violated directive

```
class domonic.events.StorageEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

STORAGE: str = 'storage'

key

Returns the key of the changed storage item

newValue

Returns the new value of the changed storage item

oldValue

Returns the old value of the changed storage item

storageArea

Returns an object representing the affected storage object

url

Returns the URL of the changed item's document

```
class domonic.events.SubmitEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

SUBMIT: str = 'submit'

```
class domonic.events.SyncEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

SYNC: str = 'sync'

lastChance

Returns whether the sync event is the last chance or not

tag

Returns the tag of the sync event

timeStamp: float

Returns the time stamp of the event

```
class domonic.events.TimerEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

TIMER: str = 'timer'

TIMER_COMPLETE: str = 'timercomplete'

TimerEvent

```
class domonic.events.TouchEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

TOUCHCANCEL: str = 'touchcancel'

TOUCHEND: str = 'touchend'

TOUCHMOVE: str = 'touchmove'

TOUCHSTART: str = 'touchstart'

```
class domonic.events.TransitionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

TRANSITIONEND: str = 'transitionend'

elapsedTime

Returns the number of seconds a transition has been running

propertyName

Returns the name of the transition

pseudoElement

Returns the name of the pseudo-element of the transition

```
class domonic.events.TweenEvent(_type, source=None, bubbles=False, cancelable=False)
```

```
    COMPLETE: str = 'onComplete'
```

```
    PAUSE: str = 'onPause'
```

```
    RESET: str = 'onReset'
```

```
    START: str = 'onStart'
```

```
    STOP: str = 'onStop'
```

```
    TIMER: str = 'onTimer'
```

```
    UNPAUSE: str = 'onUnPause'
```

```
    UPDATE_END: str = 'onUpdateEnd'
```

```
    UPDATE_START: str = 'onUpdateStart'
```

```
class domonic.events.UIEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
class domonic.events.WheelEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    WHEEL: str = 'wheel'
```

1.4.3 javascript

There's a javascript package that mimics the js API.

It's useful for things like quickly porting javascript code to python but also if you already know javascript:

```
from domonic.javascript import Math
print(Math.random())

from domonic.javascript import Array
myArr=Array(1,2,3)
print(myArr.splice(1))

from domonic.javascript import URL
url = URL('https://somesite.com/blog/article-one#some-hash')
print(url.protocol)
print(url.host)
print(url.pathname)
print(url.hash)

# from domonic.javascript import Global
# Global.decodeURIComponent(...
```

(continues on next page)

(continued from previous page)

```
# Global.encodeComponent(...  
  
# from domonic.javascript import Date, String, Number  
# etc..
```

As as all of the usual String and Numbers methods you may be familiar with.

Date class

The Date class is available...

```
from domonic.javascript import Date  
print(Date.now())
```

Array methods

All the javascript array methods you may be familiar with available python

```
myarr = Array("1", "2", 3, {"4": "four"}, 5, [6])  
  
print(myarr.length)  
print(myarr.includes("1"))  
print(myarr.includes(3))  
print(myarr.includes(10))  
print(myarr.indexOf(10))  
print(myarr.indexOf("1"))  
print(myarr.indexOf([6]))  
print(myarr[1])  
print(len(myarr))  
print(myarr.join('---')) # TODO - test some js ones  
print(myarr.lastIndexOf("1"))  
print(myarr.lastIndexOf(3))  
print(myarr.reverse())  
print(myarr.slice(0, 1))  
print(myarr.splice(1))  
# print(myarr.splice(2))  
# print(myarr.splice(3))  
# print(myarr.splice(4))  
print(myarr.splice(3, 3, "a", "b", "c"))  
print(myarr)  
print(myarr.pop())  
print(myarr)  
myarr.push(7)  
print(myarr)  
print(myarr.unshift('z'))  
print(myarr)  
print(myarr.shift())  
print(myarr)  
# print(myarr.concat())
```

(continues on next page)

(continued from previous page)

```
# myarr.sort()
# myarr.fill()
# myarr.isArray()?
# myarr.map()
# myarr.reduce()
# myarr.reduceRight()
# myarr.some()
```

String methods

A whole bunch of familiar string methods for you to enjoy...

```
mystr = String("Some String")

mystr.toLowerCase() # "some string"
mystr.toUpperCase() # "SOME STRING"
# print(mystr.length)
mystr.repeat(2) # "Some StringSome String"
print(mystr.startsWith('S'))
# mystr.endsWith('g'))

# javascript substr in python
mystr.substr(1) # 'ome String'

# javascript slice in python
# print(mystr.slice(1, 3))
mystr.slice(1, 3) # 'om'

# trim
mystr = String("  Some String  ")
mystr.trim() # "Some String"

# charAt
mystr = String("Some String")
mystr.charAt(1) # 'o'
mystr.charAt(5) # 'S'

# charCodeAt
mystr.charCodeAt(1) # 111
mystr.fromCharCode(111) # 'o'

# test
# mystr.test('a') # True
# mystr.test('b') # False

# replace
# print(mystr.replace('S', 'X'))
mystr.replace('S', 'X') # "Xome String"
mystr.replace(' ', 'X') # "SomeXString"
mystr.replace('S', 'X') != "Xome Xtring"
```

(continues on next page)

(continued from previous page)

```

# search
mystr = String("Some String")
mystr.search('a') # False
mystr.search('o') # True

# substr
print(mystr.substr(1, 2))
mystr.substr(1, 2) # 'om'
mystr.substr(1, 3) # 'ome'
mystr.substr(1, 4) # 'ome '
mystr.substr(1, 5) # 'ome S'

# toLocaleLowerCase
mystr.toLocaleLowerCase() # 'some string'
mystr.toLocaleLowerCase() # 'some string'

# toLocaleUpperCase
# print(mystr.toLocaleUpperCase())
mystr.toLocaleUpperCase() # 'SOME STRING'

# lastIndex
# print(mystr.lastIndexOf('o'))
mystr.lastIndexOf('o') # 1

assert mystr.padEnd(13) # "Some String  "
assert mystr.padStart(13) # "  Some String"
assert mystr.padStart(13, '-') # "--Some String"

mystr.includes('a') # False
mystr.includes('Some') # True

```

Plus some obsolete friends to do fancy tricks with!!... i.e

```

>>> test = String("Hello Wolrd!")
>>> test.blink()
>>> test.sub()
>>> test.sup()
>>> test.div() # ?? hang on?
>>> test.webpage() # ??? err... wait what!!!

```

You can actually transform a type String into any tag.

Simply call `()` on a stringvar to transform it into a Node

```

>>> test = String("time to take a mo")
>>> test('div', _style="font-color:red;")
>>> str(test('div', _style="font-color:red;"))

```

passing it the tag and attributes...

Object methods

Object is useful for making dicts a bit more js-like...

```
o = Object()
o.prop = 'hi'
str(o)
```

But also contains a growing list of methods you may know from javascript.

setInterval

You can use setInterval and clearInterval with params

```
x=0

def hi(inc):
    global x
    x = x+inc
    print(x)

test = window.setInterval(hi, 1000, 2)
import time
time.sleep(5)
window.clearInterval(test)
print(f"Final value of x:{x}")
```

fetch

There's a fetch implementation that uses promises. With additional multithreaded and pooled versions.

```
from domonic.javascript import *

urls = ['http://google.com', 'http://linkedin.com', 'http://eventual.technology'] # use_
↳ your own domains

print('run 1')
results = window.fetch(urls[0])
results.then(lambda r: print(r.text))
print('run 1 FINISHED')

def somefunc(response):
    print("I'm a callback", response.ok)
    return response

mydata = window.fetch(urls[0]).then(somefunc)
print(mydata)
print(mydata.data)
print(mydata.data.text)

# fetch more than one
results = window.fetch_set(urls)
```

(continues on next page)

(continued from previous page)

```
print(results)
print(list(results))
for r in results:
    if r is not None:
        print(r.ok)
        # print(r.text)

# multi-threaded
results = window.fetch_threaded(urls)
print(results)
print(list(results))
for r in results:
    if r is not None:
        print(r.ok)
        # print(r.text)

# pooled
results = window.fetch_pooled(urls, timeout=2)
print(results)
for r in results:
    if r is not None:
        print(r.ok)
        # print(r.text)

print('run 4')
results = window.fetch(urls[0])
print(results)
results.then(lambda r: print(r.text) if r is not None else None)
```

All fetch methods use requests and will pass all the kwargs along should you need to modify

keywords

If you ``import *`` you will get the js keywords... i.e.

```
print(true) # True
print(false) # False
print(undefined) # None
print(null) # None
```

As well as a function which evaluates python strings...

```
sup = function('`print("hi")`')
sup()
```

Typed arrays

js style typed arrays are even available

Styling

Styling gets passed to the style tag on render.

```
mytag = div("hi", _id="test")
mytag.style.backgroundColor = "black"
mytag.style.fontSize = "12px"
print(mytag)
# <div id="test" style="background-color:black;font-size:12px;">hi</div>
```

Many other undocumented features. Take a look at the code.

domonic.javascript

- https://www.w3schools.com/jsref/jsref_reference.asp
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

class domonic.javascript.**Array**(*args)

javascript array

at(index: *int*)

[takes an integer value and returns the item at that index, allowing for positive and negative integers. Negative integers count back from the last item in the array.]

Parameters

index (*[type]*) – [position of item]

Returns

[item at the given position]

Return type

[*type*]

concat(*args)

[Joins two or more arrays, and returns a copy of the joined arrays]

Returns

[returns a copy of the joined arrays]

Return type

[*list*]

copyWithin(target, start=0, end=None)

Copies array elements within the array, from start to end

entries()

[Returns a key/value pair Array Iteration Object]

Yields

[*type*] – [key/value pair]

every(*func*)

[Checks if every element in an array pass a test]

Parameters

func (*[type]*) – [test function]

Returns

[if every array elemnt passed the test]

Return type

[*bool*]

fill(*value=None, start=None, end=None*)

[Fills elements of an array from a start index to an end index with a static value]

filter(*func*)

Creates a new array with every element in an array that pass a test i.e. `even_numbers = someArr.filter(lambda x: x % 2 == 0)`

find(*func*)

Returns the value of the first element in an array that pass a test

findIndex(*value*)

Returns the index of the first element in an array that pass a test

findLast(*callback*)

[Returns the last element in an array that passes a test]

findLastIndex(*callback*)

[Returns the last index of an element in an array that passes a test]

flat(*depth=1*)

[Flattens an array into a single-dimensional array or a depth of arrays]

flatMap(*fn*)

[Maps a function over an array and flattens the result]

forEach(*func*)

Calls a function for each array element

static from_(*obj*)

Creates a new Array instance from an array-like or iterable object.

groupBy(*callback*) → *dict*

[Groups the elements of an array according to the result of calling a callback function on each element]

Parameters

callback (*callable*) – [the callback recieves the following paramters(value, index, target)]

Returns

[a dictionary of arrays]

Return type

[*dict*]

includes(*value*)

[Check if an array contains the specified item]

Parameters

value (*[any]*) – [any value]

Returns

[a boolean]

Return type

[bool]

indexOf(*value*)

Search the array for an element and returns its position

static isArray(*thing*)

[Checks whether an object is an array]

Parameters**thing** ([*type*]) – [thing to check]**Returns**

[True if the object is list, tuple or Array]

Return type

[bool]

join(*value*)

Joins all elements of an array into a string

keys()

Returns a Array Iteration Object, containing the keys of the original array

lastIndexOf(*value*)

Search the array for an element, starting at the end, and returns its position

property length

Sets or returns the number of elements in an array

map(*func*)

[Creates a new array with the result of calling a function for each array element]

Parameters**func** ([*type*]) – [a function to call on each array element]**Returns**

[a new array]

Return type

[list]

static of(**args*)

Creates a new Array instance with a variable number of arguments, regardless of number or type of the arguments.

pop()

Removes the last element of an array, and returns that element

prototypealias of [Array](#)**push**(*value*)

Adds new elements to the end of an array, and returns the new length

reduce(*callback*, *initialValue=None*)

Reduces the array to a single value (going left-to-right) callback recieve theses parameters: previousValue, currentValue, currentIndex, array

reduceRight(*callback*, *initialValue=None*)

Reduces the array to a single value (going right-to-left) callback recieve theses parameters: previousValue, currentValue, currentIndex, array

reverse()

Reverses the order of the elements in an array

shift()

[removes the first element from an array and returns that removed element]

Returns

[the removed array element]

Return type

[*type*]

slice(*start=0*, *stop=None*, *step=1*)

[Selects a part of an array, and returns the new array]

Parameters

- **start** ([*int*]) – [index to slice from]
- **stop** ([*int*], *optional*) – [index to slice to]. Defaults to end of the array.
- **step** (*int*, *optional*) – [description]. Defaults to 1.

Returns

[new array]

Return type

[*type*]

some(*func*)

Checks if any of the elements in an array pass a test

sort(*func=None*)

Sorts the elements of an array

splice(*start*, *delete_count=None*, **items*)

Selects a part of an array, and returns the new array

toSource()

Returns the source array.

toString()

Converts an array to a string, and returns the result

unshift(**args*)

[Adds new elements to the beginning of an array, and returns the new length]

Returns

[the length of the array]

Return type

[*int*]

class domonic.javascript.**Boolean**(*value=False*)

[Creates a Boolean Object. Warning this is NOT a boolean type. for that use Global.Boolean()]

class domonic.javascript.**Date**(*date=None, *args, formatter='python', **kwargs*)

javascript date

UTC()

Returns the number of milliseconds in a date since midnight of January 1, 1970, according to UTC time

getDate()

Returns the day of the month (from 1-31)

getDay()

Returns the day of the week (from 0-6 : Sunday-Saturday)

Returns

An integer number, between 0 and 6, corresponding to the day of the week for the given date, according to local time: 0 for Sunday, 1 for Monday, 2 for Tuesday, and so on

Return type

int

getFullYear()

Returns the year

getHours()

Returns the hour (from 0-23)

getMilliseconds()

Returns the milliseconds (from 0-999)

getMinutes()

Returns the minutes (from 0-59)

getMonth()

Returns the month (from 0-11)

getSeconds()

Returns the seconds (from 0-59)

getTime()

Returns A number representing the milliseconds elapsed between 1 January 1970 00:00:00 UTC and self.date

getTimezoneOffset()

Returns the difference, in minutes, between a date as evaluated in the UTC time zone, and the same date as evaluated in the local time zone

getUTCDate()

Returns the day of the month, according to universal time (from 1-31)

getUTCDay()

Returns the day of the week, according to universal time (from 0-6)

getUTCFullYear()

Returns the year, according to universal time

getUTCHours()

Returns the hour, according to universal time (from 0-23)

getUTCMilliseconds()

Returns the milliseconds, according to universal time (from 0-999)

getUTCMinutes()

Returns the minutes, according to universal time (from 0-59)

getUTCMonth()

Returns the month, according to universal time (from 0-11)

getUTCSeconds()

Returns the seconds, according to universal time (from 0-59)

getYear()

Deprecated. Use the `getFullYear()` method instead

static now()

Returns the number of milliseconds since midnight Jan 1, 1970

static parse(*date_string*)

Parses a date string and returns the number of milliseconds since January 1, 1970

setDate(*day: int*)

Sets the day of the month of a date object

Parameters

day (*int*) – An integer representing the day of the month.

Returns

milliseconds between epoch and updated date.

Return type

int

setFullYear(*yearValue: int, monthValue: Optional[int] = None, dateValue: Optional[int] = None*)

Sets the year of a date object

Parameters

- **yearValue** (*_type_*) – *_description_*
- **monthValue** (*int, optional*) – *_description_*. Defaults to *None*.
- **dateValue** (*int, optional*) – *_description_*. Defaults to *None*.

Returns

milliseconds between epoch and updated date.

Return type

int

setHours(*hoursValue: int, minutesValue: Optional[int] = None, secondsValue: Optional[int] = None, msValue: Optional[int] = None*)

Sets the hour of a date object

Parameters

- **hoursValue** (*int*) – an integer between 0 and 23
- **minutesValue** (*int, optional*) – an integer between 0 and 59
- **secondsValue** (*int, optional*) – an integer between 0 and 59,
- **msValue** (*int, optional*) – a number between 0 and 999,

Returns

milliseconds between epoch and updated date.

Return type

`int`

setMilliseconds(*milliseconds*: `int`)

Sets the milliseconds of a date object

Parameters

milliseconds (`int`) – Milliseconds to set i.e 123

setMinutes(*minutesValue*: `int`, *secondsValue*: `Optional[int] = None`, *msValue*: `Optional[int] = None`)

Set the minutes of a date object

Parameters

- **minutesValue** (`int`, `optional`) – an integer between 0 and 59
- **secondsValue** (`int`, `optional`) – an integer between 0 and 59,
- **msValue** (`int`, `optional`) – a number between 0 and 999,

Returns

milliseconds between epoch and updated date.

Return type

`int`

setMonth(*monthValue*: `int`, *dayValue*: `Optional[int] = None`)

Sets the month of a date object

Parameters

- **monthValue** (`int`) – a number from 0 to 11 indicating the month.
- **dayValue** (`int`, `optional`) – an optional day of the month. Defaults to 0.

Returns

milliseconds between epoch and updated date.

Return type

`int`

setSeconds(*secondsValue*: `int`, *msValue*: `Optional[int] = None`)

Sets the seconds of a date object

Parameters

- **secondsValue** (`int`) – `_description_`
- **msValue** (`int`, `optional`) – `_description_`. Defaults to None.

Returns

milliseconds between epoch and updated date.

Return type

`int`

setTime(*milliseconds*: `Optional[int] = None`)

Sets the date and time of a date object

Parameters

milliseconds (`_type_`, `optional`) – `_description_`. Defaults to None.

Returns

`_description_`

Return type

`_type_`

setUTCDate(*day*)

Sets the day of the month of a date object, according to universal time

setUTCFullYear(*year*)

Sets the year of a date object, according to universal time

setUTCHours(*hour*)

Sets the hour of a date object, according to universal time

setUTCMilliseconds(*milliseconds*)

Sets the milliseconds of a date object, according to universal time

setUTCMinutes(*minutes*)

Set the minutes of a date object, according to universal time

setUTCMonth(*month*)

Sets the month of a date object, according to universal time

setUTCSeconds(*seconds*)

Set the seconds of a date object, according to universal time

setYear(*year*)

Deprecated. Use the setFullYear() method instead

toDateString()

Converts the date portion of a Date object into a readable string

toGMTString()

Deprecated. Use the toUTCString() method instead

toISOString()

Returns the date as a string, using the ISO standard

toJSON()

Returns the date as a string, formatted as a JSON date

toLocaleDateString()

Returns the date portion of a Date object as a string, using locale conventions

toLocaleString()

Converts a Date object to a string, using locale conventions

toLocaleTimeString()

Returns the time portion of a Date object as a string, using locale conventions

toString()

Returns a string representation of the date

toTimeString()

Converts the time portion of a Date object to a string

toUTCString()

Converts a Date object to a string, according to universal time

exception domonic.javascript.**Error**(*message*, **args*, ***kwargs*)

Raise Errors

class domonic.javascript.**Function**(*func*, **args*, ***kwargs*)

a Function object

apply(*thisArg*=None, *args*=None, ***kwargs*)

[calls a function with a given this value, and arguments provided as an array]

Parameters

thisArg ([*type*]) – [The value of this provided for the call to func.]

Returns

[result of calling the function.]

Return type

[*type*]

bind(*thisArg*, **args*, ***kwargs*)

[creates a new function that, when called, has its this keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called.]

Parameters

- **thisArg** ([*type*]) – [The value to be passed as the this parameter to the target
- **called.**] (*function func when the bound function is*) –

Returns

[A copy of the given function with the specified this value, and initial arguments (if provided).]

Return type

[*type*]

call(*thisArg*=None, **args*, ***kwargs*)

[calls a function with a given this value and arguments provided individually.]

Parameters

thisArg ([*type*]) – [description]

Returns

[result of calling the function.]

Return type

[*type*]

toString()

[Returns a string representing the source code of the function. Overrides the]

class domonic.javascript.**Global**

javascript global methods

NaN()

“Not-a-Number” value

static Number(*x*)

Converts an object’s value to a number

static String(*x*)

Converts an object’s value to a string

static `clearTimeout(timeoutID)`

[cancels a timer set with `setTimeout()`]

Parameters

timeoutID (*[str]*) – [the identifier returned by `setTimeout()`]

static `decodeURI(x)`

Decodes a URI

static `decodeURIComponent(x)`

Decodes a URI component

static `encodeURI(x)`

Encodes a URI

static `encodeURIComponent(x)`

Encodes a URI component

static `eval(pythonstring)`

Evaluates a string and executes it as if it was script code

static `isFinite(x)` → *bool*

Returns true if *x* is a finite number

static `isNaN(x)`

Determines whether a value is an illegal number

static `parseFloat(x: str)`

Parses a string and returns a floating point number

static `parseInt(x: str)`

Parses a string and returns an integer

static `require(path: str)`

Loads a script from a file

static `setTimeout(callback, t, *args, **kwargs)`

[sets a timer which executes a function or evaluates an expression after a specified delay]

Parameters

- **callback** (*function*) – [method to be executed after the delay]
- **t** (*[int]*) – [milliseconds]

Returns

[an identifier for the timer]

Return type

[str]

undefined()

Indicates that a variable has not been assigned a value

class `domonic.javascript.Job(interval, execute, *args, **kwargs)`

run()

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

class domonic.javascript.**Map**(*collection*)

Map holds key-value pairs and remembers the original insertion order of the keys.

clear()

Removes all key-value pairs from the Map object.

delete(*key: str*) → **bool**

Returns true if an element in the Map object existed and has been removed, or false if the element does not exist. Map.prototype.has(key) will return false afterwards.

entries()

Returns a new Iterator object that contains an array of [key, value] for each element in the Map object in insertion order.

get(*key: str, default=None*)

Returns the value associated to the key, or undefined if there is none.

has(*key: str*) → **bool**

Returns a boolean asserting whether a value has been associated to the key in the Map object or not.

keys()

Returns a new Iterator object that contains the keys for each element in the Map object in insertion order.

set(*key: str, value*)

Sets the value for the key in the Map object. Returns the Map object.

values()

Returns a new Iterator object that contains the values for each element in the Map object in insertion order.

class domonic.javascript.**Math**(*obj=None, *args, **kwargs*)

Math class that mirrors javascript implementation.

i.e. you can pass strings and it will also work, Math.abs('-1')

static hypot(*args)

returns the square root of the sum of squares of its arguments

static log2(*args)

returns the square root of the sum of squares of its arguments

static loglp(*args)

returns the natural logarithm (base e) of 1 + a number, that is

class domonic.javascript.**Number**(*x="", *args, **kwargs*)

javascript Number methods

NEGATIVE_INFINITY = inf

Represents negative infinity (returned on overflow) Number

POSITIVE_INFINITY = -inf

Represents infinity (returned on overflow) Number

isInteger()

Checks whether a value is an integer

isSafeInteger()

Checks whether a value is a safe integer

toExponential(*num=None*)

Converts a number into an exponential notation

toFixed(*digits: int*)

[formats a number using fixed-point notation.]

Parameters

digits (*[int]*) – [The number of digits to appear after the decimal point]

Returns

[A string representing the given number using fixed-point notation.]

Return type

[str]

toPrecision(*precision*)

[returns a string representing the Number object to the specified precision.]

Parameters

precision (*[int]*) – [An integer specifying the number of significant digits.]

Returns

[A string representing a Number object in fixed-point or exponential notation rounded to precision significant digits]

Return type

[str]

toString(*base: int*)

[returns a string representing the specified Number object.]

Parameters

base (*int*) – [An integer in the range 2 through 36 specifying the base to use for representing numeric values.]

Returns

[a string representing the specified Number object]

Return type

[str]

exception domonic.javascript.**ProgramKilled**

class domonic.javascript.**Reflect**

The Reflect object provides the following static functions which have the same names as the proxy handler methods. Some of these methods are also the same as corresponding methods on Object, although they do have some subtle differences between them.

static apply(*target, thisArgument, argumentsList*)

Calls a target function with arguments as specified by the argumentsList parameter. See also Function.prototype.apply().

static construct(*target, argumentsList, newTarget*)

The new operator as a function. Equivalent to calling new target(...argumentsList). Also provides the option to specify a different prototype.

static defineProperty(*target, propertyKey, attributes*)

Similar to Object.defineProperty(). Returns a Boolean that is true if the property was successfully defined.

static deleteProperty(*target*, *propertyKey*)

The delete operator as a function. Equivalent to calling delete target[propertyKey].

static get(*target*, *propertyKey*, *receiver*)

Returns the value of the property. Works like getting a property from an object (target[propertyKey]) as a function.

static getOwnPropertyDescriptor(*target*, *propertyKey*)

Similar to Object.getOwnPropertyDescriptor(). Returns a property descriptor of the given property if it exists on the object, undefined otherwise.

getPrototypeOf()

Returns the prototype (internal [[Prototype]] property) of the specified object.

static has(*target*, *propertyKey*)

Returns a Boolean indicating whether the target has the property. Either as own or inherited. Works like the in operator as a function.

static ownKeys(*target*)

Returns an array of the target object's own (not inherited) property keys.

static preventExtensions(*target*)

Similar to Object.preventExtensions(). Returns a Boolean that is true if the update was successful.

static set(*target*, *propertyKey*, *value*, *receiver*)

A function that assigns values to properties. Returns a Boolean that is true if the update was successful.

static setPrototypeOf(*target*, *prototype*)

A function that sets the prototype of an object. Returns a Boolean that is true if the update was successful.

class domonic.javascript.Screen

screen

availHeight()

Returns the height of the screen (excluding the Windows Taskbar)

availWidth()

Returns the width of the screen (excluding the Windows Taskbar)

colorDepth()

Returns the colorDepth

height()

Returns the total height of the screen

pixelDepth()

Returns the pixelDepth

width()

Returns the total width of the screen

class domonic.javascript.String(*x*="", **args*, ***kwargs*)

javascript String methods

big()

[wraps the string in big tags]

Returns

[the string in big tags]

Return type`[str]`**blink()**`[wraps the string in blink tags]`**Returns**`[the string in blink tags]`**Return type**`[str]`**bold()**`[wraps the string in bold tags]`**Returns**`[the string in bold tags]`**Return type**`[str]`**charAt(*index: int*) → str**`[Returns the character at the specified index (position)]`**Parameters****index** (*int*) – [index position]**Returns**`[the character at the specified index. if the index is out of range, an empty string is returned.]`**Return type**`[str]`**charCodeAt(*index: int*) → int**`Returns the Unicode of the character at the specified index`**codePointAt(*index: int*)**`[Returns the Unicode code point at the specified index (position)]`**Parameters****index** (*int*) – [index position]**Returns**`[the Unicode code point at the specified index (position)]`**Return type**`[type]`**compile(*pattern: str*)**`Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.`**concat(*args, seperator: str = ") → str**`[concatenates the string arguments to the calling string and returns a new string.]`**Parameters****seperator** (*str*, *optional*) – []. Defaults to “”.**Returns**`[A new string containing the combined text of the strings provided.]`

Return type`[type]`**div**(**args*, ***kwargs*)

[wraps the string in a div tag]

Returns

[the string in a div tag]

Return type`[str]`**endsWith**(*x*: *str*, *start*: *Optional[int]* = *None*, *end*: *Optional[int]* = *None*) → *bool*

Checks whether a string ends with specified string/characters

fixed()

[wraps the string in fixed tags]

Returns

[the string in fixed tags]

Return type`[str]`**fontcolor**(*color*: *str*)

[wraps the string in font tags with a specified color]

Parameters**color** (*str*) – [the color to use]**Returns**

[the string in font tags]

Return type`[str]`**fontsize**(*size*: *str*)

[wraps the string in font tags with a specified size]

Parameters**size** (*str*) – [the size to use]**Returns**

[the string in font tags]

Return type`[str]`**fromCharCode**(**codes*) → *str*

returns a string created from the specified sequence of UTF-16 code units

static fromCodePoint(*codePoint*: *int*)

Converts a Unicode code point into a string

includes(*searchValue*: *str*, *position*: *int* = 0) → *bool*

[returns true if the specified string is found within the calling String object,]

Parameters

- **searchValue** (*str*) – [The string value to search for.]
- **position** (*int*, *optional*) – [the position to search from]. Defaults to 0.

Returns

[a boolean value indicating whether the search value was found.]

Return type

[*type*]

indexOf(*searchValue*: *str*, *fromIndex*: *int* = 0)

[returns the index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex]

Parameters

- **searchValue** (*str*) – [The string value to search for.]
- **fromIndex** (*int*) – [An integer representing the index at which to start the search]

Returns

[The index of the first occurrence of searchValue, or -1 if not found.]

Return type

[*type*]

italics()

[wraps the string in italics tags]

Returns

[the string in italics tags]

Return type

[*str*]

lastIndexOf(*searchValue*: *str*, *fromIndex*: *int* = 0)

returns the last index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex

link(*url*: *str*)

[wraps the string in a link tag]

Parameters

url (*str*) – [the url to use]

Returns

[the string in a link tag]

Return type

[*str*]

localeCompare(*comparisonString*: *str*, *locale*: *Optional*[*str*] = None, *args) → *int*

method returns a number indicating whether a reference string comes before, or after, or is the same as the given string in sort order

match(*pattern*: *str*)

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

matchAll(*pattern*: *str*)

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

padEnd(*length*: *int*, *padChar*: *str* = ' ') → *str*

[Pads the end of a string with a specified character (repeated, if needed) to create a new string.]

Parameters

- **length** (*int*) – [the length of the resulting string]
- **padChar** (*str*, *optional*) – [the character to use for padding. Defaults to " "].

Returns

[the padded string]

Return type

[*str*]

padStart(*length*: *int*, *padChar*: *str* = ' ') → *str*

[Pads the start of a string with a specified character]

Parameters

- **length** (*int*) – [the length of the resulting string]
- **padChar** (*str*, *optional*) – [the character to use for padding. Defaults to " "].

Returns

[the padded string]

Return type

[*str*]

static raw(*string*)

Returns the string as-is

repeat(*count*: *int*) → *str*

Returns a new string with a specified number of copies of an existing string

replace(*old*: *str*, *new*) → *str*

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

replaceAll(*old*: *str*, *new*: *str*)

[returns a new string where the specified values are replaced. ES2021]

Parameters

- **old** ([*str*]) – [word to remove]
- **new** ([*str*]) – [word to replace it with]

Returns

[new string with all occurrences of old word replaced]

Return type

[*str*]

search(*searchValue*: *str*, *position*: *int* = 0) → *bool*

[returns true if the specified string is found within the calling String object,] starting at the specified position.
:param searchValue: [The string value to search for.] :type searchValue: *str* :param position: [the position to search from]. Defaults to 0. :type position: *int*, optional

Returns

[a boolean value indicating whether the search value was found.]

Return type`[type]`**slice**(*start*: `int` = 0, *end*: `Optional[int]` = None) → `str`

Selects a part of an string, and returns the new string

small()

[wraps the string in small tags]

Returns

[the string in small tags]

Return type`[str]`**split**(*expr*) → `list`

[can split a string based on a regex]

Parameters**expr** (`[str]`) – [valid regex or string to split on]**Returns**

[list of str]

Return type`[list]`**startsWith**(*x*: `str`, *start*: `Optional[int]` = None, *end*: `Optional[int]` = None) → `bool`

Checks whether a string begins with specified characters

strike()

[wraps the string in strike tags]

Returns

[the string in strike tags]

Return type`[str]`**sub**()

[wraps the string in sub tags]

Returns

[the string in sub tags]

Return type`[str]`**substr**(*start*: `int` = 0, *end*: `Optional[int]` = None)

Extracts the characters from a string, beginning at a specified start position, and through the specified number of character

substring(*start*: `int`, *end*: `Optional[int]` = None) → `str`

Extracts the characters from a string, between two specified indices

sup()

[wraps the string in sup tags]

Returns

[the string in sup tags]

Return type`[str]`**static** `toCharCode(char: str)`

Converts a Unicode string into a code point

static `toCodePoint(char: str)`

Converts a Unicode string into a code point

`toLocaleLowerCase()` → `str`

Converts a string to lowercase letters, according to the host's locale

`toLocaleUpperCase()` → `str`

Converts a string to uppercase letters, according to the host's locale

`toLowerCase()` → `str`

Converts a string to lowercase letters

`toUpperCase()` → `str`

Converts a string to uppercase letters

`trim()`

Removes whitespace from both ends of a string

`trimEnd(length: int)`

[Removes whitespace from the end of a string]

Parameters**length** (`int`) – [the length of the resulting string]**Returns**

[the trimmed string]

Return type`[type]``trimStart(length: int)`

[Removes whitespace from the beginning of a string.]

Parameters**length** (`int`) – [the length of the resulting string]**Returns**

[the trimmed string]

Return type`[str]``webpage()`

[wraps the string in a webpage]

Returns

[the string as a webpage]

Return type`[str]`**class** `domonic.javascript.Window(*args, **kwargs)`

window

static alert(*msg*)

Displays an alert box with a message and an OK button

static atob(*dataString*)

Decodes a base-64 encoded string

static btoa(*dataString*)

Encodes a string in base-64

clearTimeout()

[cancels a timer set with setTimeout()]

Parameters

timeoutID ([*str*]) – [the identifier returned by setTimeout()]

static prompt(*msg*, *default_text*="")

Displays a dialog box that prompts the visitor for input

static requestAnimationFrame(*callback*)

[requests a frame of an animation]

Parameters

callback (*callable*) – [the callback function]

Returns

[description]

Return type

[*type*]

setTimeout(*t*, **args*, ***kwargs*)

[sets a timer which executes a function or evaluates an expression after a specified delay]

Parameters

- **callback** (*function*) – [method to be executed after the delay]
- **t** ([*int*]) – [milliseconds]

Returns

[an identifier for the timer]

Return type

[*str*]

class domonic.javascript.Worker(*script*)

[A background task that can be created via script, which can send messages back to its creator. Creating a worker is done by calling the Worker("path/to/worker/script") constructor.] TODO - JSWorker - Node :param object: [takes a path to a python script] :type object: [*str*]

postMessage()

Sends a message — consisting of any object — to the worker's inner scope.

terminate()

Immediately terminates the worker. This does not let worker finish its operations; it is halted at once. ServiceWorker instances do not support this method.

domonic.javascript.as_signed(*value*, *bits*)

Converts an unsigned integer to a signed integer.

`domonic.javascript.clearTimeout(timeoutID)`

[cancels a timer set with `setTimeout()`]

Parameters

`timeoutID` (`[str]`) – [the identifier returned by `setTimeout()`]

`domonic.javascript.decodeURI(x)`

Decodes a URI

`domonic.javascript.decodeURIComponent(x)`

Decodes a URI component

`domonic.javascript.encodeURI(x)`

Encodes a URI

`domonic.javascript.encodeURIComponent(x)`

Encodes a URI component

`domonic.javascript.function(python_str: str) → str`

[evals a string i.e.

`sup = function(“print(hi)”) sup()`

]

Parameters

`python_str` (`[str]`) – [some valid python code as a string]

`domonic.javascript.parseFloat(x: str)`

Parses a string and returns a floating point number

`domonic.javascript.parseInt(x: str)`

Parses a string and returns an integer

`domonic.javascript.setTimeout(callback, t, *args, **kwargs)`

[sets a timer which executes a function or evaluates an expression after a specified delay]

Parameters

- **`callback`** (*function*) – [method to be executed after the delay]
- **`t`** (`[int]`) – [milliseconds]

Returns

[an identifier for the timer]

Return type

[*str*]

`domonic.javascript.window`

alias of [Window](#)

1.4.4 events

There's an Event Dispatcher i.e

```
from domonic.events import *

class SomeEventHandler(EventDispatcher):

    def __init__(self):
        super().__init__(self)
        self.addEventListener('some_event', self.on_custom_event)

    def on_custom_event(evt):
        print('that just happened')

my_handler = SomeEventHandler()
my_handler.dispatchEvent('some_event')
```

If you create a python virtual DOM with domonic you can listen for events like so...

```
def on_page_clicked(evt):
    print('the page was just clicked', evt)
    print('mouseX', evt.x)
    print('mouseY', evt.y)

page.addEventListener( MouseEvent.CLICK, on_page_clicked )
```

- above example needs a MouseEvent relaying or dispatching from somewhere. i.e. examples/events

domonic.events

<https://developer.mozilla.org/en-US/docs/Web/Events>

```
class domonic.events.AnimationEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ANIMATIONEND: str = 'animationend'

    ANIMATIONITERATION: str = 'animationiteration'

    ANIMATIONSTART: str = 'animationstart'

    animationName
        Returns the name of the animation

    elapsedTime
        Returns the number of seconds an animation has been running

    pseudoElement
        Returns the name of the pseudo-element of the animation

class domonic.events.BeforeUnloadEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    BEFOREUNLOAD: str = 'beforeunload'

    BeforeUnloadEvent

class domonic.events.ClipboardEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```



```

COPY: str = 'copy'

CUT: str = 'cut'

PASTE: str = 'paste'

clipboardData
    Returns an object containing the data affected by the clipboard operation
class domonic.events.CompositionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
class domonic.events.CustomEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
class domonic.events.DOMContentLoadedEvent(_type: str, options: Optional[dict] = None, *args,
                                             **kwargs)

    DOMCONTENTLOADED: str = 'DOMContentLoaded'

document
    Returns the document that was loaded
class domonic.events.DragEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    DRAG: str = 'drag'

    DROP: str = 'drop'

    END: str = 'dragend'

    ENTER: str = 'dragenter'

    EXIT: str = 'dragexit'

    LEAVE: str = 'dragleave'

    OVER: str = 'dragover'

    START: str = 'dragstart'

dataTransfer
    Returns the data that is dragged/dropped
class domonic.events.ErrorEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ERROR: str = 'error'

class domonic.events.Event(_type: str, options: Optional[dict] = None, *args, **kwargs)
    event

    ABORT: str = 'abort'

    AFTERPRINT: str = 'afterprint'

    BEFOREPRINT: str = 'beforeprint'

    BEFOREUNLOAD: str = 'beforeunload'

    CANPLAY: str = 'canplay'

    CANPLAYTHROUGH: str = 'canplaythrough'

```

```
CHANGE: str = 'change'
DURATIONCHANGE: str = 'durationchange'
EMPTIED: str = 'emptied'
ENDED: str = 'ended'
ERROR: str = 'error'
FULLSCREENCHANGE: str = 'fullscreenchange'
FULLSCREENERROR: str = 'fullscreenerror'
INPUT: str = 'input'
INVALID: str = 'invalid'
LOAD: str = 'load'
LOADEDATA: str = 'loadeddata'
LOADEDMETADATA: str = 'loadedmetadata'
MESSAGE: str = 'message'
OFFLINE: str = 'offline'
ONLINE: str = 'online'
OPEN: str = 'open'
PAUSE: str = 'pause'
PLAY: str = 'play'
PLAYING: str = 'playing'
PROGRESS: str = 'progress'
RATECHANGE: str = 'ratechange'
READYSTATECHANGE: str = 'readystatechange'
RESET: str = 'reset'
RESIZE: str = 'resize'
SCROLL: str = 'scroll'
SEARCH: str = 'search'
SEEKED: str = 'seeked'
SEEKING: str = 'seeking'
SELECT: str = 'select'
SHOW: str = 'show'
STALLED: str = 'stalled'
```

```

    SUBMIT: str = 'submit'

    SUSPEND: str = 'suspend'

    TOGGLE: str = 'toggle'

    UNLOAD: str = 'unload'

    VOLUMECHANGE: str = 'volumechange'

    WAITING: str = 'waiting'

    stopPropagation()
        [prevents further propagation of the current event in the capturing and bubbling phases]
domonic.events.EventDispatcher
    legacy alias
class domonic.events.EventTarget(*args, **kwargs)
    EventTarget is a class you can extend to give your obj event dispatching abilities
class domonic.events.ExtensibleEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    extendable
        Returns whether the event is extendable or not

    timeStamp: float
        Returns the time stamp of the event
class domonic.events.FetchEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    FETCH: str = 'fetch'

    clientId
        Returns the client ID of the fetch request

    property isReload
        Returns whether the request is a reload or not

    request
        Returns the request object

    respondWith(response)
        Returns a promise that resolves to the response object

    waitUntil(promise)
        Returns a promise that resolves when the response is available
class domonic.events.FocusEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    BLUR: str = 'blur'

    FOCUS: str = 'focus'

    FOCUSIN: str = 'focusin'

    FOCUSOUT: str = 'focusout'

class domonic.events.GamePadEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

```

```
START: str = 'gamepadconnected'

STOP: str = 'gamepaddisconnected'

class domonic.events.HashChangeEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    CHANGE: str = 'hashchange'

class domonic.events.InputEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    CHANGE: str = 'change'

    INPUT: str = 'input'

    SELECT: str = 'select'

    data
        Returns the inserted characters

    dataTransfer
        Returns an object containing information about the inserted/deleted data

    getTargetRanges()
        Returns an array containing target ranges that will be affected by the insertion/deletion

    inputType
        Returns the type of the change (i.e “inserting” or “deleting”)

    isComposing
        Returns whether the state of the event is composing or not

class domonic.events.KeyboardEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
    keyboard events

    DOM_KEY_LOCATION_JOYSTICK: int = 5

    DOM_KEY_LOCATION_LEFT: int = 0

    DOM_KEY_LOCATION_MOBILE: int = 4

    DOM_KEY_LOCATION_NUMPAD: int = 3

    DOM_KEY_LOCATION_RIGHT: int = 2

    DOM_KEY_LOCATION_STANDARD: int = 1

    KEYDOWN: str = 'keydown'

    KEYPRESS: str = 'keypress'

    KEYUP: str = 'keyup'

class domonic.events.MessageEvent(_type, options: Optional[dict] = None, *args, **kwargs)

    CONNECT: str = 'connect'

    DISCONNECT: str = 'disconnect'

    MESSAGE: str = 'message'
```

data

Returns the data of the message

lastEventId

Returns the last event id of the message

origin

Returns the origin of the message

ports

Returns the ports of the message

source

Returns the source of the message

```
class domonic.events.MouseEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

mouse events

CLICK: str = 'click'

CONTEXTMENU: str = 'contextmenu'

DBLCLICK: str = 'dblclick'

MOUSEDOWN: str = 'mousedown'

MOUSEENTER: str = 'mouseenter'

MOUSELEAVE: str = 'mouseleave'

MOUSEMOVE: str = 'mousemove'

MOUSEOUT: str = 'mouseout'

MOUSEOVER: str = 'mouseover'

MOUSEUP: str = 'mouseup'

getModifierState()

Returns an array containing target ranges that will be affected by the insertion/deletion

```
class domonic.events.PageTransitionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

PAGEHIDE: str = 'pagehide'

PAGESHOW: str = 'pageshow'

persisted

Returns whether the webpage was cached by the browser

```
class domonic.events.PointerEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

POINTER: str = 'pointer'

POINTERCANCEL: str = 'pointercancel'

POINTERDOWN: str = 'pointerdown'

POINTERENTER: str = 'pointerenter'

POINTERLEAVE: `str` = 'pointerleave'

POINTERMOVE: `str` = 'pointermove'

POINTEROUT: `str` = 'pointerout'

POINTEROVER: `str` = 'pointerover'

POINTERUP: `str` = 'pointerup'

`class domonic.events.PopStateEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

POPSTATE: `str` = 'popstate'

`state`

Returns an object containing a copy of the history entries

`class domonic.events.ProgressEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

ABORT: `str` = 'abort'

ERROR: `str` = 'error'

LOAD: `str` = 'load'

LOADED: `str` = 'loaded'

LOADEND: `str` = 'loadend'

LOADSTART: `str` = 'loadstart'

PROGRESS: `str` = 'progress'

TIMEOUT: `str` = 'timeout'

`class domonic.events.PromiseRejectionEvent(_type, options=None, *args, **kwargs)`

HANDLED: `str` = 'rejectionhandled'

UNHANDLED: `str` = 'unhandledrejection'

`isRejected`

Returns whether the promise was rejected or not

`promise`

Returns the promise that was rejected

`reason`

Returns the reason of the rejection

`class domonic.events.SVGEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

ABORT: `str` = 'abort'

ERROR: `str` = 'error'

LOAD: `str` = 'load'

LOADEDATA: `str` = 'loadeddata'

LOADEDMETADATA: `str` = 'loadedmetadata'

LOADSTART: `str` = 'loadstart'

PROGRESS: `str` = 'progress'

SCROLL: `str` = 'scroll'

UNLOAD: `str` = 'unload'

```
class domonic.events.SecurityPolicyViolationEvent(_type: str, options: Optional[dict] = None, *args,
                                                  **kwargs)
```

SECURITY_POLICY_VIOLATION: `str` = 'securitypolicyviolation'

blockedURI

Returns the blocked URI

frame

Returns the frame that violated the policy

isFrameAncestor

Returns whether the frame is an ancestor of the frame that violated the policy

isMainFrame

Returns whether the frame is the main frame

originalPolicy

Returns the original policy

timeStamp: `float`

Returns the time stamp of the event

violatedDirective

Returns the violated directive

```
class domonic.events.StorageEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

STORAGE: `str` = 'storage'

key

Returns the key of the changed storage item

newValue

Returns the new value of the changed storage item

oldValue

Returns the old value of the changed storage item

storageArea

Returns an object representing the affected storage object

url

Returns the URL of the changed item's document

```
class domonic.events.SubmitEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

SUBMIT: `str` = 'submit'

```
class domonic.events.SyncEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    SYNC: str = 'sync'

    lastChance
        Returns whether the sync event is the last chance or not

    tag
        Returns the tag of the sync event

    timeStamp: float
        Returns the time stamp of the event

class domonic.events.TimerEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    TIMER: str = 'timer'

    TIMER_COMPLETE: str = 'timercomplete'
        TimerEvent

class domonic.events.TouchEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    TOUCHCANCEL: str = 'touchcancel'

    TOUCHEND: str = 'touchend'

    TOUCHMOVE: str = 'touchmove'

    TOUCHSTART: str = 'touchstart'

class domonic.events.TransitionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    TRANSITIONEND: str = 'transitionend'

    elapsedTime
        Returns the number of seconds a transition has been running

    propertyName
        Returns the name of the transition

    pseudoElement
        Returns the name of the pseudo-element of the transition

class domonic.events.TweenEvent(_type, source=None, bubbles=False, cancelable=False)

    COMPLETE: str = 'onComplete'

    PAUSE: str = 'onPause'

    RESET: str = 'onReset'

    START: str = 'onStart'

    STOP: str = 'onStop'

    TIMER: str = 'onTimer'

    UNPAUSE: str = 'onUnPause'

    UPDATE_END: str = 'onUpdateEnd'
```



```
UPDATE_START: str = 'onUpdateStart'
```

```
class domonic.events.UIEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
class domonic.events.WheelEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
WHEEL: str = 'wheel'
```

1.4.5 sitemap

domonic can help create a sitemap or sitemapindex for your website.

A sitemap contains a list of urls for your website. Whereas a sitemap index contains a list of sitemaps.

You can see below How to make sitemaps with python and domonic.

creating a sitemapindex

A sitemap index contains a list of sitemaps. A minimal one might look something like this:

```
'<?xml version="1.0" encoding="UTF-8"?>'
<sitemapindex xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<sitemap>
  <loc>http://www.example.com/sitemap1.xml.gz</loc>
  <lastmod>2004-10-01T18:23:17+00:00</lastmod>
</sitemap>
</sitemapindex>
```

With domonic we can create one in a number ways depending on our needs.

```
from domonic.xml.sitemap import sitemapindex, sitemap, url, loc, lastmod, changefreq,
    priority

doc = sitemapindex(
    sitemap(
        loc(https://xyz.net/sitemap1.xml)
        lastmod('2021-07-08T13:12:16+00:00') # pass a date as string. if no
    ) # data is passed the current date is used
)

render(f'{doc}', 'sitemap.xml')
```

Create a sitemap

A sitemap contains a list of urls for your website and is limited to 50,000 urls.

A minimal one might look something like this:

```
'<?xml version="1.0" encoding="UTF-8"?>'
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
  <loc>https://xyz.net/</loc>
```

(continues on next page)

(continued from previous page)

```
<lastmod>2004-10-01T18:23:17+00:00</lastmod>
<changefreq>monthly</changefreq>
<priority>0.8</priority>
</url>
</urlset>
```

With domonic we can create one in a number ways depending on our needs.

```
from domonic.xml.sitemap import sitemapindex, sitemap, url, loc, lastmod, changefreq,
priority

doc = urlset(
    url(
        loc('https://xyz.net')
        lastmod('2021-07-08T13:12:16+00:00') # pass a date as string. if no
        data is passed the current date is used
        changefreq('weekly')
        priority(0.5)
    )
)

# use f-string to call format on the doc to prettify
render(f"{doc}", 'sitemap1.xml')
```

utils

domonic also has some utils for quickly creating sitemaps with default values.

```
mypages = []
sm = sitemap_from_urls(mypages)
print(sm)
```

but you will likely want a little more control. So use any dom manipulating methods you like.

Here's some more examples.

creating a sitemap from scratch

```
sm = urlset()
sm += url(loc('https://abc.net/sitemap.xml'), lastmod('2020-07-08T13:12:16+00:00'))

print(sm)
```

formatting

You can format with following normal python methods which are regonised by domonic:

```
print(f"{sm}") # f string will call __format__ and run through a prettify
print(f"{sm!s}") # str will not be prettified
print(f"{sm!r}") # r show the object as a repr
# print(f"{sm!a}")
```

more

For more infor on sitemaps see...

<https://www.sitemaps.org/protocol.html>

domonic.sitemap

generate or load sitemaps

warning - when using image and video tags from this package they will be namespaced i.e <image:image> and <video:video> so i'd advise to only import them within the def that you use them in to avoid conflict with html.image

```
class domonic.xml.sitemap.changefreq(*args, **kwargs)
```

```
class domonic.xml.sitemap.lastmod(*args, **kwargs)
```

```
class domonic.xml.sitemap.loc(*args, **kwargs)
```

```
class domonic.xml.sitemap.priority(*args, **kwargs)
```

```
class domonic.xml.sitemap.sitemap(*args, **kwargs)
```

```
domonic.xml.sitemap.sitemap_format(self, *args, **kwargs)
```

attempts to prettify the output of the sitemap.

```
domonic.xml.sitemap.sitemap_from_urls(urls)
```

Create a sitemap from a list of urls.add()

Note: This won't allow you to add priority or changefreq of the urls. or add images etc tho u could loop the nodes afterwards and do that.

WARNING: there's a difference between a sitemap index and a sitemap. make sure you know what you want.

```
class domonic.xml.sitemap.sitemapindex(*args, **kwargs)
```

```
domonic.xml.sitemap.sitemapindex_from_urls(urls)
```

Create a sitemap index from a list of urls.

WARNING: there's a difference between a sitemap index and a sitemap. make sure you know what you want.

i.e

```
# <?xml version="1.0" encoding="UTF-8"?> # <sitemapindex xmlns="http://www.sitemaps.org/
schemas/sitemap/0.9"> # <sitemap> # <loc>https://xyz.com/sitemap1.xml</loc> # <lastmod>2021-07-
08T13:12:16+00:00</lastmod> # </sitemap> # </sitemapindex>
```

```
class domonic.xml.sitemap.url(*args, **kwargs)
```

```
class domonic.xml.sitemap.urlset(*args, **kwargs)
```

1.4.6 dQuery

dQuery is for querying and manipulating your server-side dom.

querying

dQuery uses the ° symbol (alt+0).

```
from domonic.html import *
from domonic.dQuery import °

d = html(head(body(li(_class='things'), div(_id="test"))))

°(d) # you need to init a dom first. i.e. a html element

# now you can use it
print( °('#test') )
print( °('.things') )
a = °('<div class="test2"></div>')
print( a )

b = °('#test').append(a)
print(b)
```

You can quickly access returned elements as if it were a list...

```
somehtml = °('<html><table id="mytable" class="one"></table></html>')
str(°('html')[0])
```

But you don't need a DOM fragment to use dquery. It also contains useful static methods. i.e.

```
first = ["a", "b", "c"]
second = ["d", "e", "f"]
result = °.merge(first, second)
print(result)

obj1 = {'a':1, 'b':2}
obj2 = {'c':1, 'b':5}
print(°.extend(obj1,obj2))

print(°.trim(" some tst \n TEST."))

print(°.now())
```

domonic.dQuery

alt + 0

domonic.dQuery.dQuery

alias of o

class domonic.dQuery.dQuery_el(dom, *args, **kwargs)

alt + 0

dQuery - methods for querying domonic

add(elements)

Create a new dQuery object with elements added to the set of matched elements.

addBack(selector)

Add the previous set of elements on the stack to the current set, optionally filtered by a selector.

addClass(name: str)

Adds the specified class to each element in the set of matched elements.

after(newnode)

Insert content, specified by the parameter, after each element in the set of matched elements.

ajaxComplete(handler)

Register a handler to be called when Ajax requests complete. This is an Ajax Event

ajaxError(handler)

Register a handler to be called when Ajax requests complete with an error. This is an Ajax Event

ajaxSend(handler)

Register a handler to be called when Ajax requests complete successfully. This is an Ajax Event

ajaxStart(handler)

Register a handler to be called when the first Ajax request begins. This is an Ajax Event

ajaxStop(handler)

Register a handler to be called when all Ajax requests have completed. This is an Ajax Event

ajaxSuccess(handler)

Attach a function to be executed whenever an Ajax request completes successfully. This is an Ajax Event.

andSelf()

Add the previous set of elements on the stack to the current set.

animate()

Perform a custom animation of a set of CSS properties.

append(html)

Insert content, specified by the parameter, to the end of each element in the set of matched elements.

appendTo(target)

Insert every element in the set of matched elements to the end of the target.

attr(property, value=None)

Get the value of an attribute for the first element in the set of matched elements or set one or more attributes for every matched element.

before(*content*)

Insert content, specified by the parameter, before each element in the set of matched elements.

bind(*event, handler*)

Attach a function to be executed when an event occurs on a set of matched elements.

blur(*handler*)

Bind an event handler to the “blur” JavaScript event, or trigger that event on an element.

change(*handler*)

Bind an event handler to the “change” JavaScript event, or trigger that event on an element.

children(*selector=None*)

Get the children of each element in the set of matched elements, optionally filtered by a selector.

clearQueue()

Remove from the queue all items that have not yet been run.

click(*handler*)

Bind an event handler to the “click” JavaScript event, or trigger that event on an element.

clone()

Create a deep copy of the set of matched elements.

closest(*selector=None*)

For each element in the set, get the first element that matches the selector by testing the element itself and traversing up through its ancestors in the DOM tree.

contents(*selector=None*)

Get the children of each element in the set of matched elements, including text and comment nodes.

property context

The DOM node context originally passed to dQuery if none was passed then context will likely be the document.

contextmenu()

Bind an event handler to the “contextmenu” JavaScript event, or trigger that event on an element.

css(*property, value=None*)

Get the value of a computed style property for the first element in the set of matched elements or set one or more CSS properties for every matched element.

data(*key, value=None*)

Store arbitrary data associated with the matched elements or return the value at the named data store for the first element in the set of matched elements.

dblclick(*handler*)

Bind an event handler to the “dblclick” JavaScript event, or trigger that event on an element.

delay(*time*)

Set a timer to delay execution of subsequent items in the queue.

delegate(*selector, event, handler*)

Attach a handler to one or more events for all elements that match the selector, now or in the future, based on a specific set of root elements.

dequeue()

Execute the next function on the queue for the matched elements.

detach()

Remove the set of matched elements from the DOM.

die()

Remove event handlers previously attached using `.live` from the elements.

each(*func*)

Iterate over a dQuery object, executing a function for each matched element.

empty()

Remove all child nodes of the set of matched elements from the DOM.

end()

End the most recent filtering operation in the current chain and return the set of matched elements to its previous state.

eq(*index*)

Reduce the set of matched elements to the one at the specified index.

error(*handler*)

Bind an event handler to the “error” JavaScript event.

even()

Reduce the set of matched elements to the even ones in the set, numbered from zero.

fadeIn()

Display the matched elements by fading them to opaque.

fadeOut()

Hide the matched elements by fading them to transparent.

fadeTo()

Adjust the opacity of the matched elements.

fadeToggle()

Display or hide the matched elements by animating their opacity.

filter(*selector*)

Reduce the set of matched elements to those that match the selector or pass the function’s test.

find(*selector*)

Get the descendants of each element in the current set of matched elements, filtered by a selector, dQuery object, or element.

finish()

Stop the currently-running animation, remove all queued animations, and complete all animations for the matched elements.

first()

Reduce the set of matched elements to the first in the set.

focus()

Bind an event handler to the “focus” JavaScript event, or trigger that event on an element.

focusin()

Bind an event handler to the “focusin” event.

focusout()

Bind an event handler to the “focusout” JavaScript event.

has(selector)

Reduce the set of matched elements to those that have a descendant that matches the selector or DOM element.

hasClass(classname)

Determine whether any of the matched elements are assigned the given class.

height()

Get the current computed height for the first element in the set of matched elements or set the height of every matched element.

hide()

Hide the matched elements.

hover()

Bind one or two handlers to the matched elements, to be executed when the mouse pointer enters and leaves the elements.

html(html=None)

Get the HTML contents of the first element in the set of matched elements or set the HTML contents of every matched element.

index()

Search for a given element from among the matched elements.

innerHeight()

Get the current computed inner height (including padding but not border) for the first element in the set of matched elements or set the inner height of every matched element.

innerWidth()

Get the current computed inner width (including padding but not border) for the first element in the set of matched elements or set the inner width of every matched element.

insertAfter(target)

Insert the matched elements after the specified target element.

insertBefore(target)

Insert every element in the set of matched elements before the target.

keydown()

Bind an event handler to the “keydown” JavaScript event, or trigger that event on an element.

keypress()

Bind an event handler to the “keypress” JavaScript event, or trigger that event on an element.

keyup()

Bind an event handler to the “keyup” JavaScript event, or trigger that event on an element.

last()

Reduce the set of matched elements to the final one in the set.

property length

The number of elements in the dQuery object.

live()

Attach an event handler for all elements which match the current selector, now and in the future.

load(*url*, *data=None*, *complete=None*)

Load data from the server and place the returned HTML into the matched elements.

map(*func*)

Pass each element in the current matched set through a function, producing a new dQuery object containing the return values.

mousedown()

Bind an event handler to the “mousedown” JavaScript event, or trigger that event on an element.

mouseenter()

Bind an event handler to be fired when the mouse enters an element, or trigger that handler on an element.

mouseleave()

Bind an event handler to be fired when the mouse leaves an element, or trigger that handler on an element.

mousemove()

Bind an event handler to the “mousemove” JavaScript event, or trigger that event on an element.

mouseout()

Bind an event handler to the “mouseout” JavaScript event, or trigger that event on an element.

mouseover()

Bind an event handler to the “mouseover” JavaScript event, or trigger that event on an element.

mouseup()

Bind an event handler to the “mouseup” JavaScript event, or trigger that event on an element.

next(*selector*)

Get the immediately following sibling of each element in the set of matched elements. If a selector is provided, it retrieves the next sibling only if it matches that selector.

nextAll(*selector*)

Get all following siblings of each element in the set of matched elements, optionally filtered by a selector.

nextUntil(*selector*)

Get all following siblings of each element up to but not including the element matched by the selector, DOM node, or dQuery object passed.

odd()

Reduce the set of matched elements to the odd ones in the set, numbered from zero.

off(*event*)

Remove an event handler.

offset(*coordinates*)

Get the current coordinates of the first element, or set the coordinates of every element, in the set of matched elements, relative to the document.

offsetParent()

Get the closest ancestor element that is positioned.

on(*event*, *callback*)

Attach an event handler function for one or more events to the selected elements.

one()

Attach a handler to an event for the elements. The handler is executed at most once per element per event type.

outerHeight()

Get the current computed outer height (including padding, border, and optionally margin) for the first element in the set of matched elements or set the outer height of every matched element.

outerWidth()

Get the current computed outer width (including padding, border, and optionally margin) for the first element in the set of matched elements or set the outer width of every matched element.

parent(*selector*)

Get the parent of each element in the current set of matched elements, optionally filtered by a selector.

parents(*selector*)

Get the ancestors of each element in the current set of matched elements, optionally filtered by a selector.

parentsUntil(*selector*)

Get the ancestors of each element in the current set of matched elements, up to but not including the element matched by the selector, DOM node, or dQuery object.

position()

Get the current coordinates of the first element in the set of matched elements, relative to the offset parent.

prepend(*html*)

Insert content, specified by the parameter, to the beginning of each element in the set of matched elements.

prependTo(*target*)

Insert every element in the set of matched elements to the beginning of the target.

prev(*selector*)

Get the immediately preceding sibling of each element in the set of matched elements. If a selector is provided, it retrieves the previous sibling only if it matches that selector.

prevAll(*selector*)

Get all preceding siblings of each element in the set of matched elements, optionally filtered by a selector.

prevUntil(*selector*)

Get all preceding siblings of each element up to but not including the element matched by the selector, DOM node, or dQuery object.

promise()

Return a Promise object to observe when all actions of a certain type bound to the collection, queued or not, have finished.

prop(*property, value*)

Get the value of a property for the first element in the set of matched elements or set one or more properties for every matched element.

pushStack(*stack*)

Add a collection of DOM elements onto the dQuery.

queue()

Show or manipulate the queue of functions to be executed on the matched elements.

ready(*callback*)

Specify a function to execute when the DOM is fully loaded.

remove(*items*)

Remove the set of matched elements from the DOM.

removeAttr(*attr: str*)

Remove an attribute from each element in the set of matched elements.

removeClass(*classname: str*)

Remove a single class, multiple classes, or all classes from each element in the set of matched elements.

removeData(*name: str*)

Remove a previously-stored piece of data.

removeProp(*prop: str*)

Remove a property for the set of matched elements.

replaceAll(*elements*)

Replace each target element with the set of matched elements.

replaceWith(*replacement*)

Replace each element in the set of matched elements with the provided new content and return the set of elements that was removed.

resize(*callback*)

Bind an event handler to the “resize” JavaScript event, or trigger that event on an element.

scroll(*callback*)

Bind an event handler to the “scroll” JavaScript event, or trigger that event on an element.

scrollLeft()

Get the current horizontal position of the scroll bar for the first element in the set of matched elements or set the horizontal position of the scroll bar for every matched element.

scrollTop()

Get the current vertical position of the scroll bar for the first element in the set of matched elements or set the vertical position of the scroll bar for every matched element.

select(*selector*)

Bind an event handler to the “select” JavaScript event, or trigger that event on an element.

serialize()

Encode a set of form elements as a string for submission.

serializeArray(*array*)

Encode an array of form elements as a string for submission.

show()

Display the matched elements.

siblings(*selector*)

Return the siblings of the matched elements. filter by selector.

size()

Return the number of elements in the dQuery object.

slice(*start, end*)

Return a new dQuery object containing the set of matched elements starting at the specified index and ending at the specified index.

slideDown()

Display the matched elements with a sliding motion.

slideToggle()

Display or hide the matched elements with a sliding motion.

slideUp()

Hide the matched elements with a sliding motion.

stop()

Stop the currently-running animation on the matched elements.

submit()

Bind an event handler to the “submit” JavaScript event, or trigger that event on an element.

text(*newVal: Optional[str] = None*)

Get the combined text contents of each element in the set of matched elements, including their descendants, or set the text contents of the matched elements.

toArray()

Retrieve all the elements contained in the dQuery set, as an array.

toggle()

Display or hide the matched elements.

toggleClass(*className*)

Add or remove one or more classes from each element in the set of matched elements

trigger(*eventName, eventArg=None*)

Execute all handlers and behaviors attached to the matched elements for the given event type.

triggerHandler()

Execute all handlers attached to an element for an event.

unbind()

Remove a previously-attached event handler from the elements.

undelegate()

Remove a handler from the event for all elements which match the current selector, based upon a specific set of root elements.

unload()

Bind an event handler to the “unload” JavaScript event.

unwrap()

Remove the parents of the set of matched elements from the DOM, leaving the matched elements in their place.

val(*newVal=None*)

Get the current value of the first element in the set of matched elements or set the value of every matched element.

width()

Get the current computed width for the first element in the set of matched elements or set the width of every matched element.

wrap(*wrappingElement*)

Wrap an HTML structure around each element in the set of matched elements.

wrapAll(*wrappingElement*)

Wrap an HTML structure around all elements in the set of matched elements.

wrapInner()

Wrap an HTML struct

class domonic.dQuery.o(*selector=None, *args, **kwargs*)

static Callbacks()

A multi-purpose callbacks list object that provides a powerful way to manage callback lists.

static Deferred()

A factory function that returns a chainable utility object with methods to register multiple callbacks into callback queues, invoke callback queues, and relay the success or failure state of any synchronous or asynchronous function.

static ajax(*url='/', type='GET', data=None, contentType=False, processData=False, cache=False, success=None, error=None*)

make an ajax request

static ajaxPrefilter()

Handle custom Ajax options or modify existing options before each request is sent and before they are processed by .ajax

static ajaxSetup()

Set default values for future Ajax requests. Its use is not recommended.

static ajaxTransport()

Creates an object that handles the actual transmission of Ajax data.

static contains(*parent, child*)

Check to see if a DOM element is a descendant of another DOM element.

static data()

Store arbitrary data associated with the specified element and/or return the value that was set.

static dequeue()

Execute the next function on the queue for the matched element.

static each(*arr, func*)

A generic iterator function, which can be used to seamlessly iterate over both objects and arrays.

static error(*msg*)

Takes a string and throws an exception containing it.

static escapeSelector(*selector*)

Returns a string with all special characters replaced with their respective character codes.

static extend(**args*)

Merge the contents of two or more objects together into the first object.

static get(*url: str, data=None, dataType=False, success=None, error=None*)

[simplified ajax request]

Parameters

- **url** (*str*) – [the url to request]
- **data** (*[type]*) – [the data to send]

- **dataType** (*bool*, *optional*) – [the dataType]. Defaults to False.
- **success** (*[type]*, *optional*) – [a success function]. Defaults to None.
- **error** (*[type]*, *optional*) – [an error method]. Defaults to None.

Returns

[the response]

Return type

[*type*]

static getJSON()

Load JSON-encoded data from the server using a GET HTTP request.

static getScript(*filename*, **args*)

execute another python file.

static globalEval(*code*)

Execute some python code globally.

static grep(*arr*, *func*)

Returns an array of elements from the original array which satisfy a filter function.

static hasData(*element*)

Determine whether an element has any dQuery data associated with it.

static holdReady()

Holds or releases the execution of dQuery's ready event.

static htmlPrefilter()

Modify and filter HTML strings passed through dQuery manipulation methods.

static inArray(*thing*, *arr*)

Search for a specified value within an array and return its index or -1 if not found.

static isArray(*item*)

Determine whether the argument is an array.

static isEmptyObject(*dct*)

Check to see if an object is empty (contains no enumerable properties).

static isFunction(*obj*)

Determines if its argument is callable as a function.

static isNumeric(*thing*)

Determine whether the argument is numeric.

static isPlainObject(*obj*)

Check to see if an object is a plain object created using '{ }'

static isWindow(*obj*)

Determine whether the argument is a window object.

static isXMLDoc(*obj*)

Check to see if a DOM node is within an XML document (or is an XML document).

static makeArray(*somelist*)

Convert an array-like object into a true JavaScript array.

static **map**(*arr, func*)

Translate all items in an array or object to new array of items.

static **merge**(*one, *args*)

Merge the contents of arrays into the first array.

static **noConflict**()

Relinquish dQuery's control of the `^` variable.

static **noop**()

An empty function.

static **now**()

Return a number representing the current time.

static **param**(*obj*)

Create a serialized representation of an array, a plain object, or a dQuery object suitable for use in a URL query string or Ajax request. In case a dQuery object is passed, it should contain input elements with name/value properties.

static **parseHTML**(*string*)

Parses a string into an array of DOM nodes.

static **parseJSON**(*string: str*)

Takes a well-formed JSON string and returns the resulting JavaScript value.

static **parseXML**(*string: str*)

Parses an XMLstring into a `pyml`

static **post**(*url, data, success*)

Send data to the server using a HTTP POST request.

static **proxy**(*func*)

Takes a function and returns a new one that will always have a particular context.

static **queue**(*func*)

Show or manipulate the queue of functions to be executed on the matched element.

static **readyException**()

Handles errors thrown synchronously in functions wrapped in dQuery

static **removeData**()

Remove a previously-stored piece of data.

static **sub**()

Creates a new copy of dQuery whose properties and methods can be modified without affecting the original dQuery object.

static **trim**(*content: str*) → *str*

Remove the whitespace from the beginning and end of a string.

static **unique**(*arr*)

[removes duplicate elements.]

Parameters

arr (*[type]*) – [list of elements]

Returns

[a sorted array without duplicates]

Return type
[*type*]

static uniqueSort(*arr*)

Sorts an array of DOM elements, in place, with the duplicates removed. Note that this only works on arrays of DOM elements, not strings or numbers.

static when()

Provides a way to execute callback functions based on zero or more Thenable objects, usually Deferred objects that represent asynchronous events.

1.4.7 d3

WARNING: this package is still in dev. expect bugs

d3 is another popular DOM manipulation library available in javascript. This is the python port of that.

It has several utilities for working with the DOM.

selection

d3's selection API is very powerful.

```
from domonic.dom import *
from domonic.html import *
from domonic.d3.selection import *

import domonic.d3 as d3
```

format

d3's has a library for formatting numbers to avoid things like floating precision

```
from domonic.dom import *
from domonic.html import *
from domonic.d3.format import *
from domonic.d3.format import format
```

dispatch

d3 has an event system

```
from domonic.dom import *
from domonic.html import *
from domonic.d3.dispatch import dispatch, Dispatch
```


path

d3's path API converts SVG paths to a d3.path object which is compatible with canvas.

```
from domonic.svg import *
from domonic.d3.path import Path
```

polygon

d3's has shape API

```
from domonic.d3.polygon import *
```

D3 expects a "polygon" is a 2D array of integers listed counterclockwise (clockwise works too). This does not work with the polygon as defined by SVG, SVG polygons have a points method you can parse into a 2D array and use here.

polygonArea() finds the area of a given polygon. It returns a float/int.

```
def test_polygonArea(self):
    irreg_0 = [[5,11],[12,4],[7,7],[6,1]]          # area: 15
    irreg_1 = [[-6,12],[23,2],[19,-8],[-7,-6]]     # area: 400
    irreg_2 = [[0,4],[12,8],[23,-5],[-5,-3]]       # area: 203
    square = [[0,4],[4,4],[4,0],[0,0]]             # area: 16
    triangle = [[-4,0],[0,4],[2,0]]                 # area: 12

    self.assertEqual(polygonArea(irreg_0), 15)
    self.assertEqual(polygonArea(irreg_1), 400)
    self.assertEqual(polygonArea(irreg_2), 203)
    self.assertEqual(polygonArea(square), 16)
    self.assertEqual(polygonArea(triangle), 12)
```

polygonCentroid() finds the center of mass of a given polygon. It returns 2 ints in an array, representing the point of the polygons center of mass.

```
def test_polygonCentroid(self):
    irreg = [[-4,0],[8,12],[4,8],[-4,-4],[0,0]]    # centroid: [0, 4]
    square = [[0,4],[4,4],[4,0],[0,0]]            # centroid: [2, 2]
    triangle = [[-4,0],[0,4],[4,2]]                # centroid: [0, 2]

    self.assertEqual(polygonCentroid(irreg), [0, 4])
    self.assertEqual(polygonCentroid(square), [2, 2])
    self.assertEqual(polygonCentroid(triangle), [0, 2])
```

polygonHull() takes a 2D array of points and determines the polygon hull using andrew's monotone chain algorithm. It returns the polygon hull as a 2D array of points.

```
def test_polygonHull(self):
    points_0 = [[0,6],[12,8],[23,-5],[-5,-3],[5,11],[12,4],[7,7],[6,1]]
    points_1 = [[-4,0],[8,12],[4,8],[-4,-4],[0,0],[-6,12],[23,2],[19,-8],[-7,-6]]

    hull_0 = [[-5,-3],[0,6],[5,11],[12,8],[23,-5]]
    hull_1 = [[-7,-6],[-6,12],[8,12],[19,-8],[23,2]]
```

(continues on next page)

(continued from previous page)

```
self.assertEqual(polygonHull(points_0), hull_0)
self.assertEqual(polygonHull(points_1), hull_1)
```

polygonContains() takes a polygon and a point, it returns true if the point is inside the polygon, false otherwise.

```
def test_polygonContains(self):
    irreg_0 = [[5,11],[12,4],[7,7],[6,1]]
    square = [[0,4],[4,4],[4,0],[0,0]]
    triangle = [[-4,0],[0,4],[2,0]]

    self.assertTrue(polygonContains(irreg_0, [6,8]))
    self.assertFalse(polygonContains(irreg_0, [0,0]))
    self.assertTrue(polygonContains(square, [2,2]))
    self.assertFalse(polygonContains(square, [-2,-2]))
    self.assertTrue(polygonContains(triangle, [0,1]))
    self.assertFalse(polygonContains(triangle, [0,-1]))
```

polygonLength() takes a polygon and returns the sum of the length of the polygon's sides.

```
def test_polygonLength(self):
    irreg_0 = [[5,11],[12,4],[7,7],[6,1]] # length = 31.863
    square = [[0,4],[4,4],[4,0],[0,0]] # length = 16
    triangle = [[-4,0],[0,4],[2,0]] # length ~ 16.129

    self.assertEqual(round(polygonLength(irreg_0), 3), 31.863)
    self.assertEqual(polygonLength(square), 16)
    self.assertEqual(round(polygonLength(triangle), 3), 16.129)
```

domonic.d3

1.4.8 JSON

decorate any function that returns python objects to return json instead

```
from domonic.JSON import *

@return_json
def somefunc():
    myObj = {"hi": [1,2,3]}
    return myObj

print( somefunc() )
print( is_json(somefunc()) )
```

convert json arrays into html tables...

```
import domonic.JSON as JSON

# i.e. containing flat json array of dicts... [{"id": "01", "name": "some item"}, {"id": "02", "name": "some other item"}]
```

(continues on next page)

(continued from previous page)

```
json_data = JSON.parse_file('somefile.json')
mytable = JSON.tablify(json_data)
print(mytable)
```

convert json arrays into csv files...

```
import domonic.JSON as JSON

json_data = JSON.parse_file('somefile.json')
JSON.csvify(json_data, 'data.csv')
```

convert csv files to json...

```
import domonic.JSON as JSON

json_data =JSON.csv2json("data.csv")
print(json_data)
```

more to come...

domonic.JSON

`domonic.JSON.csv2json(csv_filepath, json_filepath=None)`

convert a CSV to JSON.

`domonic.JSON.csvify(arr, outfile='data.csv')`

takes a json array and dumps a csv file

Parameters

- **arr** (*list*) – the json array
- **outfile** (*list*) – the output file

Returns

a csv file

Return type

str

`domonic.JSON.flatten(b, delim='__')`

i.e. input = map(lambda x: JSON.flatten(x, “__”), input)

`domonic.JSON.parse(json_string: str)`

[take a json string and return a python object]

Parameters

json_string (*str*) – [a json string]

Returns

[a python object]

Return type

[*type*]

`domonic.JSON.parse_file(filepath: str)`

[loads a json file and returns a python object]

Parameters

filepath (*str*) – [path to json file]

Returns

[a python object]

Return type

[*type*]

`domonic.JSON.stringify(data, filepath: Optional[str] = None, **kwargs)`

[stringify a python object]

Parameters

- **data** ([*type*]) – [the python object]
- **filepath** (*str*, *optional*) – [an optional filepath to save the stringified object] [default: None]

Returns

[the stringified object]

Return type

[*type*]

`domonic.JSON.tablify(arr)`

takes a json array and returns a html table # TODO - reverse. table to json

Parameters

arr (*list*) – the json array

Returns

a html table

Return type

str

1.4.9 constants

`domonic.constants` contains references to make your life a little easier.

Color

```
from domonic.constants.color import Color
print(Color.shit)
```

Color is also a util with a few methods...

```
c1 = Color('#ff00ff')
print(c1)

c2 = Color(255,255,255)
print(c1)
```

```
class domonic.constants.color.Color(*args, **kwargs)
```

```
    Color Functions
```

```
    AliceBlue:  str = '#F0F8FF'
```

```
    AntiqueWhite:  str = '#FAEBD7'
```

```
    Aqua:  str = '#00FFFF'
```

```
    Aquamarine:  str = '#7FFFD4'
```

```
    Azure:  str = '#F0FFFF'
```

```
    Beige:  str = '#F5F5DC'
```

```
    Bisque:  str = '#FFE4C4'
```

```
    Black:  str = '#000000'
```

```
    BlanchedAlmond:  str = '#FFEBCD'
```

```
    Blue:  str = '#0000FF'
```

```
    BlueViolet:  str = '#8A2BE2'
```

```
    Brown:  str = '#A52A2A'
```

```
    BurlyWood:  str = '#DEB887'
```

```
    CadetBlue:  str = '#5F9EA0'
```

```
    Chartreuse:  str = '#7FFF00'
```

```
    Chocolate:  str = '#D2691E'
```

```
    Coral:  str = '#FF7F50'
```

```
    CornflowerBlue:  str = '#6495ED'
```

```
    Cornsilk:  str = '#FFF8DC'
```

```
    Crimson:  str = '#DC143C'
```

```
    Cyan:  str = '#00FFFF'
```

```
    DarkBlue:  str = '#00008B'
```

```
    DarkCyan:  str = '#008B8B'
```

```
    DarkGoldenRod:  str = '#B8860B'
```

```
    DarkGray:  str = '#A9A9A9'
```

```
    DarkGreen:  str = '#006400'
```

```
    DarkGrey:  str = '#A9A9A9'
```

```
    DarkKhaki:  str = '#BDB76B'
```

```
    DarkMagenta:  str = '#8B008B'
```

```
DarkOliveGreen: str = '#556B2F'
DarkOrange: str = '#FF8C00'
DarkOrchid: str = '#9932CC'
DarkRed: str = '#8B0000'
DarkSalmon: str = '#E9967A'
DarkSeaGreen: str = '#8FBC8F'
DarkSlateBlue: str = '#483D8B'
DarkSlateGray: str = '#2F4F4F'
DarkSlateGrey: str = '#2F4F4F'
DarkTurquoise: str = '#00CED1'
DarkViolet: str = '#9400D3'
DeepPink: str = '#FF1493'
DeepSkyBlue: str = '#00BFFF'
DimGray: str = '#696969'
DimGrey: str = '#696969'
DodgerBlue: str = '#1E90FF'
FireBrick: str = '#B22222'
FloralWhite: str = '#FFFAF0'
ForestGreen: str = '#228B22'
Fuchsia: str = '#FF00FF'
Gainsboro: str = '#DCDCDC'
GhostWhite: str = '#F8F8FF'
Gold: str = '#FFD700'
GoldenRod: str = '#DAA520'
Gray: str = '#808080'
Green: str = '#008000'
GreenYellow: str = '#ADFF2F'
Grey: str = '#808080'
HoneyDew: str = '#F0FFF0'
HotPink: str = '#FF69B4'
IndianRed: str = '#CD5C5C'
```

```
Indigo:  str = '#4B0082'
Ivory:   str = '#FFFFFF0'
Khaki:   str = '#F0E68C'
Lavender: str = '#E6E6FA'
LavenderBlush: str = '#FFF0F5'
LawnGreen: str = '#7CFC00'
LemonChiffon: str = '#FFFACD'
LightBlue: str = '#ADD8E6'
LightCoral: str = '#F08080'
LightCyan: str = '#E0FFFF'
LightGoldenRodYellow: str = '#FAFAD2'
LightGray: str = '#D3D3D3'
LightGreen: str = '#90EE90'
LightGrey: str = '#D3D3D3'
LightPink: str = '#FFB6C1'
LightSalmon: str = '#FFA07A'
LightSeaGreen: str = '#20B2AA'
LightSkyBlue: str = '#87CEFA'
LightSlateGray: str = '#778899'
LightSlateGrey: str = '#778899'
LightSteelBlue: str = '#B0C4DE'
LightYellow: str = '#FFFFE0'
Lime: str = '#00FF00'
LimeGreen: str = '#32CD32'
Linen: str = '#FAF0E6'
Magenta: str = '#FF00FF'
Maroon: str = '#800000'
MediumAquaMarine: str = '#66CDAA'
MediumBlue: str = '#0000CD'
MediumOrchid: str = '#BA55D3'
MediumPurple: str = '#9370DB'
```

```
MediumSeaGreen:  str = '#3CB371'
MediumSlateBlue: str = '#7B68EE'
MediumSpringGreen: str = '#00FA9A'
MediumTurquoise: str = '#48D1CC'
MediumVioletRed: str = '#C71585'
MidnightBlue:   str = '#191970'
MintCream:      str = '#F5FFFA'
MistyRose:      str = '#FFE4E1'
Moccasin:       str = '#FFE4B5'
NavajoWhite:    str = '#FFDEAD'
Navy:           str = '#000080'
OldLace:        str = '#FDF5E6'
Olive:          str = '#808000'
OliveDrab:      str = '#6B8E23'
Orange:         str = '#FFA500'
OrangeRed:      str = '#FF4500'
Orchid:         str = '#DA70D6'
PaleGoldenRod:  str = '#EEE8AA'
PaleGreen:      str = '#98FB98'
PaleTurquoise:  str = '#AFEEEE'
PaleVioletRed:  str = '#DB7093'
PapayaWhip:     str = '#FFEFD5'
PeachPuff:      str = '#FFDAB9'
Peru:           str = '#CD853F'
Pink:           str = '#FFC0CB'
Plum:           str = '#DDA0DD'
PowderBlue:     str = '#B0E0E6'
Purple:         str = '#800080'
RebeccaPurple:  str = '#663399'
Red:            str = '#FF0000'
RosyBrown:      str = '#BC8F8F'
```



```
RoyalBlue:  str = '#4169E1'
SaddleBrown: str = '#8B4513'
Salmon:     str = '#FA8072'
SandyBrown: str = '#F4A460'
SeaGreen:   str = '#2E8B57'
SeaShell:   str = '#FFF5EE'
Sienna:     str = '#A0522D'
Silver:     str = '#C0C0C0'
SkyBlue:    str = '#87CEEB'
SlateBlue:  str = '#6A5ACD'
SlateGray:  str = '#708090'
SlateGrey:  str = '#708090'
Snow:       str = '#FFFAFA'
SpringGreen: str = '#00FF7F'
SteelBlue:  str = '#4682B4'
Tan:        str = '#D2B48C'
Teal:       str = '#008080'
Thistle:    str = '#D8BFD8'
Tomato:     str = '#FF6347'
Turquoise:  str = '#40E0D0'
Violet:     str = '#EE82EE'
Wheat:      str = '#F5DEB3'
White:      str = '#FFFFFF'
WhiteSmoke: str = '#F5F5F5'
Yellow:     str = '#FFFF00'
YellowGreen: str = '#9ACD32'
acidgreen:  str = '#8ffe09'
adobe:      str = '#bd6c48'
algae:      str = '#54ac68'
algaegreen: str = '#21c36f'
almostblack: str = '#070d0d'
```

```
amber: str = '#feb308'
amethyst: str = '#9b5fc0'
apple: str = '#6ecb3c'
applegreen: str = '#76cd26'
apricot: str = '#ffb16d'
aqua: str = '#13eac9'
aquablue: str = '#02d8e9'
aquagreen: str = '#12e193'
aquamarine: str = '#2ee8bb'
armygreen: str = '#4b5d16'
asparagus: str = '#77ab56'
aubergine: str = '#3d0734'
auburn: str = '#9a3001'
avocado: str = '#90b134'
avocadogreen: str = '#87a922'
azul: str = '#1d5dec'
azure: str = '#069af3'
babyblue: str = '#a2cffe'
babygreen: str = '#8cff9e'
babypink: str = '#ffb7ce'
babypoo: str = '#ab9004'
babypoop: str = '#937c00'
babypoopgreen: str = '#8f9805'
babypukegreen: str = '#b6c406'
babypurple: str = '#ca9bf7'
babyshitbrown: str = '#ad900d'
babyshitgreen: str = '#889717'
banana: str = '#ffff7e'
bananayellow: str = '#fafa4b'
barbiepink: str = '#fe46a5'
barfgreen: str = '#94ac02'
```

```
barney:  str = '#ac1db8'
barneypurple:  str = '#a00498'
battleshipgrey:  str = '#6b7c85'
beige:  str = '#e6daa6'
berry:  str = '#990f4b'
bile:  str = '#b5c306'
black:  str = '#000000'
bland:  str = '#afa88b'
blood:  str = '#770001'
bloodorange:  str = '#fe4b03'
bloodred:  str = '#980002'
blue:  str = '#0343df'
blueberry:  str = '#464196'
blueblue:  str = '#2242c7'
bluegreen:  str = '#0f9b8e'
bluegrey:  str = '#85a3b2'
bluepurple:  str = '#5a06ef'
blueviolet:  str = '#5d06e9'
bluewithahintofpurple:  str = '#533cc6'
blueygreen:  str = '#2bb179'
blueygrey:  str = '#89a0b0'
blueypurple:  str = '#6241c7'
bluish:  str = '#2976bb'
bluishgreen:  str = '#10a674'
bluishgrey:  str = '#748b97'
bluishpurple:  str = '#703be7'
blurple:  str = '#5539cc'
blush:  str = '#f29e8e'
blushpink:  str = '#fe828c'
booger:  str = '#9bb53c'
boogergreen:  str = '#96b403'
```

```
bordeaux:  str = '#7b002c'
boringgreen: str = '#63b365'
bottlegreen: str = '#044a05'
brick:  str = '#a03623'
brickorange: str = '#c14a09'
brickred: str = '#8f1402'
brightaqua: str = '#0bf9ea'
brightblue: str = '#0165fc'
brightcyan: str = '#41fdfe'
brightgreen: str = '#01ff07'
brightlavender: str = '#c760ff'
brightlightblue: str = '#26f7fd'
brightlightgreen: str = '#2dfe54'
brightlilac: str = '#c95efb'
brightlime: str = '#87fd05'
brightlimegreen: str = '#65fe08'
brightmagenta: str = '#ff08e8'
brightolive: str = '#9cbb04'
brightorange: str = '#ff5b00'
brightpink: str = '#fe01b1'
brightpurple: str = '#be03fd'
brightred: str = '#ff000d'
brightseagreen: str = '#05ffa6'
brightskyblue: str = '#02ccfe'
brightteal: str = '#01f9c6'
brightturquoise: str = '#0ffef9'
brightviolet: str = '#ad0afd'
brightyellow: str = '#fffd01'
brightyellowgreen: str = '#9dff00'
britishracinggreen: str = '#05480d'
bronze: str = '#a87900'
```

```
brown: str = '#653700'
browngreen: str = '#706c11'
browngrey: str = '#8d8468'
brownish: str = '#9c6d57'
brownishgreen: str = '#6a6e09'
brownishgrey: str = '#86775f'
brownishorange: str = '#cb7723'
brownishpink: str = '#c27e79'
brownishpurple: str = '#76424e'
brownishred: str = '#9e3623'
brownishyellow: str = '#c9b003'
brownorange: str = '#b96902'
brownred: str = '#922b05'
brownyellow: str = '#b29705'
brownyellowgreen: str = '#6f6c0a'
brownyelloworange: str = '#ca6b02'
bruise: str = '#7e4071'
bubblegum: str = '#ff6cb5'
bubblegumpink: str = '#ff69af'
buff: str = '#fef69e'
burgundy: str = '#610023'
burntorange: str = '#c04e01'
burntred: str = '#9f2305'
burntsiena: str = '#b75203'
burntsienna: str = '#b04e0f'
burntumber: str = '#a0450e'
burntyellow: str = '#d5ab09'
burple: str = '#6832e3'
butter: str = '#ffff81'
butterscotch: str = '#fdb147'
butteryellow: str = '#fffd74'
```

```
cadetblue: str = '#4e7496'
camel: str = '#c69f59'
camo: str = '#7f8f4e'
camogreen: str = '#526525'
camouflagegreen: str = '#4b6113'
canary: str = '#fdff63'
canaryyellow: str = '#fffe40'
candypink: str = '#ff63e9'
caramel: str = '#af6f09'
carmine: str = '#9d0216'
carnation: str = '#fd798f'
carnationpink: str = '#ff7fa7'
carolinablue: str = '#8ab8fe'
celadon: str = '#befdb7'
celery: str = '#c1fd95'
cement: str = '#a5a391'
cerise: str = '#de0c62'
cerulean: str = '#0485d1'
ceruleanblue: str = '#056eee'
charcoal: str = '#343837'
charcoalgrey: str = '#3c4142'
chartreuse: str = '#c1f80a'
cherry: str = '#cf0234'
cherryred: str = '#f7022a'
chestnut: str = '#742802'
chocolate: str = '#3d1c02'
chocolatebrown: str = '#411900'
cinnamon: str = '#ac4f06'
claret: str = '#680018'
clay: str = '#b66a50'
claybrown: str = '#b2713d'
```

clearblue: `str = '#247afd'`

cobalt: `str = '#1e488f'`

cobaltblue: `str = '#030aa7'`

cocoa: `str = '#875f42'`

coffee: `str = '#a6814c'`

`convert(to: str)`

convert the color to a different color space

Parameters

`to ([str])` – [can be one of the following: 'rgb', 'hsl', 'hsv', 'hex']

coolblue: `str = '#4984b8'`

coolgreen: `str = '#33b864'`

coolgrey: `str = '#95a3a6'`

copper: `str = '#b66325'`

coral: `str = '#fc5a50'`

coralpink: `str = '#ff6163'`

cornflower: `str = '#6a79f7'`

cornflowerblue: `str = '#5170d7'`

cranberry: `str = '#9e003a'`

cream: `str = '#ffffc2'`

creme: `str = '#ffffb6'`

crimson: `str = '#8c000f'`

custard: `str = '#fffd78'`

cyan: `str = '#00ffff'`

dandelion: `str = '#fedf08'`

dark: `str = '#1b2431'`

darkaqua: `str = '#05696b'`

darkaquamarine: `str = '#017371'`

darkbeige: `str = '#ac9362'`

darkblue: `str = '#030764'`

darkbluegreen: `str = '#005249'`

darkbluegrey: `str = '#1f3b4d'`

darkbrown: `str = '#341c02'`

```
darkcoral: str = '#cf524e'
darkcream: str = '#fff39a'
darkcyan: str = '#0a888a'
darkforestgreen: str = '#002d04'
darkfuchsia: str = '#9d0759'
darkgold: str = '#b59410'
darkgrassgreen: str = '#388004'
darkgreen: str = '#054907'
darkgreenblue: str = '#1f6357'
darkgrey: str = '#363737'
darkgreyblue: str = '#29465b'
darkhotpink: str = '#d90166'
darkindigo: str = '#1f0954'
darkishblue: str = '#014182'
darkishgreen: str = '#287c37'
darkishpink: str = '#da467d'
darkishpurple: str = '#751973'
darkishred: str = '#a90308'
darkkhaki: str = '#9b8f55'
darklavender: str = '#856798'
darklilac: str = '#9c6da5'
darklime: str = '#84b701'
darklimegreen: str = '#7ebd01'
darkmagenta: str = '#960056'
darkmaroon: str = '#3c0008'
darkmauve: str = '#874c62'
darkmint: str = '#48c072'
darkmintgreen: str = '#20c073'
darkmustard: str = '#a88905'
darknavy: str = '#000435'
darknavyblue: str = '#00022e'
```



```
darkolive: str = '#373e02'
darkolivegreen: str = '#3c4d03'
darkorange: str = '#c65102'
darkpastelgreen: str = '#56ae57'
darkpeach: str = '#de7e5d'
darkperiwinkle: str = '#665fd1'
darkpink: str = '#cb416b'
darkplum: str = '#3f012c'
darkpurple: str = '#35063e'
darkred: str = '#840000'
darkrose: str = '#b5485d'
darkroyalblue: str = '#02066f'
darksage: str = '#598556'
darksalmon: str = '#c85a53'
darksand: str = '#a88f59'
darkseafoam: str = '#1fb57a'
darkseafoamgreen: str = '#3eaf76'
darkseagreen: str = '#11875d'
darkskyblue: str = '#448ee4'
darkslateblue: str = '#214761'
darktan: str = '#af884a'
darktaupe: str = '#7f684e'
darkteal: str = '#014d4e'
darkturquoise: str = '#045c5a'
darkviolet: str = '#34013f'
darkyellow: str = '#d5b60a'
darkyellowgreen: str = '#728f02'
deepaqua: str = '#08787f'
deepblue: str = '#040273'
deepbrown: str = '#410200'
deepgreen: str = '#02590f'
```

```
deeplavender: str = '#8d5eb7'
deeplilac: str = '#966ebd'
deepmagenta: str = '#a0025c'
deeporange: str = '#dc4d01'
deeppink: str = '#cb0162'
deeppurple: str = '#36013f'
deepred: str = '#9a0200'
deeprose: str = '#c74767'
deepseablue: str = '#015482'
deepskyblue: str = '#0d75f8'
deepteal: str = '#00555a'
deepturquoise: str = '#017374'
deepviolet: str = '#490648'
denim: str = '#3b638c'
denimblue: str = '#3b5b92'
desert: str = '#ccad60'
diarrhea: str = '#9f8303'
dirt: str = '#8a6e45'
dirtbrown: str = '#836539'
dirtyblue: str = '#3f829d'
dirtygreen: str = '#667e2c'
dirtyorange: str = '#c87606'
dirtypink: str = '#ca7b80'
dirtypurple: str = '#734a65'
dirtyyellow: str = '#cdc50a'
dodgerblue: str = '#3e82fc'
drab: str = '#828344'
drabgreen: str = '#749551'
driedblood: str = '#4b0101'
duckeggblue: str = '#c3fbf4'
dullblue: str = '#49759c'
```

```
dullbrown:  str = '#876e4b'
dullgreen:  str = '#74a662'
dullorange: str = '#d8863b'
dullpink:   str = '#d5869d'
dullpurple: str = '#84597e'
dullred:    str = '#bb3f3f'
dullteal:   str = '#5f9e8f'
dullyellow: str = '#eedc5b'
dusk:       str = '#4e5481'
duskblue:   str = '#26538d'
duskyblue:  str = '#475f94'
duskypink:  str = '#cc7a8b'
duskypurple: str = '#895b7b'
duskyrose:  str = '#ba6873'
dust:       str = '#b2996e'
dustyblue:  str = '#5a86ad'
dustygreen: str = '#76a973'
dustylavender: str = '#ac86a8'
dustyorange: str = '#f0833a'
dustypink:  str = '#d58a94'
dustypurple: str = '#825f87'
dustyred:   str = '#b9484e'
dustyrose:  str = '#c0737a'
dustyteal:  str = '#4c9085'
earth:      str = '#a2653e'
eastergreen: str = '#8cfd7e'
easterpurple: str = '#c071fe'
ecru:       str = '#feffca'
eggplant:   str = '#380835'
eggplantpurple: str = '#430541'
eggshell:   str = '#fffcc4'
```

```
eggshellblue: str = '#c4fff7'
electricblue: str = '#0652ff'
electricgreen: str = '#21fc0d'
electriclime: str = '#a8ff04'
electricpink: str = '#ff0490'
electricpurple: str = '#aa23ff'
emerald: str = '#01a049'
emeraldgreen: str = '#028f1e'
evergreen: str = '#05472a'
fadedblue: str = '#658cbb'
fadedgreen: str = '#7bb274'
fadedorange: str = '#f0944d'
fadedpink: str = '#de9dac'
fadedpurple: str = '#916e99'
fadedred: str = '#d3494e'
fadedyellow: str = '#feff7f'
fawn: str = '#cfaf7b'
fern: str = '#63a950'
ferngreen: str = '#548d44'
fireengineered: str = '#fe0002'
flatblue: str = '#3c73a8'
flatgreen: str = '#699d4c'
fluorescentgreen: str = '#08ff08'
flurogreen: str = '#0aff02'
foamgreen: str = '#90fda9'
forest: str = '#0b5509'
forestgreen: str = '#06470c'
forrestgreen: str = '#154406'
frenchblue: str = '#436bad'
freshgreen: str = '#69d84f'
froggreen: str = '#58bc08'
```

static fromHex(*hex: str*) → Color

create a Color from a hex string i.e. #ffffff

static fromRGBA(*r: int, g: int, b: int, a: int = 255*) → Color

[creates a Color from rgba values]

Parameters

- **r** ([*int*]) – [a value between 0 and 255]
- **g** ([*int*]) – [a value between 0 and 255]
- **b** ([*int*]) – [a value between 0 and 255]
- **a** ([*int*]) – [a value between 0 and 255]

Returns

[a Color object]

Return type

[*type*]

fuchsia: **str** = '#ed0dd9'

gold: **str** = '#d4b48c'

golden: **str** = '#f5bf03'

goldenbrown: **str** = '#b27a01'

goldenrod: **str** = '#f9bc08'

goldenyellow: **str** = '#fec615'

grape: **str** = '#6c3461'

grapefruit: **str** = '#fd5956'

grapepurple: **str** = '#5d1451'

grass: **str** = '#5cac2d'

grassgreen: **str** = '#3f9b0b'

grassygreen: **str** = '#419c03'

green: **str** = '#15b01a'

greenapple: **str** = '#5edc1f'

greenblue: **str** = '#01c08d'

greenbrown: **str** = '#544e03'

greengrey: **str** = '#77926f'

greenish: **str** = '#40a368'

greenishbeige: **str** = '#c9d179'

greenishblue: **str** = '#0b8b87'

```
greenishbrown: str = '#696112'
greenishcyan: str = '#2afeb7'
greenishgrey: str = '#96ae8d'
greenishtan: str = '#bccb7a'
greenishteal: str = '#32bf84'
greenishturquoise: str = '#00fbb0'
greenishyellow: str = '#cdfd02'
greenteal: str = '#0cb577'
greenyblue: str = '#42b395'
greenybrown: str = '#696006'
greenyyellow: str = '#b5ce08'
greenygrey: str = '#7ea07a'
greenyyellow: str = '#c6f808'
grey: str = '#929591'
greyblue: str = '#647d8e'
greybrown: str = '#7f7053'
greengreen: str = '#86a17d'
greyish: str = '#a8a495'
greyishblue: str = '#5e819d'
greyishbrown: str = '#7a6a4f'
greyishgreen: str = '#82a67d'
greyishpink: str = '#c88d94'
greyishpurple: str = '#887191'
greyishteal: str = '#719f91'
greypink: str = '#c3909b'
greypurple: str = '#826d8c'
greyteal: str = '#5e9b8a'
grossgreen: str = '#a0bf16'
gunmetal: str = '#536267'
```

hasAlpha() → bool

[does the color have an alpha channel]

Returns

[True if alpha channel exists else False]

Return type

[bool]

hazel: `str = '#8e7618'`

heather: `str = '#a484ac'`

heliotrope: `str = '#d94ff5'`

static hex2rgb(h: str) → Tuple[int, int, int]

[takes a hex color in the form of #RRGGBB and returns the rgb values as a tuple i.e (r, g, b)]

Parameters

h ([str]) – [hex string i.e #ffffff]

Returns

[rgb tuple i.e. (255, 255, 255)]

Return type

[tuple]

highlightergreen: `str = '#1bfc06'`

hospitalgreen: `str = '#9be5aa'`

hotgreen: `str = '#25ff29'`

hotmagenta: `str = '#f504c9'`

hotpink: `str = '#ff028d'`

hotpurple: `str = '#cb00f5'`

huntergreen: `str = '#0b4008'`

ice: `str = '#d6fffa'`

iceblue: `str = '#d7fffe'`

ickygreen: `str = '#8fae22'`

indianred: `str = '#850e04'`

indigo: `str = '#380282'`

indigoblue: `str = '#3a18b1'`

iris: `str = '#6258c4'`

irishgreen: `str = '#019529'`

ivory: `str = '#ffffcb'`

jade: `str = '#1fa774'`

```
jadegreen: str = '#2baf6a'
junglegreen: str = '#048243'
kelleygreen: str = '#009337'
kellygreen: str = '#02ab2e'
kermitgreen: str = '#5cb200'
keylime: str = '#aeff6e'
khaki: str = '#aaa662'
khakigreen: str = '#728639'
kiwi: str = '#9cef43'
kiwigreen: str = '#8ee53f'
lavender: str = '#c79fef'
lavenderblue: str = '#8b88f8'
lavenderpink: str = '#dd85d7'
lawngreen: str = '#4da409'
leaf: str = '#71aa34'
leafgreen: str = '#5ca904'
leafygreen: str = '#51b73b'
leather: str = '#ac7434'
lemon: str = '#fdff52'
lemongreen: str = '#adf802'
lemonlime: str = '#bffe28'
lemonyellow: str = '#fdff38'
lichen: str = '#8fb67b'
lightaqua: str = '#8cffdb'
lightaquamarine: str = '#7bfdc7'
lightbeige: str = '#fffeb6'
lightblue: str = '#7bc8f6'
lightbluegreen: str = '#7efbb3'
lightbluegrey: str = '#b7c9e2'
lightbluishgreen: str = '#76fda8'
lightbrightgreen: str = '#53fe5c'
```



```
lightbrown:  str = '#ad8150'
lightburgundy: str = '#a8415b'
lightcyan:   str = '#acfffc'
lighteggplant: str = '#894585'
lightergreen: str = '#75fd63'
lighterpurple: str = '#a55af4'
lightforestgreen: str = '#4f9153'
lightgold:   str = '#fddc5c'
lightgrassgreen: str = '#9af764'
lightgreen:  str = '#76ff7b'
lightgreenblue: str = '#56fca2'
lightgreenishblue: str = '#63f7b4'
lightgrey:   str = '#d8dcd6'
lightgreyblue: str = '#9dbcd4'
lightgreygreen: str = '#b7e1a1'
lightindigo:  str = '#6d5acf'
lightishblue: str = '#3d7afd'
lightishgreen: str = '#61e160'
lightishpurple: str = '#a552e6'
lightishred:  str = '#fe2f4a'
lightkhaki:   str = '#e6f2a2'
lightlavendar: str = '#efc0fe'
lightlavender: str = '#dfc5fe'
lightlightblue: str = '#caffff'
lightlightgreen: str = '#c8ffb0'
lightlilac:   str = '#edc8ff'
lightlime:    str = '#aefd6c'
lightlimegreen: str = '#b9ff66'
lightmagenta: str = '#fa5ff7'
lightmaroon:  str = '#a24857'
lightmauve:   str = '#c292a1'
```

```
lightmint: str = '#b6ffbb'
lightmintgreen: str = '#a6fbb2'
lightmossgreen: str = '#a6c875'
lightmustard: str = '#f7d560'
lightnavy: str = '#155084'
lightnavyblue: str = '#2e5a88'
lightneongreen: str = '#4efd54'
lightolive: str = '#acbf69'
lightolivegreen: str = '#a4be5c'
lightorange: str = '#fdaa48'
lightpastelgreen: str = '#b2fba5'
lightpeach: str = '#ffd8b1'
lightpeagreen: str = '#c4fe82'
lightperiwinkle: str = '#c1c6fc'
lightpink: str = '#ffd1df'
lightplum: str = '#9d5783'
lightpurple: str = '#bf77f6'
lightred: str = '#ff474c'
lightrose: str = '#ffc5cb'
lightroyalblue: str = '#3a2efe'
lightsage: str = '#bcecac'
lightsalmon: str = '#fea993'
lightseafoam: str = '#a0febf'
lightseafoamgreen: str = '#a7ffb5'
lightseagreen: str = '#98f6b0'
lightskyblue: str = '#c6fcff'
lighttan: str = '#fbeeac'
lightteal: str = '#90e4c1'
lightturquoise: str = '#7ef4cc'
lightviolet: str = '#d6b4fc'
```

```
lightyellow: str = '#ffffe7a'
lightyellowgreen: str = '#ccfd7f'
lightyellowishgreen: str = '#c2ff89'
lilac: str = '#cea2fd'
lilac: str = '#c48efd'
lime: str = '#aaff32'
limegreen: str = '#89fe05'
limeyellow: str = '#d0fe1d'
lipstick: str = '#d5174e'
lipstickred: str = '#c0022f'
macaroniandcheese: str = '#efb435'
magenta: str = '#c20078'
mahogany: str = '#4a0100'
maize: str = '#f4d054'
mango: str = '#ffa62b'
manilla: str = '#fffa86'
marigold: str = '#fcc006'
marine: str = '#042e60'
marineblue: str = '#01386a'
maroon: str = '#650021'
mauve: str = '#ae7181'
mediumblue: str = '#2c6fbb'
mediumbrown: str = '#7f5112'
mediumgreen: str = '#39ad48'
mediumgrey: str = '#7d7f7c'
mediumpink: str = '#f36196'
mediumpurple: str = '#9e43a2'
melon: str = '#ff7855'
merlot: str = '#730039'
metallicblue: str = '#4f738e'
midblue: str = '#276ab3'
```

```
midgreen: str = '#50a747'
midnight: str = '#03012d'
midnightblue: str = '#020035'
midnightpurple: str = '#280137'
militarygreen: str = '#667c3e'
milkchocolate: str = '#7f4e1e'
mint: str = '#9ffeb0'
mintgreen: str = '#8fff9f'
mintygreen: str = '#0bf77d'
mocha: str = '#9d7651'
moss: str = '#769958'
mossgreen: str = '#658b38'
mossygreen: str = '#638b27'
mud: str = '#735c12'
mudbrown: str = '#60460f'
muddybrown: str = '#886806'
muddygreen: str = '#657432'
muddyyellow: str = '#bfac05'
mudgreen: str = '#606602'
mulberry: str = '#920a4e'
murkygreen: str = '#6c7a0e'
mushroom: str = '#ba9e88'
mustard: str = '#ceb301'
mustardbrown: str = '#ac7e04'
mustardgreen: str = '#a8b504'
mustardyellow: str = '#d2bd0a'
mutedblue: str = '#3b719f'
mutedgreen: str = '#5fa052'
mutedpink: str = '#d1768f'
mutedpurple: str = '#805b87'
nastygreen: str = '#70b23f'
```

```
navy: str = '#01153e'
navyblue: str = '#001146'
navygreen: str = '#35530a'
neonblue: str = '#04d9ff'
neongreen: str = '#0cff0c'
neonpink: str = '#fe019a'
neonpurple: str = '#bc13fe'
neonred: str = '#ff073a'
neonyellow: str = '#cfff04'
niceblue: str = '#107ab0'
nightblue: str = '#040348'
ocean: str = '#017b92'
oceanblue: str = '#03719c'
oceangreen: str = '#3d9973'
ocher: str = '#bf9b0c'
ochre: str = '#bf9005'
ocre: str = '#c69c04'
offblue: str = '#5684ae'
offgreen: str = '#6ba353'
offwhite: str = '#ffffe4'
offyellow: str = '#f1f33f'
oldpink: str = '#c77986'
oldrose: str = '#c87f89'
olive: str = '#6e750e'
olivebrown: str = '#645403'
olivedrab: str = '#6f7632'
olivegreen: str = '#677a04'
oliveyellow: str = '#c2b709'
orange: str = '#f97306'
orangebrown: str = '#be6400'
orangeish: str = '#fd8d49'
```

```
orangepink: str = '#ff6f52'
orangered: str = '#fe420f'
orangeybrown: str = '#b16002'
orangeyellow: str = '#ffad01'
orangeyred: str = '#fa4224'
orangeyyellow: str = '#fdb915'
orangish: str = '#fc824a'
orangishbrown: str = '#b25f03'
orangishred: str = '#f43605'
orchid: str = '#c875c4'
pale: str = '#fff9d0'
paleaqua: str = '#b8ffeb'
paleblue: str = '#d0fefe'
palebrown: str = '#b1916e'
palecyan: str = '#b7fffa'
palegold: str = '#fdde6c'
palegreen: str = '#c7fdb5'
palegrey: str = '#fdfdfe'
palelavender: str = '#eecffe'
palelightgreen: str = '#b1fc99'
palelilac: str = '#e4cbff'
palelime: str = '#befd73'
palelimegreen: str = '#b1ff65'
palemagenta: str = '#d767ad'
palemauve: str = '#fed0fc'
paleolive: str = '#b9cc81'
paleolivegreen: str = '#b1d27b'
paleorange: str = '#ffa756'
palepeach: str = '#ffe5ad'
palepink: str = '#ffcfdc'
palepurple: str = '#b790d4'
```

```
palered:  str = '#d9544d'
palerose: str = '#fdc1c5'
palesalmon: str = '#ffb19a'
paleskyblue: str = '#bdf6fe'
paleteal: str = '#82cbb2'
paleturquoise: str = '#a5fbd5'
paleviolet: str = '#ceaefa'
paleyellow: str = '#ffff84'
parchment: str = '#fefcaf'
pastelblue: str = '#a2bffe'
pastelgreen: str = '#b0ff9d'
pastelorange: str = '#ff964f'
pastelpink: str = '#ffbacd'
pastelpurple: str = '#caa0ff'
pastelred: str = '#db5856'
pastelyellow: str = '#fffe71'
pea: str = '#a4bf20'
peach: str = '#ffb07c'
peachypink: str = '#ff9a8a'
peacockblue: str = '#016795'
peagreen: str = '#8eab12'
pear: str = '#cbf85f'
peasoup: str = '#929901'
peasoupgreen: str = '#94a617'
periwinkle: str = '#8e82fe'
periwinkleblue: str = '#8f99fb'
perrywinkle: str = '#8f8ce7'
petrol: str = '#005f6a'
pigpink: str = '#e78ea5'
pine: str = '#2b5d34'
pinegreen: str = '#0a481e'
```

```
pink: str = '#ff81c0'
pinkish: str = '#d46a7e'
pinkishbrown: str = '#b17261'
pinkishgrey: str = '#c8aca9'
pinkishorange: str = '#ff724c'
pinkishpurple: str = '#d648d7'
pinkishred: str = '#f10c45'
pinkishtan: str = '#d99b82'
pinkpurple: str = '#ef1de7'
pinkred: str = '#f5054f'
pinky: str = '#fc86aa'
pinkypurple: str = '#c94cbe'
pinkyred: str = '#fc2647'
pissyyellow: str = '#ddd618'
pistachio: str = '#c0fa8b'
plum: str = '#580f41'
plumpurple: str = '#4e0550'
poisongreen: str = '#40fd14'
poo: str = '#8f7303'
poobrown: str = '#885f01'
poop: str = '#7f5e00'
poopbrown: str = '#7a5901'
poopgreen: str = '#6f7c00'
powderblue: str = '#b1d1fc'
powderpink: str = '#ffb2d0'
primaryblue: str = '#0804f9'
prussianblue: str = '#004577'
puce: str = '#a57e52'
puke: str = '#a5a502'
pukebrown: str = '#947706'
pukegreen: str = '#9aae07'
```



```
pukeyellow: str = '#c2be0e'
pumpkin: str = '#e17701'
pumpkinorange: str = '#fb7d07'
pureblue: str = '#0203e2'
purple: str = '#7e1e9c'
purpleblue: str = '#5d21d0'
purplebrown: str = '#673a3f'
purplegrey: str = '#866f85'
purpleish: str = '#98568d'
purpleishblue: str = '#6140ef'
purpleishpink: str = '#df4ec8'
purplepink: str = '#d725de'
purplered: str = '#990147'
purpley: str = '#8756e4'
purpleyblue: str = '#5f34e7'
purpleygrey: str = '#947e94'
purpleypink: str = '#c83cb9'
purplish: str = '#94568c'
purplishblue: str = '#601ef9'
purplishbrown: str = '#6b4247'
purplishgrey: str = '#7a687f'
purplishpink: str = '#ce5dae'
purplishred: str = '#b0054b'
purply: str = '#983fb2'
purplyblue: str = '#661aee'
purplypink: str = '#f075e6'
putty: str = '#beae8a'
racinggreen: str = '#014600'
radioactivegreen: str = '#2cfa1f'
```

static random_hex() → *str*

[returns a random hex color i.e. #000000]

Returns

[random hex color i.e. #000000]

Return type

[*str*]

raspberry: *str* = '#b00149'

rawsienna: *str* = '#9a6200'

rawumber: *str* = '#a75e09'

reallylightblue: *str* = '#d4ffff'

red: *str* = '#e50000'

redbrown: *str* = '#8b2e16'

reddish: *str* = '#c44240'

reddishbrown: *str* = '#7f2b0a'

reddishgrey: *str* = '#997570'

reddishorange: *str* = '#f8481c'

reddishpink: *str* = '#fe2c54'

reddishpurple: *str* = '#910951'

reddybrown: *str* = '#6e1005'

redorange: *str* = '#fd3c06'

redpink: *str* = '#fa2a55'

redpurple: *str* = '#820747'

redviolet: *str* = '#9e0168'

redwine: *str* = '#8c0034'

static rgb2hex(r: *int*, g: *int*, b: *int*) → *str*

[takes 3 rgb values and returns a hex string i.e. #000000]

Parameters

- **r** ([*int*]) – [a value between 0 and 255]
- **g** ([*int*]) – [a value between 0 and 255]
- **b** ([*int*]) – [a value between 0 and 255]

Returns

[retuns a hex string i.e #ffffff]

Return type

[*str*]

```
richblue:  str = '#021bf9'
richpurple: str = '#720058'
robineggblue: str = '#8af1fe'
robinsegg: str = '#6dedfd'
robinseggblue: str = '#98eff9'
rosa: str = '#fe86a4'
rose: str = '#cf6275'
rosepink: str = '#f7879a'
rosered: str = '#be013c'
rosypink: str = '#f6688e'
rouge: str = '#ab1239'
royal: str = '#0c1793'
royalblue: str = '#0504aa'
royalpurple: str = '#4b006e'
ruby: str = '#ca0147'
russet: str = '#a13905'
rust: str = '#a83c09'
rustbrown: str = '#8b3103'
rustorange: str = '#c45508'
rustred: str = '#aa2704'
rustyorange: str = '#cd5909'
rustyred: str = '#af2f0d'
saffron: str = '#feb209'
sage: str = '#87ae73'
sagegreen: str = '#88b378'
salmon: str = '#ff796c'
salmonpink: str = '#fe7b7c'
sand: str = '#e2ca76'
sandbrown: str = '#cba560'
sandstone: str = '#c9ae74'
sandy: str = '#f1da7a'
```

```
sandybrown: str = '#c4a661'
sandyellow: str = '#fce166'
sandyyellow: str = '#fdee73'
sapgreen: str = '#5c8b15'
sapphire: str = '#2138ab'
scarlet: str = '#be0119'
sea: str = '#3c9992'
seablue: str = '#047495'
seafoam: str = '#80f9ad'
seafoamblue: str = '#78d1b6'
seafoamgreen: str = '#7af9ab'
seagreen: str = '#53fca1'
seaweed: str = '#18d17b'
seaweedgreen: str = '#35ad6b'
sepia: str = '#985e2b'
shamrock: str = '#01b44c'
shamrockgreen: str = '#02c14d'
shit: str = '#7f5f00'
shitbrown: str = '#7b5804'
shitgreen: str = '#758000'
shockingpink: str = '#fe02a2'
sickgreen: str = '#9db92c'
sicklygreen: str = '#94b21c'
sicklyyellow: str = '#d0e429'
sienna: str = '#a9561e'
silver: str = '#c5c9c7'
sky: str = '#82cafc'
skyblue: str = '#75bbfd'
slate: str = '#516572'
slateblue: str = '#5b7c99'
slategreen: str = '#658d6d'
```

```
slategrey: str = '#59656d'
slimegreen: str = '#99cc04'
snot: str = '#acbb0d'
snotgreen: str = '#9dc100'
softblue: str = '#6488ea'
softgreen: str = '#6fc276'
softpink: str = '#fdb0c0'
softpurple: str = '#a66fb5'
spearmint: str = '#1ef876'
springgreen: str = '#a9f971'
spruce: str = '#0a5f38'
squash: str = '#f2ab15'
steel: str = '#738595'
steelblue: str = '#5a7d9a'
steelgrey: str = '#6f828a'
stone: str = '#ada587'
stormyblue: str = '#507b9c'
straw: str = '#fcf679'
strawberry: str = '#fb2943'
strongblue: str = '#0c06f7'
strongpink: str = '#ff0789'
sunflower: str = '#ffc512'
sunfloweryellow: str = '#ffda03'
sunnyyellow: str = '#fff917'
sunshineyellow: str = '#fffd37'
sunnyellow: str = '#ffdf22'
swamp: str = '#698339'
swampgreen: str = '#748500'
tan: str = '#d1b26f'
tanbrown: str = '#ab7e4c'
tangerine: str = '#ff9408'
```

tangreen: `str = '#a9be70'`

taupe: `str = '#b9a281'`

tea: `str = '#65ab7c'`

teagreen: `str = '#bdf8a3'`

teal: `str = '#029386'`

tealblue: `str = '#01889f'`

tealgreen: `str = '#25a36f'`

tealish: `str = '#24bca8'`

tealishgreen: `str = '#0cdc73'`

terracota: `str = '#cb6843'`

terracotta: `str = '#c9643b'`

tiffanyblue: `str = '#7bf2da'`

`toCSS()` → `str`

return the color as a CSS string

`toHsl()`

returns the hsl for the color

`toHsv()`

get the hsv for the color

`toRGB()`

[returns the color as RGB]

Returns

[(r, g, b)]

Return type

[tuple]

`toSVG(shape='circle', size=10)`

returns the color as an svg string :param shape: [can be circle or square] :type shape: [str] :param size: [size in pixels] :type size: [int]

tomato: `str = '#ef4026'`

tomatoired: `str = '#ec2d01'`

topaz: `str = '#13bbaf'`

toupe: `str = '#c7ac7d'`

toxicgreen: `str = '#61de2a'`

treegreen: `str = '#2a7e19'`

trueblue: `str = '#010fcc'`

```
truegreen:  str = '#089404'
turquoise:  str = '#06c2ac'
turquoiseblue:  str = '#06b1c4'
turquoisegreen:  str = '#04f489'
turtlegreen:  str = '#75b84f'
twilight:  str = '#4e518b'
twilightblue:  str = '#0a437a'
uglyblue:  str = '#31668a'
uglybrown:  str = '#7d7103'
uglygreen:  str = '#7a9703'
uglypink:  str = '#cd7584'
uglypurple:  str = '#a442a0'
uglyyellow:  str = '#d0c101'
ultramarine:  str = '#2000b1'
ultramarineblue:  str = '#1805db'
umber:  str = '#b26400'
velvet:  str = '#750851'
vermillion:  str = '#f4320c'
verydarkblue:  str = '#000133'
verydarkbrown:  str = '#1d0200'
verydarkgreen:  str = '#062e03'
verydarkpurple:  str = '#2a0134'
verylightblue:  str = '#d5ffff'
verylightbrown:  str = '#d3b683'
verylightgreen:  str = '#d1ffbd'
verylightpink:  str = '#fff4f2'
verylightpurple:  str = '#f6cefc'
verypaleblue:  str = '#d6fffe'
verypalegreen:  str = '#cffdbc'
vibrantblue:  str = '#0339f8'
vibrantgreen:  str = '#0add08'
```

```
vibrantpurple: str = '#ad03de'
violet: str = '#9a0eea'
violetblue: str = '#510ac9'
violetpink: str = '#fb5ffc'
violetred: str = '#a50055'
viridian: str = '#1e9167'
vividblue: str = '#152eff'
vividgreen: str = '#2fef10'
vividpurple: str = '#9900fa'
vomit: str = '#a2a415'
vomitgreen: str = '#89a203'
vomityellow: str = '#c7c10c'
warmblue: str = '#4b57db'
warmbrown: str = '#964e02'
warmgrey: str = '#978a84'
warmpink: str = '#fb5581'
warmpurple: str = '#952e8f'
washedoutgreen: str = '#bcf5a6'
waterblue: str = '#0e87cc'
watermelon: str = '#fd4659'
weirdgreen: str = '#3ae57f'
wheat: str = '#fbdd7e'
white: str = '#ffffff'
windowsblue: str = '#3778bf'
wine: str = '#80013f'
winered: str = '#7b0323'
wintergreen: str = '#20f986'
wisteria: str = '#a87dc2'
yellow: str = '#ffff14'
yellowbrown: str = '#b79400'
yellowgreen: str = '#bbf90f'
```



```

yellowish:  str = '#faee66'
yellowishbrown: str = '#9b7a01'
yellowishgreen: str = '#b0dd16'
yellowishorange: str = '#ffab0f'
yellowishtan: str = '#fcfc81'
yellowochre: str = '#cb9d06'
yelloworange: str = '#fcb001'
yellowtan: str = '#ffe36e'
yellowybrown: str = '#ae8b0c'
yellowygreen: str = '#bfff12'

```

Char

```

from domonic.constants.entities import Char
print(Char.AMPERSAND)

```

```
class domonic.constants.entities.Char(character: str)
```

```

AACUTE: str = '&aacute;'
    á
ACIRC: str = '&acirc;'
    â
ACUTE: str = '&acute;'
    ´
AELIG: str = '&aelig;'
    æ
AGRAVE: str = '&agrave;'
    à
ALPHA: str = '&alpha;'
AMPERSAND: str = '&amp;'
    &
AND: str = '&and;'
ANG: str = '&ang;'
APOSTROPHE: str = '&#39;'
    ´
ARING: str = '&aring;'
    å

```

```
ASTERISK: str = '&#42;'  
    •  
ASYMP: str = '&asymp;'  
ATILDE: str = '&atilde;'  
    ã  
AT_SIGN: str = '&#64;'  
    @  
AUML: str = '&auml;'  
    ä  
BACKSLASH: str = '&#92;'  
BDQUO: str = '&bdquo;'  
    ”  
BETA: str = '&beta;'  
BRVBAR: str = '&brvbar;'  
    |  
BULL: str = '&bull;'  
    •  
CAP: str = '&cap;'  
CARET: str = '&#94;'  
    ^  
CCEDIL: str = '&ccedil;'  
    ç  
CEDIL: str = '&cedil;'  
    ¸  
CENT: str = '&cent;'  
    ¢  
CHECK: str = '&check;'  
    ✓  
CHECKMARK: str = '&checkmark;'  
    ✓  
CHI: str = '&chi;'  
CIRC: str = '&circ;'  
    ^  
CLOSING_CURLY_BRACE: str = '&#125;'  
    }  
CLOSING_PARENTHESIS: str = '&#41;'  
    )
```

```

CLOSING_SQUARE_BRACKET: str = '&#93;'
    ]
CLUBS: str = '&clubs;'
COLON: str = '&#58;'
    :
COMMA: str = '&#44;'
    ,
CONG: str = '&cong;'
COPY: str = '&copy;'
    ©
COPYRIGHT: str = '&copy;'
    ©
CRARR: str = '&crarr;'
CROSS: str = '&cross;'
CUP: str = '&cup;'
CURREN: str = '&curren;'
    ₤
DAGGER: str = '&Dagger;'
    †
DARR: str = '&darr;'
    ↓
DEG: str = '&deg;'
    °
DELTA: str = '&delta;'
DIAMONDS: str = '&diams;'
DIAMS: str = '&diams;'
DIVIDE: str = '&divide;'
    ÷
DOLLAR_SIGN: str = '&#36;'
    $
DOWN: str = '&darr;'
    ↓
EACUTE: str = '&eacute;'
    é
ECIRC: str = '&ecirc;'
    ê

```

Egrave: `str = '`'`
è
Eight: `str = '8'`
8
Empty: `str = '∅'`
Epsilon: `str = 'ε'`
Equiv: `str = '≡'`
Eta: `str = 'η'`
Eth: `str = 'ð'`
ð
Euml: `str = 'ë'`
ë
Euro: `str = '€'`
€
Exclamation_Mark: `str = '!'`
!
Exist: `str = '∃'`
Female: `str = '♀'`
Five: `str = '5'`
5
Flat: `str = '♭'`
Fnof: `str = 'ƒ'`
f
Forall: `str = '∀'`
Four: `str = '4'`
4
Frac12: `str = '½'`
½
Frac13: `str = '⅓'`
Frac14: `str = '¼'`
¼
Frac15: `str = '⅕'`
Frac16: `str = '⅙'`
Frac18: `str = '⅛'`
Frac23: `str = '⅔'`

```

FRAC25: str = '&frac25;'
FRAC34: str = '&frac34;'
      ¾
FRAC35: str = '&frac35;'
FRAC38: str = '&frac38;'
FRAC45: str = '&frac45;'
FRAC56: str = '&frac56;'
FRAC58: str = '&frac58;'
FRAC78: str = '&frac78;'

GAMMA: str = '&gamma;'

GE: str = '&ge;'

GRAVE_ACCENT: str = '&#96;'

GREATER_THAN: str = '&gt;'
      >

HARR: str = '&harr;'

HEARTS: str = '&hearts;'

HELLIP: str = '&hellip;'
      ...

HYPHEN: str = '&#45;'
      •

IACUTE: str = '&iacute;'
      í

ICIRC: str = '&icirc;'
      î

IEXCL: str = '&iexcl;'
      ¡

IGRAVE: str = '&igrave;'
      ì

INFIN: str = '&infin;'
      ∞

INT: str = '&int;'

IOTA: str = '&iota;'

IQUEST: str = '&iquest;'
      ¿

ISIN: str = '&isin;'

```

```
IUML: str = '&iuml;'  
i  
KAPPA: str = '&kappa;'  
LAMBDA: str = '&lambda;'  
LAQUO: str = '&laquo;'  
«  
LARR: str = '&larr;'  
←  
LCEIL: str = '&lceil;'  
LDQUO: str = '&ldquo;'  
“  
LE: str = '&le;'  
LEFT: str = '&larr;'  
←  
LEFT_CURLY_BRACE: str = '&#123;'  
{  
LEFT_PARENTHESIS: str = '&#40;'  
(  
LESS_THAN: str = '&lt;'  
<  
LFLOOR: str = '&lfloor;'  
LOWAST: str = '&lowast;'  
LOWERCASE_A: str = '&#97;'  
a  
LOWERCASE_B: str = '&#98;'  
b  
LOWERCASE_C: str = '&#99;'  
c  
LOWERCASE_D: str = '&#100;'  
d  
LOWERCASE_E: str = '&#101;'  
e  
LOWERCASE_F: str = '&#102;'  
f  
LOWERCASE_G: str = '&#103;'  
g  
LOWERCASE_H: str = '&#104;'  
h
```

```
LOWERCASE_I: str = 'i'
i
LOWERCASE_J: str = 'j'
j
LOWERCASE_K: str = 'k'
k
LOWERCASE_L: str = 'l'
l
LOWERCASE_M: str = 'm'
m
LOWERCASE_N: str = 'n'
n
LOWERCASE_O: str = 'o'
o
LOWERCASE_P: str = 'p'
p
LOWERCASE_Q: str = 'q'
q
LOWERCASE_R: str = 'r'
r
LOWERCASE_S: str = 's'
s
LOWERCASE_T: str = 't'
t
LOWERCASE_U: str = 'u'
u
LOWERCASE_V: str = 'v'
v
LOWERCASE_W: str = 'w'
w
LOWERCASE_X: str = 'x'
x
LOWERCASE_Y: str = 'y'
y
LOWERCASE_Z: str = 'z'
z
LOZ: str = '&loz;'
LSAQUO: str = '&lsaquo;'
<
```

```
LSQUO: str = '&lsquo;'
    ´

MACR: str = '&macr;'
    ¯

MALE: str = '&male;'

MDASH: str = '&mdash;'
    —

MHO: str = '&mho;'
    Ω

MICRO: str = '&micro;'
    μ

MINUS: str = '&minus;'

MU: str = '&mu;'

NABLA: str = '&nabla;'

NATUR: str = '&natur;'

NATURAL: str = '&natural;'

NBSP: str = '&nbsp;'

NDASH: str = '&ndash;'
    –

NE: str = '&ne;'

NI: str = '&ni;'

NINE: str = '&#57;'
    9

NOT: str = '&not;'
    ¬

NOTIN: str = '&notin;'

NSUB: str = '&nsub;'

NTILDE: str = '&ntilde;'
    ñ

NU: str = '&nu;'

NUMBER_SIGN: str = '&#35;'
    #

OACUTE: str = '&oacute;'
    ó

OCIRC: str = '&ocirc;'
    ô
```



```

OELIG: str = '&OElig;'
    Œ
OGRAVE: str = '&ograve;'
    ò
OHM: str = '&ohm;'
    Ω
OLINE: str = '&oline;'
OMEGA: str = '&omega;'
OMICRON: str = '&omicron;'
ONE: str = '&#49;'
    1
OPENING_CURLY_BRACE: str = '&#123;'
    {
OPENING_PARENTHESIS: str = '&#40;'
    (
OPENING_SQUARE_BRACKET: str = '&#91;'
    [
OPLUS: str = '&oplus;'
OR: str = '&or;'
ORDF: str = '&ordf;'
    ª
ORDM: str = '&ordm;'
    º
OSLASH: str = '&oslash;'
    ø
OTILDE: str = '&otilde;'
    õ
OTIMES: str = '&otimes;'
Ouml: str = '&ouml;'
    ö
PARA: str = '&para;'
    ¶
PART: str = '&part;'
PERCENT_SIGN: str = '&#37;'
    %
PERIOD: str = '&#46;'
    .

```

```
PERMIL: str = '&permil;'
    ‰
PERP: str = '&perp;'
PHI: str = '&phi;'
PHONE: str = '&phone;'
PI: str = '&pi;'
PIV: str = '&piv;'
PLUSMN: str = '&plusmn;'
    ±
PLUS_SIGN: str = '&#43;'
    •
POUND: str = '&pound;'
    £
PRIME: str = '&Prime;'
PROD: str = '&prod;'
PROP: str = '&prop;'
PSI: str = '&psi;'
QUESTION_MARK: str = '&#63;'
    ?
QUOTATION_MARK: str = '&#34;'
    “
RADIC: str = '&radic;'
RAQUO: str = '&raquo;'
    »
RARR: str = '&rarr;'
    →
RCEIL: str = '&rceil;'
RDQUO: str = '&rdquo;'
    ”
REG: str = '&reg;'
    ®
RFLOOR: str = '&rfloor;'
RHO: str = '&rho;'
RIGHT: str = '&rarr;'
    →
```

```

RIGHT_CURLY_BRACE: str = '&#125;'
    }
RIGHT_PARENTHESIS: str = '&#41;'
    )
RSAQUO: str = '&rsaquo;'
    >
RSQUO: str = '&rsquo;'
    ,
SBQUO: str = '&sbquo;'
    ,
SCARON: str = '&Scaron;'
    Š
SDOT: str = '&sdot;'
SECT: str = '&sect;'
    §
SEMICOLON: str = '&#59;'
    ;
SEVEN: str = '&#55;'
    7
SHARP: str = '&sharp;'
SIGMA: str = '&sigma;'
SIGMAF: str = '&sigmaf;'
SIM: str = '&sim;'
SIX: str = '&#54;'
    6
SLASH: str = '&#47;'
    /
SPADES: str = '&spades;'
STAR: str = '&star;'
STARF: str = '&starf;'
SUB: str = '&sub;'
SUBE: str = '&sube;'
SUM: str = '&sum;'
SUNG: str = '&sung;'
    ♪
SUP: str = '&sup;'

```

SUP1: `str` = '¹'
1

SUP2: `str` = '²'
2

SUP3: `str` = '³'
3

SUPE: `str` = '⊇'

SZLIG: `str` = 'ß'
ß

Scaron: `str` = 'Š'
Š

TAU: `str` = 'τ'

THERE4: `str` = '∴'

THETA: `str` = 'θ'

THETASYM: `str` = 'ϑ'

THORN: `str` = 'þ'
þ

THREE: `str` = '3'
3

TICK: `str` = '✓'
✓

TILDE: `str` = '˜'
~

TIMES: `str` = '×'
×

TRADE: `str` = '™'
™

TRADEMARK: `str` = '™'
™

TWO: `str` = '2'
2

UACUTE: `str` = 'ú'
ú

UARR: `str` = '↑'
↑

UCIRC: `str` = 'û'
û

```

UGRAVE: str = '&ugrave;'
    ù
UML: str = '&uml;'
    ..
UNDERSCORE: str = '&#95;'
    -
UP: str = '&uarr;'
    ↑
UPPERCASE_A: str = '&#65;'
    A
UPPERCASE_B: str = '&#66;'
    B
UPPERCASE_C: str = '&#67;'
    C
UPPERCASE_D: str = '&#68;'
    D
UPPERCASE_E: str = '&#69;'
    E
UPPERCASE_F: str = '&#70;'
    F
UPPERCASE_G: str = '&#71;'
    G
UPPERCASE_H: str = '&#72;'
    H
UPPERCASE_I: str = '&#73;'
    I
UPPERCASE_J: str = '&#74;'
    J
UPPERCASE_K: str = '&#75;'
    K
UPPERCASE_L: str = '&#76;'
    L
UPPERCASE_M: str = '&#77;'
    M
UPPERCASE_N: str = '&#78;'
    N
UPPERCASE_O: str = '&#79;'
    O
UPPERCASE_P: str = '&#80;'
    P

```

```
UPPERCASE_Q: str = '&#81;'
    Q
UPPERCASE_R: str = '&#82;'
    R
UPPERCASE_S: str = '&#83;'
    S
UPPERCASE_T: str = '&#84;'
    T
UPPERCASE_U: str = '&#85;'
    U
UPPERCASE_V: str = '&#86;'
    V
UPPERCASE_W: str = '&#87;'
    W
UPPERCASE_X: str = '&#88;'
    X
UPPERCASE_Y: str = '&#89;'
    Y
UPPERCASE_Z: str = '&#90;'
    Z
UPSIH: str = '&upsih;'
UPSILON: str = '&epsilon;'
Uuml: str = '&uuml;'
    ü
VERTICAL_BAR: str = '&#124;'

XI: str = '&xi;'
YACUTE: str = '&yacute;'
    ý
YEN: str = '&yen;'
    ¥
Yuml: str = '&Yuml;'
    Ÿ
ZERO: str = '&#48;'
    0
ZETA: str = '&zeta;'
```

```
oeLIG: str = '&oelig;'
      œ
scaron: str = '&scaron;'
      š
```

KeyCode

```
from domonic.constants.keyboard import KeyCode
print(KeyCode.DOWN)
print(KeyCode.ENTER)
```

```
class domonic.constants.keyboard.KeyCode
```

```
    A: str = '65'
    ALTERNATE: str = '18'
    B: str = '66'
    BACKQUOTE: str = '192'
    BACKSLASH: str = '220'
    BACKSPACE: str = '8'
    C: str = '67'
    CAPS_LOCK: str = '20'
    COMMA: str = '188'
    COMMAND: str = '15'
    CONTROL: str = '17'
    D: str = '68'
    DELETE: str = '46'
    DOWN: str = '40'
    E: str = '69'
    END: str = '35'
    ENTER: str = '13'
    EQUAL: str = '187'
    ESCAPE: str = '27'
    F: str = '70'
    F1: str = '112'
    F10: str = '121'
```

```
F11: str = '122'
F12: str = '123'
F13: str = '124'
F14: str = '125'
F15: str = '126'
F2: str = '113'
F3: str = '114'
F4: str = '115'
F5: str = '116'
F6: str = '117'
F7: str = '118'
F8: str = '119'
F9: str = '120'
G: str = '71'
H: str = '72'
HOME: str = '36'
I: str = '73'
INSERT: str = '45'
J: str = '74'
K: str = '75'
L: str = '76'
LEFT: str = '37'
LEFTBRACKET: str = '219'
M: str = '77'
MINUS: str = '189'
N: str = '78'
NUMBER_0: str = '48'
NUMBER_1: str = '49'
NUMBER_2: str = '50'
NUMBER_3: str = '51'
NUMBER_4: str = '52'
```



```
NUMBER_5: str = '53'
NUMBER_6: str = '54'
NUMBER_7: str = '55'
NUMBER_8: str = '56'
NUMBER_9: str = '57'
NUMPAD: str = '21'
NUMPAD_0: str = '96'
NUMPAD_1: str = '97'
NUMPAD_2: str = '98'
NUMPAD_3: str = '99'
NUMPAD_4: str = '100'
NUMPAD_5: str = '101'
NUMPAD_6: str = '102'
NUMPAD_7: str = '103'
NUMPAD_8: str = '104'
NUMPAD_9: str = '105'
NUMPAD_ADD: str = '107'
NUMPAD_DECIMAL: str = '110'
NUMPAD_DIVIDE: str = '111'
NUMPAD_ENTER: str = '108'
NUMPAD_MULTIPLY: str = '106'
NUMPAD_SUBTRACT: str = '109'
O: str = '79'
P: str = '80'
PAGE_DOWN: str = '34'
PAGE_UP: str = '33'
PERIOD: str = '190'
Q: str = '81'
QUOTE: str = '222'
R: str = '82'
RETURN: str = '13'
```

```
RIGHT: str = '39'

RIGHTBRACKET: str = '221'

S: str = '83'

SEMICOLON: str = '186'

SHIFT: str = '16'
    ?? left or right or both?

SLASH: str = '191'

SPACE: str = '32'

T: str = '84'

TAB: str = '9'

U: str = '85'

UP: str = '38'

V: str = '86'

W: str = '87'

X: str = '88'

Y: str = '89'

Z: str = '9'
```

1.4.10 terminal

There is a command line package that can call bash/unix/posix and other apps on the command line.

This package only works on nix systems as it effectively just passes stuff off to subprocess.

```
from domonic.terminal import *

print(ls())
print(ls("-al"))
print(ls("../"))

print(pwd())

print(mkdir('somedir'))
print(touch('somefile'))
print(git('status'))

for file in ls( "-al" ):
    print("Line : ", file)

for f in ls():
    try:
        print(f)
```

(continues on next page)

(continued from previous page)

```

        print(cat(f))
    except Exception as e:
        pass

for i, l in enumerate(cat('LICENSE.txt')):
    print(i,l)

print(man("ls"))
print(echo('test'))
print(df())
print(du())

for thing in du():
    print(thing)

print(find('.'))
# print(ping('eventual.technology'))# < TODO - need to stream output
print(cowsay('moo'))
print(wget('eventual.technology'))
print(date())
print(cal())

```

run arbitrary commands...

```

from domonic.terminal import command
command.run("echo hi")

```

Take a look at the code in ‘terminal.py’ to see all the commands as there’s loads. (Disclaimer: not all tested.)

domonic.terminal

- call command line functions in python 3

exception domonic.terminal.TerminalException(*error, message='An error message was recieved from terminal'*)

raised if the terminal throws an exception

```

class domonic.terminal.alias(*args, **kwargs)

class domonic.terminal.apt(*args, **kwargs)

class domonic.terminal.ar(*args, **kwargs)

class domonic.terminal.asa(*args, **kwargs)

class domonic.terminal.at(*args, **kwargs)

class domonic.terminal.awk(*args, **kwargs)

class domonic.terminal.banner(*args, **kwargs)

class domonic.terminal.basename(*args, **kwargs)

```

```
class domonic.terminal.bash(*args, **kwargs)
```

```
class domonic.terminal.batch(*args, **kwargs)
```

```
class domonic.terminal.bc(*args, **kwargs)
```

```
class domonic.terminal.bg(*args, **kwargs)
```

```
class domonic.terminal.bind(*args, **kwargs)
```

```
class domonic.terminal.builtin(*args, **kwargs)
```

```
class domonic.terminal.cal(*args, **kwargs)
```

```
class domonic.terminal.caller(*args, **kwargs)
```

```
class domonic.terminal.cat(*args, **kwargs)
```

```
class domonic.terminal.cc(*args, **kwargs)
```

```
class domonic.terminal.cd(*args, **kwargs)
```

NOTE - 'cd' does not run on terminal - cd is pointless as session opens and closes - so is overridden to change dirs via pure python

```
class domonic.terminal.cflow(*args, **kwargs)
```

```
class domonic.terminal.chgrp(*args, **kwargs)
```

```
class domonic.terminal.chmod(*args, **kwargs)
```

```
class domonic.terminal.chown(*args, **kwargs)
```

```
class domonic.terminal.cksum(*args, **kwargs)
```

```
class domonic.terminal.comm(*args, **kwargs)
```

```
class domonic.terminal.command(*args, **kwargs)
```

wrapper class for all terminal commands

```
static run(cmd: str)
```

runs any command on the terminal

Parameters

cmd (*str*) – The command you want to run on the terminal

Returns

the response as a string

Return type

str

```
class domonic.terminal.compgen(*args, **kwargs)
```

```
class domonic.terminal.complete(*args, **kwargs)
```

```
class domonic.terminal.compopt(*args, **kwargs)
```

```
class domonic.terminal.compress(*args, **kwargs)
```

```
class domonic.terminal.convert(*args, **kwargs)
```

```
class domonic.terminal.cowsay(*args, **kwargs)
class domonic.terminal.cp(*args, **kwargs)
class domonic.terminal.cron(*args, **kwargs)
class domonic.terminal.crontab(*args, **kwargs)
class domonic.terminal.csplit(*args, **kwargs)
class domonic.terminal.ctags(*args, **kwargs)
class domonic.terminal.curl(*args, **kwargs)
class domonic.terminal.cut(*args, **kwargs)
class domonic.terminal.cxref(*args, **kwargs)
class domonic.terminal.date(*args, **kwargs)
class domonic.terminal.dd(*args, **kwargs)
class domonic.terminal.declare(*args, **kwargs)
class domonic.terminal.delta(*args, **kwargs)
class domonic.terminal.df(*args, **kwargs)
class domonic.terminal.diff(*args, **kwargs)
class domonic.terminal.dirname(*args, **kwargs)
class domonic.terminal.dirs(*args, **kwargs)
class domonic.terminal.disown(*args, **kwargs)
class domonic.terminal.du(*args, **kwargs)
class domonic.terminal.echo(*args, **kwargs)
class domonic.terminal.ed(*args, **kwargs)
class domonic.terminal.enable(*args, **kwargs)
class domonic.terminal.env(*args, **kwargs)
class domonic.terminal.ex(*args, **kwargs)
class domonic.terminal.exit(*args, **kwargs)
class domonic.terminal.expand(*args, **kwargs)
class domonic.terminal.export(*args, **kwargs)
class domonic.terminal.expr(*args, **kwargs)
class domonic.terminal.fc(*args, **kwargs)
class domonic.terminal.ffmpeg(*args, **kwargs)
class domonic.terminal.fg(*args, **kwargs)
```

```
class domonic.terminal.figlet(*args, **kwargs)
class domonic.terminal.file(*args, **kwargs)
class domonic.terminal.find(*args, **kwargs)
class domonic.terminal.finger(*args, **kwargs)
class domonic.terminal.fold(*args, **kwargs)
class domonic.terminal.fort77(*args, **kwargs)
class domonic.terminal.fuser(*args, **kwargs)
class domonic.terminal.gcc(*args, **kwargs)
class domonic.terminal.gencat(*args, **kwargs)
class domonic.terminal.get(*args, **kwargs)
class domonic.terminal.getconf(*args, **kwargs)
class domonic.terminal.getopts(*args, **kwargs)
class domonic.terminal.git(*args, **kwargs)
class domonic.terminal.grep(*args, **kwargs)
class domonic.terminal.groupadd(*args, **kwargs)
class domonic.terminal.groupdel(*args, **kwargs)
class domonic.terminal.groups(*args, **kwargs)
class domonic.terminal.gunzip(*args, **kwargs)
class domonic.terminal.gzip(*args, **kwargs)
class domonic.terminal.head(*args, **kwargs)
class domonic.terminal.history(*args, **kwargs)
class domonic.terminal.iconv(*args, **kwargs)
class domonic.terminal.ifconfig(*args, **kwargs)
class domonic.terminal.ipconfig(*args, **kwargs)
class domonic.terminal.ipcrm(*args, **kwargs)
class domonic.terminal.ipcs(*args, **kwargs)
class domonic.terminal.jobs(*args, **kwargs)
class domonic.terminal.join(*args, **kwargs)
class domonic.terminal.jq(*args, **kwargs)
class domonic.terminal.kill(*args, **kwargs)
class domonic.terminal.killall(*args, **kwargs)
```

```
class domonic.terminal.less(*args, **kwargs)
class domonic.terminal.let(*args, **kwargs)
class domonic.terminal.lex(*args, **kwargs)
class domonic.terminal.link(*args, **kwargs)
class domonic.terminal.ln(*args, **kwargs)
class domonic.terminal.local(*args, **kwargs)
class domonic.terminal.locale(*args, **kwargs)
class domonic.terminal.localedef(*args, **kwargs)
class domonic.terminal.logger(*args, **kwargs)
class domonic.terminal.logname(*args, **kwargs)
class domonic.terminal.logout(*args, **kwargs)
class domonic.terminal.lp(*args, **kwargs)
class domonic.terminal.ls(*args, **kwargs)
class domonic.terminal.m4(*args, **kwargs)
class domonic.terminal.mailx(*args, **kwargs)
class domonic.terminal.make(*args, **kwargs)
class domonic.terminal.man(*args, **kwargs)
class domonic.terminal.mapfile(*args, **kwargs)
class domonic.terminal.mesg(*args, **kwargs)
class domonic.terminal.mkdir(*args, **kwargs)
class domonic.terminal.mkfifo(*args, **kwargs)
class domonic.terminal.mkfile(*args, **kwargs)
class domonic.terminal.more(*args, **kwargs)
class domonic.terminal.mv(*args, **kwargs)
class domonic.terminal.nautilus(*args, **kwargs)
class domonic.terminal.newgrp(*args, **kwargs)
class domonic.terminal.nice(*args, **kwargs)
class domonic.terminal.nl(*args, **kwargs)
class domonic.terminal.nm(*args, **kwargs)
class domonic.terminal.nmap(*args, **kwargs)
class domonic.terminal.nohup(*args, **kwargs)
```

```
class domonic.terminal.npm(*args, **kwargs)
class domonic.terminal.od(*args, **kwargs)
class domonic.terminal.passwd(*args, **kwargs)
class domonic.terminal.paste(*args, **kwargs)
class domonic.terminal.patch(*args, **kwargs)
class domonic.terminal.pathchk(*args, **kwargs)
class domonic.terminal.pax(*args, **kwargs)
class domonic.terminal.ping(*args, **kwargs)
class domonic.terminal.pip(*args, **kwargs)
class domonic.terminal.popd(*args, **kwargs)
class domonic.terminal.pr(*args, **kwargs)
class domonic.terminal.printf(*args, **kwargs)
class domonic.terminal.prs(*args, **kwargs)
class domonic.terminal.ps(*args, **kwargs)
class domonic.terminal.pushd(*args, **kwargs)
class domonic.terminal.pwd(*args, **kwargs)
class domonic.terminal.python(*args, **kwargs)
class domonic.terminal.qalter(*args, **kwargs)
class domonic.terminal.qdel(*args, **kwargs)
class domonic.terminal.qhold(*args, **kwargs)
class domonic.terminal.qmove(*args, **kwargs)
class domonic.terminal.qmsg(*args, **kwargs)
class domonic.terminal.qrerun(*args, **kwargs)
class domonic.terminal.qrls(*args, **kwargs)
class domonic.terminal.qselect(*args, **kwargs)
class domonic.terminal.qsig(*args, **kwargs)
class domonic.terminal.qstat(*args, **kwargs)
class domonic.terminal.qsub(*args, **kwargs)
class domonic.terminal.read(*args, **kwargs)
class domonic.terminal.readarray(*args, **kwargs)
class domonic.terminal.readonly(*args, **kwargs)
```



```
class domonic.terminal.reboot(*args, **kwargs)
class domonic.terminal.renice(*args, **kwargs)
class domonic.terminal.rm(*args, **kwargs)
class domonic.terminal.rmdel(*args, **kwargs)
class domonic.terminal.rmdir(*args, **kwargs)
class domonic.terminal.rsync(*args, **kwargs)
class domonic.terminal.sact(*args, **kwargs)
class domonic.terminal.say(*args, **kwargs)
class domonic.terminal.sccs(*args, **kwargs)
class domonic.terminal.scp(*args, **kwargs)
class domonic.terminal.sed(*args, **kwargs)
class domonic.terminal.sh(*args, **kwargs)
class domonic.terminal.shift(*args, **kwargs)
class domonic.terminal.shopt(*args, **kwargs)
class domonic.terminal.shutdown(*args, **kwargs)
class domonic.terminal.sleep(*args, **kwargs)
class domonic.terminal.sort(*args, **kwargs)
class domonic.terminal.source(*args, **kwargs)
class domonic.terminal.split(*args, **kwargs)
class domonic.terminal.ssh(*args, **kwargs)
class domonic.terminal.strings(*args, **kwargs)
class domonic.terminal.strip(*args, **kwargs)
class domonic.terminal.stty(*args, **kwargs)
class domonic.terminal.suspend(*args, **kwargs)
class domonic.terminal.tabs(*args, **kwargs)
class domonic.terminal.tail(*args, **kwargs)
class domonic.terminal.talk(*args, **kwargs)
class domonic.terminal.tar(*args, **kwargs)
class domonic.terminal.tee(*args, **kwargs)
class domonic.terminal.test(*args, **kwargs)
class domonic.terminal.time(*args, **kwargs)
```

```
class domonic.terminal.times(*args, **kwargs)
class domonic.terminal.touch(*args, **kwargs)
class domonic.terminal.tput(*args, **kwargs)
class domonic.terminal.tr(*args, **kwargs)
class domonic.terminal.trap(*args, **kwargs)
class domonic.terminal.true(*args, **kwargs)
class domonic.terminal.tsort(*args, **kwargs)
class domonic.terminal.tty(*args, **kwargs)
class domonic.terminal.typeset(*args, **kwargs)
class domonic.terminal.ulimit(*args, **kwargs)
class domonic.terminal.umask(*args, **kwargs)
class domonic.terminal.unalias(*args, **kwargs)
class domonic.terminal.uname(*args, **kwargs)
class domonic.terminal.uncompress(*args, **kwargs)
class domonic.terminal.unexpand(*args, **kwargs)
class domonic.terminal.unget(*args, **kwargs)
class domonic.terminal.uniq(*args, **kwargs)
class domonic.terminal.unlink(*args, **kwargs)
class domonic.terminal.unset(*args, **kwargs)
class domonic.terminal.uptime(*args, **kwargs)
class domonic.terminal.useradd(*args, **kwargs)
class domonic.terminal.userdel(*args, **kwargs)
class domonic.terminal.users(*args, **kwargs)
class domonic.terminal.uucp(*args, **kwargs)
class domonic.terminal.uudecode(*args, **kwargs)
class domonic.terminal.uuencode(*args, **kwargs)
class domonic.terminal.uustat(*args, **kwargs)
class domonic.terminal.uux(*args, **kwargs)
class domonic.terminal.val(*args, **kwargs)
class domonic.terminal.wait(*args, **kwargs)
class domonic.terminal.wc(*args, **kwargs)
```

```

class domonic.terminal.wget(*args, **kwargs)
class domonic.terminal.what(*args, **kwargs)
class domonic.terminal.who(*args, **kwargs)
class domonic.terminal.whoami(*args, **kwargs)
class domonic.terminal.write(*args, **kwargs)
class domonic.terminal.xargs(*args, **kwargs)
class domonic.terminal.yacc(*args, **kwargs)
class domonic.terminal.zcat(*args, **kwargs)

```

1.4.11 cmd

There is a cmd package for calling windows commands within python3

This package only works on windows systems as it effectively just passes stuff off to subprocess.

```

from domonic.cmd import *

print(dir())

print(mkdir('somedir'))
print(touch('somefile'))

for file in dir():
    print("Line : ", file)

```

run arbitrary commands...

```

from domonic.cmd import Cmdcommand
Cmdcommand.run("echo hi")

```

Take a look at the code in 'cmd.py' to see all the commands. (Disclaimer: not tested.)

domonic.cmd

- call cmd commands from python 3

exception domonic.cmd.CmdException(error, message: *str* = 'An error message was recieved from cmd')
 raised if cmd throws an exception

class domonic.cmd.Cmdcommand(*args, **kwargs)
 wrapper class for all cmd commands

static run(cmd: *str*) → *str*
 [runs any command on the cmd]

Parameters

cmd (*str*) – The command you want to run on cmd

Returns

the response as a string

Return type

str

class domonic.cmd.**attrib**(*args, **kwargs)

display file attributes

class domonic.cmd.**cd**(*args, **kwargs)

NOTE - 'cd' does not run on cmd - cd is pointless as session opens and closes - so is overridden to change dirs via pure python

class domonic.cmd.**chdir**(*args, **kwargs)**class** domonic.cmd.**chkdsk**(*args, **kwargs)

check volumes

class domonic.cmd.**comp**(*args, **kwargs)

compare file contents

class domonic.cmd.**copy**(*args, **kwargs)

copy files

class domonic.cmd.**dir**(*args, **kwargs)

list directory content

class domonic.cmd.**driverquery**(*args, **kwargs)

display installed devices and their properties

class domonic.cmd.**echo**(*args, **kwargs)

text output

class domonic.cmd.**erase**(*args, **kwargs)**class** domonic.cmd.**fc**(*args, **kwargs)**class** domonic.cmd.**fsutil**(*args, **kwargs)**class** domonic.cmd.**getmac**(*args, **kwargs)

display MAC address

class domonic.cmd.**gpresult**(*args, **kwargs)

display group policies

class domonic.cmd.**hostname**(*args, **kwargs)

display host name

class domonic.cmd.**ipconfig**(*args, **kwargs)

display IP network settings

class domonic.cmd.**logoff**(*args, **kwargs)

Logs the user out of Windows.

class domonic.cmd.**mkdir**(*args, **kwargs)

create a new directory

class domonic.cmd.**move**(*args, **kwargs)

move/rename files

```
class domonic.cmd.mrinfo(*args, **kwargs)
```

Provides information on the router

```
class domonic.cmd.ping(*args, **kwargs)
```

pings the network

```
class domonic.cmd.rename(*args, **kwargs)
```

rename files

```
class domonic.cmd.replace(*args, **kwargs)
```

replace files

```
class domonic.cmd.rmdir(*args, **kwargs)
```

delete directory

```
class domonic.cmd.shutdown(*args, **kwargs)
```

shutdown the computer. (/s), triggers a restart (/r), or logs the user out (/l).

```
class domonic.cmd.systeminfo(*args, **kwargs)
```

displays computer-specific properties and configurations

```
class domonic.cmd.tasklist(*args, **kwargs)
```

Lists all running processes

```
class domonic.cmd.title(*args, **kwargs)
```

Changes the title of the command prompt

```
class domonic.cmd.touch(*args, **kwargs)
```

```
class domonic.cmd.type_(*args, **kwargs)
```

display content of text files

```
class domonic.cmd.tzutil(*args, **kwargs)
```

bitsadmin Creates and monitors downloads and uploads chcp Changes the current code page choice Creates a selection list clip Forwards the result of a command to the clipboard. color Changes the background command Starts CMD.COM. 32-bit/DOS date Displays the current date and allows you to change it. debug Starts debug doskey Creates macros, recalls commands, and edits command input dosshell Opens the DOS shell edit Starts the MS-DOS editor edlin Creates and edits text files within the command prompt fasthelp Displays helpful information about commands fastopen Writes the position of a program into a specified list find Searches through a file or multiple files for a particular character sequence. findstr Finds character sequences in one or multiple files. forcedos Starts a program in the MS-DOS partial system graftabl Enables the option to use extended characters of a specific code page in graphics mode. 32-bit/DOS graphics Starts a program that can print graphics. 32-bit/DOS help Displays help text for a specific command (you can also use the /? command) kb16 Changes the country settings of the keyboard for DOS programs keyb Changes the country settings of the keyboard for DOS programs lpq Displays the status of a printer queue for computers that use a "line Printer Daemon" (LPD) lpr Sends a file to a computer that uses a line printer daemon (LPD). md Creates a new directory on the specified path. more Outputs the content of a file (for example, a text file) by the page. msg Sends a message to another user. nlsfunc Provides country-specific information for language support. 32-bit/DOS ntbackup Runs backup services directly from the command line or as part of batch or script files. XP path Creates and displays the path for searching executable files pause Pauses execution in batch files and scripts. popd Changes to the folder saved by the pushd command. print Prints a text file. prompt Changes the display of the command prompt pushd Saves a specific path into a script or batch file. qbasic Starts qbasic, a program environment based on the BASIC programming language rd Deletes a directory. rem Writes comments in batch and script files that aren't taken into account when executing runas Allows a user to run commands with the rights of another user. scandisk Starts Microsoft ScanDisk. The program searches data carriers for errors schtasks Sets the execution of specified programs and commands for a specified point in time. set Displays environmental variables of CMD.EXE and lets you configure them shift Moves variables within batch files and scripts sort Lists out data (from a file

or command) and outputs it again sorted start Opens a new command prompt window in which you can run a specific program or command subst Assigns a drive letter to a path to create a virtual drive taskkill Ends one or more running tasks. time Displays the current time and allows it to be changed. timeout Stops a process for a specified time. Files CMD command Description Windows version append Sets the path in which files will be searched for. 32-bit/DOS assoc Changes the program that's linked with a particular file ending. attrib Changes attributes of specified files. With the parameter +R you can protect a file from changes cipher Displays and changes the encryption status of files and directories on NTFS partitions. comp Compares the content of two files or two file sets. compact Displays and changes the compression status of files and directories on NTFS partitions. copy Copies a file or multiple files to another location. cscript Runs scripts over the Microsoft Script Host. del Deletes a file or multiple files. deltree Deletes a directory as well as all subdirectories and files within diantz Compresses files without any loss (command has the same function as makecab) diskcomp Compares the content of two disks (not 10)/DOS diskcopy Copies the content of a disk to another (not 10)/DOS endlocal Ends the valid range of changes to batch files or scripts. erase Function is the same as del exe2bin Converts an EXE file to a BIN file. 32-bit expand Extracts files and folders stored in CAB files (not 64-bit XP)/DOS extrac32 Extracts files and folders stored in CAB files. fc Compares two individual files or two sets of files with one another and displays the differences forfiles Selects one or more files and runs a command that refers to these files. ftype Specifies a program for opening a specific file type. goto Skips the execution within a batch program to a specific line (marker) if Represents a conditional statement and executes expressions within batch files only under certain conditions. makecab Compresses files without loss in CAB format (you can also use the diantz command). mklink Creates a symbolic link to a file. With /D you can also create connections to directories. move Moves a file or multiple files from one directory to another. openfiles Displays and separates open system files and folders. recover Restores readable files that were on a defective data drive. ren Changes the name of a particular file. rename Function is the same as ren replace Replaces the selected file or files with one or more other files. robocopy Allows so-called robust file copying. rsm Manages media on removable storage devices. setlocal Limits the valid range of changes to batch files or scripts. share Installs file sharing and file locking. 32-bit/DOS sxstrace Starts the WinSxs Tracing Utility, a tool for programming diagnostics. takeown Restores administrator access rights to a file that have been lost when reassigning a user. undelete Undoes the deletion of a file verify When enabled, checks whether files are written correctly on a data drive. where Finds files that match a particular search topic. xcopy Copies files and entire directory structures. System CMD command Description Windows version at Starts commands and programs at a particular time. auditpol Displays current monitoring policies. backup Creates backups of files. These can be recovered with restore (replaced by msbackup) bcdboot Creates and repairs start files bcdedit Allows users to make changes to start configuration data storage bdehdcfg Prepares a hard drive for BitLocker Drive Encryption. bootcfg Creates, edits, or displays the content of boot.ini bootsect Modifies the master boot code so that it's compatible with the Windows Boot Manager or NT Loader cacls Edits and displays the access control list. chkdsk Checks and repairs (with the parameter /R) a data drive chkntfs Changes or displays the data driver check at startup. cmdkey Can display (/list), create (/add), or delete (/delete) login information. convert Converts partitions from FAT/FAT32 to NTFS. ctty Changes the standard input and output for the system dblspace Creates or configures compresses drives (a newer version of the command is called drvspace) defrag Defragments all or only specified drives. Use /U to observe the progress. diskpart Manages, creates, and deletes partitions from the hard drive. diskperf Allows users to remotely control the disk performance counter. diskraid Manages RAID systems. dism Manages and integrates Windows images dispdiag Creates a file in the current directory in which you'll find information about your display. dosx Starts the DOS Protected Mode Interface driverquery Creates a list with all installed drivers. drvspace Creates or configures compressed drives. emm386 Provides DOS with more than 640 KB of RAM esentutl Manages databases within the extensible storage engine. eventcreate Creates an entry (ID and message) in an event log. eventtriggers Configures and displays event trigger. XP fdisk Creates, deletes, and manages partitions on the hard drive. fltmc Allows users to manage and display filter drivers. fondue Installs additional Windows features. format Formats a drive to the file system specified by the user hwrcomp Compiles self-created dictionaries for handwriting recognition hwrreg Installs a compiled dictionary for handwriting recognition icacls Edits and displays the access control list. ktmutil Starts the kernel transaction manager. label Changes or deletes a drive's label lh Loads a program into the high memory area (UMB) – has the same function as loadhigh licensingdiag Creates an XML and a CAB file that contain information on the Windows product licence loadfix Ensures that a program is loaded and executed above the first 64 KB of RAM. 32-bit/DOS loadhigh Has the same function as lh lock Locks a drive so that only a user-selected program can access it directly. 98/95 lodctr Updates all registry entries that

have to do with performance indicators logman Creates and manages event trace sessions and performance logs. manage-bde Configures drive encryption with BitLocker. mem Displays information about the RAM and indicates which programs are currently loaded in it. 32-bit/DOS memmaker Optimises the RAM mode Configures system devices – primarily on the COM or LPT port mofcomp Analyses files in managed object format (MOF) and adds the classes and instances to the WMI repository mountvol Creates and deletes mount points for drives and displays them. msav Starts Microsoft Antivirus msbackup Starts Microsoft Backup (replaces backup and restores) mscdex Loads the CD-ROM support for MS-DOS msd Starts the program Microsoft Diagnostics, with which system information can be displayed msixexec Starts the Windows installer muiunattend Starts an automatic setup process for the multilingual user interface (MUI) netcfg Installs the minimal operating system Microsoft Windows PE. ocsetup Installs additional Windows functions. 8/7/Vista pentnt Recognises floating point division errors pkgmgr Installs, uninstalls, and configures packages and functions for Windows. pnpunattend Automates the installation of device drivers pnputil Installs plug-and-play devices from the command prompt. power Uses the IDLE status of a processor to reduce energy consumption powercfg Allows the user to change the computer's energy options and control energy conservation plans. pwlauncher Configures the startup options for Windows To Go with which you can boot Windows from a USB drive qprocess Provides information on running processes. query Displays the status of a particular service. quser Provides information on the currently logged-in users. reagentc Configures the Windows recovery environment recimg Creates a user-defined Windows image to restore the system. 8 reg Manages the registry of the command prompt. regini Changes registry authorisations. register-cimprovider Registers a common information model provider (CIM provider) in Windows regsvr32 Registers a DLL file in the registry. relog Creates new performance indicator protocols from the data in the existing protocols. repair-bde Repairs and decrypts defective drives that are encrypted with BitLocker. reset Resets a session. You can also use the rwinsta command. restore Restores backups that were created with the backup command (replaced by msbackup) rwinsta Command has the same function as reset. sc Manages services by connecting to the Service Controller. scanreg Repairs the registry and allows a backup to be created of it. 98/95 sdbinst Applies user-defined database files (SDB). secdit Analyses the security settings by comparing the current configurations with templates. setver Sets a version number of MS-DOS that's forwarded to a program setx Creates or changes environmental variable in the user of system environment. sfc Checks all important and protected system files. smartdrv Starts and manages the hard drive cache program SMARTDrive sys Copies system files from MS-DOS and the command interpreter to another hard drive. This makes it bootable systeminfo Displays information about the Windows installation, including all installed service packages. tpmvscmgr Creates and deletes TPM virtual smart cards. tracerpt Processes logs or real-time data generated during the tracing of computer programs. typeperf Displays performance counter data or writes it into a file. unformat Undoes the drive formatting done by the format command unlock Unlocks a drive that was locked with the lock command. 98/95 unlodctr Deletes names as well as descriptions for extensible performance counters in the Windows registry. vaultcmd Creates, deletes, and displays saved registration information vol Displays the label and serial number of a drive vsafe Starts the antivirus software VSafe vssadmin Manages the volume shadow copy services that can be used to store different versions (snapshots) of drives. wadmin Creates backups of the operating system and delivers information to the created backup copies. wevtutil Manages event logs and event log files. winmgmt Manages WMI repositories. Backups (/backup) are possible with the command, for example winsat Evaluates various system factors – for example, processor performance or graphical capabilities. wmic Starts the Windows Management Instrumentation in the command prompt. xwizard Registers Windows data in the form of XML files Network CMD command Description Windows version arp Displays and edits entries in the Address Resolution Protocol cache atmadm Displays information on asynchronous transfer mode (ATM). XP certreq Manages and creates certificate registration requirements for certification authorities. certutil Manages services related to certificate authentication. change Changes the settings of a terminal server and can be used together with the parameters logon, port, or user checknetisolation Checks the network capability of apps from the Windows Store chglogon Enables, disables, or adjusts logins for terminal server sessions. chgport Displays or changes the COM pin assignment of terminal servers for DOS compatibility. chgusr Changes the installation mode of a terminal server. cmstp Installs or uninstalls profiles for the connection manager. djoin Creates a new computer account in the Active Directory Domain Services (AD DS). finger Provides information about users on remote devices using the Finger service. ftp Transfers data to an FTP server or from this to a PC. gpresult Displays information on the Group Policy. gpupdate Updates information on the Group Policy. interlnk Connects two computers via serial or parallel connection to share files or printers intersvr Starts an interlnk server and transfers data from one computer to another via serial or parallel connection ipxroute Changes and displays

information on the IPX routing tables. XP irftp Transfers files via infrared connection, if one is available. iscsi-
cli Manages iSCSI, which enables connections via the SCSI protocol. klist Displays all tickets authenticated by
the Kerberos service. Also enables the command to delete tickets (purge) ksetup Configures a connection to a
Kerberos server mount Enables network sharing under the Network File System. (To use the command, enable
NFS services) nbtstat Displays statistics and information on the TCP/IP connections on remote computers net
Configures and displays network settings net1 Configures and displays network settings netsh Starts the network
shell, which allows for network settings to be changed on local and remote computers. netstat Displays statistics
and information on TCP/IP connections on the local computer nfsadmin Manages NFS servers and clients (to be
able to use the command, you first have to enable NFS services in Windows) nltest Displays information related
to secure channels in the Active Directory Domain Services (AD DS) and tests the connections nslookup Sends
a DNS query to a specific IP or host name on the preconfigured DNS server. You can also specify another DNS
server ntsd Runs debugging. XP pathping Provides information on forwarding and package loss when sending
over a network and also specifies the latency. qappsrv Displays all available remote computers in the network.
qwinsta Displays information on the open remote desktop sessions. rasautou Manages autodial addresses. rasdial
Starts and ends network connections for Microsoft clients. rcp Copies files from a Windows computer to a server
that's running a RSDH daemon, and vice versa rdp sign Signs a remote desktop protocol file (RDP file)/7 rexec
Runs commands on a remote computer that's running a Rexec daemon. Vista/XP route Displays routing tables
and makes it possible to change, add, or delete entries rpcinfo Sends a remote procedure call (RPC) to an RPC
server. rpcping Sends a ping via remote procedure call (RPC) and checks whether a connection is possible. rsh
Runs commands on remote computers that are running the Unix program Remote Shell (RSH) setspn Creates,
deletes, and changes SPNs. shadow Monitors a session on a remote computer. showmount Provides information
on NFS file systems (to use the command, you first have to activate NFS services in Windows) tcmsetup Enables
or disables a client for the Telephony Application Programming Interface (TAPI) telnet Enables communication
with another computer that also uses the telnet protocol tftp Enables a file exchange between the local computer
and a server that supports the Trivial File Transfer Protocol (TFTP). tlntadmn Manages a telnet server on a local
or remote computer tracert Tracks a data package on the way through the network to a server. tscon Connects the
current local user session with a session on a remote computer. tsdiscon Ends the connection between a local user
session and a session on a remote computer.XP tskill Ends a process on a remote computer. tsshutdn Shuts down
or restarts a remote terminal server. umount Removes mounted network file system drives. w32tm Manages the
Windows time service that synchronises dates and times on all computers that share an AD DS domain. waitfor
Sends or waits on a single. wecutil Creates and managements subscriptions for events. winrm Manages secure
connections between local and remote computers via the WS management protocol. winrs Enables access to the
command line of a remote computer via a secure connection to implement changes. wsmanhttpconfig Manages
functions of the Windows Remote Management (winrm).

```
class domonic.cmd.ver(*args, **kwargs)
    display operating system version - TODO seemed to hang

class domonic.cmd.vol(*args, **kwargs)
    show volume description and serial numbers of the HDDs

class domonic.cmd.whoami(*args, **kwargs)
    information about the current user. /GROUP parameter
```

1.4.12 tween

For tweening there's now a port of the penner equations and function to use them.

```
import time

from domonic.lerpy.easing import *
from domonic.lerpy.tween import *
```

(continues on next page)

(continued from previous page)

```

someObj = {'x':0,'y':0,'z':0}
tw = Tween( someObj, { 'x':10, 'y':5, 'z':3 }, 6, Linear.easeIn )
tw.start()

time.sleep(1)
print(someObj)
time.sleep(1)
print(someObj)
time.sleep(1)
print(someObj)
# tw.pause()
time.sleep(1)
print(someObj)
time.sleep(1)
# tw.unpause()
print(someObj)
time.sleep(1)
print(someObj)
time.sleep(1)
print(someObj)

```

equations can be an array **with** different tweens per **property**

```
# tw.equations = [ Expo.easeOut, Expo.easeIn, { ease:Back.easeOut, a:0.5, b:1.5 } ]
```

Note* Tweens use a domonic 'setInterval' which runs on a thread so won't be affected by sleep.

Easing equations you can pass in:

Back.easeIn Bounce.easeIn Circ.easeIn Cubic.easeIn Elastic.easeIn Expo.easeIn Linear.easeIn Quad.easeIn
 Quart.easeIn Quint.easeIn Sine.easeIn

Back.easeOut Bounce.easeOut Circ.easeOut Cubic.easeOut Elastic.easeOut Expo.easeOut Linear.easeOut
 Quad.easeOut Quart.easeOut Quint.easeOut Sine.easeOut

Back.easeInOut Bounce.easeInOut Circ.easeInOut Cubic.easeInOut Elastic.easeInOut Expo.easeInOut Lin-
 ear.easeInOut Quad.easeInOut Quart.easeInOut Quint.easeInOut Sine.easeInOut

get_timer

The tween engine makes use of timer method which shows how long the game has been running.

```

from domonic.lerpy import get_timer
print(get_timer())

```

```
class domonic.lerpy.tween.Tween(target=None, values=None, duration=0, equations=None, delay=0,
                                loop=False)
```

Tween is a complex lerp but you don't have to do the math cos robert penner did already. Just pass an easing equation from the easing package

i.e

```
tw = Tween(someObj, { 'x':10, 'y':5, 'z':3 }, 6, Linear.easeIn )
```

will tween the objects properties over the given time using `Linear.easeIn`

pause()

Pauses the tween from changing the values

unpause()

unpauses the tween

1.4.13 geom

I've been using `.ai` to casually create a geom package.

It's already amazing. You can add shapes together etc.

Changes a lot atm so not documented yet.

domonic.geom

written by.ai

class domonic.geom.matrix(*m*)

[matrixs]

rotate(*pt*)

Rotates the point on the vector defined by the vectors `m[0]` and `m[1]`.

scale(*pt*)

Scales the point on the vector defined by the vectors `m[0]` and `m[1]`.

translate(*pt*)

Translates the point on the vector defined by the vectors `m[0]` and `m[1]`.

1.4.14 x3d

If using `x3d` you must use 'append' rather than 'html' when adding children 'inline' when templating.

This is because they are 'Nodes' not 'Elements' so won't inherit that custom innerHTML shortcut.

Instead they currently Mixin the 'ParentNode' which grants them 'append' and 'prepend' methods

See below...

```
from domonic.xml.x3d import *

x3d(_width='500px', _height='400px').append(
    scene(
        transform(_DEF="ball").append(
            shape(
                appearance(
                    material(_diffuseColor='1 0 0')
                ),
                sphere()
            )
        ),
        timeSensor(_DEF="time", _cycleInterval="2", _loop="true"),
```

(continues on next page)

(continued from previous page)

```

        PositionInterpolator(_DEF="move", _key="0 0.5 1", _keyValue="0 0 0 0 3 0 0 0 0
↪"),
        Route(_fromNode="time", _fromField ="fraction_changed", _toNode="move", _toField=
↪"set_fraction"),
        Route(_fromNode="move", _fromField ="value_changed", _toNode="ball", _toField=
↪"translation")
    )
)

```

Alternatively you can just put them directly in the first parameter and move your kwargs to the end.

aframe

aframe is a simliar library and its tags can be used if you import the js

```

from domonic.html import *
from domonic.xml.aframe import *
from domonic.CDN import *

_scene = scene(
    box(_position="-1 0.5 -3", _rotation="0 45 0", _color="#4CC3D9"),
    sphere(_position="0 1.25 -5", _radius="1.25", _color="#EF2D5E"),
    cylinder(_position="1 0.75 -3", _radius="0.5", _height="1.5", _color="#FFC65D"),
    plane(_position="0 0 -4", _rotation="-90 0 0", _width="4", _height="4", _color="
↪#7BC8A4"),
    sky(_color="#ECECEC")
)

_webpage = html(head(),body(
    script(_src=CDN_JS.AFRAME_1_2), # < NOTICE you need to import aframe to use it
    str(_scene)
)
)

render( _webpage, 'hello.html' )

```

domonic.x3d

Generate x3d with python 3

```
class domonic.xml.x3d.Anchor(*args, **kwargs)
```

domonic.xml.x3d.Appearance

alias of *appearance*

```
class domonic.xml.x3d.Arc2D(*args, **kwargs)
```

```
class domonic.xml.x3d.ArcClose2D(*args, **kwargs)
```

```
class domonic.xml.x3d.AudioClip(*args, **kwargs)
```

```
class domonic.xml.x3d.Background(*args, **kwargs)
```

```
class domonic.xml.x3d.BallJoint(*args, **kwargs)
```

```
class domonic.xml.x3d.Billboard(*args, **kwargs)
class domonic.xml.x3d.BinaryGeometry(*args, **kwargs)
class domonic.xml.x3d.BlendMode(*args, **kwargs)
class domonic.xml.x3d.BlendedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Block(*args, **kwargs)
class domonic.xml.x3d.BoundaryEnhancementVolumeStyle(*args, **kwargs)
domonic.xml.x3d.Box
    alias of box
class domonic.xml.x3d.BufferAccessor(*args, **kwargs)
class domonic.xml.x3d.BufferGeometry(*args, **kwargs)
class domonic.xml.x3d.BufferView(*args, **kwargs)
class domonic.xml.x3d.CADAssembly(*args, **kwargs)
class domonic.xml.x3d.CADFace(*args, **kwargs)
class domonic.xml.x3d.CADLayer(*args, **kwargs)
class domonic.xml.x3d.CADPart(*args, **kwargs)
class domonic.xml.x3d.CartoonVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Circle2D(*args, **kwargs)
class domonic.xml.x3d.ClipPlane(*args, **kwargs)
class domonic.xml.x3d.CollidableShape(*args, **kwargs)
class domonic.xml.x3d.Collision(*args, **kwargs)
class domonic.xml.x3d.CollisionCollection(*args, **kwargs)
class domonic.xml.x3d.CollisionSensor(*args, **kwargs)
class domonic.xml.x3d.Color(*args, **kwargs)
class domonic.xml.x3d.ColorChaser(*args, **kwargs)
class domonic.xml.x3d.ColorDamper(*args, **kwargs)
class domonic.xml.x3d.ColorInterpolator(*args, **kwargs)
class domonic.xml.x3d.ColorMaskMode(*args, **kwargs)
class domonic.xml.x3d.ColorRGBA(*args, **kwargs)
class domonic.xml.x3d.CommonSurfaceShader(*args, **kwargs)
class domonic.xml.x3d.ComposedCubeMapTexture(*args, **kwargs)
class domonic.xml.x3d.ComposedShader(*args, **kwargs)
```

```
class domonic.xml.x3d.ComposedTexture3D(*args, **kwargs)
class domonic.xml.x3d.ComposedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Cone(*args, **kwargs)
class domonic.xml.x3d.Coordinate(*args, **kwargs)
class domonic.xml.x3d.CoordinateDamper(*args, **kwargs)
class domonic.xml.x3d.CoordinateDouble(*args, **kwargs)
class domonic.xml.x3d.CoordinateInterpolator(*args, **kwargs)
class domonic.xml.x3d.Cylinder(*args, **kwargs)
class domonic.xml.x3d.CylinderSensor(*args, **kwargs)
class domonic.xml.x3d.DepthMode(*args, **kwargs)
class domonic.xml.x3d.DirectionallLight(*args, **kwargs)
class domonic.xml.x3d.Dish(*args, **kwargs)
class domonic.xml.x3d.Disk2D(*args, **kwargs)
class domonic.xml.x3d.DoubleAxisHingeJoint(*args, **kwargs)
class domonic.xml.x3d.DynamicLOD(*args, **kwargs)
class domonic.xml.x3d.EdgeEnhancementVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.ElevationGrid(*args, **kwargs)
class domonic.xml.x3d.Environment(*args, **kwargs)
class domonic.xml.x3d.Extrusion(*args, **kwargs)
class domonic.xml.x3d.Field(*args, **kwargs)
class domonic.xml.x3d.FloatVertexAttribute(*args, **kwargs)
class domonic.xml.x3d.Fog(*args, **kwargs)
class domonic.xml.x3d.FontStyle(*args, **kwargs)
class domonic.xml.x3d.GeneratedCubeMapTexture(*args, **kwargs)
class domonic.xml.x3d.GeoCoordinate(*args, **kwargs)
class domonic.xml.x3d.GeoElevationGrid(*args, **kwargs)
class domonic.xml.x3d.GeoLOD(*args, **kwargs)
class domonic.xml.x3d.GeoLocation(*args, **kwargs)
class domonic.xml.x3d.GeoMetadata(*args, **kwargs)
class domonic.xml.x3d.GeoOrigin(*args, **kwargs)
class domonic.xml.x3d.GeoPositionInterpolator(*args, **kwargs)
```

```
class domonic.xml.x3d.GeoTransform(*args, **kwargs)
class domonic.xml.x3d.GeoViewpoint(*args, **kwargs)
class domonic.xml.x3d.Group(*args, **kwargs)
class domonic.xml.x3d.HAnimDisplacer(*args, **kwargs)
class domonic.xml.x3d.HAnimHumanoid(*args, **kwargs)
class domonic.xml.x3d.HAnimJoint(*args, **kwargs)
class domonic.xml.x3d.HAnimSegment(*args, **kwargs)
class domonic.xml.x3d.HAnimSite(*args, **kwargs)
class domonic.xml.x3d.ImageTexture(*args, **kwargs)
class domonic.xml.x3d.ImageTexture3D(*args, **kwargs)
class domonic.xml.x3d.ImageTextureAtlas(*args, **kwargs)
class domonic.xml.x3d.IndexedFaceSet(*args, **kwargs)
class domonic.xml.x3d.IndexedLineSet(*args, **kwargs)
class domonic.xml.x3d.IndexedQuadSet(*args, **kwargs)
class domonic.xml.x3d.IndexedTriangleSet(*args, **kwargs)
class domonic.xml.x3d.IndexedTriangleStripSet(*args, **kwargs)
domonic.xml.x3d.Inline
    alias of inline
class domonic.xml.x3d.IsoSurfaceVolumeData(*args, **kwargs)
class domonic.xml.x3d.LOD(*args, **kwargs)
class domonic.xml.x3d.LineProperties(*args, **kwargs)
class domonic.xml.x3d.LineSet(*args, **kwargs)
class domonic.xml.x3d.MPRPlane(*args, **kwargs)
class domonic.xml.x3d.MPRVolumeStyle(*args, **kwargs)
domonic.xml.x3d.Material
    alias of material
class domonic.xml.x3d.MatrixTextureTransform(*args, **kwargs)
class domonic.xml.x3d.MatrixTransform(*args, **kwargs)
class domonic.xml.x3d.Mesh(*args, **kwargs)
class domonic.xml.x3d.MetadataBoolean(*args, **kwargs)
class domonic.xml.x3d.MetadataDouble(*args, **kwargs)
class domonic.xml.x3d.MetadataFloat(*args, **kwargs)
```

```

class domonic.xml.x3d.MetadataInteger(*args, **kwargs)
class domonic.xml.x3d.MetadataSet(*args, **kwargs)
class domonic.xml.x3d.MetadataString(*args, **kwargs)
class domonic.xml.x3d.MotorJoint(*args, **kwargs)
class domonic.xml.x3d.MovieTexture(*args, **kwargs)
class domonic.xml.x3d.MultiTexture(*args, **kwargs)
class domonic.xml.x3d.MultiTextureCoordinate(*args, **kwargs)
class domonic.xml.x3d.NavigationInfo(*args, **kwargs)
class domonic.xml.x3d.Normal(*args, **kwargs)
class domonic.xml.x3d.NormalInterpolator(*args, **kwargs)
class domonic.xml.x3d.Nozzle(*args, **kwargs)
class domonic.xml.x3d.OpacityMapVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.OrientationChaser(*args, **kwargs)
class domonic.xml.x3d.OrientationDamper(*args, **kwargs)
class domonic.xml.x3d.OrientationInterpolator(*args, **kwargs)
class domonic.xml.x3d.OrthoViewpoint(*args, **kwargs)
class domonic.xml.x3d.Param(*args, **kwargs)
class domonic.xml.x3d.ParticleSet(*args, **kwargs)
class domonic.xml.x3d.PhysicalEnvironmentLight(*args, **kwargs)
class domonic.xml.x3d.PhysicalMaterial(*args, **kwargs)
class domonic.xml.x3d.PixelTexture(*args, **kwargs)
class domonic.xml.x3d.PixelTexture3D(*args, **kwargs)
domonic.xml.x3d.Plane
    alias of plane
class domonic.xml.x3d.PlaneSensor(*args, **kwargs)
class domonic.xml.x3d.PointLight(*args, **kwargs)
class domonic.xml.x3d.PointSet(*args, **kwargs)
class domonic.xml.x3d.Polyline2D(*args, **kwargs)
class domonic.xml.x3d.Polypoint2D(*args, **kwargs)
class domonic.xml.x3d.PopGeometry(*args, **kwargs)
class domonic.xml.x3d.PopGeometryLevel(*args, **kwargs)

```

```
class domonic.xml.x3d.PositionChaser(*args, **kwargs)
class domonic.xml.x3d.PositionChaser2D(*args, **kwargs)
class domonic.xml.x3d.PositionDamper(*args, **kwargs)
class domonic.xml.x3d.PositionDamper2D(*args, **kwargs)
class domonic.xml.x3d.PositionInterpolator(*args, **kwargs)
class domonic.xml.x3d.PositionInterpolator2D(*args, **kwargs)
class domonic.xml.x3d.ProjectionVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Pyramid(*args, **kwargs)
class domonic.xml.x3d.QuadSet(*args, **kwargs)
class domonic.xml.x3d.RadarVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Rectangle2D(*args, **kwargs)
class domonic.xml.x3d.RectangularTorus(*args, **kwargs)
class domonic.xml.x3d.RefinementTexture(*args, **kwargs)
class domonic.xml.x3d.RemoteSelectionGroup(*args, **kwargs)
class domonic.xml.x3d.RenderedTexture(*args, **kwargs)
class domonic.xml.x3d.RigidBody(*args, **kwargs)
class domonic.xml.x3d.RigidBodyCollection(*args, **kwargs)
class domonic.xml.x3d.Route(*args, **kwargs)
class domonic.xml.x3d.ScalarChaser(*args, **kwargs)
class domonic.xml.x3d.ScalarDamper(*args, **kwargs)
class domonic.xml.x3d.ScalarInterpolator(*args, **kwargs)
domonic.xml.x3d.Scene
    alias of scene
class domonic.xml.x3d.SegmentedVolumeData(*args, **kwargs)
class domonic.xml.x3d.ShadedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.ShaderPart(*args, **kwargs)
domonic.xml.x3d.Shape
    alias of shape
class domonic.xml.x3d.SilhouetteEnhancementVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.SingleAxisHingeJoint(*args, **kwargs)
class domonic.xml.x3d.SliderJoint(*args, **kwargs)
class domonic.xml.x3d.SlopedCylinder(*args, **kwargs)
```



```

class domonic.xml.x3d.Snout(*args, **kwargs)
class domonic.xml.x3d.SolidOfRevolution(*args, **kwargs)
class domonic.xml.x3d.Sound(*args, **kwargs)
domonic.xml.x3d.Sphere
    alias of sphere
class domonic.xml.x3d.SphereSegment(*args, **kwargs)
class domonic.xml.x3d.SphereSensor(*args, **kwargs)
class domonic.xml.x3d.SplinePositionInterpolator(*args, **kwargs)
class domonic.xml.x3d.SpotLight(*args, **kwargs)
class domonic.xml.x3d.StaticGroup(*args, **kwargs)
class domonic.xml.x3d.StippleVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.SurfaceShaderTexture(*args, **kwargs)
class domonic.xml.x3d.Switch(*args, **kwargs)
class domonic.xml.x3d.TexCoordDamper2D(*args, **kwargs)
class domonic.xml.x3d.Text(*args, **kwargs)
class domonic.xml.x3d.Texture(*args, **kwargs)
class domonic.xml.x3d.TextureCoordinate(*args, **kwargs)
class domonic.xml.x3d.TextureCoordinate3D(*args, **kwargs)
class domonic.xml.x3d.TextureCoordinateGenerator(*args, **kwargs)
class domonic.xml.x3d.TextureProperties(*args, **kwargs)
class domonic.xml.x3d.TextureTransform(*args, **kwargs)
class domonic.xml.x3d.TextureTransform3D(*args, **kwargs)
class domonic.xml.x3d.TextureTransformMatrix3D(*args, **kwargs)
domonic.xml.x3d.TimeSensor
    alias of timeSensor
class domonic.xml.x3d.ToneMappedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Torus(*args, **kwargs)
class domonic.xml.x3d.TouchSensor(*args, **kwargs)
domonic.xml.x3d.Transform
    alias of transform
class domonic.xml.x3d.TriangleSet(*args, **kwargs)
class domonic.xml.x3d.TriangleSet2D(*args, **kwargs)

```

```
class domonic.xml.x3d.TwoSidedMaterial(*args, **kwargs)
class domonic.xml.x3d.Uniform(*args, **kwargs)
class domonic.xml.x3d.UniversalJoint(*args, **kwargs)
class domonic.xml.x3d.Viewfrustum(*args, **kwargs)
class domonic.xml.x3d.Viewpoint(*args, **kwargs)
class domonic.xml.x3d.VolumeData(*args, **kwargs)
class domonic.xml.x3d.WorldInfo(*args, **kwargs)
domonic.xml.x3d.X3D
    alias of x3d
class domonic.xml.x3d.X3DAppearanceChildNode(*args, **kwargs)
class domonic.xml.x3d.X3DAppearanceNode(*args, **kwargs)
class domonic.xml.x3d.X3DBackgroundNode(*args, **kwargs)
class domonic.xml.x3d.X3DBinaryContainerGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DBindableNode(*args, **kwargs)
class domonic.xml.x3d.X3DBoundedObject(*args, **kwargs)
class domonic.xml.x3d.X3DChaserNode(*args, **kwargs)
class domonic.xml.x3d.X3DChildNode(*args, **kwargs)
class domonic.xml.x3d.X3DColorNode(*args, **kwargs)
class domonic.xml.x3d.X3DComposableVolumeRenderStyleNode(*args, **kwargs)
class domonic.xml.x3d.X3DComposedGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DCoordinateNode(*args, **kwargs)
class domonic.xml.x3d.X3DDamperNode(*args, **kwargs)
class domonic.xml.x3d.X3DDragSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DEnvironmentNode(*args, **kwargs)
class domonic.xml.x3d.X3DEnvironmentTextureNode(*args, **kwargs)
class domonic.xml.x3d.X3DFogNode(*args, **kwargs)
class domonic.xml.x3d.X3DFollowerNode(*args, **kwargs)
class domonic.xml.x3d.X3DFontStyleNode(*args, **kwargs)
class domonic.xml.x3d.X3DGeometricPropertyNode(*args, **kwargs)
class domonic.xml.x3d.X3DGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DGroupingNode(*args, **kwargs)
```

```

class domonic.xml.x3d.X3DInfoNode(*args, **kwargs)
class domonic.xml.x3d.X3DInterpolatorNode(*args, **kwargs)
class domonic.xml.x3d.X3DLODNode(*args, **kwargs)
class domonic.xml.x3d.X3DLightNode(*args, **kwargs)
class domonic.xml.x3d.X3DMaterialNode(*args, **kwargs)
class domonic.xml.x3d.X3DMetadataObject(*args, **kwargs)
class domonic.xml.x3d.X3DNBodyCollidableNode(*args, **kwargs)
class domonic.xml.x3d.X3DNavigationInfoNode(*args, **kwargs)
class domonic.xml.x3d.X3DNode(*args, **kwargs)
class domonic.xml.x3d.X3DPlanarGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DPointingDeviceSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DRigidJointNode(*args, **kwargs)
class domonic.xml.x3d.X3DSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DShaderNode(*args, **kwargs)
class domonic.xml.x3d.X3DShapeNode(*args, **kwargs)
class domonic.xml.x3d.X3DSoundNode(*args, **kwargs)
class domonic.xml.x3d.X3DSoundSourceNode(*args, **kwargs)
class domonic.xml.x3d.X3DSpatialGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DTexture3DNode(*args, **kwargs)
class domonic.xml.x3d.X3DTextureCoordinateNode(*args, **kwargs)
class domonic.xml.x3d.X3DTextureNode(*args, **kwargs)
class domonic.xml.x3d.X3DTextureTransformNode(*args, **kwargs)
class domonic.xml.x3d.X3DTimeDependentNode(*args, **kwargs)
class domonic.xml.x3d.X3DTouchSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DTransformNode(*args, **kwargs)
class domonic.xml.x3d.X3DVertexAttributeNode(*args, **kwargs)
class domonic.xml.x3d.X3DViewpointNode(*args, **kwargs)
class domonic.xml.x3d.X3DVolumeDataNode(*args, **kwargs)
class domonic.xml.x3d.X3DVolumeRenderStyleNode(*args, **kwargs)
domonic.xml.x3d.anchor
    alias of Anchor

```

class domonic.xml.x3d.appearance(*args, **kwargs)

domonic.xml.x3d.arc2D

alias of *Arc2D*

domonic.xml.x3d.arcClose2D

alias of *ArcClose2D*

domonic.xml.x3d.audioClip

alias of *AudioClip*

domonic.xml.x3d.background

alias of *Background*

domonic.xml.x3d.ballJoint

alias of *BallJoint*

domonic.xml.x3d.billboard

alias of *Billboard*

domonic.xml.x3d.binaryGeometry

alias of *BinaryGeometry*

domonic.xml.x3d.blendMode

alias of *BlendMode*

domonic.xml.x3d.blendedVolumeStyle

alias of *BlendedVolumeStyle*

domonic.xml.x3d.block

alias of *Block*

domonic.xml.x3d.boundaryEnhancementVolumeStyle

alias of *BoundaryEnhancementVolumeStyle*

class domonic.xml.x3d.box(*args, **kwargs)

domonic.xml.x3d.bufferAccessor

alias of *BufferAccessor*

domonic.xml.x3d.bufferGeometry

alias of *BufferGeometry*

domonic.xml.x3d.bufferView

alias of *BufferView*

domonic.xml.x3d.cADAssembly

alias of *CADAssembly*

domonic.xml.x3d.cADFace

alias of *CADFace*

domonic.xml.x3d.cADLayer

alias of *CADLayer*

domonic.xml.x3d.cADPart

alias of *CADPart*

`domonic.xml.x3d.cartoonVolumeStyle`
alias of *CartoonVolumeStyle*

`domonic.xml.x3d.circle2D`
alias of *Circle2D*

`domonic.xml.x3d.clipPlane`
alias of *ClipPlane*

`domonic.xml.x3d.collidableShape`
alias of *CollidableShape*

`domonic.xml.x3d.collision`
alias of *Collision*

`domonic.xml.x3d.collisionCollection`
alias of *CollisionCollection*

`domonic.xml.x3d.collisionSensor`
alias of *CollisionSensor*

`domonic.xml.x3d.color`
alias of *Color*

`domonic.xml.x3d.colorChaser`
alias of *ColorChaser*

`domonic.xml.x3d.colorDamper`
alias of *ColorDamper*

`domonic.xml.x3d.colorInterpolator`
alias of *ColorInterpolator*

`domonic.xml.x3d.colorMaskMode`
alias of *ColorMaskMode*

`domonic.xml.x3d.colorRGBA`
alias of *ColorRGBA*

`domonic.xml.x3d.commonSurfaceShader`
alias of *CommonSurfaceShader*

`domonic.xml.x3d.composedCubeMapTexture`
alias of *ComposedCubeMapTexture*

`domonic.xml.x3d.composedShader`
alias of *ComposedShader*

`domonic.xml.x3d.composedTexture3D`
alias of *ComposedTexture3D*

`domonic.xml.x3d.composedVolumeStyle`
alias of *ComposedVolumeStyle*

`domonic.xml.x3d.cone`
alias of *Cone*

`domonic.xml.x3d.coordinate`
alias of *Coordinate*

domonic.xml.x3d.coordinateDamper
alias of *CoordinateDamper*

domonic.xml.x3d.coordinateDouble
alias of *CoordinateDouble*

domonic.xml.x3d.coordinateInterpolator
alias of *CoordinateInterpolator*

domonic.xml.x3d.cylinder
alias of *Cylinder*

domonic.xml.x3d.cylinderSensor
alias of *CylinderSensor*

domonic.xml.x3d.depthMode
alias of *DepthMode*

domonic.xml.x3d.directionalLight
alias of *DirectionalLight*

domonic.xml.x3d.dish
alias of *Dish*

domonic.xml.x3d.disk2D
alias of *Disk2D*

domonic.xml.x3d.doubleAxisHingeJoint
alias of *DoubleAxisHingeJoint*

domonic.xml.x3d.dynamicLOD
alias of *DynamicLOD*

domonic.xml.x3d.edgeEnhancementVolumeStyle
alias of *EdgeEnhancementVolumeStyle*

domonic.xml.x3d.elevationGrid
alias of *ElevationGrid*

domonic.xml.x3d.environment
alias of *Environment*

domonic.xml.x3d.extrusion
alias of *Extrusion*

domonic.xml.x3d.field
alias of *Field*

domonic.xml.x3d.floatVertexAttribute
alias of *FloatVertexAttribute*

domonic.xml.x3d.fog
alias of *Fog*

domonic.xml.x3d.fontStyle
alias of *FontStyle*

`domonic.xml.x3d.generatedCubeMapTexture`

alias of *GeneratedCubeMapTexture*

`domonic.xml.x3d.geoCoordinate`

alias of *GeoCoordinate*

`domonic.xml.x3d.geoElevationGrid`

alias of *GeoElevationGrid*

`domonic.xml.x3d.geoLOD`

alias of *GeoLOD*

`domonic.xml.x3d.geoLocation`

alias of *GeoLocation*

`domonic.xml.x3d.geoMetadata`

alias of *GeoMetadata*

`domonic.xml.x3d.geoOrigin`

alias of *GeoOrigin*

`domonic.xml.x3d.geoPositionInterpolator`

alias of *GeoPositionInterpolator*

`domonic.xml.x3d.geoTransform`

alias of *GeoTransform*

`domonic.xml.x3d.geoViewpoint`

alias of *GeoViewpoint*

`domonic.xml.x3d.group`

alias of *Group*

`domonic.xml.x3d.hAnimDisplacer`

alias of *HAnimDisplacer*

`domonic.xml.x3d.hAnimHumanoid`

alias of *HAnimHumanoid*

`domonic.xml.x3d.hAnimJoint`

alias of *HAnimJoint*

`domonic.xml.x3d.hAnimSegment`

alias of *HAnimSegment*

`domonic.xml.x3d.hAnimSite`

alias of *HAnimSite*

`domonic.xml.x3d.imageTexture`

alias of *ImageTexture*

`domonic.xml.x3d.imageTexture3D`

alias of *ImageTexture3D*

`domonic.xml.x3d.imageTextureAtlas`

alias of *ImageTextureAtlas*

`domonic.xml.x3d.indexedFaceSet`

alias of *IndexedFaceSet*

domonic.xml.x3d.indexedLineSet

alias of *IndexedLineSet*

domonic.xml.x3d.indexedQuadSet

alias of *IndexedQuadSet*

domonic.xml.x3d.indexedTriangleSet

alias of *IndexedTriangleSet*

domonic.xml.x3d.indexedTriangleStripSet

alias of *IndexedTriangleStripSet*

class domonic.xml.x3d.inline(*args, **kwargs)

domonic.xml.x3d.isoSurfaceVolumeData

alias of *IsoSurfaceVolumeData*

domonic.xml.x3d.lod

alias of *LOD*

domonic.xml.x3d.lineProperties

alias of *LineProperties*

domonic.xml.x3d.lineSet

alias of *LineSet*

domonic.xml.x3d.mprPlane

alias of *MPRPlane*

domonic.xml.x3d.mprVolumeStyle

alias of *MPRVolumeStyle*

class domonic.xml.x3d.material(*args, **kwargs)

domonic.xml.x3d.matrixTextureTransform

alias of *MatrixTextureTransform*

domonic.xml.x3d.matrixTransform

alias of *MatrixTransform*

domonic.xml.x3d.mesh

alias of *Mesh*

domonic.xml.x3d.metadataBoolean

alias of *MetadataBoolean*

domonic.xml.x3d.metadataDouble

alias of *MetadataDouble*

domonic.xml.x3d.metadataFloat

alias of *MetadataFloat*

domonic.xml.x3d.metadataInteger

alias of *MetadataInteger*

domonic.xml.x3d.metadataSet

alias of *MetadataSet*

`domonic.xml.x3d.metadataString`
alias of *MetadataString*

`domonic.xml.x3d.motorJoint`
alias of *MotorJoint*

`domonic.xml.x3d.movieTexture`
alias of *MovieTexture*

`domonic.xml.x3d.multiTexture`
alias of *MultiTexture*

`domonic.xml.x3d.multiTextureCoordinate`
alias of *MultiTextureCoordinate*

`domonic.xml.x3d.navigationInfo`
alias of *NavigationInfo*

`domonic.xml.x3d.normal`
alias of *Normal*

`domonic.xml.x3d.normalInterpolator`
alias of *NormalInterpolator*

`domonic.xml.x3d.nozzle`
alias of *Nozzle*

`domonic.xml.x3d.opacityMapVolumeStyle`
alias of *OpacityMapVolumeStyle*

`domonic.xml.x3d.orientationChaser`
alias of *OrientationChaser*

`domonic.xml.x3d.orientationDamper`
alias of *OrientationDamper*

`domonic.xml.x3d.orientationInterpolator`
alias of *OrientationInterpolator*

`domonic.xml.x3d.orthoViewpoint`
alias of *OrthoViewpoint*

`domonic.xml.x3d.param`
alias of *Param*

`domonic.xml.x3d.particleSet`
alias of *ParticleSystem*

`domonic.xml.x3d.physicalEnvironmentLight`
alias of *PhysicalEnvironmentLight*

`domonic.xml.x3d.physicalMaterial`
alias of *PhysicalMaterial*

`domonic.xml.x3d.pixelTexture`
alias of *PixelTexture*

`domonic.xml.x3d.pixelTexture3D`
alias of *PixelTexture3D*

class domonic.xml.x3d.plane(*args, **kwargs)

domonic.xml.x3d.planeSensor

alias of *PlaneSensor*

domonic.xml.x3d.pointLight

alias of *PointLight*

domonic.xml.x3d.pointSet

alias of *PointSet*

domonic.xml.x3d.polyline2D

alias of *Polyline2D*

domonic.xml.x3d.polypoint2D

alias of *Polypoint2D*

domonic.xml.x3d.popGeometry

alias of *PopGeometry*

domonic.xml.x3d.popGeometryLevel

alias of *PopGeometryLevel*

domonic.xml.x3d.positionChaser

alias of *PositionChaser*

domonic.xml.x3d.positionChaser2D

alias of *PositionChaser2D*

domonic.xml.x3d.positionDamper

alias of *PositionDamper*

domonic.xml.x3d.positionDamper2D

alias of *PositionDamper2D*

domonic.xml.x3d.positionInterpolator

alias of *PositionInterpolator*

domonic.xml.x3d.positionInterpolator2D

alias of *PositionInterpolator2D*

domonic.xml.x3d.projectionVolumeStyle

alias of *ProjectionVolumeStyle*

domonic.xml.x3d.pyramid

alias of *Pyramid*

domonic.xml.x3d.quadSet

alias of *QuadSet*

domonic.xml.x3d.radarVolumeStyle

alias of *RadarVolumeStyle*

domonic.xml.x3d.rectangle2D

alias of *Rectangle2D*

domonic.xml.x3d.rectangularTorus

alias of *RectangularTorus*

domonic.xml.x3d.refinementTexture
alias of *RefinementTexture*

domonic.xml.x3d.remoteSelectionGroup
alias of *RemoteSelectionGroup*

domonic.xml.x3d.renderedTexture
alias of *RenderedTexture*

domonic.xml.x3d.rigidBody
alias of *RigidBody*

domonic.xml.x3d.rigidBodyCollection
alias of *RigidBodyCollection*

domonic.xml.x3d.route
alias of *Route*

domonic.xml.x3d.scalarChaser
alias of *ScalarChaser*

domonic.xml.x3d.scalarDamper
alias of *ScalarDamper*

domonic.xml.x3d.scalarInterpolator
alias of *ScalarInterpolator*

class domonic.xml.x3d.scene(*args, **kwargs)

domonic.xml.x3d.segmentedVolumeData
alias of *SegmentedVolumeData*

domonic.xml.x3d.shadedVolumeStyle
alias of *ShadedVolumeStyle*

domonic.xml.x3d.shaderPart
alias of *ShaderPart*

class domonic.xml.x3d.shape(*args, **kwargs)

domonic.xml.x3d.silhouetteEnhancementVolumeStyle
alias of *SilhouetteEnhancementVolumeStyle*

domonic.xml.x3d.singleAxisHingeJoint
alias of *SingleAxisHingeJoint*

domonic.xml.x3d.sliderJoint
alias of *SliderJoint*

domonic.xml.x3d.slopedCylinder
alias of *SlopedCylinder*

domonic.xml.x3d.snout
alias of *Snout*

domonic.xml.x3d.solidOfRevolution
alias of *SolidOfRevolution*

`domonic.xml.x3d.sound`
alias of *Sound*

class `domonic.xml.x3d.sphere(*args, **kwargs)`

`domonic.xml.x3d.sphereSegment`
alias of *SphereSegment*

`domonic.xml.x3d.sphereSensor`
alias of *SphereSensor*

`domonic.xml.x3d.splinePositionInterpolator`
alias of *SplinePositionInterpolator*

`domonic.xml.x3d.spotLight`
alias of *SpotLight*

`domonic.xml.x3d.staticGroup`
alias of *StaticGroup*

`domonic.xml.x3d.stippleVolumeStyle`
alias of *StippleVolumeStyle*

`domonic.xml.x3d.surfaceShaderTexture`
alias of *SurfaceShaderTexture*

`domonic.xml.x3d.switch`
alias of *Switch*

`domonic.xml.x3d.texCoordDamper2D`
alias of *TexCoordDamper2D*

`domonic.xml.x3d.text`
alias of *Text*

`domonic.xml.x3d.texture`
alias of *Texture*

`domonic.xml.x3d.textureCoordinate`
alias of *TextureCoordinate*

`domonic.xml.x3d.textureCoordinate3D`
alias of *TextureCoordinate3D*

`domonic.xml.x3d.textureCoordinateGenerator`
alias of *TextureCoordinateGenerator*

`domonic.xml.x3d.textureProperties`
alias of *TextureProperties*

`domonic.xml.x3d.textureTransform`
alias of *TextureTransform*

`domonic.xml.x3d.textureTransform3D`
alias of *TextureTransform3D*

`domonic.xml.x3d.textureTransformMatrix3D`
alias of *TextureTransformMatrix3D*

```

class domonic.xml.x3d.timeSensor(*args, **kwargs)

domonic.xml.x3d.toneMappedVolumeStyle
    alias of ToneMappedVolumeStyle

domonic.xml.x3d.torus
    alias of Torus

domonic.xml.x3d.touchSensor
    alias of TouchSensor

class domonic.xml.x3d.transform(*args, **kwargs)

domonic.xml.x3d.triangleSet
    alias of TriangleSet

domonic.xml.x3d.triangleSet2D
    alias of TriangleSet2D

domonic.xml.x3d.twoSidedMaterial
    alias of TwoSidedMaterial

domonic.xml.x3d.uniform
    alias of Uniform

domonic.xml.x3d.universalJoint
    alias of UniversalJoint

domonic.xml.x3d.viewfrustum
    alias of Viewfrustum

domonic.xml.x3d.viewpoint
    alias of Viewpoint

domonic.xml.x3d.volumeData
    alias of VolumeData

domonic.xml.x3d.worldInfo
    alias of WorldInfo

domonic.xml.x3d.x3DAppearanceChildNode
    alias of X3DAppearanceChildNode

domonic.xml.x3d.x3DAppearanceNode
    alias of X3DAppearanceNode

domonic.xml.x3d.x3DBackgroundNode
    alias of X3DBackgroundNode

domonic.xml.x3d.x3DBinaryContainerGeometryNode
    alias of X3DBinaryContainerGeometryNode

domonic.xml.x3d.x3DBindableNode
    alias of X3DBindableNode

domonic.xml.x3d.x3DBoundedObject
    alias of X3DBoundedObject

```

domonic.xml.x3d.**x3DChaserNode**
alias of *X3DChaserNode*

domonic.xml.x3d.**x3DChildNode**
alias of *X3DChildNode*

domonic.xml.x3d.**x3DColorNode**
alias of *X3DColorNode*

domonic.xml.x3d.**x3DComposableVolumeRenderStyleNode**
alias of *X3DComposableVolumeRenderStyleNode*

domonic.xml.x3d.**x3DComposedGeometryNode**
alias of *X3DComposedGeometryNode*

domonic.xml.x3d.**x3DCoordinateNode**
alias of *X3DCoordinateNode*

domonic.xml.x3d.**x3DDamperNode**
alias of *X3DDamperNode*

domonic.xml.x3d.**x3DDragSensorNode**
alias of *X3DDragSensorNode*

domonic.xml.x3d.**x3DEnvironmentNode**
alias of *X3DEnvironmentNode*

domonic.xml.x3d.**x3DEnvironmentTextureNode**
alias of *X3DEnvironmentTextureNode*

domonic.xml.x3d.**x3DFogNode**
alias of *X3DFogNode*

domonic.xml.x3d.**x3DFollowerNode**
alias of *X3DFollowerNode*

domonic.xml.x3d.**x3DFontStyleNode**
alias of *X3DFontStyleNode*

domonic.xml.x3d.**x3DGeometricPropertyNode**
alias of *X3DGeometricPropertyNode*

domonic.xml.x3d.**x3DGeometryNode**
alias of *X3DGeometryNode*

domonic.xml.x3d.**x3DGroupingNode**
alias of *X3DGroupingNode*

domonic.xml.x3d.**x3DInfoNode**
alias of *X3DInfoNode*

domonic.xml.x3d.**x3DInterpolatorNode**
alias of *X3DInterpolatorNode*

domonic.xml.x3d.**x3DLODNode**
alias of *X3DLODNode*

domonic.xml.x3d.x3DLightNode
alias of *X3DLightNode*

domonic.xml.x3d.x3DMaterialNode
alias of *X3DMaterialNode*

domonic.xml.x3d.x3DMetadataObject
alias of *X3DMetadataObject*

domonic.xml.x3d.x3DNBodyCollidableNode
alias of *X3DNBodyCollidableNode*

domonic.xml.x3d.x3DNavigationInfoNode
alias of *X3DNavigationInfoNode*

domonic.xml.x3d.x3DNode
alias of *X3DNode*

domonic.xml.x3d.x3DPlanarGeometryNode
alias of *X3DPlanarGeometryNode*

domonic.xml.x3d.x3DPointingDeviceSensorNode
alias of *X3DPointingDeviceSensorNode*

domonic.xml.x3d.x3DRigidJointNode
alias of *X3DRigidJointNode*

domonic.xml.x3d.x3DSensorNode
alias of *X3DSensorNode*

domonic.xml.x3d.x3DShaderNode
alias of *X3DShaderNode*

domonic.xml.x3d.x3DShapeNode
alias of *X3DShapeNode*

domonic.xml.x3d.x3DSoundNode
alias of *X3DSoundNode*

domonic.xml.x3d.x3DSoundSourceNode
alias of *X3DSoundSourceNode*

domonic.xml.x3d.x3DSpatialGeometryNode
alias of *X3DSpatialGeometryNode*

domonic.xml.x3d.x3DTexture3DNode
alias of *X3DTexture3DNode*

domonic.xml.x3d.x3DTextureCoordinateNode
alias of *X3DTextureCoordinateNode*

domonic.xml.x3d.x3DTextureNode
alias of *X3DTextureNode*

domonic.xml.x3d.x3DTextureTransformNode
alias of *X3DTextureTransformNode*

`domonic.xml.x3d.x3DTimeDependentNode`

alias of *X3DTimeDependentNode*

`domonic.xml.x3d.x3DTouchSensorNode`

alias of *X3DTouchSensorNode*

`domonic.xml.x3d.x3DTransformNode`

alias of *X3DTransformNode*

`domonic.xml.x3d.x3DVertexAttributeNode`

alias of *X3DVertexAttributeNode*

`domonic.xml.x3d.x3DViewpointNode`

alias of *X3DViewpointNode*

`domonic.xml.x3d.x3DVolumeDataNode`

alias of *X3DVolumeDataNode*

`domonic.xml.x3d.x3DVolumeRenderStyleNode`

alias of *X3DVolumeRenderStyleNode*

class `domonic.xml.x3d.x3d(*args, **kwargs)`

domonic.aframe

Generate aframe tags with python 3 <https://aframe.io/>

class `domonic.xml.aframe.aframe_tag(*args, **kwargs)`

class `domonic.xml.aframe.alink(*args, **kwargs)`

class `domonic.xml.aframe.appearance(*args, **kwargs)`

class `domonic.xml.aframe.box(*args, **kwargs)`

class `domonic.xml.aframe.camera(*args, **kwargs)`

class `domonic.xml.aframe.circle(*args, **kwargs)`

class `domonic.xml.aframe.cone(*args, **kwargs)`

class `domonic.xml.aframe.cursor(*args, **kwargs)`

class `domonic.xml.aframe.curvedimage(*args, **kwargs)`

class `domonic.xml.aframe.cylinder(*args, **kwargs)`

class `domonic.xml.aframe.dodecahedron(*args, **kwargs)`

class `domonic.xml.aframe.entity(*args, **kwargs)`

class `domonic.xml.aframe.gltf(*args, **kwargs)`

class `domonic.xml.aframe.gltf_model(*args, **kwargs)`

class `domonic.xml.aframe.icosahedron(*args, **kwargs)`

class `domonic.xml.aframe.image(*args, **kwargs)`


```

class domonic.xml.aframe.light(*args, **kwargs)
class domonic.xml.aframe.material(*args, **kwargs)
class domonic.xml.aframe.mixin(*args, **kwargs)
class domonic.xml.aframe.obj_model(*args, **kwargs)
class domonic.xml.aframe.octahedron(*args, **kwargs)
class domonic.xml.aframe.plane(*args, **kwargs)
class domonic.xml.aframe.ring(*args, **kwargs)
class domonic.xml.aframe.scene(*args, **kwargs)
class domonic.xml.aframe.shape(*args, **kwargs)
class domonic.xml.aframe.sky(*args, **kwargs)
class domonic.xml.aframe.sound(*args, **kwargs)
class domonic.xml.aframe.sphere(*args, **kwargs)
class domonic.xml.aframe.tetrahedron(*args, **kwargs)
class domonic.xml.aframe.text(*args, **kwargs)
class domonic.xml.aframe.torus(*args, **kwargs)
class domonic.xml.aframe.torus_knot(*args, **kwargs)
class domonic.xml.aframe.triangle(*args, **kwargs)
class domonic.xml.aframe.video(*args, **kwargs)
class domonic.xml.aframe.videosphere(*args, **kwargs)

```

1.4.15 CDN

For quick reference when prototyping you can use the CDN package.

To use a CDN class it must be imported.

```
from domonic.CDN import *
```

Or just classes you need...

```
from domonic.CDN import CDN_JS, CDN_CSS
```

CDN_JS

```
script(_src=CDN_JS.JQUERY_3_5_1)
```

CDN_CSS

```
classless_css = link(_rel="stylesheet", _href=CDN_CSS.WATER_LATEST)
```

currently constants exist for:

- BOOTSTRAP_5_ALPHA
- BOOTSTRAP_4
- MARX
- MVP
- WATER_LATEST
- BALLOON
- THREE_DOTS_0_2_0
- MILLIGRAM_1_3_0
- X3DOM
- FONTAWESOME_5_7_1
- MDI_5_4_55

CDN_IMG

CDN_IMG has a placeholder service.

```
# to change it. do this...
CDN_IMG.PLACEHOLDER_SERVICE = "placebear.com/g"

img(_src=CDN_IMG.PLACEHOLDER(300,100))

# optional seperator if the site uses x instead of slash between dimensions
img(_src=CDN_IMG.PLACEHOLDER(300,100,'x'))
```

there's tons to pick from. (NOT ALL ARE HTTPS):

- <http://placeholder.it/350x150>
- <http://unsplash.it/200/300>
- <http://lorempixel.com/400/200>
- <http://dummyimage.com/600x300/000/fff>
- <https://dummyimage.com/420x320/ff7f7f/333333.png&text=Sample>
- <http://placekitten.com/200/300>
- <https://placeimg.com/640/480/any>
- <http://placebear.com/g/200/300>

- <https://ipsumimage.appspot.com/140x100, ff7700>
- <https://www.fillmurray.com/640/360>
- <https://baconmockup.com/640/360>
- <https://placebeard.it/640x360>
- <https://www.placecage.com/640/360>
- <https://www.stevensegallery.com/640/360>
- <https://fakeimg.pl/640x360>
- <https://fakeimg.pl/420x320/ff0000,128/333333,255/?text=Sample&font=lobster>
- <https://picsum.photos/640/360>
- <https://via.placeholder.com/420x320/ff7f7f/333333?text=Sample>
- <https://keywordimg.com/420x320/random>
- <http://www.dummysrc.com/430x320.png/22c5fc/17202A>

domonic.CDN

For quick reference when prototyping you can use the CDN package. (Don't rely on a CDN package for production code. wget a local copy.)

TODO - integrity/cross origin/module?

class domonic.CDN.CDN_CSS

Preferably use version numbers if available. use LATEST if it always gets the latest

BALLOON: **str** = 'https://unpkg.com/balloon-css/balloon.min.css'

BOOTSTRAP_4: **str** =
'https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css'

BOOTSTRAP_5_ALPHA: **str** =
'https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js'

FONTAWESOME_5_7_1: **str** = 'https://use.fontawesome.com/releases/v5.7.1/css/all.css'

MARX: **str** = 'https://unpkg.com/marx-css/css/marx.min.css'

MDI_5_4_55: **str** =
'https://cdn.materialdesignicons.com/5.4.55/css/materialdesignicons.min.css'

MILLIGRAM_1_3_0: **str** =
'https://cdnjs.cloudflare.com/ajax/libs/milligram/1.3.0/milligram.css'

MVP: **str** = 'https://unpkg.com/mvp.css'

SIMPLE: **str** = 'https://cdn.simplecss.org/simple.min.css'

TAILWIND_2_2_15: **str** =
'https://unpkg.com/tailwindcss@^2.2.15/dist/tailwind.min.css'

THREE_DOTS_0_2_0: **str** =
'https://cdnjs.cloudflare.com/ajax/libs/three-dots/0.2.0/three-dots.min.css'

```
WATER_LATEST: str =  
'https://cdn.jsdelivr.net/gh/kognise/water.css@latest/water.min.css'
```

```
X3DOM: str = 'https://www.x3dom.org/download/x3dom.css'
```

```
class domonic.CDN.CDN_IMG
```

```
    CDN images
```

```
    static PLACEHOLDER(width: int = 100, height: int = 100, HTTP: str = "", separator: str = '/') → str  
        to update do CDN_IMG.PLACEHOLDER_SERVICE = "placebear.com/g" usage :  
        img(_src=CDN_IMG.PLACEHOLDER(300,100)) default HTTP is none, to let the browser decide # use  
        optional separator if the site uses x instead of slash img(_src=CDN_IMG.PLACEHOLDER(300,100,'x'))
```

```
class domonic.CDN.CDN_JS
```

```
    js libs
```

```
    AFRAME_1_2: str = 'https://aframe.io/releases/1.2.0/aframe.min.js'
```

```
    BOOTSTRAP_4: str =  
'https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js'
```

```
    BOOTSTRAP_5 =  
'https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js'  
    latest
```

```
    BOOTSTRAP_5_ALPHA: str =  
'https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js'
```

```
    BRYTHON_3_9_5: str =  
'https://cdnjs.cloudflare.com/ajax/libs/brython/3.9.5/brython.min.js'
```

```
    D3: str = 'https://cdnjs.cloudflare.com/ajax/libs/d3/7.4.3/d3.min.js'  
    latest
```

```
    D3_6_1_0: str = 'https://cdnjs.cloudflare.com/ajax/libs/d3/6.1.0/d3.min.js'
```

```
    HTMX: str = 'https://unpkg.com/htmx.org@1.7.0'  
    latest
```

```
    JQUERY: str = 'https://code.jquery.com/jquery-3.6.0.min.js'  
    latest
```

```
    JQUERY_3_5_1: str = 'https://code.jquery.com/jquery-3.5.1.min.js'
```

```
    JQUERY_UI: str = 'https://code.jquery.com/ui/1.12.0/jquery-ui.min.js'
```

```
    MATHML: str = 'https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?  
    config=MML_HTMLorMML'
```

```
    MODERNIZER_2_8_3: str =  
'https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js'
```

```
    MOMENT_2_27_0: str =  
'https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.27.0/moment.min.js'
```

```
    PIXI_5_3_3: str =  
'https://cdnjs.cloudflare.com/ajax/libs/pixi.js/5.3.3/pixi.min.js'
```

```

POPPER_1_16_1: str =
'https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js'

SOCKET_1_4_5: str =
'https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.5/socket.io.min.js'

UNDERSCORE: str = 'https://cdn.jsdelivr.net/npm/underscore@1.11.0/underscore-min.js'

X3DOM: str = 'https://www.x3dom.org/download/x3dom.js'

```

domonic.utils

snippets etc

class domonic.utils.Utils

static acronym(sentence: str) → str

[pass a sentence, returns the acronym]

Parameters

sentence ([str]) – [typically 3 words]

Returns

[a TLA (three letter acronym)]

Return type

[str]

static case_camel(s: str) → str

case_camel('camel-case') > 'camelCase'

static case_kebab(s: str) → str

kebab('camelCase') # 'camel-case'

static case_snake(s: str) → str

snake('camelCase') # 'camel_case'

static chunk(list: list, size: int) → list

chunk a list into batches

static chunks(iterable, size: int, format=<built-in function iter>)

Iterate over any iterable (list, set, file, stream, strings, whatever), of ANY size

static clean(lst: list) → list

[removes falsy values (False, None, 0 and "") from a list]

Parameters

lst ([list]) – [lst to operate on]

Returns

[a new list with falsy values removed]

Return type

[list]

static dictify(arr: list) → dict

[turns a list into a dictionary where the list items are the keys]

Parameters**arr** (*[list]*) – [list to change]**Returns**

[a new dict where the list items are now the keys]

Return type

[dict]

static digits(*text: str = ""*) → *str*

[takes a string of mix of digits and letters and returns a string of digits]

Parameters**text** (*str, optional*) – [the text to change]. Defaults to “.”**Returns**

[a string of digits]

Return type

[str]

static escape(*s: str*) → *str*

[escape a string]

Parameters**s** (*[str]*) – [the string to escape]**Returns**

[the escaped string]

Return type

[str]

static frequency(*data*)

[check the frequency of elements in the data]

Parameters**data** (*[type]*) – [the data to check]**Returns**

[a dict of elements and their frequency]

Return type

[dict]

static get_vowels(*string: str*) → *list*

[get a list of vowels from the word]

Parameters**string** (*[str]*) – [the word to check]**Returns**

[a list of vowels]

Return type

[list]

static has_internet(*url: str = 'http://www.google.com/', timeout: int = 5*) → *bool*

[check if you have internet connection]

Parameters• **url** (*str, optional*) – [the url to check]. Defaults to ‘http://www.google.com/’.

- **timeout** (*int*, *optional*) – [the timeout]. Defaults to 5.

Returns

[True if you have internet]

Return type

[bool]

static **init_assets**(*dir*: *str* = 'assets') → None

[creates an assets directory with nested js/css/img dirs]

Parameters

dir (*str*, *optional*) – [default directory name]. Defaults to 'assets'.

static **is_linux**() → bool

[check if the system is a linux]

Returns

[description]

Return type

[bool]

static **is_mac**() → bool

[check if the system is a mac]

Returns

[True if the system is a mac]

Return type

[bool]

static **is_nix**() → bool

[check if the system is a nix based system]

Returns

[True if it is a nix based system]

Return type

[bool]

static **is_windows**() → bool

[check if the system is a windows]

Returns

[True if windows]

Return type

[bool]

static **merge_dictionaries**(*a*: *dict*, *b*: *dict*) → dict

[merges 2 dicts]

Parameters

- **a** ([*dict*]) – [dict a]
- **b** ([*dict*]) – [dict b]

Returns

[a new dict]

Return type

[dict]

static `permutations(word: str) → list`

[provides all the possible permutations of a given word]

Parameters

word (`[str]`) – [the word to get permutations for]

Returns

[a list of permutations]

Return type

[list]

static `replace_between(content: str, match: str, replacement: str, start: int = 0, end: int = 0)`

[replace some text but only between certain indexes]

Parameters

- **content** (`str`) – [the content whos text you will be replacing]
- **match** (`str`) – [the string to find]
- **replacement** (`str`) – [the string to replace it with]
- **start** (`int`, *optional*) – [start index]. Defaults to 0.
- **end** (`int`, *optional*) – [end index]. Defaults to 0.

Returns

[the new string]

Return type

[str]

static `squash(the_list: list) → list`

[turns a 2d array into a flat one]

Parameters

the_list (`[list]`) – [a 2d array]

Returns

[a flattened 1d array]

Return type

[list]

static `to_dictionary(keys: list, values: list) → dict`

[take a list of keys and values and returns a dict]

Parameters

- **keys** (`[list]`) – [a list of keys]
- **values** (`[list]`) – [a list of value]

Returns

[a dictionary]

Return type

[dict]

static `truncate(text: str = "", length: int = 0) → str`

[truncates a string and appends 3 dots]

Parameters

- **text** (*str*, *optional*) – [the text to truncate]. Defaults to ‘.’.
- **length** (*int*, *optional*) – [the max length]. Defaults to 0.

Returns

[the truncated string]

Return type[*str*]**static unescape**(*s*: *str*) → *str*

[unescape a string]

Parameters**s** ([*str*]) – [the string to unescape]**Returns**

[the unescaped string]

Return type[*str*]**static unique**(*some_arr*: *list*) → *list*

[removes duplicates from a list]

Parameters**some_arr** ([*list*]) – [list containing duplicates]**Returns**

[a list containing no duplicates]

Return type[*list*]**static untitled**(*string*: *str*) → *str*

[the opposite of title]

Parameters**str** ([*str*]) – [the string to change]**Returns**

[a string with the first character set to lowercase]

Return type[*str*]**static url2file**(*url*: *str*) → *str*

[gen a safe filename from a url. by replacing ‘/’ for ‘_’ and ‘:’ for ‘__’]

Parameters**url** ([*str*]) – [the url to turn into a filename]**Returns**

[description]

Return type[*str*]

domonic.decorators

`domonic.decorators.as_json(func)`

decorate any function to return json instead of a python obj note - used by JSON.py

`domonic.decorators.called(before=None, error: Optional[Callable[[Exception], None]] = None)`

Decorator to call a function before and after a function call.

`domonic.decorators.check(f)`

Prints entry and exit of a function to the console

`domonic.decorators.deprecated(func)`

marks a function as deprecated.

`domonic.decorators.el(element='div', string: bool = False)`

[wraps the results of a function in an element]

`domonic.decorators.iiife(before=None, error: Optional[Callable[[Exception], None]] = None)`

Decorator to call a function before and after a function call.

`domonic.decorators.instead(f, somethingelse)`

what to return if it fails

`domonic.decorators.log(logger, level='info')`

@log(logging.getLogger('main'), level='warning')

`domonic.decorators.silence(*args, **kwargs)`

stop a function from doing anything

1.4.16 decorators

Everyone loves python decorators

We have a few in domonic to make life more fun!

el

You can use the el decorator to wrap elements around function results.

```
from domonic.decorators import el

@el(html, True)
@el(body)
@el(div)
def test():
    return 'hi!'

print(test())
# <html><body><div>hi!</div></body></html>

# returns pyl objects so call str to render
assert str(test()) == '<html><body><div>hi!</div></body></html>'
```

It returns the tag object by default.

You can pass True as a second param to the decorator to return a rendered string instead. Also accepts strings as first param i.e. custom tags.

silence

Want that unit test to stfu?

```
from domonic.decorators import silence

@silence
def test_that_wont_pass():
    assert True == False
```

called

Python's lambda restrictions may force you to create anonymous success methods above calling functions.

domonic uses a unique type of decorator to call anonymous methods immediately after calling the passed method.

To use it, pass 2 functions, something to call BEFORE hand, and an error method

Then your decorated anonymous function will receive the data of the first function you passed in as a parameter.

Let me show you...

```
from domonic.decorators import called

@called(
    lambda: °.ajax('https://www.google.com'),
    lambda err: print('error:', err))
def success(data=None):
    print("Sweet as a Nut!")
    print(data.text)
```

It's meant for anonymous functions and calls immediately. So don't go using it on class methods.

It's also called iffe. (so you can know when ur just passing nothing)

```
@iife()
def sup():
    print("sup!")
    return True
```

check

logs the entry and exit of a function and is useful for debugging. i.e.

```
@check
def somefunc():
    return True

somefunc()

# would output this to the console
# Entering somefunc
# Exited somefunc
```

domonic.decorators

alias of <module 'domonic.decorators' from '/home/docs/checkouts/readthedocs.org/user_builds/domonic/checkouts/stable/domonic.decorators'>

1.4.17 Templates and Components

With all these pieces you can now build templates or components...

Templates

Some notes on templates

domonic mixed with lambdas can create templates without needing to make a class...

```
# create a template
some_tmpl = lambda somevar: div( _style=f"display:inline;margin:{MARGIN}px;").html(
    button(somevar, _style="background-color:white;color:black;")
)
```

then you can use it like this...

```
some_tmpl("some content")
```

Here is another template. This one is larger and uses a Class. It can take content as input.

```
class Webpage:

    def __init__(self, content=None):
        self.content = content

    def __str__(self):
        classless_css = link(_rel="stylesheet", _href="https://unpkg.com/marx-css/css/
↳ marx.min.css")
        jquery = script(_src="https://code.jquery.com/jquery-3.5.1.min.js"),
        script(_src=domonic.JS_MASTER),
        link(_rel="stylesheet", _type="text/css", _href=domonic.CSS_STYLE),
        code = script('''
            $(document).on( "click", ".close", function() {
                var _id = $(this).data('ref');
                $('#'+_id).css("display","none");
            })
        ''')
```

(continues on next page)

(continued from previous page)

```

    });
    $(document).on( "click", ".open", function() {
        var _id = $(this).data('ref');
        $('#'+_id).css("display","block");
    });

    // pass an ElementID and an endpoint to redraw that div with the endpoints.
↪response
    window.redraw = function( _id, endpoint ){
        $.get( endpoint, function( data ) {
            window.console.log(data)
            $( "#"+_id ).html( $(data).html() );
        });
    }

    ''')
    styles = style(''
        .domonic-container {
            padding:20px;
        }
        .modal {
            display: none;
            position: fixed;
            z-index: 1;
            left: 0;
            top: 0;
            width: 100%;
            height: 100%;
            overflow: auto;
            background-color: rgb(0,0,0);
            background-color: rgba(0,0,0,0.4);
        }
        .modal-content {
            background-color: #fefefe;
            margin: 15% auto;
            padding: 20px;
            border: 1px solid #888;
            width: 80%;
        }
        .btn-sm {
            font-size:10px;
            padding: 0px;
            padding-left: 2px;
            padding-right: 2px;
        }
        .del {
            background-color:red;
        }
        .go {
            background-color:green;
        }

```

(continues on next page)

(continued from previous page)

```
'''
return str(
    html(
        '<!DOCTYPE HTML>',
        head(classless_css, jquery, code, styles),
        body(div(self.content, _class="domonic-container"))
    )
)
```

You can now render your template. Which can take content as input.

```
@app.route('/')
async def home(request):
    page = article(
        div(h1("my homepage!"))
    )
    return response.html( render( Webpage(page) ) )
```

Important notes on templating

while you can create a div with content like :

```
div("some content")
```

python doesn't allow named params before unnamed ones. So you can't do this:

```
div(_class="container", p("Some content") )
```

or it will complain the params are in the wrong order. You have to instead put content before attributes:

```
div( p("Some content"), _class="container")
```

which is annoying when a div gets long.

You can get around this by using 'html' which is available on every Element:

```
div( _class="container" ).html("Some content")
```

This is NOT like jQuery html func that returns just the inner content. use innerHTML for that.

It is used specifically for rendering.

Common Errors

When you first start templating this way you can make a lot of common mistakes. Usually missing underscores or commas between attributes.

Refer back to this page for a few days until you get used to it.

Here are the 4 most common ones I experienced when creating large templates...

(i.e. bootstrap5 examples in test_domonic.py)

IndexError: list index out of range

- You most likely didn't put a underscore on an attribute.
- THIS ALSO APPLIES TO `**{"_data-tags": "x"}`

SyntaxError: invalid syntax

- You are Missing a comma between attributes

SyntaxError: positional argument follows keyword argument

- You have to pass attributes LAST. and strings and objects first. *see docs*

TypeError: unsupported operand type(s) for ** or pow(): 'str' and 'dict'

- You are Missing a comma between attributes. before the `**{ }`

Components**Some notes on components**

A component 'might' look something like this...

```
from domonic.html import *
from domonic.javascript import Math
from domonic.terminal import ifconfig

class My_Component:

    def __init__(self, request, *args, **kwargs):
        self.id = 'launcher'

    def __str__(self):
        return str(
            div(
                div(_id=self.id).html(
                    "CONTENT"
                ),
                script(''
                    ''
                )
            )
        )
```

Now you will need a server as domonic only provides a view.

These example use Sanic. But it could be Flask or any other that can provide routing.

A component could, for example, take a request directly as input and returns html

```
@app.route("/component/<component>")
async def component(request, component):
    try:
        module = __import__(f'app.components.{component}')
        my_class = getattr(module, component.title())
        return response.html( str( my_class(request) ) )
    except Exception as e:
```

(continues on next page)

(continued from previous page)

```
print(e)
return response.html( str( div("COMPONENT NOT FOUND!") ) ) )
```

for this to work the component would need to be in a file called: app/components/my_component.py

Then a given component or template can just return html and render directly into your page using a bit of javascript.

```
// pass an ElementID and an endpoint to redraw that div with the endpoints response
window.redraw = function( _id, endpoint ){
    $.get( endpoint, function( data ) {
        window.console.log(data)
        $( "#"+_id ).html( $(data).html() );
    });
}
```

built-in components

There is a built in components package but its use is discouraged as they may change or be buggy and untested.

You should use domonic to make your own components.

Some that may kick around a while due to being used in examples are listed here...

SpriteCSS

For a working example see... /examples/ken/sf2.py

pass a UID. w, h, path, duration, steps, looping, y_offset

```
animated_monster = SpriteCSS('ken', 70, 80, 'assets/spritesheets/ken.png', 0.8, 4, True,
↪80)
```

1.4.18 utils

The utils libray attempts to be relevant to dom related things.

It's currently mostly string utils but there's some other bits and bobs in there.

If you can think of any missing that would be good then get in touch.

domonic.utils

snippets etc

```
class domonic.utils.Utils
```

```
    static acronym(sentence: str) → str
        [pass a sentence, returns the acronym]
```

Parameters

sentence ([str]) – [typically 3 words]

Returns

[a TLA (three letter acronym)]

Return type`[str]`**static** `case_camel(s: str) → str``case_camel('camel-case') > 'camelCase'`**static** `case_kebab(s: str) → str``kebab('camelCase') # 'camel-case'`**static** `case_snake(s: str) → str``snake('camelCase') # 'camel_case'`**static** `chunk(list: list, size: int) → list``chunk a list into batches`**static** `chunks(iterable, size: int, format=<built-in function iter>)``Iterate over any iterable (list, set, file, stream, strings, whatever), of ANY size`**static** `clean(lst: list) → list``[removes falsy values (False, None, 0 and "") from a list]`**Parameters**`lst ([list])` – [lst to operate on]**Returns**`[a new list with falsy values removed]`**Return type**`[list]`**static** `dictify(arr: list) → dict``[turns a list into a dictionary where the list items are the keys]`**Parameters**`arr ([list])` – [list to change]**Returns**`[a new dict where the list items are now the keys]`**Return type**`[dict]`**static** `digits(text: str = "") → str``[takes a string of mix of digits and letters and returns a string of digits]`**Parameters**`text (str, optional)` – [the text to change]. Defaults to ‘’.**Returns**`[a string of digits]`**Return type**`[str]`**static** `escape(s: str) → str``[escape a string]`**Parameters**`s ([str])` – [the string to escape]**Returns**`[the escaped string]`

Return type`[str]`**static frequency(*data*)**`[check the frequency of elements in the data]`**Parameters****data** (`[type]`) – [the data to check]**Returns**`[a dict of elements and their frequency]`**Return type**`[dict]`**static get_vowels(*string: str*) → list**`[get a list of vowels from the word]`**Parameters****string** (`[str]`) – [the word to check]**Returns**`[a list of vowels]`**Return type**`[list]`**static has_internet(*url: str = 'http://www.google.com/', timeout: int = 5*) → bool**`[check if you have internet connection]`**Parameters**

- **url** (`str`, *optional*) – [the url to check]. Defaults to 'http://www.google.com/'.
- **timeout** (`int`, *optional*) – [the timeout]. Defaults to 5.

Returns`[True if you have internet]`**Return type**`[bool]`**static init_assets(*dir: str = 'assets'*) → None**`[creates an assets directory with nested js/css/img dirs]`**Parameters****dir** (`str`, *optional*) – [default directory name]. Defaults to 'assets'.**static is_linux() → bool**`[check if the system is a linux]`**Returns**`[description]`**Return type**`[bool]`**static is_mac() → bool**`[check if the system is a mac]`**Returns**`[True if the system is a mac]`

Return type`[bool]`**static** `is_nix()` → `bool``[check if the system is a nix based system]`**Returns**`[True if it is a nix based system]`**Return type**`[bool]`**static** `is_windows()` → `bool``[check if the system is a windows]`**Returns**`[True if windows]`**Return type**`[bool]`**static** `merge_dictionaries(a: dict, b: dict)` → `dict``[merges 2 dicts]`**Parameters**

- **a** (`[dict]`) – [dict a]
- **b** (`[dict]`) – [dict b]

Returns`[a new dict]`**Return type**`[dict]`**static** `permutations(word: str)` → `list``[provides all the possible permutations of a given word]`**Parameters****word** (`[str]`) – [the word to get permutations for]**Returns**`[a list of permutations]`**Return type**`[list]`**static** `replace_between(content: str, match: str, replacement: str, start: int = 0, end: int = 0)``[replace some text but only between certain indexes]`**Parameters**

- **content** (`str`) – [the content whos text you will be replacing]
- **match** (`str`) – [the string to find]
- **replacement** (`str`) – [the string to replace it with]
- **start** (`int`, *optional*) – [start index]. Defaults to 0.
- **end** (`int`, *optional*) – [end index]. Defaults to 0.

Returns`[the new string]`

Return type`[str]`**static** `squash(the_list: list) → list``[turns a 2d array into a flat one]`**Parameters****the_list** (`[list]`) – [a 2d array]**Returns**`[a flattened 1d array]`**Return type**`[list]`**static** `to_dictionary(keys: list, values: list) → dict``[take a list of keys and values and returns a dict]`**Parameters**

- **keys** (`[list]`) – [a list of keys]
- **values** (`[list]`) – [a list of value]

Returns`[a dictionary]`**Return type**`[dict]`**static** `truncate(text: str = "", length: int = 0) → str``[truncates a string and appends 3 dots]`**Parameters**

- **text** (`str`, `optional`) – [the text to truncate]. Defaults to ‘’.
- **length** (`int`, `optional`) – [the max length]. Defaults to 0.

Returns`[the truncated string]`**Return type**`[str]`**static** `unescape(s: str) → str``[unescape a string]`**Parameters****s** (`[str]`) – [the string to unescape]**Returns**`[the unescaped string]`**Return type**`[str]`**static** `unique(some_arr: list) → list``[removes duplicates from a list]`**Parameters****some_arr** (`[list]`) – [list containing duplicates]

Returns

[a list containing no duplicates]

Return type

[list]

static untitled(*string: str*) → str

[the opposite of title]

Parameters**str** ([*str*]) – [the string to change]**Returns**

[a string with the first character set to lowercase]

Return type

[str]

static url2file(*url: str*) → str

[gen a safe filename from a url. by replacing '/' for '_' and ':' for '__']

Parameters**url** ([*str*]) – [the url to turn into a filename]**Returns**

[description]

Return type

[str]

1.4.19 webapi

The webapi is the latest package to be added to domonic.

console

```
from domonic.webapi.console import console
console.log("Hello World")
```

encoding

```
from domonic.webapi.encoding import TextEncoder, TextDecoder
encoder = TextEncoder()
```

fetch

```
from domonic.webapi.fetch import fetch
```

URL

url is a wrapper around the urlparse and urlencode functions in python.

```
from domonic.webapi.url import URL

myurl = URL("http://www.google.com/search?q=domonic")
print(myurl.host)
print(myurl.query)
print(myurl.query.q)
print(myurl.query.q.value)
```

for more information see mdn docs... # TODO - link

XPATH

Here's a quick example of using xpath...

```
from domonic.webapi.xpath import XPathEvaluator, XPathResult

somehtml = '''
<div>XPath example</div>
<div>Number of <div>&gt;s: <output></output></div>
'''

page = domonic.parseString(somehtml) # NOTE - probably requires html5lib install
evaluator = XPathEvaluator()
expression = evaluator.createExpression("//div")
result = expression.evaluate(page, XPathResult.ORDERED_NODE_SNAPSHOT_TYPE)
assert result.snapshotLength == 2
```

for more information see mdn docs... # TODO - link

domonic.webapi.console

https://developer.mozilla.org/en-US/docs/Web/API/Console_API

domonic.webapi.encoding

https://developer.mozilla.org/en-US/docs/Web/API/Encoding_API

```
class domonic.webapi.encoding.TextDecoderStream(encoding='utf-8')
```

```
class domonic.webapi.encoding.TextEncoderStream(encoding='utf-8')
```

domonic.webapi.fetch

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

domonic.webapi.url

<https://developer.mozilla.org/en-US/docs/Web/API/URL>

TODO - move the unit tests for this class from javascript to webapi # TODO - untested

class domonic.webapi.url.**URL**(*url: str = "", *args, **kwargs*)

a-tag extends from URL

property hash

” hash Sets or returns the anchor part (#) of a URL

class domonic.webapi.url.**URLSearchParams**(*paramString*)

[utility methods to work with the query string of a URL]

append(*key, value*)

Appends a specified key/value pair as a new search parameter

delete(*key*)

Deletes the given search parameter, and its associated value, from the list of all search parameters.

entries()

Returns an iterator allowing iteration through all key/value pairs contained in this object.

forEach(*func*)

Allows iteration through all values contained in this object via a callback function.

get(*key*)

Returns the first value associated with the given search parameter.

getAll(*key*)

Returns all the values associated with a given search parameter.

has(*key*)

Returns a Boolean indicating if such a given parameter exists.

keys()

Returns an iterator allowing iteration through all keys of the key/value pairs contained in this object.

set(*key, value*)

Sets the value associated with a given search parameter to the given value. If there are several values, the others are deleted.

sort()

Sorts all key/value pairs, if any, by their keys.

toString()

Returns a string containing a query string suitable for use in a URL.

values()

Returns an iterator allowing iteration through all values of the key/value pairs contained in this object.

domonic.webapi.XMLHttpRequest

<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

domonic.webapi.xpath

<https://developer.mozilla.org/en-US/docs/Glossary/XPath>

uses elementpath lib.

TODO - content strings must be TextNodes for it to work.
so will have to iterate and update them. i.e. Treewalker

1.4.20 styles

domonic supports the style attribute.

Styling Elements

Styling gets passed to the style tag on render.

```
mytag = div("hi", _id="test")
mytag.style.backgroundColor = "black"
mytag.style.fontSize = "12px"
print(mytag)
# <div id="test" style="background-color:black;font-size:12px;">hi</div>
```

CSSOM is also stubbed out. See the styles.py class. Feel free to make a PR

stylesheets

```
from domonic.window import *
window.location = "https://www.facebook.com"
window.document.stylesheets
```

domonic.style

TODO - should this be moved to the webapi in a future revision?

class domonic.style.CSSColorProfileRule

The CSSColorProfileRule interface represents an @color-profile rule.

class domonic.style.CSSConditionRule

The CSSConditionRule interface represents a condition rule.

class domonic.style.CSSCounterStyleRule

The CSSCounterStyleRule interface represents a @counter-style rule.

class domonic.style.CSSDocumentRule

The CSSDocumentRule interface represents an @document rule.

class domonic.style.CSSFontFaceRule

The CSSFontFaceRule interface represents a @font-face rule in a CSS style sheet.

class domonic.style.CSSFontFeatureValuesRule

The CSSFontFeatureValuesRule interface represents a @font-feature-values rule.

class domonic.style.CSSGroupingRule

The CSSGroupingRule interface represents a @grouping rule.

class domonic.style.CSSImportRule

The CSSImportRule interface represents an @import rule in a CSS style sheet.

class domonic.style.CSSKeyframeRule

The CSSKeyframeRule interface represents a single @keyframes at-rule.

class domonic.style.CSSKeyframesRule

The CSSKeyframesRule interface represents a @keyframes at-rule.

class domonic.style.CSSMediaRule

The CSSMediaRule interface represents a @media rule in a CSS style sheet.

class domonic.style.CSSNamespaceRule

The CSSNamespaceRule interface represents an @namespace rule in a CSS style sheet.

class domonic.style.CSSPageRule

The CSSPageRule interface represents a @page rule in a CSS style sheet.

class domonic.style.CSSRule

The CSSRule interface represents a single CSS rule. There are several types of rules which inherit properties from CSSRule.

CSSFontFeatureValuesRule CSSViewportRule

property cssText

16pt }” or “@import ‘url’”. To access or modify parts of the rule (e.g. the value of “font-size” in the example) use the properties on the specialized interface for the rule’s type.

Type

Represents the textual representation of the rule, e.g. “h1,h2 { font-size

class domonic.style.CSSRuleList(*iterable=()*, /)

A CSSRuleList represents an ordered collection of read-only CSSRule objects. While the CSSRuleList object is read-only, and cannot be directly modified, it is considered a live object, as the content can change over time.

item(*index: int*) → *CSSRule*

Gets a single CSSRule.

property length: int

Returns an integer representing the number of CSSRule objects in the collection.

class domonic.style.CSSStyleDeclaration(*parentNode=None, *args, **kwargs*)

The CSSStyleDeclaration interface represents an object that is a CSS declaration block, and exposes style information and various style-related methods and properties.

A CSSStyleDeclaration object can be exposed using three different APIs:

Via HTMLElement.style, which deals with the inline styles of a single element (e.g., <div style=“...”>). Via the CSSStyleSheet API. For example, document.styleSheets[0].cssRules[0].style returns a CSSStyleDeclaration object on the first CSS rule in the document’s first stylesheet. Via Window.getComputedStyle(), which exposes the CSSStyleDeclaration object as a read-only interface.

getPropertyCSSValue()

Only supported via `getComputedStyle` in Firefox. Returns the property value as a `CSSPrimitiveValue` or null for shorthand properties.

getPropertyPriority()

Returns the optional priority, “important”.

getPropertyValue(*propertyName: str*) → *str*

Returns the value of the property with the specified name.

item(*index: int*) → *str*

Returns a CSS property name by its index, or the empty string if the index is out-of-bounds. An alternative to accessing `nodeList[i]` (which instead returns undefined when `i` is out-of-bounds). This is mostly useful for non-JavaScript DOM implementations.

property length

The number of properties. See the `item()` method below.

removeProperty()

Removes a property from the CSS declaration block.

class domonic.style.CSSStyleRule

The `CSSStyleRule` interface represents a single style rule in a CSS style sheet.

property cssText

Returns the textual representation of the rule.

class domonic.style.CSSStyleSheet

Creates a new `CSSStyleSheet` object.

addRule(*selectorText: str, style: str, index: int*)

Adds a new rule to the stylesheet given the selector to which the style applies and the style block to apply to the matching elements. This differs from `insertRule()`, which takes the textual representation of the entire rule as a single string.

deleteRule(*index: int*)

Deletes the rule at the specified index into the stylesheet’s rule list.

insertRule(*rule: str, index: Optional[int] = None*)

Inserts a new rule at the specified position in the stylesheet, given the textual representation of the rule.

removeRule(*index: int*)

Functionally identical to `deleteRule()`; removes the rule at the specified index from the stylesheet’s rule list.

replace(*text: str*)

Asynchronously replaces the content of the stylesheet and returns a Promise that resolves with the updated `CSSStyleSheet`.

replaceSync(*text: str*)

Synchronously replaces the content of the stylesheet.

class domonic.style.CSSSupportsRule

The `CSSSupportsRule` interface represents a `@supports` at-rule.

class domonic.style.MediaList(*iterable=(, /)*)

The `MediaList` interface represents a list of media. It is used in the `@media` at-rule.

appendMedium(*newMedium: str*) → None

Appends a new medium to the end of the list.

deleteMedium(*oldMedium: str*) → None

Removes medium in the media list. If the medium is not found nothing happens.

item(*index: int*) → str

Returns the media at the given index in the MediaList.

property length: int

Returns the number of media in the list.

property mediaText: str

Returns a string containing the text of the media query.

class domonic.style.Style(*parent_node=None*)

[js syntax styles] # TODO - just add normal float? # TODO - consider camel case for hyphen params? # TODO
- not json serialisable due to the decorators.

property alignContent

Sets or returns the alignment between the lines inside a flexible container when the items do not use all available space

property alignItems

Sets or returns the alignment for items inside a flexible container

property alignSelf

Sets or returns the alignment for selected items inside a flexible container

property animation

shorthand property for all the animation properties below, except the animationPlayState property

property animationDelay

Sets or returns when the animation will start

property animationDirection

Sets or returns whether or not the animation should play in reverse on alternate cycles

property animationDuration

Sets or returns how many seconds or milliseconds an animation takes to complete one cycle

property animationFillMode

Sets or returns what values are applied by the animation outside the time it is executing

property animationIterationCount

Sets or returns the number of times an animation should be played

property animationName

Sets or returns a name for the @keyframes animation

property animationPlayState

Sets or returns whether the animation is running or paused

property animationTimingFunction

Sets or returns the speed curve of the animation

property backfaceVisibility

Sets or returns whether or not an element should be visible when not facing the screen

property background

Sets or returns all the background properties in one declaration

property backgroundAttachment

Sets or returns whether a background-image is fixed or scrolls with the page

property backgroundClip

Sets or returns the painting area of the background

property backgroundColor

Sets or returns the background-color of an element

property backgroundImage

Sets or returns the background-image for an element

property backgroundOrigin

Sets or returns the positioning area of the background images

property backgroundPosition

Sets or returns the starting position of a background-image

property backgroundRepeat

Sets or returns how to repeat (tile) a background-image

property backgroundSize

Sets or returns the size of the background image

property border

Sets or returns borderWidth, borderStyle, and borderColor in one declaration

property borderBottom

Sets or returns all the borderBottom properties in one declaration

property borderBottomColor

Sets or returns the color of the bottom border 1

property borderBottomLeftRadius

Sets or returns the shape of the border of the bottom-left corner

property borderBottomRightRadius

Sets or returns the shape of the border of the bottom-right corner

property borderBottomStyle

Sets or returns the style of the bottom border

property borderBottomWidth

Sets or returns the width of the bottom border

property borderCollapse

Sets or returns whether the table border should be collapsed into a single border, or not

property borderColor

Sets or returns the color of an element's border (can have up to four values)

property borderImage

horthand property for setting or returning all the borderImage properties

property `borderImageOutset`

Sets or returns the amount by which the border image area extends beyond the border box

property `borderImageRepeat`

Sets or returns whether the image-border should be repeated, rounded or stretched

property `borderImageSlice`

Sets or returns the inward offsets of the image-border

property `borderImageSource`

Sets or returns the image to be used as a border

property `borderImageWidth`

Sets or returns the widths of the image-border

property `borderLeft`

Sets or returns all the `borderLeft` properties in one declaration

property `borderLeftColor`

Sets or returns the color of the left border

property `borderLeftStyle`

Sets or returns the style of the left border

property `borderLeftWidth`

Sets or returns the width of the left border

property `borderRadius`

A shorthand property for setting or returning all the four `borderRadius` properties

property `borderRight`

Sets or returns all the `borderRight` properties in one declaration

property `borderRightColor`

Sets or returns the color of the right border

property `borderRightStyle`

Sets or returns the style of the right border

property `borderRightWidth`

Sets or returns the width of the right border

property `borderSpacing`

Sets or returns the space between cells in a table

property `borderStyle`

Sets or returns the style of an element's border (can have up to four values)

property `borderTop`

Sets or returns all the `borderTop` properties in one declaration

property `borderTopColor`

Sets or returns the color of the top border

property `borderTopLeftRadius`

Sets or returns the shape of the border of the top-left corner

property borderTopRightRadius

Sets or returns the shape of the border of the top-right corner

property borderTopStyle

Sets or returns the style of the top border

property borderTopWidth

Sets or returns the width of the top border

property borderWidth

Sets or returns the width of an element's border (can have up to four values)

property bottom

Sets or returns the bottom position of a positioned element

property boxDecorationBreak

Sets or returns the behaviour of the background and border of an element at page-break, or, for in-line elements, at line-break.

property boxShadow

ttaches one or more drop-shadows to the box

property boxSizing

llows you to define certain elements to fit an area in a certain way

property captionSide

Sets or returns the position of the table caption

property clear

Sets or returns the position of the element relative to floating objects

property clip

Sets or returns which part of a positioned element is visible

property color

Sets or returns the color of the text

property columnCount

Sets or returns the number of columns an element should be divided into

property columnFill

Sets or returns how to fill columns

property columnGap

Sets or returns the gap between the columns

property columnRule

shorthand property for setting or returning all the columnRule properties

property columnRuleColor

Sets or returns the color of the rule between columns

property columnRuleStyle

Sets or returns the style of the rule between columns

property columnRuleWidth

Sets or returns the width of the rule between columns

property columnSpan

Sets or returns how many columns an element should span across

property columnWidth

Sets or returns the width of the columns

property columns

horthand property for setting or returning columnWidth and columnCount

property content

after pseudo-elements, to insert generated content

Type

d with the

Type

before and

property counterIncrement

Increments one or more counters

property counterReset

Creates or resets one or more counters

property cssFloat

Sets or returns the horizontal alignment of an element

property cursor

Sets or returns the type of cursor to display for the mouse pointer

property direction

Sets or returns the text direction

property display

Sets or returns an element's display type

property emptyCells

Sets or returns whether to show the border and background of empty cells, or not

property filter

Sets or returns image filters (visual effects, like blur and saturation)

property flex

Sets or returns the length of the item, relative to the rest

property flexBasis

Sets or returns the initial length of a flexible item

property flexDirection

Sets or returns the direction of the flexible items

property flexFlow

A shorthand property for the flexDirection and the flexWrap properties

property flexGrow

Sets or returns how much the item will grow relative to the rest

property flexShrink

Sets or returns how the item will shrink relative to the rest

property flexWrap

Sets or returns whether the flexible items should wrap or not

property font

Sets or returns fontStyle, fontVariant, fontWeight, fontSize, lineHeight, and fontFamily in one declaration

property fontFamily

Sets or returns the font family for text

property fontSize

Sets or returns the font size of the text

property fontSizeAdjust

reserves the readability of text when font fallback occurs

property fontStretch

selects a normal, condensed, or expanded face from a font family

property fontStyle

Sets or returns whether the style of the font is normal, italic or oblique

property fontVariant

Sets or returns whether the font should be displayed in small capital letters

property fontWeight

Sets or returns the boldness of the font

property hangingPunctuation

Specifies whether a punctuation character may be placed outside the line box

property height

Sets or returns the height of an element

property hyphens

Sets how to split words to improve the layout of paragraphs

property icon

Provides the author the ability to style an element with an iconic equivalent

property imageOrientation

Specifies a rotation in the right or clockwise direction that a user agent applies to an image

property isolation

Defines whether an element must create a new stacking context

property justifyContent

Sets or returns the alignment between the items inside a flexible container when the items do not use all available space.

property left

Sets or returns the left position of a positioned element

property letterSpacing

Sets or returns the space between characters in a text

property lineHeight

Sets or returns the distance between lines in a text

property listStyle

Sets or returns listStyleImage, listStylePosition, and listStyleType in one declaration

property listStyleImage

Sets or returns an image as the list-item marker

property listStylePosition

Sets or returns the position of the list-item marker

property listStyleType

Sets or returns the list-item marker type

property margin

Sets or returns the margins of an element (can have up to four values)

property marginBottom

Sets or returns the bottom margin of an element

property marginLeft

Sets or returns the left margin of an element

property marginRight

Sets or returns the right margin of an element

property marginTop

Sets or returns the top margin of an element

property maxHeight

Sets or returns the maximum height of an element

property maxWidth

Sets or returns the maximum width of an element

property minHeight

Sets or returns the minimum height of an element

property minWidth

Sets or returns the minimum width of an element

property navDown

Sets or returns where to navigate when using the arrow-down navigation key

property navIndex

Sets or returns the tabbing order for an element

property navLeft

Sets or returns where to navigate when using the arrow-left navigation key

property navRight

Sets or returns where to navigate when using the arrow-right navigation key

property navUp

Sets or returns where to navigate when using the arrow-up navigation key

property objectFit

pecifies how the contents of a replaced element should be fitted to the box established by its used height and width

property objectPosition

ecifies the alignment of the replaced element inside its box

property opacity

Sets or returns the opacity level for an element

property order

Sets or returns the order of the flexible item, relative to the rest

property orphans

Sets or returns the minimum number of lines for an element that must be left at the bottom of a page when a page break occurs inside an element

property outline

Sets or returns all the outline properties in one declaration

property outlineColor

Sets or returns the color of the outline around a element

property outlineOffset

ffsets an outline, and draws it beyond the border edge

property outlineStyle

Sets or returns the style of the outline around an element

property outlineWidth

Sets or returns the width of the outline around an element

property overflow

Sets or returns what to do with content that renders outside the element box

property overflowX

pecifies what to do with the left/right edges of the content, if it overflows the element's content area

property overflowY

pecifies what to do with the top/bottom edges of the content, if it overflows the element's content area

property padding

Sets or returns the padding of an element (can have up to four values)

property paddingBottom

Sets or returns the bottom padding of an element

property paddingLeft

Sets or returns the left padding of an element

property paddingRight

Sets or returns the right padding of an element

property paddingTop

Sets or returns the top padding of an element

property pageBreakAfter

Sets or returns the page-break behavior after an element

property pageBreakBefore

Sets or returns the page-break behavior before an element

property pageBreakInside

Sets or returns the page-break behavior inside an element

property perspective

Sets or returns the perspective on how 3D elements are viewed

property perspectiveOrigin

Sets or returns the bottom position of 3D elements

property position

Sets or returns the type of positioning method used for an element (static, relative, absolute or fixed)

property quotes

Sets or returns the type of quotation marks for embedded quotations

property resize

Sets or returns whether or not an element is resizable by the user

property right

Sets or returns the right position of a positioned element

property tabSize

Sets or returns the length of the tab-character

property tableLayout

Sets or returns the way to lay out table cells, rows, and columns

property textAlign

Sets or returns the horizontal alignment of text

property textAlignLast

Sets or returns how the last line of a block or a line right before a forced line break is aligned when text-align is justify

property textDecoration

Sets or returns the decoration of a text

property textDecorationColor

Sets or returns the color of the text-decoration

property text-decorationLine

Sets or returns the type of line in a text-decoration

property text-decorationStyle

Sets or returns the style of the line in a text decoration

property textIndent

Sets or returns the indentation of the first line of text

property textJustify

Sets or returns the justification method used when text-align is justify

property textOverflow

Sets or returns what should happen when text overflows the containing element

property textShadow

Sets or returns the shadow effect of a text

property textTransform

Sets or returns the capitalization of a text

property top

Sets or returns the top position of a positioned element

property transform

Applies a 2D or 3D transformation to an element

property transformOrigin

Sets or returns the position of transformed elements

property transformStyle

Sets or returns how nested elements are rendered in 3D space

property transition

Shorthand property for setting or returning the four transition properties

property transitionDelay

Sets or returns when the transition effect will start

property transitionDuration

Sets or returns how many seconds or milliseconds a transition effect takes to complete

property transitionProperty

Sets or returns the CSS property that the transition effect is for

property transitionTimingFunction

Sets or returns the speed curve of the transition effect

property unicodeBidi

Sets or returns whether the text should be overridden to support multiple languages in the same document

property userSelect

Sets or returns whether the text of an element can be selected or not

property verticalAlign

Sets or returns the vertical alignment of the content in an element

property visibility

Sets or returns whether an element should be visible

property whiteSpace

Sets or returns how to handle tabs, line breaks and whitespace in a text

property widows

Sets or returns the minimum number of lines for an element that must be visible at the top of a page

property width

Sets or returns the width of an element

property wordBreak

Sets or returns line breaking rules for non-CJK scripts

property wordSpacing

Sets or returns the spacing between words in a text

property wordWrap

Allows long, unbreakable words to be broken and wrap to the next line

property zIndex

Sets or returns the stack order of a positioned element

class domonic.style.StyleSheet

An object implementing the StyleSheet interface represents a single style sheet. CSS style sheets will further implement the more specialized CSSStyleSheet interface.

class domonic.style.StyleSheetList(iterable=(),/)

An instance of this object can be returned by Document.styleSheets. it can be iterated over in a standard for loop over its indices, or converted to an Array.

item(index)

Returns the CSSStyleSheet object at the index passed in, or null if no item exists for that index.

property length

Returns the number of CSSStyleSheet objects in the collection.

1.4.21 servers

Generating static html files with domonic is fun.

Python has a built-in server allowing you to view your generated files:

Running a python server to view static pages

```
cd Desktop/yourproject
python3 -m http.server 8080
```

now go to localhost:8080 and view your website.

Serving dynamic content

For dynamic content you will need a better webserver.

Domonic does not come with a webserver but there are plenty of great ones in the python community to choose from.

Below are some examples of how to use domonic with some popular webrowsers

WARNING: When generating dynamic content make sure to escape any user generated content to avoid [XSS attacks](#)..

Using domonic with CherryPy

```
python3 -m venv venv
. venv/bin/activate
pip install cherrypy
pip install domonic
```

now create a file called app.py

```
import cherrypy
from domonic.html import *

class HelloWorld:

    @cherrypy.expose
    def index(self):
        return str(
            html(
                head(),
                body(
                    div(span("Hello, World!"))
                )
            )
        )

cherrypy.quickstart(HelloWorld())
```

to run it do this:

```
python app.py
```

and visit localhost:8080 in your browser

[Find out more about Cherrypy here...](#)

Using domonic with Pyramid

```
python3 -m venv venv
. venv/bin/activate
pip install pyramid
pip install domonic
```

now create a file called app.py

```
from wsgiref.simple_server import make_server
from pyramid.config import Configurator
from pyramid.response import Response
from domonic.html import *

def hello_world(request):
    return Response(str(
        html(
            head(),
            body(
                div(span("Hello, World!"))
            )
        )
    ))

if __name__ == '__main__':
```

(continues on next page)

(continued from previous page)

```

with Configurator() as config:
    config.add_route('hello', '/')
    config.add_view(hello_world, route_name='hello')
    app = config.make_wsgi_app()
    server = make_server('0.0.0.0', 8080, app)
    server.serve_forever()

```

to run it do this:

```
python app.py
```

and visit localhost:8080 in your browser

[Find out more about Pyramid here...](#)

Using domonic with Bottle

```

python3 -m venv venv
. venv/bin/activate
pip install bottle
pip install domonic

```

now create a file called app.py

```

from bottle import route, run
from domonic.html import *

@route('/hello/<name>')
def index(name):
    return str(
        html(
            head(),
            body(
                div(span(f"Hello, {name}!"))
            )
        )
    )

run(host='localhost', port=8080)

```

to run it do this:

```
python app.py
```

and visit <http://localhost:8080/hello/yourname> in your browser

[Find out more about Bottle here...](#)

Using domonic with Sanic

A lot of the examples in the repo use Sanic. It's like Flask and is async

```
python3 -m venv venv
. venv/bin/activate
pip install sanic
pip install domonic
```

now create a file called app.py

```
from sanic import Sanic
from sanic import response
from domonic.html import *

app = Sanic("My Hello, world app")

@app.route('/')
async def test(request):
    return response.html(str(
        html(
            head(),
            body(
                div(span("Hello World!"))
            )
        ))
    )

if __name__ == '__main__':
    app.run()
```

to run it do this:

```
python app.py
```

and visit <http://localhost:8000> in your browser

[Find out more about Sanic here...](#)

Using domonic with Flask

Flask comes with Jinja already but it's still possible...

```
python3 -m venv venv
. venv/bin/activate
pip install flask
pip install domonic
```

now create a file called app.py

```
from flask import Flask
from domonic.html import *

app = Flask(__name__)
```

(continues on next page)

(continued from previous page)

```
@app.route("/")
def hello():
    return str(
        html(
            head(),
            body(
                div(span("Hello World!"))
            )
        )
    )

if __name__ == '__main__':
    app.run()
```

to run it do this:

```
python app.py
```

and visit <http://localhost:5000> in your browser

[Find out more about Flask here...](#)

Using domonic with FastAPI

```
python3 -m venv venv
. venv/bin/activate
pip install fastapi
pip install uvicorn
pip install domonic
```

now create a file called app.py

```
from fastapi import FastAPI
from fastapi.responses import HTMLResponse
from domonic.html import *

app = FastAPI()

@app.get("/", response_class=HTMLResponse)
def read_root():
    return str(
        html(
            head(),
            body(
                div(span("Hello World!"))
            )
        )
    )
```

to run it do this:

```
uvicorn app:app --reload
```

and visit <http://localhost:8000> in your browser

Find out more about FastAPI [here](#)...

Using domonic with Werkzeug

```
python3 -m venv venv
. venv/bin/activate
pip install werkzeug
pip install domonic
```

now create a file called app.py

```
from werkzeug.wrappers import Request, Response
from domonic.html import *

@Request.application
def application(request):
    return Response(str(
        html(
            head(),
            body(
                div(span("Hello World!"))
            )
        )), mimetype='text/html')

if __name__ == '__main__':
    from werkzeug.serving import run_simple
    run_simple('localhost', 4000, application)
```

to run it do this:

```
python app.py
```

and visit <http://localhost:4000/> in your browser

Find out more about Werkzeug [here](#)...

Using domonic with Starlette

```
python3 -m venv venv
. venv/bin/activate
pip install starlette
pip install uvicorn
pip install domonic
```

now create a file called app.py

```
from starlette.applications import Starlette
from starlette.responses import HTMLResponse
from starlette.routing import Route
from domonic.html import *

async def homepage(request):
```

(continues on next page)

(continued from previous page)

```

    return HTMLResponse(str(
        html(
            head(),
            body(
                div(span("Hello World!"))
            )
        ))
    )

routes = [
    Route("/", endpoint=homepage)
]

app = Starlette(debug=True, routes=routes)

```

to run it do this:

```
uvicorn app:app --reload
```

and visit <http://localhost:8000> in your browser

[Find out more about Starlette here...](#)

Using domonic with Tornado

```

python3 -m venv venv
. venv/bin/activate
pip install tornado
pip install domonic

```

now create a file called app.py

```

import tornado.ioloop
import tornado.web
from domonic.html import *

class MainHandler(tornado.web.RequestHandler):
    def get(self):
        self.write(str(
            html(
                head(),
                body(
                    div(span("Hello World!"))
                )
            )))

def make_app():
    return tornado.web.Application([
        (r"/", MainHandler),
    ])

if __name__ == "__main__":

```

(continues on next page)

(continued from previous page)

```
app = make_app()
app.listen(8888)
tornado.ioloop.IOLoop.current().start()
```

to run it do this:

```
python app.py
```

and visit <http://localhost:8888/> in your browser

[Find out more about Tornado here...](#)

Using domonic with Django

Django already has some kind of Jinja but more restrictive.

```
python3 -m venv venv
. venv/bin/activate
pip install django
pip install domonic
django-admin startproject mysite
```

now cd into mysite and edit urls.py

```
from django.contrib import admin
from django.urls import path
from django.http import HttpResponse
from domonic import div, span

def index(request):
    mywebpage = str(
        div(span("Hello World!"))
    )
    return HttpResponse(mywebpage)

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index, name='index'),
]
```

to run it do this from within mysite folder:

```
python manage.py runserver
```

and visit <http://localhost:8000/> in your browser

Note: Django didn't allow import * so there's a conflict somewhere. I resolved by importing what I needed.

[Find out more about Django here...](#)

Using domonic with aiohttp

```
python3 -m venv venv
. venv/bin/activate
pip install aiohttp
pip install domonic
```

now create a file called app.py

```
from domonic.html import *
from aiohttp import web

async def handle(request):
    name = request.match_info.get('name', "Anonymous")
    page = html(head(), body(div(span("Hello, World!"))))
    return web.Response(text=str(page), content_type='text/html')

app = web.Application()
app.add_routes([web.get('/', handle),
                web.get('/{name}', handle)])

if __name__ == '__main__':
    web.run_app(app)
```

to run it do this:

```
python app.py
```

and visit <http://localhost:8080/> in your browser

[Find out more about aiohttp here...](#)

and if that wasn't enough webservers to try out this isn't even a [more complete list!!!](#).

SPA's

Now you have a framework you can use some simple javascript to call on endpoints to redraw parts of the dom.

```
function redraw(_id, endpoint) {
    fetch(endpoint)
        .then(function(response){return response.text();})
        .then(function(data){
            document.getElementById(_id).innerHTML = data;
        })
}
```

Checkout the 'templates and components' section to see how you can take your templating skills to the next level.

Another alternative to running a webserver is running a serverless function. See below for more details.

Using domonic with AWS lambda

The original version of domonic was tags only and written to be used in an aws lambda function.

the original POC code for that is [here in the archive](#).

You can just create a file called tags.py alongside your lambda with the AWS GUI and paste in the tags then import and use them.

Alternatively to upload entire packages people tend to drop their lambda_function.py into the /site-packages folder of their virtualenv.

Then zip and upload the whole thing.

[Find out more about AWS Lambda here...](#)

or even try an ASGI adapter on your lambda with magnum!... [<https://mangum.io/>](https://mangum.io/)`_.

Using domonic with Google Cloud Functions

Google have 'cloud functions'.

Dealing with Package dependencies is here in their documentation.

<https://cloud.google.com/functions/docs/writing/specifying-dependencies-python>

[Find out more about Google Cloud Functions here...](#)

1.4.22 autodocs

domonic

A library for creating html

domonic.html

Generate HTML using python.

```
exception domonic.html.TemplateError(error, message='TemplateError: ')
```

```
class domonic.html.a(*args, **kwargs)
```

```
class domonic.html.abbrev(*args, **kwargs)
```

```
class domonic.html.address(*args, **kwargs)
```

```
class domonic.html.applet(*args, **kwargs)
```

```
class domonic.html.area(*args, **kwargs)
```

```
class domonic.html.article(*args, **kwargs)
```

```
class domonic.html.aside(*args, **kwargs)
```

```
class domonic.html.audio(*args, autoplay: Optional[bool] = None, controls=None, loop=None, muted=None, preload=None, src=None, **kwargs)
```

```
class domonic.html.b(*args, **kwargs)
```

```

class domonic.html.base(*args, href=None, target=None, **kwargs)
class domonic.html.basefont(*args, **kwargs)
class domonic.html.bdi(*args, **kwargs)
class domonic.html.bdo(*args, **kwargs)
class domonic.html.blockquote(*args, **kwargs)
class domonic.html.body(*args, aLink=None, background=None, bgColor=None, link=None, onload=None,
                        onunload=None, text=None, vLink=None, **kwargs)
class domonic.html.br(*args, **kwargs)
class domonic.html.button(*args, disabled: Optional[bool] = None, form=None, formaction: Optional[str] =
                        None, formenctype=None, formmethod=None, formnovalidate=None,
                        formtarget=None, name=None, type=None, value=None, **kwargs)
class domonic.html.canvas(*args, width: Optional[int] = None, height: Optional[int] = None, **kwargs)
class domonic.html.caption(*args, **kwargs)
class domonic.html.center(*args, **kwargs)
class domonic.html.cite(*args, **kwargs)
class domonic.html.closed_tag(*args, **kwargs)
class domonic.html.code(*args, **kwargs)
class domonic.html.col(*args, **kwargs)
class domonic.html.colgroup(*args, **kwargs)
class domonic.html.command(*args, **kwargs)
class domonic.html.comment(data)
domonic.html.create_element(name='custom_tag', *args, **kwargs)
    A method for creating custom tags
    tag name needs to be set due to custom tags with hyphens can't be classnames. i.e. hypenated tags <some-
    custom-tag></some-custom-tag>
class domonic.html.data(*args, **kwargs)
class domonic.html.datalist(*args, **kwargs)
class domonic.html.dd(*args, **kwargs)
class domonic.html.details(*args, **kwargs)
class domonic.html.dfn(*args, **kwargs)
class domonic.html.dialog(*args, open=None, **kwargs)
class domonic.html.div(*args, **kwargs)
class domonic.html.dl(*args, **kwargs)

```

```
class domonic.html.doctype(name: str = 'html', publicId: str = "", systemId: str = "")
class domonic.html.dt(*args, **kwargs)
class domonic.html.em(*args, **kwargs)
class domonic.html.embed(*args, **kwargs)
class domonic.html.fieldset(*args, **kwargs)
class domonic.html.figcaption(*args, **kwargs)
class domonic.html.figure(*args, **kwargs)
class domonic.html.font(*args, **kwargs)
class domonic.html.footer(*args, **kwargs)
class domonic.html.form(*args, **kwargs)
class domonic.html.h1(*args, **kwargs)
class domonic.html.h2(*args, **kwargs)
class domonic.html.h3(*args, **kwargs)
class domonic.html.h4(*args, **kwargs)
class domonic.html.h5(*args, **kwargs)
class domonic.html.h6(*args, **kwargs)
class domonic.html.head(*args, **kwargs)
class domonic.html.header(*args, **kwargs)
class domonic.html.hgroup(*args, **kwargs)
class domonic.html.hr(*args, **kwargs)
class domonic.html.html(*args, **kwargs)
class domonic.html.i(*args, **kwargs)
class domonic.html.iframe(*args, **kwargs)
class domonic.html.img(*args, alt=None, src=None, crossorigin=None, height=None, ismap=None,
                      longdesc=None, sizes=None, srcset=None, usemap=None, width=None, **kwargs)
class domonic.html.input(*args, accept=None, alt=None, autocomplete=None, autofocus=None,
                        checked=None, dirname=None, disabled=None, form=None, formaction=None,
                        formenctype=None, formmethod=None, formnovalidate=None, formtarget=None,
                        height=None, _list=None, _max=None, maxlength=None, _min=None,
                        multiple=None, name=None, pattern=None, placeholder=None, readonly=None,
                        required=None, size=None, src=None, step=None, type=None, value=None,
                        width=None, **kwargs)
class domonic.html.ins(*args, **kwargs)
class domonic.html.isindex(*args, **kwargs)
```



```

class domonic.html.kbd(*args, **kwargs)
class domonic.html.keygen(*args, **kwargs)
class domonic.html.label(*args, **kwargs)
class domonic.html.legend(*args, **kwargs)
class domonic.html.li(*args, **kwargs)
class domonic.html.link(*args, **kwargs)
class domonic.html.listing(*args, **kwargs)
class domonic.html.main(*args, **kwargs)
class domonic.html.mark(*args, **kwargs)
class domonic.html.menu(*args, **kwargs)
class domonic.html.menuitem(*args, **kwargs)
class domonic.html.meta(*args, charset=None, content=None, http_equiv=None, name=None, **kwargs)
class domonic.html.meter(*args, value=None, _min=None, _max=None, low=None, high=None,
                        optimum=None, **kwargs)
class domonic.html.nav(*args, **kwargs)
class domonic.html.noscript(*args, **kwargs)
class domonic.html.ol(*args, **kwargs)
class domonic.html.optgroup(*args, **kwargs)
class domonic.html.option(*args, disabled=None, label=None, selected=None, value=None, **kwargs)
class domonic.html.output(*args, **kwargs)
class domonic.html.p(*args, **kwargs)
class domonic.html.param(*args, **kwargs)
class domonic.html.picture(*args, **kwargs)
class domonic.html.plaintext(*args, **kwargs)
class domonic.html.portal(*args, **kwargs)
class domonic.html.pre(*args, **kwargs)
class domonic.html.progress(*args, **kwargs)
class domonic.html.q(*args, **kwargs)
domonic.html.render(inp, outp="", to=None)
    write the input to string or to a file.

```

Parameters

- **inp** (*obj*) – A domonic tag. For example `div()`

- **outp** (*str*) – An optional output filename
- **to** (*str*) – An optional output type. if ‘pym1’ is specified then pym1 is returned instead of html.

Returns

A HTML rendered string

Return type

str

```
class domonic.html.rp(*args, **kwargs)
class domonic.html.rt(*args, **kwargs)
class domonic.html.ruby(*args, **kwargs)
class domonic.html.s(*args, **kwargs)
class domonic.html.samp(*args, **kwargs)
class domonic.html.script(*args, **kwargs)
class domonic.html.section(*args, **kwargs)
class domonic.html.select(*args, autofocus: Optional[bool] = None, disabled: Optional[bool] = None,
                           multiple: Optional[bool] = None, name: Optional[str] = None, required:
                           Optional[bool] = None, size: Optional[int] = None, **kwargs)
class domonic.html.small(*args, **kwargs)
class domonic.html.source(*args, **kwargs)
class domonic.html.span(*args, **kwargs)
class domonic.html.strike(*args, **kwargs)
class domonic.html.strong(*args, **kwargs)
class domonic.html.style(*args, **kwargs)
class domonic.html.sub(*args, **kwargs)
class domonic.html.submit(*args, **kwargs)
class domonic.html.summary(*args, **kwargs)
class domonic.html.sup(*args, **kwargs)
class domonic.html.table(*args, align: Optional[str] = None, bgcolor=None, border=None,
                           cellpadding=None, cellspacing=None, frame=None, rules=None, summary=None,
                           width=None, **kwargs)
class domonic.html.tbody(*args, **kwargs)
class domonic.html.td(*args, **kwargs)
class domonic.html.template(*args, **kwargs)
```

```

class domonic.html.textarea(*args, autofocus=None, cols=None, disabled=None, form=None,
                             maxlength=None, name=None, placeholder=None, readonly=None,
                             required=None, rows=None, wrap=None, **kwargs)

class domonic.html.tfoot(*args, **kwargs)

class domonic.html.th(*args, **kwargs)

class domonic.html.thead(*args, **kwargs)

class domonic.html.title(*args, **kwargs)

class domonic.html.tr(*args, **kwargs)

class domonic.html.track(*args, **kwargs)

class domonic.html.u(*args, **kwargs)

class domonic.html.ul(*args, **kwargs)

class domonic.html.var(*args, **kwargs)

class domonic.html.video(*args, autoplay=None, controls=None, height=None, loop=None, muted=None,
                          poster=None, preload=None, src=None, width=None, **kwargs)

class domonic.html.wbr(*args, **kwargs)

class domonic.html.xmp(*args, **kwargs)

```

domonic.dom

The DOM represents a document with a logical tree. https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

```
class domonic.dom.Attr(name: str, value="", *args, **kwargs)
```

```
    getNamedItem(name: str)
```

Returns a specified attribute node from a NamedNodeMap

```
    removeNamedItem(name: str) → bool
```

Removes a specified attribute node

```
    setNamedItem(name: str, value) → bool
```

Sets the specified attribute node (by name)

```
class domonic.dom.CDATASection(data)
```

```
class domonic.dom.CharacterData(*args, **kwargs)
```

The CharacterData abstract interface represents a Node object that contains characters. This is an abstract interface, meaning there aren't any objects of type CharacterData: it is implemented by other interfaces like Text, Comment, or ProcessingInstruction, which aren't abstract.

```
    after(newChild)
```

Inserts a newChild node immediately after this ChildNode.

```
    appendData(data)
```

Appends the given DOMString to the CharacterData.data string; when this method returns, data contains the concatenated DOMString.

before(*newChild*)

Inserts a newChild node immediately before this ChildNode.

deleteData(*offset: int, count: int*)

Removes the specified amount of characters, starting at the specified offset, from the CharacterData.data string; when this method returns, data contains the shortened DOMString.

insertData(*offset: int, data*)

Inserts the specified characters, at the specified offset, in the CharacterData.data string; when this method returns, data contains the modified DOMString.

property nextElementSibling

Returns the next element at the same node tree level

property previousElementSibling

returns the Element immediately prior to the specified one in its parent's children list, or None if the specified element is the first one in the list.

remove()

Removes this ChildNode from the children list of its parent.

replaceData(*offset: int, count: int, data*)

Replaces the specified amount of characters, starting at the specified offset, with the specified DOMString; when this method returns, data contains the modified DOMString.

replaceWith(*newChild*)

Replaces this ChildNode with a new one.

substringData(*offset: int, length: int*)

Returns a DOMString containing the part of CharacterData.data of the specified length and starting at the specified offset.

class domonic.dom.**ChildNode**(*args, **kwargs)

not tested yet

after(*newChild*)

Inserts a newChild node immediately after this ChildNode.

before(*newChild*)

Inserts a newChild node immediately before this ChildNode.

remove()

Removes this ChildNode from the children list of its parent.

replaceWith(*newChild*)

Replaces this ChildNode with a new one.

class domonic.dom.**Comment**(*data*)

class domonic.dom.**DOMConfig**

DOMConfig: Not to be confused with the obsolete DOMConfiguration.

This class is used to set some global rules for our dom.

GLOBAL_AUTOESCAPE - If this is set to True, then all text nodes will be automatically escaped.
HTML5_MODE - doesn't render close tags on certain elements HTMX_ENABLED - includes htmx attributes into domonic for quicker notation

exception domonic.dom.DOMException(*code*, *message*: *Optional[str] = None*)

The DOMException interface represents an anormal event related to the DOM.

class domonic.dom.DOMPoint(*x*: *float*, *y*: *float*, *z*: *float* = 0, *w*: *float* = 1)

The DOMPoint interface represents a point specified by x and y coordinates.

class domonic.dom.DOMPointReadOnly(*x*: *float*, *y*: *float*, *z*: *float* = 0, *w*: *float* = 1)

The DOMPointReadOnly interface represents a point specified by x and y coordinates.

class domonic.dom.DOMQuad(*p1*, *p2*, *p3*, *p4*)

The DOMQuad interface represents a quadrilateral on the plane with its four corners represented as Cartesian coordinates.

class domonic.dom.DOMStringMap(**args*, ***kwargs*)

TODO - not tested yet TODO - make this like a dict

delete(*name*: *str*) → *bool*

Deletes the item with the specified name

get(*name*: *str*)

Returns the value of the item with the specified name

set(*name*: *str*, *value*)

Sets the value of the item with the specified name

class domonic.dom.DOMTimeStamp(*value*)

The DOMTimeStamp interface represents a numeric value which represents the number of milliseconds since the epoch.

class domonic.dom.DOMTokenList(*element*: *Node*)

DOMTokenList represents a set of space-separated tokens.

add(**args*)

Adds the given tokens to the list

contains(*token*) → *bool*

Returns true if the token is in the list, and false otherwise

item(*index*: *int*)

Returns the token at the specified index

remove(**args*)

Removes the given tokens from the list

toString() → *str*

Returns a string containing all tokens in the list, with spaces separating each token

toggle(*token*, *force*=None)

If force is not given, removes token from list if present, otherwise adds token to list. If force is true, adds token to list, and if force is false, removes token from list if present.

class domonic.dom.Document(**args*, ***kwargs*)

The Document interface represents the entire HTML or XML document.

property anchors

[get the anchors in the document]

property applets

Returns a collection of all <applet> elements in the document

property body

Returns the <body> element in the document

property characterSet

Returns the character encoding for the document

property charset

Use characterSet instead.

Type

Returns the character encoding for the document. Deprecated

close()

Closes the output stream previously opened with document.open()

static createAttribute(*name*)

Creates an attribute node

static createCDATASection(*data*)

Creates a CDATASection node with the specified data

static createComment(*message*)

Creates a Comment node with the specified text

static createDocumentFragment(args*)**

Creates an empty DocumentFragment node if not content passed. I added args as optional to pass content

static createElement(*_type: str, *args, **kwargs*)

Creates an Element node

static createElementNS(*namespaceURI, qualifiedName, options=None*)

Creates an element with the specified namespace URI and qualified name.

static createEntityReference(*name*)

Creates an EntityReference node with the specified name

static createEvent(*event_type=None*)

[Creates a new event]

Parameters

event_type (*[type]*, *optional*) – [description]. Defaults to None.

Returns

[a new event]

Return type

[type]

static createExpression(*xpath, nsResolver*)

Creates an XPathExpression object for the given XPath string.

static createNodeIterator(*root, whatToShow=None, filter=None*)

Creates a NodeIterator that can be used to traverse the document tree or subtree under root.

static createProcessingInstruction(*target, data*)

Creates a ProcessingInstruction node with the specified target and data

static createRange()

Creates a Range

static createTextNode(*text*)

[Creates a Text node with the specified text.

Parameters

text (*[str]*) – [the text to be inserted]

Returns

[a new Text node]

Return type

[*type*]

static createTreeWalker(*root*, *whatToShow=*None, *filter=*None, *entityReferenceExpansion=*None)

[creates a TreeWalker object]

Parameters

- **root** (*[type]*) – [the root node at which to begin traversal]
- **whatToShow** (*[type]*, *optional*) – [what types of nodes to show]. Defaults to None.
- **filter** (*[type]*, *optional*) – [a NodeFilter or a function to be called for each node]. Defaults to None.

Returns

[a new TreeWalker object]

Return type

[*type*]

property doctype

Returns the Document Type Declaration associated with the document

domConfig()

Returns the DOMConfig which has settings for how html content is rendered

domain()

Returns the domain name of the server that loaded the document

elementFromPoint(*x*, *y*)

Returns the topmost element at the specified coordinates.

elementsFromPoint(*x*, *y*)

Returns an array of all elements at the specified coordinates.

property embeds

[Returns a collection of all <embed> elements the document]

Returns

[a collection of all <embed> elements the document]

Return type

[*type*]

evaluate(*xpathExpression: str*, *contextNode: Optional[Node]* = None, *namespaceResolver=*None, *resultType=*7, *result=*None)

Evaluates an XPath expression and returns the result.

property forms

Returns a collection of all <form> elements in the document

fullscreenElement()

Returns the current element that is displayed in fullscreen mode

fullscreenEnabled()

Returns a Boolean value indicating whether the document can be viewed in fullscreen mode

getElementById(*_id*)

[Returns the element that has the ID attribute with the specified value]

Parameters

_id (*[str]*) – [the value of the ID attribute]

Returns

[the element that has the ID attribute with the specified value]

Return type

[*type*]

getElementsByName(*name: str*)

[Returns a NodeList containing all elements with a specified name]

Parameters

name (*str*) – [the name to search for]

Returns

[the matching elements]

Return type

[*type*]

property head: *HTMLHeadElement*

Returns the <head> element of the document

property images

Returns a collection of all elements in the document

property implementation

Returns the DOMImplementation object that handles this document

importNode(*node, deep=False*)

Imports a node from another document to this document.

property links

Returns a collection of all <a> and <area> elements in the document that have a href attribute

normalizeDocument()

Removes empty Text nodes, and joins adjacent nodes

open(*index='index.html'*)

Opens an HTML output stream to collect output from document.write()

property pictureInPictureEnabled

Returns whether Picture-in-Picture mode is enabled.

renameNode(*node, namespaceURI: str, nodename: str*)

[Renames the specified node, and returns the renamed node.]

Parameters

- **node** (*[type]*) – [the node to rename]
- **namespaceURI** (*[type]*) – [a namespace URI]
- **nodename** (*[type]*) – [a node name]

Returns

[description]

Return type*[type]***property scripts**

[Returns a collection of <script> elements in the document]

Returns

[a collection of <script> elements in the document]

Return type*[type]***strictErrorChecking()**

Returns a Boolean value indicating whether to stop on the first error

property title: str

[gets the title of the document]

Returns

The title of the document

Return type*[str]***property visibilityState**

Returns the visibility state of the document

write(html: str = "")

[writes HTML text to a document]

Parameters**html** (*str*, *optional*) – [the content to write to the document]**writeln(html: str = "")**

[writes HTML text to a document, followed by a line break]

Parameters**html** (*str*, *optional*) – [the content to write to the document]**property xmlversion**

Returns the version of XML used for the document

class domonic.dom.DocumentFragment(*args)**property attributes: NamedNodeMap**

Returns a NamedNodeMap of an element's attributes

getElementById(_id)

[Returns the element that has the ID attribute with the specified value]

Parameters**_id** (*[str]*) – [the value of the ID attribute]

Returns

[the element that has the ID attribute with the specified value]

Return type

[*type*]

getElementsByTagName(*tagName: str*) → *HTMLCollection*

[Returns a collection of all child elements with the specified tag name]

Parameters

tagName (*str*) – [a DOMString representing the tag name to match]

Returns

[method returns a live HTMLCollection of elements with the given tag name.]

Return type

[*type*]

querySelector(*query: str*)

[Returns the first child element that matches a specified CSS selector(s) of an element]

Parameters

query (*str*) – [a CSS selector string]

Returns

[an Element object]

Return type

[*type*]

querySelectorAll(*query: str*)

[Returns all child elements that matches a specified CSS selector(s) of an element]

Parameters

query (*str*) – [a CSS selector string]

Returns

[a list of Element objects]

Return type

[*type*]

replaceChildren(*newChildren*) → *None*

Replaces the childNodes of the DocumentFragment object.

class domonic.dom.**DocumentType**(*name: str = 'html', publicId: str = "", systemId: str = ""*)

internalSubset()

A DOMString of the internal subset, or None. Eg “<!ELEMENT foo (bar)>”.

notations() → *NamedNodeMap*

A *NamedNodeMap* with notations declared in the DTD.

class domonic.dom.**Element**(*args, **kwargs)

Baseclass for all html tags

append(*args)

Inserts a set of Node objects or DOMString objects after the last child of the Element.

property attributes: *NamedNodeMap*

Returns a *NamedNodeMap* of an element’s attributes

blur()

Removes focus from an element

property classList

Returns the value of the classList attribute of an element

property className

Sets or returns the value of the className attribute of an element

click()

Simulates a mouse-click on an element

property clientHeight

Returns the height of an element, including padding

property clientLeft

Returns the width of the left border of an element

property clientTop

Returns the width of the top border of an element

property clientWidth

Returns the width of an element, including padding

property contentEditable: `bool`

Sets or returns whether an element is editable

property dataset

Returns the value of the dataset attribute of an element

property dir

returns the value of the dir attribute of an element

exitFullscreen()

Cancels an element in fullscreen mode

firstElementChild()

Returns the first child element of an element

focus()

Sets focus on an element

getAttribute(*attribute: `str`*) → `str`

Returns the specified attribute value of an element node

getAttributeNS(*namespaceURI, localName*)

Returns the value of the specified attribute

getAttributeNode(*attribute: `str`*) → `str`

Returns the specified attribute node

getAttributeNodeNS(*attr*)

Sets the attribute node of an element

getBoundingClientRect()

Returns the size of an element and its position relative to the viewport

getElementsByClassName(*className: str*) → *HTMLCollection*

[Returns a collection of all child elements with the specified class name]

Parameters

className (*str*) – [a DOMString representing the class name to match]

Returns

[a NodeList of all child elements with the specified class name]

Return type

[*type*]

getElementsBySelector(*all_selectors, document*)

Get DOM elements based on the given CSS Selector <https://simonwillison.net/2003/Mar/25/getElementsBySelector/> < original author http://www.openjs.com/scripts/dom/css_selector/ < ported to support ‘,’ <https://bin-co.com/python/scripts/getelementsbyselector-html-css-query.php> < ported to py2 (broken/bugs) *BSD LICENSED*

note - always needs a tag in the query i.e. (‘a.classname’) will work. but just (‘.classname’) wont fixed and ported to py3 here. quite cool means other peoples code works on my dom # TODO - needs to work in conjunctions with `_matchElement` so `querySelector` works a bit better and `dQuery` picks it up # TOOD - `*` = node content

Parameters

- **all_selectors** ([*type*]) – [description]
- **document** ([*type*]) – [description]

Returns

[description]

Return type

[*type*]

getElementsByTagName(*tagName: str*) → *HTMLCollection*

[Returns a collection of all child elements with the specified tag name]

Parameters

tagName (*str*) – [a DOMString representing the tag name to match]

Returns

[method returns a live HTMLCollection of elements with the given tag name.]

Return type

[*type*]

hasAttribute(*attribute: str*) → *bool*

Returns True if an element has the specified attribute, otherwise False

Parameters

attribute (*str*) – [the attribute to test for]

Returns

[True if an element has the specified attribute, otherwise False]

Return type

bool

hasAttributes() → *bool*

Returns true if an element has any attributes, otherwise false

property id

Sets or returns the value of the id attribute of an element

property innerHTML

Sets or returns the content of an element

insertAdjacentElement(*position: str, element*)

Inserts an element adjacent to the current element

insertAdjacentHTML(*position: str, html: str*)

Inserts raw HTML adjacent to the current element

insertAdjacentText(*position: str, text: str*)

Inserts text adjacent to the current element

isContentEditable() → *bool*

Returns true if the content of an element is editable, otherwise false

lastElementChild()

[Returns the last child element of an element]

Returns

[the last child element of an element]

Return type

[*type*]

matches(*s: str*) → *bool*

[checks to see if the Element would be selected by the provided selectorString]

<https://developer.mozilla.org/en-US/docs/Web/API/Element/matches>

Parameters

s (*str*) – [css selector]

Returns

[True if selector matches Element otherwise False]

Return type

[*bool*]

namespaceURI()

Returns the namespace URI of an element

property nextElementSibling

Returns the next element at the same node tree level

property nextSibling

Returns the next node at the same node tree level

normalize()

Joins adjacent text nodes and removes empty text nodes in an element

offsetHeight()

Returns the height of an element, including padding, border and scrollbar

offsetLeft()

Returns the horizontal offset position of an element

offsetParent()

Returns the offset container of an element

offsetTop()

Returns the vertical offset position of an element

offsetWidth()

Returns the width of an element, including padding, border and scrollbar

property parentElement

Returns the parent element node of an element

prepend(*args)

Prepends a node to the current element

property previousElementSibling

returns the Element immediately prior to the specified one in its parent's children list, or None if the specified element is the first one in the list.

querySelector(query: *str*)

[Returns the first child element that matches a specified CSS selector(s) of an element]

Parameters

query (*str*) – [a CSS selector string]

Returns

[an Element object]

Return type

[*type*]

querySelectorAll(query: *str*)

[Returns all child elements that matches a specified CSS selector(s) of an element]

Parameters

query (*str*) – [a CSS selector string]

Returns

[a list of Element objects]

Return type

[*type*]

remove()

Removes the element from the DOM

removeAttribute(attribute: *str*)

Removes a specified attribute from an element

removeAttributeNS(namespaceURI, localName)

Removes an attribute from an element

removeAttributeNode(attribute)

Removes a specified attribute node, and returns the removed node

requestFullscreen()

Shows an element in fullscreen mode

scrollHeight()

Returns the entire height of an element, including padding

scrollIntoView()

Scrolls the specified element into the visible area of the browser window

scrollLeft()

Sets or returns the number of pixels an element's content is scrolled horizontally

scrollTop()

Sets or returns the number of pixels an element's content is scrolled vertically

scrollWidth()

Returns the entire width of an element, including padding

setAttribute(*attribute*, *value*)

Sets or changes the specified attribute, to the specified value

setAttributeNS(*namespaceURI*, *localName*, *value*)

Sets an attribute in the given namespace

setAttributeNode(*attr*)

[Sets or changes the specified attribute node]

Parameters

attr (*[type]*) – [an Attr object]

setAttributeNodeNS(*attr*)

Sets the attribute node of an element

property style

returns the value of the style attribute of an element

property title

returns the value of the title attribute of an element

toString() → str

Converts an element to a string

class domonic.dom.Entity(*args)**static fromChar(*char*: str) → str**

Returns the character corresponding to the given entity name.

static fromName(*entityName*: str) → str

Returns the entity name corresponding to the given character.

class domonic.dom.EntityReference(*args)

The EntityReference interface represents a reference to an entity, either parsed or unparsed, in an Entity Node. Note that this is not a CharacterData node, and does not have any child nodes.

static fromOrdinal(*ordinal*: int)

Returns the entity name corresponding to the given character.

static ordinal(*entityName*: str)

Returns the character corresponding to the given entity name.

class domonic.dom.HTMLAnchorElement(*args, **kwargs)

```
class domonic.dom.HTMLAreaElement(*args, **kwargs)

class domonic.dom.HTMLAudioElement(*args, autoplay: Optional[bool] = None, controls=None, loop=None,
                                     muted=None, preload=None, src=None, **kwargs)

class domonic.dom.HTMLBRElement(*args, **kwargs)

class domonic.dom.HTMLBaseElement(*args, href=None, target=None, **kwargs)

class domonic.dom.HTMLBaseFontElement(*args, **kwargs)

class domonic.dom.HTMLBodyElement(*args, aLink=None, background=None, bgColor=None, link=None,
                                    onload=None, onunload=None, text=None, vLink=None, **kwargs)

class domonic.dom.HTMLButtonElement(*args, disabled: Optional[bool] = None, form=None, formaction:
                                     Optional[str] = None, formenctype=None, formmethod=None,
                                     formnovalidate=None, formtarget=None, name=None, type=None,
                                     value=None, **kwargs)

class domonic.dom.HTMLCanvasElement(*args, width: Optional[int] = None, height: Optional[int] = None,
                                     **kwargs)

class domonic.dom.HTMLCollection(iterable=(),/)

    item(index: int)
        [gets the indexth item in the collection. If index is greater than or equal to the number of nodes in the list,
         this returns null.]

        Parameters
            index ([type]) – [the index of the item to return.]

        Returns
            [the node at the indexth position, or None]

        Return type
            [type]

    namedItem(name: str)
        Returns the specific node whose ID or, as a fallback, name matches the string specified by name.

class domonic.dom.HTMLContentElement(*args, **kwargs)

class domonic.dom.HTMLDListElement(*args, **kwargs)

class domonic.dom.HTMLDataElement(*args, **kwargs)

class domonic.dom.HTMLDataListElement(*args, **kwargs)

class domonic.dom.HTMLDialogElement(*args, open=None, **kwargs)

class domonic.dom.HTMLDivElement(*args, **kwargs)

class domonic.dom.HTMLDocument(*args, **kwargs)

class domonic.dom.HTMLElement(*args, **kwargs)

class domonic.dom.HTMLEmbedElement(*args, **kwargs)

class domonic.dom.HTMLFieldSetElement(*args, **kwargs)
```



```

class domonic.dom.HTMLFormControlsCollection(*args, **kwargs)

class domonic.dom.HTMLFormElement(*args, action: Optional[str] = None, autocomplete=None, enctype:
    Optional[str] = None, method: Optional[str] = None, name:
    Optional[str] = None, novalidate: Optional[bool] = None, target=None,
    **kwargs)

class domonic.dom.HTMLFrameSetElement(*args, **kwargs)

class domonic.dom.HTMLHRElement(*args, **kwargs)

class domonic.dom.HTMLHeadElement(*args, **kwargs)

class domonic.dom.HTMLHeadingElement(*args, **kwargs)

class domonic.dom.HTMLIFrameElement(*args, src=None, name=None, sandbox=None,
    allowfullscreen=None, **kwargs)

class domonic.dom.HTMLImageElement(*args, alt=None, src=None, crossorigin=None, height=None,
    ismap=None, longdesc=None, sizes=None, srcset=None,
    usemap=None, width=None, **kwargs)

class domonic.dom.HTMLInputElement(*args, accept=None, alt=None, autocomplete=None, autofocus=None,
    checked=None, dirname=None, disabled=None, form=None,
    formaction=None, formenctype=None, formmethod=None,
    formnovalidate=None, formtarget=None, height=None, _list=None,
    _max=None, maxlength=None, _min=None, multiple=None,
    name=None, pattern=None, placeholder=None, readonly=None,
    required=None, size=None, src=None, step=None, type=None,
    value=None, width=None, **kwargs)

class domonic.dom.HTMLIsIndexElement(*args, **kwargs)

class domonic.dom.HTMLKeygenElement(*args, **kwargs)

class domonic.dom.HTMLLIElement(*args, **kwargs)

class domonic.dom.HTMLLabelElement(*args, **kwargs)

class domonic.dom.HTMLLegendElement(*args, **kwargs)

class domonic.dom.HTMLLinkElement(*args, **kwargs)

class domonic.dom.HTMLMapElement(*args, **kwargs)

class domonic.dom.HTMLMediaElement(*args, **kwargs)

class domonic.dom.HTMLMetaElement(*args, charset=None, content=None, http_equiv=None, name=None,
    **kwargs)

class domonic.dom.HTMLMeterElement(*args, value=None, _min=None, _max=None, low=None, high=None,
    optimum=None, **kwargs)

class domonic.dom.HTMLModElement(*args, **kwargs)

class domonic.dom.HTMLOLListElement(*args, **kwargs)

class domonic.dom.HTMLObjectElement(*args, **kwargs)

```

```
class domonic.dom.HTMLOptGroupElement(*args, **kwargs)

class domonic.dom.HTMLOptionElement(*args, disabled=None, label=None, selected=None, value=None,
                                     **kwargs)

class domonic.dom.HTMLOutputElement(*args, **kwargs)

class domonic.dom.HTMLParagraphElement(*args, **kwargs)

class domonic.dom.HTMLParamElement(*args, **kwargs)

class domonic.dom.HTMLPictureElement(*args, **kwargs)

class domonic.dom.HTMLPortalElement(*args, **kwargs)

class domonic.dom.HTMLPreElement(*args, **kwargs)

class domonic.dom.HTMLProgressElement(*args, **kwargs)

class domonic.dom.HTMLQuoteElement(*args, **kwargs)

class domonic.dom.HTMLScriptElement(*args, **kwargs)

class domonic.dom.HTMLSelectElement(*args, autofocus: Optional[bool] = None, disabled: Optional[bool] =
                                     None, multiple: Optional[bool] = None, name: Optional[str] =
                                     None, required: Optional[bool] = None, size: Optional[int] = None,
                                     **kwargs)

class domonic.dom.HTMLShadowElement(*args, **kwargs)

class domonic.dom.HTMLSourceElement(*args, **kwargs)

class domonic.dom.HTMLSpanElement(*args, **kwargs)

class domonic.dom.HTMLStyleElement(*args, **kwargs)

class domonic.dom.HTMLTableCaptionElement(*args, **kwargs)

class domonic.dom.HTMLTableCellElement(*args, **kwargs)

class domonic.dom.HTMLTableColElement(*args, **kwargs)

class domonic.dom.HTMLTableDataCellElement(*args, **kwargs)

class domonic.dom.HTMLTableElement(*args, align: Optional[str] = None, bgcolor=None, border=None,
                                    cellpadding=None, cellspacing=None, frame=None, rules=None,
                                    summary=None, width=None, **kwargs)

class domonic.dom.HTMLTableHeaderCellElement(*args, **kwargs)

class domonic.dom.HTMLTableRowElement(*args, **kwargs)

class domonic.dom.HTMLTableSectionElement(*args, **kwargs)

class domonic.dom.HTMLTemplateElement(*args, **kwargs)

class domonic.dom.HTMLTextAreaElement(*args, autofocus=None, cols=None, disabled=None, form=None,
                                       maxlength=None, name=None, placeholder=None,
                                       readonly=None, required=None, rows=None, wrap=None,
                                       **kwargs)
```

```

class domonic.dom.HTMLTimeElement(*args, **kwargs)
class domonic.dom.HTMLTitleElement(*args, **kwargs)
class domonic.dom.HTMLTrackElement(*args, **kwargs)
class domonic.dom.HTMLULListElement(*args, **kwargs)
class domonic.dom.HTMLUnknownElement(*args, **kwargs)
class domonic.dom.HTMLVideoElement(*args, autoplay=None, controls=None, height=None, loop=None,
                                     muted=None, poster=None, preload=None, src=None, width=None,
                                     **kwargs)

```

```
class domonic.dom.Node(*args, **kwargs)
```

An abstract base class upon which many other DOM API objects are based

appendChild(*aChild*: [Node](#)) → [Node](#)

Adds a child to the current element. If item is a DocumentFragment, all its children are added.

Parameters

item ([Node](#)) – The Node to add.

property attrib

Returns the attributes of the current node as a dict not a NamedNodeMap

property childElementCount: [int](#)

Returns the number of child elements an element has

property childNodes: [NodeList](#)

Returns a live NodeList containing all the children of this node

property children

Returns a collection of an element's child element (excluding text and comment nodes)

cloneNode(*deep*: [bool](#) = [True](#))

Returns a copy.

compareDocumentPosition(*otherElement*) → [int](#)

An integer value representing otherNode's position relative to node as a bitmask combining the following constant properties of Node:

<https://stackoverflow.com/questions/8334286/cross-browser-compare-document-position>

contains(*node*)

Check whether a node is a descendant of a given node

property firstChild

Returns the first child node of an element

hasChildNodes() → [bool](#)

Returns true if an element has any child nodes, otherwise false

insertBefore(*new_node*, *reference_node*=[None](#))

inserts a node before a reference node as a child of a specified parent node. this will remove the node from its previous parent node, if any.

TODO - can throw value error if wrong ordered params. may be helpful to catch to say so.

isDefaultNamespace(*ns*)

Checks if a namespace is the default namespace

isEqualNode(*node*)

Checks if two elements are equal

isSameNode(*node*)

Checks if two elements are the same node

iter(*tag=None*)

Creates a tree iterator with the current element as the root. The iterator iterates over this element and all elements below it, in document (depth first) order. If tag is not None or '*', only elements whose tag equals tag are returned from the iterator. If the tree structure is modified during iteration, the result is undefined.

property lastChild

Returns the last child node of an element

lookupNamespaceURI(*ns: str*)

Returns the namespace URI for a given prefix

Parameters

ns – prefix - i.e 'xml', 'xlink', 'svg', etc

lookupPrefix(*ns*)

Returns the prefix for a given namespace URI

property nextSibling

[returns the next sibling of the current node.]

property nodeName

Returns the name of a node

property nodeValue

Sets or returns the value of a node

normalize()

Normalize a node's value

property ownerDocument

Returns the root element (document object) for an element

property previousSibling

[returns the previous sibling of the current node.]

removeChild(*node*)

removes a child node from the DOM and returns the removed node.

replaceChild(*newChild, oldChild*)

[Replaces a child node within the given (parent) node.]

Parameters

- **newChild** ([*type*]) – [a Node object]
- **oldChild** ([*type*]) – [a Node object]

Returns

[the old child node]

Return type

[*type*]

property rootNode

[read-only property returns a Node object representing the topmost node in the tree, or the current node if it's the topmost node in the tree]

Returns

[the topmost Node in the tree]

Return type

[*Node*]

property tag

Returns the tag name of the current node

property tail

Returns the text content of the current node

property text

Returns the text content of the current node

property textContent

Returns the text content of a node and its descendants

```
class domonic.dom.NodeIterator(root, whatToShow=4294967295, filter=None,
                               entityReferenceExpansion=False)
```

[NodeIterator is an iterator object that iterates over the descendants of a node, in tree order.]

nextNode()

Returns the next Node in the document, or null if there are none.

pointerBeforeReferenceNode() → *bool*

Returns a boolean flag that indicates whether the NodeIterator is anchored before, the flag being true, or after, the flag being false, the anchor node.

previousNode()

Returns the previous Node in the document, or null if there are none.

referenceNode() → *Node*

Returns the Node that is being iterated over.

```
class domonic.dom.NodeList(iterable=(None), /)
```

NodeList objects are collections of nodes

entries() → *Iterable*[*Tuple*[*int*, *Node*]]

Returns an iterator, allowing code to go through all key/value pairs contained in the collection. (In this case, the keys are numbers starting from 0 and the values are nodes.)

forEach(*func*, *thisArg*=None) → *None*

Calls a function for each item in the NodeList.

item(*index*) → *Node*

Returns an item in the list by its index, or null if the index is out-of-bounds.

keys() → *Iterable*[*int*]

Returns an iterator, allowing code to go through all the keys of the key/value pairs contained in the collection. (In this case, the keys are numbers starting from 0.)

values() → *Iterable*[*Node*]

Returns an iterator allowing code to go through all values (nodes) of the key/value pairs contained in the collection.

```
class domonic.dom.ParentNode(*args, **kwargs)
    not tested yet

    property children: NodeList
        Return list of child nodes.

    property firstElementChild
        First Element child node.

    property lastElementChild
        Last Element child node.

class domonic.dom.ProcessingInstruction(target, data)

class domonic.dom.RadioNodeList(name: str)

    property value
        Returns the value of the first element in the collection, or null if there are no elements in the collection.

class domonic.dom.ShadowRoot(host, mode='open')
    property on element that has hidden DOM

    caretPositionFromPoint()
        Returns a CaretPosition object containing the DOM node containing the caret, and caret's character offset
        within that node.

    elementFromPoint(x, y)
        Returns the topmost element at the specified coordinates.

    elementsFromPoint(x, y)
        Returns an array of all elements at the specified coordinates.

    getSelection()
        Returns a Selection object representing the range of text selected by the user, or the current position of the
        caret.

class domonic.dom.Text(*args, **kwargs)
    Text Node

    property assignedSlot
        Returns the slot whose assignedNodes contains this node.

    property childNodes
        Returns a live NodeList containing all the children of this node

    property firstChild
        Returns the first child node of an element

    property nodeName
        Returns the name of a node

    splitText(offset: int)
        Splits the Text node into two Text nodes at the specified offset, keeping both in the tree as siblings. The
        first node is returned, while the second node is discarded and exists outside the tree.

    property wholeText
        Returns a DOMString containing all the text content of the node and its descendants.
```

```
class domonic.dom.TreeWalker(node, whatToShow=4294967295, _filter=None,
                             expandEntityReferences=False)
```

The TreeWalker object represents the nodes of a document subtree and a position within them.

firstChild()

Moves the current Node to the first visible child of the current node, and returns the found child. It also moves the current node to this child. If no such child exists, returns null and the current node is not changed.

lastChild()

Moves the current Node to the last visible child of the current node, and returns the found child. It also moves the current node to this child. If no such child exists, null is returned and the current node is not changed.

nextNode()

Moves the current Node to the next visible node in the document order, and returns the found node. It also moves the current node to this one. If no such node exists, returns None and the current node is not changed. can be used in a while loop to iterate over all the nodes in the document order.

nextSibling()

Moves the current Node to its next sibling, if any, and returns the found sibling. If there is no such node, null is returned and the current node is not changed.

parentNode()

Moves the current Node to the first visible ancestor node in the document order, and returns the found node. It also moves the current node to this one. If no such node exists, or if it is before that the root node defined at the object construction, returns null and the current node is not changed.

previousNode()

Moves the current Node to the previous visible node in the document order, and returns the found node. It also moves the current node to this one. If no such node exists, or if it is before that the root node defined at the object construction, returns null and the current node is not changed.

previousSibling()

Moves the current Node to its previous sibling, if any, and returns the found sibling. If there is no such node, return null and the current node is not changed.

property root

Returns a Node representing the root node as specified when the TreeWalker was created.

tree

Is a boolean value indicating, when discarding an entity reference its whole sub-tree must be discarded at the same time.

whatToShow(options)

Returns an unsigned long being a bitmask made of constants describing the types of Node that must be presented. Non-matching nodes are skipped, but their children may be included, if relevant. The possible values are:

```
domonic.dom.console
```

```
# self.screen = type('screen', (DOM,), {'name':'screen'})
```

domonic.javascript

- https://www.w3schools.com/jsref/jsref_reference.asp
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>

class domonic.javascript.**Array**(*args)

javascript array

at(index: *int*)

[takes an integer value and returns the item at that index, allowing for positive and negative integers. Negative integers count back from the last item in the array.]

Parameters**index** ([*type*]) – [position of item]**Returns**

[item at the given position]

Return type[*type*]**concat**(*args)

[Joins two or more arrays, and returns a copy of the joined arrays]

Returns

[returns a copy of the joined arrays]

Return type[*list*]**copyWithin**(target, start=0, end=None)

Copies array elements within the array, from start to end

entries()

[Returns a key/value pair Array Iteration Object]

Yields[*type*] – [key/value pair]**every**(func)

[Checks if every element in an array pass a test]

Parameters**func** ([*type*]) – [test function]**Returns**

[if every array elemnt passed the test]

Return type[*bool*]**fill**(value=None, start=None, end=None)

[Fills elements of an array from a start index to an end index with a static value]

filter(func)

Creates a new array with every element in an array that pass a test i.e. even_numbers = someArr.filter(lambda x: x % 2 == 0)

find(func)

Returns the value of the first element in an array that pass a test

findIndex(*value*)

Returns the index of the first element in an array that pass a test

findLast(*callback*)

[Returns the last element in an array that passes a test]

findLastIndex(*callback*)

[Returns the last index of an element in an array that passes a test]

flat(*depth=1*)

[Flattens an array into a single-dimensional array or a depth of arrays]

flatMap(*fn*)

[Maps a function over an array and flattens the result]

forEach(*func*)

Calls a function for each array element

static from_(*obj*)

Creates a new Array instance from an array-like or iterable object.

groupBy(*callback*) → *dict*

[Groups the elements of an array according to the result of calling a callback function on each element]

Parameters

callback (*callable*) – [the callback receives the following parameters (value, index, target)]

Returns

[a dictionary of arrays]

Return type

[*dict*]

includes(*value*)

[Check if an array contains the specified item]

Parameters

value (*[any]*) – [any value]

Returns

[a boolean]

Return type

[*bool*]

indexOf(*value*)

Search the array for an element and returns its position

static isArray(*thing*)

[Checks whether an object is an array]

Parameters

thing (*[type]*) – [thing to check]

Returns

[True if the object is list, tuple or Array]

Return type

[*bool*]

join(*value*)

Joins all elements of an array into a string

keys()

Returns a Array Iteration Object, containing the keys of the original array

lastIndexOf(*value*)

Search the array for an element, starting at the end, and returns its position

property length

Sets or returns the number of elements in an array

map(*func*)

[Creates a new array with the result of calling a function for each array element]

Parameters

func (*[type]*) – [a function to call on each array element]

Returns

[a new array]

Return type

[*list*]

static of(**args*)

Creates a new Array instance with a variable number of arguments, regardless of number or type of the arguments.

pop()

Removes the last element of an array, and returns that element

prototype

alias of *Array*

push(*value*)

Adds new elements to the end of an array, and returns the new length

reduce(*callback, initialValue=None*)

Reduces the array to a single value (going left-to-right) callback recieve theses parameters: previousValue, currentValue, currentIndex, array

reduceRight(*callback, initialValue=None*)

Reduces the array to a single value (going right-to-left) callback recieve theses parameters: previousValue, currentValue, currentIndex, array

reverse()

Reverses the order of the elements in an array

shift()

[removes the first element from an array and returns that removed element]

Returns

[the removed array element]

Return type

[*type*]

slice(*start=0, stop=None, step=1*)

[Selects a part of an array, and returns the new array]

Parameters

- **start** (*[int]*) – [index to slice from]
- **stop** (*[int], optional*) – [index to slice to]. Defaults to end of the array.
- **step** (*int, optional*) – [description]. Defaults to 1.

Returns

[new array]

Return type

[*type*]

some(*func*)

Checks if any of the elements in an array pass a test

sort(*func=None*)

Sorts the elements of an array

splice(*start, delete_count=None, *items*)

Selects a part of an array, and returns the new array

toSource()

Returns the source array.

toString()

Converts an array to a string, and returns the result

unshift(**args*)

[Adds new elements to the beginning of an array, and returns the new length]

Returns

[the length of the array]

Return type

[*int*]

class domonic.javascript.**Boolean**(*value=False*)

[Creates a Boolean Object. Warning this is NOT a boolean type. for that use Global.Boolean()]

class domonic.javascript.**Date**(*date=None, *args, formatter='python', **kwargs*)

javascript date

UTC()

Returns the number of milliseconds in a date since midnight of January 1, 1970, according to UTC time

getDate()

Returns the day of the month (from 1-31)

getDay()

Returns the day of the week (from 0-6 : Sunday-Saturday)

Returns

An integer number, between 0 and 6, corresponding to the day of the week for the given date, according to local time: 0 for Sunday, 1 for Monday, 2 for Tuesday, and so on

Return type

int

getFullYear()

Returns the year

getHours()

Returns the hour (from 0-23)

getMilliseconds()

Returns the milliseconds (from 0-999)

getMinutes()

Returns the minutes (from 0-59)

getMonth()

Returns the month (from 0-11)

getSeconds()

Returns the seconds (from 0-59)

getTime()

Returns A number representing the milliseconds elapsed between 1 January 1970 00:00:00 UTC and self.date

getTimezoneOffset()

Returns the difference, in minutes, between a date as evaluated in the UTC time zone, and the same date as evaluated in the local time zone

getUTCDate()

Returns the day of the month, according to universal time (from 1-31)

getUTCDay()

Returns the day of the week, according to universal time (from 0-6)

getUTCFullYear()

Returns the year, according to universal time

getUTCHours()

Returns the hour, according to universal time (from 0-23)

getUTCMilliseconds()

Returns the milliseconds, according to universal time (from 0-999)

getUTCMinutes()

Returns the minutes, according to universal time (from 0-59)

getUTCMonth()

Returns the month, according to universal time (from 0-11)

getUTCSeconds()

Returns the seconds, according to universal time (from 0-59)

getYear()

Deprecated. Use the `getFullYear()` method instead

static now()

Returns the number of milliseconds since midnight Jan 1, 1970

static parse(*date_string*)

Parses a date string and returns the number of milliseconds since January 1, 1970

setDate(*day*: *int*)

Sets the day of the month of a date object

Parameters

day (*int*) – An integer representing the day of the month.

Returns

milliseconds between epoch and updated date.

Return type

int

setFullYear(*yearValue*: *int*, *monthValue*: *Optional*[*int*] = *None*, *dateValue*: *Optional*[*int*] = *None*)

Sets the year of a date object

Parameters

- **yearValue** (*_type_*) – *_description_*
- **monthValue** (*int*, *optional*) – *_description_*. Defaults to *None*.
- **dateValue** (*int*, *optional*) – *_description_*. Defaults to *None*.

Returns

milliseconds between epoch and updated date.

Return type

int

setHours(*hoursValue*: *int*, *minutesValue*: *Optional*[*int*] = *None*, *secondsValue*: *Optional*[*int*] = *None*, *msValue*: *Optional*[*int*] = *None*)

Sets the hour of a date object

Parameters

- **hoursValue** (*int*) – an integer between 0 and 23
- **minutesValue** (*int*, *optional*) – an integer between 0 and 59
- **secondsValue** (*int*, *optional*) – an integer between 0 and 59,
- **msValue** (*int*, *optional*) – a number between 0 and 999,

Returns

milliseconds between epoch and updated date.

Return type

int

setMilliseconds(*milliseconds*: *int*)

Sets the milliseconds of a date object

Parameters

milliseconds (*int*) – Milliseconds to set i.e 123

setMinutes(*minutesValue*: *int*, *secondsValue*: *Optional*[*int*] = *None*, *msValue*: *Optional*[*int*] = *None*)

Set the minutes of a date object

Parameters

- **minutesValue** (*int*, *optional*) – an integer between 0 and 59
- **secondsValue** (*int*, *optional*) – an integer between 0 and 59,
- **msValue** (*int*, *optional*) – a number between 0 and 999,

Returns

milliseconds between epoch and updated date.

Return type

`int`

setMonth(*monthValue*: `int`, *dayValue*: `Optional[int]` = `None`)

Sets the month of a date object

Parameters

- **monthValue** (`int`) – a number from 0 to 11 indicating the month.
- **dayValue** (`int`, `optional`) – an optional day of the month. Defaults to 0.

Returns

milliseconds between epoch and updated date.

Return type

`int`

setSeconds(*secondsValue*: `int`, *msValue*: `Optional[int]` = `None`)

Sets the seconds of a date object

Parameters

- **secondsValue** (`int`) – `_description_`
- **msValue** (`int`, `optional`) – `_description_`. Defaults to `None`.

Returns

milliseconds between epoch and updated date.

Return type

`int`

setTime(*milliseconds*: `Optional[int]` = `None`)

Sets the date and time of a date object

Parameters

milliseconds (`_type_`, `optional`) – `_description_`. Defaults to `None`.

Returns

`_description_`

Return type

`_type_`

setUTCDate(*day*)

Sets the day of the month of a date object, according to universal time

setUTCFullYear(*year*)

Sets the year of a date object, according to universal time

setUTCHours(*hour*)

Sets the hour of a date object, according to universal time

setUTCMilliseconds(*milliseconds*)

Sets the milliseconds of a date object, according to universal time

setUTCMinutes(*minutes*)

Set the minutes of a date object, according to universal time

setUTCMonth(*month*)

Sets the month of a date object, according to universal time

setUTCSeconds(*seconds*)

Set the seconds of a date object, according to universal time

setYear(*year*)

Deprecated. Use the `setFullYear()` method instead

toDateString()

Converts the date portion of a Date object into a readable string

toGMTString()

Deprecated. Use the `toUTCString()` method instead

toISOString()

Returns the date as a string, using the ISO standard

toJSON()

Returns the date as a string, formatted as a JSON date

toLocaleDateString()

Returns the date portion of a Date object as a string, using locale conventions

toLocaleString()

Converts a Date object to a string, using locale conventions

toLocaleTimeString()

Returns the time portion of a Date object as a string, using locale conventions

toString()

Returns a string representation of the date

getTimeString()

Converts the time portion of a Date object to a string

toUTCString()

Converts a Date object to a string, according to universal time

exception `domonic.javascript.Error`(*message*, **args*, ***kwargs*)

Raise Errors

class `domonic.javascript.Function`(*func*, **args*, ***kwargs*)

a Function object

apply(*thisArg=None*, *args=None*, ***kwargs*)

[calls a function with a given this value, and arguments provided as an array]

Parameters

thisArg ([*type*]) – [The value of this provided for the call to func.]

Returns

[result of calling the function.]

Return type

[*type*]

bind(*thisArg*, **args*, ***kwargs*)

[creates a new function that, when called, has its *this* keyword set to the provided value, with a given sequence of arguments preceding any provided when the new function is called.]

Parameters

- **thisArg** ([*type*]) – [The value to be passed as the *this* parameter to the target]
- **called.**] (*function func when the bound function is*) –

Returns

[A copy of the given function with the specified *this* value, and initial arguments (if provided).]

Return type

[*type*]

call(*thisArg=**None*, **args*, ***kwargs*)

[calls a function with a given *this* value and arguments provided individually.]

Parameters

thisArg ([*type*]) – [description]

Returns

[result of calling the function.]

Return type

[*type*]

toString()

[Returns a string representing the source code of the function. Overrides the]

class domonic.javascript.Global

javascript global methods

NaN()

“Not-a-Number” value

static Number(*x*)

Converts an object’s value to a number

static String(*x*)

Converts an object’s value to a string

static clearTimeout(*timeoutID*)

[cancels a timer set with `setTimeout()`]

Parameters

timeoutID ([*str*]) – [the identifier returned by `setTimeout()`]

static decodeURI(*x*)

Decodes a URI

static decodeURIComponent(*x*)

Decodes a URI component

static encodeURI(*x*)

Encodes a URI

static encodeURIComponent(*x*)

Encodes a URI component

static eval(*pythonstring*)

Evaluates a string and executes it as if it was script code

static isFinite(*x*) → *bool*

Returns true if *x* is a finite number

static isNaN(*x*)

Determines whether a value is an illegal number

static parseFloat(*x: str*)

Parses a string and returns a floating point number

static parseInt(*x: str*)

Parses a string and returns an integer

static require(*path: str*)

Loads a script from a file

static setTimeout(*callback, t, *args, **kwargs*)

[sets a timer which executes a function or evaluates an expression after a specified delay]

Parameters

- **callback** (*function*) – [method to be executed after the delay]
- **t** (*[int]*) – [milliseconds]

Returns

[an identifier for the timer]

Return type

[*str*]

undefined()

Indicates that a variable has not been assigned a value

class domonic.javascript.Job(*interval, execute, *args, **kwargs*)

run()

Method representing the thread's activity.

You may override this method in a subclass. The standard `run()` method invokes the callable object passed to the object's constructor as the target argument, if any, with sequential and keyword arguments taken from the `args` and `kwargs` arguments, respectively.

class domonic.javascript.Map(*collection*)

Map holds key-value pairs and remembers the original insertion order of the keys.

clear()

Removes all key-value pairs from the Map object.

delete(*key: str*) → *bool*

Returns true if an element in the Map object existed and has been removed, or false if the element does not exist. `Map.prototype.has(key)` will return false afterwards.

entries()

Returns a new Iterator object that contains an array of [`key`, `value`] for each element in the Map object in insertion order.

get(key: *str*, default=None)

Returns the value associated to the key, or undefined if there is none.

has(key: *str*) → bool

Returns a boolean asserting whether a value has been associated to the key in the Map object or not.

keys()

Returns a new Iterator object that contains the keys for each element in the Map object in insertion order.

set(key: *str*, value)

Sets the value for the key in the Map object. Returns the Map object.

values()

Returns a new Iterator object that contains the values for each element in the Map object in insertion order.

class domonic.javascript.Math(obj=None, *args, **kwargs)

Math class that mirrors javascript implementation.

i.e. you can pass strings and it will also work, Math.abs('-1')

static hypot(*args)

returns the square root of the sum of squares of its arguments

static log2(*args)

returns the square root of the sum of squares of its arguments

static loglp(*args)

returns the natural logarithm (base e) of 1 + a number, that is

class domonic.javascript.Number(x="", *args, **kwargs)

javascript Number methods

NEGATIVE_INFINITY = inf

Represents negative infinity (returned on overflow) Number

POSITIVE_INFINITY = -inf

Represents infinity (returned on overflow) Number

isInteger()

Checks whether a value is an integer

isSafeInteger()

Checks whether a value is a safe integer

toExponential(num=None)

Converts a number into an exponential notation

toFixed(digits: int)

[formats a number using fixed-point notation.]

Parameters

digits ([int]) – [The number of digits to appear after the decimal point]

Returns

[A string representing the given number using fixed-point notation.]

Return type

[str]

toPrecision(*precision*)

[returns a string representing the Number object to the specified precision.]

Parameters

precision (*[int]*) – [An integer specifying the number of significant digits.]

Returns

[A string representing a Number object in fixed-point or exponential notation rounded to precision significant digits]

Return type

[*str*]

toString(*base: int*)

[returns a string representing the specified Number object.]

Parameters

base (*int*) – [An integer in the range 2 through 36 specifying the base to use for representing numeric values.]

Returns

[a string representing the specified Number object]

Return type

[*str*]

exception domonic.javascript.**ProgramKilled****class** domonic.javascript.**Reflect**

The Reflect object provides the following static functions which have the same names as the proxy handler methods. Some of these methods are also the same as corresponding methods on Object, although they do have some subtle differences between them.

static **apply**(*target, thisArgument, argumentsList*)

Calls a target function with arguments as specified by the argumentsList parameter. See also Function.prototype.apply().

static **construct**(*target, argumentsList, newTarget*)

The new operator as a function. Equivalent to calling new target(...argumentsList). Also provides the option to specify a different prototype.

static **defineProperty**(*target, propertyKey, attributes*)

Similar to Object.defineProperty(). Returns a Boolean that is true if the property was successfully defined.

static **deleteProperty**(*target, propertyKey*)

The delete operator as a function. Equivalent to calling delete target[propertyKey].

static **get**(*target, propertyKey, receiver*)

Returns the value of the property. Works like getting a property from an object (target[propertyKey]) as a function.

static **getOwnPropertyDescriptor**(*target, propertyKey*)

Similar to Object.getOwnPropertyDescriptor(). Returns a property descriptor of the given property if it exists on the object, undefined otherwise.

getPrototypeOf()

Returns the prototype (internal [[Prototype]] property) of the specified object.

static has(*target*, *propertyKey*)

Returns a Boolean indicating whether the target has the property. Either as own or inherited. Works like the in operator as a function.

static ownKeys(*target*)

Returns an array of the target object's own (not inherited) property keys.

static preventExtensions(*target*)

Similar to Object.preventExtensions(). Returns a Boolean that is true if the update was successful.

static set(*target*, *propertyKey*, *value*, *receiver*)

A function that assigns values to properties. Returns a Boolean that is true if the update was successful.

static setPrototypeOf(*target*, *prototype*)

A function that sets the prototype of an object. Returns a Boolean that is true if the update was successful.

class domonic.javascript.Screen

screen

availHeight()

Returns the height of the screen (excluding the Windows Taskbar)

availWidth()

Returns the width of the screen (excluding the Windows Taskbar)

colorDepth()

Returns the colorDepth

height()

Returns the total height of the screen

pixelDepth()

Returns the pixelDepth

width()

Returns the total width of the screen

class domonic.javascript.String(*x*="", **args*, ***kwargs*)

javascript String methods

big()

[wraps the string in big tags]

Returns

[the string in big tags]

Return type

[str]

blink()

[wraps the string in blink tags]

Returns

[the string in blink tags]

Return type

[str]

bold()

[wraps the string in bold tags]

Returns

[the string in bold tags]

Return type

[str]

charAt(*index: int*) → str

[Returns the character at the specified index (position)]

Parameters

index (*int*) – [index position]

Returns

[the character at the specified index. if the index is out of range, an empty string is returned.]

Return type

[str]

charCodeAt(*index: int*) → int

Returns the Unicode of the character at the specified index

codePointAt(*index: int*)

[Returns the Unicode code point at the specified index (position)]

Parameters

index (*int*) – [index position]

Returns

[the Unicode code point at the specified index (position)]

Return type

[type]

compile(*pattern: str*)

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

concat(*args, *separator: str = ""*) → str

[concatenates the string arguments to the calling string and returns a new string.]

Parameters

separator (*str, optional*) – []. Defaults to “”.

Returns

[A new string containing the combined text of the strings provided.]

Return type

[type]

div(*args, **kwargs)

[wraps the string in a div tag]

Returns

[the string in a div tag]

Return type

[str]

endsWith(*x: str, start: Optional[int] = None, end: Optional[int] = None*) → bool

Checks whether a string ends with specified string/characters

fixed()

[wraps the string in fixed tags]

Returns

[the string in fixed tags]

Return type

[str]

fontcolor(*color: str*)

[wraps the string in font tags with a specified color]

Parameters

color (*str*) – [the color to use]

Returns

[the string in font tags]

Return type

[str]

fontsize(*size: str*)

[wraps the string in font tags with a specified size]

Parameters

size (*str*) – [the size to use]

Returns

[the string in font tags]

Return type

[str]

fromCharCode(**codes*) → str

returns a string created from the specified sequence of UTF-16 code units

static fromCodePoint(*codePoint: int*)

Converts a Unicode code point into a string

includes(*searchValue: str, position: int = 0*) → bool

[returns true if the specified string is found within the calling String object,]

Parameters

- **searchValue** (*str*) – [The string value to search for.]
- **position** (*int, optional*) – [the position to search from]. Defaults to 0.

Returns

[a boolean value indicating whether the search value was found.]

Return type

[type]

indexOf(*searchValue: str, fromIndex: int = 0*)

[returns the index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex]

Parameters

- **searchValue** (*str*) – [The string value to search for.]
- **fromIndex** (*int*) – [An integer representing the index at which to start the search]

Returns

[The index of the first occurrence of searchValue, or -1 if not found.]

Return type

[*type*]

italics()

[wraps the string in italics tags]

Returns

[the string in italics tags]

Return type

[*str*]

lastIndexOf(*searchValue: str, fromIndex: int = 0*)

returns the last index within the calling String object of the first occurrence of the specified value, starting the search at fromIndex

link(*url: str*)

[wraps the string in a link tag]

Parameters

url (*str*) – [the url to use]

Returns

[the string in a link tag]

Return type

[*str*]

localeCompare(*comparisonString: str, locale: Optional[str] = None, *args*) → *int*

method returns a number indicating whether a reference string comes before, or after, or is the same as the given string in sort order

match(*pattern: str*)

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

matchAll(*pattern: str*)

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

padEnd(*length: int, padChar: str = ' ')* → *str*

[Pads the end of a string with a specified character (repeated, if needed) to create a new string.]

Parameters

- **length** (*int*) – [the length of the resulting string]
- **padChar** (*str, optional*) – [the character to use for padding. Defaults to " "].

Returns

[the padded string]

Return type

[*str*]

padStart(*length*: *int*, *padChar*: *str* = ' ') → *str*

[Pads the start of a string with a specified character]

Parameters

- **length** (*int*) – [the length of the resulting string]
- **padChar** (*str*, *optional*) – [the character to use for padding. Defaults to " "].

Returns

[the padded string]

Return type

[*str*]

static raw(*string*)

Returns the string as-is

repeat(*count*: *int*) → *str*

Returns a new string with a specified number of copies of an existing string

replace(*old*: *str*, *new*) → *str*

Searches a string for a specified value, or a regular expression, and returns a new string where the specified values are replaced. only replaces first one.

replaceAll(*old*: *str*, *new*: *str*)

[returns a new string where the specified values are replaced. ES2021]

Parameters

- **old** ([*str*]) – [word to remove]
- **new** ([*str*]) – [word to replace it with]

Returns

[new string with all occurrences of old word replaced]

Return type

[*str*]

search(*searchValue*: *str*, *position*: *int* = 0) → *bool*

[returns true if the specified string is found within the calling String object,] starting at the specified position.
:param searchValue: [The string value to search for.] :type searchValue: *str* :param position: [the position to search from]. Defaults to 0. :type position: *int*, optional

Returns

[a boolean value indicating whether the search value was found.]

Return type

[*type*]

slice(*start*: *int* = 0, *end*: *Optional[int]* = *None*) → *str*

Selects a part of an string, and returns the new string

small()

[wraps the string in small tags]

Returns

[the string in small tags]

Return type

[*str*]

split(*expr*) → *list*

[can split a string based on a regex]

Parameters

expr (*[str]*) – [valid regex or string to split on]

Returns

[list of str]

Return type

[*list*]

startsWith(*x: str, start: Optional[int] = None, end: Optional[int] = None*) → *bool*

Checks whether a string begins with specified characters

strike()

[wraps the string in strike tags]

Returns

[the string in strike tags]

Return type

[*str*]

sub()

[wraps the string in sub tags]

Returns

[the string in sub tags]

Return type

[*str*]

substr(*start: int = 0, end: Optional[int] = None*)

Extracts the characters from a string, beginning at a specified start position, and through the specified number of character

substring(*start: int, end: Optional[int] = None*) → *str*

Extracts the characters from a string, between two specified indices

sup()

[wraps the string in sup tags]

Returns

[the string in sup tags]

Return type

[*str*]

static toCharCode(*char: str*)

Converts a Unicode string into a code point

static toCodePoint(*char: str*)

Converts a Unicode string into a code point

toLocaleLowerCase() → *str*

Converts a string to lowercase letters, according to the host's locale

toLocaleUpperCase() → *str*

Converts a string to uppercase letters, according to the host's locale

toLowerCase() → *str*

Converts a string to lowercase letters

toUpperCase() → *str*

Converts a string to uppercase letters

trim()

Removes whitespace from both ends of a string

trimEnd(*length: int*)

[Removes whitespace from the end of a string]

Parameters

length (*int*) – [the length of the resulting string]

Returns

[the trimmed string]

Return type

[*type*]

trimStart(*length: int*)

[Removes whitespace from the beginning of a string.]

Parameters

length (*int*) – [the length of the resulting string]

Returns

[the trimmed string]

Return type

[*str*]

webpage()

[wraps the string in a webpage]

Returns

[the string as a webpage]

Return type

[*str*]

class domonic.javascript.**Window**(*args, **kwargs)

window

static alert(*msg*)

Displays an alert box with a message and an OK button

static atob(*dataString*)

Decodes a base-64 encoded string

static btoa(*dataString*)

Encodes a string in base-64

clearTimeout()

[cancels a timer set with setTimeout()]

Parameters

timeoutID ([*str*]) – [the identifier returned by setTimeout()]

static prompt(*msg*, *default_text*="")

Displays a dialog box that prompts the visitor for input

static requestAnimationFrame(*callback*)

[requests a frame of an animation]

Parameters

callback (*callable*) – [the callback function]

Returns

[description]

Return type

[*type*]

setTimeout(*t*, **args*, ***kwargs*)

[sets a timer which executes a function or evaluates an expression after a specified delay]

Parameters

- **callback** (*function*) – [method to be executed after the delay]
- **t** ([*int*]) – [milliseconds]

Returns

[an identifier for the timer]

Return type

[*str*]

class domonic.javascript.Worker(*script*)

[A background task that can be created via script, which can send messages back to its creator. Creating a worker is done by calling the Worker(“path/to/worker/script”) constructor.] TODO - JSWorker - Node :param object: [takes a path to a python script] :type object: [*str*]

postMessage()

Sends a message — consisting of any object — to the worker’s inner scope.

terminate()

Immediately terminates the worker. This does not let worker finish its operations; it is halted at once. ServiceWorker instances do not support this method.

domonic.javascript.as_signed(*value*, *bits*)

Converts an unsigned integer to a signed integer.

domonic.javascript.clearTimeout(*timeoutID*)

[cancels a timer set with setTimeout()]

Parameters

timeoutID ([*str*]) – [the identifier returned by setTimeout()]

domonic.javascript.decodeURI(*x*)

Decodes a URI

domonic.javascript.decodeURIComponent(*x*)

Decodes a URI component

domonic.javascript.encodeURI(*x*)

Encodes a URI

`domonic.javascript.encodeURIComponent(x)`

Encodes a URI component

`domonic.javascript.function(python_str: str) → str`

[evals a string i.e.

`sup = function("print(hi)") sup()`

]

Parameters

`python_str ([str])` – [some valid python code as a string]

`domonic.javascript.parseFloat(x: str)`

Parses a string and returns a floating point number

`domonic.javascript.parseInt(x: str)`

Parses a string and returns an integer

`domonic.javascript.setTimeout(callback, t, *args, **kwargs)`

[sets a timer which executes a function or evaluates an expression after a specified delay]

Parameters

- `callback (function)` – [method to be executed after the delay]
- `t ([int])` – [milliseconds]

Returns

[an identifier for the timer]

Return type

[str]

`domonic.javascript.window`

alias of [Window](#)

domonic.terminal

- call command line functions in python 3

exception `domonic.terminal.TerminalException(error, message='An error message was recieved from terminal')`

raised if the terminal throws an exception

`class domonic.terminal.alias(*args, **kwargs)`

`class domonic.terminal.apt(*args, **kwargs)`

`class domonic.terminal.ar(*args, **kwargs)`

`class domonic.terminal.asa(*args, **kwargs)`

`class domonic.terminal.at(*args, **kwargs)`

`class domonic.terminal.awk(*args, **kwargs)`

`class domonic.terminal.banner(*args, **kwargs)`

`class domonic.terminal.basename(*args, **kwargs)`

```
class domonic.terminal.bash(*args, **kwargs)
```

```
class domonic.terminal.batch(*args, **kwargs)
```

```
class domonic.terminal.bc(*args, **kwargs)
```

```
class domonic.terminal.bg(*args, **kwargs)
```

```
class domonic.terminal.bind(*args, **kwargs)
```

```
class domonic.terminal.builtin(*args, **kwargs)
```

```
class domonic.terminal.cal(*args, **kwargs)
```

```
class domonic.terminal.caller(*args, **kwargs)
```

```
class domonic.terminal.cat(*args, **kwargs)
```

```
class domonic.terminal.cc(*args, **kwargs)
```

```
class domonic.terminal.cd(*args, **kwargs)
```

NOTE - 'cd' does not run on terminal - cd is pointless as session opens and closes - so is overridden to change dirs via pure python

```
class domonic.terminal.cflow(*args, **kwargs)
```

```
class domonic.terminal.chgrp(*args, **kwargs)
```

```
class domonic.terminal.chmod(*args, **kwargs)
```

```
class domonic.terminal.chown(*args, **kwargs)
```

```
class domonic.terminal.cksum(*args, **kwargs)
```

```
class domonic.terminal.comb(*args, **kwargs)
```

```
class domonic.terminal.command(*args, **kwargs)
```

wrapper class for all terminal commands

```
static run(cmd: str)
```

runs any command on the terminal

Parameters

cmd (*str*) – The command you want to run on the terminal

Returns

the response as a string

Return type

str

```
class domonic.terminal.compgen(*args, **kwargs)
```

```
class domonic.terminal.complete(*args, **kwargs)
```

```
class domonic.terminal.compopt(*args, **kwargs)
```

```
class domonic.terminal.compress(*args, **kwargs)
```

```
class domonic.terminal.convert(*args, **kwargs)
```

```
class domonic.terminal.cowsay(*args, **kwargs)
class domonic.terminal.cp(*args, **kwargs)
class domonic.terminal.cron(*args, **kwargs)
class domonic.terminal.crontab(*args, **kwargs)
class domonic.terminal.csplit(*args, **kwargs)
class domonic.terminal.ctags(*args, **kwargs)
class domonic.terminal.curl(*args, **kwargs)
class domonic.terminal.cut(*args, **kwargs)
class domonic.terminal.cxref(*args, **kwargs)
class domonic.terminal.date(*args, **kwargs)
class domonic.terminal.dd(*args, **kwargs)
class domonic.terminal.declare(*args, **kwargs)
class domonic.terminal.delta(*args, **kwargs)
class domonic.terminal.df(*args, **kwargs)
class domonic.terminal.diff(*args, **kwargs)
class domonic.terminal.dirname(*args, **kwargs)
class domonic.terminal.dirs(*args, **kwargs)
class domonic.terminal.disown(*args, **kwargs)
class domonic.terminal.du(*args, **kwargs)
class domonic.terminal.echo(*args, **kwargs)
class domonic.terminal.ed(*args, **kwargs)
class domonic.terminal.enable(*args, **kwargs)
class domonic.terminal.env(*args, **kwargs)
class domonic.terminal.ex(*args, **kwargs)
class domonic.terminal.exit(*args, **kwargs)
class domonic.terminal.expand(*args, **kwargs)
class domonic.terminal.export(*args, **kwargs)
class domonic.terminal.expr(*args, **kwargs)
class domonic.terminal.fc(*args, **kwargs)
class domonic.terminal.ffmpeg(*args, **kwargs)
class domonic.terminal.fg(*args, **kwargs)
```

```
class domonic.terminal.figlet(*args, **kwargs)
class domonic.terminal.file(*args, **kwargs)
class domonic.terminal.find(*args, **kwargs)
class domonic.terminal.finger(*args, **kwargs)
class domonic.terminal.fold(*args, **kwargs)
class domonic.terminal.fort77(*args, **kwargs)
class domonic.terminal.fuser(*args, **kwargs)
class domonic.terminal.gcc(*args, **kwargs)
class domonic.terminal.gencat(*args, **kwargs)
class domonic.terminal.get(*args, **kwargs)
class domonic.terminal.getconf(*args, **kwargs)
class domonic.terminal.getopts(*args, **kwargs)
class domonic.terminal.git(*args, **kwargs)
class domonic.terminal.grep(*args, **kwargs)
class domonic.terminal.groupadd(*args, **kwargs)
class domonic.terminal.groupdel(*args, **kwargs)
class domonic.terminal.groups(*args, **kwargs)
class domonic.terminal.gunzip(*args, **kwargs)
class domonic.terminal.gzip(*args, **kwargs)
class domonic.terminal.head(*args, **kwargs)
class domonic.terminal.history(*args, **kwargs)
class domonic.terminal.iconv(*args, **kwargs)
class domonic.terminal.ifconfig(*args, **kwargs)
class domonic.terminal.ipconfig(*args, **kwargs)
class domonic.terminal.ipcrm(*args, **kwargs)
class domonic.terminal.ipcs(*args, **kwargs)
class domonic.terminal.jobs(*args, **kwargs)
class domonic.terminal.join(*args, **kwargs)
class domonic.terminal.jq(*args, **kwargs)
class domonic.terminal.kill(*args, **kwargs)
class domonic.terminal.killall(*args, **kwargs)
```

```
class domonic.terminal.less(*args, **kwargs)
class domonic.terminal.let(*args, **kwargs)
class domonic.terminal.lex(*args, **kwargs)
class domonic.terminal.link(*args, **kwargs)
class domonic.terminal.ln(*args, **kwargs)
class domonic.terminal.local(*args, **kwargs)
class domonic.terminal.locale(*args, **kwargs)
class domonic.terminal.localedef(*args, **kwargs)
class domonic.terminal.logger(*args, **kwargs)
class domonic.terminal.logname(*args, **kwargs)
class domonic.terminal.logout(*args, **kwargs)
class domonic.terminal.lp(*args, **kwargs)
class domonic.terminal.ls(*args, **kwargs)
class domonic.terminal.m4(*args, **kwargs)
class domonic.terminal.mailx(*args, **kwargs)
class domonic.terminal.make(*args, **kwargs)
class domonic.terminal.man(*args, **kwargs)
class domonic.terminal.mapfile(*args, **kwargs)
class domonic.terminal.mesg(*args, **kwargs)
class domonic.terminal.mkdir(*args, **kwargs)
class domonic.terminal.mkfifo(*args, **kwargs)
class domonic.terminal.mkfile(*args, **kwargs)
class domonic.terminal.more(*args, **kwargs)
class domonic.terminal.mv(*args, **kwargs)
class domonic.terminal.nautilus(*args, **kwargs)
class domonic.terminal.newgrp(*args, **kwargs)
class domonic.terminal.nice(*args, **kwargs)
class domonic.terminal.nl(*args, **kwargs)
class domonic.terminal.nm(*args, **kwargs)
class domonic.terminal.nmap(*args, **kwargs)
class domonic.terminal.nohup(*args, **kwargs)
```

```
class domonic.terminal.npm(*args, **kwargs)
class domonic.terminal.od(*args, **kwargs)
class domonic.terminal.passwd(*args, **kwargs)
class domonic.terminal.paste(*args, **kwargs)
class domonic.terminal.patch(*args, **kwargs)
class domonic.terminal.pathchk(*args, **kwargs)
class domonic.terminal.pax(*args, **kwargs)
class domonic.terminal.ping(*args, **kwargs)
class domonic.terminal.pip(*args, **kwargs)
class domonic.terminal.popd(*args, **kwargs)
class domonic.terminal.pr(*args, **kwargs)
class domonic.terminal.printf(*args, **kwargs)
class domonic.terminal.prs(*args, **kwargs)
class domonic.terminal.ps(*args, **kwargs)
class domonic.terminal.pushd(*args, **kwargs)
class domonic.terminal.pwd(*args, **kwargs)
class domonic.terminal.python(*args, **kwargs)
class domonic.terminal.qalter(*args, **kwargs)
class domonic.terminal.qdel(*args, **kwargs)
class domonic.terminal.qhold(*args, **kwargs)
class domonic.terminal.qmove(*args, **kwargs)
class domonic.terminal.qmsg(*args, **kwargs)
class domonic.terminal.qrerun(*args, **kwargs)
class domonic.terminal.qrls(*args, **kwargs)
class domonic.terminal.qselect(*args, **kwargs)
class domonic.terminal.qsig(*args, **kwargs)
class domonic.terminal.qstat(*args, **kwargs)
class domonic.terminal.qsub(*args, **kwargs)
class domonic.terminal.read(*args, **kwargs)
class domonic.terminal.readarray(*args, **kwargs)
class domonic.terminal.readonly(*args, **kwargs)
```

```
class domonic.terminal.reboot(*args, **kwargs)
class domonic.terminal.renice(*args, **kwargs)
class domonic.terminal.rm(*args, **kwargs)
class domonic.terminal.rmdel(*args, **kwargs)
class domonic.terminal.rmdir(*args, **kwargs)
class domonic.terminal.rsync(*args, **kwargs)
class domonic.terminal.sact(*args, **kwargs)
class domonic.terminal.say(*args, **kwargs)
class domonic.terminal.sccs(*args, **kwargs)
class domonic.terminal.scp(*args, **kwargs)
class domonic.terminal.sed(*args, **kwargs)
class domonic.terminal.sh(*args, **kwargs)
class domonic.terminal.shift(*args, **kwargs)
class domonic.terminal.shopt(*args, **kwargs)
class domonic.terminal.shutdown(*args, **kwargs)
class domonic.terminal.sleep(*args, **kwargs)
class domonic.terminal.sort(*args, **kwargs)
class domonic.terminal.source(*args, **kwargs)
class domonic.terminal.split(*args, **kwargs)
class domonic.terminal.ssh(*args, **kwargs)
class domonic.terminal.strings(*args, **kwargs)
class domonic.terminal.strip(*args, **kwargs)
class domonic.terminal.stty(*args, **kwargs)
class domonic.terminal.suspend(*args, **kwargs)
class domonic.terminal.tabs(*args, **kwargs)
class domonic.terminal.tail(*args, **kwargs)
class domonic.terminal.talk(*args, **kwargs)
class domonic.terminal.tar(*args, **kwargs)
class domonic.terminal.tee(*args, **kwargs)
class domonic.terminal.test(*args, **kwargs)
class domonic.terminal.time(*args, **kwargs)
```

```
class domonic.terminal.times(*args, **kwargs)
class domonic.terminal.touch(*args, **kwargs)
class domonic.terminal.tput(*args, **kwargs)
class domonic.terminal.tr(*args, **kwargs)
class domonic.terminal.trap(*args, **kwargs)
class domonic.terminal.true(*args, **kwargs)
class domonic.terminal.tsort(*args, **kwargs)
class domonic.terminal.tty(*args, **kwargs)
class domonic.terminal.typeset(*args, **kwargs)
class domonic.terminal.ulimit(*args, **kwargs)
class domonic.terminal.umask(*args, **kwargs)
class domonic.terminal.unalias(*args, **kwargs)
class domonic.terminal.uname(*args, **kwargs)
class domonic.terminal.uncompress(*args, **kwargs)
class domonic.terminal.unexpand(*args, **kwargs)
class domonic.terminal.unget(*args, **kwargs)
class domonic.terminal.uniq(*args, **kwargs)
class domonic.terminal.unlink(*args, **kwargs)
class domonic.terminal.unset(*args, **kwargs)
class domonic.terminal.uptime(*args, **kwargs)
class domonic.terminal.useradd(*args, **kwargs)
class domonic.terminal.userdel(*args, **kwargs)
class domonic.terminal.users(*args, **kwargs)
class domonic.terminal.uucp(*args, **kwargs)
class domonic.terminal.uudecode(*args, **kwargs)
class domonic.terminal.uuencode(*args, **kwargs)
class domonic.terminal.uustat(*args, **kwargs)
class domonic.terminal.uux(*args, **kwargs)
class domonic.terminal.val(*args, **kwargs)
class domonic.terminal.wait(*args, **kwargs)
class domonic.terminal.wc(*args, **kwargs)
```

```
class domonic.terminal.wget(*args, **kwargs)
class domonic.terminal.what(*args, **kwargs)
class domonic.terminal.who(*args, **kwargs)
class domonic.terminal.whoami(*args, **kwargs)
class domonic.terminal.write(*args, **kwargs)
class domonic.terminal.xargs(*args, **kwargs)
class domonic.terminal.yacc(*args, **kwargs)
class domonic.terminal.zcat(*args, **kwargs)
```

domonic.JSON

domonic.JSON.csv2json(csv_filepath, json_filepath=None)
convert a CSV to JSON.

domonic.JSON.csvify(arr, outfile='data.csv')
takes a json array and dumps a csv file

Parameters

- **arr** (*list*) – the json array
- **outfile** (*list*) – the output file

Returns

a csv file

Return type

str

domonic.JSON.flatten(b, delim='__')
i.e. input = map(lambda x: JSON.flatten(x, “__”), input)

domonic.JSON.parse(json_string: str)
[take a json string and return a python object]

Parameters

json_string (*str*) – [a json string]

Returns

[a python object]

Return type

[type]

domonic.JSON.parse_file(filepath: str)
[loads a json file and returns a python object]

Parameters

filepath (*str*) – [path to json file]

Returns

[a python object]

Return type`[type]``domonic.JSON.stringify(data, filepath: Optional[str] = None, **kwargs)`

[stringify a python object]

Parameters

- **data** (`[type]`) – [the python object]
- **filepath** (`str`, *optional*) – [an optional filepath to save the stringified object] [default: None]

Returns

[the stringified object]

Return type`[type]``domonic.JSON.tablify(arr)`

takes a json array and returns a html table # TODO - reverse. table to json

Parameters**arr** (`list`) – the json array**Returns**

a html table

Return type`str`**domonic.CDN**

For quick reference when prototyping you can use the CDN package. (Don't rely on a CDN package for production code. wget a local copy.)

TODO - integrity/cross origin/module?

```
class domonic.CDN.CDN_CSS
```

Preferably use version numbers if available. use LATEST if it always gets the latest

```
BALLOON: str = 'https://unpkg.com/balloon-css/balloon.min.css'
```

```
BOOTSTRAP_4: str =
```

```
'https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css'
```

```
BOOTSTRAP_5_ALPHA: str =
```

```
'https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js'
```

```
FONTAWESOME_5_7_1: str = 'https://use.fontawesome.com/releases/v5.7.1/css/all.css'
```

```
MARX: str = 'https://unpkg.com/marx-css/css/marx.min.css'
```

```
MDI_5_4_55: str =
```

```
'https://cdn.materialdesignicons.com/5.4.55/css/materialdesignicons.min.css'
```

```
MILLIGRAM_1_3_0: str =
```

```
'https://cdnjs.cloudflare.com/ajax/libs/milligram/1.3.0/milligram.css'
```

```
MVP: str = 'https://unpkg.com/mvp.css'
```

```
SIMPLE: str = 'https://cdn.simplecss.org/simple.min.css'
```

```
TAILWIND_2_2_15: str =  
'https://unpkg.com/tailwindcss@^2.2.15/dist/tailwind.min.css'
```

```
THREE_DOTS_0_2_0: str =  
'https://cdnjs.cloudflare.com/ajax/libs/three-dots/0.2.0/three-dots.min.css'
```

```
WATER_LATEST: str =  
'https://cdn.jsdelivr.net/gh/kognise/water.css@latest/water.min.css'
```

```
X3DOM: str = 'https://www.x3dom.org/download/x3dom.css'
```

```
class domonic.CDN.CDN_IMG
```

```
    CDN images
```

```
    static PLACEHOLDER(width: int = 100, height: int = 100, HTTP: str = "", seperator: str = '/') → str  
        to update do CDN_IMG.PLACEHOLDER_SERVICE = "placebear.com/g" usage :  
        img(_src=CDN_IMG.PLACEHOLDER(300,100)) default HTTP is none, to let the browser decide # use  
        optional seperator if the site uses x instead of slash img(_src=CDN_IMG.PLACEHOLDER(300,100,'x'))
```

```
class domonic.CDN.CDN_JS
```

```
    js libs
```

```
    AFRAME_1_2: str = 'https://aframe.io/releases/1.2.0/aframe.min.js'
```

```
    BOOTSTRAP_4: str =  
'https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js'
```

```
    BOOTSTRAP_5 =  
'https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js'  
        latest
```

```
    BOOTSTRAP_5_ALPHA: str =  
'https://stackpath.bootstrapcdn.com/bootstrap/5.0.0-alpha1/js/bootstrap.min.js'
```

```
    BRYTHON_3_9_5: str =  
'https://cdnjs.cloudflare.com/ajax/libs/brython/3.9.5/brython.min.js'
```

```
    D3: str = 'https://cdnjs.cloudflare.com/ajax/libs/d3/7.4.3/d3.min.js'  
        latest
```

```
    D3_6_1_0: str = 'https://cdnjs.cloudflare.com/ajax/libs/d3/6.1.0/d3.min.js'
```

```
    HTMX: str = 'https://unpkg.com/htmx.org@1.7.0'  
        latest
```

```
    JQUERY: str = 'https://code.jquery.com/jquery-3.6.0.min.js'  
        latest
```

```
    JQUERY_3_5_1: str = 'https://code.jquery.com/jquery-3.5.1.min.js'
```

```
    JQUERY_UI: str = 'https://code.jquery.com/ui/1.12.0/jquery-ui.min.js'
```

```
    MATHML: str = 'https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.0/MathJax.js?  
config=MML_HTMLorMML'
```

```

MODERNIZER_2_8_3: str =
'https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.8.3/modernizr.min.js'

MOMENT_2_27_0: str =
'https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.27.0/moment.min.js'

PIXI_5_3_3: str =
'https://cdnjs.cloudflare.com/ajax/libs/pixi.js/5.3.3/pixi.min.js'

POPPER_1_16_1: str =
'https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js'

SOCKET_1_4_5: str =
'https://cdnjs.cloudflare.com/ajax/libs/socket.io/1.4.5/socket.io.min.js'

UNDERScore: str = 'https://cdn.jsdelivr.net/npm/underscore@1.11.0/underscore-min.js'

X3DOM: str = 'https://www.x3dom.org/download/x3dom.js'

```

domonic.events

<https://developer.mozilla.org/en-US/docs/Web/Events>

```
class domonic.events.AnimationEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    ANIMATIONEND: str = 'animationend'
```

```
    ANIMATIONITERATION: str = 'animationiteration'
```

```
    ANIMATIONSTART: str = 'animationstart'
```

```
    animationName
```

Returns the name of the animation

```
    elapsedTime
```

Returns the number of seconds an animation has been running

```
    pseudoElement
```

Returns the name of the pseudo-element of the animation

```
class domonic.events.BeforeUnloadEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    BEFOREUNLOAD: str = 'beforeunload'
```

BeforeUnloadEvent

```
class domonic.events.ClipboardEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    COPY: str = 'copy'
```

```
    CUT: str = 'cut'
```

```
    PASTE: str = 'paste'
```

```
    clipboardData
```

Returns an object containing the data affected by the clipboard operation

```
class domonic.events.CompositionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
class domonic.events.CustomEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

class domonic.events.DOMContentLoadedEvent(_type: str, options: Optional[dict] = None, *args,
                                           **kwargs)

    DOMCONTENTLOADED: str = 'DOMContentLoaded'

    document
        Returns the document that was loaded

class domonic.events.DragEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    DRAG: str = 'drag'

    DROP: str = 'drop'

    END: str = 'dragend'

    ENTER: str = 'dragenter'

    EXIT: str = 'dragexit'

    LEAVE: str = 'dragleave'

    OVER: str = 'dragover'

    START: str = 'dragstart'

    dataTransfer
        Returns the data that is dragged/dropped

class domonic.events.ErrorEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ERROR: str = 'error'

class domonic.events.Event(_type: str, options: Optional[dict] = None, *args, **kwargs)

    event

    ABORT: str = 'abort'

    AFTERPRINT: str = 'afterprint'

    BEFOREPRINT: str = 'beforeprint'

    BEFOREUNLOAD: str = 'beforeunload'

    CANPLAY: str = 'canplay'

    CANPLAYTHROUGH: str = 'canplaythrough'

    CHANGE: str = 'change'

    DURATIONCHANGE: str = 'durationchange'

    EMPTIED: str = 'emptied'

    ENDED: str = 'ended'

    ERROR: str = 'error'
```



```
FULLSCREENCHANGE: str = 'fullscreenchange'
FULLSCREENERROR: str = 'fullscreenerror'
INPUT: str = 'input'
INVALID: str = 'invalid'
LOAD: str = 'load'
LOADEDATA: str = 'loadeddata'
LOADEDMETADATA: str = 'loadedmetadata'
MESSAGE: str = 'message'
OFFLINE: str = 'offline'
ONLINE: str = 'online'
OPEN: str = 'open'
PAUSE: str = 'pause'
PLAY: str = 'play'
PLAYING: str = 'playing'
PROGRESS: str = 'progress'
RATECHANGE: str = 'ratechange'
READYSTATECHANGE: str = 'readystatechange'
RESET: str = 'reset'
RESIZE: str = 'resize'
SCROLL: str = 'scroll'
SEARCH: str = 'search'
SEEKED: str = 'seeked'
SEEKING: str = 'seeking'
SELECT: str = 'select'
SHOW: str = 'show'
STALLED: str = 'stalled'
SUBMIT: str = 'submit'
SUSPEND: str = 'suspend'
TOGGLE: str = 'toggle'
UNLOAD: str = 'unload'
VOLUMECHANGE: str = 'volumechange'
```

WAITING: `str` = 'waiting'

stopPropagation()

[prevents further propagation of the current event in the capturing and bubbling phases]

`domonic.events.EventDispatcher`

legacy alias

class `domonic.events.EventTarget(*args, **kwargs)`

EventTarget is a class you can extend to give your obj event dispatching abilities

class `domonic.events.ExtendableEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

extendable

Returns whether the event is extendable or not

timeStamp: `float`

Returns the time stamp of the event

class `domonic.events.FetchEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

FETCH: `str` = 'fetch'

clientId

Returns the client ID of the fetch request

property isReload

Returns whether the request is a reload or not

request

Returns the request object

respondWith(response)

Returns a promise that resolves to the response object

waitUntil(promise)

Returns a promise that resolves when the response is available

class `domonic.events.FocusEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

BLUR: `str` = 'blur'

FOCUS: `str` = 'focus'

FOCUSIN: `str` = 'focusin'

FOCUSOUT: `str` = 'focusout'

class `domonic.events.GamePadEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

START: `str` = 'gamepadconnected'

STOP: `str` = 'gamepaddisconnected'

class `domonic.events.HashChangeEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

CHANGE: `str` = 'hashchange'

class `domonic.events.InputEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)`

CHANGE: `str` = 'change'

INPUT: `str` = 'input'

SELECT: `str` = 'select'

data

Returns the inserted characters

dataTransfer

Returns an object containing information about the inserted/deleted data

getTargetRanges()

Returns an array containing target ranges that will be affected by the insertion/deletion

inputType

Returns the type of the change (i.e “inserting” or “deleting”)

isComposing

Returns whether the state of the event is composing or not

class domonic.events.KeyboardEvent(*_type: str, options: Optional[dict] = None, *args, **kwargs*)

keyboard events

DOM_KEY_LOCATION_JOYSTICK: `int` = 5

DOM_KEY_LOCATION_LEFT: `int` = 0

DOM_KEY_LOCATION_MOBILE: `int` = 4

DOM_KEY_LOCATION_NUMPAD: `int` = 3

DOM_KEY_LOCATION_RIGHT: `int` = 2

DOM_KEY_LOCATION_STANDARD: `int` = 1

KEYDOWN: `str` = 'keydown'

KEYPRESS: `str` = 'keypress'

KEYUP: `str` = 'keyup'

class domonic.events.MessageEvent(*_type, options: Optional[dict] = None, *args, **kwargs*)

CONNECT: `str` = 'connect'

DISCONNECT: `str` = 'disconnect'

MESSAGE: `str` = 'message'

data

Returns the data of the message

lastEventId

Returns the last event id of the message

origin

Returns the origin of the message

ports

Returns the ports of the message

source

Returns the source of the message

```
class domonic.events.MouseEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

mouse events

CLICK: str = 'click'

CONTEXTMENU: str = 'contextmenu'

DBLCLICK: str = 'dblclick'

MOUSEDOWN: str = 'mousedown'

MOUSEENTER: str = 'mouseenter'

MOUSELEAVE: str = 'mouseleave'

MOUSEMOVE: str = 'mousemove'

MOUSEOUT: str = 'mouseout'

MOUSEOVER: str = 'mouseover'

MOUSEUP: str = 'mouseup'

getModifierState()

Returns an array containing target ranges that will be affected by the insertion/deletion

```
class domonic.events.PageTransitionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

PAGEHIDE: str = 'pagehide'

PAGESHOW: str = 'pageshow'

persisted

Returns whether the webpage was cached by the browser

```
class domonic.events.PointerEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

POINTER: str = 'pointer'

POINTERCANCEL: str = 'pointercancel'

POINTERDOWN: str = 'pointerdown'

POINTERENTER: str = 'pointerenter'

POINTERLEAVE: str = 'pointerleave'

POINTERMOVE: str = 'pointermove'

POINTEROUT: str = 'pointerout'

POINTEROVER: str = 'pointerover'

POINTERUP: str = 'pointerup'

```

class domonic.events.PopStateEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    POPSTATE: str = 'popstate'

    state
        Returns an object containing a copy of the history entries

class domonic.events.ProgressEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ABORT: str = 'abort'

    ERROR: str = 'error'

    LOAD: str = 'load'

    LOADED: str = 'loaded'

    LOADEND: str = 'loadend'

    LOADSTART: str = 'loadstart'

    PROGRESS: str = 'progress'

    TIMEOUT: str = 'timeout'

class domonic.events.PromiseRejectionEvent(_type, options=None, *args, **kwargs)

    HANDLED: str = 'rejectionhandled'

    UNHANDLED: str = 'unhandledrejection'

    isRejected
        Returns whether the promise was rejected or not

    promise
        Returns the promise that was rejected

    reason
        Returns the reason of the rejection

class domonic.events.SVGEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)

    ABORT: str = 'abort'

    ERROR: str = 'error'

    LOAD: str = 'load'

    LOADEDATA: str = 'loadeddata'

    LOADEDMETADATA: str = 'loadedmetadata'

    LOADSTART: str = 'loadstart'

    PROGRESS: str = 'progress'

    SCROLL: str = 'scroll'

    UNLOAD: str = 'unload'

```

```
class domonic.events.SecurityPolicyViolationEvent(_type: str, options: Optional[dict] = None, *args,
                                                  **kwargs)
```

```
    SECURITY_POLICY_VIOLATION: str = 'securitypolicyviolation'
```

```
    blockedURI
```

```
        Returns the blocked URI
```

```
    frame
```

```
        Returns the frame that violated the policy
```

```
    isFrameAncestor
```

```
        Returns whether the frame is an ancestor of the frame that violated the policy
```

```
    isMainFrame
```

```
        Returns whether the frame is the main frame
```

```
    originalPolicy
```

```
        Returns the original policy
```

```
    violatedDirective
```

```
        Returns the violated directive
```

```
class domonic.events.StorageEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    STORAGE: str = 'storage'
```

```
    key
```

```
        Returns the key of the changed storage item
```

```
    newValue
```

```
        Returns the new value of the changed storage item
```

```
    oldValue
```

```
        Returns the old value of the changed storage item
```

```
    storageArea
```

```
        Returns an object representing the affected storage object
```

```
    url
```

```
        Returns the URL of the changed item's document
```

```
class domonic.events.SubmitEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    SUBMIT: str = 'submit'
```

```
class domonic.events.SyncEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    SYNC: str = 'sync'
```

```
    lastChance
```

```
        Returns whether the sync event is the last chance or not
```

```
    tag
```

```
        Returns the tag of the sync event
```

```
class domonic.events.TimerEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
```

```
    TIMER: str = 'timer'
```

```

    TIMER_COMPLETE: str = 'timercomplete'
    TimerEvent
class domonic.events.TouchEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
    TOUCHCANCEL: str = 'touchcancel'
    TOUCHEND: str = 'touchend'
    TOUCHMOVE: str = 'touchmove'
    TOUCHSTART: str = 'touchstart'
class domonic.events.TransitionEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
    TRANSITIONEND: str = 'transitionend'
    elapsedTime
        Returns the number of seconds a transition has been running
    propertyName
        Returns the name of the transition
    pseudoElement
        Returns the name of the pseudo-element of the transition
class domonic.events.TweenEvent(_type, source=None, bubbles=False, cancelable=False)
    COMPLETE: str = 'onComplete'
    PAUSE: str = 'onPause'
    RESET: str = 'onReset'
    START: str = 'onStart'
    STOP: str = 'onStop'
    TIMER: str = 'onTimer'
    UNPAUSE: str = 'onUnPause'
    UPDATE_END: str = 'onUpdateEnd'
    UPDATE_START: str = 'onUpdateStart'
class domonic.events.UIEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
class domonic.events.WheelEvent(_type: str, options: Optional[dict] = None, *args, **kwargs)
    WHEEL: str = 'wheel'

```

domonic.style

TODO - should this be moved to the webapi in a future revision?

class domonic.style.CSSColorProfileRule

The CSSColorProfileRule interface represents an @color-profile rule.

class domonic.style.CSSConditionRule

The CSSConditionRule interface represents a condition rule.

class domonic.style.CSSCounterStyleRule

The CSSCounterStyleRule interface represents a @counter-style rule.

class domonic.style.CSSDocumentRule

The CSSDocumentRule interface represents an @document rule.

class domonic.style.CSSFontFaceRule

The CSSFontFaceRule interface represents a @font-face rule in a CSS style sheet.

class domonic.style.CSSFontFeatureValuesRule

The CSSFontFeatureValuesRule interface represents a @font-feature-values rule.

class domonic.style.CSSGroupingRule

The CSSGroupingRule interface represents a @grouping rule.

class domonic.style.CSSImportRule

The CSSImportRule interface represents an @import rule in a CSS style sheet.

class domonic.style.CSSKeyframeRule

The CSSKeyframeRule interface represents a single @keyframes at-rule.

class domonic.style.CSSKeyframesRule

The CSSKeyframesRule interface represents a @keyframes at-rule.

class domonic.style.CSSMediaRule

The CSSMediaRule interface represents a @media rule in a CSS style sheet.

class domonic.style.CSSNamespaceRule

The CSSNamespaceRule interface represents an @namespace rule in a CSS style sheet.

class domonic.style.CSSPageRule

The CSSPageRule interface represents a @page rule in a CSS style sheet.

class domonic.style.CSSRule

The CSSRule interface represents a single CSS rule. There are several types of rules which inherit properties from CSSRule.

CSSFontFeatureValuesRule CSSViewportRule

property cssText

16pt }” or “@import ‘url’”. To access or modify parts of the rule (e.g. the value of “font-size” in the example) use the properties on the specialized interface for the rule’s type.

Type

Represents the textual representation of the rule, e.g. “h1,h2 { font-size

class domonic.style.CSSRuleList(*iterable*=(), /)

A CSSRuleList represents an ordered collection of read-only CSSRule objects. While the CSSRuleList object is read-only, and cannot be directly modified, it is considered a live object, as the content can change over time.

item(*index: int*) → *CSSRule*

Gets a single *CSSRule*.

property length: *int*

Returns an integer representing the number of *CSSRule* objects in the collection.

class domonic.style.*CSSStyleDeclaration*(*parentNode=None, *args, **kwargs*)

The *CSSStyleDeclaration* interface represents an object that is a CSS declaration block, and exposes style information and various style-related methods and properties.

A *CSSStyleDeclaration* object can be exposed using three different APIs:

Via *HTMLElement.style*, which deals with the inline styles of a single element (e.g., <div style="...">). Via the *CSSStyleSheet* API. For example, `document.styleSheets[0].cssRules[0].style` returns a *CSSStyleDeclaration* object on the first CSS rule in the document's first stylesheet. Via *Window.getComputedStyle()*, which exposes the *CSSStyleDeclaration* object as a read-only interface.

getPropertyCSSValue()

Only supported via *getComputedStyle* in Firefox. Returns the property value as a *CSSPrimitiveValue* or null for shorthand properties.

getPropertyPriority()

Returns the optional priority, "important".

getPropertyValue(*propertyName: str*) → *str*

Returns the value of the property with the specified name.

item(*index: int*) → *str*

Returns a CSS property name by its index, or the empty string if the index is out-of-bounds. An alternative to accessing `nodeList[i]` (which instead returns undefined when *i* is out-of-bounds). This is mostly useful for non-JavaScript DOM implementations.

property length

The number of properties. See the `item()` method below.

removeProperty()

Removes a property from the CSS declaration block.

class domonic.style.*CSSStyleRule*

The *CSSStyleRule* interface represents a single style rule in a CSS style sheet.

property cssText

Returns the textual representation of the rule.

class domonic.style.*CSSStyleSheet*

Creates a new *CSSStyleSheet* object.

addRule(*selectorText: str, style: str, index: int*)

Adds a new rule to the stylesheet given the selector to which the style applies and the style block to apply to the matching elements. This differs from `insertRule()`, which takes the textual representation of the entire rule as a single string.

deleteRule(*index: int*)

Deletes the rule at the specified index into the stylesheet's rule list.

insertRule(*rule: str, index: Optional[int] = None*)

Inserts a new rule at the specified position in the stylesheet, given the textual representation of the rule.

removeRule(*index: int*)

Functionally identical to deleteRule(); removes the rule at the specified index from the stylesheet's rule list.

replace(*text: str*)

Asynchronously replaces the content of the stylesheet and returns a Promise that resolves with the updated CSSStyleSheet.

replaceSync(*text: str*)

Synchronously replaces the content of the stylesheet.

class domonic.style.CSSSupportsRule

The CSSSupportsRule interface represents a @supports at-rule.

class domonic.style.MediaList(*iterable=()*, /)

The MediaList interface represents a list of media. It is used in the @media at-rule.

appendMedium(*newMedium: str*) → None

Appends a new medium to the end of the list.

deleteMedium(*oldMedium: str*) → None

Removes medium in the media list. If the medium is not found nothing happens.

item(*index: int*) → str

Returns the media at the given index in the MediaList.

property length: int

Returns the number of media in the list.

property mediaText: str

Returns a string containing the text of the media query.

class domonic.style.Style(*parent_node=None*)

[js syntax styles] # TODO - just add normal float? # TODO - consider camel case for hyphen params? # TODO
- not json serialisable due to the decorators.

property alignContent

Sets or returns the alignment between the lines inside a flexible container when the items do not use all available space

property alignItems

Sets or returns the alignment for items inside a flexible container

property alignSelf

Sets or returns the alignment for selected items inside a flexible container

property animation

shorthand property for all the animation properties below, except the animationPlayState property

property animationDelay

Sets or returns when the animation will start

property animationDirection

Sets or returns whether or not the animation should play in reverse on alternate cycles

property animationDuration

Sets or returns how many seconds or milliseconds an animation takes to complete one cycle

property animationFillMode

Sets or returns what values are applied by the animation outside the time it is executing

property animationIterationCount

Sets or returns the number of times an animation should be played

property animationName

Sets or returns a name for the @keyframes animation

property animationPlayState

Sets or returns whether the animation is running or paused

property animationTimingFunction

Sets or returns the speed curve of the animation

property backfaceVisibility

Sets or returns whether or not an element should be visible when not facing the screen

property background

Sets or returns all the background properties in one declaration

property backgroundAttachment

Sets or returns whether a background-image is fixed or scrolls with the page

property backgroundClip

Sets or returns the painting area of the background

property backgroundColor

Sets or returns the background-color of an element

property backgroundImage

Sets or returns the background-image for an element

property backgroundOrigin

Sets or returns the positioning area of the background images

property backgroundPosition

Sets or returns the starting position of a background-image

property backgroundRepeat

Sets or returns how to repeat (tile) a background-image

property backgroundSize

Sets or returns the size of the background image

property border

Sets or returns borderWidth, borderStyle, and borderColor in one declaration

property borderBottom

Sets or returns all the borderBottom properties in one declaration

property borderBottomColor

Sets or returns the color of the bottom border 1

property borderBottomLeftRadius

Sets or returns the shape of the border of the bottom-left corner

property borderBottomRightRadius

Sets or returns the shape of the border of the bottom-right corner

property borderBottomStyle

Sets or returns the style of the bottom border

property borderBottomWidth

Sets or returns the width of the bottom border

property borderCollapse

Sets or returns whether the table border should be collapsed into a single border, or not

property borderColor

Sets or returns the color of an element's border (can have up to four values)

property borderImage

horthand property for setting or returning all the borderImage properties

property borderImageOutset

Sets or returns the amount by which the border image area extends beyond the border box

property borderImageRepeat

Sets or returns whether the image-border should be repeated, rounded or stretched

property borderImageSlice

Sets or returns the inward offsets of the image-border

property borderImageSource

Sets or returns the image to be used as a border

property borderImageWidth

Sets or returns the widths of the image-border

property borderLeft

Sets or returns all the borderLeft properties in one declaration

property borderLeftColor

Sets or returns the color of the left border

property borderLeftStyle

Sets or returns the style of the left border

property borderLeftWidth

Sets or returns the width of the left border

property borderRadius

A shorthand property for setting or returning all the four borderRadius properties

property borderRight

Sets or returns all the borderRight properties in one declaration

property borderRightColor

Sets or returns the color of the right border

property borderRightStyle

Sets or returns the style of the right border

property `borderRightWidth`

Sets or returns the width of the right border

property `borderSpacing`

Sets or returns the space between cells in a table

property `borderStyle`

Sets or returns the style of an element's border (can have up to four values)

property `borderTop`

Sets or returns all the `borderTop` properties in one declaration

property `borderTopColor`

Sets or returns the color of the top border

property `borderTopLeftRadius`

Sets or returns the shape of the border of the top-left corner

property `borderTopRightRadius`

Sets or returns the shape of the border of the top-right corner

property `borderTopStyle`

Sets or returns the style of the top border

property `borderTopWidth`

Sets or returns the width of the top border

property `borderWidth`

Sets or returns the width of an element's border (can have up to four values)

property `bottom`

Sets or returns the bottom position of a positioned element

property `boxDecorationBreak`

Sets or returns the behaviour of the background and border of an element at page-break, or, for in-line elements, at line-break.

property `boxShadow`

Attaches one or more drop-shadows to the box

property `boxSizing`

Allows you to define certain elements to fit an area in a certain way

property `captionSide`

Sets or returns the position of the table caption

property `clear`

Sets or returns the position of the element relative to floating objects

property `clip`

Sets or returns which part of a positioned element is visible

property `color`

Sets or returns the color of the text

property `columnCount`

Sets or returns the number of columns an element should be divided into

property columnFill

Sets or returns how to fill columns

property columnGap

Sets or returns the gap between the columns

property columnRule

shorthand property for setting or returning all the columnRule properties

property columnRuleColor

Sets or returns the color of the rule between columns

property columnRuleStyle

Sets or returns the style of the rule between columns

property columnRuleWidth

Sets or returns the width of the rule between columns

property columnSpan

Sets or returns how many columns an element should span across

property columnWidth

Sets or returns the width of the columns

property columns

horthand property for setting or returning columnWidth and columnCount

property content

after pseudo-elements, to insert generated content

Type

d with the

Type

before and

property counterIncrement

Increments one or more counters

property counterReset

Creates or resets one or more counters

property cssFloat

Sets or returns the horizontal alignment of an element

property cursor

Sets or returns the type of cursor to display for the mouse pointer

property direction

Sets or returns the text direction

property display

Sets or returns an element's display type

property emptyCells

Sets or returns whether to show the border and background of empty cells, or not

property filter

Sets or returns image filters (visual effects, like blur and saturation)

property flex

Sets or returns the length of the item, relative to the rest

property flexBasis

Sets or returns the initial length of a flexible item

property flexDirection

Sets or returns the direction of the flexible items

property flexFlow

A shorthand property for the flexDirection and the flexWrap properties

property flexGrow

Sets or returns how much the item will grow relative to the rest

property flexShrink

Sets or returns how the item will shrink relative to the rest

property flexWrap

Sets or returns whether the flexible items should wrap or not

property font

Sets or returns fontStyle, fontVariant, fontWeight, fontSize, lineHeight, and fontFamily in one declaration

property fontFamily

Sets or returns the font family for text

property fontSize

Sets or returns the font size of the text

property fontSizeAdjust

reserves the readability of text when font fallback occurs

property fontStretch

ects a normal, condensed, or expanded face from a font family

property fontStyle

Sets or returns whether the style of the font is normal, italic or oblique

property fontVariant

Sets or returns whether the font should be displayed in small capital letters

property fontWeight

Sets or returns the boldness of the font

property hangingPunctuation

ecifies whether a punctuation character may be placed outside the line box

property height

Sets or returns the height of an element

property hyphens

Sets how to split words to improve the layout of paragraphs

property icon

Provides the author the ability to style an element with an iconic equivalent

property imageOrientation

Specifies a rotation in the right or clockwise direction that a user agent applies to an image

property isolation

Defines whether an element must create a new stacking context

property justifyContent

Sets or returns the alignment between the items inside a flexible container when the items do not use all available space.

property left

Sets or returns the left position of a positioned element

property letterSpacing

Sets or returns the space between characters in a text

property lineHeight

Sets or returns the distance between lines in a text

property listStyle

Sets or returns listStyleImage, listStylePosition, and listStyleType in one declaration

property listStyleImage

Sets or returns an image as the list-item marker

property listStylePosition

Sets or returns the position of the list-item marker

property listStyleType

Sets or returns the list-item marker type

property margin

Sets or returns the margins of an element (can have up to four values)

property marginBottom

Sets or returns the bottom margin of an element

property marginLeft

Sets or returns the left margin of an element

property marginRight

Sets or returns the right margin of an element

property marginTop

Sets or returns the top margin of an element

property maxHeight

Sets or returns the maximum height of an element

property maxWidth

Sets or returns the maximum width of an element

property minHeight

Sets or returns the minimum height of an element

property minWidth

Sets or returns the minimum width of an element

property navDown

Sets or returns where to navigate when using the arrow-down navigation key

property navIndex

Sets or returns the tabbing order for an element

property navLeft

Sets or returns where to navigate when using the arrow-left navigation key

property navRight

Sets or returns where to navigate when using the arrow-right navigation key

property navUp

Sets or returns where to navigate when using the arrow-up navigation key

property objectFit

pecifies how the contents of a replaced element should be fitted to the box established by its used height and width

property objectPosition

ecifies the alignment of the replaced element inside its box

property opacity

Sets or returns the opacity level for an element

property order

Sets or returns the order of the flexible item, relative to the rest

property orphans

Sets or returns the minimum number of lines for an element that must be left at the bottom of a page when a page break occurs inside an element

property outline

Sets or returns all the outline properties in one declaration

property outlineColor

Sets or returns the color of the outline around a element

property outlineOffset

ffsets an outline, and draws it beyond the border edge

property outlineStyle

Sets or returns the style of the outline around an element

property outlineWidth

Sets or returns the width of the outline around an element

property overflow

Sets or returns what to do with content that renders outside the element box

property overflowX

pecifies what to do with the left/right edges of the content, if it overflows the element's content area

property overflowY

pecifies what to do with the top/bottom edges of the content, if it overflows the element's content area

property padding

Sets or returns the padding of an element (can have up to four values)

property paddingBottom

Sets or returns the bottom padding of an element

property paddingLeft

Sets or returns the left padding of an element

property paddingRight

Sets or returns the right padding of an element

property paddingTop

Sets or returns the top padding of an element

property pageBreakAfter

Sets or returns the page-break behavior after an element

property pageBreakBefore

Sets or returns the page-break behavior before an element

property pageBreakInside

Sets or returns the page-break behavior inside an element

property perspective

Sets or returns the perspective on how 3D elements are viewed

property perspectiveOrigin

Sets or returns the bottom position of 3D elements

property position

Sets or returns the type of positioning method used for an element (static, relative, absolute or fixed)

property quotes

Sets or returns the type of quotation marks for embedded quotations

property resize

Sets or returns whether or not an element is resizable by the user

property right

Sets or returns the right position of a positioned element

property tabSize

Sets or returns the length of the tab-character

property tableLayout

Sets or returns the way to lay out table cells, rows, and columns

property textAlign

Sets or returns the horizontal alignment of text

property textAlignLast

Sets or returns how the last line of a block or a line right before a forced line break is aligned when text-align is justify

property textDecoration

Sets or returns the decoration of a text

property textDecorationColor

Sets or returns the color of the text-decoration

property textDecorationLine

Sets or returns the type of line in a text-decoration

property textDecorationStyle

Sets or returns the style of the line in a text decoration

property textIndent

Sets or returns the indentation of the first line of text

property textJustify

Sets or returns the justification method used when text-align is justify

property textOverflow

Sets or returns what should happen when text overflows the containing element

property textShadow

Sets or returns the shadow effect of a text

property textTransform

Sets or returns the capitalization of a text

property top

Sets or returns the top position of a positioned element

property transform

Applies a 2D or 3D transformation to an element

property transformOrigin

Sets or returns the position of transformed elements

property transformStyle

Sets or returns how nested elements are rendered in 3D space

property transition

Shorthand property for setting or returning the four transition properties

property transitionDelay

Sets or returns when the transition effect will start

property transitionDuration

Sets or returns how many seconds or milliseconds a transition effect takes to complete

property transitionProperty

Sets or returns the CSS property that the transition effect is for

property transitionTimingFunction

Sets or returns the speed curve of the transition effect

property unicodeBidi

Sets or returns whether the text should be overridden to support multiple languages in the same document

property userSelect

Sets or returns whether the text of an element can be selected or not

property verticalAlign

Sets or returns the vertical alignment of the content in an element

property visibility

Sets or returns whether an element should be visible

property `whiteSpace`

Sets or returns how to handle tabs, line breaks and whitespace in a text 1

property `widows`

Sets or returns the minimum number of lines for an element that must be visible at the top of a page

property `width`

Sets or returns the width of an element

property `wordBreak`

Sets or returns line breaking rules for non-CJK scripts

property `wordSpacing`

Sets or returns the spacing between words in a text

property `wordWrap`

Allows long, unbreakable words to be broken and wrap to the next line

property `zIndex`

Sets or returns the stack order of a positioned element

class `domonic.style.StyleSheet`

An object implementing the `StyleSheet` interface represents a single style sheet. CSS style sheets will further implement the more specialized `CSSStyleSheet` interface.

class `domonic.style.StyleSheetList`(*iterable=()*, /)

An instance of this object can be returned by `Document.styleSheets`. it can be iterated over in a standard for loop over its indices, or converted to an `Array`.

item(*index*)

Returns the `CSSStyleSheet` object at the index passed in, or null if no item exists for that index.

property `length`

Returns the number of `CSSStyleSheet` objects in the collection.

domonic.utils

snippets etc

class `domonic.utils.Utils`

static **acronym**(*sentence: str*) → *str*

[pass a sentence, returns the acronym]

Parameters

sentence (*[str]*) – [typically 3 words]

Returns

[a TLA (three letter acronym)]

Return type

[*str*]

static **case_camel**(*s: str*) → *str*

`case_camel('camel-case') > 'camelCase'`

static **case_kebab**(*s: str*) → *str*

`kebab('camelCase') # 'camel-case'`

static case_snake(*s: str*) → str
snake('camelCase') # 'camel_case'

static chunk(*list: list, size: int*) → list
chunk a list into batches

static chunks(*iterable, size: int, format=<built-in function iter>*)
Iterate over any iterable (list, set, file, stream, strings, whatever), of ANY size

static clean(*lst: list*) → list
[removes falsy values (False, None, 0 and "") from a list]

Parameters

lst ([*list*]) – [lst to operate on]

Returns

[a new list with falsy values removed]

Return type

[list]

static dictify(*arr: list*) → dict
[turns a list into a dictionary where the list items are the keys]

Parameters

arr ([*list*]) – [list to change]

Returns

[a new dict where the list items are now the keys]

Return type

[dict]

static digits(*text: str = ""*) → str
[takes a string of mix of digits and letters and returns a string of digits]

Parameters

text (*str, optional*) – [the text to change]. Defaults to ‘’.

Returns

[a string of digits]

Return type

[str]

static escape(*s: str*) → str
[escape a string]

Parameters

s ([*str*]) – [the string to escape]

Returns

[the escaped string]

Return type

[str]

static frequency(*data*)
[check the frequency of elements in the data]

Parameters

data ([*type*]) – [the data to check]

Returns

[a dict of elements and their frequency]

Return type

[dict]

static `get_vowels(string: str) → list`

[get a list of vowels from the word]

Parameters

string ([str]) – [the word to check]

Returns

[a list of vowels]

Return type

[list]

static `has_internet(url: str = 'http://www.google.com/', timeout: int = 5) → bool`

[check if you have internet connection]

Parameters

- **url** (str, optional) – [the url to check]. Defaults to 'http://www.google.com/'.
- **timeout** (int, optional) – [the timeout]. Defaults to 5.

Returns

[True if you have internet]

Return type

[bool]

static `init_assets(dir: str = 'assets') → None`

[creates an assets directory with nested js/css/img dirs]

Parameters

dir (str, optional) – [default directory name]. Defaults to 'assets'.

static `is_linux() → bool`

[check if the system is a linux]

Returns

[description]

Return type

[bool]

static `is_mac() → bool`

[check if the system is a mac]

Returns

[True if the system is a mac]

Return type

[bool]

static `is_nix() → bool`

[check if the system is a nix based system]

Returns

[True if it is a nix based system]

Return type`[bool]`**static** `is_windows()` \rightarrow `bool``[check if the system is a windows]`**Returns**`[True if windows]`**Return type**`[bool]`**static** `merge_dictionaries(a: dict, b: dict)` \rightarrow `dict``[merges 2 dicts]`**Parameters**

- **a** (`[dict]`) – [dict a]
- **b** (`[dict]`) – [dict b]

Returns`[a new dict]`**Return type**`[dict]`**static** `permutations(word: str)` \rightarrow `list``[provides all the possible permutations of a given word]`**Parameters****word** (`[str]`) – [the word to get permutations for]**Returns**`[a list of permutations]`**Return type**`[list]`**static** `replace_between(content: str, match: str, replacement: str, start: int = 0, end: int = 0)``[replace some text but only between certain indexes]`**Parameters**

- **content** (`str`) – [the content whos text you will be replacing]
- **match** (`str`) – [the string to find]
- **replacement** (`str`) – [the string to replace it with]
- **start** (`int`, *optional*) – [start index]. Defaults to 0.
- **end** (`int`, *optional*) – [end index]. Defaults to 0.

Returns`[the new string]`**Return type**`[str]`**static** `squash(the_list: list)` \rightarrow `list``[turns a 2d array into a flat one]`**Parameters****the_list** (`[list]`) – [a 2d array]

Returns

[a flattened 1d array]

Return type

[list]

static to_dictionary(*keys: list, values: list*) → dict

[take a list of keys and values and returns a dict]

Parameters

- **keys** ([list]) – [a list of keys]
- **values** ([list]) – [a list of value]

Returns

[a dictionary]

Return type

[dict]

static truncate(*text: str = "", length: int = 0*) → str

[truncates a string and appends 3 dots]

Parameters

- **text** (str, optional) – [the text to truncate]. Defaults to ‘’.
- **length** (int, optional) – [the max length]. Defaults to 0.

Returns

[the truncated string]

Return type

[str]

static unescape(*s: str*) → str

[unescape a string]

Parameters

s ([str]) – [the string to unescape]

Returns

[the unescaped string]

Return type

[str]

static unique(*some_arr: list*) → list

[removes duplicates from a list]

Parameters

some_arr ([list]) – [list containing duplicates]

Returns

[a list containing no duplicates]

Return type

[list]

static untittle(*string: str*) → str

[the opposite of title]

Parameters**str** (*[str]*) – [the string to change]**Returns**

[a string with the first character set to lowercase]

Return type*[str]***static url2file**(*url: str*) → *str*

[gen a safe filename from a url. by replacing '/' for '_' and ':' for '__']

Parameters**url** (*[str]*) – [the url to turn into a filename]**Returns**

[description]

Return type*[str]***domonic.decorators****domonic.decorators.as_json**(*func*)

decorate any function to return json instead of a python obj note - used by JSON.py

domonic.decorators.called(*before=None, error: Optional[Callable[[Exception], None]] = None*)

Decorator to call a function before and after a function call.

domonic.decorators.check(*f*)

Prints entry and exit of a function to the console

domonic.decorators.deprecated(*func*)

marks a function as deprecated.

domonic.decorators.el(*element='div', string: bool = False*)

[wraps the results of a function in an element]

domonic.decorators.iife(*before=None, error: Optional[Callable[[Exception], None]] = None*)

Decorator to call a function before and after a function call.

domonic.decorators.instead(*f, somethingelse*)

what to return if it fails

domonic.decorators.log(*logger, level='info'*)

@log(logging.getLogger('main'), level='warning')

domonic.decorators.silence(**args, **kwargs*)

stop a function from doing anything

domonic.svg

Generate SVG with python 3

WARNING - totally not tested. except circle. I just assumed it would work...

<https://www.w3.org/TR/SVG2/eltindex.html>

```
class domonic.svg.altGlyph(*args, **kwargs)
class domonic.svg.altGlyphDef(*args, **kwargs)
class domonic.svg.altGlyphItem(*args, **kwargs)
class domonic.svg.animate(*args, **kwargs)
class domonic.svg.animateColor(*args, **kwargs)
class domonic.svg.animateMotion(*args, **kwargs)
class domonic.svg.animateTransform(*args, **kwargs)
class domonic.svg.circle(*args, **kwargs)
class domonic.svg.clipPath(*args, **kwargs)
class domonic.svg.cursor(*args, **kwargs)
class domonic.svg.defs(*args, **kwargs)
class domonic.svg.desc(*args, **kwargs)
class domonic.svg.discard(*args, **kwargs)
class domonic.svg.ellipse(*args, **kwargs)
class domonic.svg.feBlend(*args, **kwargs)
class domonic.svg.feColorMatrix(*args, **kwargs)
class domonic.svg.feComponentTransfer(*args, **kwargs)
class domonic.svg.feComposite(*args, **kwargs)
class domonic.svg.feConvolveMatrix(*args, **kwargs)
class domonic.svg.feDiffuseLighting(*args, **kwargs)
class domonic.svg.feDisplacementMap(*args, **kwargs)
class domonic.svg.feDistantLight(*args, **kwargs)
class domonic.svg.feDropShadow(*args, **kwargs)
class domonic.svg.feFlood(*args, **kwargs)
class domonic.svg.feFuncA(*args, **kwargs)
class domonic.svg.feFuncB(*args, **kwargs)
class domonic.svg.feFuncG(*args, **kwargs)
```

```
class domonic.svg.feFuncR(*args, **kwargs)
class domonic.svg.feGaussianBlur(*args, **kwargs)
class domonic.svg.feImage(*args, **kwargs)
class domonic.svg.feMerge(*args, **kwargs)
class domonic.svg.feMergeNode(*args, **kwargs)
class domonic.svg.feMorphology(*args, **kwargs)
class domonic.svg.feOffset(*args, **kwargs)
class domonic.svg.fePointLight(*args, **kwargs)
class domonic.svg.feSpecularLighting(*args, **kwargs)
class domonic.svg.feSpotLight(*args, **kwargs)
class domonic.svg.feTile(*args, **kwargs)
class domonic.svg.feTurbulence(*args, **kwargs)
class domonic.svg.font(*args, **kwargs)
class domonic.svg.foreignObject(*args, **kwargs)
class domonic.svg.g(*args, **kwargs)
class domonic.svg.glyph(*args, **kwargs)
class domonic.svg.glyphRef(*args, **kwargs)
class domonic.svg.hatch(*args, **kwargs)
class domonic.svg.hatchpath(*args, **kwargs)
class domonic.svg.hkern(*args, **kwargs)
class domonic.svg.image(*args, **kwargs)
class domonic.svg.line(*args, **kwargs)
class domonic.svg.linearGradient(*args, **kwargs)
class domonic.svg.marker(*args, **kwargs)
class domonic.svg.mask(*args, **kwargs)
class domonic.svg.metadata(*args, **kwargs)
class domonic.svg.mpath(*args, **kwargs)
class domonic.svg.path(*args, **kwargs)
class domonic.svg.pattern(*args, **kwargs)
class domonic.svg.polygon(*args, **kwargs)
class domonic.svg.polyline(*args, **kwargs)
```

```
class domonic.svg.radialGradient(*args, **kwargs)
class domonic.svg.rect(*args, **kwargs)
class domonic.svg.solidcolor(*args, **kwargs)
class domonic.svg.stop(*args, **kwargs)
class domonic.svg.svg(*args, **kwargs)
class domonic.svg.switch(*args, **kwargs)
class domonic.svg.symbol(*args, **kwargs)
class domonic.svg.text(*args, **kwargs)
class domonic.svg.textPath(*args, **kwargs)
class domonic.svg.title(*args, **kwargs)
class domonic.svg.tref(*args, **kwargs)
class domonic.svg.tspan(*args, **kwargs)
class domonic.svg.unknown(*args, **kwargs)
class domonic.svg.use(*args, **kwargs)
class domonic.svg.view(*args, **kwargs)
class domonic.svg.vkern(*args, **kwargs)
class domonic.lerpy.tween.Tween(target=None, values=None, duration=0, equations=None, delay=0,
                                loop=False)
```

Tween is a complex lerp but you don't have to do the math cos robert penner did already. Just pass an easing equation from the easing package

i.e

```
tw = Tween(someObj, { 'x':10, 'y':5, 'z':3}, 6, Linear.easeIn )
```

will tween the objects properties over the given time using Linear.easeIn

pause()

Pauses the tween from changing the values

unpause()

unpauses the tween

domonic.x3d

Generate x3d with python 3

```
class domonic.xml.x3d.Anchor(*args, **kwargs)
```

domonic.xml.x3d.Appearance

alias of [appearance](#)

```
class domonic.xml.x3d.Arc2D(*args, **kwargs)
```

```

class domonic.xml.x3d.ArcClose2D(*args, **kwargs)
class domonic.xml.x3d.AudioClip(*args, **kwargs)
class domonic.xml.x3d.Background(*args, **kwargs)
class domonic.xml.x3d.BallJoint(*args, **kwargs)
class domonic.xml.x3d.Billboard(*args, **kwargs)
class domonic.xml.x3d.BinaryGeometry(*args, **kwargs)
class domonic.xml.x3d.BlendMode(*args, **kwargs)
class domonic.xml.x3d.BlendedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Block(*args, **kwargs)
class domonic.xml.x3d.BoundaryEnhancementVolumeStyle(*args, **kwargs)
domonic.xml.x3d.Box
    alias of box
class domonic.xml.x3d.BufferAccessor(*args, **kwargs)
class domonic.xml.x3d.BufferGeometry(*args, **kwargs)
class domonic.xml.x3d.BufferView(*args, **kwargs)
class domonic.xml.x3d.CADAssembly(*args, **kwargs)
class domonic.xml.x3d.CADFace(*args, **kwargs)
class domonic.xml.x3d.CADLayer(*args, **kwargs)
class domonic.xml.x3d.CADPart(*args, **kwargs)
class domonic.xml.x3d.CartoonVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Circle2D(*args, **kwargs)
class domonic.xml.x3d.ClipPlane(*args, **kwargs)
class domonic.xml.x3d.CollidableShape(*args, **kwargs)
class domonic.xml.x3d.Collision(*args, **kwargs)
class domonic.xml.x3d.CollisionCollection(*args, **kwargs)
class domonic.xml.x3d.CollisionSensor(*args, **kwargs)
class domonic.xml.x3d.Color(*args, **kwargs)
class domonic.xml.x3d.ColorChaser(*args, **kwargs)
class domonic.xml.x3d.ColorDamper(*args, **kwargs)
class domonic.xml.x3d.ColorInterpolator(*args, **kwargs)
class domonic.xml.x3d.ColorMaskMode(*args, **kwargs)

```

```
class domonic.xml.x3d.ColorRGBA(*args, **kwargs)
class domonic.xml.x3d.CommonSurfaceShader(*args, **kwargs)
class domonic.xml.x3d.ComposedCubeMapTexture(*args, **kwargs)
class domonic.xml.x3d.ComposedShader(*args, **kwargs)
class domonic.xml.x3d.ComposedTexture3D(*args, **kwargs)
class domonic.xml.x3d.ComposedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Cone(*args, **kwargs)
class domonic.xml.x3d.Coordinate(*args, **kwargs)
class domonic.xml.x3d.CoordinateDamper(*args, **kwargs)
class domonic.xml.x3d.CoordinateDouble(*args, **kwargs)
class domonic.xml.x3d.CoordinateInterpolator(*args, **kwargs)
class domonic.xml.x3d.Cylinder(*args, **kwargs)
class domonic.xml.x3d.CylinderSensor(*args, **kwargs)
class domonic.xml.x3d.DepthMode(*args, **kwargs)
class domonic.xml.x3d.DirectionallLight(*args, **kwargs)
class domonic.xml.x3d.Dish(*args, **kwargs)
class domonic.xml.x3d.Disk2D(*args, **kwargs)
class domonic.xml.x3d.DoubleAxisHingeJoint(*args, **kwargs)
class domonic.xml.x3d.DynamicLOD(*args, **kwargs)
class domonic.xml.x3d.EdgeEnhancementVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.ElevationGrid(*args, **kwargs)
class domonic.xml.x3d.Environment(*args, **kwargs)
class domonic.xml.x3d.Extrusion(*args, **kwargs)
class domonic.xml.x3d.Field(*args, **kwargs)
class domonic.xml.x3d.FloatVertexAttribute(*args, **kwargs)
class domonic.xml.x3d.Fog(*args, **kwargs)
class domonic.xml.x3d.FontStyle(*args, **kwargs)
class domonic.xml.x3d.GeneratedCubeMapTexture(*args, **kwargs)
class domonic.xml.x3d.GeoCoordinate(*args, **kwargs)
class domonic.xml.x3d.GeoElevationGrid(*args, **kwargs)
class domonic.xml.x3d.GeoLOD(*args, **kwargs)
```

```

class domonic.xml.x3d.GeoLocation(*args, **kwargs)
class domonic.xml.x3d.GeoMetadata(*args, **kwargs)
class domonic.xml.x3d.GeoOrigin(*args, **kwargs)
class domonic.xml.x3d.GeoPositionInterpolator(*args, **kwargs)
class domonic.xml.x3d.GeoTransform(*args, **kwargs)
class domonic.xml.x3d.GeoViewpoint(*args, **kwargs)
class domonic.xml.x3d.Group(*args, **kwargs)
class domonic.xml.x3d.HAnimDisplacer(*args, **kwargs)
class domonic.xml.x3d.HAnimHumanoid(*args, **kwargs)
class domonic.xml.x3d.HAnimJoint(*args, **kwargs)
class domonic.xml.x3d.HAnimSegment(*args, **kwargs)
class domonic.xml.x3d.HAnimSite(*args, **kwargs)
class domonic.xml.x3d.ImageTexture(*args, **kwargs)
class domonic.xml.x3d.ImageTexture3D(*args, **kwargs)
class domonic.xml.x3d.ImageTextureAtlas(*args, **kwargs)
class domonic.xml.x3d.IndexedFaceSet(*args, **kwargs)
class domonic.xml.x3d.IndexedLineSet(*args, **kwargs)
class domonic.xml.x3d.IndexedQuadSet(*args, **kwargs)
class domonic.xml.x3d.IndexedTriangleSet(*args, **kwargs)
class domonic.xml.x3d.IndexedTriangleStripSet(*args, **kwargs)
domonic.xml.x3d.Inline
    alias of inline
class domonic.xml.x3d.IsoSurfaceVolumeData(*args, **kwargs)
class domonic.xml.x3d.LOD(*args, **kwargs)
class domonic.xml.x3d.LineProperties(*args, **kwargs)
class domonic.xml.x3d.LineSet(*args, **kwargs)
class domonic.xml.x3d.MPRPlane(*args, **kwargs)
class domonic.xml.x3d.MPRVolumeStyle(*args, **kwargs)
domonic.xml.x3d.Material
    alias of material
class domonic.xml.x3d.MatrixTextureTransform(*args, **kwargs)
class domonic.xml.x3d.MatrixTransform(*args, **kwargs)

```

```
class domonic.xml.x3d.Mesh(*args, **kwargs)
class domonic.xml.x3d.MetadataBoolean(*args, **kwargs)
class domonic.xml.x3d.MetadataDouble(*args, **kwargs)
class domonic.xml.x3d.MetadataFloat(*args, **kwargs)
class domonic.xml.x3d.MetadataInteger(*args, **kwargs)
class domonic.xml.x3d.MetadataSet(*args, **kwargs)
class domonic.xml.x3d.MetadataString(*args, **kwargs)
class domonic.xml.x3d.MotorJoint(*args, **kwargs)
class domonic.xml.x3d.MovieTexture(*args, **kwargs)
class domonic.xml.x3d.MultiTexture(*args, **kwargs)
class domonic.xml.x3d.MultiTextureCoordinate(*args, **kwargs)
class domonic.xml.x3d.NavigationInfo(*args, **kwargs)
class domonic.xml.x3d.Normal(*args, **kwargs)
class domonic.xml.x3d.NormalInterpolator(*args, **kwargs)
class domonic.xml.x3d.Nozzle(*args, **kwargs)
class domonic.xml.x3d.OpacityMapVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.OrientationChaser(*args, **kwargs)
class domonic.xml.x3d.OrientationDamper(*args, **kwargs)
class domonic.xml.x3d.OrientationInterpolator(*args, **kwargs)
class domonic.xml.x3d.OrthoViewpoint(*args, **kwargs)
class domonic.xml.x3d.Param(*args, **kwargs)
class domonic.xml.x3d.ParticleSet(*args, **kwargs)
class domonic.xml.x3d.PhysicalEnvironmentLight(*args, **kwargs)
class domonic.xml.x3d.PhysicalMaterial(*args, **kwargs)
class domonic.xml.x3d.PixelTexture(*args, **kwargs)
class domonic.xml.x3d.PixelTexture3D(*args, **kwargs)
domonic.xml.x3d.Plane
    alias of plane
class domonic.xml.x3d.PlaneSensor(*args, **kwargs)
class domonic.xml.x3d.PointLight(*args, **kwargs)
class domonic.xml.x3d.PointSet(*args, **kwargs)
```



```

class domonic.xml.x3d.Polyline2D(*args, **kwargs)
class domonic.xml.x3d.Polypoint2D(*args, **kwargs)
class domonic.xml.x3d.PopGeometry(*args, **kwargs)
class domonic.xml.x3d.PopGeometryLevel(*args, **kwargs)
class domonic.xml.x3d.PositionChaser(*args, **kwargs)
class domonic.xml.x3d.PositionChaser2D(*args, **kwargs)
class domonic.xml.x3d.PositionDamper(*args, **kwargs)
class domonic.xml.x3d.PositionDamper2D(*args, **kwargs)
class domonic.xml.x3d.PositionInterpolator(*args, **kwargs)
class domonic.xml.x3d.PositionInterpolator2D(*args, **kwargs)
class domonic.xml.x3d.ProjectionVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Pyramid(*args, **kwargs)
class domonic.xml.x3d.QuadSet(*args, **kwargs)
class domonic.xml.x3d.RadarVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Rectangle2D(*args, **kwargs)
class domonic.xml.x3d.RectangularTorus(*args, **kwargs)
class domonic.xml.x3d.RefinementTexture(*args, **kwargs)
class domonic.xml.x3d.RemoteSelectionGroup(*args, **kwargs)
class domonic.xml.x3d.RenderedTexture(*args, **kwargs)
class domonic.xml.x3d.RigidBody(*args, **kwargs)
class domonic.xml.x3d.RigidBodyCollection(*args, **kwargs)
class domonic.xml.x3d.Route(*args, **kwargs)
class domonic.xml.x3d.ScalarChaser(*args, **kwargs)
class domonic.xml.x3d.ScalarDamper(*args, **kwargs)
class domonic.xml.x3d.ScalarInterpolator(*args, **kwargs)
domonic.xml.x3d.Scene
    alias of scene
class domonic.xml.x3d.SegmentedVolumeData(*args, **kwargs)
class domonic.xml.x3d.ShadedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.ShaderPart(*args, **kwargs)
domonic.xml.x3d.Shape
    alias of shape

```

```
class domonic.xml.x3d.SilhouetteEnhancementVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.SingleAxisHingeJoint(*args, **kwargs)
class domonic.xml.x3d.SliderJoint(*args, **kwargs)
class domonic.xml.x3d.SlopedCylinder(*args, **kwargs)
class domonic.xml.x3d.Snout(*args, **kwargs)
class domonic.xml.x3d.SolidOfRevolution(*args, **kwargs)
class domonic.xml.x3d.Sound(*args, **kwargs)
domonic.xml.x3d.Sphere
    alias of sphere
class domonic.xml.x3d.SphereSegment(*args, **kwargs)
class domonic.xml.x3d.SphereSensor(*args, **kwargs)
class domonic.xml.x3d.SplinePositionInterpolator(*args, **kwargs)
class domonic.xml.x3d.SpotLight(*args, **kwargs)
class domonic.xml.x3d.StaticGroup(*args, **kwargs)
class domonic.xml.x3d.StippleVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.SurfaceShaderTexture(*args, **kwargs)
class domonic.xml.x3d.Switch(*args, **kwargs)
class domonic.xml.x3d.TexCoordDamper2D(*args, **kwargs)
class domonic.xml.x3d.Text(*args, **kwargs)
class domonic.xml.x3d.Texture(*args, **kwargs)
class domonic.xml.x3d.TextureCoordinate(*args, **kwargs)
class domonic.xml.x3d.TextureCoordinate3D(*args, **kwargs)
class domonic.xml.x3d.TextureCoordinateGenerator(*args, **kwargs)
class domonic.xml.x3d.TextureProperties(*args, **kwargs)
class domonic.xml.x3d.TextureTransform(*args, **kwargs)
class domonic.xml.x3d.TextureTransform3D(*args, **kwargs)
class domonic.xml.x3d.TextureTransformMatrix3D(*args, **kwargs)
domonic.xml.x3d.TimeSensor
    alias of timeSensor
class domonic.xml.x3d.ToneMappedVolumeStyle(*args, **kwargs)
class domonic.xml.x3d.Torus(*args, **kwargs)
class domonic.xml.x3d.TouchSensor(*args, **kwargs)
```

domonic.xml.x3d.Transform

alias of *transform*

class domonic.xml.x3d.TriangleSet(*args, **kwargs)

class domonic.xml.x3d.TriangleSet2D(*args, **kwargs)

class domonic.xml.x3d.TwoSidedMaterial(*args, **kwargs)

class domonic.xml.x3d.Uniform(*args, **kwargs)

class domonic.xml.x3d.UniversalJoint(*args, **kwargs)

class domonic.xml.x3d.Viewfrustum(*args, **kwargs)

class domonic.xml.x3d.Viewpoint(*args, **kwargs)

class domonic.xml.x3d.VolumeData(*args, **kwargs)

class domonic.xml.x3d.WorldInfo(*args, **kwargs)

domonic.xml.x3d.X3D

alias of *x3d*

class domonic.xml.x3d.X3DAppearanceChildNode(*args, **kwargs)

class domonic.xml.x3d.X3DAppearanceNode(*args, **kwargs)

class domonic.xml.x3d.X3DBackgroundNode(*args, **kwargs)

class domonic.xml.x3d.X3DBinaryContainerGeometryNode(*args, **kwargs)

class domonic.xml.x3d.X3DBindableNode(*args, **kwargs)

class domonic.xml.x3d.X3DBoundedObject(*args, **kwargs)

class domonic.xml.x3d.X3DChaserNode(*args, **kwargs)

class domonic.xml.x3d.X3DChildNode(*args, **kwargs)

class domonic.xml.x3d.X3DColorNode(*args, **kwargs)

class domonic.xml.x3d.X3DComposableVolumeRenderStyleNode(*args, **kwargs)

class domonic.xml.x3d.X3DComposedGeometryNode(*args, **kwargs)

class domonic.xml.x3d.X3DCoordinateNode(*args, **kwargs)

class domonic.xml.x3d.X3DDamperNode(*args, **kwargs)

class domonic.xml.x3d.X3DDragSensorNode(*args, **kwargs)

class domonic.xml.x3d.X3DEnvironmentNode(*args, **kwargs)

class domonic.xml.x3d.X3DEnvironmentTextureNode(*args, **kwargs)

class domonic.xml.x3d.X3DFogNode(*args, **kwargs)

class domonic.xml.x3d.X3DFollowerNode(*args, **kwargs)

class domonic.xml.x3d.X3DFontStyleNode(*args, **kwargs)

```
class domonic.xml.x3d.X3DGeometricPropertyNode(*args, **kwargs)
class domonic.xml.x3d.X3DGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DGroupingNode(*args, **kwargs)
class domonic.xml.x3d.X3DInfoNode(*args, **kwargs)
class domonic.xml.x3d.X3DInterpolatorNode(*args, **kwargs)
class domonic.xml.x3d.X3DLODNode(*args, **kwargs)
class domonic.xml.x3d.X3DLightNode(*args, **kwargs)
class domonic.xml.x3d.X3DMaterialNode(*args, **kwargs)
class domonic.xml.x3d.X3DMetadataObject(*args, **kwargs)
class domonic.xml.x3d.X3DNBodyCollidableNode(*args, **kwargs)
class domonic.xml.x3d.X3DNavigationInfoNode(*args, **kwargs)
class domonic.xml.x3d.X3DNode(*args, **kwargs)
class domonic.xml.x3d.X3DPlanarGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DPointingDeviceSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DRigidJointNode(*args, **kwargs)
class domonic.xml.x3d.X3DSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DShaderNode(*args, **kwargs)
class domonic.xml.x3d.X3DShapeNode(*args, **kwargs)
class domonic.xml.x3d.X3DSoundNode(*args, **kwargs)
class domonic.xml.x3d.X3DSoundSourceNode(*args, **kwargs)
class domonic.xml.x3d.X3DSpatialGeometryNode(*args, **kwargs)
class domonic.xml.x3d.X3DTexture3DNode(*args, **kwargs)
class domonic.xml.x3d.X3DTextureCoordinateNode(*args, **kwargs)
class domonic.xml.x3d.X3DTextureNode(*args, **kwargs)
class domonic.xml.x3d.X3DTextureTransformNode(*args, **kwargs)
class domonic.xml.x3d.X3DTimeDependentNode(*args, **kwargs)
class domonic.xml.x3d.X3DTouchSensorNode(*args, **kwargs)
class domonic.xml.x3d.X3DTransformNode(*args, **kwargs)
class domonic.xml.x3d.X3DVertexAttributeNode(*args, **kwargs)
class domonic.xml.x3d.X3DViewpointNode(*args, **kwargs)
class domonic.xml.x3d.X3DVolumeDataNode(*args, **kwargs)
```

```
class domonic.xml.x3d.X3DVolumeRenderStyleNode(*args, **kwargs)
```

```
domonic.xml.x3d.anchor
```

```
    alias of Anchor
```

```
class domonic.xml.x3d.appearance(*args, **kwargs)
```

```
domonic.xml.x3d.arc2D
```

```
    alias of Arc2D
```

```
domonic.xml.x3d.arcClose2D
```

```
    alias of ArcClose2D
```

```
domonic.xml.x3d.audioClip
```

```
    alias of AudioClip
```

```
domonic.xml.x3d.background
```

```
    alias of Background
```

```
domonic.xml.x3d.ballJoint
```

```
    alias of BallJoint
```

```
domonic.xml.x3d.billboard
```

```
    alias of Billboard
```

```
domonic.xml.x3d.binaryGeometry
```

```
    alias of BinaryGeometry
```

```
domonic.xml.x3d.blendMode
```

```
    alias of BlendMode
```

```
domonic.xml.x3d.blendedVolumeStyle
```

```
    alias of BlendedVolumeStyle
```

```
domonic.xml.x3d.block
```

```
    alias of Block
```

```
domonic.xml.x3d.boundaryEnhancementVolumeStyle
```

```
    alias of BoundaryEnhancementVolumeStyle
```

```
class domonic.xml.x3d.box(*args, **kwargs)
```

```
domonic.xml.x3d.bufferAccessor
```

```
    alias of BufferAccessor
```

```
domonic.xml.x3d.bufferGeometry
```

```
    alias of BufferGeometry
```

```
domonic.xml.x3d.bufferView
```

```
    alias of BufferView
```

```
domonic.xml.x3d.CADAssembly
```

```
    alias of CADAssembly
```

```
domonic.xml.x3d.CADFace
```

```
    alias of CADFace
```

```
domonic.xml.x3d.CADLayer
```

```
    alias of CADLayer
```

domonic.xml.x3d.cADPart
alias of *CADPart*

domonic.xml.x3d.cartoonVolumeStyle
alias of *CartoonVolumeStyle*

domonic.xml.x3d.circle2D
alias of *Circle2D*

domonic.xml.x3d.clipPlane
alias of *ClipPlane*

domonic.xml.x3d.collidableShape
alias of *CollidableShape*

domonic.xml.x3d.collision
alias of *Collision*

domonic.xml.x3d.collisionCollection
alias of *CollisionCollection*

domonic.xml.x3d.collisionSensor
alias of *CollisionSensor*

domonic.xml.x3d.color
alias of *Color*

domonic.xml.x3d.colorChaser
alias of *ColorChaser*

domonic.xml.x3d.colorDamper
alias of *ColorDamper*

domonic.xml.x3d.colorInterpolator
alias of *ColorInterpolator*

domonic.xml.x3d.colorMaskMode
alias of *ColorMaskMode*

domonic.xml.x3d.colorRGBA
alias of *ColorRGBA*

domonic.xml.x3d.commonSurfaceShader
alias of *CommonSurfaceShader*

domonic.xml.x3d.composedCubeMapTexture
alias of *ComposedCubeMapTexture*

domonic.xml.x3d.composedShader
alias of *ComposedShader*

domonic.xml.x3d.composedTexture3D
alias of *ComposedTexture3D*

domonic.xml.x3d.composedVolumeStyle
alias of *ComposedVolumeStyle*

domonic.xml.x3d.cone
alias of *Cone*

domonic.xml.x3d.coordinate
alias of *Coordinate*

domonic.xml.x3d.coordinateDamper
alias of *CoordinateDamper*

domonic.xml.x3d.coordinateDouble
alias of *CoordinateDouble*

domonic.xml.x3d.coordinateInterpolator
alias of *CoordinateInterpolator*

domonic.xml.x3d.cylinder
alias of *Cylinder*

domonic.xml.x3d.cylinderSensor
alias of *CylinderSensor*

domonic.xml.x3d.depthMode
alias of *DepthMode*

domonic.xml.x3d.directionalLight
alias of *DirectionalLight*

domonic.xml.x3d.dish
alias of *Dish*

domonic.xml.x3d.disk2D
alias of *Disk2D*

domonic.xml.x3d.doubleAxisHingeJoint
alias of *DoubleAxisHingeJoint*

domonic.xml.x3d.dynamicLOD
alias of *DynamicLOD*

domonic.xml.x3d.edgeEnhancementVolumeStyle
alias of *EdgeEnhancementVolumeStyle*

domonic.xml.x3d.elevationGrid
alias of *ElevationGrid*

domonic.xml.x3d.environment
alias of *Environment*

domonic.xml.x3d.extrusion
alias of *Extrusion*

domonic.xml.x3d.field
alias of *Field*

domonic.xml.x3d.floatVertexAttribute
alias of *FloatVertexAttribute*

domonic.xml.x3d.fog
alias of *Fog*

domonic.xml.x3d.fontStyle
alias of *FontStyle*

domonic.xml.x3d.generatedCubeMapTexture
alias of *GeneratedCubeMapTexture*

domonic.xml.x3d.geoCoordinate
alias of *GeoCoordinate*

domonic.xml.x3d.geoElevationGrid
alias of *GeoElevationGrid*

domonic.xml.x3d.geoLOD
alias of *GeoLOD*

domonic.xml.x3d.geoLocation
alias of *GeoLocation*

domonic.xml.x3d.geoMetadata
alias of *GeoMetadata*

domonic.xml.x3d.geoOrigin
alias of *GeoOrigin*

domonic.xml.x3d.geoPositionInterpolator
alias of *GeoPositionInterpolator*

domonic.xml.x3d.geoTransform
alias of *GeoTransform*

domonic.xml.x3d.geoViewpoint
alias of *GeoViewpoint*

domonic.xml.x3d.group
alias of *Group*

domonic.xml.x3d.hAnimDisplacer
alias of *HAnimDisplacer*

domonic.xml.x3d.hAnimHumanoid
alias of *HAnimHumanoid*

domonic.xml.x3d.hAnimJoint
alias of *HAnimJoint*

domonic.xml.x3d.hAnimSegment
alias of *HAnimSegment*

domonic.xml.x3d.hAnimSite
alias of *HAnimSite*

domonic.xml.x3d.imageTexture
alias of *ImageTexture*

domonic.xml.x3d.imageTexture3D
alias of *ImageTexture3D*

`domonic.xml.x3d.imageTextureAtlas`

alias of *ImageTextureAtlas*

`domonic.xml.x3d.indexedFaceSet`

alias of *IndexedFaceSet*

`domonic.xml.x3d.indexedLineSet`

alias of *IndexedLineSet*

`domonic.xml.x3d.indexedQuadSet`

alias of *IndexedQuadSet*

`domonic.xml.x3d.indexedTriangleSet`

alias of *IndexedTriangleSet*

`domonic.xml.x3d.indexedTriangleStripSet`

alias of *IndexedTriangleStripSet*

class `domonic.xml.x3d.inline(*args, **kwargs)`

`domonic.xml.x3d.isoSurfaceVolumeData`

alias of *IsoSurfaceVolumeData*

`domonic.xml.x3d.lod`

alias of *LOD*

`domonic.xml.x3d.lineProperties`

alias of *LineProperties*

`domonic.xml.x3d.lineSet`

alias of *LineSet*

`domonic.xml.x3d.mprPlane`

alias of *MPRPlane*

`domonic.xml.x3d.mprVolumeStyle`

alias of *MPRVolumeStyle*

class `domonic.xml.x3d.material(*args, **kwargs)`

`domonic.xml.x3d.matrixTextureTransform`

alias of *MatrixTextureTransform*

`domonic.xml.x3d.matrixTransform`

alias of *MatrixTransform*

`domonic.xml.x3d.mesh`

alias of *Mesh*

`domonic.xml.x3d.metadataBoolean`

alias of *MetadataBoolean*

`domonic.xml.x3d.metadataDouble`

alias of *MetadataDouble*

`domonic.xml.x3d.metadataFloat`

alias of *MetadataFloat*

`domonic.xml.x3d.metadataInteger`
alias of *MetadataInteger*

`domonic.xml.x3d.metadataSet`
alias of *MetadataSet*

`domonic.xml.x3d.metadataString`
alias of *MetadataString*

`domonic.xml.x3d.motorJoint`
alias of *MotorJoint*

`domonic.xml.x3d.movieTexture`
alias of *MovieTexture*

`domonic.xml.x3d.multiTexture`
alias of *MultiTexture*

`domonic.xml.x3d.multiTextureCoordinate`
alias of *MultiTextureCoordinate*

`domonic.xml.x3d.navigationInfo`
alias of *NavigationInfo*

`domonic.xml.x3d.normal`
alias of *Normal*

`domonic.xml.x3d.normalInterpolator`
alias of *NormalInterpolator*

`domonic.xml.x3d.nozzle`
alias of *Nozzle*

`domonic.xml.x3d.opacityMapVolumeStyle`
alias of *OpacityMapVolumeStyle*

`domonic.xml.x3d.orientationChaser`
alias of *OrientationChaser*

`domonic.xml.x3d.orientationDamper`
alias of *OrientationDamper*

`domonic.xml.x3d.orientationInterpolator`
alias of *OrientationInterpolator*

`domonic.xml.x3d.orthoViewpoint`
alias of *OrthoViewpoint*

`domonic.xml.x3d.param`
alias of *Param*

`domonic.xml.x3d.particleSet`
alias of *ParticleSet*

`domonic.xml.x3d.physicalEnvironmentLight`
alias of *PhysicalEnvironmentLight*

`domonic.xml.x3d.physicalMaterial`
alias of *PhysicalMaterial*

`domonic.xml.x3d.pixelTexture`
alias of *PixelTexture*

`domonic.xml.x3d.pixelTexture3D`
alias of *PixelTexture3D*

class `domonic.xml.x3d.plane(*args, **kwargs)`

`domonic.xml.x3d.planeSensor`
alias of *PlaneSensor*

`domonic.xml.x3d.pointLight`
alias of *PointLight*

`domonic.xml.x3d.pointSet`
alias of *PointSet*

`domonic.xml.x3d.polyline2D`
alias of *Polyline2D*

`domonic.xml.x3d.polypoint2D`
alias of *Polypoint2D*

`domonic.xml.x3d.popGeometry`
alias of *PopGeometry*

`domonic.xml.x3d.popGeometryLevel`
alias of *PopGeometryLevel*

`domonic.xml.x3d.positionChaser`
alias of *PositionChaser*

`domonic.xml.x3d.positionChaser2D`
alias of *PositionChaser2D*

`domonic.xml.x3d.positionDamper`
alias of *PositionDamper*

`domonic.xml.x3d.positionDamper2D`
alias of *PositionDamper2D*

`domonic.xml.x3d.positionInterpolator`
alias of *PositionInterpolator*

`domonic.xml.x3d.positionInterpolator2D`
alias of *PositionInterpolator2D*

`domonic.xml.x3d.projectionVolumeStyle`
alias of *ProjectionVolumeStyle*

`domonic.xml.x3d.pyramid`
alias of *Pyramid*

`domonic.xml.x3d.quadSet`
alias of *QuadSet*

domonic.xml.x3d.radarVolumeStyle
alias of *RadarVolumeStyle*

domonic.xml.x3d.rectangle2D
alias of *Rectangle2D*

domonic.xml.x3d.rectangularTorus
alias of *RectangularTorus*

domonic.xml.x3d.refinementTexture
alias of *RefinementTexture*

domonic.xml.x3d.remoteSelectionGroup
alias of *RemoteSelectionGroup*

domonic.xml.x3d.renderedTexture
alias of *RenderedTexture*

domonic.xml.x3d.rigidBody
alias of *RigidBody*

domonic.xml.x3d.rigidBodyCollection
alias of *RigidBodyCollection*

domonic.xml.x3d.route
alias of *Route*

domonic.xml.x3d.scalarChaser
alias of *ScalarChaser*

domonic.xml.x3d.scalarDamper
alias of *ScalarDamper*

domonic.xml.x3d.scalarInterpolator
alias of *ScalarInterpolator*

class domonic.xml.x3d.scene(*args, **kwargs)

domonic.xml.x3d.segmentedVolumeData
alias of *SegmentedVolumeData*

domonic.xml.x3d.shadedVolumeStyle
alias of *ShadedVolumeStyle*

domonic.xml.x3d.shaderPart
alias of *ShaderPart*

class domonic.xml.x3d.shape(*args, **kwargs)

domonic.xml.x3d.silhouetteEnhancementVolumeStyle
alias of *SilhouetteEnhancementVolumeStyle*

domonic.xml.x3d.singleAxisHingeJoint
alias of *SingleAxisHingeJoint*

domonic.xml.x3d.sliderJoint
alias of *SliderJoint*

`domonic.xml.x3d.slopedCylinder`
alias of *SlopedCylinder*

`domonic.xml.x3d.snout`
alias of *Snout*

`domonic.xml.x3d.solidOfRevolution`
alias of *SolidOfRevolution*

`domonic.xml.x3d.sound`
alias of *Sound*

class `domonic.xml.x3d.sphere(*args, **kwargs)`

`domonic.xml.x3d.sphereSegment`
alias of *SphereSegment*

`domonic.xml.x3d.sphereSensor`
alias of *SphereSensor*

`domonic.xml.x3d.splinePositionInterpolator`
alias of *SplinePositionInterpolator*

`domonic.xml.x3d.spotLight`
alias of *SpotLight*

`domonic.xml.x3d.staticGroup`
alias of *StaticGroup*

`domonic.xml.x3d.stippleVolumeStyle`
alias of *StippleVolumeStyle*

`domonic.xml.x3d.surfaceShaderTexture`
alias of *SurfaceShaderTexture*

`domonic.xml.x3d.switch`
alias of *Switch*

`domonic.xml.x3d.texCoordDamper2D`
alias of *TexCoordDamper2D*

`domonic.xml.x3d.text`
alias of *Text*

`domonic.xml.x3d.texture`
alias of *Texture*

`domonic.xml.x3d.textureCoordinate`
alias of *TextureCoordinate*

`domonic.xml.x3d.textureCoordinate3D`
alias of *TextureCoordinate3D*

`domonic.xml.x3d.textureCoordinateGenerator`
alias of *TextureCoordinateGenerator*

`domonic.xml.x3d.textureProperties`
alias of *TextureProperties*

`domonic.xml.x3d.textureTransform`
alias of *TextureTransform*

`domonic.xml.x3d.textureTransform3D`
alias of *TextureTransform3D*

`domonic.xml.x3d.textureTransformMatrix3D`
alias of *TextureTransformMatrix3D*

class `domonic.xml.x3d.timeSensor(*args, **kwargs)`

`domonic.xml.x3d.toneMappedVolumeStyle`
alias of *ToneMappedVolumeStyle*

`domonic.xml.x3d.torus`
alias of *Torus*

`domonic.xml.x3d.touchSensor`
alias of *TouchSensor*

class `domonic.xml.x3d.transform(*args, **kwargs)`

`domonic.xml.x3d.triangleSet`
alias of *TriangleSet*

`domonic.xml.x3d.triangleSet2D`
alias of *TriangleSet2D*

`domonic.xml.x3d.twoSidedMaterial`
alias of *TwoSidedMaterial*

`domonic.xml.x3d.uniform`
alias of *Uniform*

`domonic.xml.x3d.universalJoint`
alias of *UniversalJoint*

`domonic.xml.x3d.viewfrustum`
alias of *Viewfrustum*

`domonic.xml.x3d.viewpoint`
alias of *Viewpoint*

`domonic.xml.x3d.volumeData`
alias of *VolumeData*

`domonic.xml.x3d.worldInfo`
alias of *WorldInfo*

`domonic.xml.x3d.x3DAppearanceChildNode`
alias of *X3DAppearanceChildNode*

`domonic.xml.x3d.x3DAppearanceNode`
alias of *X3DAppearanceNode*

`domonic.xml.x3d.x3DBackgroundNode`
alias of *X3DBackgroundNode*

`domonic.xml.x3d.x3DBinaryContainerGeometryNode`
alias of *X3DBinaryContainerGeometryNode*

`domonic.xml.x3d.x3DBindableNode`
alias of *X3DBindableNode*

`domonic.xml.x3d.x3DBoundedObject`
alias of *X3DBoundedObject*

`domonic.xml.x3d.x3DChaserNode`
alias of *X3DChaserNode*

`domonic.xml.x3d.x3DChildNode`
alias of *X3DChildNode*

`domonic.xml.x3d.x3DColorNode`
alias of *X3DColorNode*

`domonic.xml.x3d.x3DComposableVolumeRenderStyleNode`
alias of *X3DComposableVolumeRenderStyleNode*

`domonic.xml.x3d.x3DComposedGeometryNode`
alias of *X3DComposedGeometryNode*

`domonic.xml.x3d.x3DCoordinateNode`
alias of *X3DCoordinateNode*

`domonic.xml.x3d.x3DDamperNode`
alias of *X3DDamperNode*

`domonic.xml.x3d.x3DDragSensorNode`
alias of *X3DDragSensorNode*

`domonic.xml.x3d.x3DEnvironmentNode`
alias of *X3DEnvironmentNode*

`domonic.xml.x3d.x3DEnvironmentTextureNode`
alias of *X3DEnvironmentTextureNode*

`domonic.xml.x3d.x3DFogNode`
alias of *X3DFogNode*

`domonic.xml.x3d.x3DFollowerNode`
alias of *X3DFollowerNode*

`domonic.xml.x3d.x3DFontStyleNode`
alias of *X3DFontStyleNode*

`domonic.xml.x3d.x3DGeometricPropertyNode`
alias of *X3DGeometricPropertyNode*

`domonic.xml.x3d.x3DGeometryNode`
alias of *X3DGeometryNode*

`domonic.xml.x3d.x3DGroupingNode`
alias of *X3DGroupingNode*

`domonic.xml.x3d.x3DInfoNode`
alias of *X3DInfoNode*

`domonic.xml.x3d.x3DInterpolatorNode`
alias of *X3DInterpolatorNode*

`domonic.xml.x3d.x3DLODNode`
alias of *X3DLODNode*

`domonic.xml.x3d.x3DLightNode`
alias of *X3DLightNode*

`domonic.xml.x3d.x3DMaterialNode`
alias of *X3DMaterialNode*

`domonic.xml.x3d.x3DMetadataObject`
alias of *X3DMetadataObject*

`domonic.xml.x3d.x3DNBodyCollidableNode`
alias of *X3DNBodyCollidableNode*

`domonic.xml.x3d.x3DNavigationInfoNode`
alias of *X3DNavigationInfoNode*

`domonic.xml.x3d.x3DNode`
alias of *X3DNode*

`domonic.xml.x3d.x3DPlanarGeometryNode`
alias of *X3DPlanarGeometryNode*

`domonic.xml.x3d.x3DPointingDeviceSensorNode`
alias of *X3DPointingDeviceSensorNode*

`domonic.xml.x3d.x3DRigidJointNode`
alias of *X3DRigidJointNode*

`domonic.xml.x3d.x3DSensorNode`
alias of *X3DSensorNode*

`domonic.xml.x3d.x3DShaderNode`
alias of *X3DShaderNode*

`domonic.xml.x3d.x3DShapeNode`
alias of *X3DShapeNode*

`domonic.xml.x3d.x3DSoundNode`
alias of *X3DSoundNode*

`domonic.xml.x3d.x3DSoundSourceNode`
alias of *X3DSoundSourceNode*

`domonic.xml.x3d.x3DSpatialGeometryNode`
alias of *X3DSpatialGeometryNode*

`domonic.xml.x3d.x3DTexture3DNode`
alias of *X3DTexture3DNode*

`domonic.xml.x3d.x3DTextureCoordinateNode`
alias of *X3DTextureCoordinateNode*

`domonic.xml.x3d.x3DTextureNode`
alias of *X3DTextureNode*

`domonic.xml.x3d.x3DTextureTransformNode`

alias of *X3DTextureTransformNode*

`domonic.xml.x3d.x3DTimeDependentNode`

alias of *X3DTimeDependentNode*

`domonic.xml.x3d.x3DTouchSensorNode`

alias of *X3DTouchSensorNode*

`domonic.xml.x3d.x3DTransformNode`

alias of *X3DTransformNode*

`domonic.xml.x3d.x3DVertexAttributeNode`

alias of *X3DVertexAttributeNode*

`domonic.xml.x3d.x3DViewpointNode`

alias of *X3DViewpointNode*

`domonic.xml.x3d.x3DVolumeDataNode`

alias of *X3DVolumeDataNode*

`domonic.xml.x3d.x3DVolumeRenderStyleNode`

alias of *X3DVolumeRenderStyleNode*

class `domonic.xml.x3d.x3d(*args, **kwargs)`

domonic.geom

written by.ai

class `domonic.geom.matrix(m)`

[matrixs]

rotate(*pt*)

Rotates the point on the vector defined by the vectors m[0] and m[1].

scale(*pt*)

Scales the point on the vector defined by the vectors m[0] and m[1].

translate(*pt*)

Translates the point on the vector defined by the vectors m[0] and m[1].

domonic.sitemap

generate or load sitemaps

warning - when using image and video tags from this package they will be namespaced i.e <image:image> and <video:video> so i'd advise to only import them within the def that you use them in to avoid conflict with html.image

class `domonic.xml.sitemap.changefreq(*args, **kwargs)`

class `domonic.xml.sitemap.lastmod(*args, **kwargs)`

class `domonic.xml.sitemap.loc(*args, **kwargs)`

class `domonic.xml.sitemap.priority(*args, **kwargs)`

```
class domonic.xml.sitemap.sitemap(*args, **kwargs)
```

```
domonic.xml.sitemap.sitemap_format(self, *args, **kwargs)
```

attempts to prettify the output of the sitemap.

```
domonic.xml.sitemap.sitemap_from_urls(urls)
```

Create a sitemap from a list of urls.add()

Note: This won't allow you to add priority or changefreq of the urls. or add images etc tho u could loop the nodes afterwards and do that.

WARNING: there's a difference between a sitemap index and a sitemap. make sure you know what you want.

```
class domonic.xml.sitemap.sitemapindex(*args, **kwargs)
```

```
domonic.xml.sitemap.sitemapindex_from_urls(urls)
```

Create a sitemap index from a list of urls.

WARNING: there's a difference between a sitemap index and a sitemap. make sure you know what you want.

i.e

```
# <?xml version="1.0" encoding="UTF-8"?> # <sitemapindex xmlns="http://www.sitemaps.org/
schemas/sitemap/0.9"> # <sitemap> # <loc>https://xyz.com/sitemap1.xml</loc> # <lastmod>2021-07-
08T13:12:16+00:00</lastmod> # </sitemap> # </sitemapindex>
```

```
class domonic.xml.sitemap.url(*args, **kwargs)
```

```
class domonic.xml.sitemap.urlset(*args, **kwargs)
```

domonic.webapi.url

<https://developer.mozilla.org/en-US/docs/Web/API/URL>

TODO - move the unit tests for this class from javascript to webapi # TODO - untested

```
class domonic.webapi.url.URL(url: str = "", *args, **kwargs)
```

a-tag extends from URL

property hash

” hash Sets or returns the anchor part (#) of a URL

```
class domonic.webapi.url.URLSearchParams(paramString)
```

[utility methods to work with the query string of a URL]

```
append(key, value)
```

Appends a specified key/value pair as a new search parameter

```
delete(key)
```

Deletes the given search parameter, and its associated value, from the list of all search parameters.

```
entries()
```

Returns an iterator allowing iteration through all key/value pairs contained in this object.

```
forEach(func)
```

Allows iteration through all values contained in this object via a callback function.

```
get(key)
```

Returns the first value associated with the given search parameter.

getAll(*key*)

Returns all the values associated with a given search parameter.

has(*key*)

Returns a Boolean indicating if such a given parameter exists.

keys()

Returns an iterator allowing iteration through all keys of the key/value pairs contained in this object.

set(*key*, *value*)

Sets the value associated with a given search parameter to the given value. If there are several values, the others are deleted.

sort()

Sorts all key/value pairs, if any, by their keys.

toString()

Returns a string containing a query string suitable for use in a URL.

values()

Returns an iterator allowing iteration through all values of the key/value pairs contained in this object.

domonic.webapi.fetch

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

1.4.23 Contribute

domonic is an open-source port of the popular html/dom/js api's to Python.

I actively encourages new and experienced users to join in and contribute.

How can you contribute?

Reporting bugs

If you find a bug, an issue where the product doesn't behave as you expect, please file a bug on the github issue tracker.

Fixing Bugs

domonic has a list of issues in the github issue tracker perhaps comment on one if you'd like to help. Feel free to also create an issue and ask for support.

Writing Pull Requests

To contribute to the domonic project, you can create a pull request.

If you have added or fixed any code just send a pull request to the github repository.

If all works out, I will merge your pull request as soon as possible.

Docs

The docs get more traffic than the repo. So any support here is also welcome. All methods could do with a working example in the docs.

We use Sphinx for the docs. It's easy to publish.

Just edit the .rst files then navigate to the docs folder and run *make html*.

Writing tests

Test coverage is a huge plus as it fixes bugs and increases the quality of the code.

More information

Check the notes in the README.md file. and in the CONTRIBUTING.md file.

Firstly pull the repo and try to run the tests. Try with 'make test'.

Read the Makefile to see some examples of running the tests.

1.5 Join-In

Feel free to join in if you find it useful.

If there's any methods you want that are missing or not complete yet. Just update the code and send a pull request.

I'll merge and release asap.

1.6 EXAMPLE PROJECTS

A browser based file browser. Working example of how components can work: <https://github.com/byteface/Blueberry/>

A cron viewer: <https://github.com/byteface/ezcron/>

1.7 Disclaimer

There's several more widely supported libraries doing HTML generation, DOM reading/manipulation, terminal wrappers etc. Maybe use one of those for production due to strictness and support.

This is becoming more of a fast prototyping library.

-
- `genindex`
 - `modindex`
 - `search`

PYTHON MODULE INDEX

d

- `domonic`, 242
- `domonic.CDN`, 297
- `domonic.decorators`, 325
- `domonic.dom`, 247
- `domonic.events`, 299
- `domonic.geom`, 349
- `domonic.html`, 242
- `domonic.javascript`, 267
- `domonic.JSON`, 296
- `domonic.lerpy.easing`, 328
- `domonic.lerpy.tween`, 328
- `domonic.style`, 307
- `domonic.svg`, 325
- `domonic.terminal`, 288
- `domonic.utils`, 320
- `domonic.webapi.fetch`, 351
- `domonic.webapi.url`, 350
- `domonic.xml.sitemap`, 349
- `domonic.xml.x3d`, 328

A

- `a` (class in *domonic.html*), 242
- `abbr` (class in *domonic.html*), 242
- `ABORT` (*domonic.events.Event* attribute), 300
- `ABORT` (*domonic.events.ProgressEvent* attribute), 305
- `ABORT` (*domonic.events.SVGEvent* attribute), 305
- `acronym()` (*domonic.utils.Utils* static method), 320
- `add()` (*domonic.dom.DOMTokenList* method), 249
- `address` (class in *domonic.html*), 242
- `addRule()` (*domonic.style.CSSStyleSheet* method), 309
- `AFRAME_1_2` (*domonic.CDN.CDN_JS* attribute), 298
- `after()` (*domonic.dom.CharacterData* method), 247
- `after()` (*domonic.dom.ChildNode* method), 248
- `AFTERPRINT` (*domonic.events.Event* attribute), 300
- `alert()` (*domonic.javascript.Window* static method), 286
- `alias` (class in *domonic.terminal*), 288
- `alignContent` (*domonic.style.Style* property), 310
- `alignItems` (*domonic.style.Style* property), 310
- `alignSelf` (*domonic.style.Style* property), 310
- `altGlyph` (class in *domonic.svg*), 326
- `altGlyphDef` (class in *domonic.svg*), 326
- `altGlyphItem` (class in *domonic.svg*), 326
- `Anchor` (class in *domonic.xml.x3d*), 328
- `anchor` (in module *domonic.xml.x3d*), 337
- `anchors` (*domonic.dom.Document* property), 249
- `animate` (class in *domonic.svg*), 326
- `animateColor` (class in *domonic.svg*), 326
- `animateMotion` (class in *domonic.svg*), 326
- `animateTransform` (class in *domonic.svg*), 326
- `animation` (*domonic.style.Style* property), 310
- `animationDelay` (*domonic.style.Style* property), 310
- `animationDirection` (*domonic.style.Style* property), 310
- `animationDuration` (*domonic.style.Style* property), 310
- `ANIMATIONEND` (*domonic.events.AnimationEvent* attribute), 299
- `AnimationEvent` (class in *domonic.events*), 299
- `animationFillMode` (*domonic.style.Style* property), 310
- `ANIMATIONITERATION` (*domonic.events.AnimationEvent* attribute), 299
- `animationIterationCount` (*domonic.style.Style* property), 311
- `animationName` (*domonic.events.AnimationEvent* attribute), 299
- `animationName` (*domonic.style.Style* property), 311
- `animationPlayState` (*domonic.style.Style* property), 311
- `ANIMATIONSTART` (*domonic.events.AnimationEvent* attribute), 299
- `animationTimingFunction` (*domonic.style.Style* property), 311
- `appearance` (class in *domonic.xml.x3d*), 337
- `Appearance` (in module *domonic.xml.x3d*), 328
- `append()` (*domonic.dom.Element* method), 254
- `append()` (*domonic.webapi.url.URLSearchParams* method), 350
- `appendChild()` (*domonic.dom.Node* method), 263
- `appendData()` (*domonic.dom.CharacterData* method), 247
- `appendMedium()` (*domonic.style.MediaList* method), 310
- `applet` (class in *domonic.html*), 242
- `applets` (*domonic.dom.Document* property), 249
- `apply()` (*domonic.javascript.Function* method), 275
- `apply()` (*domonic.javascript.Reflect* static method), 279
- `apt` (class in *domonic.terminal*), 288
- `ar` (class in *domonic.terminal*), 288
- `Arc2D` (class in *domonic.xml.x3d*), 328
- `arc2D` (in module *domonic.xml.x3d*), 337
- `ArcClose2D` (class in *domonic.xml.x3d*), 328
- `arcClose2D` (in module *domonic.xml.x3d*), 337
- `area` (class in *domonic.html*), 242
- `Array` (class in *domonic.javascript*), 268
- `article` (class in *domonic.html*), 242
- `as_json()` (in module *domonic.decorators*), 325
- `as_signed()` (in module *domonic.javascript*), 287
- `asa` (class in *domonic.terminal*), 288
- `aside` (class in *domonic.html*), 242
- `assignedSlot` (*domonic.dom.Text* property), 266
- `at` (class in *domonic.terminal*), 288
- `at()` (*domonic.javascript.Array* method), 268
- `atob()` (*domonic.javascript.Window* static method), 286

Attr (class in *domonic.dom*), 247
attrib (*domonic.dom.Node* property), 263
attributes (*domonic.dom.DocumentFragment* property), 253
attributes (*domonic.dom.Element* property), 254
audio (class in *domonic.html*), 242
AudioClip (class in *domonic.xml.x3d*), 329
audioClip (in module *domonic.xml.x3d*), 337
availHeight() (*domonic.javascript.Screen* method), 280
availWidth() (*domonic.javascript.Screen* method), 280
awk (class in *domonic.terminal*), 288

B

b (class in *domonic.html*), 242
backfaceVisibility (*domonic.style.Style* property), 311
Background (class in *domonic.xml.x3d*), 329
background (*domonic.style.Style* property), 311
background (in module *domonic.xml.x3d*), 337
backgroundAttachment (*domonic.style.Style* property), 311
backgroundClip (*domonic.style.Style* property), 311
backgroundColor (*domonic.style.Style* property), 311
backgroundImage (*domonic.style.Style* property), 311
backgroundOrigin (*domonic.style.Style* property), 311
backgroundPosition (*domonic.style.Style* property), 311
backgroundRepeat (*domonic.style.Style* property), 311
backgroundSize (*domonic.style.Style* property), 311
BallJoint (class in *domonic.xml.x3d*), 329
ballJoint (in module *domonic.xml.x3d*), 337
BALLOON (*domonic.CDN.CDN_CSS* attribute), 297
banner (class in *domonic.terminal*), 288
base (class in *domonic.html*), 242
basefont (class in *domonic.html*), 243
basename (class in *domonic.terminal*), 288
bash (class in *domonic.terminal*), 288
batch (class in *domonic.terminal*), 289
bc (class in *domonic.terminal*), 289
bdi (class in *domonic.html*), 243
bdo (class in *domonic.html*), 243
before() (*domonic.dom.CharacterData* method), 247
before() (*domonic.dom.ChildNode* method), 248
BEFOREPRINT (*domonic.events.Event* attribute), 300
BEFOREUNLOAD (*domonic.events.BeforeUnloadEvent* attribute), 299
BEFOREUNLOAD (*domonic.events.Event* attribute), 300
BeforeUnloadEvent (class in *domonic.events*), 299
bg (class in *domonic.terminal*), 289
big() (*domonic.javascript.String* method), 280
Billboard (class in *domonic.xml.x3d*), 329
billboard (in module *domonic.xml.x3d*), 337
BinaryGeometry (class in *domonic.xml.x3d*), 329

binaryGeometry (in module *domonic.xml.x3d*), 337
bind (class in *domonic.terminal*), 289
bind() (*domonic.javascript.Function* method), 275
BlendedVolumeStyle (class in *domonic.xml.x3d*), 329
blendedVolumeStyle (in module *domonic.xml.x3d*), 337
BlendMode (class in *domonic.xml.x3d*), 329
blendMode (in module *domonic.xml.x3d*), 337
blink() (*domonic.javascript.String* method), 280
Block (class in *domonic.xml.x3d*), 329
block (in module *domonic.xml.x3d*), 337
blockedURI (*domonic.events.SecurityPolicyViolationEvent* attribute), 306
blockquote (class in *domonic.html*), 243
BLUR (*domonic.events.FocusEvent* attribute), 302
blur() (*domonic.dom.Element* method), 254
body (class in *domonic.html*), 243
body (*domonic.dom.Document* property), 250
bold() (*domonic.javascript.String* method), 280
Boolean (class in *domonic.javascript*), 271
BOOTSTRAP_4 (*domonic.CDN.CDN_CSS* attribute), 297
BOOTSTRAP_4 (*domonic.CDN.CDN_JS* attribute), 298
BOOTSTRAP_5 (*domonic.CDN.CDN_JS* attribute), 298
BOOTSTRAP_5_ALPHA (*domonic.CDN.CDN_CSS* attribute), 297
BOOTSTRAP_5_ALPHA (*domonic.CDN.CDN_JS* attribute), 298
border (*domonic.style.Style* property), 311
borderBottom (*domonic.style.Style* property), 311
borderBottomColor (*domonic.style.Style* property), 311
borderBottomLeftRadius (*domonic.style.Style* property), 311
borderBottomRightRadius (*domonic.style.Style* property), 311
borderBottomStyle (*domonic.style.Style* property), 312
borderBottomWidth (*domonic.style.Style* property), 312
borderCollapse (*domonic.style.Style* property), 312
borderColor (*domonic.style.Style* property), 312
borderImage (*domonic.style.Style* property), 312
borderImageOutset (*domonic.style.Style* property), 312
borderImageRepeat (*domonic.style.Style* property), 312
borderImageSlice (*domonic.style.Style* property), 312
borderImageSource (*domonic.style.Style* property), 312
borderImageWidth (*domonic.style.Style* property), 312
borderLeft (*domonic.style.Style* property), 312
borderLeftColor (*domonic.style.Style* property), 312
borderLeftStyle (*domonic.style.Style* property), 312
borderLeftWidth (*domonic.style.Style* property), 312
borderRadius (*domonic.style.Style* property), 312
borderRight (*domonic.style.Style* property), 312
borderRightColor (*domonic.style.Style* property), 312
borderRightStyle (*domonic.style.Style* property), 312
borderRightWidth (*domonic.style.Style* property), 312
borderSpacing (*domonic.style.Style* property), 313

- `borderStyle` (*domonic.style.Style* property), 313
 - `borderTop` (*domonic.style.Style* property), 313
 - `borderTopColor` (*domonic.style.Style* property), 313
 - `borderTopLeftRadius` (*domonic.style.Style* property), 313
 - `borderTopRightRadius` (*domonic.style.Style* property), 313
 - `borderTopStyle` (*domonic.style.Style* property), 313
 - `borderTopWidth` (*domonic.style.Style* property), 313
 - `borderWidth` (*domonic.style.Style* property), 313
 - `bottom` (*domonic.style.Style* property), 313
 - `BoundaryEnhancementVolumeStyle` (class in *domonic.xml.x3d*), 329
 - `boundaryEnhancementVolumeStyle` (in module *domonic.xml.x3d*), 337
 - `box` (class in *domonic.xml.x3d*), 337
 - `Box` (in module *domonic.xml.x3d*), 329
 - `boxDecorationBreak` (*domonic.style.Style* property), 313
 - `boxShadow` (*domonic.style.Style* property), 313
 - `boxSizing` (*domonic.style.Style* property), 313
 - `br` (class in *domonic.html*), 243
 - `BRYTHON_3_9_5` (*domonic.CDN.CDN_JS* attribute), 298
 - `btoa()` (*domonic.javascript.Window* static method), 286
 - `BufferAccessor` (class in *domonic.xml.x3d*), 329
 - `bufferAccessor` (in module *domonic.xml.x3d*), 337
 - `BufferGeometry` (class in *domonic.xml.x3d*), 329
 - `bufferGeometry` (in module *domonic.xml.x3d*), 337
 - `BufferView` (class in *domonic.xml.x3d*), 329
 - `bufferView` (in module *domonic.xml.x3d*), 337
 - `builtin` (class in *domonic.terminal*), 289
 - `button` (class in *domonic.html*), 243
- ## C
- `CADAssembly` (class in *domonic.xml.x3d*), 329
 - `cADAssembly` (in module *domonic.xml.x3d*), 337
 - `CADFace` (class in *domonic.xml.x3d*), 329
 - `cADFace` (in module *domonic.xml.x3d*), 337
 - `CADLayer` (class in *domonic.xml.x3d*), 329
 - `cADLayer` (in module *domonic.xml.x3d*), 337
 - `CADPart` (class in *domonic.xml.x3d*), 329
 - `cADPart` (in module *domonic.xml.x3d*), 337
 - `cal` (class in *domonic.terminal*), 289
 - `call()` (*domonic.javascript.Function* method), 276
 - `called()` (in module *domonic.decorators*), 325
 - `caller` (class in *domonic.terminal*), 289
 - `CANPLAY` (*domonic.events.Event* attribute), 300
 - `CANPLAYTHROUGH` (*domonic.events.Event* attribute), 300
 - `canvas` (class in *domonic.html*), 243
 - `caption` (class in *domonic.html*), 243
 - `captionSide` (*domonic.style.Style* property), 313
 - `caretPositionFromPoint()` (*domonic.dom.ShadowRoot* method), 266
 - `CartoonVolumeStyle` (class in *domonic.xml.x3d*), 329
 - `cartoonVolumeStyle` (in module *domonic.xml.x3d*), 338
 - `case_camel()` (*domonic.utils.Utils* static method), 320
 - `case_kebab()` (*domonic.utils.Utils* static method), 320
 - `case_snake()` (*domonic.utils.Utils* static method), 320
 - `cat` (class in *domonic.terminal*), 289
 - `cc` (class in *domonic.terminal*), 289
 - `cd` (class in *domonic.terminal*), 289
 - `CDATASection` (class in *domonic.dom*), 247
 - `CDN_CSS` (class in *domonic.CDN*), 297
 - `CDN_IMG` (class in *domonic.CDN*), 298
 - `CDN_JS` (class in *domonic.CDN*), 298
 - `center` (class in *domonic.html*), 243
 - `cflow` (class in *domonic.terminal*), 289
 - `CHANGE` (*domonic.events.Event* attribute), 300
 - `CHANGE` (*domonic.events.HashChangeEvent* attribute), 302
 - `CHANGE` (*domonic.events.InputEvent* attribute), 302
 - `changeFreq` (class in *domonic.xml.sitemap*), 349
 - `CharacterData` (class in *domonic.dom*), 247
 - `characterSet` (*domonic.dom.Document* property), 250
 - `charAt()` (*domonic.javascript.String* method), 281
 - `charCodeAt()` (*domonic.javascript.String* method), 281
 - `charset` (*domonic.dom.Document* property), 250
 - `check()` (in module *domonic.decorators*), 325
 - `chgrp` (class in *domonic.terminal*), 289
 - `childElementCount` (*domonic.dom.Node* property), 263
 - `ChildNode` (class in *domonic.dom*), 248
 - `childNodes` (*domonic.dom.Node* property), 263
 - `childNodes` (*domonic.dom.Text* property), 266
 - `children` (*domonic.dom.Node* property), 263
 - `children` (*domonic.dom.ParentNode* property), 266
 - `chmod` (class in *domonic.terminal*), 289
 - `chown` (class in *domonic.terminal*), 289
 - `chunk()` (*domonic.utils.Utils* static method), 321
 - `chunks()` (*domonic.utils.Utils* static method), 321
 - `circle` (class in *domonic.svg*), 326
 - `Circle2D` (class in *domonic.xml.x3d*), 329
 - `circle2D` (in module *domonic.xml.x3d*), 338
 - `cite` (class in *domonic.html*), 243
 - `cksum` (class in *domonic.terminal*), 289
 - `classList` (*domonic.dom.Element* property), 255
 - `className` (*domonic.dom.Element* property), 255
 - `clean()` (*domonic.utils.Utils* static method), 321
 - `clear` (*domonic.style.Style* property), 313
 - `clear()` (*domonic.javascript.Map* method), 277
 - `clearTimeout()` (*domonic.javascript.Global* static method), 276
 - `clearTimeout()` (*domonic.javascript.Window* method), 286
 - `clearTimeout()` (in module *domonic.javascript*), 287
 - `CLICK` (*domonic.events.MouseEvent* attribute), 304
 - `click()` (*domonic.dom.Element* method), 255

- `clientHeight` (*domonic.dom.Element* property), 255
- `clientId` (*domonic.events.FetchEvent* attribute), 302
- `clientLeft` (*domonic.dom.Element* property), 255
- `clientTop` (*domonic.dom.Element* property), 255
- `clientWidth` (*domonic.dom.Element* property), 255
- `clip` (*domonic.style.Style* property), 313
- `clipboardData` (*domonic.events.ClipboardEvent* attribute), 299
- `ClipboardEvent` (class in *domonic.events*), 299
- `clipPath` (class in *domonic.svg*), 326
- `ClipPlane` (class in *domonic.xml.x3d*), 329
- `clipPlane` (in module *domonic.xml.x3d*), 338
- `cloneNode()` (*domonic.dom.Node* method), 263
- `close()` (*domonic.dom.Document* method), 250
- `closed_tag` (class in *domonic.html*), 243
- `code` (class in *domonic.html*), 243
- `codePointAt()` (*domonic.javascript.String* method), 281
- `col` (class in *domonic.html*), 243
- `colgroup` (class in *domonic.html*), 243
- `CollidableShape` (class in *domonic.xml.x3d*), 329
- `collidableShape` (in module *domonic.xml.x3d*), 338
- `Collision` (class in *domonic.xml.x3d*), 329
- `collision` (in module *domonic.xml.x3d*), 338
- `CollisionCollection` (class in *domonic.xml.x3d*), 329
- `collisionCollection` (in module *domonic.xml.x3d*), 338
- `CollisionSensor` (class in *domonic.xml.x3d*), 329
- `collisionSensor` (in module *domonic.xml.x3d*), 338
- `Color` (class in *domonic.xml.x3d*), 329
- `color` (*domonic.style.Style* property), 313
- `color` (in module *domonic.xml.x3d*), 338
- `ColorChaser` (class in *domonic.xml.x3d*), 329
- `colorChaser` (in module *domonic.xml.x3d*), 338
- `ColorDamper` (class in *domonic.xml.x3d*), 329
- `colorDamper` (in module *domonic.xml.x3d*), 338
- `colorDepth()` (*domonic.javascript.Screen* method), 280
- `ColorInterpolator` (class in *domonic.xml.x3d*), 329
- `colorInterpolator` (in module *domonic.xml.x3d*), 338
- `ColorMaskMode` (class in *domonic.xml.x3d*), 329
- `colorMaskMode` (in module *domonic.xml.x3d*), 338
- `ColorRGBA` (class in *domonic.xml.x3d*), 329
- `colorRGBA` (in module *domonic.xml.x3d*), 338
- `columnCount` (*domonic.style.Style* property), 313
- `columnFill` (*domonic.style.Style* property), 313
- `columnGap` (*domonic.style.Style* property), 314
- `columnRule` (*domonic.style.Style* property), 314
- `columnRuleColor` (*domonic.style.Style* property), 314
- `columnRuleStyle` (*domonic.style.Style* property), 314
- `columnRuleWidth` (*domonic.style.Style* property), 314
- `columns` (*domonic.style.Style* property), 314
- `columnSpan` (*domonic.style.Style* property), 314
- `columnWidth` (*domonic.style.Style* property), 314
- `comm` (class in *domonic.terminal*), 289
- `command` (class in *domonic.html*), 243
- `command` (class in *domonic.terminal*), 289
- `Comment` (class in *domonic.dom*), 248
- `comment` (class in *domonic.html*), 243
- `CommonSurfaceShader` (class in *domonic.xml.x3d*), 330
- `commonSurfaceShader` (in module *domonic.xml.x3d*), 338
- `compareDocumentPosition()` (*domonic.dom.Node* method), 263
- `compgen` (class in *domonic.terminal*), 289
- `compile()` (*domonic.javascript.String* method), 281
- `complete` (class in *domonic.terminal*), 289
- `COMPLETE` (*domonic.events.TweenEvent* attribute), 307
- `compopt` (class in *domonic.terminal*), 289
- `ComposedCubeMapTexture` (class in *domonic.xml.x3d*), 330
- `composedCubeMapTexture` (in module *domonic.xml.x3d*), 338
- `ComposedShader` (class in *domonic.xml.x3d*), 330
- `composedShader` (in module *domonic.xml.x3d*), 338
- `ComposedTexture3D` (class in *domonic.xml.x3d*), 330
- `composedTexture3D` (in module *domonic.xml.x3d*), 338
- `ComposedVolumeStyle` (class in *domonic.xml.x3d*), 330
- `composedVolumeStyle` (in module *domonic.xml.x3d*), 338
- `CompositionEvent` (class in *domonic.events*), 299
- `compress` (class in *domonic.terminal*), 289
- `concat()` (*domonic.javascript.Array* method), 268
- `concat()` (*domonic.javascript.String* method), 281
- `Cone` (class in *domonic.xml.x3d*), 330
- `cone` (in module *domonic.xml.x3d*), 338
- `CONNECT` (*domonic.events.MessageEvent* attribute), 303
- `console` (in module *domonic.dom*), 267
- `construct()` (*domonic.javascript.Reflect* static method), 279
- `contains()` (*domonic.dom.DOMTokenList* method), 249
- `contains()` (*domonic.dom.Node* method), 263
- `content` (*domonic.style.Style* property), 314
- `contentEditable` (*domonic.dom.Element* property), 255
- `CONTEXTMENU` (*domonic.events.MouseEvent* attribute), 304
- `convert` (class in *domonic.terminal*), 289
- `Coordinate` (class in *domonic.xml.x3d*), 330
- `coordinate` (in module *domonic.xml.x3d*), 338
- `CoordinateDamper` (class in *domonic.xml.x3d*), 330
- `coordinateDamper` (in module *domonic.xml.x3d*), 339
- `CoordinateDouble` (class in *domonic.xml.x3d*), 330
- `coordinateDouble` (in module *domonic.xml.x3d*), 339
- `CoordinateInterpolator` (class in *domonic.xml.x3d*), 330
- `coordinateInterpolator` (in module *domonic.xml.x3d*), 339
- `COPY` (*domonic.events.ClipboardEvent* attribute), 299

- copyWithin() (*domonic.javascript.Array* method), 268
 counterIncrement (*domonic.style.Style* property), 314
 counterReset (*domonic.style.Style* property), 314
 cowsay (*class in domonic.terminal*), 289
 cp (*class in domonic.terminal*), 290
 create_element() (*in module domonic.html*), 243
 createAttribute() (*domonic.dom.Document* static method), 250
 createCDATASection() (*domonic.dom.Document* static method), 250
 createComment() (*domonic.dom.Document* static method), 250
 createDocumentFragment() (*domonic.dom.Document* static method), 250
 createElement() (*domonic.dom.Document* static method), 250
 createElementNS() (*domonic.dom.Document* static method), 250
 createEntityReference() (*domonic.dom.Document* static method), 250
 createEvent() (*domonic.dom.Document* static method), 250
 createExpression() (*domonic.dom.Document* static method), 250
 createNodeIterator() (*domonic.dom.Document* static method), 250
 createProcessingInstruction() (*domonic.dom.Document* static method), 250
 createRange() (*domonic.dom.Document* static method), 250
 createTextNode() (*domonic.dom.Document* static method), 251
 createTreeWalker() (*domonic.dom.Document* static method), 251
 cron (*class in domonic.terminal*), 290
 crontab (*class in domonic.terminal*), 290
 csplit (*class in domonic.terminal*), 290
 CSSColorProfileRule (*class in domonic.style*), 308
 CSSConditionRule (*class in domonic.style*), 308
 CSSCounterStyleRule (*class in domonic.style*), 308
 CSSDocumentRule (*class in domonic.style*), 308
 cssFloat (*domonic.style.Style* property), 314
 CSSFontFaceRule (*class in domonic.style*), 308
 CSSFontFeatureValuesRule (*class in domonic.style*), 308
 CSSGroupingRule (*class in domonic.style*), 308
 CSSImportRule (*class in domonic.style*), 308
 CSSKeyframeRule (*class in domonic.style*), 308
 CSSKeyframesRule (*class in domonic.style*), 308
 CSSMediaRule (*class in domonic.style*), 308
 CSSNamespaceRule (*class in domonic.style*), 308
 CSSPageRule (*class in domonic.style*), 308
 CSSRule (*class in domonic.style*), 308
 CSSRuleList (*class in domonic.style*), 308
 CSSStyleDeclaration (*class in domonic.style*), 309
 CSSStyleRule (*class in domonic.style*), 309
 CSSStyleSheet (*class in domonic.style*), 309
 CSSSupportsRule (*class in domonic.style*), 310
 cssText (*domonic.style.CSSRule* property), 308
 cssText (*domonic.style.CSSStyleRule* property), 309
 csv2json() (*in module domonic.JSON*), 296
 csvify() (*in module domonic.JSON*), 296
 ctags (*class in domonic.terminal*), 290
 curl (*class in domonic.terminal*), 290
 cursor (*class in domonic.svg*), 326
 cursor (*domonic.style.Style* property), 314
 CustomEvent (*class in domonic.events*), 299
 cut (*class in domonic.terminal*), 290
 CUT (*domonic.events.ClipboardEvent* attribute), 299
 cxref (*class in domonic.terminal*), 290
 Cylinder (*class in domonic.xml.x3d*), 330
 cylinder (*in module domonic.xml.x3d*), 339
 CylinderSensor (*class in domonic.xml.x3d*), 330
 cylinderSensor (*in module domonic.xml.x3d*), 339
- ## D
- D3 (*domonic.CDN.CDN_JS* attribute), 298
 D3_6_1_0 (*domonic.CDN.CDN_JS* attribute), 298
 data (*class in domonic.html*), 243
 data (*domonic.events.InputEvent* attribute), 303
 data (*domonic.events.MessageEvent* attribute), 303
 datalist (*class in domonic.html*), 243
 dataset (*domonic.dom.Element* property), 255
 dataTransfer (*domonic.events.DragEvent* attribute), 300
 dataTransfer (*domonic.events.InputEvent* attribute), 303
 Date (*class in domonic.javascript*), 271
 date (*class in domonic.terminal*), 290
 DBLCLICK (*domonic.events.MouseEvent* attribute), 304
 dd (*class in domonic.html*), 243
 dd (*class in domonic.terminal*), 290
 declare (*class in domonic.terminal*), 290
 decodeURI() (*domonic.javascript.Global* static method), 276
 decodeURI() (*in module domonic.javascript*), 287
 decodeURIComponent() (*domonic.javascript.Global* static method), 276
 decodeURIComponent() (*in module domonic.javascript*), 287
 defineProperty() (*domonic.javascript.Reflect* static method), 279
 defs (*class in domonic.svg*), 326
 delete() (*domonic.dom.DOMStringMap* method), 249
 delete() (*domonic.javascript.Map* method), 277
 delete() (*domonic.webapi.url.URLSearchParams* method), 350

`deleteData()` (*domonic.dom.CharacterData* method), 248

`deleteMedium()` (*domonic.style.MediaList* method), 310

`deleteProperty()` (*domonic.javascript.Reflect* static method), 279

`deleteRule()` (*domonic.style.CSSStyleSheet* method), 309

`delta` (class in *domonic.terminal*), 290

`deprecated()` (in module *domonic.decorators*), 325

`DepthMode` (class in *domonic.xml.x3d*), 330

`depthMode` (in module *domonic.xml.x3d*), 339

`desc` (class in *domonic.svg*), 326

`details` (class in *domonic.html*), 243

`df` (class in *domonic.terminal*), 290

`dfn` (class in *domonic.html*), 243

`dialog` (class in *domonic.html*), 243

`dictify()` (*domonic.utils.Utils* static method), 321

`diff` (class in *domonic.terminal*), 290

`digits()` (*domonic.utils.Utils* static method), 321

`dir` (*domonic.dom.Element* property), 255

`direction` (*domonic.style.Style* property), 314

`DirectionalLight` (class in *domonic.xml.x3d*), 330

`directionalLight` (in module *domonic.xml.x3d*), 339

`dirname` (class in *domonic.terminal*), 290

`dirs` (class in *domonic.terminal*), 290

`discard` (class in *domonic.svg*), 326

`DISCONNECT` (*domonic.events.MessageEvent* attribute), 303

`Dish` (class in *domonic.xml.x3d*), 330

`dish` (in module *domonic.xml.x3d*), 339

`Disk2D` (class in *domonic.xml.x3d*), 330

`disk2D` (in module *domonic.xml.x3d*), 339

`disown` (class in *domonic.terminal*), 290

`display` (*domonic.style.Style* property), 314

`div` (class in *domonic.html*), 243

`div()` (*domonic.javascript.String* method), 281

`dl` (class in *domonic.html*), 243

`doctype` (class in *domonic.html*), 243

`doctype` (*domonic.dom.Document* property), 251

`Document` (class in *domonic.dom*), 249

`document` (*domonic.events.DOMContentLoadedEvent* attribute), 300

`DocumentFragment` (class in *domonic.dom*), 253

`DocumentType` (class in *domonic.dom*), 254

`DOM_KEY_LOCATION_JOYSTICK` (*domonic.events.KeyboardEvent* attribute), 303

`DOM_KEY_LOCATION_LEFT` (*domonic.events.KeyboardEvent* attribute), 303

`DOM_KEY_LOCATION_MOBILE` (*domonic.events.KeyboardEvent* attribute), 303

`DOM_KEY_LOCATION_NUMPAD` (*domonic.events.KeyboardEvent* attribute), 303

`DOM_KEY_LOCATION_RIGHT` (*domonic.events.KeyboardEvent* attribute), 303

`DOM_KEY_LOCATION_STANDARD` (*domonic.events.KeyboardEvent* attribute), 303

`domain()` (*domonic.dom.Document* method), 251

`DOMConfig` (class in *domonic.dom*), 248

`domConfig()` (*domonic.dom.Document* method), 251

`DOMCONTENTLOADED` (*domonic.events.DOMContentLoadedEvent* attribute), 300

`DOMContentLoadedEvent` (class in *domonic.events*), 300

`DOMException`, 248

`domonic`
 module, 242

`domonic.CDN`
 module, 297

`domonic.decorators`
 module, 325

`domonic.dom`
 module, 247

`domonic.events`
 module, 299

`domonic.geom`
 module, 349

`domonic.html`
 module, 242

`domonic.javascript`
 module, 267

`domonic.JSON`
 module, 296

`domonic.lerpy.easing`
 module, 328

`domonic.lerpy.tween`
 module, 328

`domonic.style`
 module, 307

`domonic.svg`
 module, 325

`domonic.terminal`
 module, 288

`domonic.utils`
 module, 320

`domonic.webapi.fetch`
 module, 351

`domonic.webapi.url`
 module, 350

`domonic.xml.sitemap`
 module, 349

`domonic.xml.x3d`

- module, 328
- DOMPoint (class in domonic.dom), 249
- DOMPointReadOnly (class in domonic.dom), 249
- DOMQuad (class in domonic.dom), 249
- DOMStringMap (class in domonic.dom), 249
- DOMTimeStamp (class in domonic.dom), 249
- DOMTokenList (class in domonic.dom), 249
- DoubleAxisHingeJoint (class in domonic.xml.x3d), 330
- doubleAxisHingeJoint (in module domonic.xml.x3d), 339
- DRAG (domonic.events.DragEvent attribute), 300
- DragEvent (class in domonic.events), 300
- DROP (domonic.events.DragEvent attribute), 300
- dt (class in domonic.html), 244
- du (class in domonic.terminal), 290
- DURATIONCHANGE (domonic.events.Event attribute), 300
- DynamicLOD (class in domonic.xml.x3d), 330
- dynamicLOD (in module domonic.xml.x3d), 339
- E**
- echo (class in domonic.terminal), 290
- ed (class in domonic.terminal), 290
- EdgeEnhancementVolumeStyle (class in domonic.xml.x3d), 330
- edgeEnhancementVolumeStyle (in module domonic.xml.x3d), 339
- el() (in module domonic.decorators), 325
- elapsedTime (domonic.events.AnimationEvent attribute), 299
- elapsedTime (domonic.events.TransitionEvent attribute), 307
- Element (class in domonic.dom), 254
- elementFromPoint() (domonic.dom.Document method), 251
- elementFromPoint() (domonic.dom.ShadowRoot method), 266
- elementsFromPoint() (domonic.dom.Document method), 251
- elementsFromPoint() (domonic.dom.ShadowRoot method), 266
- ElevationGrid (class in domonic.xml.x3d), 330
- elevationGrid (in module domonic.xml.x3d), 339
- ellipse (class in domonic.svg), 326
- em (class in domonic.html), 244
- embed (class in domonic.html), 244
- embeds (domonic.dom.Document property), 251
- EMPTYED (domonic.events.Event attribute), 300
- emptyCells (domonic.style.Style property), 314
- enable (class in domonic.terminal), 290
- encodeURIComponent (domonic.javascript.Global static method), 276
- encodeURIComponent (in module domonic.javascript), 287
- encodeURIComponent() (domonic.javascript.Global static method), 276
- encodeURIComponent() (in module domonic.javascript), 287
- END (domonic.events.DragEvent attribute), 300
- ENDED (domonic.events.Event attribute), 300
- endsWith() (domonic.javascript.String method), 281
- ENTER (domonic.events.DragEvent attribute), 300
- Entity (class in domonic.dom), 259
- EntityReference (class in domonic.dom), 259
- entries() (domonic.dom.NodeList method), 265
- entries() (domonic.javascript.Array method), 268
- entries() (domonic.javascript.Map method), 277
- entries() (domonic.webapi.url.URLSearchParams method), 350
- env (class in domonic.terminal), 290
- Environment (class in domonic.xml.x3d), 330
- environment (in module domonic.xml.x3d), 339
- Error, 275
- ERROR (domonic.events.ErrorEvent attribute), 300
- ERROR (domonic.events.Event attribute), 300
- ERROR (domonic.events.ProgressEvent attribute), 305
- ERROR (domonic.events.SVGEvent attribute), 305
- ErrorEvent (class in domonic.events), 300
- escape() (domonic.utils.Utils static method), 321
- eval() (domonic.javascript.Global static method), 276
- evaluate() (domonic.dom.Document method), 251
- Event (class in domonic.events), 300
- EventDispatcher (in module domonic.events), 302
- EventTarget (class in domonic.events), 302
- every() (domonic.javascript.Array method), 268
- ex (class in domonic.terminal), 290
- exit (class in domonic.terminal), 290
- EXIT (domonic.events.DragEvent attribute), 300
- exitFullscreen() (domonic.dom.Element method), 255
- expand (class in domonic.terminal), 290
- export (class in domonic.terminal), 290
- expr (class in domonic.terminal), 290
- extendable (domonic.events.ExtendableEvent attribute), 302
- ExtendableEvent (class in domonic.events), 302
- Extrusion (class in domonic.xml.x3d), 330
- extrusion (in module domonic.xml.x3d), 339
- F**
- fc (class in domonic.terminal), 290
- feBlend (class in domonic.svg), 326
- feColorMatrix (class in domonic.svg), 326
- feComponentTransfer (class in domonic.svg), 326
- feComposite (class in domonic.svg), 326
- feConvolveMatrix (class in domonic.svg), 326
- feDiffuseLighting (class in domonic.svg), 326
- feDisplacementMap (class in domonic.svg), 326

- `feDistantLight` (class in *domonic.svg*), 326
- `feDropShadow` (class in *domonic.svg*), 326
- `feFlood` (class in *domonic.svg*), 326
- `feFuncA` (class in *domonic.svg*), 326
- `feFuncB` (class in *domonic.svg*), 326
- `feFuncG` (class in *domonic.svg*), 326
- `feFuncR` (class in *domonic.svg*), 326
- `feGaussianBlur` (class in *domonic.svg*), 327
- `feImage` (class in *domonic.svg*), 327
- `feMerge` (class in *domonic.svg*), 327
- `feMergeNode` (class in *domonic.svg*), 327
- `feMorphology` (class in *domonic.svg*), 327
- `feOffset` (class in *domonic.svg*), 327
- `fePointLight` (class in *domonic.svg*), 327
- `feSpecularLighting` (class in *domonic.svg*), 327
- `feSpotLight` (class in *domonic.svg*), 327
- `FETCH` (*domonic.events.FetchEvent* attribute), 302
- `FetchEvent` (class in *domonic.events*), 302
- `feTile` (class in *domonic.svg*), 327
- `feTurbulence` (class in *domonic.svg*), 327
- `ffmpeg` (class in *domonic.terminal*), 290
- `fg` (class in *domonic.terminal*), 290
- `Field` (class in *domonic.xml.x3d*), 330
- `field` (in module *domonic.xml.x3d*), 339
- `fieldset` (class in *domonic.html*), 244
- `figcaption` (class in *domonic.html*), 244
- `figlet` (class in *domonic.terminal*), 290
- `figure` (class in *domonic.html*), 244
- `file` (class in *domonic.terminal*), 291
- `fill()` (*domonic.javascript.Array* method), 268
- `filter` (*domonic.style.Style* property), 314
- `filter()` (*domonic.javascript.Array* method), 268
- `find` (class in *domonic.terminal*), 291
- `find()` (*domonic.javascript.Array* method), 268
- `findIndex()` (*domonic.javascript.Array* method), 268
- `findLast()` (*domonic.javascript.Array* method), 269
- `findLastIndex()` (*domonic.javascript.Array* method), 269
- `finger` (class in *domonic.terminal*), 291
- `firstChild` (*domonic.dom.Node* property), 263
- `firstChild` (*domonic.dom.Text* property), 266
- `firstChild()` (*domonic.dom.TreeWalker* method), 267
- `firstElementChild` (*domonic.dom.ParentNode* property), 266
- `firstElementChild()` (*domonic.dom.Element* method), 255
- `fixed()` (*domonic.javascript.String* method), 282
- `flat()` (*domonic.javascript.Array* method), 269
- `flatMap()` (*domonic.javascript.Array* method), 269
- `flatten()` (in module *domonic.JSON*), 296
- `flex` (*domonic.style.Style* property), 314
- `flexBasis` (*domonic.style.Style* property), 315
- `flexDirection` (*domonic.style.Style* property), 315
- `flexFlow` (*domonic.style.Style* property), 315
- `flexGrow` (*domonic.style.Style* property), 315
- `flexShrink` (*domonic.style.Style* property), 315
- `flexWrap` (*domonic.style.Style* property), 315
- `FloatVertexAttribute` (class in *domonic.xml.x3d*), 330
- `floatVertexAttribute` (in module *domonic.xml.x3d*), 339
- `FOCUS` (*domonic.events.FocusEvent* attribute), 302
- `focus()` (*domonic.dom.Element* method), 255
- `FocusEvent` (class in *domonic.events*), 302
- `FOCUSIN` (*domonic.events.FocusEvent* attribute), 302
- `FOCUSOUT` (*domonic.events.FocusEvent* attribute), 302
- `Fog` (class in *domonic.xml.x3d*), 330
- `fog` (in module *domonic.xml.x3d*), 339
- `fold` (class in *domonic.terminal*), 291
- `font` (class in *domonic.html*), 244
- `font` (class in *domonic.svg*), 327
- `font` (*domonic.style.Style* property), 315
- `FONTAWESOME_5_7_1` (*domonic.CDN.CDN_CSS* attribute), 297
- `fontcolor()` (*domonic.javascript.String* method), 282
- `fontFamily` (*domonic.style.Style* property), 315
- `fontSize` (*domonic.style.Style* property), 315
- `fontsize()` (*domonic.javascript.String* method), 282
- `fontSizeAdjust` (*domonic.style.Style* property), 315
- `fontStretch` (*domonic.style.Style* property), 315
- `FontStyle` (class in *domonic.xml.x3d*), 330
- `fontStyle` (*domonic.style.Style* property), 315
- `fontStyle` (in module *domonic.xml.x3d*), 339
- `fontVariant` (*domonic.style.Style* property), 315
- `fontWeight` (*domonic.style.Style* property), 315
- `footer` (class in *domonic.html*), 244
- `forEach()` (*domonic.dom.NodeList* method), 265
- `forEach()` (*domonic.javascript.Array* method), 269
- `forEach()` (*domonic.webapi.url.URLSearchParams* method), 350
- `foreignObject` (class in *domonic.svg*), 327
- `form` (class in *domonic.html*), 244
- `forms` (*domonic.dom.Document* property), 251
- `fort77` (class in *domonic.terminal*), 291
- `frame` (*domonic.events.SecurityPolicyViolationEvent* attribute), 306
- `frequency()` (*domonic.utils.Utils* static method), 321
- `from_()` (*domonic.javascript.Array* static method), 269
- `fromChar()` (*domonic.dom.Entity* static method), 259
- `fromCharCode()` (*domonic.javascript.String* method), 282
- `fromCodePoint()` (*domonic.javascript.String* static method), 282
- `fromName()` (*domonic.dom.Entity* static method), 259
- `fromOrdinal()` (*domonic.dom.EntityReference* static method), 259
- `FULLSCREENCHANGE` (*domonic.events.Event* attribute), 300

- fullscreenElement() (*domonic.dom.Document* method), 252
 fullscreenEnabled() (*domonic.dom.Document* method), 252
 FULLSCREENERROR (*domonic.events.Event* attribute), 301
 Function (class in *domonic.javascript*), 275
 function() (in module *domonic.javascript*), 288
 fuser (class in *domonic.terminal*), 291
- ## G
- g (class in *domonic.svg*), 327
 GamePadEvent (class in *domonic.events*), 302
 gcc (class in *domonic.terminal*), 291
 gencat (class in *domonic.terminal*), 291
 GeneratedCubeMapTexture (class in *domonic.xml.x3d*), 330
 generatedCubeMapTexture (in module *domonic.xml.x3d*), 340
 GeoCoordinate (class in *domonic.xml.x3d*), 330
 geoCoordinate (in module *domonic.xml.x3d*), 340
 GeoElevationGrid (class in *domonic.xml.x3d*), 330
 geoElevationGrid (in module *domonic.xml.x3d*), 340
 GeoLocation (class in *domonic.xml.x3d*), 330
 geoLocation (in module *domonic.xml.x3d*), 340
 GeoLOD (class in *domonic.xml.x3d*), 330
 geoLOD (in module *domonic.xml.x3d*), 340
 GeoMetadata (class in *domonic.xml.x3d*), 331
 geoMetadata (in module *domonic.xml.x3d*), 340
 GeoOrigin (class in *domonic.xml.x3d*), 331
 geoOrigin (in module *domonic.xml.x3d*), 340
 GeoPositionInterpolator (class in *domonic.xml.x3d*), 331
 geoPositionInterpolator (in module *domonic.xml.x3d*), 340
 GeoTransform (class in *domonic.xml.x3d*), 331
 geoTransform (in module *domonic.xml.x3d*), 340
 GeoViewpoint (class in *domonic.xml.x3d*), 331
 geoViewpoint (in module *domonic.xml.x3d*), 340
 get (class in *domonic.terminal*), 291
 get() (*domonic.dom.DOMStringMap* method), 249
 get() (*domonic.javascript.Map* method), 277
 get() (*domonic.javascript.Reflect* static method), 279
 get() (*domonic.webapi.url.URLSearchParams* method), 350
 get_vowels() (*domonic.utils.Utils* static method), 322
 getAll() (*domonic.webapi.url.URLSearchParams* method), 350
 getAttribute() (*domonic.dom.Element* method), 255
 getAttributeNode() (*domonic.dom.Element* method), 255
 getAttributeNodeNS() (*domonic.dom.Element* method), 255
 getAttributeNS() (*domonic.dom.Element* method), 255
 getBoundingClientRect() (*domonic.dom.Element* method), 255
 getconf (class in *domonic.terminal*), 291
 getDate() (*domonic.javascript.Date* method), 271
 getDay() (*domonic.javascript.Date* method), 271
 getElementById() (*domonic.dom.Document* method), 252
 getElementById() (*domonic.dom.DocumentFragment* method), 253
 getElementsByClassName() (*domonic.dom.Element* method), 255
 getElementsByName() (*domonic.dom.Document* method), 252
 getElementsByTagName() (*domonic.dom.Element* method), 256
 getElementsByTagName() (*domonic.dom.DocumentFragment* method), 254
 getElementsByTagName() (*domonic.dom.Element* method), 256
 getFullYear() (*domonic.javascript.Date* method), 271
 getHours() (*domonic.javascript.Date* method), 272
 getMilliseconds() (*domonic.javascript.Date* method), 272
 getMinutes() (*domonic.javascript.Date* method), 272
 getModifierState() (*domonic.events.MouseEvent* method), 304
 getMonth() (*domonic.javascript.Date* method), 272
 getNamedItem() (*domonic.dom.Attr* method), 247
 getopts (class in *domonic.terminal*), 291
 getOwnPropertyDescriptor() (*domonic.javascript.Reflect* static method), 279
 getPropertyCSSValue() (*domonic.style.CSSStyleDeclaration* method), 309
 getPropertyPriority() (*domonic.style.CSSStyleDeclaration* method), 309
 getPropertyValue() (*domonic.style.CSSStyleDeclaration* method), 309
 getPrototypeOf() (*domonic.javascript.Reflect* method), 279
 getSeconds() (*domonic.javascript.Date* method), 272
 getSelection() (*domonic.dom.ShadowRoot* method), 266
 getTargetRanges() (*domonic.events.InputEvent* method), 303
 getTime() (*domonic.javascript.Date* method), 272
 getTimezoneOffset() (*domonic.javascript.Date* method), 272
 getUTCDate() (*domonic.javascript.Date* method), 272
 getUTCDay() (*domonic.javascript.Date* method), 272
 getUTCFullYear() (*domonic.javascript.Date* method),

272
getUTCHours() (*domonic.javascript.Date* method), 272
getUTCMilliseconds() (*domonic.javascript.Date* method), 272
getUTCMinutes() (*domonic.javascript.Date* method), 272
getUTCMonth() (*domonic.javascript.Date* method), 272
getUTCSeconds() (*domonic.javascript.Date* method), 272
getYear() (*domonic.javascript.Date* method), 272
git (class in *domonic.terminal*), 291
Global (class in *domonic.javascript*), 276
glyph (class in *domonic.svg*), 327
glyphRef (class in *domonic.svg*), 327
grep (class in *domonic.terminal*), 291
Group (class in *domonic.xml.x3d*), 331
group (in module *domonic.xml.x3d*), 340
groupadd (class in *domonic.terminal*), 291
groupBy() (*domonic.javascript.Array* method), 269
groupdel (class in *domonic.terminal*), 291
groups (class in *domonic.terminal*), 291
gunzip (class in *domonic.terminal*), 291
gzip (class in *domonic.terminal*), 291

H

h1 (class in *domonic.html*), 244
h2 (class in *domonic.html*), 244
h3 (class in *domonic.html*), 244
h4 (class in *domonic.html*), 244
h5 (class in *domonic.html*), 244
h6 (class in *domonic.html*), 244
HANDLED (*domonic.events.PromiseRejectionEvent* attribute), 305
hangingPunctuation (*domonic.style.Style* property), 315
HAnimDisplacer (class in *domonic.xml.x3d*), 331
hAnimDisplacer (in module *domonic.xml.x3d*), 340
HAnimHumanoid (class in *domonic.xml.x3d*), 331
hAnimHumanoid (in module *domonic.xml.x3d*), 340
HAnimJoint (class in *domonic.xml.x3d*), 331
hAnimJoint (in module *domonic.xml.x3d*), 340
HAnimSegment (class in *domonic.xml.x3d*), 331
hAnimSegment (in module *domonic.xml.x3d*), 340
HAnimSite (class in *domonic.xml.x3d*), 331
hAnimSite (in module *domonic.xml.x3d*), 340
has() (*domonic.javascript.Map* method), 278
has() (*domonic.javascript.Reflect* static method), 279
has() (*domonic.webapi.url.URLSearchParams* method), 351
has_internet() (*domonic.utils.Utils* static method), 322
hasAttribute() (*domonic.dom.Element* method), 256
hasAttributes() (*domonic.dom.Element* method), 256
hasChildNodes() (*domonic.dom.Node* method), 263

hash (*domonic.webapi.url.URL* property), 350
HashChangeEvent (class in *domonic.events*), 302
hatch (class in *domonic.svg*), 327
hatchpath (class in *domonic.svg*), 327
head (class in *domonic.html*), 244
head (class in *domonic.terminal*), 291
head (*domonic.dom.Document* property), 252
header (class in *domonic.html*), 244
height (*domonic.style.Style* property), 315
height() (*domonic.javascript.Screen* method), 280
hgroup (class in *domonic.html*), 244
history (class in *domonic.terminal*), 291
hkern (class in *domonic.svg*), 327
hr (class in *domonic.html*), 244
html (class in *domonic.html*), 244
HTMLAnchorElement (class in *domonic.dom*), 259
HTMLAreaElement (class in *domonic.dom*), 259
HTMLAudioElement (class in *domonic.dom*), 260
HTMLBaseElement (class in *domonic.dom*), 260
HTMLBaseFontElement (class in *domonic.dom*), 260
HTMLBodyElement (class in *domonic.dom*), 260
HTMLBRElement (class in *domonic.dom*), 260
HTMLButtonElement (class in *domonic.dom*), 260
HTMLCanvasElement (class in *domonic.dom*), 260
HTMLCollection (class in *domonic.dom*), 260
HTMLContentElement (class in *domonic.dom*), 260
HTMLDataElement (class in *domonic.dom*), 260
HTMLDataListElement (class in *domonic.dom*), 260
HTMLDialogElement (class in *domonic.dom*), 260
HTMLDivElement (class in *domonic.dom*), 260
HTMLDListElement (class in *domonic.dom*), 260
HTMLDocument (class in *domonic.dom*), 260
HTMLElement (class in *domonic.dom*), 260
HTMLEmbedElement (class in *domonic.dom*), 260
HTMLFieldSetElement (class in *domonic.dom*), 260
HTMLFormControlsCollection (class in *domonic.dom*), 260
HTMLFormElement (class in *domonic.dom*), 261
HTMLFrameSetElement (class in *domonic.dom*), 261
HTMLHeadElement (class in *domonic.dom*), 261
HTMLHeadingElement (class in *domonic.dom*), 261
HTMLHRElement (class in *domonic.dom*), 261
HTMLIFrameElement (class in *domonic.dom*), 261
HTMLImageElement (class in *domonic.dom*), 261
HTMLInputElement (class in *domonic.dom*), 261
HTMLIsIndexElement (class in *domonic.dom*), 261
HTMLKeygenElement (class in *domonic.dom*), 261
HTMLLabelElement (class in *domonic.dom*), 261
HTMLLegendElement (class in *domonic.dom*), 261
HTMLLIElement (class in *domonic.dom*), 261
HTMLLinkElement (class in *domonic.dom*), 261
HTMLMapElement (class in *domonic.dom*), 261
HTMLMediaElement (class in *domonic.dom*), 261
HTMLMetaElement (class in *domonic.dom*), 261

- HTMLMeterElement (class in domonic.dom), 261
- HTMLModElement (class in domonic.dom), 261
- HTMLObjectElement (class in domonic.dom), 261
- HTMLLOListElement (class in domonic.dom), 261
- HTMLOptGroupElement (class in domonic.dom), 261
- HTMLOptionElement (class in domonic.dom), 262
- HTMLOutputElement (class in domonic.dom), 262
- HTMLParagraphElement (class in domonic.dom), 262
- HTMLParamElement (class in domonic.dom), 262
- HTMLPictureElement (class in domonic.dom), 262
- HTMLPortalElement (class in domonic.dom), 262
- HTMLPreElement (class in domonic.dom), 262
- HTMLProgressElement (class in domonic.dom), 262
- HTMLQuoteElement (class in domonic.dom), 262
- HTMLScriptElement (class in domonic.dom), 262
- HTMLSelectElement (class in domonic.dom), 262
- HTMLShadowElement (class in domonic.dom), 262
- HTMLSourceElement (class in domonic.dom), 262
- HTMLSpanElement (class in domonic.dom), 262
- HTMLStyleElement (class in domonic.dom), 262
- HTMLTableCaptionElement (class in domonic.dom), 262
- HTMLTableCellElement (class in domonic.dom), 262
- HTMLTableColElement (class in domonic.dom), 262
- HTMLTableDataCellElement (class in domonic.dom), 262
- HTMLTableElement (class in domonic.dom), 262
- HTMLTableHeaderCellElement (class in domonic.dom), 262
- HTMLTableRowElement (class in domonic.dom), 262
- HTMLTableSectionElement (class in domonic.dom), 262
- HTMLTemplateElement (class in domonic.dom), 262
- HTMLTextAreaElement (class in domonic.dom), 262
- HTMLTimeElement (class in domonic.dom), 262
- HTMLTitleElement (class in domonic.dom), 263
- HTMLTrackElement (class in domonic.dom), 263
- HTMLULListElement (class in domonic.dom), 263
- HTMLUnknownElement (class in domonic.dom), 263
- HTMLVideoElement (class in domonic.dom), 263
- HTMX (domonic.CDN.CDN_JS attribute), 298
- hyphens (domonic.style.Style property), 315
- hypot() (domonic.javascript.Math static method), 278
- I
- i (class in domonic.html), 244
- icon (domonic.style.Style property), 315
- iconv (class in domonic.terminal), 291
- id (domonic.dom.Element property), 257
- ifconfig (class in domonic.terminal), 291
- iframe (class in domonic.html), 244
- iife() (in module domonic.decorators), 325
- image (class in domonic.svg), 327
- imageOrientation (domonic.style.Style property), 315
- images (domonic.dom.Document property), 252
- ImageTexture (class in domonic.xml.x3d), 331
- imageTexture (in module domonic.xml.x3d), 340
- ImageTexture3D (class in domonic.xml.x3d), 331
- imageTexture3D (in module domonic.xml.x3d), 340
- ImageTextureAtlas (class in domonic.xml.x3d), 331
- imageTextureAtlas (in module domonic.xml.x3d), 340
- img (class in domonic.html), 244
- implementation (domonic.dom.Document property), 252
- importNode() (domonic.dom.Document method), 252
- includes() (domonic.javascript.Array method), 269
- includes() (domonic.javascript.String method), 282
- IndexedFaceSet (class in domonic.xml.x3d), 331
- indexedFaceSet (in module domonic.xml.x3d), 341
- IndexedLineSet (class in domonic.xml.x3d), 331
- indexedLineSet (in module domonic.xml.x3d), 341
- IndexedQuadSet (class in domonic.xml.x3d), 331
- indexedQuadSet (in module domonic.xml.x3d), 341
- IndexedTriangleSet (class in domonic.xml.x3d), 331
- indexedTriangleSet (in module domonic.xml.x3d), 341
- IndexedTriangleStripSet (class in domonic.xml.x3d), 331
- indexedTriangleStripSet (in module domonic.xml.x3d), 341
- indexOf() (domonic.javascript.Array method), 269
- indexOf() (domonic.javascript.String method), 282
- init_assets() (domonic.utils.Utils static method), 322
- inline (class in domonic.xml.x3d), 341
- Inline (in module domonic.xml.x3d), 331
- innerHTML (domonic.dom.Element property), 257
- input (class in domonic.html), 244
- INPUT (domonic.events.Event attribute), 301
- INPUT (domonic.events.InputEvent attribute), 303
- InputEvent (class in domonic.events), 302
- inputType (domonic.events.InputEvent attribute), 303
- ins (class in domonic.html), 244
- insertAdjacentElement() (domonic.dom.Element method), 257
- insertAdjacentHTML() (domonic.dom.Element method), 257
- insertAdjacentText() (domonic.dom.Element method), 257
- insertBefore() (domonic.dom.Node method), 263
- insertData() (domonic.dom.CharacterData method), 248
- insertRule() (domonic.style.CSSStyleSheet method), 309
- instead() (in module domonic.decorators), 325
- internalSubset() (domonic.dom.DocumentType method), 254
- INVALID (domonic.events.Event attribute), 301
- ipconfig (class in domonic.terminal), 291

`ipcrm` (class in *domonic.terminal*), 291
`ipcs` (class in *domonic.terminal*), 291
`is_linux()` (*domonic.utils.Utils* static method), 322
`is_mac()` (*domonic.utils.Utils* static method), 322
`is_nix()` (*domonic.utils.Utils* static method), 322
`is_windows()` (*domonic.utils.Utils* static method), 323
`isArray()` (*domonic.javascript.Array* static method), 269
`isComposing` (*domonic.events.InputEvent* attribute), 303
`isContentEditable()` (*domonic.dom.Element* method), 257
`isDefaultNamespace()` (*domonic.dom.Node* method), 263
`isEqualNode()` (*domonic.dom.Node* method), 264
`isFinite()` (*domonic.javascript.Global* static method), 277
`isFrameAncestor` (*domonic.events.SecurityPolicyViolationEvent* attribute), 306
`isindex` (class in *domonic.html*), 244
`isInteger()` (*domonic.javascript.Number* method), 278
`isMainFrame` (*domonic.events.SecurityPolicyViolationEvent* attribute), 306
`isNaN()` (*domonic.javascript.Global* static method), 277
`isolation` (*domonic.style.Style* property), 316
`IsoSurfaceVolumeData` (class in *domonic.xml.x3d*), 331
`isoSurfaceVolumeData` (in module *domonic.xml.x3d*), 341
`isRejected` (*domonic.events.PromiseRejectionEvent* attribute), 305
`isReload` (*domonic.events.FetchEvent* property), 302
`isSafeInteger()` (*domonic.javascript.Number* method), 278
`isSameNode()` (*domonic.dom.Node* method), 264
`italics()` (*domonic.javascript.String* method), 283
`item()` (*domonic.dom.DOMTokenList* method), 249
`item()` (*domonic.dom.HTMLCollection* method), 260
`item()` (*domonic.dom.NodeList* method), 265
`item()` (*domonic.style.CSSRuleList* method), 308
`item()` (*domonic.style.CSSStyleDeclaration* method), 309
`item()` (*domonic.style.MediaList* method), 310
`item()` (*domonic.style.StyleSheetList* method), 320
`iter()` (*domonic.dom.Node* method), 264

J

`Job` (class in *domonic.javascript*), 277
`jobs` (class in *domonic.terminal*), 291
`join` (class in *domonic.terminal*), 291
`join()` (*domonic.javascript.Array* method), 269
`jq` (class in *domonic.terminal*), 291
`JQUERY` (*domonic.CDN.CDN_JS* attribute), 298
`JQUERY_3_5_1` (*domonic.CDN.CDN_JS* attribute), 298
`JQUERY_UI` (*domonic.CDN.CDN_JS* attribute), 298

`justifyContent` (*domonic.style.Style* property), 316

K

`kbd` (class in *domonic.html*), 244
`key` (*domonic.events.StorageEvent* attribute), 306
`KeyboardEvent` (class in *domonic.events*), 303
`KEYDOWN` (*domonic.events.KeyboardEvent* attribute), 303
`keygen` (class in *domonic.html*), 245
`KEYPRESS` (*domonic.events.KeyboardEvent* attribute), 303
`keys()` (*domonic.dom.NodeList* method), 265
`keys()` (*domonic.javascript.Array* method), 270
`keys()` (*domonic.javascript.Map* method), 278
`keys()` (*domonic.webapi.url.URLSearchParams* method), 351
`KEYUP` (*domonic.events.KeyboardEvent* attribute), 303
`kill` (class in *domonic.terminal*), 291
`killall` (class in *domonic.terminal*), 291

L

`label` (class in *domonic.html*), 245
`lastChance` (*domonic.events.SyncEvent* attribute), 306
`lastChild` (*domonic.dom.Node* property), 264
`lastChild()` (*domonic.dom.TreeWalker* method), 267
`lastElementChild` (*domonic.dom.ParentNode* property), 266
`lastElementChild()` (*domonic.dom.Element* method), 257
`lastEventId` (*domonic.events.MessageEvent* attribute), 303
`lastIndexOf()` (*domonic.javascript.Array* method), 270
`lastIndexOf()` (*domonic.javascript.String* method), 283
`lastmod` (class in *domonic.xml.sitemap*), 349
`LEAVE` (*domonic.events.DragEvent* attribute), 300
`left` (*domonic.style.Style* property), 316
`legend` (class in *domonic.html*), 245
`length` (*domonic.javascript.Array* property), 270
`length` (*domonic.style.CSSRuleList* property), 309
`length` (*domonic.style.CSSStyleDeclaration* property), 309
`length` (*domonic.style.MediaList* property), 310
`length` (*domonic.style.StyleSheetList* property), 320
`less` (class in *domonic.terminal*), 291
`let` (class in *domonic.terminal*), 292
`letterSpacing` (*domonic.style.Style* property), 316
`lex` (class in *domonic.terminal*), 292
`li` (class in *domonic.html*), 245
`line` (class in *domonic.svg*), 327
`linearGradient` (class in *domonic.svg*), 327
`lineHeight` (*domonic.style.Style* property), 316
`LineProperties` (class in *domonic.xml.x3d*), 331
`lineProperties` (in module *domonic.xml.x3d*), 341
`LineSet` (class in *domonic.xml.x3d*), 331

- `lineSet` (in module *domonic.xml.x3d*), 341
 - `link` (class in *domonic.html*), 245
 - `link` (class in *domonic.terminal*), 292
 - `link()` (*domonic.javascript.String* method), 283
 - `links` (*domonic.dom.Document* property), 252
 - `listing` (class in *domonic.html*), 245
 - `listStyle` (*domonic.style.Style* property), 316
 - `listStyleImage` (*domonic.style.Style* property), 316
 - `listStylePosition` (*domonic.style.Style* property), 316
 - `listStyleType` (*domonic.style.Style* property), 316
 - `ln` (class in *domonic.terminal*), 292
 - `LOAD` (*domonic.events.Event* attribute), 301
 - `LOAD` (*domonic.events.ProgressEvent* attribute), 305
 - `LOAD` (*domonic.events.SVGEvent* attribute), 305
 - `LOADED` (*domonic.events.ProgressEvent* attribute), 305
 - `LOADEDATA` (*domonic.events.Event* attribute), 301
 - `LOADEDATA` (*domonic.events.SVGEvent* attribute), 305
 - `LOADEDMETADATA` (*domonic.events.Event* attribute), 301
 - `LOADEDMETADATA` (*domonic.events.SVGEvent* attribute), 305
 - `LOADEND` (*domonic.events.ProgressEvent* attribute), 305
 - `LOADSTART` (*domonic.events.ProgressEvent* attribute), 305
 - `LOADSTART` (*domonic.events.SVGEvent* attribute), 305
 - `loc` (class in *domonic.xml.sitemap*), 349
 - `local` (class in *domonic.terminal*), 292
 - `locale` (class in *domonic.terminal*), 292
 - `localeCompare()` (*domonic.javascript.String* method), 283
 - `localedef` (class in *domonic.terminal*), 292
 - `LOD` (class in *domonic.xml.x3d*), 331
 - `LOD` (in module *domonic.xml.x3d*), 341
 - `log()` (in module *domonic.decorators*), 325
 - `log2()` (*domonic.javascript.Math* static method), 278
 - `logger` (class in *domonic.terminal*), 292
 - `loglp()` (*domonic.javascript.Math* static method), 278
 - `logname` (class in *domonic.terminal*), 292
 - `logout` (class in *domonic.terminal*), 292
 - `lookupNamespaceURI()` (*domonic.dom.Node* method), 264
 - `lookupPrefix()` (*domonic.dom.Node* method), 264
 - `lp` (class in *domonic.terminal*), 292
 - `ls` (class in *domonic.terminal*), 292
- ## M
- `m4` (class in *domonic.terminal*), 292
 - `mailx` (class in *domonic.terminal*), 292
 - `main` (class in *domonic.html*), 245
 - `make` (class in *domonic.terminal*), 292
 - `man` (class in *domonic.terminal*), 292
 - `Map` (class in *domonic.javascript*), 277
 - `map()` (*domonic.javascript.Array* method), 270
 - `mapfile` (class in *domonic.terminal*), 292
 - `margin` (*domonic.style.Style* property), 316
 - `marginBottom` (*domonic.style.Style* property), 316
 - `marginLeft` (*domonic.style.Style* property), 316
 - `marginRight` (*domonic.style.Style* property), 316
 - `marginTop` (*domonic.style.Style* property), 316
 - `mark` (class in *domonic.html*), 245
 - `marker` (class in *domonic.svg*), 327
 - `MARX` (*domonic.CDN.CDN_CSS* attribute), 297
 - `mask` (class in *domonic.svg*), 327
 - `match()` (*domonic.javascript.String* method), 283
 - `matchAll()` (*domonic.javascript.String* method), 283
 - `matches()` (*domonic.dom.Element* method), 257
 - `material` (class in *domonic.xml.x3d*), 341
 - `Material` (in module *domonic.xml.x3d*), 331
 - `Math` (class in *domonic.javascript*), 278
 - `MATHML` (*domonic.CDN.CDN_JS* attribute), 298
 - `matrix` (class in *domonic.geom*), 349
 - `MatrixTextureTransform` (class in *domonic.xml.x3d*), 331
 - `matrixTextureTransform` (in module *domonic.xml.x3d*), 341
 - `MatrixTransform` (class in *domonic.xml.x3d*), 331
 - `matrixTransform` (in module *domonic.xml.x3d*), 341
 - `maxHeight` (*domonic.style.Style* property), 316
 - `maxWidth` (*domonic.style.Style* property), 316
 - `MDI_5_4_55` (*domonic.CDN.CDN_CSS* attribute), 297
 - `MediaList` (class in *domonic.style*), 310
 - `mediaText` (*domonic.style.MediaList* property), 310
 - `menu` (class in *domonic.html*), 245
 - `menuItem` (class in *domonic.html*), 245
 - `merge_dictionaries()` (*domonic.utils.Utils* static method), 323
 - `mesg` (class in *domonic.terminal*), 292
 - `Mesh` (class in *domonic.xml.x3d*), 331
 - `mesh` (in module *domonic.xml.x3d*), 341
 - `MESSAGE` (*domonic.events.Event* attribute), 301
 - `MESSAGE` (*domonic.events.MessageEvent* attribute), 303
 - `MessageEvent` (class in *domonic.events*), 303
 - `meta` (class in *domonic.html*), 245
 - `metadata` (class in *domonic.svg*), 327
 - `MetadataBoolean` (class in *domonic.xml.x3d*), 332
 - `MetadataBoolean` (in module *domonic.xml.x3d*), 341
 - `MetadataDouble` (class in *domonic.xml.x3d*), 332
 - `MetadataDouble` (in module *domonic.xml.x3d*), 341
 - `MetadataFloat` (class in *domonic.xml.x3d*), 332
 - `MetadataFloat` (in module *domonic.xml.x3d*), 341
 - `MetadataInteger` (class in *domonic.xml.x3d*), 332
 - `MetadataInteger` (in module *domonic.xml.x3d*), 341
 - `MetadataSet` (class in *domonic.xml.x3d*), 332
 - `metadataSet` (in module *domonic.xml.x3d*), 342
 - `MetadataString` (class in *domonic.xml.x3d*), 332
 - `metadataString` (in module *domonic.xml.x3d*), 342
 - `meter` (class in *domonic.html*), 245
 - `MILLIGRAM_1_3_0` (*domonic.CDN.CDN_CSS* attribute), 297

minHeight (*domonic.style.Style* property), 316
 minWidth (*domonic.style.Style* property), 316
 mkdir (*class in domonic.terminal*), 292
 mkfifo (*class in domonic.terminal*), 292
 mkfile (*class in domonic.terminal*), 292
 MODERNIZER_2_8_3 (*domonic.CDN.CDN_JS* attribute), 298
 module
 domonic, 242
 domonic.CDN, 297
 domonic.decorators, 325
 domonic.dom, 247
 domonic.events, 299
 domonic.geom, 349
 domonic.html, 242
 domonic.javascript, 267
 domonic.JSON, 296
 domonic.lerpy.easing, 328
 domonic.lerpy.tween, 328
 domonic.style, 307
 domonic.svg, 325
 domonic.terminal, 288
 domonic.utils, 320
 domonic.webapi.fetch, 351
 domonic.webapi.url, 350
 domonic.xml.sitemap, 349
 domonic.xml.x3d, 328
 MOMENT_2_27_0 (*domonic.CDN.CDN_JS* attribute), 299
 more (*class in domonic.terminal*), 292
 MotorJoint (*class in domonic.xml.x3d*), 332
 motorJoint (*in module domonic.xml.x3d*), 342
 MOUSEDOWN (*domonic.events.MouseEvent* attribute), 304
 MOUSEENTER (*domonic.events.MouseEvent* attribute), 304
 MouseEvent (*class in domonic.events*), 304
 MOUSELEAVE (*domonic.events.MouseEvent* attribute), 304
 MOUSEMOVE (*domonic.events.MouseEvent* attribute), 304
 MOUSEOUT (*domonic.events.MouseEvent* attribute), 304
 MOUSEOVER (*domonic.events.MouseEvent* attribute), 304
 MOUSEUP (*domonic.events.MouseEvent* attribute), 304
 MovieTexture (*class in domonic.xml.x3d*), 332
 movieTexture (*in module domonic.xml.x3d*), 342
 mpath (*class in domonic.svg*), 327
 MPRPlane (*class in domonic.xml.x3d*), 331
 mPRPlane (*in module domonic.xml.x3d*), 341
 MPRVolumeStyle (*class in domonic.xml.x3d*), 331
 mPRVolumeStyle (*in module domonic.xml.x3d*), 341
 MultiTexture (*class in domonic.xml.x3d*), 332
 multiTexture (*in module domonic.xml.x3d*), 342
 MultiTextureCoordinate (*class in domonic.xml.x3d*), 332
 multiTextureCoordinate (*in module domonic.xml.x3d*), 342
 mv (*class in domonic.terminal*), 292
 MVP (*domonic.CDN.CDN_CSS* attribute), 297

N

namedItem() (*domonic.dom.HTMLCollection* method), 260
 namespaceURI() (*domonic.dom.Element* method), 257
 NaN() (*domonic.javascript.Global* method), 276
 nautilus (*class in domonic.terminal*), 292
 nav (*class in domonic.html*), 245
 navDown (*domonic.style.Style* property), 316
 NavigationInfo (*class in domonic.xml.x3d*), 332
 navigationInfo (*in module domonic.xml.x3d*), 342
 navIndex (*domonic.style.Style* property), 317
 navLeft (*domonic.style.Style* property), 317
 navRight (*domonic.style.Style* property), 317
 navUp (*domonic.style.Style* property), 317
 NEGATIVE_INFINITY (*domonic.javascript.Number* attribute), 278
 newgrp (*class in domonic.terminal*), 292
 newValue (*domonic.events.StorageEvent* attribute), 306
 nextElementSibling (*domonic.dom.CharacterData* property), 248
 nextElementSibling (*domonic.dom.Element* property), 257
 nextNode() (*domonic.dom.NodeIterator* method), 265
 nextNode() (*domonic.dom.TreeWalker* method), 267
 nextSibling (*domonic.dom.Element* property), 257
 nextSibling (*domonic.dom.Node* property), 264
 nextSibling() (*domonic.dom.TreeWalker* method), 267
 nice (*class in domonic.terminal*), 292
 nl (*class in domonic.terminal*), 292
 nm (*class in domonic.terminal*), 292
 nmap (*class in domonic.terminal*), 292
 Node (*class in domonic.dom*), 263
 NodeIterator (*class in domonic.dom*), 265
 NodeList (*class in domonic.dom*), 265
 nodeName (*domonic.dom.Node* property), 264
 nodeName (*domonic.dom.Text* property), 266
 nodeValue (*domonic.dom.Node* property), 264
 nohup (*class in domonic.terminal*), 292
 Normal (*class in domonic.xml.x3d*), 332
 normal (*in module domonic.xml.x3d*), 342
 NormalInterpolator (*class in domonic.xml.x3d*), 332
 normalInterpolator (*in module domonic.xml.x3d*), 342
 normalize() (*domonic.dom.Element* method), 257
 normalize() (*domonic.dom.Node* method), 264
 normalizeDocument() (*domonic.dom.Document* method), 252
 noscript (*class in domonic.html*), 245
 notations() (*domonic.dom.DocumentType* method), 254
 now() (*domonic.javascript.Date* static method), 272
 Nozzle (*class in domonic.xml.x3d*), 332
 nozzle (*in module domonic.xml.x3d*), 342
 npm (*class in domonic.terminal*), 292

Number (class in domonic.javascript), 278
 Number() (domonic.javascript.Global static method), 276

O

objectFit (domonic.style.Style property), 317
 objectPosition (domonic.style.Style property), 317
 od (class in domonic.terminal), 293
 of() (domonic.javascript.Array static method), 270
 OFFLINE (domonic.events.Event attribute), 301
 offsetHeight() (domonic.dom.Element method), 257
 offsetLeft() (domonic.dom.Element method), 257
 offsetParent() (domonic.dom.Element method), 258
 offsetTop() (domonic.dom.Element method), 258
 offsetWidth() (domonic.dom.Element method), 258
 ol (class in domonic.html), 245
 oldValue (domonic.events.StorageEvent attribute), 306
 ONLINE (domonic.events.Event attribute), 301
 opacity (domonic.style.Style property), 317
 OpacityMapVolumeStyle (class in domonic.xml.x3d), 332
 opacityMapVolumeStyle (in module domonic.xml.x3d), 342
 OPEN (domonic.events.Event attribute), 301
 open() (domonic.dom.Document method), 252
 optgroup (class in domonic.html), 245
 option (class in domonic.html), 245
 order (domonic.style.Style property), 317
 ordinal() (domonic.dom.EntityReference static method), 259
 OrientationChaser (class in domonic.xml.x3d), 332
 orientationChaser (in module domonic.xml.x3d), 342
 OrientationDamper (class in domonic.xml.x3d), 332
 orientationDamper (in module domonic.xml.x3d), 342
 OrientationInterpolator (class in domonic.xml.x3d), 332
 orientationInterpolator (in module domonic.xml.x3d), 342
 origin (domonic.events.MessageEvent attribute), 303
 originalPolicy (domonic.events.SecurityPolicyViolationEvent attribute), 306
 orphans (domonic.style.Style property), 317
 OrthoViewpoint (class in domonic.xml.x3d), 332
 orthoViewpoint (in module domonic.xml.x3d), 342
 outline (domonic.style.Style property), 317
 outlineColor (domonic.style.Style property), 317
 outlineOffset (domonic.style.Style property), 317
 outlineStyle (domonic.style.Style property), 317
 outlineWidth (domonic.style.Style property), 317
 output (class in domonic.html), 245
 OVER (domonic.events.DragEvent attribute), 300
 overflow (domonic.style.Style property), 317
 overflowX (domonic.style.Style property), 317
 overflowY (domonic.style.Style property), 317

ownerDocument (domonic.dom.Node property), 264
 ownKeys() (domonic.javascript.Reflect static method), 280

P

p (class in domonic.html), 245
 padding (domonic.style.Style property), 317
 paddingBottom (domonic.style.Style property), 317
 paddingLeft (domonic.style.Style property), 318
 paddingRight (domonic.style.Style property), 318
 paddingTop (domonic.style.Style property), 318
 padEnd() (domonic.javascript.String method), 283
 padStart() (domonic.javascript.String method), 283
 pageBreakAfter (domonic.style.Style property), 318
 pageBreakBefore (domonic.style.Style property), 318
 pageBreakInside (domonic.style.Style property), 318
 PAGEHIDE (domonic.events.PageTransitionEvent attribute), 304
 PAGESHOW (domonic.events.PageTransitionEvent attribute), 304
 PageTransitionEvent (class in domonic.events), 304
 param (class in domonic.html), 245
 Param (class in domonic.xml.x3d), 332
 param (in module domonic.xml.x3d), 342
 parentElement (domonic.dom.Element property), 258
 ParentNode (class in domonic.dom), 265
 parentNode() (domonic.dom.TreeWalker method), 267
 parse() (domonic.javascript.Date static method), 272
 parse() (in module domonic.JSON), 296
 parse_file() (in module domonic.JSON), 296
 parseFloat() (domonic.javascript.Global static method), 277
 parseFloat() (in module domonic.javascript), 288
 parseInt() (domonic.javascript.Global static method), 277
 parseInt() (in module domonic.javascript), 288
 ParticleSet (class in domonic.xml.x3d), 332
 particleSet (in module domonic.xml.x3d), 342
 passwd (class in domonic.terminal), 293
 Paste (class in domonic.terminal), 293
 PASTE (domonic.events.ClipboardEvent attribute), 299
 patch (class in domonic.terminal), 293
 path (class in domonic.svg), 327
 pathchk (class in domonic.terminal), 293
 pattern (class in domonic.svg), 327
 PAUSE (domonic.events.Event attribute), 301
 PAUSE (domonic.events.TweenEvent attribute), 307
 pause() (domonic.lerpy.tween.Tween method), 328
 pax (class in domonic.terminal), 293
 permutations() (domonic.utils.Utils static method), 323
 persisted (domonic.events.PageTransitionEvent attribute), 304
 perspective (domonic.style.Style property), 318

`perspectiveOrigin` (*domonic.style.Style* property), 318
`PhysicalEnvironmentLight` (class in *domonic.xml.x3d*), 332
`physicalEnvironmentLight` (in module *domonic.xml.x3d*), 342
`PhysicalMaterial` (class in *domonic.xml.x3d*), 332
`physicalMaterial` (in module *domonic.xml.x3d*), 342
`picture` (class in *domonic.html*), 245
`pictureInPictureEnabled` (*domonic.dom.Document* property), 252
`ping` (class in *domonic.terminal*), 293
`pip` (class in *domonic.terminal*), 293
`pixelDepth()` (*domonic.javascript.Screen* method), 280
`PixelTexture` (class in *domonic.xml.x3d*), 332
`pixelTexture` (in module *domonic.xml.x3d*), 343
`PixelTexture3D` (class in *domonic.xml.x3d*), 332
`pixelTexture3D` (in module *domonic.xml.x3d*), 343
`PIXI_5_3_3` (*domonic.CDN.CDN_JS* attribute), 299
`PLACEHOLDER()` (*domonic.CDN.CDN_IMG* static method), 298
`plaintext` (class in *domonic.html*), 245
`plane` (class in *domonic.xml.x3d*), 343
`Plane` (in module *domonic.xml.x3d*), 332
`PlaneSensor` (class in *domonic.xml.x3d*), 332
`planeSensor` (in module *domonic.xml.x3d*), 343
`PLAY` (*domonic.events.Event* attribute), 301
`PLAYING` (*domonic.events.Event* attribute), 301
`POINTER` (*domonic.events.PointerEvent* attribute), 304
`pointerBeforeReferenceNode()` (*domonic.dom.NodeIterator* method), 265
`POINTERCANCEL` (*domonic.events.PointerEvent* attribute), 304
`POINTERDOWN` (*domonic.events.PointerEvent* attribute), 304
`POINTERENTER` (*domonic.events.PointerEvent* attribute), 304
`PointerEvent` (class in *domonic.events*), 304
`POINTERLEAVE` (*domonic.events.PointerEvent* attribute), 304
`POINTERMOVE` (*domonic.events.PointerEvent* attribute), 304
`POINTEROUT` (*domonic.events.PointerEvent* attribute), 304
`POINTEROVER` (*domonic.events.PointerEvent* attribute), 304
`POINTERUP` (*domonic.events.PointerEvent* attribute), 304
`PointLight` (class in *domonic.xml.x3d*), 332
`pointLight` (in module *domonic.xml.x3d*), 343
`PointSet` (class in *domonic.xml.x3d*), 332
`pointSet` (in module *domonic.xml.x3d*), 343
`polygon` (class in *domonic.svg*), 327
`polyline` (class in *domonic.svg*), 327
`Polyline2D` (class in *domonic.xml.x3d*), 332
`polyline2D` (in module *domonic.xml.x3d*), 343
`Polypoint2D` (class in *domonic.xml.x3d*), 333
`polypoint2D` (in module *domonic.xml.x3d*), 343
`pop()` (*domonic.javascript.Array* method), 270
`popd` (class in *domonic.terminal*), 293
`PopGeometry` (class in *domonic.xml.x3d*), 333
`popGeometry` (in module *domonic.xml.x3d*), 343
`PopGeometryLevel` (class in *domonic.xml.x3d*), 333
`popGeometryLevel` (in module *domonic.xml.x3d*), 343
`POPPER_1_16_1` (*domonic.CDN.CDN_JS* attribute), 299
`POPSTATE` (*domonic.events.PopStateEvent* attribute), 305
`PopStateEvent` (class in *domonic.events*), 304
`portal` (class in *domonic.html*), 245
`ports` (*domonic.events.MessageEvent* attribute), 303
`position` (*domonic.style.Style* property), 318
`PositionChaser` (class in *domonic.xml.x3d*), 333
`positionChaser` (in module *domonic.xml.x3d*), 343
`PositionChaser2D` (class in *domonic.xml.x3d*), 333
`positionChaser2D` (in module *domonic.xml.x3d*), 343
`PositionDamper` (class in *domonic.xml.x3d*), 333
`positionDamper` (in module *domonic.xml.x3d*), 343
`PositionDamper2D` (class in *domonic.xml.x3d*), 333
`positionDamper2D` (in module *domonic.xml.x3d*), 343
`PositionInterpolator` (class in *domonic.xml.x3d*), 333
`positionInterpolator` (in module *domonic.xml.x3d*), 343
`PositionInterpolator2D` (class in *domonic.xml.x3d*), 333
`positionInterpolator2D` (in module *domonic.xml.x3d*), 343
`POSITIVE_INFINITY` (*domonic.javascript.Number* attribute), 278
`postMessage()` (*domonic.javascript.Worker* method), 287
`pr` (class in *domonic.terminal*), 293
`pre` (class in *domonic.html*), 245
`prepend()` (*domonic.dom.Element* method), 258
`preventExtensions()` (*domonic.javascript.Reflect* static method), 280
`previousElementSibling` (*domonic.dom.CharacterData* property), 248
`previousElementSibling` (*domonic.dom.Element* property), 258
`previousNode()` (*domonic.dom.NodeIterator* method), 265
`previousNode()` (*domonic.dom.TreeWalker* method), 267
`previousSibling` (*domonic.dom.Node* property), 264
`previousSibling()` (*domonic.dom.TreeWalker* method), 267
`printf` (class in *domonic.terminal*), 293
`priority` (class in *domonic.xml sitemap*), 349
`ProcessingInstruction` (class in *domonic.dom*), 266

ProgramKilled, 279
 progress (class in domonic.html), 245
 PROGRESS (domonic.events.Event attribute), 301
 PROGRESS (domonic.events.ProgressEvent attribute), 305
 PROGRESS (domonic.events.SVGEvent attribute), 305
 ProgressEvent (class in domonic.events), 305
 ProjectionVolumeStyle (class in domonic.xml.x3d), 333
 projectionVolumeStyle (in module domonic.xml.x3d), 343
 promise (domonic.events.PromiseRejectionEvent attribute), 305
 PromiseRejectionEvent (class in domonic.events), 305
 prompt() (domonic.javascript.Window static method), 286
 propertyName (domonic.events.TransitionEvent attribute), 307
 prototype (domonic.javascript.Array attribute), 270
 prs (class in domonic.terminal), 293
 ps (class in domonic.terminal), 293
 pseudoElement (domonic.events.AnimationEvent attribute), 299
 pseudoElement (domonic.events.TransitionEvent attribute), 307
 push() (domonic.javascript.Array method), 270
 pushd (class in domonic.terminal), 293
 pwd (class in domonic.terminal), 293
 Pyramid (class in domonic.xml.x3d), 333
 pyramid (in module domonic.xml.x3d), 343
 python (class in domonic.terminal), 293

Q

q (class in domonic.html), 245
 qalter (class in domonic.terminal), 293
 qdel (class in domonic.terminal), 293
 qhold (class in domonic.terminal), 293
 qmove (class in domonic.terminal), 293
 qmsg (class in domonic.terminal), 293
 qrerun (class in domonic.terminal), 293
 qrls (class in domonic.terminal), 293
 qselect (class in domonic.terminal), 293
 qsig (class in domonic.terminal), 293
 qstat (class in domonic.terminal), 293
 qsub (class in domonic.terminal), 293
 QuadSet (class in domonic.xml.x3d), 333
 quadSet (in module domonic.xml.x3d), 343
 querySelector() (domonic.dom.DocumentFragment method), 254
 querySelector() (domonic.dom.Element method), 258
 querySelectorAll() (domonic.dom.DocumentFragment method), 254
 querySelectorAll() (domonic.dom.Element method), 258

quotes (domonic.style.Style property), 318

R

RadarVolumeStyle (class in domonic.xml.x3d), 333
 radarVolumeStyle (in module domonic.xml.x3d), 343
 radialGradient (class in domonic.svg), 327
 RadioNodeList (class in domonic.dom), 266
 RATECHANGE (domonic.events.Event attribute), 301
 raw() (domonic.javascript.String static method), 284
 read (class in domonic.terminal), 293
 readarray (class in domonic.terminal), 293
 readonly (class in domonic.terminal), 293
 READYSTATECHANGE (domonic.events.Event attribute), 301
 reason (domonic.events.PromiseRejectionEvent attribute), 305
 reboot (class in domonic.terminal), 293
 rect (class in domonic.svg), 328
 Rectangle2D (class in domonic.xml.x3d), 333
 rectangle2D (in module domonic.xml.x3d), 344
 RectangularTorus (class in domonic.xml.x3d), 333
 rectangularTorus (in module domonic.xml.x3d), 344
 reduce() (domonic.javascript.Array method), 270
 reduceRight() (domonic.javascript.Array method), 270
 referenceNode() (domonic.dom.NodeIterator method), 265
 RefinementTexture (class in domonic.xml.x3d), 333
 refinementTexture (in module domonic.xml.x3d), 344
 Reflect (class in domonic.javascript), 279
 RemoteSelectionGroup (class in domonic.xml.x3d), 333
 remoteSelectionGroup (in module domonic.xml.x3d), 344
 remove() (domonic.dom.CharacterData method), 248
 remove() (domonic.dom.ChildNode method), 248
 remove() (domonic.dom.DOMTokenList method), 249
 remove() (domonic.dom.Element method), 258
 removeAttribute() (domonic.dom.Element method), 258
 removeAttributeNode() (domonic.dom.Element method), 258
 removeAttributeNS() (domonic.dom.Element method), 258
 removeChild() (domonic.dom.Node method), 264
 removeNamedItem() (domonic.dom.Attr method), 247
 removeProperty() (domonic.style.CSSStyleDeclaration method), 309
 removeRule() (domonic.style.CSSStyleSheet method), 309
 renameNode() (domonic.dom.Document method), 252
 render() (in module domonic.html), 245
 RenderedTexture (class in domonic.xml.x3d), 333
 renderedTexture (in module domonic.xml.x3d), 344
 renice (class in domonic.terminal), 294

- `repeat()` (*domonic.javascript.String* method), 284
- `replace()` (*domonic.javascript.String* method), 284
- `replace()` (*domonic.style.CSSStyleSheet* method), 310
- `replace_between()` (*domonic.utils.Utils* static method), 323
- `replaceAll()` (*domonic.javascript.String* method), 284
- `replaceChild()` (*domonic.dom.Node* method), 264
- `replaceChildren()` (*domonic.dom.DocumentFragment* method), 254
- `replaceData()` (*domonic.dom.CharacterData* method), 248
- `replaceSync()` (*domonic.style.CSSStyleSheet* method), 310
- `replaceWith()` (*domonic.dom.CharacterData* method), 248
- `replaceWith()` (*domonic.dom.ChildNode* method), 248
- `request` (*domonic.events.FetchEvent* attribute), 302
- `requestAnimationFrame()` (*domonic.javascript.Window* static method), 287
- `requestFullscreen()` (*domonic.dom.Element* method), 258
- `require()` (*domonic.javascript.Global* static method), 277
- `RESET` (*domonic.events.Event* attribute), 301
- `RESET` (*domonic.events.TweenEvent* attribute), 307
- `RESIZE` (*domonic.events.Event* attribute), 301
- `resize` (*domonic.style.Style* property), 318
- `respondWith()` (*domonic.events.FetchEvent* method), 302
- `reverse()` (*domonic.javascript.Array* method), 270
- `right` (*domonic.style.Style* property), 318
- `RigidBody` (*class in domonic.xml.x3d*), 333
- `rigidBody` (*in module domonic.xml.x3d*), 344
- `RigidBodyCollection` (*class in domonic.xml.x3d*), 333
- `rigidBodyCollection` (*in module domonic.xml.x3d*), 344
- `rm` (*class in domonic.terminal*), 294
- `rm del` (*class in domonic.terminal*), 294
- `rmdir` (*class in domonic.terminal*), 294
- `root` (*domonic.dom.TreeWalker* property), 267
- `rootNode` (*domonic.dom.Node* property), 265
- `rotate()` (*domonic.geom.matrix* method), 349
- `Route` (*class in domonic.xml.x3d*), 333
- `route` (*in module domonic.xml.x3d*), 344
- `rp` (*class in domonic.html*), 246
- `rsync` (*class in domonic.terminal*), 294
- `rt` (*class in domonic.html*), 246
- `ruby` (*class in domonic.html*), 246
- `run()` (*domonic.javascript.Job* method), 277
- `run()` (*domonic.terminal.command* static method), 289
- `sact` (*class in domonic.terminal*), 294
- `samp` (*class in domonic.html*), 246
- `say` (*class in domonic.terminal*), 294
- `ScalarChaser` (*class in domonic.xml.x3d*), 333
- `scalarChaser` (*in module domonic.xml.x3d*), 344
- `ScalarDamper` (*class in domonic.xml.x3d*), 333
- `scalarDamper` (*in module domonic.xml.x3d*), 344
- `ScalarInterpolator` (*class in domonic.xml.x3d*), 333
- `scalarInterpolator` (*in module domonic.xml.x3d*), 344
- `scale()` (*domonic.geom.matrix* method), 349
- `sccs` (*class in domonic.terminal*), 294
- `scene` (*class in domonic.xml.x3d*), 344
- `Scene` (*in module domonic.xml.x3d*), 333
- `scp` (*class in domonic.terminal*), 294
- `Screen` (*class in domonic.javascript*), 280
- `script` (*class in domonic.html*), 246
- `scripts` (*domonic.dom.Document* property), 253
- `SCROLL` (*domonic.events.Event* attribute), 301
- `SCROLL` (*domonic.events.SVGEvent* attribute), 305
- `scrollHeight()` (*domonic.dom.Element* method), 258
- `scrollIntoView()` (*domonic.dom.Element* method), 259
- `scrollLeft()` (*domonic.dom.Element* method), 259
- `scrollTop()` (*domonic.dom.Element* method), 259
- `scrollWidth()` (*domonic.dom.Element* method), 259
- `SEARCH` (*domonic.events.Event* attribute), 301
- `search()` (*domonic.javascript.String* method), 284
- `section` (*class in domonic.html*), 246
- `SECURITY_POLICY_VIOLATION` (*domonic.events.SecurityPolicyViolationEvent* attribute), 306
- `SecurityPolicyViolationEvent` (*class in domonic.events*), 305
- `sed` (*class in domonic.terminal*), 294
- `SEEKED` (*domonic.events.Event* attribute), 301
- `SEEKING` (*domonic.events.Event* attribute), 301
- `SegmentedVolumeData` (*class in domonic.xml.x3d*), 333
- `segmentedVolumeData` (*in module domonic.xml.x3d*), 344
- `select` (*class in domonic.html*), 246
- `SELECT` (*domonic.events.Event* attribute), 301
- `SELECT` (*domonic.events.InputEvent* attribute), 303
- `set()` (*domonic.dom.DOMStringMap* method), 249
- `set()` (*domonic.javascript.Map* method), 278
- `set()` (*domonic.javascript.Reflect* static method), 280
- `set()` (*domonic.webapi.url.URLSearchParams* method), 351
- `setAttribute()` (*domonic.dom.Element* method), 259
- `setAttributeNode()` (*domonic.dom.Element* method), 259
- `setAttributeNodeNS()` (*domonic.dom.Element* method), 259
- `s` (*class in domonic.html*), 246

- setAttributeNS() (*domonic.dom.Element* method), 259
- setDate() (*domonic.javascript.Date* method), 272
- setFullYear() (*domonic.javascript.Date* method), 273
- setHours() (*domonic.javascript.Date* method), 273
- setMilliseconds() (*domonic.javascript.Date* method), 273
- setMinutes() (*domonic.javascript.Date* method), 273
- setMonth() (*domonic.javascript.Date* method), 274
- setNamedItem() (*domonic.dom.Attr* method), 247
- setPrototypeOf() (*domonic.javascript.Reflect* static method), 280
- setSeconds() (*domonic.javascript.Date* method), 274
- setTime() (*domonic.javascript.Date* method), 274
- setTimeout() (*domonic.javascript.Global* static method), 277
- setTimeout() (*domonic.javascript.Window* method), 287
- setTimeout() (in module *domonic.javascript*), 288
- setUTCDate() (*domonic.javascript.Date* method), 274
- setUTCFullYear() (*domonic.javascript.Date* method), 274
- setUTCHours() (*domonic.javascript.Date* method), 274
- setUTCMilliseconds() (*domonic.javascript.Date* method), 274
- setUTCMinutes() (*domonic.javascript.Date* method), 274
- setUTCMonth() (*domonic.javascript.Date* method), 274
- setUTCSeconds() (*domonic.javascript.Date* method), 275
- setYear() (*domonic.javascript.Date* method), 275
- sh (*class in domonic.terminal*), 294
- ShadedVolumeStyle (*class in domonic.xml.x3d*), 333
- shadedVolumeStyle (in module *domonic.xml.x3d*), 344
- ShaderPart (*class in domonic.xml.x3d*), 333
- shaderPart (in module *domonic.xml.x3d*), 344
- ShadowRoot (*class in domonic.dom*), 266
- shape (*class in domonic.xml.x3d*), 344
- Shape (in module *domonic.xml.x3d*), 333
- shift (*class in domonic.terminal*), 294
- shift() (*domonic.javascript.Array* method), 270
- shopt (*class in domonic.terminal*), 294
- SHOW (*domonic.events.Event* attribute), 301
- shutdown (*class in domonic.terminal*), 294
- silence() (in module *domonic.decorators*), 325
- SilhouetteEnhancementVolumeStyle (*class in domonic.xml.x3d*), 333
- silhouetteEnhancementVolumeStyle (in module *domonic.xml.x3d*), 344
- SIMPLE (*domonic.CDN.CDN_CSS* attribute), 297
- SingleAxisHingeJoint (*class in domonic.xml.x3d*), 334
- singleAxisHingeJoint (in module *domonic.xml.x3d*), 344
- sitemap (*class in domonic.xml.sitemap*), 349
- sitemap_format() (in module *domonic.xml.sitemap*), 350
- sitemap_from_urls() (in module *domonic.xml.sitemap*), 350
- sitemapindex (*class in domonic.xml.sitemap*), 350
- sitemapindex_from_urls() (in module *domonic.xml.sitemap*), 350
- sleep (*class in domonic.terminal*), 294
- slice() (*domonic.javascript.Array* method), 270
- slice() (*domonic.javascript.String* method), 284
- SliderJoint (*class in domonic.xml.x3d*), 334
- sliderJoint (in module *domonic.xml.x3d*), 344
- SlopedCylinder (*class in domonic.xml.x3d*), 334
- slopedCylinder (in module *domonic.xml.x3d*), 344
- small (*class in domonic.html*), 246
- small() (*domonic.javascript.String* method), 284
- Snout (*class in domonic.xml.x3d*), 334
- snout (in module *domonic.xml.x3d*), 345
- SOCKET_1_4_5 (*domonic.CDN.CDN_JS* attribute), 299
- solidcolor (*class in domonic.svg*), 328
- SolidOfRevolution (*class in domonic.xml.x3d*), 334
- solidOfRevolution (in module *domonic.xml.x3d*), 345
- some() (*domonic.javascript.Array* method), 271
- sort (*class in domonic.terminal*), 294
- sort() (*domonic.javascript.Array* method), 271
- sort() (*domonic.webapi.url.URLSearchParams* method), 351
- Sound (*class in domonic.xml.x3d*), 334
- sound (in module *domonic.xml.x3d*), 345
- source (*class in domonic.html*), 246
- source (*class in domonic.terminal*), 294
- source (*domonic.events.MessageEvent* attribute), 304
- span (*class in domonic.html*), 246
- sphere (*class in domonic.xml.x3d*), 345
- Sphere (in module *domonic.xml.x3d*), 334
- SphereSegment (*class in domonic.xml.x3d*), 334
- sphereSegment (in module *domonic.xml.x3d*), 345
- SphereSensor (*class in domonic.xml.x3d*), 334
- sphereSensor (in module *domonic.xml.x3d*), 345
- splice() (*domonic.javascript.Array* method), 271
- SplinePositionInterpolator (*class in domonic.xml.x3d*), 334
- splinePositionInterpolator (in module *domonic.xml.x3d*), 345
- split (*class in domonic.terminal*), 294
- split() (*domonic.javascript.String* method), 284
- splitText() (*domonic.dom.Text* method), 266
- SpotLight (*class in domonic.xml.x3d*), 334
- spotLight (in module *domonic.xml.x3d*), 345
- squash() (*domonic.utils.Utils* static method), 323
- ssh (*class in domonic.terminal*), 294
- STALLED (*domonic.events.Event* attribute), 301
- START (*domonic.events.DragEvent* attribute), 300

- START (*domonic.events.GamePadEvent* attribute), 302
 - START (*domonic.events.TweenEvent* attribute), 307
 - startsWith() (*domonic.javascript.String* method), 285
 - state (*domonic.events.PopStateEvent* attribute), 305
 - StaticGroup (*class in domonic.xml.x3d*), 334
 - staticGroup (*in module domonic.xml.x3d*), 345
 - StippleVolumeStyle (*class in domonic.xml.x3d*), 334
 - stippleVolumeStyle (*in module domonic.xml.x3d*), 345
 - stop (*class in domonic.svg*), 328
 - STOP (*domonic.events.GamePadEvent* attribute), 302
 - STOP (*domonic.events.TweenEvent* attribute), 307
 - stopPropagation() (*domonic.events.Event* method), 302
 - STORAGE (*domonic.events.StorageEvent* attribute), 306
 - storageArea (*domonic.events.StorageEvent* attribute), 306
 - StorageEvent (*class in domonic.events*), 306
 - strictErrorChecking() (*domonic.dom.Document* method), 253
 - strike (*class in domonic.html*), 246
 - strike() (*domonic.javascript.String* method), 285
 - String (*class in domonic.javascript*), 280
 - String() (*domonic.javascript.Global* static method), 276
 - stringify() (*in module domonic.JSON*), 297
 - strings (*class in domonic.terminal*), 294
 - strip (*class in domonic.terminal*), 294
 - strong (*class in domonic.html*), 246
 - stty (*class in domonic.terminal*), 294
 - style (*class in domonic.html*), 246
 - Style (*class in domonic.style*), 310
 - style (*domonic.dom.Element* property), 259
 - StyleSheet (*class in domonic.style*), 320
 - StyleSheetList (*class in domonic.style*), 320
 - sub (*class in domonic.html*), 246
 - sub() (*domonic.javascript.String* method), 285
 - submit (*class in domonic.html*), 246
 - SUBMIT (*domonic.events.Event* attribute), 301
 - SUBMIT (*domonic.events.SubmitEvent* attribute), 306
 - SubmitEvent (*class in domonic.events*), 306
 - substr() (*domonic.javascript.String* method), 285
 - substring() (*domonic.javascript.String* method), 285
 - substringData() (*domonic.dom.CharacterData* method), 248
 - summary (*class in domonic.html*), 246
 - sup (*class in domonic.html*), 246
 - sup() (*domonic.javascript.String* method), 285
 - SurfaceShaderTexture (*class in domonic.xml.x3d*), 334
 - surfaceShaderTexture (*in module domonic.xml.x3d*), 345
 - suspend (*class in domonic.terminal*), 294
 - SUSPEND (*domonic.events.Event* attribute), 301
 - svg (*class in domonic.svg*), 328
 - SVGEvent (*class in domonic.events*), 305
 - switch (*class in domonic.svg*), 328
 - Switch (*class in domonic.xml.x3d*), 334
 - switch (*in module domonic.xml.x3d*), 345
 - symbol (*class in domonic.svg*), 328
 - SYNC (*domonic.events.SyncEvent* attribute), 306
 - SyncEvent (*class in domonic.events*), 306
- ## T
- table (*class in domonic.html*), 246
 - tableLayout (*domonic.style.Style* property), 318
 - tablify() (*in module domonic.JSON*), 297
 - tabs (*class in domonic.terminal*), 294
 - tabSize (*domonic.style.Style* property), 318
 - tag (*domonic.dom.Node* property), 265
 - tag (*domonic.events.SyncEvent* attribute), 306
 - tail (*class in domonic.terminal*), 294
 - tail (*domonic.dom.Node* property), 265
 - TAILWIND_2_2_15 (*domonic.CDN.CDN_CSS* attribute), 298
 - talk (*class in domonic.terminal*), 294
 - tar (*class in domonic.terminal*), 294
 - tbody (*class in domonic.html*), 246
 - td (*class in domonic.html*), 246
 - tee (*class in domonic.terminal*), 294
 - template (*class in domonic.html*), 246
 - TemplateError, 242
 - TerminalException, 288
 - terminate() (*domonic.javascript.Worker* method), 287
 - test (*class in domonic.terminal*), 294
 - TexCoordDamper2D (*class in domonic.xml.x3d*), 334
 - texCoordDamper2D (*in module domonic.xml.x3d*), 345
 - Text (*class in domonic.dom*), 266
 - text (*class in domonic.svg*), 328
 - Text (*class in domonic.xml.x3d*), 334
 - text (*domonic.dom.Node* property), 265
 - text (*in module domonic.xml.x3d*), 345
 - textAlign (*domonic.style.Style* property), 318
 - textAlignLast (*domonic.style.Style* property), 318
 - textarea (*class in domonic.html*), 246
 - textContent (*domonic.dom.Node* property), 265
 - textDecoration (*domonic.style.Style* property), 318
 - textDecorationColor (*domonic.style.Style* property), 318
 - textDecorationLine (*domonic.style.Style* property), 318
 - textDecorationStyle (*domonic.style.Style* property), 319
 - textIndent (*domonic.style.Style* property), 319
 - textJustify (*domonic.style.Style* property), 319
 - textOverflow (*domonic.style.Style* property), 319
 - textPath (*class in domonic.svg*), 328
 - textShadow (*domonic.style.Style* property), 319

- textTransform (*domonic.style.Style* property), 319
 Texture (*class* in *domonic.xml.x3d*), 334
 texture (*in module domonic.xml.x3d*), 345
 TextureCoordinate (*class* in *domonic.xml.x3d*), 334
 textureCoordinate (*in module domonic.xml.x3d*), 345
 TextureCoordinate3D (*class* in *domonic.xml.x3d*), 334
 textureCoordinate3D (*in module domonic.xml.x3d*), 345
 TextureCoordinateGenerator (*class* in *domonic.xml.x3d*), 334
 textureCoordinateGenerator (*in module domonic.xml.x3d*), 345
 TextureProperties (*class* in *domonic.xml.x3d*), 334
 textureProperties (*in module domonic.xml.x3d*), 345
 TextureTransform (*class* in *domonic.xml.x3d*), 334
 textureTransform (*in module domonic.xml.x3d*), 345
 TextureTransform3D (*class* in *domonic.xml.x3d*), 334
 textureTransform3D (*in module domonic.xml.x3d*), 346
 TextureTransformMatrix3D (*class* in *domonic.xml.x3d*), 334
 textureTransformMatrix3D (*in module domonic.xml.x3d*), 346
 tfoot (*class* in *domonic.html*), 247
 th (*class* in *domonic.html*), 247
 thead (*class* in *domonic.html*), 247
 THREE_DOTS_0_2_0 (*domonic.CDN.CDN_CSS* attribute), 298
 time (*class* in *domonic.terminal*), 294
 TIMEOUT (*domonic.events.ProgressEvent* attribute), 305
 TIMER (*domonic.events.TimerEvent* attribute), 306
 TIMER (*domonic.events.TweenEvent* attribute), 307
 TIMER_COMPLETE (*domonic.events.TimerEvent* attribute), 306
 TimerEvent (*class* in *domonic.events*), 306
 times (*class* in *domonic.terminal*), 294
 timeSensor (*class* in *domonic.xml.x3d*), 346
 TimeSensor (*in module domonic.xml.x3d*), 334
 timeStamp (*domonic.events.ExtendableEvent* attribute), 302
 title (*class* in *domonic.html*), 247
 title (*class* in *domonic.svg*), 328
 title (*domonic.dom.Document* property), 253
 title (*domonic.dom.Element* property), 259
 to_dictionary() (*domonic.utils.Utils* static method), 324
 toCharCode() (*domonic.javascript.String* static method), 285
 toCodePoint() (*domonic.javascript.String* static method), 285
 toDateString() (*domonic.javascript.Date* method), 275
 toExponential() (*domonic.javascript.Number* method), 278
 toFixed() (*domonic.javascript.Number* method), 278
 TOGGLE (*domonic.events.Event* attribute), 301
 toggle() (*domonic.dom.DOMTokenList* method), 249
 toGMTString() (*domonic.javascript.Date* method), 275
 toISOString() (*domonic.javascript.Date* method), 275
 toJSON() (*domonic.javascript.Date* method), 275
 toLocaleDateString() (*domonic.javascript.Date* method), 275
 toLocaleLowerCase() (*domonic.javascript.String* method), 285
 toLocaleString() (*domonic.javascript.Date* method), 275
 toLocaleTimeString() (*domonic.javascript.Date* method), 275
 toLocaleUpperCase() (*domonic.javascript.String* method), 285
 toLowerCase() (*domonic.javascript.String* method), 285
 ToneMappedVolumeStyle (*class* in *domonic.xml.x3d*), 334
 toneMappedVolumeStyle (*in module domonic.xml.x3d*), 346
 top (*domonic.style.Style* property), 319
 toPrecision() (*domonic.javascript.Number* method), 278
 Torus (*class* in *domonic.xml.x3d*), 334
 torus (*in module domonic.xml.x3d*), 346
 toSource() (*domonic.javascript.Array* method), 271
 toString() (*domonic.dom.DOMTokenList* method), 249
 toString() (*domonic.dom.Element* method), 259
 toString() (*domonic.javascript.Array* method), 271
 toString() (*domonic.javascript.Date* method), 275
 toString() (*domonic.javascript.Function* method), 276
 toString() (*domonic.javascript.Number* method), 279
 toString() (*domonic.webapi.url.URLSearchParams* method), 351
 toTimeString() (*domonic.javascript.Date* method), 275
 touch (*class* in *domonic.terminal*), 295
 TOUCHCANCEL (*domonic.events.TouchEvent* attribute), 307
 TOUCHEND (*domonic.events.TouchEvent* attribute), 307
 TouchEvent (*class* in *domonic.events*), 307
 TOUCHMOVE (*domonic.events.TouchEvent* attribute), 307
 TouchSensor (*class* in *domonic.xml.x3d*), 334
 touchSensor (*in module domonic.xml.x3d*), 346
 TOUCHSTART (*domonic.events.TouchEvent* attribute), 307
 toUpperCase() (*domonic.javascript.String* method), 286
 toUTCString() (*domonic.javascript.Date* method), 275
 tput (*class* in *domonic.terminal*), 295
 tr (*class* in *domonic.html*), 247
 tr (*class* in *domonic.terminal*), 295
 track (*class* in *domonic.html*), 247

`transform` (class in `domonic.xml.x3d`), 346
`transform` (*domonic.style.Style* property), 319
`Transform` (in module `domonic.xml.x3d`), 334
`transformOrigin` (*domonic.style.Style* property), 319
`transformStyle` (*domonic.style.Style* property), 319
`transition` (*domonic.style.Style* property), 319
`transitionDelay` (*domonic.style.Style* property), 319
`transitionDuration` (*domonic.style.Style* property), 319
`TRANSITIONEND` (*domonic.events.TransitionEvent* attribute), 307
`TransitionEvent` (class in *domonic.events*), 307
`transitionProperty` (*domonic.style.Style* property), 319
`transitionTimingFunction` (*domonic.style.Style* property), 319
`translate()` (*domonic.geom.matrix* method), 349
`trap` (class in *domonic.terminal*), 295
`tree` (*domonic.dom.TreeWalker* attribute), 267
`TreeWalker` (class in *domonic.dom*), 266
`tref` (class in *domonic.svg*), 328
`TriangleSet` (class in *domonic.xml.x3d*), 335
`triangleSet` (in module *domonic.xml.x3d*), 346
`TriangleSet2D` (class in *domonic.xml.x3d*), 335
`triangleSet2D` (in module *domonic.xml.x3d*), 346
`trim()` (*domonic.javascript.String* method), 286
`trimEnd()` (*domonic.javascript.String* method), 286
`trimStart()` (*domonic.javascript.String* method), 286
`true` (class in *domonic.terminal*), 295
`truncate()` (*domonic.utils.Utils* static method), 324
`tsort` (class in *domonic.terminal*), 295
`tspan` (class in *domonic.svg*), 328
`tty` (class in *domonic.terminal*), 295
`Tween` (class in *domonic.lerpy.tween*), 328
`TweenEvent` (class in *domonic.events*), 307
`TwoSidedMaterial` (class in *domonic.xml.x3d*), 335
`twoSidedMaterial` (in module *domonic.xml.x3d*), 346
`typeset` (class in *domonic.terminal*), 295

U

`u` (class in *domonic.html*), 247
`UIEvent` (class in *domonic.events*), 307
`ul` (class in *domonic.html*), 247
`ulimit` (class in *domonic.terminal*), 295
`umask` (class in *domonic.terminal*), 295
`unalias` (class in *domonic.terminal*), 295
`uname` (class in *domonic.terminal*), 295
`uncompress` (class in *domonic.terminal*), 295
`undefined()` (*domonic.javascript.Global* method), 277
`UNDERSCORE` (*domonic.CDN.CDN_JS* attribute), 299
`unescape()` (*domonic.utils.Utils* static method), 324
`unexpand` (class in *domonic.terminal*), 295
`unget` (class in *domonic.terminal*), 295

`UNHANDLED` (*domonic.events.PromiseRejectionEvent* attribute), 305
`unicodeBidi` (*domonic.style.Style* property), 319
`Uniform` (class in *domonic.xml.x3d*), 335
`uniform` (in module *domonic.xml.x3d*), 346
`uniq` (class in *domonic.terminal*), 295
`unique()` (*domonic.utils.Utils* static method), 324
`UniversalJoint` (class in *domonic.xml.x3d*), 335
`universalJoint` (in module *domonic.xml.x3d*), 346
`unknown` (class in *domonic.svg*), 328
`unlink` (class in *domonic.terminal*), 295
`UNLOAD` (*domonic.events.Event* attribute), 301
`UNLOAD` (*domonic.events.SVGEvent* attribute), 305
`UNPAUSE` (*domonic.events.TweenEvent* attribute), 307
`unpause()` (*domonic.lerpy.tween.Tween* method), 328
`unset` (class in *domonic.terminal*), 295
`unshift()` (*domonic.javascript.Array* method), 271
`untitled()` (*domonic.utils.Utils* static method), 324
`UPDATE_END` (*domonic.events.TweenEvent* attribute), 307
`UPDATE_START` (*domonic.events.TweenEvent* attribute), 307
`uptime` (class in *domonic.terminal*), 295
`URL` (class in *domonic.webapi.url*), 350
`url` (class in *domonic.xml.sitemap*), 350
`url` (*domonic.events.StorageEvent* attribute), 306
`url2file()` (*domonic.utils.Utils* static method), 325
`URLSearchParams` (class in *domonic.webapi.url*), 350
`urlset` (class in *domonic.xml.sitemap*), 350
`use` (class in *domonic.svg*), 328
`useradd` (class in *domonic.terminal*), 295
`userdel` (class in *domonic.terminal*), 295
`users` (class in *domonic.terminal*), 295
`userSelect` (*domonic.style.Style* property), 319
`UTC()` (*domonic.javascript.Date* method), 271
`Utils` (class in *domonic.utils*), 320
`uucp` (class in *domonic.terminal*), 295
`uudecode` (class in *domonic.terminal*), 295
`uuencode` (class in *domonic.terminal*), 295
`uustat` (class in *domonic.terminal*), 295
`uux` (class in *domonic.terminal*), 295

V

`val` (class in *domonic.terminal*), 295
`value` (*domonic.dom.RadioNodeList* property), 266
`values()` (*domonic.dom.NodeList* method), 265
`values()` (*domonic.javascript.Map* method), 278
`values()` (*domonic.webapi.url.URLSearchParams* method), 351
`var` (class in *domonic.html*), 247
`verticalAlign` (*domonic.style.Style* property), 319
`video` (class in *domonic.html*), 247
`view` (class in *domonic.svg*), 328
`Viewfrustum` (class in *domonic.xml.x3d*), 335
`viewfrustum` (in module *domonic.xml.x3d*), 346

Viewpoint (class in domonic.xml.x3d), 335
 viewpoint (in module domonic.xml.x3d), 346
 violatedDirective (domonic.events.SecurityPolicyViolationEvent domonic.xml.x3d), 335
 attribute), 306
 visibility (domonic.style.Style property), 319
 visibilityState (domonic.dom.Document property), 253
 vkern (class in domonic.svg), 328
 VOLUMECHANGE (domonic.events.Event attribute), 301
 VolumeData (class in domonic.xml.x3d), 335
 volumeData (in module domonic.xml.x3d), 346

W

wait (class in domonic.terminal), 295
 WAITING (domonic.events.Event attribute), 301
 waitUntil() (domonic.events.FetchEvent method), 302
 WATER_LATEST (domonic.CDN.CDN_CSS attribute), 298
 wbr (class in domonic.html), 247
 wc (class in domonic.terminal), 295
 webpage() (domonic.javascript.String method), 286
 wget (class in domonic.terminal), 295
 what (class in domonic.terminal), 296
 whatToShow() (domonic.dom.TreeWalker method), 267
 WHEEL (domonic.events.WheelEvent attribute), 307
 WheelEvent (class in domonic.events), 307
 whiteSpace (domonic.style.Style property), 319
 who (class in domonic.terminal), 296
 whoami (class in domonic.terminal), 296
 wholeText (domonic.dom.Text property), 266
 widows (domonic.style.Style property), 320
 width (domonic.style.Style property), 320
 width() (domonic.javascript.Screen method), 280
 Window (class in domonic.javascript), 286
 window (in module domonic.javascript), 288
 wordBreak (domonic.style.Style property), 320
 wordSpacing (domonic.style.Style property), 320
 wordWrap (domonic.style.Style property), 320
 Worker (class in domonic.javascript), 287
 WorldInfo (class in domonic.xml.x3d), 335
 worldInfo (in module domonic.xml.x3d), 346
 write (class in domonic.terminal), 296
 write() (domonic.dom.Document method), 253
 writeln() (domonic.dom.Document method), 253

X

x3d (class in domonic.xml.x3d), 349
 X3D (in module domonic.xml.x3d), 335
 X3DAppearanceChildNode (class in domonic.xml.x3d), 335
 x3DAppearanceChildNode (in module domonic.xml.x3d), 346
 X3DAppearanceNode (class in domonic.xml.x3d), 335
 x3DAppearanceNode (in module domonic.xml.x3d), 346
 X3DBackgroundNode (class in domonic.xml.x3d), 335

x3DBackgroundNode (in module domonic.xml.x3d), 346
 X3DBinaryContainerGeometryNode (class in domonic.xml.x3d), 335
 x3DBinaryContainerGeometryNode (in module domonic.xml.x3d), 346
 X3DBindableNode (class in domonic.xml.x3d), 335
 x3DBindableNode (in module domonic.xml.x3d), 347
 X3DBoundedObject (class in domonic.xml.x3d), 335
 x3DBoundedObject (in module domonic.xml.x3d), 347
 X3DChaserNode (class in domonic.xml.x3d), 335
 x3DChaserNode (in module domonic.xml.x3d), 347
 X3DChildNode (class in domonic.xml.x3d), 335
 x3DChildNode (in module domonic.xml.x3d), 347
 X3DColorNode (class in domonic.xml.x3d), 335
 x3DColorNode (in module domonic.xml.x3d), 347
 X3DComposableVolumeRenderStyleNode (class in domonic.xml.x3d), 335
 x3DComposableVolumeRenderStyleNode (in module domonic.xml.x3d), 347
 X3DComposedGeometryNode (class in domonic.xml.x3d), 335
 x3DComposedGeometryNode (in module domonic.xml.x3d), 347
 X3DCoordinateNode (class in domonic.xml.x3d), 335
 x3DCoordinateNode (in module domonic.xml.x3d), 347
 X3DDamperNode (class in domonic.xml.x3d), 335
 x3DDamperNode (in module domonic.xml.x3d), 347
 X3DDragSensorNode (class in domonic.xml.x3d), 335
 x3DDragSensorNode (in module domonic.xml.x3d), 347
 X3DEnvironmentNode (class in domonic.xml.x3d), 335
 x3DEnvironmentNode (in module domonic.xml.x3d), 347
 X3DEnvironmentTextureNode (class in domonic.xml.x3d), 335
 x3DEnvironmentTextureNode (in module domonic.xml.x3d), 347
 X3DFogNode (class in domonic.xml.x3d), 335
 x3DFogNode (in module domonic.xml.x3d), 347
 X3DFollowerNode (class in domonic.xml.x3d), 335
 x3DFollowerNode (in module domonic.xml.x3d), 347
 X3DFontStyleNode (class in domonic.xml.x3d), 335
 x3DFontStyleNode (in module domonic.xml.x3d), 347
 X3DGeometricPropertyNode (class in domonic.xml.x3d), 335
 x3DGeometricPropertyNode (in module domonic.xml.x3d), 347
 X3DGeometryNode (class in domonic.xml.x3d), 336
 x3DGeometryNode (in module domonic.xml.x3d), 347
 X3DGroupingNode (class in domonic.xml.x3d), 336
 x3DGroupingNode (in module domonic.xml.x3d), 347
 X3DInfoNode (class in domonic.xml.x3d), 336
 x3DInfoNode (in module domonic.xml.x3d), 347
 X3DInterpolatorNode (class in domonic.xml.x3d), 336

`x3DInterpolatorNode` (in module *domonic.xml.x3d*), 347

`X3DLightNode` (class in *domonic.xml.x3d*), 336

`x3DLightNode` (in module *domonic.xml.x3d*), 348

`X3DLODNode` (class in *domonic.xml.x3d*), 336

`x3DLODNode` (in module *domonic.xml.x3d*), 348

`X3DMaterialNode` (class in *domonic.xml.x3d*), 336

`x3DMaterialNode` (in module *domonic.xml.x3d*), 348

`X3DMetadataObject` (class in *domonic.xml.x3d*), 336

`x3DMetadataObject` (in module *domonic.xml.x3d*), 348

`X3DNavigationInfoNode` (class in *domonic.xml.x3d*), 336

`x3DNavigationInfoNode` (in module *domonic.xml.x3d*), 348

`X3DNBodyCollidableNode` (class in *domonic.xml.x3d*), 336

`x3DNBodyCollidableNode` (in module *domonic.xml.x3d*), 348

`X3DNode` (class in *domonic.xml.x3d*), 336

`x3DNode` (in module *domonic.xml.x3d*), 348

`X3DOM` (*domonic.CDN.CDN_CSS* attribute), 298

`X3DOM` (*domonic.CDN.CDN_JS* attribute), 299

`X3DPlanarGeometryNode` (class in *domonic.xml.x3d*), 336

`x3DPlanarGeometryNode` (in module *domonic.xml.x3d*), 348

`X3DPointingDeviceSensorNode` (class in *domonic.xml.x3d*), 336

`x3DPointingDeviceSensorNode` (in module *domonic.xml.x3d*), 348

`X3DRigidJointNode` (class in *domonic.xml.x3d*), 336

`x3DRigidJointNode` (in module *domonic.xml.x3d*), 348

`X3DSensorNode` (class in *domonic.xml.x3d*), 336

`x3DSensorNode` (in module *domonic.xml.x3d*), 348

`X3DShaderNode` (class in *domonic.xml.x3d*), 336

`x3DShaderNode` (in module *domonic.xml.x3d*), 348

`X3DShapeNode` (class in *domonic.xml.x3d*), 336

`x3DShapeNode` (in module *domonic.xml.x3d*), 348

`X3DSoundNode` (class in *domonic.xml.x3d*), 336

`x3DSoundNode` (in module *domonic.xml.x3d*), 348

`X3DSoundSourceNode` (class in *domonic.xml.x3d*), 336

`x3DSoundSourceNode` (in module *domonic.xml.x3d*), 348

`X3DSpatialGeometryNode` (class in *domonic.xml.x3d*), 336

`x3DSpatialGeometryNode` (in module *domonic.xml.x3d*), 348

`X3DTexture3DNode` (class in *domonic.xml.x3d*), 336

`x3DTexture3DNode` (in module *domonic.xml.x3d*), 348

`X3DTextureCoordinateNode` (class in *domonic.xml.x3d*), 336

`x3DTextureCoordinateNode` (in module *domonic.xml.x3d*), 348

`X3DTextureNode` (class in *domonic.xml.x3d*), 336

`x3DTextureNode` (in module *domonic.xml.x3d*), 348

`X3DTextureTransformNode` (class in *domonic.xml.x3d*), 336

`x3DTextureTransformNode` (in module *domonic.xml.x3d*), 348

`X3DTimeDependentNode` (class in *domonic.xml.x3d*), 336

`x3DTimeDependentNode` (in module *domonic.xml.x3d*), 349

`X3DTouchSensorNode` (class in *domonic.xml.x3d*), 336

`x3DTouchSensorNode` (in module *domonic.xml.x3d*), 349

`X3DTransformNode` (class in *domonic.xml.x3d*), 336

`x3DTransformNode` (in module *domonic.xml.x3d*), 349

`X3DVertexAttributeNode` (class in *domonic.xml.x3d*), 336

`x3DVertexAttributeNode` (in module *domonic.xml.x3d*), 349

`X3DViewpointNode` (class in *domonic.xml.x3d*), 336

`x3DViewpointNode` (in module *domonic.xml.x3d*), 349

`X3DVolumeDataNode` (class in *domonic.xml.x3d*), 336

`x3DVolumeDataNode` (in module *domonic.xml.x3d*), 349

`X3DVolumeRenderStyleNode` (class in *domonic.xml.x3d*), 336

`x3DVolumeRenderStyleNode` (in module *domonic.xml.x3d*), 349

`xargs` (class in *domonic.terminal*), 296

`xmlversion` (*domonic.dom.Document* property), 253

`xmp` (class in *domonic.html*), 247

Y

`yacc` (class in *domonic.terminal*), 296

Z

`zcat` (class in *domonic.terminal*), 296

`zIndex` (*domonic.style.Style* property), 320