

webbrowser — Displays web pages

Purpose: Use the *webbrowser* module to display web pages to your users.

The *webbrowser* module includes functions to open URLs in interactive browser applications. The module includes a registry of available browsers, in case multiple options are available on the system. It can also be controlled with the `BROWSER` environment variable.

Simple Example

To open a page in the browser, use the `open()` function.

```
# webbrowser_open.py

import webbrowser

webbrowser.open(
    'https://docs.python.org/3/library/webbrowser.html'
)
```

The URL is opened in a browser window, and that window is raised to the top of the window stack. The documentation says that an existing window will be reused, if possible, but the actual behavior may depend on your browser's settings. Using Firefox on Mac OS X, a new window was always created.

Windows vs. Tabs

If you always want a new window used, use `open_new()`.

```
# webbrowser_open_new.py

import webbrowser

webbrowser.open_new(
    'https://docs.python.org/3/library/webbrowser.html'
)
```

If you would rather create a new tab, use `open_new_tab()` instead.

Using a specific browser

If for some reason your application needs to use a specific browser, you can access the set of registered browser controllers using the `get()` function. The browser controller has methods to `open()`, `open_new()`, and `open_new_tab()`. This example forces the use of the *lynx* browser:

```
# webbrowser_get.py

import webbrowser

b = webbrowser.get('lynx')
b.open('https://docs.python.org/3/library/webbrowser.html')
```

Refer to the module documentation for a list of available browser types.

BROWSER variable

Users can control the module from outside your application by setting the environment variable `BROWSER` to the browser names or commands to try. The value should consist of a series of browser names separated by `os.pathsep`. If the name includes `%s`, the name is interpreted as a literal command and executed directly with the `%s` replaced by the URL. Otherwise, the name is passed to `get()` to obtain a controller object from the registry.

For example, this command opens the web page in lynx, assuming it is available, no matter what other browsers are registered.

```
$ BROWSER=lynx python3 webbrowser_open.py
```

If none of the names in BROWSER work, webbrowser falls back to its default behavior.

Command Line Interface

All of the features of the webbrowser module are available via the command line as well as from within your Python program.

```
$ python3 -m webbrowser
```

```
Usage: .../lib/python3.7/webbrowser.py [-n | -t] url
```

```
-n: open new window
```

```
-t: open new tab
```

See also

- [Standard library documentation for webbrowser](#)
- [What the What?](#) - Runs your Python program and then launches a Google search for any exception message produced.

[http.cookies — HTTP Cookies](#)

[uuid — Universally Unique Identifiers](#)

Quick Links

[Simple Example](#)

[Windows vs. Tabs](#)

[Using a specific browser](#)

[BROWSER](#)

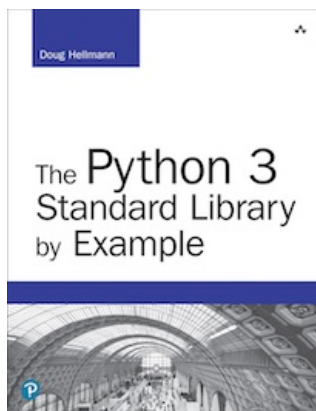
[Command Line Interface](#)

This page was last updated 2018-12-09.

Navigation

[http.cookies — HTTP Cookies](#)

[uuid — Universally Unique Identifiers](#)



[Get the book](#)

The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for [examples for Python 2?](#)

This Site

[Module Index](#)

[Index](#)



© Copyright 2019, Doug Hellmann



Other Writing

 [Blog](#)

 [The Python Standard Library By Example](#)