

Webbrowser Module Python 2.x 2.0–2.7, Python 3.x 3.0–3.6

According to Python's standard documentation, the `webbrowser` module provides a high-level interface to allow displaying Web-based documents to users. This topic explains and demonstrates proper usage of the `webbrowser` module.

Examples

Opening a URL with Default Browser

To simply open a URL, use the `webbrowser.open()` method:

```
import webbrowser
webbrowser.open("http://stackoverflow.com")
```

If a browser window is currently open, the method will open a new tab at the specified URL. If no window is open, the method will open the operating system's default browser and navigate to the URL in the parameter. The `open` method supports the following parameters:

- `url` - the URL to open in the web browser (string) **[required]**
- `new` - 0 opens in existing tab, 1 opens new window, 2 opens new tab (integer) **[default 0]**
- `autoraise` - if set to True, the window will be moved on top of other applications' windows (Boolean) **[default False]**

Note, the `new` and `autoraise` arguments rarely work as the majority of modern browsers refuse these commands.

Webbrowser can also try to open URLs in new windows with the `open_new` method:

```
import webbrowser
webbrowser.open_new("http://stackoverflow.com")
```

This method is commonly ignored by modern browsers and the URL is usually opened in a new tab. Opening a new tab can be tried by the module using the `open_new_tab` method:

```
import webbrowser
webbrowser.open_new_tab("http://stackoverflow.com")
```

Opening a URL with Different Browsers

The `webbrowser` module also supports different browsers using the `register()` and `get()` methods. The `get` method is used to create a browser controller using a specific executable's path and the `register` method is used to attach these executables to preset browser types for future use, commonly when multiple browser types are used.

```
import webbrowser
ff_path = webbrowser.get("C:/Program Files/Mozilla Firefox/firefox.exe")
ff = webbrowser.get(ff_path)
ff.open("http://stackoverflow.com/")
```

Registering a browser type:

```
import webbrowser
ff_path = webbrowser.get("C:/Program Files/Mozilla Firefox/firefox.exe")
ff = webbrowser.get(ff_path)
webbrowser.register('firefox', None, ff)
# Now to refer to use Firefox in the future you can use this
webbrowser.get('firefox').open("https://stackoverflow.com/")
```

Syntax

```
webbrowser.open(url, new=0, autoraise=False)
```

```
webbrowser.open_new(url)
```

```
webbrowser.open_new_tab(url)
```

```
webbrowser.get(usage=None)
```

```
webbrowser.register(name, constructor, instance=None)
```

Parameters

Parameter	Details
<code>webbrowser.open()</code>	
<code>url</code>	the URL to open in the web browser
<code>new</code>	0 opens the URL in the existing tab, 1 opens in a new window, 2 opens in new tab
<code>autoraise</code>	if set to True, the window will be moved on top of the other windows
<code>webbrowser.open_new()</code>	
<code>url</code>	the URL to open in the web browser
<code>webbrowser.open_new_tab()</code>	
<code>url</code>	the URL to open in the web browser
<code>webbrowser.get()</code>	
<code>using</code>	the browser to use
<code>webbrowser.register()</code>	
<code>url</code>	browser name
<code>constructor</code>	path to the executable browser (help)
<code>instance</code>	An instance of a web browser returned from the <code>webbrowser.get()</code> method

Remarks

The following table lists predefined browser types. The left column are names that can be passed into the `webbrowser.get()` method and the right column lists the class names for each browser type.

Type Name	Class Name
'mozilla'	Mozilla('mozilla')
'firefox'	Mozilla('mozilla')
'netscape'	Mozilla('netscape')
'galeon'	Galeon('galeon')
'epiphany'	Galeon('epiphany')
'skipstone'	BackgroundBrowser('skipstone')
'kfmclient'	Konqueror()
'konqueror'	Konqueror()
'kfm'	Konqueror()
'mosaic'	BackgroundBrowser('mosaic')
'opera'	Opera()
'grail'	Grail()
'links'	GenericBrowser('links')
'elinks'	Elinks('elinks')
'lynx'	GenericBrowser('lynx')
'w3m'	GenericBrowser('w3m')
'windows-default'	WindowsDefault
'macosx'	MacOSX('default')
'safari'	MacOSX('safari')
'google-chrome'	Chrome('google-chrome')

Type Name	Class Name
'chromium'	Chromium('chromium')
'chromium-browser'	Chromium('chromium-browser')