# Browser commands in Selenium Python bindings

Table of content

## Open Browser:

For any browser-related automation testing, opening a browser is a must. Selenium does not have the capability to work with an already existing browser.

So whenever we want to test something with selenium, we have to open a new browser and test it.

### Open Firefox in selenium python

Till firefox 48, selenium was supporting firefox by default but from the firefox 48 version, the firefox team has changed their internal structure, so selenium stopped supporting Firefox by default.

We have to set the connecting executable file between Selenium python and Firefox browser. We can pass the **executable_path** property to the Firefox class constructor, to open the firefox browser in selenium python.

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Firefox(executable_path='D:\PATH\geckodriver.exe');
```

## Open Chrome in python

From the beginning years, Chrome provided an executable file to connect the python bindings of selenium with the chrome browser

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Chrome(executable_path=r'D:\PATH\chromedriver.exe');
```

## Open Internet Explorer in python selenium

We can also open Internet Explorer by setting the correct path of the executables.

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Ie(executable_path=r'D:\PATH\IEdriverserver.exe');
```

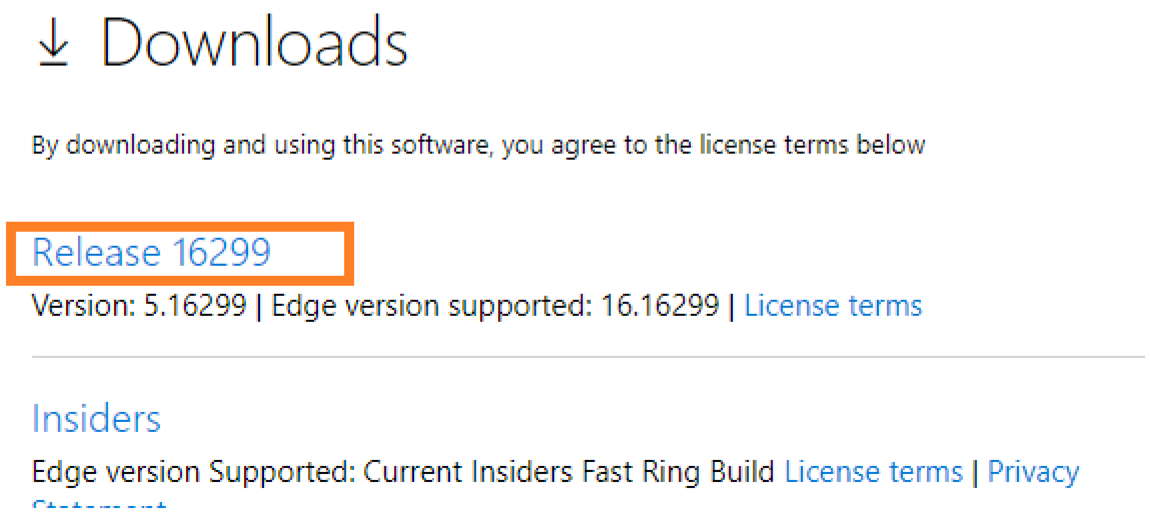
## Open Microsoft Edge Browser in python selenium

For opening Edge browser we have to set the driverserver executable like all other browsers.

But for the Microsoft Edge browser, we have to provide the exact driver server based on your operating system build.

You can download the edge driver server from this

URL [https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/](https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/)



↓ Downloads

By downloading and using this software, you agree to the license terms below

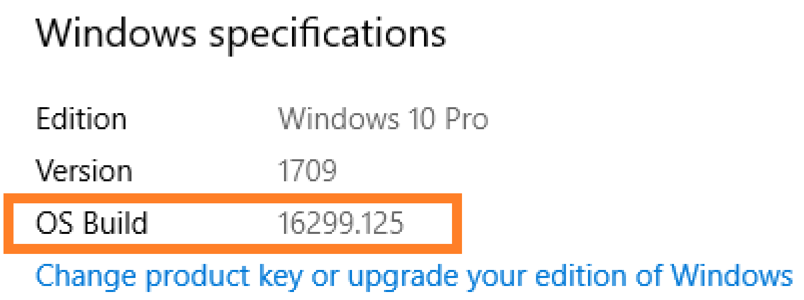
Release 16299

Version: 5.16299 | Edge version supported: 16.16299 | License terms

Insiders

Edge version Supported: Current Insiders Fast Ring Build License terms | Privacy Statement

To find your correct build number: Go to **Start** > **Settings** > **System** > **About** and locate the number next to OS Build on the screen.



Windows specifications

| | |
|---|---|
| Edition | Windows 10 Pro |
| Version | 1709 |
| OS Build | 16299.125 |

Change product key or upgrade your edition of Windows

If you do not have the correct build for the driverserver executable, then the Edge browser may not work as expected

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Ie(executable_path=r'D:\PATH\IEdriverserver.exe');
```

## Close and Quit Browsers

python selenium Bindings provides an option to close the browser as well.

## Close the browser

When calling the method **close()** selenium python closes the browser window. **close()** method closes only the current window on which the selenium has control.

For example, Selenium opened three windows and control is on the third window, if call the close() method selenium closes the third window alone, window 1 and two 2 are not affected.

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Edge(executable_path=r'D:\PATH\MicrosoftWebDriver.exe');
// closes current window
driver.close()
```

## Close the browser in python selenium bindings

quit() method not only closes current window, it closes all the windows opened by the driver. In the above example, if you call the quit() method, it closes window 1, 2, 3

quit() method terminates the driverserver executable process as well.

```
# closes all browser windows
driver.quit()
```

# Open Webpage in selenium python

Opening a webpage is a more important part of web application testing, we can use get() present in the selenium to open a webpage.

get() will not pass the control of the program until the page loads, when page loading is finished then control goes to the next line of code.

Whenever we pass the website address to the get() method, it first checks for the protocol in the address. Selenium python throws exceptions If the protocol is not present in the URL of the website.

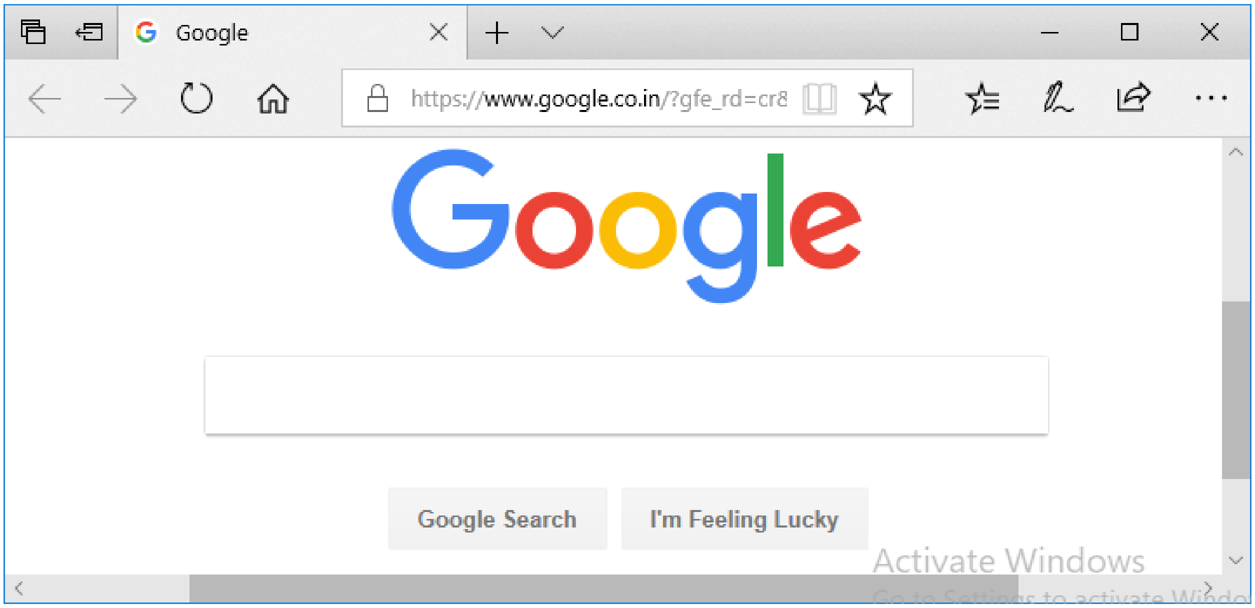
## What is protocol?

Protocol is prefixed to mention what kind of site is. Examples for protocol:

- http
- https
- ftp
- file

Program to open a website

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Edge(executable_path=r'D:\PATH\MicrosoftWebDriver.exe');
# open website
driver.get("http://google.com")
```


Google
https://www.google.co.in/?gfe_rd=cr8
Google
Google Search
I'm Feeling Lucky
Activate Windows
Go to Settings to activate Windo

## Work offline in python selenium

If you are not connected to the internet but if you have a page, which you had stored when there was an internet. You can access such kinds of offline pages in selenium.

Instead of giving the website address, give the local address of the stored page.

```
# open website
driver.get("file:///C:/Users/path/selenium.html")
```

## Browser Size in selenium python

Nowadays every almost every website is developed using Bootstrap or some other responsive frameworks.

The responsive website fits into all size screens, according to the screen size, all the web elements are wrapped.

To test this kind of websites we have to change the screen size in selenium

## Methods to change the screen Size

Below are few methods, with which we can manipulate the browser window sizes

1. maximize_window
2. fullscreen_window
3. set_window_size
4. set_window_rect (may not work)
5. set_window_position
6. get_window_position
7. get_window_rect (may not work)
8. get_window_size
9. minimize_window (may not work)

### *maximize_window*

maximize_window() method in selenium python, maximizes the current browser window

Few browsers open in the maximized state and few browsers may not open in maximized state. Sometimes we have scenarios to be performed on the maximized browser. In such cases, we can use the maximize_window method to maximize the window.

If you call the maximize_window() method on the browser which is already in a maximized state, then this method will not have any effect on that browser.

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Edge(executable_path=r'D:\PATH\MicrosoftWebDriver.exe');
# open website
driver.get("http://google.com")
# maximize browser window
driver.maximize_window()
```

## *fullscreen_window*

fullscreen_window() in python selenium bindings, helps the user to set the browser size to full screen. When you make the browser to full screen, you cannot see the title bar, address bar, URL bar in the browser, you can only view the webpage.

By pressing the F11 key, you can achieve full screen manually. Selenium will not open the full screen on browser start-up itself, based on the occurrence of the fullscreen_window method; selenium changes the size to full screen.

In the below example python selenium bindings will change the browser to full screen once, the browser opens the Google page.

```
# open website
driver.get("http://google.com")
# browser full window
driver.fullscreen_window()
```

## *set_window_size(width, height, windowHandle)*

set_window_size method helps the user to resize the window according to the need of the user, this method will be helpful, when you want to test the responsive website.

This method accepts the height, width of the browser also accepts which browser window we want to resize, if we do not pass the browser window parameter then selenium python bindings consider the current window as the target window.

```
# open website
driver.get("http://google.com")
# browser window size
driver.set_window_size(200, 300)
```

## *set_window_rect(x, y, width, height)*

set_window_rect method not only resizes the browser window but also can set the position of the browser window by providing the x and y coordinate.

Note: Sometimes this command may not work in the new version of selenium.

```
# browser window rectangle
driver.set_window_rect(x=10, y=10, width=100, height=200)
```

## *set_window_position*

set_window_position method sets the coordinates for the browser starting point, top right corner is considered as (0, 0) position.

```
# open website
driver.get("http://google.com")
```

```
# browser window position
driver.set_window_position(x=200, y=50)
```

### *get_window_position*

Sometimes not only setting a browser size is required, but we may also need to get the size as well to perform some operations based on the size of the window.

We can use the get_window_position() method to retrieve the position of the window. This method returns a dictionary containing x and y.

```
# get position of the window
driver.get_window_position()
```

### *get_window_size*

get_window_size method returns the width and height of the current window, this method returns a dictionary.

```
# get size of the window
driver.get_window_size()
```

## WebPage Properties

Webpage properties are nothing but the URL, Page title, page source code, you can use these values for assertion.

### Page URL in Selenium python

Page is URL is nothing but the address of the website which is present in the address bar of the browser. Using the current_url variable, we can fetch the URL of the page.

```
# open website
driver.get("http://google.com")
print("URL : " + driver.current_url)
```

## Page title in Selenium python

We can retrieve the webpage title using the title variable present in the driver, the title is a variable, not a method.

```
driver.get("http://google.com")
print("Title : " + driver.title)
```

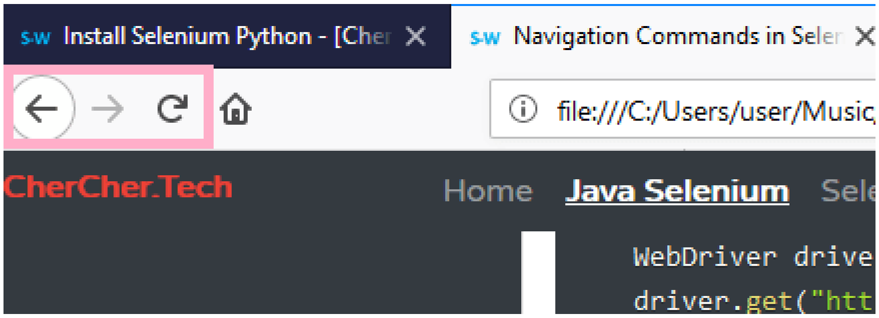
## Page title in Selenium python

We can retrieve the page source code using the page_source variable present in the driver, page source is nothing but the HTML code used to develop the webpage. If they have used javascript ways of replacing the text then you may get only the javascript HTML

```
# open website
driver.get("http://google.com")
print("Title : " + driver.title)
print("URL : " + driver.current_url)
print("Page source : " + driver.page_source)
```

# Navigation Commands in selenium python

Navigation commands are some which enable the user to navigate through the history of the browser like back, forth, refresh. Basically, these navigation commands are nothing but the browser navigation buttons, We can use these commands only when these buttons are enabled. Buttons are shown in the below image.


Install Selenium Python - [Cher
Navigation Commands in Seler
file:///C:/Users/user/Music
CherCher.Tech
Home
Java Selenium
Sele

```
WebDriver drive
driver.get("htt
```

## back()

back() method navigates the user back to the last page, For example, you navigate from Google to bing, and if you use the back() method on the bing page, the selenium navigates the page to the google page.

To use the back() method you must at least moved from one page to another page at least once, his method will not work if you have not opened any page but you trying to use the back() method.

```
#CherCherTech
# import the webdriver
from selenium import webdriver
# set exe path and open the browser.
driver = webdriver.Chrome(executable_path=r'D:\PATH\chromedriver.exe');
# open website
driver.get("http://google.com")
print("1. Title is :"+driver.title)
driver.get("http://bing.com")
print("2. Title is :"+driver.title)
# navigate back
driver.back()
print("After navigating back Title is :"+driver.title)
```

## forward()

forward() helps the user to navigate forward when already is moved back, using the back() method is a must before using forward() method otherwise there is no use.

```
# open website
driver.get("http://google.com")
print("1. Title is :"+driver.title)
driver.get("http://bing.com")
print("2. Title is :"+driver.title)
# navigate back
driver.back()
print("After navigating back Title is :"+ driver.title)
driver.forward()
print("After moving forward Title is : "+ driver.title)
```

**refresh(): refresh method refreshes the current webpage and loads the same page.**

```
# open website
driver.get("http://google.com")
print("1. Title is :"+driver.title)
driver.get("http://bing.com")
print("2. Title is :"+driver.title)
# navigate back
driver.back()
print("After navigating back Title is :"+ driver.title)
driver.forward()
print("After moving forward Title is : "+ driver.title)
driver.refresh()
print("After refreshing Title is : "+driver.title)
```

-----END-----

Muntasir Abdullah Mizan
muntasir.abdullah01@gmail.com