Open Browser in selenium python

For any browser-related automation testing, opening a browser is a must.**Selenium does not have the capability to work with the 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 the firefox by default but from firefox 48 version, 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 Firefox class constructor**, to open the firefox browser in selenium python.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Firefox(executable\_path=r'D:PATHgeckodriver.exe');**

**Open Chrome in python**

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

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**Open Internet Explorer in python selenium**

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

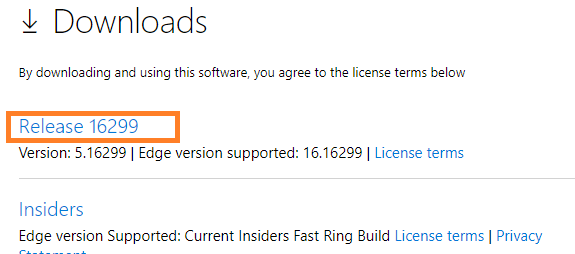
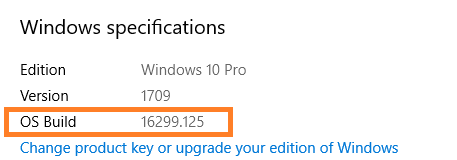
**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Ie(executable\_path=r'D:PATHIEdriverserver.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 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/>  
  
  
  
To find your correct build number: Go to Start > Settings > System > About and locate the number next to OS Build on the screen.  
  
  
  
If you donot have correct build for the driverserver executable, then Edge browser may not work as expected

**​**

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Ie(executable\_path=r'D:PATHIEdriverserver.exe');**

[Handle Alerts in Selenium Python](https://chercher.tech/python/alerts-selenium-python)

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 bindings close the browser window. close() method closes only the current window on which the selenium have 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.

**​**

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**// closes current window**

**driver.close()**

**Close the browser in python selenium bindings**

quit() method not only closes the current window, but it also closes all the windows opened by the driver. In the above example, if you call quit() method, it closes window 1, 2, 3  
  
quit() method terminates the driverserver executable process as well.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# closes all browser windows**

**driver.quit()**

Open Webpage in selenium python

Opening a webpage is a more important part in the web application testing, we can use **get()** present in the selenium to open a webpage.  
  
**get()** will not pass the control of 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 of checks for the protocol in the address  
  
**Selenium python throws exception If the protocol is not present in the URL of the website.**  
  
**What is the protocol ?**  
The protocol is prefix to mention what kind of site is that. Examples for protocol:

* *http*
* *https*
* *ftp*
* *file*

Program to open a website

**​**

**# import the webdriver**

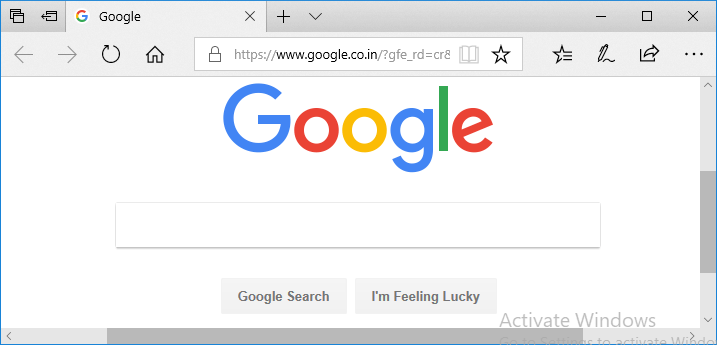
**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# open website**

**driver.get("https://google.com")**



**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 the internet. You can access such kind o offline pages in selenium.  
  
Instead of giving the website address, give the local address of the stored page.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# open website**

**driver.get("file:///C:/Users/path/selenium.html")**

[Screenshots in Selenium Python](https://chercher.tech/python/screenshot-selenium-python)

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 webelements 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 a few methods, with which we can manipulate the browser window sizes

* *maximize\_window*
* *fullscreen\_window*
* *set\_window\_size*
* *set\_window\_rect (may not work)*
* *set\_window\_position*
* *get\_window\_position*
* *get\_window\_rect (may not work)*
* *get\_window\_size*
* *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 a scenario to be performed on the maximized browser. In such cases, we can use maximize\_window method to maximize the window.  
  
If you call **maximize\_window()**method on the browser which is already in a maximized state, then this method will not have any effect on that browser.

**​**

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# open website**

**driver.get("https://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 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 below example python selenium bindings will change the browser to full screen once, the browser opens the Google page.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://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 donot pass browser window parameter then selenium python bindings considers the current window as the target window.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'DPATHchromedriver.exe');**

**# open website**

**driver.get("https://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 we can set the position of the browser window by providing the x and y coordinate.

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.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

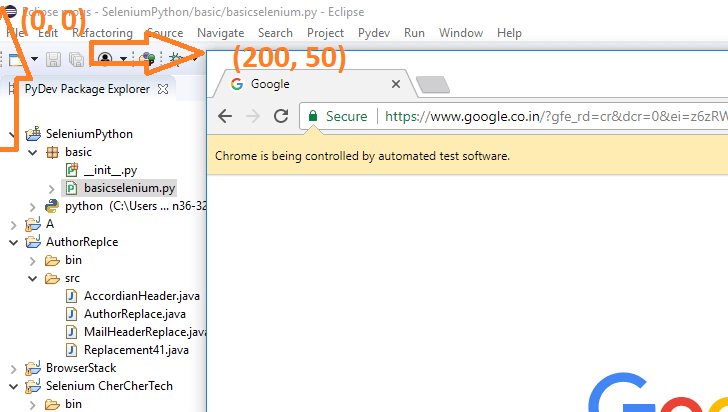
**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://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 **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()**

[Xpath In Selenium Python](https://chercher.tech/python/relative-xpath-selenium-python)

WebPage Properties

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

**Page URL in Selenium python**

The page is URL is nothing but the address of the website which is present in the address bar of the browser. Using current\_url variable, we can fetch the url of the page.

**​# open website**

**driver.get("https://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 variable not a method.

**driver.get("https://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 javascripted HTML

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

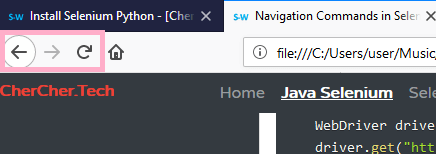
**print("Title : " + driver.title)**

**print("URL : " + driver.current\_url)**

**print("Page source : " + driver.page\_source)**

[WebElement Locators in Selenium Python](https://chercher.tech/python/webelement-locator)

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.  
  


**back()**

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

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

**print("1. Title is :"+driver.title)**

**driver.get("https://bing.com")**

**print("2. Title is :"+driver.title)**

**# navigate back**

**driver.back()**

**print("After navigating back Title is :"+driver.title)**

**forward()**

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

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

**print("1. Title is :"+driver.title)**

**driver.get("https://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.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

**print("1. Title is :"+driver.title)**

**driver.get("https://bing.com")**

**print("2. Title is :"+driver.title)**

**# navigate back**

**driver.back()**

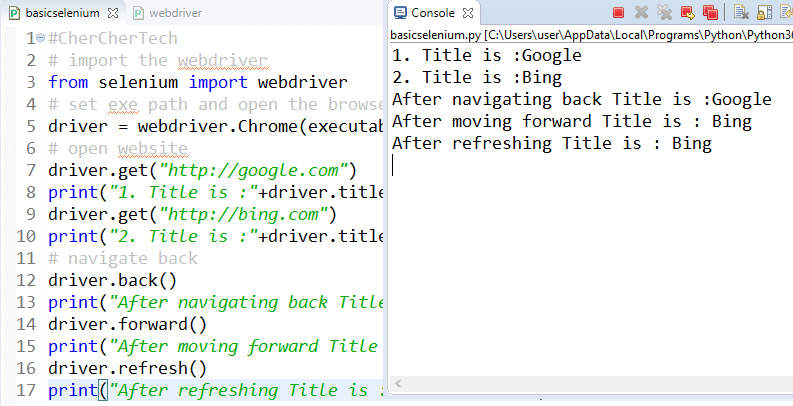
**print("After navigating back Title is :"+ driver.title)**

**driver.forward()**

**print("After moving forward Title is : "+ driver.title)**

**driver.refresh()**

**print("After refreshing Title is : "+driver.title)**



**Different ways to Refresh a webpage**

We can refresh a webpage using multiple ways other than refresh() methods, I have listed a few hereinbelow.

**1. driver.get(driver.current\_url)**

**2. driver.find\_element\_by\_tag\_name("body").send\_keys(Keys.F5)**

**3. driver.refresh()**

**4. driver.find\_element\_by\_tag\_name("body").send\_keys("uE035")**

Open Browser in selenium python

For any browser-related automation testing, opening a browser is a must.**Selenium does not have the capability to work with the 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 the firefox by default but from firefox 48 version, 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 Firefox class constructor**, to open the firefox browser in selenium python.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Firefox(executable\_path=r'D:PATHgeckodriver.exe');**

**Open Chrome in python**

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

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**Open Internet Explorer in python selenium**

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

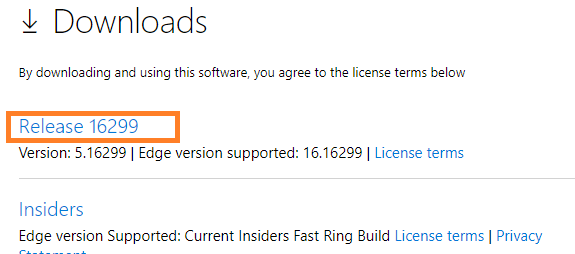
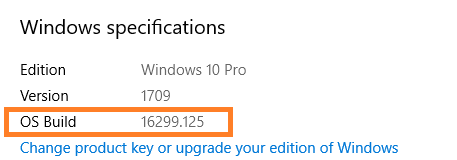
**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Ie(executable\_path=r'D:PATHIEdriverserver.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 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/>  
  
  
  
To find your correct build number: Go to Start > Settings > System > About and locate the number next to OS Build on the screen.  
  
  
  
If you donot have correct build for the driverserver executable, then Edge browser may not work as expected

**​**

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Ie(executable\_path=r'D:PATHIEdriverserver.exe');**

[Handle Alerts in Selenium Python](https://chercher.tech/python/alerts-selenium-python)

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 bindings close the browser window. close() method closes only the current window on which the selenium have 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.

**​**

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**// closes current window**

**driver.close()**

**Close the browser in python selenium bindings**

quit() method not only closes the current window, but it also closes all the windows opened by the driver. In the above example, if you call quit() method, it closes window 1, 2, 3  
  
quit() method terminates the driverserver executable process as well.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# closes all browser windows**

**driver.quit()**

[Handle Browser Windows in Selenium Python](https://chercher.tech/python/windows-selenium-python)

Open Webpage in selenium python

Opening a webpage is a more important part in the web application testing, we can use **get()** present in the selenium to open a webpage.  
  
**get()** will not pass the control of 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 of checks for the protocol in the address  
  
**Selenium python throws exception If the protocol is not present in the URL of the website.**  
  
**What is the protocol ?**  
The protocol is prefix to mention what kind of site is that. Examples for protocol:

* *http*
* *https*
* *ftp*
* *file*

Program to open a website

**​**

**# import the webdriver**

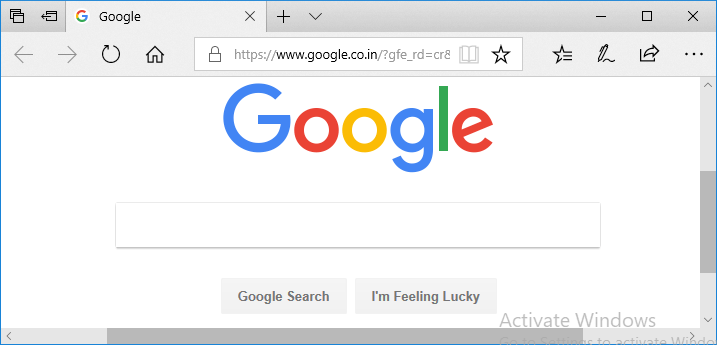
**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# open website**

**driver.get("https://google.com")**



**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 the internet. You can access such kind o offline pages in selenium.  
  
Instead of giving the website address, give the local address of the stored page.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# open website**

**driver.get("file:///C:/Users/path/selenium.html")**

[Screenshots in Selenium Python](https://chercher.tech/python/screenshot-selenium-python)

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 webelements 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 a few methods, with which we can manipulate the browser window sizes

* *maximize\_window*
* *fullscreen\_window*
* *set\_window\_size*
* *set\_window\_rect (may not work)*
* *set\_window\_position*
* *get\_window\_position*
* *get\_window\_rect (may not work)*
* *get\_window\_size*
* *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 a scenario to be performed on the maximized browser. In such cases, we can use maximize\_window method to maximize the window.  
  
If you call **maximize\_window()**method on the browser which is already in a maximized state, then this method will not have any effect on that browser.

**​**

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Edge(executable\_path=r'D:PATHMicrosoftWebDriver.exe');**

**# open website**

**driver.get("https://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 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 below example python selenium bindings will change the browser to full screen once, the browser opens the Google page.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://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 donot pass browser window parameter then selenium python bindings considers the current window as the target window.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'DPATHchromedriver.exe');**

**# open website**

**driver.get("https://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 we can set the position of the browser window by providing the x and y coordinate.

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.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

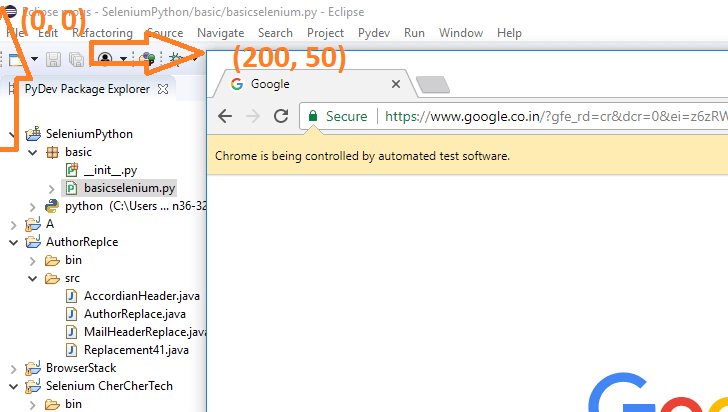
**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://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 **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()**

[Xpath In Selenium Python](https://chercher.tech/python/relative-xpath-selenium-python)

WebPage Properties

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

**Page URL in Selenium python**

The page is URL is nothing but the address of the website which is present in the address bar of the browser. Using current\_url variable, we can fetch the url of the page.

**​# open website**

**driver.get("https://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 variable not a method.

**driver.get("https://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 javascripted HTML

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

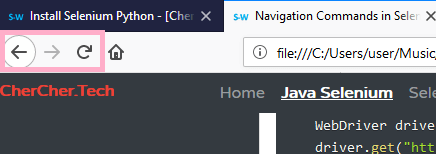
**print("Title : " + driver.title)**

**print("URL : " + driver.current\_url)**

**print("Page source : " + driver.page\_source)**

[WebElement Locators in Selenium Python](https://chercher.tech/python/webelement-locator)

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.  
  


**back()**

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

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

**print("1. Title is :"+driver.title)**

**driver.get("https://bing.com")**

**print("2. Title is :"+driver.title)**

**# navigate back**

**driver.back()**

**print("After navigating back Title is :"+driver.title)**

**forward()**

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

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

**print("1. Title is :"+driver.title)**

**driver.get("https://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.

**# import the webdriver**

**from selenium import webdriver**

**# set exe path and open the browser.**

**driver = webdriver.Chrome(executable\_path=r'D:PATHchromedriver.exe');**

**# open website**

**driver.get("https://google.com")**

**print("1. Title is :"+driver.title)**

**driver.get("https://bing.com")**

**print("2. Title is :"+driver.title)**

**# navigate back**

**driver.back()**

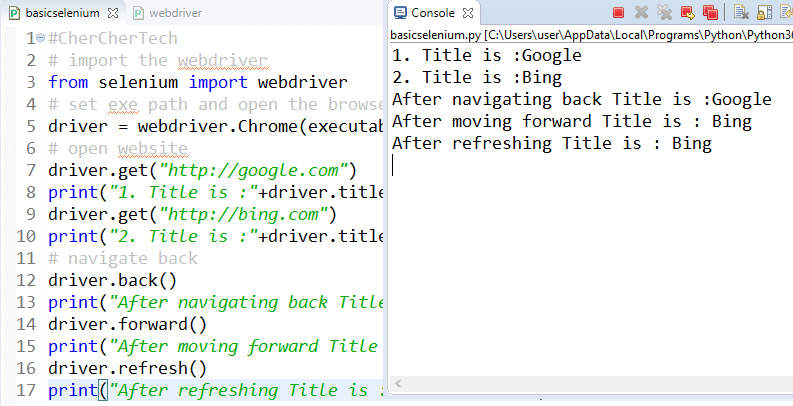
**print("After navigating back Title is :"+ driver.title)**

**driver.forward()**

**print("After moving forward Title is : "+ driver.title)**

**driver.refresh()**

**print("After refreshing Title is : "+driver.title)**



**Different ways to Refresh a webpage**

We can refresh a webpage using multiple ways other than refresh() methods, I have listed a few hereinbelow.

**1. driver.get(driver.current\_url)**

**2. driver.find\_element\_by\_tag\_name("body").send\_keys(Keys.F5)**

**3. driver.refresh()**

**4. driver.find\_element\_by\_tag\_name("body").send\_keys("uE035")**