# Selenium Python Tutorial

Selenium is a powerful tool for controlling web browsers through programs and performing browser automation. It is functional for all browsers, works on all major OS and its scripts are written in various languages i.e Python, Java, C#, etc, we will be working with Python. Selenium Tutorial covers all topics such as – WebDriver, WebElement, Unit Testing with selenium. This Python Selenium Tutorial covers Selenium from basics to advanced and professional uses.

### Why learn Selenium Python ?

* Open Source and Portable – Selenium is an open source and portable Web testing Framework.
* Combination of tool and DSL – Selenium is combination of tools and DSL (Domain Specific Language) in order to carry out various types of tests.
* Easier to understand and implement – Selenium commands are categorized in terms of different classes which make it easier to understand and implement.
* Less burden and stress for testers – As mentioned above, the amount of time required to do testing repeated test scenarios on each and every new build is reduced to zero, almost. Hence, the burden of tester gets reduced.
* Cost reduction for the Business Clients – The Business needs to pay the testers their salary, which is saved using automation testing tool. The automation not only saves time but gets cost benefits too, to the business.

### Selenium WebDriver

Selenium Basics

* Selenium Basics
* Components of Selenium
* Applications and Uses
* Features
* Limitations

Selenium Python Basics

* Selenium Python Introduction and Installation
* Navigating links using get method
* Interacting with Webpage
* Locating single elements
* Locating multiple elements
* Locator Strategies – Selenium Python
* Writing Tests using Selenium Python

Locating Strategies

* Locating Single Elements –
  + find\_element\_by\_id()
  + find\_element\_by\_name()
  + find\_element\_by\_xpath()
  + find\_element\_by\_link\_text()
  + find\_element\_by\_partial\_link\_text()
  + find\_element\_by\_tag\_name()
  + find\_element\_by\_class\_name()
  + find\_element\_by\_css\_selector()
* Locating Multiple Elements –
  + find\_elements\_by\_name()
  + find\_elements\_by\_xpath()
  + find\_elements\_by\_link\_text()
  + find\_element\_by\_partial\_link\_text()
  + find\_elements\_by\_tag\_name()
  + find\_elements\_by\_class\_name()
  + find\_elements\_by\_css\_selector()

Waits

* Explicit waits
* Implicit Waits

Action Chains

* Action Chains Basics
* click
* click\_and\_hold
* context\_click
* double\_click
* drag\_and\_drop
* key\_down
* key\_up
* move\_by\_offset
* move\_to\_element
* move\_to\_element\_with\_offset
* release
* reset\_actions
* send\_keys

Advanced in Selenium Python –

* Handling Exceptions – Selenium Python
* Special Keys in Selenium Python
* How to handle alert prompts in Selenium Python ?
* Adding and Deleting Cookies in Selenium Python
* How to move back and forward in History using Selenium Python ?
* Special Keys in Selenium Python
* Assertion in Selenium WebDriver using TestNg
* Selenium Python Tricks
* Page Object Model (POM)

Project Examples

* Whatsapp using Python!
* Browser Automation Using Selenium
* Facebook Login using Python
* Automating Happy Birthday post on Facebook using Selenium
* How to access popup login window in selenium using Python
* SMS Bomber using Selenium

Selenium Webdriver is the parent of all methods and classes used in Selenium Python. It is the driving force of Selenium that allows us to perform various operations on multiple elements on a webpage. Driver has various methods and attributes one can use to automate testing in Selenium Python. To check how to use webdriver, visit – Web Driver in Selenium Python . Various methods one can use in selenium Python are –

| Method | Description |
| --- | --- |
| add\_cookie | Adds a cookie to your current session. |
| back | Goes one step backward in the browser history. |
| close | Closes the current window. |
| create\_web\_element | Creates a web element with the specified element\_id. |
| delete\_all\_cookies | Delete all cookies in the scope of the session. |
| delete\_cookie | Deletes a single cookie with the given name. |
| execute\_async\_script | Asynchronously Executes JavaScript in the current window/frame. |
| execute\_script | Synchronously Executes JavaScript in the current window/frame. |
| forward | Goes one step forward in the browser history. |
| fullscreen\_window | Invokes the window manager-specific ‘full screen’ operation |
| get\_cookie | Get a single cookie by name. Returns the cookie if found, None if not. |
| get\_cookies | Returns a set of dictionaries, corresponding to cookies visible in the current session. |
| get\_log | Gets the log for a given log type |
| get\_screenshot\_as\_base64 | Gets the screenshot of the current window as a base64 encoded string which is useful in embedded images in HTML. |
| get\_screenshot\_as\_file | Saves a screenshot of the current window to a PNG image file. |
| get\_screenshot\_as\_png | Gets the screenshot of the current window as a binary data. |
| get\_window\_position | Gets the x, y position of the current window. |
| get\_window\_rect | Gets the x, y coordinates of the window as well as height and width of the current window. |
| get\_window\_size | Gets the width and height of the current window. |
| implicitly\_wait | Sets a sticky timeout to implicitly wait for an element to be found, |
| maximize\_window | Maximizes the current window that webdriver is using |
| minimize\_window | Invokes the window manager-specific ‘minimize’ operation |
| quit | Quits the driver and closes every associated window. |
| refresh | Refreshes the current page. |
| set\_page\_load\_timeout | Set the amount of time to wait for a page load to complete before throwing an error. |
| set\_script\_timeout | Set the amount of time that the script should wait during an execute\_async\_script call before throwing an error. |
| set\_window\_position | Sets the x, y position of the current window. (window.moveTo) |
| set\_window\_rect | Sets the x, y coordinates of the window as well as height and width of the current window. |
| current\_url | Gets the URL of the current page. |
| current\_window\_handle | Returns the handle of the current window. |
| page\_source | Gets the source of the current page. |
| title | Returns the title of the current page. |

Selenium WebElement

An element can be a tag, property, or anything, it is an instance of class selenium.webdriver.remote.webelement.WebElement. After you find an element on screen using selenium, you might want to click it or find sub-elements, etc. Selenium provides methods around this WebElement of Selenium. To checkout how to use element object in selenium, visit – WebElement in Selenium Python. Various methods one can use with an element in Selenium Python are discussed below –

| Element Methods | Description |
| --- | --- |
| is\_selected() | is\_selected method is used to check if element is selected or not. It returns a boolean value True or False. |
| is\_displayed() | is\_displayed method is used to check if element it visible to user or not. It returns a boolean value True or False. |
| is\_enabled() | is\_enabled method is used to check if element is enabled or not. It returns a boolean value True or False. |
| get\_property() | get\_property method is used to get properties of an element, such as getting text\_length property of anchor tag. |
| get\_attribute() | get\_attribute method is used to get attributes of an element, such as getting href attribute of anchor tag. |
| send\_keys() | send\_keys method is used to send text to any field, such as input field of a form or even to anchor tag paragraph, etc. |
| click() | click method is used to click on any element, such as an anchor tag, a link, etc. |
| clear() | clear method is used to clear text of any field, such as input field of a form or even to anchor tag paragraph, etc. |
| screenshot() | screenshot method is used to save a screenshot of current element to a PNG file. |
| submit() | submit method is used to submit a form after you have sent data to a form. |
| value\_of\_css\_property() | value\_of\_css\_property method is used to get value of a css property for a element. |
| location | location method is used to get location of element in renderable canvas. |
| screenshot\_as\_png | screenshot\_as\_png method is used to gets the screenshot of the current element as binary data. |
| parent | parent method is used to get internal reference to the WebDriver instance this element was found from. |
| size | size method is used to get size of current element. |
| tag\_name | tag\_name method is used to get name of tag you are referring to. |
| text | text method is used to get text of current element. |
| rect | rect method is used to get a dictionary with the size and location of the element. |
| screenshot\_as\_base64 | screenshot\_as\_base64 method is used to gets the screenshot of the current element as a base64 encoded string. |