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 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

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.

Method	Description
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

Element Methods	Description
	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.