Here’s a complete list of the **most commonly used attributes and methods** for **Selenium WebDriver** and **WebElement** objects in Python.

**✅ 1. Selenium WebDriver – Common Methods**

The WebDriver object controls the browser.

**🔹 Browser Navigation & Control**

| **Method** | **Description** |
| --- | --- |
| get(url) | Opens a webpage |
| back() | Goes to the previous page |
| forward() | Moves forward in browser history |
| refresh() | Reloads the page |
| quit() | Closes the browser entirely |
| close() | Closes current browser tab |

**🔹 Window and Frame Management**

| **Method** | **Description** |
| --- | --- |
| maximize\_window() | Maximizes the browser window |
| minimize\_window() | Minimizes the window |
| set\_window\_size(w, h) | Sets window size |
| current\_window\_handle | Gets the current window handle |
| window\_handles | List of all open window handles |
| switch\_to.window(handle) | Switch to a new browser window/tab |
| switch\_to.frame(frame\_name\_or\_id) | Switches to iframe |
| switch\_to.default\_content() | Goes back to main content |

**🔹 Alerts and Popups**

| **Method** | **Description** |
| --- | --- |
| switch\_to.alert | Switch to JavaScript alert |
| accept() | Accepts alert |
| dismiss() | Dismisses alert |
| text | Gets alert text |
| send\_keys("text") | Sends text to alert input box |

**🔹 Finding Elements**

| **Method** | **Description** |
| --- | --- |
| find\_element(by, value) | Finds a single element |
| find\_elements(by, value) | Finds all matching elements |

**✅ 2. Selenium WebElement – Common Methods**

The WebElement object refers to HTML elements like buttons, text boxes, etc.

**🔹 Interaction Methods**

| **Method** | **Description** |
| --- | --- |
| click() | Clicks the element |
| send\_keys("text") | Sends text to input field |
| clear() | Clears input field |
| submit() | Submits a form |
| get\_attribute("value") | Gets value of an attribute |
| get\_property("value") | Gets property (JS-style) |

**🔹 State Checking**

| **Property** | **Description** |
| --- | --- |
| is\_displayed() | True if visible on page |
| is\_enabled() | True if enabled for interaction |
| is\_selected() | True if checkbox or radio is selected |
| tag\_name | Returns tag name of element |
| text | Returns visible text of element |

**🔹 Location and Size**

| **Attribute** | **Description** |
| --- | --- |
| location | X, Y coordinates of element |
| size | Height and width |
| rect | Full position and size info |

**✅ Locator Strategies (used in find\_element()):**

from selenium.webdriver.common.by import By

By.ID

By.NAME

By.CLASS\_NAME

By.TAG\_NAME

By.LINK\_TEXT

By.PARTIAL\_LINK\_TEXT

By.XPATH

By.CSS\_SELECTOR

**✅ Example:**

from selenium import webdriver

from selenium.webdriver.common.by import By

driver = webdriver.Chrome()

driver.get("https://example.com")

element = driver.find\_element(By.ID, "username")

element.send\_keys("suresh")

print(element.is\_displayed())

Do you want a printable PDF version of all these Selenium methods and attributes for your notes?

from selenium import webdriver

driver = webdriver.Chrome() # or Firefox(), Edge(), etc.

# Common methods:

driver.get(url) # Navigate to URL

driver.current\_url # Get current URL

driver.title # Get page title

driver.page\_source # Get page HTML source

driver.close() # Close current window

driver.quit() # Quit the entire browser session

driver.back() # Go back in browser history

driver.forward() # Go forward in browser history

driver.refresh() # Refresh page

driver.maximize\_window() # Maximize browser window

driver.minimize\_window() # Minimize browser window

driver.set\_window\_size(x, y) # Set window dimensions

driver.get\_window\_size() # Get window dimensions

driver.get\_window\_position() # Get window position

driver.set\_window\_position(x, y) # Set window position

driver.save\_screenshot(path) # Take screenshot

driver.execute\_script(js) # Execute JavaScriptelement = driver.find\_element(By.CSS\_SELECTOR, "selector")

# Common attributes/methods:

element.text # Get visible text

element.tag\_name # Get HTML tag name

element.size # Get element dimensions

element.location # Get element coordinates

element.rect # Get element rect (size + location)

element.is\_displayed() # Check if visible

element.is\_enabled() # Check if enabled

element.is\_selected() # Check if selected (for checkboxes/radio)

element.get\_attribute(attr) # Get HTML attribute value

element.get\_property(prop) # Get DOM property value

element.send\_keys(text) # Type text

element.clear() # Clear input field

element.click() # Click element

element.submit() # Submit form

element.value\_of\_css\_property(prop) # Get CSS property value

element.screenshot(path) # Take element screenshot from selenium.webdriver.common.by import By

# Common locator strategies:

By.ID # "id"

By.NAME # "name"

By.XPATH # "xpath"

By.LINK\_TEXT # "link text"

By.PARTIAL\_LINK\_TEXT # "partial link text"

By.TAG\_NAME # "tag name"

By.CLASS\_NAME # "class name"

By.CSS\_SELECTOR # "css selector"from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

# Explicit wait example:

wait = WebDriverWait(driver, 10)

element = wait.until(EC.presence\_of\_element\_located((By.ID, "id")))

# Common expected conditions:

EC.title\_is(title)

EC.title\_contains(text)

EC.presence\_of\_element\_located(locator)

EC.visibility\_of\_element\_located(locator)

EC.invisibility\_of\_element\_located(locator)

EC.element\_to\_be\_clickable(locator)

EC.text\_to\_be\_present\_in\_element(locator, text)from selenium.webdriver.common.action\_chains import ActionChains

actions = ActionChains(driver)

actions.click(element)

actions.double\_click(element)

actions.context\_click(element) # right-click

actions.drag\_and\_drop(source, target)

actions.move\_to\_element(element)

actions.send\_keys(keys)

actions.perform() # execute the chainfrom selenium.webdriver.support.ui import Select

select = Select(driver.find\_element(By.TAG\_NAME, "select"))

select.select\_by\_index(index)

select.select\_by\_value(value)

select.select\_by\_visible\_text(text)

select.deselect\_all() # for multi-select check u pdf in case thi item not in please add this

Here’s an enhanced and organized version of the Selenium Python attributes and methods, incorporating missing elements from your reference and ensuring completeness:

---

### \*\*✅ 1. Selenium WebDriver – Complete Methods & Attributes\*\*

Controls the browser instance.

#### \*\*🔹 Browser Navigation & Control\*\*

| \*\*Method/Attribute\*\* | \*\*Description\*\* |

|----------------------------|------------------------------------------|

| `get(url)` | Opens a webpage |

| `back()` | Goes to the previous page |

| `forward()` | Moves forward in browser history |

| `refresh()` | Reloads the page |

| `quit()` | Closes all browser windows |

| `close()` | Closes the current tab |

| `current\_url` | Returns the current URL |

| `title` | Returns the page title |

| `page\_source` | Returns the HTML source |

#### \*\*🔹 Window/Frame Management\*\*

| \*\*Method/Attribute\*\* | \*\*Description\*\* |

|---------------------------------|------------------------------------------|

| `maximize\_window()` | Maximizes the browser window |

| `minimize\_window()` | Minimizes the window |

| `set\_window\_size(width, height)`| Sets window dimensions |

| `get\_window\_size()` | Returns window size |

| `set\_window\_position(x, y)` | Sets window position |

| `get\_window\_position()` | Returns window coordinates |

| `current\_window\_handle` | Gets the current window handle |

| `window\_handles` | List of all open window handles |

| `switch\_to.window(handle)` | Switches to a specific window/tab |

| `switch\_to.frame(id/name)` | Switches to an iframe |

| `switch\_to.default\_content()` | Returns to the main document |

| `switch\_to.parent\_frame()` | Switches to the parent frame |

#### \*\*🔹 Alerts & Popups\*\*

| \*\*Method/Attribute\*\* | \*\*Description\*\* |

|----------------------------|------------------------------------------|

| `switch\_to.alert` | Switches focus to an alert |

| `alert.accept()` | Accepts the alert (OK) |

| `alert.dismiss()` | Dismisses the alert (Cancel) |

| `alert.text` | Gets the alert message |

| `alert.send\_keys("text")` | Enters text in a prompt |

#### \*\*🔹 Element Location\*\*

| \*\*Method\*\* | \*\*Description\*\* |

|--------------------------------|------------------------------------------|

| `find\_element(By, value)` | Finds a single element |

| `find\_elements(By, value)` | Finds all matching elements |

#### \*\*🔹 JavaScript & Screenshots\*\*

| \*\*Method\*\* | \*\*Description\*\* |

|--------------------------------|------------------------------------------|

| `execute\_script(js\_code)` | Executes JavaScript |

| `execute\_async\_script(js\_code)`| Executes async JavaScript |

| `save\_screenshot(path)` | Saves a screenshot of the page |

---

### \*\*✅ 2. WebElement – Complete Methods & Attributes\*\*

Represents HTML elements (e.g., buttons, inputs).

#### \*\*🔹 Interaction Methods\*\*

| \*\*Method/Attribute\*\* | \*\*Description\*\* |

|-------------------------------|------------------------------------------|

| `click()` | Clicks the element |

| `send\_keys("text")` | Types text into an input |

| `clear()` | Clears an input field |

| `submit()` | Submits a form |

| `get\_attribute("attr")` | Returns an attribute value (e.g., `href`)|

| `get\_property("prop")` | Returns a DOM property (e.g., `checked`) |

| `value\_of\_css\_property("prop")`| Returns a CSS property (e.g., `color`) |

| `screenshot("path")` | Takes a screenshot of the element |

#### \*\*🔹 State Checks\*\*

| \*\*Method/Attribute\*\* | \*\*Description\*\* |

|-----------------------|------------------------------------------|

| `is\_displayed()` | Returns `True` if visible |

| `is\_enabled()` | Returns `True` if interactable |

| `is\_selected()` | Returns `True` if selected (checkboxes) |

| `text` | Returns the visible text |

| `tag\_name` | Returns the HTML tag name |

#### \*\*🔹 Location & Dimensions\*\*

| \*\*Attribute\*\* | \*\*Description\*\* |

|------------------|------------------------------------------|

| `location` | `{x, y}` coordinates of the element |

| `size` | `{width, height}` of the element |

| `rect` | Combines `location` and `size` |

---

### \*\*✅ 3. Locator Strategies (`By` Class)\*\*

Used with `find\_element()`/`find\_elements()`.

| \*\*Strategy\*\* | \*\*Example\*\* |

|-------------------------|------------------------------------------|

| `By.ID` | `find\_element(By.ID, "username")` |

| `By.NAME` | `find\_element(By.NAME, "email")` |

| `By.CLASS\_NAME` | `find\_element(By.CLASS\_NAME, "btn")` |

| `By.TAG\_NAME` | `find\_element(By.TAG\_NAME, "div")` |

| `By.LINK\_TEXT` | `find\_element(By.LINK\_TEXT, "Login")` |

| `By.PARTIAL\_LINK\_TEXT` | `find\_element(By.PARTIAL\_LINK\_TEXT, "Log")` |

| `By.XPATH` | `find\_element(By.XPATH, "//button[@id='submit']")` |

| `By.CSS\_SELECTOR` | `find\_element(By.CSS\_SELECTOR, "#submit")` |

---

### \*\*✅ 4. Waits (`WebDriverWait`)\*\*

Handles dynamic elements.

#### \*\*Common Expected Conditions (`EC`)\*\*

| \*\*Condition\*\* | \*\*Description\*\* |

|----------------------------------------|------------------------------------------|

| `EC.title\_is("title")` | Waits for a specific page title |

| `EC.title\_contains("text")` | Waits for title to contain text |

| `EC.presence\_of\_element\_located(locator)` | Waits for an element to exist in DOM |

| `EC.visibility\_of\_element\_located(locator)` | Waits for an element to be visible |

| `EC.invisibility\_of\_element\_located(locator)` | Waits for an element to disappear |

| `EC.element\_to\_be\_clickable(locator)` | Waits for an element to be clickable |

| `EC.text\_to\_be\_present\_in\_element(locator, "text")` | Waits for text to appear in an element |

---

### \*\*✅ 5. Advanced Interactions (`ActionChains`)\*\*

For complex mouse/keyboard actions.

| \*\*Method\*\* | \*\*Description\*\* |

|---------------------------|------------------------------------------|

| `click(element)` | Clicks an element |

| `double\_click(element)` | Double-clicks |

| `context\_click(element)` | Right-clicks |

| `drag\_and\_drop(source, target)` | Drags and drops an element |

| `move\_to\_element(element)`| Hovers over an element |

| `send\_keys("keys")` | Sends keyboard keys |

| `perform()` | Executes the action chain |

---

### \*\*✅ 6. Dropdowns (`Select` Class)\*\*

For `<select>` elements.

| \*\*Method\*\* | \*\*Description\*\* |

|---------------------------|------------------------------------------|

| `select\_by\_index(n)` | Selects option by index (0-based) |

| `select\_by\_value("value")`| Selects option by `value` attribute |

| `select\_by\_visible\_text("text")` | Selects option by visible text |

| `deselect\_all()` | Deselects all options (multi-select) |

| `deselect\_by\_index(n)` | Deselects by index |

| `deselect\_by\_value("value")` | Deselects by `value` |

| `deselect\_by\_visible\_text("text")` | Deselects by visible text |

| `all\_selected\_options` | Returns all selected options |

| `first\_selected\_option` | Returns the first selected option |

---

### \*\*✅ Example Usage\*\*

```python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait, Select

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.common.action\_chains import ActionChains

driver = webdriver.Chrome()

driver.get("https://example.com")

# Find and interact with elements

element = driver.find\_element(By.ID, "username")

element.send\_keys("user123")

# Handle dropdown

select = Select(driver.find\_element(By.TAG\_NAME, "select"))

select.select\_by\_visible\_text("Option 1")

# Wait for an element

wait = WebDriverWait(driver, 10)

wait.until(EC.visibility\_of\_element\_located((By.ID, "submit")))

# Advanced actions

actions = ActionChains(driver)

actions.move\_to\_element(element).click().perform()

driver.quit()

```

---

This list is \*\*comprehensive\*\* and includes all attributes/methods from your reference, plus additional ones like `switch\_to.parent\_frame()`, `execute\_async\_script()`, and `Select` class methods. Let me know if you'd like a PDF version! 🚀