# **SELENIUM WITH PYTHON FULL MATERIAL**

## **CONTENTS**

1. INTRODUCTION OF AUTOMATION TESTING
2. BROWSER LAUNCHING
3. WEB DRIVER METHODS
4. NAVIGATION COMMANDS
5. LOCATORS
6. WEBELEMENT METHODS
7. RADIO BUTTON
8. HANDLING MULTIPLE WEBELEMENTS AND AUTO SUGGESTION
9. CHECK BOX
10. DROP DOWN
11. WEB TABLE
12. JAVASCRIPT EXECUTOR
13. SCROLL UP AND SCROLL DOWN
14. SCREENSHOT
15. MOUSE HOVER ACTION
16. DRAG & DROP AND RIGHT CLICK
17. WINDOW HANDLING
18. FRAME HANDLING
19. ALERTS
20. WAIT
21. FILE UPLOADING USING PYAUROGUI AND SENDKEYS
22. BROKEN LINKS

## **1. INTRODUCTION OF AUTOMATION TESTING**

### **Important Python concepts required for Selenium:**

* Conditions (if, if-else)
* Loops (for, while)
* OOPS (Inheritance, Polymorphism, Encapsulation, Abstraction)
* Method overloading, overriding
* Constructors
* Strings
* Type casting
* Code optimization
* Collections (List, Set, Dictionary)

### **Automation:**

* Performing any task by using a tool or machine is called automation.

### **Advantages:**

1. Save time
2. Faster execution
3. Requires less manual effort
4. Can run continuously
5. Accuracy will be more
6. Multi-task capability
7. Requires less human resources

### **Disadvantages:**

1. Initial investment will be more
2. Requires constant maintenance
3. Requires additional skill sets

### **Automation Testing:**

* Testing an application by using any automation tools is called automation testing.

### **Automation Tool:**

* Software used to automate applications
* Ex: Selenium, Appium, Robot Framework, etc.

## **2. BROWSER LAUNCHING**

### **Selenium with Python Setup:**

bash

pip install selenium

### **Basic Browser Launching:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

*# Chrome Browser*

driver = webdriver.Chrome()

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

time.sleep(2)

driver.quit()

*# Firefox Browser*

driver = webdriver.Firefox()

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

time.sleep(2)

driver.quit()

*# Edge Browser*

driver = webdriver.Edge()

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

time.sleep(2)

driver.quit()

### **Difference between close() and quit():**

* close(): Closes the current browser window
* quit(): Closes all browser windows and ends WebDriver session

python

*# Example*

driver = webdriver.Chrome()

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

driver.get("https://www.facebook.com")

driver.close() *# Closes only Facebook tab*

*# driver.quit() # Would close entire browser*

## **3. WEB DRIVER METHODS**

### **Common WebDriver Methods:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

driver = webdriver.Chrome()

*# 1. get() - Navigate to URL*

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

*# 2. get\_title() - Get page title*

title = driver.title

print(f"Page Title: {title}")

*# 3. current\_url - Get current URL*

current\_url = driver.current\_url

print(f"Current URL: {current\_url}")

*# 4. page\_source - Get page source*

page\_source = driver.page\_source

print(f"Page Source Length: {len(page\_source)}")

*# 5. maximize\_window() - Maximize browser window*

driver.maximize\_window()

*# 6. set\_window\_size() - Set window size*

driver.set\_window\_size(1024, 768)

*# 7. set\_window\_position() - Set window position*

driver.set\_window\_position(100, 100)

*# 8. back() - Navigate back*

driver.back()

*# 9. forward() - Navigate forward*

driver.forward()

*# 10. refresh() - Refresh page*

driver.refresh()

*# 11. close() - Close current window*

driver.close()

*# 12. quit() - Close all windows*

driver.quit()

### **Complete Example:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def webdriver\_methods\_demo():

driver = webdriver.Chrome()

try:

*# Navigate to Google*

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

*# Get page title*

print(f"Title: {driver.title}")

*# Get current URL*

print(f"URL: {driver.current\_url}")

*# Maximize window*

driver.maximize\_window()

time.sleep(2)

*# Set window size*

driver.set\_window\_size(800, 600)

time.sleep(2)

*# Navigate to another site*

driver.get("https://www.python.org")

time.sleep(2)

*# Go back to Google*

driver.back()

time.sleep(2)

*# Go forward to Python*

driver.forward()

time.sleep(2)

*# Refresh page*

driver.refresh()

finally:

driver.quit()

webdriver\_methods\_demo()

## **4. NAVIGATION COMMANDS**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def navigation\_commands():

driver = webdriver.Chrome()

try:

driver.maximize\_window()

*# Using get() method*

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

print(f"Page 1: {driver.title}")

time.sleep(2)

*# Using navigate().to() method*

driver.get("https://www.facebook.com")

print(f"Page 2: {driver.title}")

time.sleep(2)

*# Navigate back*

driver.back()

print(f"After back: {driver.title}")

time.sleep(2)

*# Navigate forward*

driver.forward()

print(f"After forward: {driver.title}")

time.sleep(2)

*# Refresh page*

driver.refresh()

print("Page refreshed")

finally:

driver.quit()

navigation\_commands()

## **5. LOCATORS**

### **Types of Locators in Selenium Python:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def locators\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com")

*# 1. ID Locator*

username = driver.find\_element(By.ID, "user-name")

username.send\_keys("standard\_user")

*# 2. NAME Locator*

password = driver.find\_element(By.NAME, "password")

password.send\_keys("secret\_sauce")

*# 3. CLASS\_NAME Locator*

login\_btn = driver.find\_element(By.CLASS\_NAME, "submit-button")

login\_btn.click()

time.sleep(2)

*# 4. TAG\_NAME Locator*

links = driver.find\_elements(By.TAG\_NAME, "a")

print(f"Number of links: {len(links)}")

*# 5. LINK\_TEXT Locator*

driver.get("https://the-internet.herokuapp.com")

ab\_testing\_link = driver.find\_element(By.LINK\_TEXT, "A/B Testing")

ab\_testing\_link.click()

time.sleep(2)

*# 6. PARTIAL\_LINK\_TEXT Locator*

driver.back()

test\_link = driver.find\_element(By.PARTIAL\_LINK\_TEXT, "A/B")

test\_link.click()

time.sleep(2)

*# 7. CSS\_SELECTOR Locator*

driver.get("https://www.saucedemo.com")

username\_css = driver.find\_element(By.CSS\_SELECTOR, "#user-name")

username\_css.send\_keys("standard\_user")

*# 8. XPATH Locator*

password\_xpath = driver.find\_element(By.XPATH, "//input[@id='password']")

password\_xpath.send\_keys("secret\_sauce")

finally:

driver.quit()

locators\_demo()

### **XPATH Examples:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

def xpath\_examples():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com")

*# Absolute XPath*

*# username = driver.find\_element(By.XPATH, "/html/body/div/div/div[2]/div[1]/div/div/form/div[1]/input")*

*# Relative XPath by attribute*

username = driver.find\_element(By.XPATH, "//input[@id='user-name']")

username.send\_keys("standard\_user")

*# XPath with multiple attributes*

password = driver.find\_element(By.XPATH, "//input[@id='password' and @name='password']")

password.send\_keys("secret\_sauce")

*# XPath by text*

login\_btn = driver.find\_element(By.XPATH, "//input[@value='Login']")

login\_btn.click()

*# XPath contains*

menu\_btn = driver.find\_element(By.XPATH, "//button[contains(text(), 'Open Menu')]")

*# XPath starts-with*

*# menu\_btn = driver.find\_element(By.XPATH, "//button[starts-with(@id, 'react')]")*

finally:

driver.quit()

xpath\_examples()

## **6. WEBELEMENT METHODS**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def webelement\_methods():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com")

*# Find element*

username = driver.find\_element(By.ID, "user-name")

*# 1. send\_keys() - Enter text*

username.send\_keys("standard\_user")

*# 2. clear() - Clear text*

username.clear()

username.send\_keys("problem\_user")

*# 3. click() - Click element*

login\_btn = driver.find\_element(By.ID, "login-button")

login\_btn.click()

time.sleep(2)

*# 4. get\_attribute() - Get attribute value*

menu\_btn = driver.find\_element(By.ID, "react-burger-menu-btn")

class\_name = menu\_btn.get\_attribute("class")

print(f"Class attribute: {class\_name}")

*# 5. text - Get text content*

title = driver.find\_element(By.CLASS\_NAME, "title")

print(f"Title text: {title.text}")

*# 6. is\_displayed() - Check if element is displayed*

is\_displayed = menu\_btn.is\_displayed()

print(f"Is displayed: {is\_displayed}")

*# 7. is\_enabled() - Check if element is enabled*

is\_enabled = menu\_btn.is\_enabled()

print(f"Is enabled: {is\_enabled}")

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

menu\_btn.click()

time.sleep(1)

logout\_link = driver.find\_element(By.ID, "logout\_sidebar\_link")

is\_selected = logout\_link.is\_selected()

print(f"Is selected: {is\_selected}")

*# 9. location - Get element location*

location = menu\_btn.location

print(f"Element location: {location}")

*# 10. size - Get element size*

size = menu\_btn.size

print(f"Element size: {size}")

finally:

driver.quit()

webelement\_methods()

## **7. RADIO BUTTON**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def radio\_button\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/radio-button")

*# Find radio buttons*

yes\_radio = driver.find\_element(By.ID, "yesRadio")

impressive\_radio = driver.find\_element(By.ID, "impressiveRadio")

no\_radio = driver.find\_element(By.ID, "noRadio")

*# Click radio button using JavaScript (if direct click doesn't work)*

driver.execute\_script("arguments[0].click();", yes\_radio)

time.sleep(2)

*# Check if selected*

is\_selected = driver.execute\_script("return arguments[0].checked;", yes\_radio)

print(f"Yes radio selected: {is\_selected}")

*# Click another radio button*

driver.execute\_script("arguments[0].click();", impressive\_radio)

time.sleep(2)

is\_selected = driver.execute\_script("return arguments[0].checked;", impressive\_radio)

print(f"Impressive radio selected: {is\_selected}")

finally:

driver.quit()

radio\_button\_demo()

## **8. HANDLING MULTIPLE WEBELEMENTS AND AUTO SUGGESTION**

### **Handling Multiple Elements:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def multiple\_elements():

driver = webdriver.Chrome()

try:

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

*# Find all links on the page*

links = driver.find\_elements(By.TAG\_NAME, "a")

print(f"Total links found: {len(links)}")

*# Print first 10 links*

for i, link in enumerate(links[:10]):

href = link.get\_attribute("href")

text = link.text

if text: *# Only print links with visible text*

print(f"{i+1}. {text} -> {href}")

finally:

driver.quit()

multiple\_elements()

### **Auto Suggestion Handling:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

def auto\_suggestion():

driver = webdriver.Chrome()

try:

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

*# Find search box and enter text*

search\_box = driver.find\_element(By.NAME, "q")

search\_box.send\_keys("selenium")

time.sleep(2) *# Wait for suggestions to appear*

*# Get all suggestions*

suggestions = driver.find\_elements(By.CSS\_SELECTOR, "ul.G43f7e li")

print(f"Number of suggestions: {len(suggestions)}")

*# Print all suggestions*

for i, suggestion in enumerate(suggestions):

text = suggestion.text

if text:

print(f"{i+1}. {text}")

*# Click on the first suggestion*

if suggestions:

suggestions[0].click()

time.sleep(2)

finally:

driver.quit()

auto\_suggestion()

## **9. CHECK BOX**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def checkbox\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/checkbox")

*# Expand home checkbox*

home\_checkbox = driver.find\_element(By.CSS\_SELECTOR, ".rct-checkbox")

home\_checkbox.click()

time.sleep(2)

*# Find all checkboxes*

checkboxes = driver.find\_elements(By.CSS\_SELECTOR, ".rct-checkbox")

print(f"Total checkboxes: {len(checkboxes)}")

*# Click specific checkboxes*

for i, checkbox in enumerate(checkboxes[1:4]): *# Skip first one (home)*

checkbox.click()

time.sleep(1)

*# Verify selection*

selected\_checkboxes = driver.find\_elements(By.CSS\_SELECTOR, ".rct-icon-check")

print(f"Selected checkboxes: {len(selected\_checkboxes)}")

finally:

driver.quit()

checkbox\_demo()

## **10. DROP DOWN**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.select import Select

import time

def dropdown\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/select-menu")

*# Old Style Select (Dropdown)*

old\_select = Select(driver.find\_element(By.ID, "oldSelectMenu"))

*# Select by visible text*

old\_select.select\_by\_visible\_text("Green")

time.sleep(1)

*# Select by value*

old\_select.select\_by\_value("4")

time.sleep(1)

*# Select by index*

old\_select.select\_by\_index(1)

time.sleep(1)

*# Get all options*

options = old\_select.options

for option in options:

print(f"Option: {option.text}")

*# Get selected option*

selected\_option = old\_select.first\_selected\_option

print(f"Selected: {selected\_option.text}")

finally:

driver.quit()

dropdown\_demo()

### **Multi-Select Dropdown:**

python

def multi\_select\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/select-menu")

*# Multi-select dropdown*

multi\_select = Select(driver.find\_element(By.ID, "cars"))

*# Select multiple options*

multi\_select.select\_by\_visible\_text("Volvo")

multi\_select.select\_by\_visible\_text("Audi")

time.sleep(2)

*# Get all selected options*

selected\_options = multi\_select.all\_selected\_options

for option in selected\_options:

print(f"Selected: {option.text}")

*# Deselect an option*

multi\_select.deselect\_by\_visible\_text("Volvo")

time.sleep(2)

*# Deselect all*

multi\_select.deselect\_all()

finally:

driver.quit()

multi\_select\_demo()

## **11. WEB TABLE**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def web\_table\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/webtables")

*# Get all rows in the table*

rows = driver.find\_elements(By.CSS\_SELECTOR, ".rt-tr-group")

print(f"Total rows: {len(rows)}")

*# Extract data from each row*

for i, row in enumerate(rows):

cells = row.find\_elements(By.CSS\_SELECTOR, ".rt-td")

if len(cells) >= 6: *# Ensure we have enough cells*

row\_data = [cell.text for cell in cells[:6]] *# First 6 columns*

if any(row\_data): *# Skip empty rows*

print(f"Row {i+1}: {row\_data}")

*# Find specific data*

for row in rows:

cells = row.find\_elements(By.CSS\_SELECTOR, ".rt-td")

if len(cells) >= 4 and cells[0].text == "Cierra":

print(f"Found Cierra: {cells[3].text}") *# Department*

break

finally:

driver.quit()

web\_table\_demo()

## **12. JAVASCRIPT EXECUTOR**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def javascript\_executor\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com")

*# Login first*

driver.find\_element(By.ID, "user-name").send\_keys("standard\_user")

driver.find\_element(By.ID, "password").send\_keys("secret\_sauce")

driver.find\_element(By.ID, "login-button").click()

time.sleep(2)

*# JavaScript Executor examples*

*# 1. Scroll to bottom*

driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")

time.sleep(2)

*# 2. Scroll to top*

driver.execute\_script("window.scrollTo(0, 0);")

time.sleep(2)

*# 3. Scroll to specific element*

footer = driver.find\_element(By.CLASS\_NAME, "footer")

driver.execute\_script("arguments[0].scrollIntoView();", footer)

time.sleep(2)

*# 4. Highlight element*

product = driver.find\_element(By.CLASS\_NAME, "inventory\_item\_name")

driver.execute\_script("arguments[0].style.border='3px solid red'", product)

time.sleep(2)

*# 5. Click using JavaScript*

menu\_btn = driver.find\_element(By.ID, "react-burger-menu-btn")

driver.execute\_script("arguments[0].click();", menu\_btn)

time.sleep(2)

*# 6. Get page title using JavaScript*

title = driver.execute\_script("return document.title;")

print(f"Page title via JS: {title}")

*# 7. Get page URL using JavaScript*

url = driver.execute\_script("return document.URL;")

print(f"Page URL via JS: {url}")

finally:

driver.quit()

javascript\_executor\_demo()

## **13. SCROLL UP AND SCROLL DOWN**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

def scroll\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com/inventory.html")

*# Login first*

driver.find\_element(By.ID, "user-name").send\_keys("standard\_user")

driver.find\_element(By.ID, "password").send\_keys("secret\_sauce")

driver.find\_element(By.ID, "login-button").click()

time.sleep(2)

print("Scrolling methods demonstration:")

*# Method 1: Using JavaScript - Scroll by pixel*

print("1. Scrolling down by 500 pixels")

driver.execute\_script("window.scrollBy(0, 500);")

time.sleep(2)

print("2. Scrolling up by 300 pixels")

driver.execute\_script("window.scrollBy(0, -300);")

time.sleep(2)

*# Method 2: Using JavaScript - Scroll to bottom*

print("3. Scrolling to bottom")

driver.execute\_script("window.scrollTo(0, document.body.scrollHeight);")

time.sleep(2)

*# Method 3: Using JavaScript - Scroll to top*

print("4. Scrolling to top")

driver.execute\_script("window.scrollTo(0, 0);")

time.sleep(2)

*# Method 4: Using JavaScript - Scroll to specific element*

print("5. Scrolling to specific element")

footer = driver.find\_element(By.CLASS\_NAME, "footer")

driver.execute\_script("arguments[0].scrollIntoView();", footer)

time.sleep(2)

*# Method 5: Using PAGE\_DOWN key*

print("6. Scrolling using PAGE\_DOWN key")

driver.find\_element(By.TAG\_NAME, "body").send\_keys(Keys.PAGE\_DOWN)

time.sleep(1)

driver.find\_element(By.TAG\_NAME, "body").send\_keys(Keys.PAGE\_UP)

time.sleep(1)

finally:

driver.quit()

scroll\_demo()

## **14. SCREENSHOT**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

import os

def screenshot\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com")

*# Create screenshots directory if it doesn't exist*

if not os.path.exists("screenshots"):

os.makedirs("screenshots")

*# 1. Take full page screenshot*

driver.save\_screenshot("screenshots/full\_page.png")

print("Full page screenshot saved")

*# Login*

driver.find\_element(By.ID, "user-name").send\_keys("standard\_user")

driver.find\_element(By.ID, "password").send\_keys("secret\_sauce")

driver.find\_element(By.ID, "login-button").click()

time.sleep(2)

*# 2. Take screenshot after login*

driver.save\_screenshot("screenshots/after\_login.png")

print("After login screenshot saved")

*# 3. Take screenshot of specific element*

product = driver.find\_element(By.CLASS\_NAME, "inventory\_item")

product.screenshot("screenshots/specific\_element.png")

print("Specific element screenshot saved")

finally:

driver.quit()

screenshot\_demo()

## **15. MOUSE HOVER ACTION**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

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

import time

def mouse\_hover\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/menu")

actions = ActionChains(driver)

*# Hover over main menu item*

main\_item = driver.find\_element(By.LINK\_TEXT, "Main Item 2")

actions.move\_to\_element(main\_item).perform()

time.sleep(2)

*# Hover over sub menu item (if visible)*

sub\_items = driver.find\_elements(By.CSS\_SELECTOR, ".nav-item")

for item in sub\_items:

if "SUB SUB LIST" in item.text.upper():

actions.move\_to\_element(item).perform()

time.sleep(2)

break

finally:

driver.quit()

mouse\_hover\_demo()

## **16. DRAG & DROP AND RIGHT CLICK**

### **Drag and Drop:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

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

import time

def drag\_drop\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/droppable")

actions = ActionChains(driver)

*# Find source and target elements*

source = driver.find\_element(By.ID, "draggable")

target = driver.find\_element(By.ID, "droppable")

*# Perform drag and drop*

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

time.sleep(2)

*# Verify drop*

target\_text = target.text

print(f"Target text after drop: {target\_text}")

finally:

driver.quit()

drag\_drop\_demo()

### **Right Click (Context Click):**

python

def right\_click\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/buttons")

actions = ActionChains(driver)

*# Find the button for right click*

right\_click\_btn = driver.find\_element(By.ID, "rightClickBtn")

*# Perform right click*

actions.context\_click(right\_click\_btn).perform()

time.sleep(2)

*# Verify right click message*

message = driver.find\_element(By.ID, "rightClickMessage")

print(f"Right click message: {message.text}")

finally:

driver.quit()

right\_click\_demo()

### **Double Click:**

python

def double\_click\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/buttons")

actions = ActionChains(driver)

*# Find the button for double click*

double\_click\_btn = driver.find\_element(By.ID, "doubleClickBtn")

*# Perform double click*

actions.double\_click(double\_click\_btn).perform()

time.sleep(2)

*# Verify double click message*

message = driver.find\_element(By.ID, "doubleClickMessage")

print(f"Double click message: {message.text}")

finally:

driver.quit()

double\_click\_demo()

## **17. WINDOW HANDLING**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def window\_handling\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/browser-windows")

*# Store the parent window handle*

parent\_window = driver.current\_window\_handle

print(f"Parent window: {parent\_window}")

*# Click button to open new window*

button = driver.find\_element(By.ID, "windowButton")

new\_window\_btn.click()

driver.execute\_script("arguments[0].click();", button)

time.sleep(2)

*# Get all window handles*

all\_windows = driver.window\_handles

print(f"Total windows: {len(all\_windows)}")

*# Switch to new window*

for window in all\_windows:

if window != parent\_window:

driver.switch\_to.window(window)

print(f"Switched to: {driver.current\_window\_handle}")

print(f"New window title: {driver.title}")

print(f"New window URL: {driver.current\_url}")

*# Do something in new window*

text\_element = driver.find\_element(By.TAG\_NAME, "body")

print(f"New window content: {text\_element.text}")

*# Close new window*

driver.close()

break

*# Switch back to parent window*

driver.switch\_to.window(parent\_window)

print(f"Back to parent: {driver.title}")

finally:

driver.quit()

window\_handling\_demo()

### **Multiple Windows Handling:**

python

def multiple\_windows\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/browser-windows")

parent\_window = driver.current\_window\_handle

*# Open multiple windows*

driver.find\_element(By.ID, "windowButton").click()

time.sleep(1)

button =driver.find\_element(By.ID, "messageWindowButton")

driver.execute\_script("arguments[0].click();", button)

time.sleep(1)

*# Get all windows*

all\_windows = driver.window\_handles

print(f"Total windows opened: {len(all\_windows)}")

*# Switch to and close each window except parent*

for window in all\_windows:

if window != parent\_window:

driver.switch\_to.window(window)

print(f"Window: {driver.current\_window\_handle}")

print(f"Title: {driver.title}")

driver.close()

time.sleep(1)

*# Switch back to parent*

driver.switch\_to.window(parent\_window)

print(f"Final window: {driver.title}")

finally:

driver.quit()

multiple\_windows\_demo()

## **18. FRAME HANDLING**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def frame\_handling\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/frames")

*# Switch to frame by index*

driver.switch\_to.frame(0)

*# Switch to frame by ID*

driver.switch\_to.frame(“frame1”)

print("Switched to frame 0")

*# Get text from frame*

frame\_text = driver.find\_element(By.ID, "sampleHeading").text

print(f"Frame text: {frame\_text}")

*# Switch back to main content*

driver.switch\_to.default\_content()

print("Switched back to main content")

*# Switch to frame by ID/Name (if available)*

*# driver.switch\_to.frame("frame1")*

*# Switch to frame by WebElement*

frame\_element = driver.find\_element(By.ID, "frame1")

driver.switch\_to.frame(frame\_element)

print("Switched to frame by element")

frame\_text = driver.find\_element(By.ID, "sampleHeading").text

print(f"Frame text: {frame\_text}")

*# Always switch back to default content*

driver.switch\_to.default\_content()

finally:

driver.quit()

frame\_handling\_demo()

## **19. ALERTS**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def alerts\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/alerts")

*# 1. Simple Alert*

simple\_alert\_btn = driver.find\_element(By.ID, "alertButton")

simple\_alert\_btn.click()

time.sleep(1)

*# Switch to alert and accept*

alert = driver.switch\_to.alert

print(f"Alert text: {alert.text}")

alert.accept()

print("Simple alert accepted")

time.sleep(1)

*# 2. Confirmation Alert*

confirm\_alert\_btn = driver.find\_element(By.ID, "confirmButton")

confirm\_alert\_btn.click()

time.sleep(1)

alert = driver.switch\_to.alert

print(f"Confirm alert text: {alert.text}")

alert.dismiss() *# Try accept() for OK*

print("Confirmation alert dismissed")

time.sleep(1)

*# 3. Prompt Alert*

prompt\_alert\_btn = driver.find\_element(By.ID, "promtButton")

prompt\_alert\_btn.click()

time.sleep(1)

alert = driver.switch\_to.alert

alert.send\_keys("Selenium Python")

print("Text entered in prompt")

alert.accept()

print("Prompt alert accepted")

finally:

driver.quit()

alerts\_demo()

## **20. WAIT Synchronization Vs Simultaneous**

### **Implicit Wait:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

def implicit\_wait\_demo():

driver = webdriver.Chrome()

*# Set implicit wait (wait up to 10 seconds for elements)*

driver.implicitly\_wait(10)

try:

driver.get("https://www.saucedemo.com")

*# These will wait up to 10 seconds if elements are not immediately found*

username = driver.find\_element(By.ID, "user-name")

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

login\_btn = driver.find\_element(By.ID, "login-button")

username.send\_keys("standard\_user")

password.send\_keys("secret\_sauce")

login\_btn.click()

finally:

driver.quit()

implicit\_wait\_demo()

### **Explicit Wait:**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

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

import time

def explicit\_wait\_demo():

driver = webdriver.Chrome()

try:

driver.get("https://www.saucedemo.com")

*# Login first*

driver.find\_element(By.ID, "user-name").send\_keys("standard\_user")

driver.find\_element(By.ID, "password").send\_keys("secret\_sauce")

driver.find\_element(By.ID, "login-button").click()

*# Use explicit wait for specific condition*

wait = WebDriverWait(driver, 10)

*# Wait for products to be visible*

products = wait.until(

EC.visibility\_of\_element\_located((By.CLASS\_NAME, "inventory\_list"))

)

print("Products are visible")

*# Wait for element to be clickable*

product\_item = wait.until(

EC.element\_to\_be\_clickable((By.CLASS\_NAME, "inventory\_item\_name"))

)

product\_item.click()

print("Product clicked")

*# Wait for URL to contain specific text*

wait.until(EC.url\_contains("inventory-item"))

print("Navigated to product details")

finally:

driver.quit()

explicit\_wait\_demo()

### **Common Expected Conditions:**

* visibility\_of\_element\_located()
* element\_to\_be\_clickable()
* presence\_of\_element\_located()
* text\_to\_be\_present\_in\_element()
* title\_contains()
* alert\_is\_present()

## **21. FILE UPLOADING USING PYAUROGUI AND SENDKEYS**

### **File Upload using send\_keys():**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import time

import os

def file\_upload\_sendkeys():

driver = webdriver.Chrome()

try:

driver.get("https://the-internet.herokuapp.com/upload")

*# Create a test file to upload*

with open("test\_upload.txt", "w") as f:

f.write("This is a test file for upload")

*# Upload file using send\_keys*

file\_input = driver.find\_element(By.ID, "file-upload")

file\_input.send\_keys(os.path.abspath("test\_upload.txt"))

*# Click upload button*

upload\_btn = driver.find\_element(By.ID, "file-submit")

upload\_btn.click()

time.sleep(2)

*# Verify upload success*

success\_message = driver.find\_element(By.TAG\_NAME, "h3").text

print(f"Upload result: {success\_message}")

*# Clean up*

if os.path.exists("test\_upload.txt"):

os.remove("test\_upload.txt")

finally:

driver.quit()

file\_upload\_sendkeys()

### **File Upload using pyautogui (Alternative):**

python

import pyautogui

import time

def file\_upload\_pyautogui():

driver = webdriver.Chrome()

try:

driver.get("https://the-internet.herokuapp.com/upload")

*# Create test file*

with open("test\_upload.txt", "w") as f:

f.write("This is a test file for upload")

*# Click file input to open file dialog*

file\_input = driver.find\_element(By.ID, "file-upload")

file\_input.click()

time.sleep(2)

*# Use pyautogui to handle file dialog*

pyautogui.write(os.path.abspath("test\_upload.txt"))

pyautogui.press('enter')

time.sleep(2)

*# Click upload button*

upload\_btn = driver.find\_element(By.ID, "file-submit")

upload\_btn.click()

time.sleep(2)

*# Clean up*

if os.path.exists("test\_upload.txt"):

os.remove("test\_upload.txt")

finally:

driver.quit()

*# Note: Install pyautogui first: pip install pyautogui*

*# file\_upload\_pyautogui()*

## **22. BROKEN LINKS**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

import requests

from urllib.parse import urljoin, urlparse

import time

def broken\_links\_check():

driver = webdriver.Chrome()

try:

driver.get("https://demoqa.com/broken")

*# Find all links on the page*

links = driver.find\_elements(By.TAG\_NAME, "a")

print(f"Total links found: {len(links)}")

broken\_links = []

valid\_links = []

for link in links:

href = link.get\_attribute("href")

text = link.text.strip()

if href:

try:

*# Make HEAD request to check link*

response = requests.head(href, timeout=5)

if response.status\_code >= 400:

broken\_links.append((text, href, response.status\_code))

print(f"BROKEN: {text} -> {href} (Status: {response.status\_code})")

else:

valid\_links.append((text, href, response.status\_code))

print(f"VALID: {text} -> {href} (Status: {response.status\_code})")

except requests.exceptions.RequestException as e:

broken\_links.append((text, href, f"Error: {str(e)}"))

print(f"BROKEN: {text} -> {href} (Error: {str(e)})")

print(f"\nSummary:")

print(f"Total links: {len(links)}")

print(f"Valid links: {len(valid\_links)}")

print(f"Broken links: {len(broken\_links)}")

finally:

driver.quit()

broken\_links\_check()

## **COMPLETE PRACTICAL EXAMPLE**

python

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support.ui import WebDriverWait

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

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

from selenium.webdriver.support.select import Select

import time

import os

def complete\_automation\_demo():

"""Complete automation demo covering multiple concepts"""

driver = webdriver.Chrome()

wait = WebDriverWait(driver, 10)

actions = ActionChains(driver)

try:

*# 1. Navigation and Basic Operations*

print("1. Navigating to website...")

driver.get("https://www.saucedemo.com")

driver.maximize\_window()

*# 2. Login with explicit wait*

print("2. Performing login...")

username = wait.until(EC.element\_to\_be\_clickable((By.ID, "user-name")))

username.send\_keys("standard\_user")

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

password.send\_keys("secret\_sauce")

login\_btn = driver.find\_element(By.ID, "login-button")

login\_btn.click()

*# 3. Verify login success*

print("3. Verifying login success...")

wait.until(EC.url\_contains("inventory"))

products\_title = driver.find\_element(By.CLASS\_NAME, "title")

assert "Products" in products\_title.text

print("Login successful!")

*# 4. Handle dropdown (sorting)*

print("4. Handling dropdown...")

sort\_dropdown = Select(driver.find\_element(By.CLASS\_NAME, "product\_sort\_container"))

sort\_dropdown.select\_by\_visible\_text("Price (low to high)")

time.sleep(2)

*# 5. Add products to cart*

print("5. Adding products to cart...")

add\_to\_cart\_buttons = driver.find\_elements(By.CLASS\_NAME, "btn\_inventory")

for i, button in enumerate(add\_to\_cart\_buttons[:2]): *# Add first 2 products*

button.click()

print(f"Added product {i+1} to cart")

*# 6. Go to cart*

print("6. Going to cart...")

cart\_icon = driver.find\_element(By.CLASS\_NAME, "shopping\_cart\_link")

cart\_icon.click()

*# 7. Checkout*

print("7. Proceeding to checkout...")

checkout\_btn = driver.find\_element(By.ID, "checkout")

checkout\_btn.click()

*# 8. Fill checkout information*

print("8. Filling checkout information...")

driver.find\_element(By.ID, "first-name").send\_keys("John")

driver.find\_element(By.ID, "last-name").send\_keys("Doe")

driver.find\_element(By.ID, "postal-code").send\_keys("12345")

continue\_btn = driver.find\_element(By.ID, "continue")

continue\_btn.click()

*# 9. Finish checkout*

print("9. Finishing checkout...")

finish\_btn = driver.find\_element(By.ID, "finish")

finish\_btn.click()

*# 10. Verify order completion*

print("10. Verifying order completion...")

complete\_header = driver.find\_element(By.CLASS\_NAME, "complete-header")

assert "Thank you for your order!" in complete\_header.text

print("Order completed successfully!")

*# 11. Take screenshot*

print("11. Taking screenshot...")

if not os.path.exists("screenshots"):

os.makedirs("screenshots")

driver.save\_screenshot("screenshots/order\_complete.png")

*# 12. Back to products*

back\_btn = driver.find\_element(By.ID, "back-to-products")

back\_btn.click()

print("Automation completed successfully!")

except Exception as e:

print(f"Error occurred: {e}")

*# Take screenshot on error*

driver.save\_screenshot("screenshots/error.png")

finally:

driver.quit()

*# Run the complete demo*

complete\_automation\_demo()

## **KEY DIFFERENCES: JAVA vs PYTHON**

|  |  |
| --- | --- |
| Java Selenium | Python Selenium |
| driver.findElement(By.id("element")) | driver.find\_element(By.ID, "element") |
| driver.findElements() | driver.find\_elements() |
| element.sendKeys("text") | element.send\_keys("text") |
| element.isDisplayed() | element.is\_displayed() |
| element.isEnabled() | element.is\_enabled() |
| element.isSelected() | element.is\_selected() |
| element.getText() | element.text |
| element.getAttribute("attr") | element.get\_attribute("attr") |
| System.setProperty() | Not needed in Selenium 4+ |
| driver.manage().window().maximize() | driver.maximize\_window() |