**Week 1: Introduction to Web Automation and Selenium**

* Introduction to web automation
* Introduction to Selenium
* Setting up a Selenium environment in Jupyter Notebook
* Selenium's basic architecture and terminology
* Understanding web elements and web page structure
* Locating and interacting with elements using Selenium
* Handling web page navigation and URL handling using Selenium

**Week 2: Advanced Web Automation Techniques**

* Using Selenium to interact with advanced web elements (such as iframes)
* Handling alerts and popups using Selenium
* Executing JavaScript using Selenium
* Scraping data from web pages using Selenium
* Implementing implicit and explicit waits in Selenium
* Understanding and handling exceptions in Selenium

**Week 3: Web Automation Best Practices**

* Implementing good coding practices in Selenium
* Writing modular and reusable code using Selenium
* Handling multiple windows and tabs using Selenium
* Using Selenium with other Python libraries (such as Pandas and NumPy)
* Working with Selenium's advanced features (such as headless browsers)
* Tips for debugging and troubleshooting Selenium scripts

**Week 4: Advanced Topics in Web Automation with Selenium**

* Creating test cases using Selenium and Pytest
* Implementing page object models in Selenium
* Testing web applications using Selenium
* Implementing web automation using Selenium for various use cases (such as filling forms and web scraping)

### Script 1: Installing Required Packages

This script installs the necessary packages to run the Selenium tests:

pythonCopy code

!pip install selenium webdriver\_manager

### Script 2: Setting up the Selenium Web Driver

This script sets up the Selenium web driver for Chrome:

pythonCopy code

from selenium import webdriver from webdriver\_manager.chrome import ChromeDriverManager driver = webdriver.Chrome(ChromeDriverManager().install())

### Script 3: Navigating to a Webpage

This script navigates to a webpage using the Selenium web driver:

pythonCopy code

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

### Script 4: Locating Web Elements

This script uses Selenium to locate web elements on a page:

pythonCopy code

# Locate an element by ID element = driver.find\_element\_by\_id("my\_id") # Locate an element by name element = driver.find\_element\_by\_name("my\_name") # Locate an element by class name element = driver.find\_element\_by\_class\_name("my\_class") # Locate an element by tag name element = driver.find\_element\_by\_tag\_name("my\_tag") # Locate an element by link text element = driver.find\_element\_by\_link\_text("my\_link\_text") # Locate an element by partial link text element = driver.find\_element\_by\_partial\_link\_text("my\_partial\_link\_text") # Locate an element by CSS selector element = driver.find\_element\_by\_css\_selector("my\_selector") # Locate an element by XPath element = driver.find\_element\_by\_xpath("my\_xpath")

### Script 5: Interacting with Web Elements

This script interacts with web elements using Selenium:

pythonCopy code

# Click a button button = driver.find\_element\_by\_id("my\_button") button.click() # Fill in a form field field = driver.find\_element\_by\_id("my\_field") field.send\_keys("my\_text") # Select an option from a dropdown dropdown = driver.find\_element\_by\_id("my\_dropdown") option = dropdown.find\_element\_by\_xpath("//option[text()='my\_option']") option.click() # Submit a form form = driver.find\_element\_by\_id("my\_form") form.submit()

### Script 6: Waiting for Page Elements to Load

This script waits for page elements to load using Selenium:

pythonCopy code

from selenium.webdriver.common.by import By from selenium.webdriver.support.ui import WebDriverWait from selenium.webdriver.support import expected\_conditions as EC # Wait for an element to appear element = WebDriverWait(driver, 10).until( EC.presence\_of\_element\_located((By.ID, "my\_id")) ) # Wait for an element to be clickable element = WebDriverWait(driver, 10).until( EC.element\_to\_be\_clickable((By.ID, "my\_id")) ) # Wait for an element to have a certain text element = WebDriverWait(driver, 10).until( EC.text\_to\_be\_present\_in\_element((By.ID, "my\_id"), "my\_text") )