

Test Automation  
Foundation Level Hands-On Exercises  
for Python





## Test Automation – Foundation Level Hands-On Exercises

### About

1. These hands-on exercises will enable you to achieve Foundation level capability in Python.
2. As guidance, the complexity and duration that you would typically take to complete each exercise are mentioned.
3. You may use IBM Learning, online medium, or SME interaction to understand how to technically solve these exercises.

### Instructions

1. **IMPORTANT:** While attempting Automation scripting:
  - Perform Exception Handling for all exercises using the try-except block. For web pop-ups and alerts, use void dismiss/void accept/String getText/void sendKeys.
  - Use the Assertions driver.title method to match the window title, and find\_element\_by\_link\_text and find\_element\_by\_partial\_link\_text to check for the presence of the link.
  - Apply verification logic in all traversed landing pages.
  - Prepare an HTML test report covering the automation run.
  - Verify the user interface (UI) elements captured on a web page.
2. While working on these exercises, design a simple framework containing the basic aspects such as handling web elements, handling services, database connection, Excel sheet, reporting, and debugging.
3. You must successfully execute the automated script before marking an exercise complete.

### Installation and Set Up

Download and install the latest versions of following software:

Python	<a href="https://www.python.org/downloads/">https://www.python.org/downloads/</a>
Selenium Python Web Driver	<a href="https://pypi.org/project/selenium/#files">https://pypi.org/project/selenium/#files</a>
Eclipse IDE for Developers	<a href="http://www.eclipse.org/downloads/">http://www.eclipse.org/downloads/</a>
Selenium Python Client Driver	<a href="https://selenium.dev/downloads/">https://selenium.dev/downloads/</a>
Firefox Gecko Driver	<a href="https://github.com/mozilla/geckodriver/releases">https://github.com/mozilla/geckodriver/releases</a>
Chrome	<a href="https://sites.google.com/a/chromium.org/chromedriver/downloads IDE/">https://sites.google.com/a/chromium.org/chromedriver/downloads IDE/</a>
Edge	<a href="https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/">https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/</a>
Safari	<a href="https://webkit.org/blog/6900/webdriver-support-in-safari-10/">https://webkit.org/blog/6900/webdriver-support-in-safari-10/</a>

1. Download and install Python from the above link
2. Open the command prompt, navigate to the path where you have Python installed, and type **pip list**. This command will list all the libraries currently available in your Python.



## Test Automation – Foundation Level Hands-On Exercises

- =====
3. Open command prompt “PIP Install Selenium” - Once you execute the command, Selenium libraries will be downloaded and installed
  4. Verify with the “PIP list” command
  5. Download and Install PyCharm (Python IDE)

### Configuration Setup

1. Open PyCharm
2. Click on Create New Project. By default, the name of the project is taken as untitled. Enter an appropriate project name. Click on Create.
3. To verify if Selenium libraries are configured, go to **File -> Settings**. Navigate to **Project -> Project Interpreter** on the setting page.
4. Under packages, you should be seeing the Selenium package. If that is missing, hit on the “+” button in the right corner. Under available packages, search for Selenium and hit Install Package. Now verify if the Selenium package is installed.
5. Launch PyCharm, right-click on the **Project**, and **Create New Directory**.
6. Under the Main directory, create a New Python File. This will create a .py file and opens the editor.
7. Now copy the extracted .exe driver, for example, **Chromedriver.exe** and paste the file into the Drivers directory.

### Background Information

**The below website has been used for Test Automation in the following exercises. The purpose of this website is described briefly below:**

- <https://www.moneycontrol.com/>

This is a financial platform that tracks the latest updates on Indian and Global financial markets. It also covers multiple assets from the BSE, NSE, MCX, and NCDEX exchanges so that you can track Indices (Sensex and Nifty), Stocks, Futures, Options, Mutual Funds, Commodities, and Currencies with ease. It has platforms across the web, mobile, and tablets which makes it the largest online financial platform in India.

**Note: The application under test is a public website. It is possible that some of the images, texts or sub-links may change. In such cases, use the most appropriate alternative to simulate the use case conditions.**



## Test Automation – Foundation Level Hands-On Exercises

### Use Case 1 - Capture the gold commodity price over different periods on moneycontrol.com

**Details:** The commodity prices on different exchanges are displayed. There is a **Commodities** link at the top of the page that displays the commodities' prices. The **Commodity Prices** section contains the Commodity name, Price, Change, and % Chg. For example, for Gold, it displays something like 04 Dec 2020, 50,608.00, -18.00, -0.04. After clicking the Commodity name, the price of different periods is displayed.

Capture the gold price details in an Excel sheet. The testing of the whole process should be automated.

Transaction complexity	Automation complexity	Recommended duration
Low	Medium	1.5 to 3 hours
Selenium elements simulated		
<ul style="list-style-type: none"> <li>• Install Openpyxl by typing “<b>pip install robot-framework-openpyxl-lib</b>” in cmd</li> <li>• Import OpenPyxlLibrary, ExcellLibrary, and create a function to read data from an excel file.</li> <li>• Use Suite Setup and Suite Teardown methods.</li> <li>• For example, Open Browser with URL, Capabilities, and so on. can be a part of Suite Setup</li> <li>• Close All Browsers or Close Browser can be a part of Suite Teardown.</li> <li>• Sample: <ul style="list-style-type: none"> <li>Library Selenium Library</li> <li>Suite Setup    Open browser    \${URL}    \${BROWSER}</li> <li>Suite Teardown Close All Browsers</li> </ul> </li> <li>• Open moneycontrol.com using Suite Setup</li> <li>• Use Page Object Model in Robot Framework</li> <li>• Under Resources, PageObject → Keyword definition file → Locators → Testdata</li> <li>• Keep excel in one path and use load_workbook() Method</li> </ul>		
<b>The following MUST be considered in the automation script:</b> <ul style="list-style-type: none"> <li>• For all of the displayed periods, capture the Price, Change, and % Chg.</li> <li>• Note that Gold will be displayed under MCX only.</li> </ul>		

### Use Case 2 - Find which currency has high value

**Details:** In the middle of the page, there is a **Currencies** link on the website. Clicking it takes the user to the **Currency Exchange Rate** table that displays the exchange rate for various currencies.

Notice the exchange rate of the Indian rupee for different currencies in the table, for example, 1 US \$ = ₹73.3225, 1 Euro € = ₹86.4041. Capture these in an Excel sheet and mention the three currencies with the highest exchange rate with the Indian rupees. The testing of the whole process should be automated.

Transaction complexity	Automation complexity	Recommended duration
Low	Medium	1.5-3hours
Selenium elements simulated		
<ul style="list-style-type: none"> <li>• Use the steps from Use Case 1 as need be.</li> <li>• Import RPA.Browser.Selenium Library</li> <li>• Handle Cookies: Click Element When Visible css=.accept-button</li> <li>• Use Openpyxl to handle Excel-related activities</li> </ul>		



## Test Automation – Foundation Level Hands-On Exercises

<ul style="list-style-type: none"><li>• Determine elements Locator's pattern</li><li>• Use Loops to find out the highest exchange rate</li><li>• Keep excel in one path and use load_workbook() Method</li><li>• Store every data value used in a variable (rather than hard coding)</li><li>• Determine elements Locator's pattern</li></ul>
<b>The following MUST be considered in the automation script:</b> <ul style="list-style-type: none"><li>• Store the exchange rate output from the page in an Excel file</li><li>• Sort the records in descending order based on the exchange rate</li></ul>

Use Case 3 - Calculating Gratuity using Calculator		
<b>Details:</b> The moneycontrol website has the functionality to calculate Gratuity post-retirement or on leaving the service. By using the <b>Gratuity Calculator</b> link, the user can enter the details about whether <b>Have you completed 5 years in your job?</b> , <b>Monthly Salary</b> , and <b>Number of Years/months in job</b> . Then the tool calculates the gratuity amount. The testing of the whole process should be automated using the Robot framework.		
Transaction complexity	Automation complexity	Recommended duration
Low	Low	1.5-3hours
Selenium elements simulated		
<ul style="list-style-type: none"><li>• Use the steps from earlier use cases as need be.</li><li>• Suite Setup Open browser \${URL} \${BROWSER}</li><li>• Suite Teardown Close All Browsers</li><li>• Use Evaluate from Built In library to calculate the Gratuity amount</li><li>• Handle drop-down using Select/CSS/value e.g., Select From List By Index drop-down 1</li><li>• Use Page Object Model in Robot Framework</li><li>• Store each web page object as a class</li><li>• Handle check/uncheck Checkbox by using Select class with CSS to answer the questions on the page</li><li>• Use Click link "href = url"</li><li>• Use the Get text() method for the data in the sheet</li><li>• Use the Collection class</li><li>• Use the Actions class, window handling, edit boxes, drop-downs, and text boxes</li></ul>		
<b>The following MUST be considered in the automation script:</b> <ul style="list-style-type: none"><li>• Use the up/down arrow keys to increase/decrease the value.</li><li>• Verify the functioning of the <b>Reset</b> button.</li><li>• Verify the error messages when mandatory fields are left blank and when incorrect values are entered in the fields.</li><li>• Populate an Excel sheet with data values and have the Selenium script read the values from this sheet to enter on the page. Also captured in the sheet is the gratuity amount displayed on the page.</li></ul>		