

Install and Explore Selenium

Step 1: Install Java Development Kit (JDK)

Selenium requires Java to run its Web Driver server.

On Windows/Mac/Linux:

1. Download and install the JDK from the [official Oracle website](#) or install using a package manager.
2. Verify Java installation: `java -version`

Step 2: Install Python

If you plan to use **Python** with Selenium (highly recommended):

1. Download Python from [Python.org](#), or use a package manager.
2. Verify the installation: `python -version`

Step 3: Install Selenium Web Driver for Python

Selenium can be installed as a Python package.

```
pip install selenium
```

Step 4: Install Web Driver for Your Browser

Each browser requires a specific Web Driver executable to interface with Selenium:

1. **Chrome:** Download [Chrome Driver](#).
2. **Firefox:** Download [Gecko Driver](#).
3. **Edge:** Download [Edge Driver](#).

Step 5: Write and Run a Simple Selenium Test Script

Create a basic Python script to verify the setup. Save this as `test_script.py`:

```
from selenium import webdriver

from selenium.webdriver.common.by import By

import time
```

Replace 'path_to_webdriver' with the actual path to your WebDriver (e.g., 'chromedriver' for Chrome)

```
driver = webdriver.Chrome(executable_path="path_to_webdriver")

# Open a website

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

# Find the search box, type a query, and submit

search_box = driver.find_element(By.NAME, "q")

search_box.send_keys("Selenium automation")

search_box.submit()

# Wait for results to load

time.sleep(3)

# Close the browser

driver.quit()
```

Run the script: `python test_script.py`

Step 6: Explore Selenium Functionalities

1. **Locate Web Elements:** Use methods like `find_element()` and `find_elements()` with locators (e.g., `By.ID`, `By.CLASS_NAME`, `By.NAME`, `By.XPATH`).
2. **Perform Actions:** Use methods to click, send keystrokes, and submit forms.
3. **Assertions:** Compare page elements with expected values to automate testing outcomes.
4. **Handle Pop ups and Alerts:** Selenium has methods like `switch_to.alert` to manage dialogs.
5. **Manage Cookies and Sessions:** Control cookies with `driver.get_cookies()` and session handling.
6. **Take Screenshots:** Capture screenshots with `driver.save_screenshot('screenshot.png')`.

Step 7: Automate Testing Across Multiple Browsers

Set up similar test scripts for different browsers by swapping the WebDriver (e.g., `webdriver.Firefox()` for Firefox). This lets you verify cross-browser compatibility.

NAME :Hima bindu

USN :23btrcl228

BRANCH:Aiml,sec-c