Selenium

What is selenium - Selenium is a set of tools and libraries that automates web browser actions .

*Selenium is not a tool but a library of tools.

* Selenium is free and open source

📜 History of Selenium

1. Selenium Core (2004) – Jason Huggins

- Selenium was created at **ThoughtWorks** by **Jason Huggins** while testing an internal application.
- It was a **JavaScript-based framework** that allowed automated browser control.
- Known as **Selenium Core**, it injected JS code into the browser to mimic user actions.

2. Selenium RC – Remote Control (2005)

- Selenium Core had limitations due to browser security policies (same-origin policy).
- **Paul Hammant** developed **Selenium Remote Control (RC)** to overcome this.
- RC introduced a **server that injected JavaScript** into the browser from a different domain.

3. Selenium IDE (2006)

- Developed by **Shinya Kasatani** as a Firefox plugin.
- It provided a **record-and-playback** feature for non-programmers.
- Great for simple test scenarios, but not very flexible.

4. Selenium WebDriver (2009) – Simon Stewart

- WebDriver was developed by **Simon Stewart** at Google.
- It **directly communicated with the browser**, making it **faster, more robust, and modern** than RC.

- Introduced support for multiple browsers and languages.
- Later merged with Selenium in **version 2.0**.

5. Selenium Grid (2008+)

- Introduced to allow **parallel test execution** on multiple machines and browsers.
- Ideal for **cross-browser** and **cross-platform** testing.

6. Selenium 2 (2011)

- Merged **WebDriver and RC** into a single project.
- Marked the deprecation of RC in favor of WebDriver.

7. Selenium 3 (2016)

- Emphasized **WebDriver** as the standard.
- **Selenium RC was officially removed**.
- Improved support for mobile automation (via Appium) and better browser compatibility.

8. Selenium 4 (2021)

- Major release with:
    - **W3C WebDriver compliance** (more stability and performance).
    - New **Selenium IDE** (rewritten).
    - Enhanced **DevTools support** (e.g., capturing logs, network info).
    - Improved **relative locators**, better grid architecture, and easier debugging.

Features of selenium

*Flexible and extensible

*Multiple language supported

*Multiple browser support

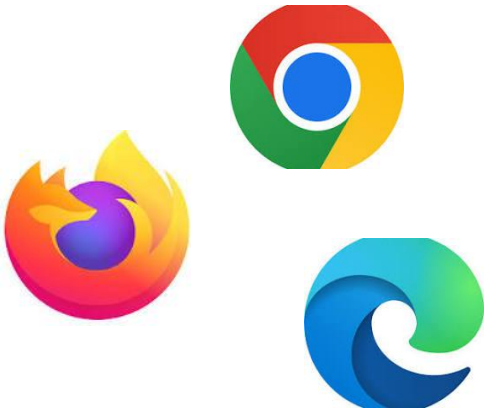
Components of selenium

- **Selenium IDE** – A record and playback plugin ,Useful for quick prototype testing
- **Selenium Rc (Remote control)** – Used to execute scripts (written in any language)
- **Web Driver** – An API used to send commands directly to the browser
- **Selenium grid** – A tool to run tests in parallel across different machines and different browsers simultaneously , Used to minimize execution time

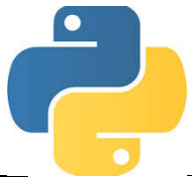
Browsers Supported

All major browsers –

- Firefox
- Chrome
- Internet explorer



Languages Supported by Selenium

Language	Description
Java 	Most widely used with Selenium. Extensive support and community resources.
Python 	Easy syntax and quick to write. Ideal for beginners and rapid automation.
C# 	Popular among Windows developers, especially with Visual Studio and .NET.

JavaScript 	Supported via **WebDriverJS** or frameworks like **Protractor** (for Angular apps).
Ruby 	Elegant syntax. Used less commonly now but still supported.
Kotlin 	Compatible with Java-based tools and works well with Selenium in Android testing contexts.
PHP 	Supported through **third-party bindings** but less commonly used.

Steps to test

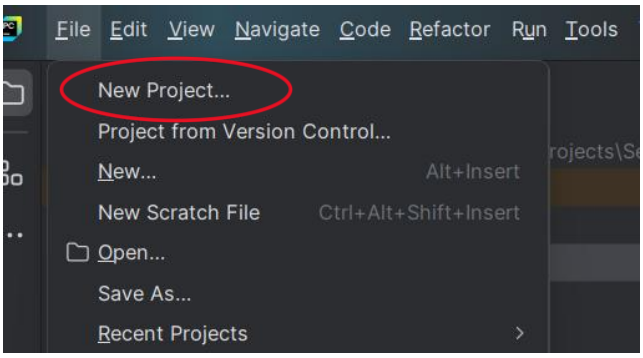
Step 1:

Make sure Python is installed. You can check by running:

python –version

* If not installed, download it from: https://www.python.org/downloads/

Step 2: Create a New Project in PyCharm



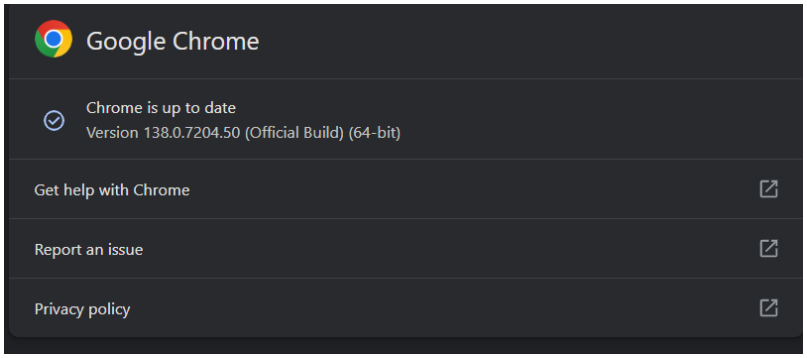
1. Open PyCharm
2. Click **"New Project"**
3. Set a project name (e.g., SeleniumTest)
4. Ensure you select **New Virtual Environment using Python interpreter**
5. Click **"Create"**

Step 3: Install Selenium Package

Using Terminal

In PyCharm terminal (bottom), run:

pip install selenium



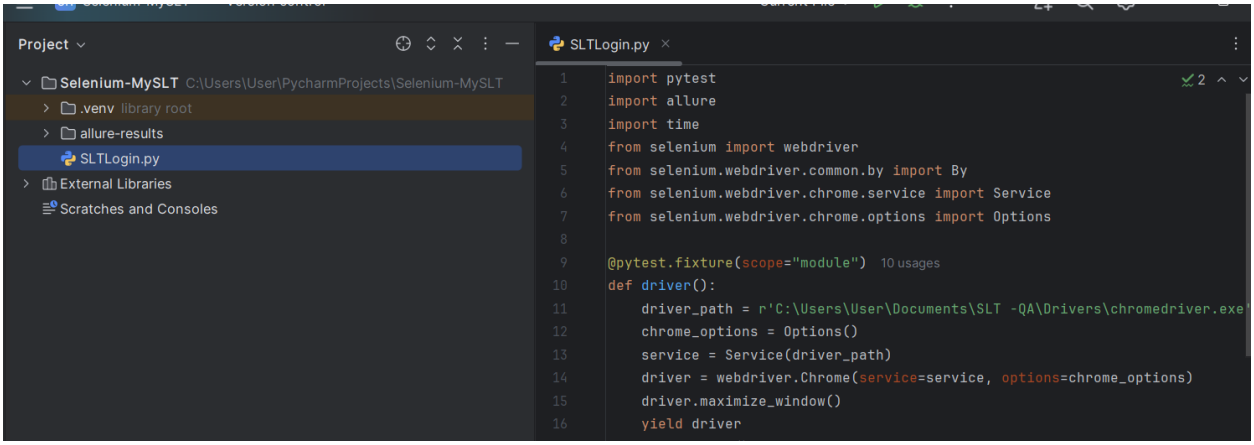
Step 4: Download ChromeDriver

1. Check your Chrome version by visiting: chrome://settings/help
2. Go to: https://googlechromelabs.github.io/chrome-for-testing/
3. Download the matching version of **ChromeDriver** (e.g., version 137)
4. Extract it and place chromedriver.exe in a known folder, like:

   C:\Users\YourName\Drivers\chromedriver.exe

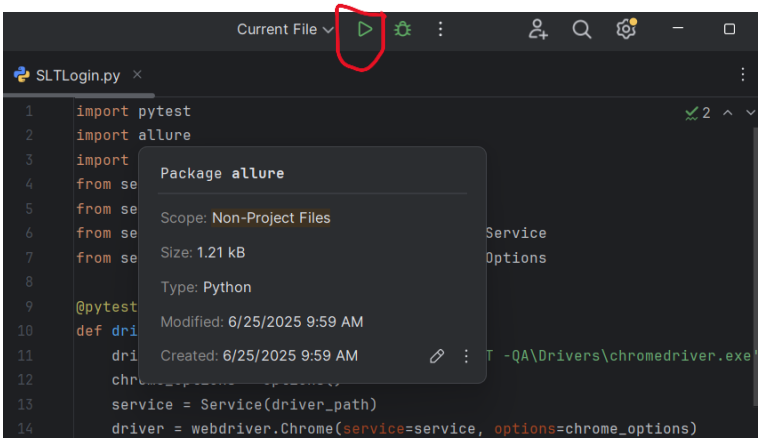
Step 5: Write Your First Selenium Script

Create a Python file in PyCharm (e.g., test_login.py) and paste this:



Step 6: Run the Script

Click on run button



Steps to get allure report

Manual Install

1. Download Allure from:
   👉 https://github.com/allure-framework/allure2/releases
2. Extract it to a folder (e.g., C:\allure)
3. Add the bin path to your system PATH:
   - Open **Environment Variables**
   - Add: C:\allure\bin to the **System or User Path**

4.Restart PyCharm or CMD and check:

allure –version

Step 1: Install Python Allure Dependencies

In your PyCharm terminal:

pip install allure-pytest

Step 3: Write a Selenium Test with Pytest

Step 4: Run the Test and Generate Results

Run this command in the terminal:

pytest --alluredir=allure-results

Step 5: Generate the Allure Report

allure serve allure-results

- Build the report
- Open it automatically in your browser

```
C:\Users\User\PycharmProjects\Selenium-MySLT\.venv\Scripts\python.exe "C:/Program Files/JetBrains/PyCharm Communi
Testing started at 10:11 AM ...
Launching pytest with arguments C:\Users\User\PycharmProjects\Selenium-MySLT\SLTLogin.py --no-header --no-summary

============================= test session starts =============================
collecting ... collected 2 items

SLTLogin.py::test_login_myslt PASSED                                     [ 50%]
SLTLogin.py::test_peo_tv_section PASSED                                  [100%]

============================== 2 passed in 30.12s ==============================

Process finished with exit code 0
```