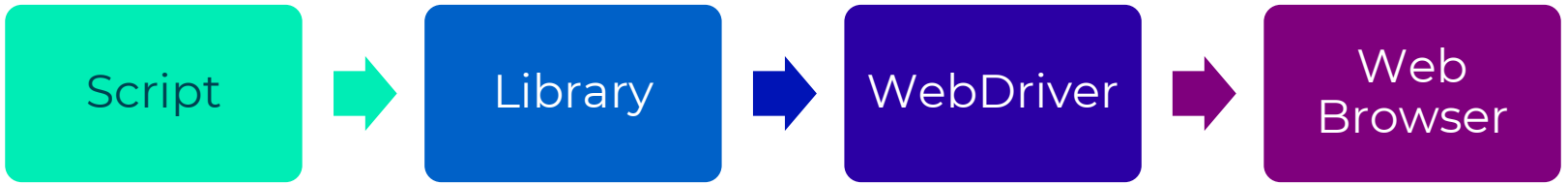
POWERING POTENTIAL

Selenium

Module 3 – WebDriver Introduction

Selenium WebDriver

- WebDriver is a W3C specification with vendor implementations abstracting and defining the process of communication with a browser.
- Has APIs in many programming languages (e.g., Python, Java, C#, Ruby).
- May need drivers for specific Web browsers (downloaded separately).



- When a script creates an instance of WebDriver library object, the library starts a WebDriver process which opens the Web browser.
- The script calls library commands, and the library, through WebDriver, sends the commands to the Web browser.

Selenium WebDriver browser support

Each browser may have a different implementation and version, so downloading the correct WebDriver type and version is important.

Examples:

- Chrome (chromedriver.exe)
- Internet Explorer (IEDriverServer.exe)
- Edge (MicrosoftWebDriver.msi)
- Firefox (geckodriver.exe)
- Safari (safaridriver)
- HtmlUnit (HtmlUnit driver)

Preparing the Chrome Driver for use

Before using a WebDriver implementation, it needs to be made available to a programming language. There are a variety of ways to accomplish this:

- Using a library for managing the WebDriver, such as WebDriverManager by bonigarcia.
- Add the driver to the system path.
- Register the WebDriver as a property.

```
System.setProperty("webdriver.chrome.driver", "/path/to/chromedriver");
```

Selenium testing in Java

In Java, you will also have to add Maven dependencies.

Each JUnit test is independent of all other tests and therefore a new window needs to be opened. After the test has finished the resources need to be released -
the @AfterAll annotation is guaranteed to run even if the @Test throws an exception.

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/junit/junit -
->
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-Jupiter-api</artifactId>
    <version>5.9.1</version>
    <scope>test</scope>
  </dependency>
  <!--
https://mvnrepository.com/artifact/org.seleniumhq.
selenium/selenium-java -->
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.7.2</version>
  </dependency>
</dependencies>
```

Setting up the WebDriver

Java

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

class WebDriverTest {

    private WebDriver driver;

    @BeforeEach
    void setup() {
        driver = new ChromeDriver();
    }

}
```

Open the browser and navigate to page

Java

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

Resizing the browser

The **Window** API can be used to programmatically manipulate a browsers settings, this can be retrieved from the **Options** object available on a **WebDriver** implementation:

```
Window window = driver.manage().window();
```

Method	Description
window.fullscreen()	Sets the browser to full screen mode, equivalent to F11 in most browsers
window.maximise()	Maximises the size of the current window
window.minimise()	Minimises the size of the current window

Closing the browser

Can close the active window (tab) , when the last window is closed, the browser quits:

driver.close()

Can close the browser and the driver at the same time.

driver.quit()

- It is important to call **quit()** on the WebDriver to ensure the resource is not left open after your test has run.

Selenium WebDriver example

```java
@Test
public void searchForPuppiesTest() {
    System.setProperty("webdriver.chrome.driver", "src/test/resources/drivers/chromedriver.exe");

    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(2));
    driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(5));

    driver.get("https://www.bing.com");

    WebElement searchBar = driver.findElement(By.name("q"));
    searchBar.sendKeys("puppies");
    searchBar.submit();

    assertEquals("puppies - Bing", driver.getTitle());

    driver.quit();
}
```

This code creates a driver and does the following:

- Opens bing.com
- Finds the search bar element by its name attribute
- Types “puppies” in the search box
- Submits the form the search bar is in
- Tests if the page title is as expected

Demo Selenium WebDriver

Exercise 1: Google search for kittens

See Exercise 1 in your Selenium WebDriver exercise book.


Thank you!
Hope you enjoyed this learning journey.