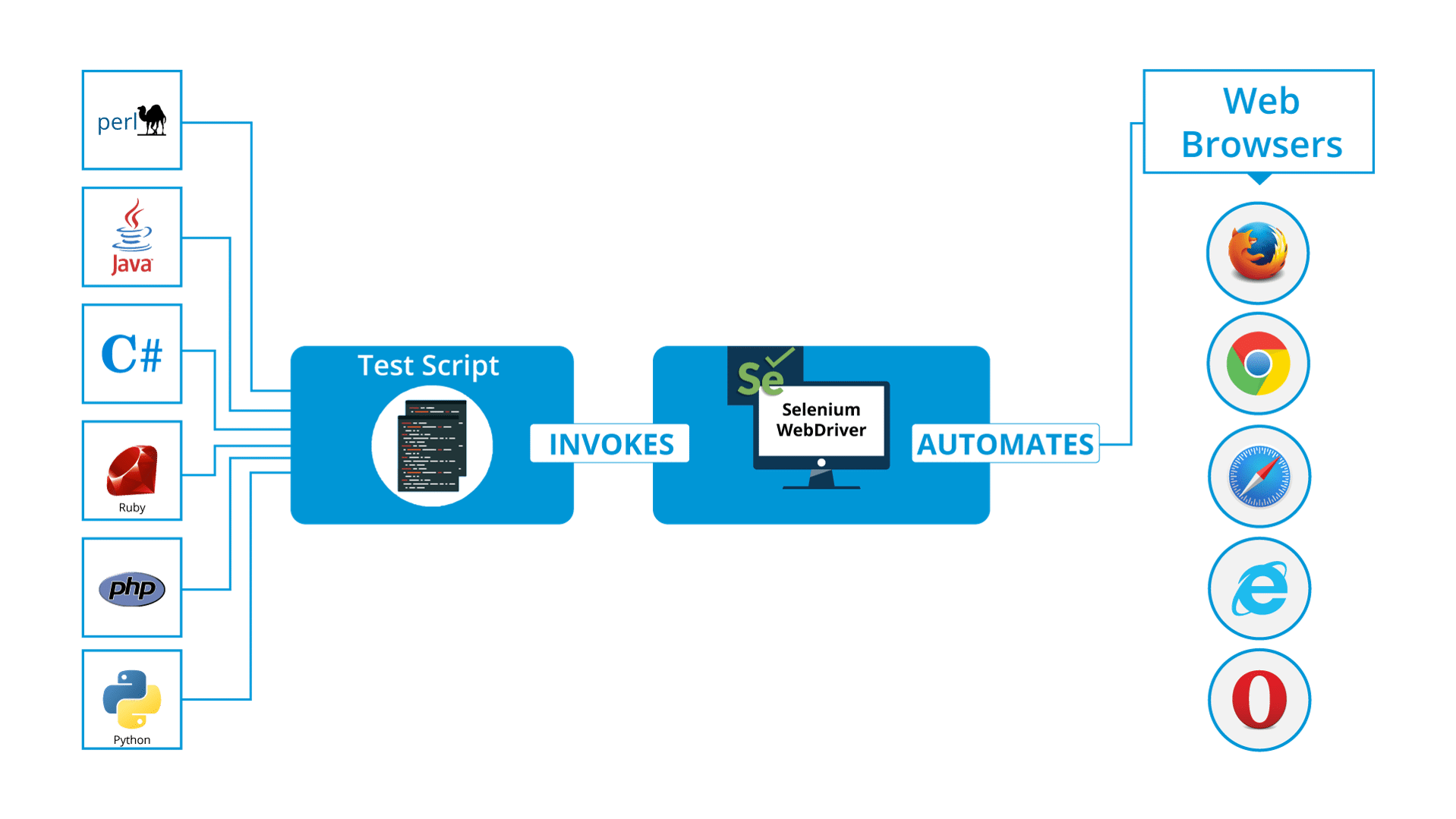
**How WebDriver works:**



Create a new Project

Create a Class

Add Selenium Java jar files through “Build path”

**Script Flow: [Copy Script from mailed document]**

Proper Import statements

Class definition

Main Method definition

Script for all Browser execution

**Selenium WebDriver Commands:**

1. ***get(String arg0): void***

– This method opensa new web page in the current browser window. Accepts String as a parameter and returns nothing.

**Command** – driver.get(appUrl);

1. **getTitle(): String**

– This method fetches the Title of the current page. Accepts nothing as a parameter and returns a String value.

**Command** – driver.getTitle();

1. **getCurrentUrl(): String**

– This method fetches the string representing the Current URL which is opened in the browser. Accepts nothing as a parameter and returns a String value.

**Command** – driver.getCurrentUrl();

1. **getPageSource(): String**

– This method returns the Source Code of the page. Accepts nothing as a parameter and returns a String value.

**Command** – driver.getPageSource();

1. **close(): void**

– This method Close only the current window the WebDriver is currently controlling. Accepts nothing as a parameter and returns nothing.

**Command** – driver.close();

1. **quit(): void**

– This method closes all windows opened by the WebDriver. Accepts nothing as a parameter and returns nothing.

**Command** – driver.quit();

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** FirstSeleniumProg {

**static** WebDriver *driver*;

**public** **static** **void** main(String[] args) {

String url = "https://www.cleartrip.com/";

//Instantiating driver object and launching browser

*driver* = **new** ChromeDriver();

//Using get() method to open a web page

*driver*.get(url);

//to get the Title of the Web page

String title = *driver*.getTitle();

System.***out***.println(title);

//to get the string representing the Current URL

String CurrentUrl = *driver*.getCurrentUrl();

System.***out***.println(CurrentUrl);

//o read the page source of the WebPage

String PageSource = *driver*.getPageSource();

System.***out***.println(PageSource);

//to close the current window in Browser

*driver*.close();

//Closing the browser

*driver*.quit();

}

}

**Practice Exercise – 1**

1. Launch a new Chrome browser.
2. Open URL: https://www.marshallspetzone.com
3. Get Page Title name and Title length [use length() method]
4. Print Page Title and Title length on the Eclipse Console.
5. Get Page URL and verify if it is a correct page opened [use equals() method]
6. Get Page Source (HTML Source code) and Page Source length
7. Print Page Length on Eclipse Console.
8. Close the Browser

**Sample OutPut: Pass**

Title of the page is: Buy pet food Online Shop for Dog, Cat & Bird Supplies

Length of the title is: 53

Verification Successful - The correct Url is opened.

Total length of the Page Source is: 206060

**Sample OutPut: Fail**

Title of the page is: Buy pet food Online Shop for Dog, Cat & Bird Supplies

Length of the title is: 53

Verification Failed - An incorrect Url is opened.

Actual URL is: https://www.marshallspetzone.com/

Expected URL is: https://www.marshallspetzone.com

Total length of the Page Source is: 206076

**Selenium Navigation Commands:**

1. **to(String arg0) : void**

– This method Loads a new web page in the current browser window. It accepts a String parameter and returns nothing.

**Command** – driver.navigate().to(appUrl);

1. **forward() : void** – This method does the same operation as clicking on the Forward Button of any browser. It neither accepts nor returns anything.

**Command** – driver.navigate().forward();

1. **back() : void** – This method does the same operation as clicking on the Back Button of any browser. It neither accepts nor returns anything.

**Command** – driver.navigate().back();

1. **refresh() : void** – This method Refresh the current page. It neither accepts nor returns anything.

**Command** – driver.navigate().refresh();

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** Navigation\_cmd {

**static** String *urlValue*;

**public** **static** **void** main(String[] args) {

// Create a new instance of the ChromeDriver

WebDriver driver = **new** ChromeDriver();

driver.manage().window().maximize();

// Open cleartrip web site

String appUrl = "https://www.cleartrip.com/";

driver.navigate().to(appUrl);

// Wait For Page To Load

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));

driver.findElement(By.*xpath*("//div[@class='pt-6 pb-6 flex flex-top flex-between ']/div[2]")).click();

// Click on Bus link

driver.findElement(By.*xpath*("//div[@class='mt-2 mb-2 d-block fs-6']/ul[@class='ls-reset']/li[3]/a")).click();

// Wait For Page To Load

driver.manage().timeouts().implicitlyWait(Duration.*ofSeconds*(10));

// Go back to Home Page

driver.navigate().back();

*urlValue* =driver.getCurrentUrl();

System.***out***.println(*urlValue*);

// Go forward to Bus page

driver.navigate().forward();

*urlValue* =driver.getCurrentUrl();

System.***out***.println(*urlValue*);

// Refresh browser

driver.navigate().refresh();

// Close browser

driver.close();

}

}

**Practice Exercise – 2**

1. Invoke Firefox Browser
2. Navigate to URL: <https://www.redbus.in/>
3. Click on the "Bus Hire" link (This link will redirect you to the bushire page)
4. Come back to the Home page using the back command
5. Again go back to the bushire page using forward command
6. Again come back to the Home page using To command
7. Refresh the Browser using Refresh command
8. Close the Browser

**What Are Browser Elements?**

Elements are the different components (fields) that are present on web pages. The most common elements we notice while browsing are:

* Text boxes
* Buttons
* Images
* Hyperlinks
* Radio buttons
* Check boxes
* Text area
* Drop down box/ List box/ Combo box
* Web Table/ HTML Table
* Frames

Only after the elements are located on the web page, we can perform operations (actions) and start testing them.

Any of the below 8 attributes can be used to locate elements uniquely.

Since the elements are located using these attributes, we refer to them as ‘Locators’.

The locators are:

1. By.id
2. By.name
3. By.className
4. By.tagName
5. By.linkText
6. By.partialLinkText
7. By.css
8. By.xpath

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** org.openqa.selenium.firefox.FirefoxDriver;

**import** org.openqa.selenium.ie.InternetExplorerDriver;

**public** **class** ChromeBrowDemo {

**static** String *browser*;

**static** WebDriver *driver*;

**public** **static** **void** setBrowser()

{

*browser* = "Chrome";

}

**public** **static** **void** setBrowserConfig()

{

**if**(*browser*.equals("Edge"))

{

//Instantiating driver object and launching browser

// Start Edge Session

*driver* = **new** EdgeDriver();

System.***out***.println("Edge browser opened");

}

**if**(*browser*.equals("Chrome"))

{

//Instantiating driver object and launching browser

*driver* = **new** ChromeDriver();

System.***out***.println("Chrome browser opened");

}

**if**(*browser*.equals("Firefox"))

{

//Instantiating driver object and launching browser

*driver* = **new** FirefoxDriver();

System.***out***.println("Firefox browser opened");

}

}

**public** **static** **void** main(String[] args) {

// Instantiation of driver with Webdriver Interface

*setBrowser*();

*setBrowserConfig*();

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

*driver*.close();

System.***out***.println("Browser Closed");

}

}

Referencces:

<https://artoftesting.com/selenium-webdriver-commands-list>

<https://www.edureka.co/blog/selenium-tutorial#WhatIsSeleniumWebDriver>

<https://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/>

<https://www.toolsqa.com/selenium-webdriver/selenium-webdriver-browser-commands/>

<https://www.seleniumeasy.com/selenium-tutorials/navigation-methods-webdriver-examples>