**Selenium Java**

**Selenium Features:**

* Open-source automation Testing Tool.
* Only for web-based applications.
* Supports multiple browsers (Chrome, Firefox, IE, Safari) and platforms (Windows, Apple, Linux).
* Can be coded in multiple languages (Java, C#,PHP,JS,Python,Ruby)

**Diff b/w Selenium and Web driver:**

**Selenium:** To automate web browsers across many platforms and it is a tool.

Selenium releases multiple tools

* Webdriver – robust and browser-based regression automation tests and suites.
* IDE – not used much used only for record and playback.

**Selenium Webdriver Architecture:**

Selenium Client Library 🡪 JSON 🡪 Browser Drivers 🡪 Browsers

(Eclipse for Java) (Converts) (Chrome Driver, IEDriver) (Chrome, IE)

* After the test is triggered complete selenium code will be converted to JSON format.
* It will send to Browser driver through HTTP Protocol, and browser drivers communicates with the browsers and executes the commands by JSON which it received on the browser.
* Browser drivers receive response back from the browser and sends to JSON response back to client.

**Quick Points to remember:**

close(); //Closes only particular window

quit(); //Closes all associated windows opened by automation.

**Code:**

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.edge.EdgeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class SelIntroduction {

public static void main(String[] args) {

//Invoking Browser

//Chrome - ChromeDriver exten->Methods close get

//Firefox- FirefoxDriver ->methods close get

// WebDriver  close  get

//WebDriver methods + class methods

// Chrome

         ChromeOptions options = **new** ChromeOptions();

         options.addArguments("--remote-allow-origins=\*");

System.setProperty("webdriver.chrome.driver", "/Users/shivani/Documents/chromedriver");

WebDriver driver = new ChromeDriver(options);

//Firefox

System.setProperty("webdriver.gecko.driver", "/Users/shivani/Documents/geckodriver");

WebDriver driver1 = new FirefoxDriver();

//Microsoft Edge

System.setProperty("webdriver.edge.driver", "/Users/shivani/Documents/msedgedriver");

WebDriver driver2 = new EdgeDriver();

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

System.out.println(driver.getTitle());

System.out.println(driver.getCurrentUrl());

driver.close();

//driver.quit();

}

}

**CORE JAVA:**

Arrays and For loop:

**public** **class** CoreJavaBrushUp {

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

// **TODO** Auto-generated method stub

**int** myNum = 5;

String website = "Sel Java Academy";

**char** letter = 'r';

**double** dec = 5.99;

**boolean** myCard = **true**;

System.**out**.println(myNum+"is the value stored in the myNum variable");

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

//Arrays -

**int**[] arr = **new** **int**[5];// 5, 10

arr[0] = 1;

arr[1] = 2;

arr[2]= 4;

arr[3]= 5;

arr[4]= 6;

**int**[] arr2 = {1,2,4,5,6};

System.**out**.println(arr2[0]);

//for loop arr.length - 5

**for**(**int** i = 0; i< arr.length; i++) //4

{

System.**out**.println(arr[i]);

}

**for** (**int** i =0;i<arr2.length; i++)

{

System.**out**.println(arr2[i]);

}

String[] name = {"vaishu", "shivani", "selenium"};

**for**(**int** i =0; i<name.length;i++)

{

System.**out**.println(name[i]);

}

**for**( String s: name)

{

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

}

}

}

**Arraylist:**

Dynamically grows the size of an array.

**Diff b/w Array and ArrayList:**

**Array** has a static size where we set initially whereas **ArrayList** dynamically grows by adding multiple items into list and have lot of operations.

**Strings:**

Object that represents sequence of characters.

Ways to define it:

1. String literal: same multiple value with same name it won’t create new object, it will refer the existing object
2. Using new memory allocation operator. – no matter if it is a same string value it will still explicitly create an object in the memory space.

**Code:**

**public** **class** CoreJavaBrushUp3 {

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

// **TODO** Auto-generated method stub

//string is an object //String literal

// String s1 = "Java Academy";

String s5 = "hello";

//new

String s2 = **new** String("Welcome");

String s3 = **new** String("Welcome");

String s = "Java Academy";

String[] splittedString = s.split(" ");

System.**out**.println(splittedString[0]);

System.**out**.println(splittedString[1]);

System.**out**.println(splittedString[1].trim());

**for**(**int** i =s.length()-1;i>=0;i--)

{

System.**out**.println(s.charAt(i));

}

}

}

//Methods Code

**public** **class** MethodsDemo {

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

// **TODO** Auto-generated method stub

MethodsDemo d = **new** MethodsDemo();

String name = d.getData();

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

MethodsDemo2 d1 = **new** MethodsDemo2();

d1.getUserData();

getData2();

}

**public** String getData()

{

System.**out**.println ("hello world");

**return** "java academy";

}

**public** **static** String getData2()

{

System.**out**.println ("hello world");

**return** "java hello";

}

}