SELENIUM

- ➢ Selenium is a tool
- ➢ It is a portable framework for testing web application
- ➢ It is an open-source tool for creating test suite
- ➢ Helps in testing web application across browsers & platforms
- ➢ It is not a single tool but suite of softwares

Benefits of Selenium
- Selenium is an open source and portable Web testing Framework
- supports various operating systems, browsers and programming languages
- supports parallel test execution that reduces time and increases the efficiency
- Selenium commands are categorized in terms of different classes

Selenium Limitations
- Selenium does not support desktop applications.
- Needed support from community forums to get the technical issues resolved
- should know at least one of the supported programming languages
- does not have any inbuilt reporting capability
- not possible to perform testing on images

Selenium WebDriver:
- ✓ test scripts can be developed using any of the supported programming languages and can be run directly in most modern web browsers.
- ✓ Languages supported by WebDriver include C#, Java, Perl, PHP, Python and Ruby
- ✓ Most of the commands used in Selenium WebDriver are simple & easy to implement
- ✓ Handles Navigation, Alerts / Dropdowns, Switching between windows, etc.

Finding Element:
It is important in a webpage; Each element has 3 important attributes: id, name & class
Will have to access HTML tags

Selenium WebDriver, uses a set of commands for performing different operations.
Commands are simply **methods** written in Java language
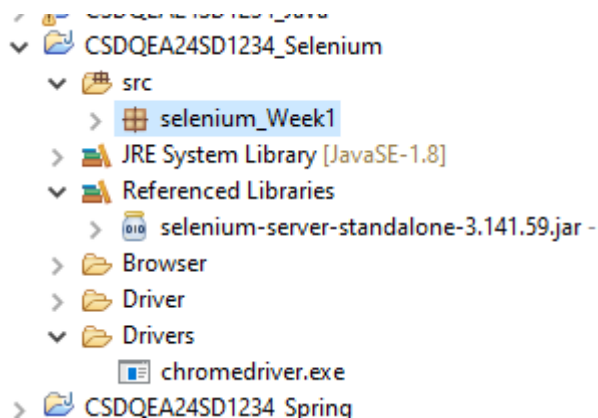
What should a script contain?
- Ensure relevant packages / classes are imported
- Invoke a browser
- Create instance of the Web Driver
- Maximize the screen window
- Give the URL of the webpage & launch the website
- Wait till the web page loads
- Find each Web Element you want to access
- Invoke an action on each of those Web Element
- Finally close the Web Driver

Locators:
1. Attribute
   a. Id

b. Name
c. Class
2. Text
   a. Link Text
   b. Partial Link Text
3. TagName
   a. TagName
4. Dynamic Locator - XPATH
   a. Absolute xpath
   b. Relative xpath

<u>Folder Structure</u>:

```
  CSDQEA24SD1234_Selenium
    src
      selenium_Week1
    JRE System Library [JavaSE-1.8]
    Referenced Libraries
      selenium-server-standalone-3.141.59.jar -
    Browser
    Driver
    Drivers
      chromedriver.exe
  CSDQEA24SD1234_Spring
```

Example 1: Basics - Locators

```java
package selenium_Week1;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Ex1_SeleniumBasics {

    public static void main(String[] args) {


        System.setProperty("webdriver.chrome.driver",
```

```java
"E:\\Selenium\\Programs\\CSDQEA24SD1234_Selenium\\D
rivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.facebook.com/");

    driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);

        System.out.println("Title of the Webpage :
" + driver.getTitle());

//      WebElement email =
driver.findElement(By.id("email"));
//      email.sendKeys("abc@gmail.com");
//
//      WebElement login =
driver.findElement(By.name("login"));
//      login.click();

//      WebElement forgotPwd =
driver.findElement(By.linkText("Forgotten
password?"));
        WebElement forgotPwd =
driver.findElement(By.partialLinkText("Forgotten"))
;
        forgotPwd.click();

        driver.close();
    }

}
```

Example 2: (Navigate WeSites in single tab of browser)

```java
package selenium_Week1;
```

```java
import java.util.concurrent.TimeUnit;

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

public class Ex2_NavWebSite {

    public static void main(String[] args) throws
InterruptedException {


    System.setProperty("webdriver.chrome.driver",
"E:\\Selenium\\Programs\\CSDQEA24SD1234_Selenium\\D
rivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.facebook.com/");

    driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);

        System.out.println("Title of the Webpage :
" + driver.getTitle());
        System.out.println("Current URL : " +
driver.getCurrentUrl());

        Thread.sleep(3000);

    driver.navigate().to("https://www.google.com/")
;
        System.out.println("Title of the Webpage :
" + driver.getTitle());
        System.out.println("Current URL : " +
driver.getCurrentUrl());

        Thread.sleep(3000);
        driver.navigate().back();
```

```java
        System.out.println("Current URL : " +
driver.getCurrentUrl());

        Thread.sleep(3000);
        driver.navigate().forward();
        System.out.println("Current URL : " +
driver.getCurrentUrl());

        Thread.sleep(3000);
        driver.navigate().refresh();
    }

}
```

Example 3: (working with Dropdown)

```java
package selenium_Week1;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

public class Ex3_UseList {

    public static void main(String[] args) throws
InterruptedException {


        System.setProperty("webdriver.chrome.driver",
"E:\\Selenium\\Programs\\CSDQEA24SD1234_Selenium\\D
rivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
```

```java
    driver.get("https://www.roboform.com/filling-test-all-fields");

    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        WebElement month = driver.findElement(By.name("66mm"));

//      Thread.sleep(3000);
//      Select selMonth = new Select(month);
//      selMonth.selectByIndex(2);
//
//      Thread.sleep(3000);
//      selMonth.selectByIndex(5);
//
//      Thread.sleep(3000);
//      selMonth.selectByVisibleText("Aug");

//      Thread.sleep(3000);
//      selMonth.selectByValue("11");
//
//      System.out.println("Selected value in list is .. " +
selMonth.getFirstSelectedOption().getText());

        month.click();
        month.sendKeys("Jun");
        month.click();
    }

}
```

Exercises to work (in Eclipse):

Exercise 1:

Access all the elements in facebook page by finding element thru Id, Name, Class & text
Fill up valid details as required for each field, but don't submit;
Print tag name, some attribute value, etc for elements

Exercise 2:

Navigate front & back between different 3 different websites
Print title & URL of each navigation

Exercise 3:

Open "Facebook" from one of three browsers (Chrome / Firefox / IE).
You have to get browser name as input from user & had to open "Facebook" in that browser

Exercise 4:

https://www.roboform.com/filling-test-all-fields

a. Extract the default values from all the dropdown & print in console
b. Work with any list element; Pick a value, extract the text & print in console; do this few time &
each time when you
 select a value do it with index, value, visibletext, sendkeys, etc; Do this for both list with numeric &
non-numeric values
c. Fill all the form elements & clear it finally

Exercise 5:

Go to this link: "https://demoqa.com/forms"
Access all the elements & fill up the form

Exercise 6:

Go to this link: https://demoqa.com/ and click on Elements.
Work on TextBox, CheckBox, RadioButton, Buttons & Links
Locate the elements & do necessary action on it
Extract values from TextBox, CheckBox & RadioButtons & print in console output

XPATH:

- It is a query language used for navigating thru XML documents to locate different elements
- It is a strategy to locate elements in Selenium
- It starts with double slash
- Uses single slash between each condition
- Uses double slash in between the xpath to by-pass parent or multiple child tags
- For attribute, use @
- While searching value, do remember that the value is case & space sensitive
- Use and, or operators to give multiple conditions

- Multiple conditions may use different attribute of same tag or same attribute with different values of a tag
- "Not" operator used find an attribute not existing in the tag
- Use "contains" and "text" functions
- Xpath index starts with 1

Syntax:

//TagName

      //div

//TagName[index]

      //div[2]

//TagName[@attribute]

      //input[@name]

//TagName[@attribute='AttributeValue']

      //input[@name='04lastname']

//TagName[text()]

//TagName[text()='TextValue']

      //div[text()='Last Name']

// TagName[@attribute1='AttributeValue' and @attribute2='AttributeValue']

      //input[@name='04fullname' and @type='text']

// TagName[@attribute='AttributeValue1' or @attribute='AttributeValue2']

      //input[@name='04fullname' or @type='text']

//TagName[not(@attribute]

      //input[not(@name)]

//TagName[not(@attribute='AttributeValue']

      //input[not(@name='04fullname')]

//TagName[contains(@attribute, 'AttributeValue')]

      //input[contains(@name,'04lastname')]

AXES:

- Use of relationship in xpath
- Child, parent, ancestor, descendant, preceding-sibling, following-sibling
- To refer relationship in xpath, use scope operator (::)

➢ While you traverse in a family tree, always travel from top to bottom
➢ Traverse to next nearest relationship

Syntax:

//TagName[relation::relativeTagName]

[xpath]/relation::relativeTagName

Example:

//input[@name='04lastname']/parent::div/parent::div/following-
sibling::div/child::div[2]/child::input

//input[@name='04lastname']/ancestor::div/following-sibling::div/child::div[2]/child::input

Two types of XPATH:

1. Absolute XPATH
   One that is traversing from one tag to next closest tag
   Also traverses top to bottom via the tree structure
   Syntax:
   //Tagname1/TagName2/TagName3

   Example:
           //body/div[2]/form/div/div/div[4]/div[2]/input

2. Relative XPATH

   One that uses relationship

   Syntax:

   //TagName[relation::relativeTagName]

Example 1: (use of XPATH)

```java
package selenium_Week2;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
```

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

public class Ex1_useXPATH {

    public static void main(String[] args) {


        System.setProperty("webdriver.chrome.driver",
"E:\\Selenium\\Programs\\CSDQEA24SD1234_Selenium\\D
rivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("https://www.roboform.com/filling-
test-all-fields");

        driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);

//       WebElement lastName =
driver.findElement(By.xpath("//input[@name='04lastn
ame']"));
//       WebElement lastName =
driver.findElement(By.xpath("//body/div[2]/form/div
/div/div[4]/div[2]/input"));
        WebElement lastName =
driver.findElement(By.xpath("//input[contains(@name
,'04lastname')]"));
        lastName.sendKeys("Jacobs");

        WebElement fullName =
driver.findElement(By.xpath("//input[@name='04lastn
ame']/parent::div/parent::div/following-
sibling::div/child::div[2]/child::input"));
        fullName.sendKeys("John Jacobs");
    }
```

}

ALERTs & POPUPs

Three types of Web Page window formats:

1. Alert
2. Popup
3. Ad

Alerts:

- A small message box which appears on screen to give the user some information or notification
- It notifies the user with some specific warning or error
- At times it asks for permission to perform certain tasks

Three types of alerts:

a) Simple Alert – that displays some information & warning on the screen; will have one OK button
b) Confirmation Alert – that asks permission to do some type of operation; will have OK & Cancel button
c) Prompt Alert – that asks some input from the user and selenium webdriver can enter test using sendkeys; will have OK & Cancel button

Example for Handling Alert:

```java
package selenium_Week2;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Ex2_HandleAlert {
```

```java
    public static void main(String[] args) throws
InterruptedException {


    System.setProperty("webdriver.chrome.driver",
"E:\\Selenium\\Programs\\CSDQEA24SD1234_Selenium\\D
rivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();

    driver.get("file:///E:/TrainerPlan/CorporateTra
iningMaterials/Batch35_CSDQEA24SD002-03-
04/JavaScriptExamples/Example2/input.html");

    driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);

        WebElement num =
driver.findElement(By.id("number"));
        num.sendKeys("6");

        WebElement value =
driver.findElement(By.xpath("//input[@value='Click'
]"));
        value.click();

        Thread.sleep(3000);
        Alert alert = driver.switchTo().alert();
        alert.accept();
    }

}
```

PopUps:

- It is a webpage driven content displayed in an extra window over the webpage
- Popups either partially or fully hide the content of web page
- Popup does not allow to operate on the webpage until popups are closed

WAITs:

- ✓ Most of web applications are developed using some Asysnchronous concepts
- ✓ When a page is loaded in browser, the elements may load at different time intervals
- ✓ Selenium script needs to be informed on these occasions, so script does not fail

Two types:

1. Implicit wait
2. Explicit wait

Implicit wait:

- ➤ It will tell the webdriver to wait for a certain amount of time before it throws an exception
- ➤ If the element is not located on the webpage within that time frame, it will throw the exception
- ➤ This wait is used for whole webpage to be loaded

Explicit wait:

- ➤ It will tell the webdriver to wait for certain conditions to happen or maximum time exceeded before throwing an exception
- ➤ It is a kind of intelligent wait that can be applied only on specified elements
- ➤ This gives better options during execution of script as it waits for dynamically loaded contents

Expected Conditions:

- (i) alertIsPresent()
- (ii) elementToBeClickable()
- (iii) elementToBeSelected()
- (iv) visibilityOfAllElements()
- (v) visibilityOfElementLocated()
- (vi) presenceOfElementLocated()
- (vii) presenceOfAllElementsLocated()
- (viii) textToBePresentInElement()

Example for Handling PopUp along with Explicit Wait:

```
package selenium_Week2;

import java.util.concurrent.TimeUnit;
```

```java
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import
org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Wait;
import
org.openqa.selenium.support.ui.WebDriverWait;

public class Ex3_HandlePopup {

    public static void main(String[] args) throws
InterruptedException {


    System.setProperty("webdriver.chrome.driver",
"E:\\Selenium\\Programs\\CSDQEA24SD1234_Selenium\\D
rivers\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://scoop.eduncle.com/jee-
mains-preparation-tips");

    driver.manage().timeouts().implicitlyWait(10,
TimeUnit.SECONDS);

        Wait<WebDriver> winWait = new
WebDriverWait(driver, 20);
//
    winWait.until(ExpectedConditions.presenceOfAllE
lementsLocatedBy(By.id("entryPopupModal")));

    winWait.until(ExpectedConditions.presenceOfAllE
lementsLocatedBy(By.xpath("//div[@class='modal
entry-popup open']")));
```

```java
        Thread.sleep(3000);
//      WebElement popUp =
driver.findElement(By.name("entryPopupButton"));
        WebElement popUp =
driver.findElement(By.xpath("//button[@class='bg-
close close-dismiss']"));
        popUp.click();
    }

}
```