

***Javascript Executor:

- 1: Sometimes selenium locators are not useful to identify component in webpage
- 2: In order to identify components, JS Executor can be used
- 3: To execute JS commands using selenium, JSExecutor interface methods are used
- 4: To execute JS commands in browser which is opened by selenium, driver object should be upcasted to JSExecutor.

Ques: Why we use javascript in selenium?

- 1: To perform mouse scroll up and scroll down action
- 2: To identify element if selenium locator are not useful
- 3: To handle disable elements.

Syntax:

```
JavascriptExecutor js = (JavascriptExecutor) driver; //casting  
js.executeScript(script,argument);
```

Script means: It is the javascript that needs to be executed

Arg. means: It is the argument to the script, its optional

-->

- 1: if element html code contains id, to identify that component JS command is:
-->document.getElementById("id property value").value=""; ----->to enter text
-->document.getElementById("id property value").click(); ----->To click
- 2: if element html code not contains id, then JS command will be as follows
--->WebElement rv=driver.findElement(Locator type);
-->js.executeScript("arguments[0].value='";rv);

Script:

```
WebDriver driver = new FirefoxDriver();  
driver.get("http://localhost:8080/login.do");  
//creating the javascript executor interface object by Type casting  
JavascriptExecutor js = (JavascriptExecutor) driver;  
//call the javascript Executor method  
js.executeScript("document.getElementById('username').value='admin'");  
WebElement ele = driver.findElement(By.name("pwd"));  
js.executeScript("arguments[0].value='manager';",ele);  
js.executeScript("document.getElementById('loginButton').click()");
```

AssignQues: WATS to login facebook application using JSExecutor

- 3: To scroll within web page, following command is used.
---> js.executeScript("window.scrollTo(x,y);");
- 4: To scroll upto specific element in web page,
---> js.executeScript("window.scrollTo(0,document.body.scrollHeight);");
- 5: To scroll top of the web page,
---> js.executeScript("window.scrollTo(0,document.body.scrollTop);");

Script:

```
WebDriver driver = new FirefoxDriver();
driver.manage().window().maximize();
driver.get("http://localhost:8080/login.do");
//creating the javascript executor interface object by Type casting
JavascriptExecutor js = (JavascriptExecutor) driver;
//call the javascript Executor method
js.executeScript("document.getElementById('username').value='admin'");
WebElement ele = driver.findElement(By.name("pwd"));
js.executeScript("arguments[0].value='manager';",ele);
js.executeScript("document.getElementById('loginButton').click()");
Thread.sleep(3000);
//scroll
js.executeScript("window.scrollTo(0,40);");
Thread.sleep(3000);
//scroll to bottom
js.executeScript("window.scrollTo(0,document.body.scrollHeight);");
Thread.sleep(3000);
//scroll to top
js.executeScript("window.scrollTo(0,document.body.scrollTop);");
```

Ques: Handling disable element by using javascript executor

```
<html>
  <body>
    UN<input type="text" id="user" disabled>
  </body>
</html>
```

Script:

```
WebDriver driver = new FirefoxDriver();
driver.get("file:///C:/Users/gehlo/Desktop/HTML/E2_IST/disable.html");
JavascriptExecutor js =(JavascriptExecutor) driver;
WebElement ele = driver.findElement(By.id("user"));
js.executeScript("arguments[0].value='admin'",ele);
```

Ques: WATS to display ALERT msg using javascript

-->print ALert popup: js.executescript("alert('msg')");

Script:

```
WebDriver driver = new ChromeDriver();
driver.get("https://www.facebook.com/");
JavascriptExecutor js = (JavascriptExecutor) driver;
WebElement ele = driver.findElement(By.id("email"));
js.executeScript("arguments[0].value=''",ele);
js.executeScript("alert('please enter email')");
```

****Screenshot****

Procedure:

- 1: upcast driver reference variable to TakeScreenshot interface.
- 2: To take screenshot, we use getScreenshotAs() which takes output type as an argument
- 3: To specify location, create file class object with path as an argument
- 4: To save screenshot, we use FileUtils class static method copyFile()

Examp:

```
WebDriver driver = new FirefoxDriver();
driver.get("http://localhost:8080/login.do");
//Take webpage screenshot
TakesScreenshot ts = (TakesScreenshot) driver;
//take screenshot
File src = ts.getScreenshotAs(OutputType.FILE);
//define loc
File dest = new File("../Selenium/screenshot/webpage.png");
//store screenshot in loc
FileUtils.copyFile(src, dest);
```

AssignQues: WATS to takescreenshot of google webpage

Ques: WATS to take screenshot of an element

Script:

```
WebDriver driver = new FirefoxDriver();
driver.get("http://localhost:8080/login.do");
WebElement tgt_ele = driver.findElement(By.id("loginButton"));
File src = tgt_ele.getScreenshotAs(OutputType.FILE);
File dest = new File("../Selenium/screenshot/login.png");
FileUtils.copyFile(src, dest);
```

ASSignQues: WATS to take screenshot of facebook login button

***Popup:

-->Sudden window which will be displayed when we click on element

**Types of Popup:

1: Hidden Division Popup:

properties:

- 1: We cannot drag this popup
- 2: We can inspect this popup

Note: Hidden division popup can be handle directly

Ques: WATS to handle hidden popup in abhibus app

Script:

```
WebDriver driver = new FirefoxDriver();
driver.manage().window().maximize();
driver.get("https://www.abhibus.com/");
driver.findElement(By.xpath("//span[text()='Login/SignUp']")).click();
```

```
Thread.sleep(2000);
driver.findElement(By.xpath("//span[text()='Sign in with google']")).click();
```

2: Alert/JavaScript Popup:

Properties:

- 1: we cannot drag this popup
- 2: we cannot inspect this popup
- 3: These popup will contain Ok or Cancel button
- 4: To handle alert Popup, we cannot use Selenium directly
- 5: Before Handling Alert POup, we need to switch selenium focus from main page to Alert using switchTo() method

```
Alert a = driver.switchTo().alert();
```

-->To handle Alert popup, we use ALert interface methods:

- 1: accept(): used to click on OK button
- 2: dismiss(): used to click on cancel button
- 3: getText(): used to get the text present in text field
- 4: sendkeys():used to enter text

***Types of ALert:

- 1: Simple Alert Popup: This popup contains text, ok & cancel button
 - 2: Prompt popup: This popup contains text, textfield & Ok & Cancel button
 - 3: Confirmation Popup: This popup will contains checkbox, text, ok & cancel button
- >To handle this poup, we use getText(),accept(),dismiss() & to handle checkbox we use Robot class

Script:

```
WebDriver driver = new FirefoxDriver();
driver.get("https://the-internet.herokuapp.com/javascript_alerts");
//simple alert popup
driver.findElement(By.xpath("//button[text()='Click for JS Alert']")).click();
Thread.sleep(2000);
Alert sa = driver.switchTo().alert();
System.out.println("Simple Alert popup text -"+sa.getText());
sa.accept();
//Confirmation popup
driver.findElement(By.xpath("//button[text()='Click for JS Confirm']")).click();
Thread.sleep(2000);
Alert ca = driver.switchTo().alert();
ca.accept();
//prompt popup
driver.findElement(By.xpath("//button[text()='Click for JS Prompt']")).click();
Thread.sleep(2000);
Alert pa = driver.switchTo().alert();
pa.sendKeys("hello");
Thread.sleep(2000);
pa.dismiss();
driver.close();
```

Note: After handling Alert popup, selenium focus by default switch to main page

3: File Upload Popup:

-->Seperate window which comes for selecting the file which need to be uploaded.

Script:

```
WebDriver driver = new FirefoxDriver();
driver.get("https://the-internet.herokuapp.com/upload");
WebElement file = driver.findElement(By.xpath("//input[@id='file-upload']"));
file.sendKeys("C:\\Users\\gehlo\\Desktop\\excel.xlsx");
driver.findElement(By.xpath("//input[@value='Upload']")).click();
```

4: Child/Window Popup:

1: we can drag this popup

2: We can inspect this popup

3: This popup will contain URL, Title, Maximize, Minimize, close options

-->to handle child browser popup we need to shift selenium focus from main page to child by using following cmd:

```
driver.switchTo().window(child);
```

**To get child address we use following methods:

1: getWindowHandle(): It is used to get parent ref. address & return type is String.

2: getWindowHandles(): It is used to get parent & child address & Return type: set<String>

Ques: WATS to print total no. of windows displayed count as an output

Script:

```
WebDriver driver = new FirefoxDriver();
driver.get("https://www.naukri.com/");
WebElement rv = driver.findElement(By.id("login_Layer"));
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("arguments[0].click()", rv);
Thread.sleep(3000);
driver.findElement(By.xpath("//span[text()='Sign in with Google']")).click();
//get window refid
Set<String> s = driver.getWindowHandles();
int count = s.size();
System.out.println(count);
driver.close();
```

Ques: WATS to print all the windows title as output

Script:

```
WebDriver driver = new FirefoxDriver();
driver.get("https://www.naukri.com/");
WebElement rv = driver.findElement(By.id("login_Layer"));
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("arguments[0].click()", rv);
```

```
Thread.sleep(3000);
driver.findElement(By.xpath("//span[text()='Sign in with Google']")).click();
//get window refid
Set<String> s = driver.getWindowHandles();
Thread.sleep(3000);
Iterator<String> itr = s.iterator();
while(itr.hasNext())
{
    String refid = itr.next();
    driver.switchTo().window(refid);
    String title = driver.getTitle();
    System.out.println(title);
}
driver.quit();
```

AssignQues: WATS to signup yatra with facebook