**G C REDDY SELENIUM WEBDRIVER**

**Contents:**

<http://www.gcreddy.com/2015/06/selenium-videos.html>

<http://www.gcreddy.com/2016/01/selenium-interview-questions-and-answers.html>

<http://www.gcreddy.com/2016/01/core-java-for-selenium.html>

**Introduction to Selenium WebDriver**  
  
i) Introduction to Selenium WebDriver  
ii) WebDriver Environment Setup  
iii) Create first Selenium Test Case  
-------------------------------------------  
**i) Introduction to Selenium WebDriver**  
Selenium Tool’s Suite  
        Selenium IDE  
        Selenium RC  
        Selenium WebDriver  
        Selenium Grid  
  
> In 2006 Selenium WebDriver was launched at Google.  
> In 2008, the whole Selenium team decided to merge Selenium WebDriver with Selenium RC in order to form more powerful tool called **Selenium 2.0**

Selenium 1.0 + WebDriver = Selenium 2.0  
Selenium 1.0  
(Selenium IDE + Selenium RC + Selenium Grid)  
Selenium 2.0  
(Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)  
Note: Now Selenium RC is only for maintenance projects.  
----------------------  
> It is a most important tool in Selenium Suite.  
> It has Programming interface only, no IDE.  
> Selenium WebDriver supports various programming languages to write programs (Test scripts)  
 Java  
 Python  
 C#  
 Ruby  
 Perl  
 PHP  
  
> Selenium WebDriver supports various Browsers to create and execute Test cases.  
 Mozilla Firefox  
 Google Chrome  
 IE  
 Safari  
 Opera etc...  
> Selenium WebDriver supports various Operating environments.  
 MS Windows  
 Linux  
 Macintosh etc...  
> Selenium WebDriver supports **Data driven Testing and Cross browser testing**.  
> Selenium WebDriver is faster than other tools of Selenium suite.  
> Selenium WebDriver supports Parallel test execution with the help of either JUnit or TestNG.  
-------------------------------  
Drawbacks of selenium Webdriver  
> Selenium WebDriver doesn't have IDE (some difficult to create test cases)  
> No Built-in Result reporting facility.  
> No other tool Integration for Test management.  
> No centralized maintenance of Elements/objects.  
--------------------------------------------------

Advantages:

* Selenium is pure open source, freeware and portable tool.
* Selenium supports variety of languages that include Java, Perl, Python, C#, Ruby, Groovy, Java Script, and VB Script. etc.
* Selenium supports many operating systems like Windows, Macintosh, Linux, Unix etc.
* Selenium supports many browsers like Internet explorer, Chrome, Firefox, Opera, Safari etc.
* Selenium can be integrated with ANT or Maven kind of framework for source code compilation.
* Selenium can be integrated with TestNG testing framework for testing our applications and generating reports.
* Selenium can be integrated with Jenkins or Hudson for continuous integration.
* Selenium can be integrated with other open source tools for supporting other features.
* Selenium can be used for Android, IPhone, Blackberry etc. based application testing.
* Selenium supports very less CPU and RAM consumption for script execution.
* Selenium comes with different component to provide support to its parent which is Selenium IDE, Selenium Grid and Selenium Remote Control (RC).

Disadvantages:

* Selenium needs very much expertise resources. The resource should also be very well versed in framework architecture.
* Selenium only supports web based application and does not support windows based application.
* It is difficult to test Image based application.
* Selenium need outside support for report generation activity like dependence on TestNG or Jenkins.
* Selenium does not support built in add-ins support.
* Selenium user lacks online support for the problems they face.
* Selenium does not provide any built in IDE for script generation and it need other IDE like Eclipse for writing scripts.
* Selenium Automation Engineers are bit in scarcity these days.
* Selenium script creation time is bit high.
* Selenium does not support file upload facility.
* Selenium partially supports for Dialog boxes.

How we create Test Cases /Test Scripts / Tests in Selenium WebDriver:  
In UFT:  
Using Objects information and Test Methods we create Tests.  
In Selenium WebDriver:  
Using Element Locators and Webdriver Commands/methods we create Test Cases.  
Selenium IDE:  
Using Element locators and Selenese / Selenium IDE commands we create Test cases  
-----------------------------------  
Element Locators - to recognize elements/identify elements.  
WebDriver commands/methods - to perform operations on elements.  
--------------------------------------  
Java Programming - for enhancing Test Cases  
TestNG Framework - for grouping Test cases, batch Testing and generating Test Reports.  
-----------------------------------------------------------  
**ii) Selenium WebDriver Environment Setup.**  
Steps:  
1) Download and Install Java (JDK) software -to create programs (Test scripts)  
2) Set Environment variable (path variable).  
3) Download Eclipse IDE and extract - to write and execute Java programs.  
4) Download Webdriver Java language binding (www.seleniumhq.org) and add WebDriver jar files to Java project in Eclipse IDE.  
---------------------------------------  
5) Install Firebug and Firepath plug ins (Mozilla Firefox) for inspecting elements.  
6) For Internet Explorer and Google Chrome, no need to install any plug in, they have  
built in developer tools(F12) for inspecting elements.  
7) Firefox driver is default driver in Seelenium Webdriver, for IE and Chrome etc... Browsers then we need to download browser drivers.  
--------------------------------------------  
Download Selenium WebDriver Java language binding from www.seleniumhq.org website and extract.

Add WebDriver jar files to Java Project in Eclipse IDE  
Navigation:  
Create Java Project  
> Select Java project and right click  
> Build path  
> Configure build path  
> Select "Libraries" tab  
> Click "Add external Jars"  
> Browse path of the WebDriver jars.  
> Add  
-----------------------------------------  
Create Selenium WebDriver Test Case  
> Import Webdriver and Firefox/IE/Chrome Libraries. (In Selenium Test Case/Program)  
> Using Element locators and Webdriver commands write test steps.  
> Insert java programming statements to enhance Test cases.  
-----------  
> Using TestNG Annotations group test cases, execute test batches and generate detailed test reports.  
--------------------------------------------------  
**iii) Write first Selenium Test Case**  
Manual Test Case  
Test Case ID: gcrshop\_admin\_TC001  
  
Test Case Name: Verify Admin Login in GCR shop Web Portal  
  
Test Steps:  
  
1) Launch the Browser and navigate to "www.gcrit.com/build3/admin"  
2) Enter User name  
3) Enter Password  
4) Click Login Button  
  
Input data:  
Username = admin  
Password =admin@123  
  
Expected URL: "www.gcrit.com/build3/admin/index.php"  
Actual: http://www.gcrit.com/build3/admin/index.php  
  
Test Result: Pass  
-------------------------------------------  
\* Verification point: Capture the Browser URL after submission of Login details and compare with expected URL.  
----------------------------------------------------------------  
Selenium WebDriver Test Case:  
**package** gcREDDY;

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

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

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

**public** **class** AdminLogin {

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

System.*setProperty*("webdriver.chrome.driver", "E:\\SOFTWARE\\chromedriver.exe");

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

driver.get("http://www.gcrit.com/build3/admin/index.php");

driver.findElement(By.*name*("username")).sendKeys("admin");

driver.findElement(By.*name*("password")).sendKeys("admin@123");

driver.findElement(By.*id*("tdb1")).click();

String url = driver.getCurrentUrl();

**if**(url.equals("http://www.gcrit.com/build3/admin/index.php"))

{

System.***out***.println("Login successful - Passed");

}

**else**

System.***out***.println("Login Failed - Failed ");

driver.close();

}

**Web Elements and Element Locators**  
   
i) Web Elements  
ii) Element Locators  
---------------------------------------

**i) Web Elements**  
Browser  
Page  
-----------------  
Edit Box  
Link  
Button  
Image, Image Link, Image Button  
Text box  
Text Area  
Check box  
Radio Button  
Drop down box  
List box  
Combo box  
Web table /HTML table  
Frame  
-----------------------------------  
1) Operations on Browser  
> Launch the browser,  
> Navigate to particular web page,  
> Close focused Browser  
> Close all Browsers that opened by WebDriver during execution  
---------------  
> Navigate from one URL to another  
> Navigate back to previous URL  
> Navigate forward  
> Refresh the Browser  
> Maximize the Browser  
Etc...  
-----------------------------------  
2) Operations on Web Page  
> Get Page Title

> get page source  
> Get Page URL  
-----------------------------------  
3) Operations on Edit box  
  
> Enter a Value,  
> Clear the Value,  
> Check enabled status,  
> Check edit box existence,  
> Get the value etc...  
-----------------------------------  
4) Operations on Link  
  
> Click Link,  
> Check the link existence,  
> Check the link enabled status,  
> Return the Link Name  
Etc...  
-----------------------------------  
5) Operations on Button  
  
> Click  
> Check Enabled status  
> Display status  
Etc...  
-----------------------------------  
6) Operations Image  
  
Three types of Image elements in Web Environment  
  
a) General Image (No functionality)  
b) Image Button (Submits)  
c) Image Link (Redirects to another page/location)  
-----------------------------------  
7) Operations on Text Area  
  
> Return / Capture Text Area or Error message from a web page  
-----------------------------------  
8) Operations on Check box  
  
> Check if the check box is displayed or not?  
> Check if the check box is enabled or not?  
> Check if the check box is Selected or not?  
> Select the Check box  
> Unselect the Check box  
-----------------------------------  
9) Operations on Radio Button  
  
> Select Radio Button  
> Verify if the Radio Button is Displayed or not?  
> Verify if the Radio Button is enabled or not?  
> Verify if the Radio Button is Selected or not?  
------------------------------------  
10) Operations on Drop down box  
  
> Check the Drop down box existence  
> Check if the Drop down is enabled or not?  
> Select an item  
> Items Count  
-----------------------------------  
11) Operations on List box

> Select one or more items

> items count

> Check the existence

> check the enabled status

-----------------------------------  
12) Operations on Combo box (Drop down & Edit/Text box)

> select items

> Enter an item

> Check the existence

> check the enabled status  
  
13) Operations on Web table /HTML Table  
> Get cell value  
> Rows Count  
> Cells Count Etc...  
-----------------------------------  
14) Operations on Frame  
-----------------------------------  
> Switch from Top window to a frame  
> Switch from a frame to Top window  
Etc...  
----------------------------------------------  
**ii) Element Locators**  
what is Locator?

Object property value

Element locator value

Edit box name email

> Locator is an address that identifies a web element uniquely within the webpage. Locators are the HTML properties of a web element.

Selenium WebDriver uses 8 element locators to find elements on Web pages.  
  
id,   
name,   
className,   
tagName,   
linkText,   
partialLinkText,   
cssSelector,   
xpath   
  
Why we need to use different locators?  
  
1) Developers may not provide all locators for all elements  
2) some locators may be duplicated.  
So we have to choose any one unique locator to recognize the element.  
  
How to inspect elements?  
Download and install Firebug and Firepath plug ins/Add ons for Firefox Browser.   
  
If it Internet Explorer or Chrome, we no need to install any Add on, they provide built -in Developer Tools (F12) to inspect elements.  
  
Note: Element Locators are common for all Browsers.  
1) id  
  
Syntax:  
By.id("id value")  
Examples:  
driver.findElement(By.id("Email"))

driver- is Object  
findElement - WebDriver method  
By - pre-defined Class  
id - Element locater  
Email - id locator value  
-------------------------------------  
driver.findElement(By.id("Email")).sendKeys("gcrindia");  
--------------------------------------------  
Or  
  
WebElement Email = driver.findElement(By.id("Email"));  
Email.sendKeys("gcrindia");  
-------------------------------------------  
id locator for Button  
  
WebElement Email = driver.findElement(By.id("signIn"));  
Email.click();  
  
Or  
  
driver.findElement(By.id("signIn")).click();  
-------------------------------------------------------  
2) name  
  
Synatx:  
  
By.name("name value/locator name")         
  
Examples:  
  
driver.findElement(By.name("Email")).sendKeys("gcrindia");  
         
Or  
  
WebElement e = driver.findElement(By.name("Email"));  
e.sendKeys("gcrindia");  
------------------------------------------  
WebElement e = driver.findElement(By.name("signIn"));  
e.click();  
------------------------------------------  
3) className  
  
Syntax: By.className("class name value")  
  
Example:  
  
driver.findElement(By.className("textboxcolor")).sendKeys("Hyderabad");  
----------------------------------------------  
4) tagName  
  
Syntax:  
By.tagName("tag name value")  
  
Example:  
driver.findElement(By.tagName("input")).sendKeys("Hyderabad");  
--------------------------------------------------------------         
5) linkText  
  
Syntax:  
By.linkText("Link Text Value")  
  
Example:  
  
driver.findElement(By.linkText("Gmail")).click();  
-----------------------------------------------------                 
6) paritialLinkText  
  
Syntax:  
By.partialLinkText("Partial Link Text Value")  
  
Example:  
  
driver.findElement(By.partialLinkText("Gma")).click();  
-----------------------------------------------  
7) cssSelector  
Syntax:  
By.cssSelector("value")  
  
Example:  
driver.findElement(By.cssSelector(".gb\_m")).click();  
---------------------------------------------------  
8) xpath   
Xpath in XML document shows the direction of software web application's element location.  
  
Syntax:  
  
By.xpath("xpath value")  
  
driver.findElement(By.xpath(".//\*[@id='Email']")).sendKeys("abcdef");  
------------------------------------------------------------------------------

**Selenium WebDriver Commands and Operations**  
  
> Selenium WebDriver Methods are used to perform operations on Web Elements.  
  
> Using Element Locators and WebDriver Methods we create Test Cases.  
  
Element Locators - for recognizing Elements  
  
WebDriver Methods - for performing operations on Elements.  
-----------------------------------  
**WebDriver Methods**  
  
1) get()   
Description: Opens a specified URL in the Browser window.  
  
Syntax:   
  
driverObject.get("URL");  
  
Example:  
  
driver.get("https://www.google.co.in");  
------------------------------------  
2) getTitle()  
Returns Title of the Browser.  
  
Syntax:  
  
String variable = driver.getTitle();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String Title = driver.getTitle();  
System.out.println(Title);  
------------------------------  
3) getPageSource()  
Returns HTML page source.  
  
Syntax:  
  
String stringName = driver.getPageSource();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String pageSource = driver.getPageSource();  
System.out.println(pageSource);  
------------------------  
4) getCurrentUrl();  
Returns Current URL of the Browser.  
  
Syntax:  
  
String stringName = driver.getCurrentUrl();  
  
Example:  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
-------------------------------  
**Browser Navigation Methods**  
5) navigate().to();  
Loads a new web page in the current browser window.  
  
Syntax:  
driverObject.navigate().to("URL");  
  
Example:  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
-------------------------------------  
6) navigate().back()  
It moves a single item back in the Browser history.  
  
Syntax:  
  
driver.navigate().back();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().back();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
Or  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().to("https://www.google.co.in");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
-----------------------------------  
7) navigate().forward();  
It moves single item forward in the Browser history.  
  
Syntax:  
driver.navigate().forward();  
  
Example:  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().back();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().forward();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
Or  
  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://www.google.co.in");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
driver.navigate().to("https://login.yahoo.com/");  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
-----------------------------------  
8) navigate().refresh()  
Refresh the current web page  
  
Syntax:  
  
driver.navigate().refresh()  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.co.in");  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
driver.navigate().refresh();  
URL = driver.getCurrentUrl();  
System.out.println(URL);  
------------------------------------  
Method Syntax in Java  
  
Object.method();  
Object.property().method();  
Class.method();  
-----------------------------------  
9) close()  
It closes the focused Browser.  
  
Syntax:  
  
driverObject.close();  
  
Example:  
  
driver.get("https://www.google.co.in");  
driver.close();  
-----------------------------------  
10) quit()  
It closes all browser that opened by WebDriver during execution.  
  
Syntax:  
  
driverObject.quit();  
  
Example:   
  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/LoginPage.html");  
driver.findElement(By.linkText("Sign In")).click();  
driver.quit();  
-----------------------------------  
11) findElement()  
It finds the first element within the current page using the give locator.  
  
driver.findElement(By.ElementLocator("Value"))  
  
Syntax:  
WebDriver driver = new FirefoxDriver();  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/LoginPage.html");  
driver.findElement(By.tagName("input")).sendKeys("abcd");  
  
Or  
  
WebElement Email = driver.findElement(By.id("Email"));  
Email.sendKeys("India");  
-----------------------------------  
12) sendkeys()  
Enters a value into Edit box/Text box  
  
Syntax:  
driver.findElement(By.ElementLocator("value").sendkeys("input data");  
  
Example:  
  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("Email")).sendKeys("India");  
}  
-----------------------------------  
13) clear()  
It clears the value   
  
Syntax:  
driver.findElement(By.ElementLocator("value").clear();  
  
Example:  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("Email")).sendKeys("India");  
Thread.sleep(5000);  
driver.findElement(By.id("Email")).clear();  
-----------------------------------  
14) click()  
Clicks an Element (Buttons, Links)  
  
Syntax:  
  
driver.findElement(By.ElementLocator("value").click;  
  
Example:  
  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("next")).click();  
-----------------------------------  
15) isEnabled()  
It checks weather the Element is in enabled state or not?  
  
Syntax:  
  
boolean variableName = driver.findElement(By.ElementLocator("value").isEnabled();  
  
Example:  
  
driver.get("https://www.gmail.com");  
boolean a = driver.findElement(By.id("next")).isEnabled();  
System.out.println(a);  
-----------------------------------  
16) isDisplayed()  
Checks if the Element is displayed or not? in the current web page.

Syntax:  
  
boolean variableName = driver.findElement(By.ElementLocator("value").isDisplayed();  
  
driver.get("https://www.gmail.com");  
boolean a = driver.findElement(By.id("next")).isDisplayed();  
System.out.println(a);

Isenabled() vs isdisplaed()

**- Enabled or not , element is displayed/at all there or not  
-----------------------------------**  
17) isSelected()  
checks if the Element is Selected or not? in the current web page.  
  
Syntax:  
  
boolean variableName = driver.findElement(By.ElementLocator("value").isSelected();  
  
Example:  
  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/MultipleCheckbox.html");  
boolean a = driver.findElement(By.xpath("html/body/input[2]")).isSelected();  
System.out.println(a);//false  
driver.findElement(By.xpath("html/body/input[2]")).click();  
a = driver.findElement(By.xpath("html/body/input[2]")).isSelected();  
System.out.println(a);//true  
-----------------------------------  
18) manage().window().maximize()  
Syntax:  
  
driverObject.manage().window().maximize()  
  
Example:  
  
driver.get("file:///C:/Users/gcreddy/Desktop/HTMLExamples/MultipleCheckbox.html");  
Thread.sleep(5000);  
driver.manage().window().maximize();

// Below is how we can minimize the window.

driver.manage().window().setPosition(new Point(100, 100));  
-------------------------------------------

**Handling Elements in Selenium WebDriver**  
  
Pre-requisites to create Test cases in Selenium  
  
i) Element Locators (To recognize/identify Elements)  
  
ii) WebDriver Methods (To perform operations on Elements)  
  
iii) Programming features (To enhance Test cases)  
----------------------------------------------  
iv) TestNG Annotations (Grouping Test Cases, Test Batch execution and generating reports.)  
----------------------------------------------

**i) Element Locators**  
   
1) id  
2) name  
3) className  
4) tagName  
5) linkText  
6) parialLinkText  
7) cssSelector  
8) xpath  
------------------------------  
Web Elements  
   
Browser -driver object  
Page-----------  
Link  
Button  
Image, Image Button, Image Link  
Edit box  
Text Area  
Check box  
Radio Button  
Drop down box  
List box  
Combo box  
Web Table / HTML Table  
Frame etc...  
-----------------------------------------  
**ii) WebDriver Methods**  
   
**a) Methods on Browser**  
1) get()  
2) getTitle()  
3) getPageSource()  
4) getCurrentUrl()  
5) getwindowHandle()  
6) close()  
7) quit()  
--------------------  
**b) Browser navigation methods**  
1) navigate().to()  
2) navigate().back()  
3) navigate().forward()  
4) navigate().refresh()  
------------------------------  
**c) Methods on Elements**  
1) findElement()  
2) findElements()  
3) sendkeys()  
4) clear()  
5) click()  
6) isEnabled()  
  
7) isDisplayed()  
8) isSelected()  
9) getText()  
10) getAttribute()  
-----------------------------------  
**d) Others**  
1) manage.window.maximize()  
2) .explicitlyWait()  
-------------------------------------  
**iii) Java Programming features**  
  
**A) Java Fundamentals**  
   
1) Comments  
2) Data Types  
3) Modifiers  
4) Variables  
5) Operators  
6) Conditional Statements  
7) Loop Statements  
8) String handling  
9) Arrays in Java  
10) Built in Methods  
11) User defined Methods  
12) Input and Output Operations, File Handling  
13) Exception Handling  
  
**B) Java OOPS**  
   
1) Inheritance  
2) Polymorphism  
3) Abstraction   
4) Encapsulation  
------------------------------------  
**Handling Elements in Selenium**  
i) Handling Browser  
  
**Operations on Browser:**  
   
> Launch the Browser  
> Navigate to specified URL  
> Return Current URL  
> Get the Page Title  
> Return Page source  
> Return Window handle  
> Close focused Browser  
> Close all browsers that opened by selenium WebDriver during execution  
-----------  
> Navigate to another URL  
> Navigate back to previous URL  
> Navigate forward  
> Refresh the Browser  
> Maximize the Browser window.  
**Examples:**  
   
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.co.in");  
  
String PageTitle = driver.getTitle();  
System.out.println(PageTitle);  
  
String URL = driver.getCurrentUrl();  
System.out.println(URL);  
  
String PageSource = driver.getPageSource();  
System.out.println(PageSource);  
  
String WindowHandle = driver.getWindowHandle();  
System.out.println(WindowHandle);  
  
driver.close();  
driver.quit();  
----------------------------------------  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.co.in");  
driver.navigate().to("https://login.yahoo.com/");  
System.out.println(driver.getCurrentUrl());  
  
driver.navigate().back();  
System.out.println(driver.getCurrentUrl());  
  
driver.navigate().forward();  
System.out.println(driver.getCurrentUrl());  
  
driver.manage().window().maximize();  
--------------------------------------------  
ii) Handling Edit box  
  
**Operations on Edit box**  
   
> Enter a value  
> Clear the Value  
> Return the Value  
> Check Displayed status  
> Check Enabled statues  
  
**Example:**  
   
//Finding Edit box using id locator  
driver.findElement(By.id("Email")).sendKeys("gcrindia");  
  
//Finding Edit box using name locator  
driver.findElement(By.name("Email")).sendKeys("gcrindia");  
  
//Finding Edit box using CSS Selector locator  
driver.findElement(By.cssSelector("#Email")).sendKeys("gcrindia");  
  
//Finding Edit box using xpath locator  
driver.findElement(By.xpath(".//\*[@id='Email']")).sendKeys("gcrindia");  
  
//Clear the Value  
driver.findElement(By.xpath(".//\*[@id='Email']")).clear();  
--------------------------------  
WebElement Email = driver.findElement(By.xpath(".//\*[@id='Email']"));  
  
Email.sendKeys("gcrindia");  
  
//Return Type of the Object  
Email.getAttribute("type"); // You get a particular attribute value here. Like id value , name value  
  
//Return the Value  
System.out.println(Email.getText());//gcrindia  
  
//Return Displayed status  
System.out.println(Email.isDisplayed());//true  
  
//Return Enabled status  
System.out.println(Email.isEnabled());//true  
  
//Clear the value  
driver.findElement(By.id("Email")).clear();  
---------------------------------------------  
iii) Handle Text Area  
  
**Capture Text Area/Capture Error Message**  
  
**Capture Text Area:**  
   
driver.get("https://www.gmail.com");  
String s = driver.findElement(By.xpath("html/body/div[1]/div[2]/div[1]/h1")).getText();  
System.out.println(s);  
  
**Capture Error Message:**  
   
WebDriver driver = new FirefoxDriver();  
driver.get("https://login.yahoo.com/");  
driver.manage().window().maximize();  
driver.findElement(By.xpath(".//\*[@id='login-signin']")).click();  
String ErrorMessage = driver.findElement(By.id("mbr-login-error")).getText();   
System.out.println(ErrorMessage);  
  
**Handle Window Popup**  
   
WebDriver driver = new FirefoxDriver();  
driver.get("https://mail.rediff.com/cgi-bin/login.cgi");  
driver.findElement(By.name("proceed")).click();  
  
Alert alert = driver.switchTo().alert();  
String Error\_Message =alert.getText();//Returns Error message  
System.out.println(Error\_Message);  
  
alert.accept();//Closes OK Button  
driver.findElement(By.id("login1")).sendKeys("Inda123");  
---------------------------------  
iv) Handle Button  
   
> Click  
> Check the Displayed status  
> Check the Enabled status  
> Return name of the Object  
> Return type of the Object  
  
**Example:**  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.gmail.co.in");  
  
System.out.println(driver.findElement(By.id("next")).isEnabled());//true  
  
driver.findElement(By.id("next")).click();  
  
System.out.println(driver.findElement(By.id("next")).isDisplayed());//true  
  
System.out.println(driver.findElement(By.id("next")).isEnabled());//false  
  
System.out.println(driver.findElement(By.id("next")).getAttribute("type"));  
  
System.out.println(driver.findElement(By.id("next")).getAttribute("name"));  
  
System.out.println(driver.findElement(By.id("next")).getAttribute("value"));

---------------------------------------------------

**Handling Elements in Selenium Part-2**

In part-1 we discussed below:  
i) Handle Browser  
ii) Handle Edit box  
iii) Handle Text Area, Error Message, Window Dialog  
iv) handle Button  
-----------------------  
**Return/Capture value from Edit box**  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.gmail.com");  
driver.findElement(By.id("Email")).sendKeys("India123");  
System.out.println(driver.findElement(By.id("Email")).getAttribute("value"));  
}  
} //in case of only error message or edit box , use gettext() orelse use getAttribute()  
-----------------------  
v) Handle Image  
  
Three types of Image elements in Web Environment.  
  
1) General Image (No functionality)  
2) Image Button (Submits)  
3) Image Link (Directs to another page/location)  
  
Example:  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
System.out.println(driver.findElement(By.id("hplogo")).isDisplayed());//true  
System.out.println(driver.findElement(By.id("hplogo")).getAttribute("title"));  
  
driver.navigate().to("http://newtours.demoaut.com/");  
driver.findElement(By.name("login")).click();  
  
driver.navigate().to("http://www.seleniumhq.org/");  
driver.findElement(By.xpath(".//\*[@id='choice']/tbody/tr/td  
  
[2]/center/a/img")).click();  
  
}  
}  
-----------------------  
vi) Handle Link  
  
Operations on Link  
  
> Click  
> Check the Link Existence  
> Check Enabled status  
> Return Link Name etc...  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
//driver.findElement(By.className("gb\_P")).click();  
//driver.findElement(By.linkText("Gmail")).click();  
//driver.findElement(By.partialLinkText("mail")).click();  
//driver.findElement(By.cssSelector(".gb\_P")).click();  
//driver.findElement(By.xpath(".//\*[@id='gbw']/div/div/div[1]/div[1]/a")).click();  
WebElement Gmail\_Link = driver.findElement(By.xpath(".//\*  
  
[@id='gbw']/div/div/div[1]/div[1]/a"));  
boolean linkStatus = Gmail\_Link.isDisplayed();  
System.out.println(linkStatus);//true  
  
linkStatus = Gmail\_Link.isEnabled();  
System.out.println(linkStatus);//true  
  
String LinkName = Gmail\_Link.getText();  
System.out.println(LinkName);  
  
Gmail\_Link.click();  
}  
}  
-----------------------  
vii) Handle Radio Button  
  
Operations on Radio Button  
  
> Select  
> Check Enabled status  
> Check Displayed status  
> Check Selected status  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.gcrit.com/build3/create\_account.php?osCsid=47gtsrhe41613u5r3eqhgdbas7");  
WebElement maleRadioButton = driver.findElement(By.xpath(".//\*[@id='bodyContent']/form/div/div[2]/table/tbody/tr[1]/td[2]/input[1]"));  
boolean elementStatus = maleRadioButton.isDisplayed();  
System.out.println(elementStatus);//true  
  
System.out.println(maleRadioButton.isEnabled());//true  
  
System.out.println(maleRadioButton.isSelected());//false  
  
maleRadioButton.click();  
  
System.out.println(maleRadioButton.isSelected());//true  
}  
}  
-----------------------  
viii) Handle Drop Down box  
  
Operations on Drop Down Box  
  
> Select an Item  
> Check Displayed status  
> Check Enabled status  
> Items count  
  
Example:  
  
public static void main(String[] args) {  
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.gcrit.com/build3/create\_account.php?osCsid=47gtsrhe41613u5r3eqhgdbas7");  
**Select dropDown = new Select (driver.findElement(By.name("country")));**

//Without select we can’t select the items in the dropdown.  
//dropDown.selectByIndex(6);//Select an item by index  
//dropDown.selectByVisibleText("India"); //select by visible text  
  
**List<WebElement> e = dropDown.getOptions();  
int itemsCount = e.size();**  
System.out.println(itemsCount);  
}  
}  
-----------------------  
ix) Handle Check box  
  
Operations on Check box  
  
> Select  
> Unselect  
> Check Displayed status  
> Check Enabled status  
> Check selected status  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("file:///E:/HTMLExamples/MultipleCheckbox.html");  
System.out.println(driver.findElement(By.xpath("html/body/input[2]")).isDisplayed());//true  
  
System.out.println(driver.findElement(By.xpath("html/body/input[2]")).isEnabled());//true  
  
System.out.println(driver.findElement(By.xpath("html/body/input[2]")).isSelected());//false  
  
driver.findElement(By.xpath("html/body/input[2]")).click();  
  
System.out.println(driver.findElement(By.xpath("html/body/input[2]")).isSelected());//true  
  
driver.findElement(By.xpath("html/body/input[2]")).click();  
  
System.out.println(driver.findElement(By.xpath("html/body/input[2]")).isSelected());//false  
-----------------------  
x) Handle Web Table / HTML Table  
  
Operations on Web Table:  
  
> Get cell value  
> Rows Count  
> Cells Count  
  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("file:///E:/HTMLExamples/htmlTable.html");  
String s = driver.findElement(By.xpath(".//\*[@id='students']/tbody/tr[2]/td[2]")).getText();  
System.out.println(s);  
WebElement htmlTable = driver.findElement(By.id("students"));  
  
List <WebElement> rows = htmlTable.findElements(By.tagName("tr"));  
int r = rows.size();  
System.out.println(r);  
  
List <WebElement> cells = htmlTable.findElements(By.tagName("td"));  
int c = cells.size();  
System.out.println(c);  
-----------------------  
xi) Handle inline Elements  
  
The span tag is used to group inline Elements in a Document.  
  
Example 1:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
driver.findElement(By.xpath(".//\*[@id='gbwa']/div[1]/a")).click();  
driver.findElement(By.xpath(".//\*[@id='gb36']/span[1]")).click();  
driver.navigate().back();  
-----------------------  
Example 2:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
driver.manage().window().maximize();  
driver.findElement(By.xpath(".//\*[@id='gbwa']/div[1]/a")).click(); // clicking on the Google apps  
driver.findElement(By.xpath(".//\*[@id='gbwa']/div[2]/a[1]")).click(); // clicking on more  
driver.findElement(By.xpath(".//\*[@id='gb300']/span[1]")).click(); // clicking on hangout  
-----------------------  
Handle Frames  
Handle Mouse Hover  
Working with Multiple browser windows.  
-----------------------

**Handling Elements in Selenium Part-3**  
  
**In Handling Elements in Selenium Part-1**  
  
i) Handle Browser  
ii) Handle Edit box  
iii) Handle Text Area, Error Message, Window Dialog  
iv) Handle Button  
-----------------------  
**In Handling Elements in Selenium Part-2**  
  
v) Handle Images (General Image, Image Button and Image Link)  
vi) Handle Link  
vii) Handle Radio Button  
viii) Handle Drop down box  
ix) Handle Check box  
x) Handle Web Table / HTML Table  
xi) Handling inline Elements  
-----------------------  
**Handling Elements in Selenium Part-3**  
  
xii) Handle Frames  
  
> HTML frames are used to divide the Browser window into multiple sections, where each section can load a separate HTML document.  
> Frames are sections of Web page displayed on top window.  
> Whenever we access the page then focus on the top window.  
  
**Switch to a frame is done in two ways**  
  
**1) Using frame index**  
  
Syntax:  
driver.swithchTo().frame(int index);  
  
Example:  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
driver.switchTo().frame(2);  
driver.findElement(By.xpath("html/body/div[3]/table/tbody[2]/tr[1]/td[1]/a")).click();  
  
**2) Using frame name**  
  
Syntax:  
driver.switchTo().frame(String frame Name);  
  
Example:  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
driver.switchTo().frame("classFrame");  
driver.findElement(By.xpath("html/body/div[3]/table/tbody[2]/tr[1]/td[1]/a")).click();  
-----------------------  
> Top window to a Frame (frame index/frame name)  
> Switch from a frame to Top window  
Syntax:  
  
driver.switchTo().defaultContent();  
---------------------  
Navigation:  
  
> Launch the page   
> Switch to 3rd frame  
> Operate an element  
> Back to Top window  
> Switch to 1st frame  
> Operate an element  
--------------------------  
Example:  
  
WebDriver driver = new FirefoxDriver();  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
  
//Switch to 3rd frame  
driver.switchTo().frame(2);  
driver.findElement(By.linkText("com.thoughtworks.selenium")).click();  
Thread.sleep(3000);  
//Switch from 3rd frame to Top window  
driver.switchTo().defaultContent();  
Thread.sleep(3000);  
//Switch to 1st frame  
driver.switchTo().frame(0);  
driver.findElement(By.linkText("org.openqa.selenium")).click();  
-----------------------  
WebDriver driver = new FirefoxDriver();  
driver.get("http://seleniumhq.github.io/selenium/docs/api/java/index.html");  
Thread.sleep(3000);  
//Switch to 3rd frame  
driver.switchTo().frame("classFrame");  
driver.findElement(By.linkText("com.thoughtworks.selenium")).click();  
Thread.sleep(3000);  
//Switch from 3rd frame to Top window  
driver.switchTo().defaultContent();  
Thread.sleep(3000);  
//Switch to 1st frame  
driver.switchTo().frame("packageListFrame");  
driver.findElement(By.linkText("org.openqa.selenium")).click();  
}  
}  
-----------------------  
xiii) Handle Mouse hover  
   
WebDriver driver = new FirefoxDriver();  
driver.get("http://www.carmax.com/");  
//create Action builder instance by passing WebDriver instance  
Actions builder = new Actions(driver);  
WebElement menuElement = driver.findElement(By.linkText("Sell Us Your Car"));  
builder.moveToElement(menuElement).build().perform();  
driver.findElement(By.linkText("FAQ")).click();  
-----------------------  
xiv) Handle Multiple Browsers  
  
String parent = driver.getWindowHandle();  
//System.out.println(parent);  
  
Set <String> Handles = driver.getWindowHandles();  
int BrowserCount = Handles.size();  
System.out.println(BrowserCount);  
  
for (String s1:Handles){  
if (! s1.equals(parent)){  
driver.switchTo().window(s1);  
System.out.println(driver.getCurrentUrl());  
}  
}  
driver.switchTo().window(parent);  
System.out.println(driver.getCurrentUrl());  
-----------------------  
xv) Handle Duplicate objects  
  
Redirecting to GCR Shop User Interface from Admin Interface  
(After Login to Admin Interface then Redirect to User Interface)  
  
Example:  
  
driver.get("http://www.gcrit.com/build3/admin/");  
driver.findElement(By.name("username")).sendKeys("admina");  
driver.findElement(By.name("password")).sendKeys("admin@123");  
driver.findElement(By.id("tdb1")).click();  
String url = driver.getCurrentUrl();  
if (url.equals("http://www.gcrit.com/build3/admin/index.php")){  
driver.findElement(By.linkText("Online Catalog")).click();  
}  
System.out.println(driver.getCurrentUrl());  
}  
}  
-----------------------  
**Assignment:**  
  
Enter Password in to 2nd Password Edit box in www.infibeam.com Registration page.  
-----------------------

**Cross Browser Testing with Selenium**  
  
1) What is Cross Browser Testing?  
  
Cross Browser Testing is a type of Test to check that our Web Application works as expected in different Browsers.  
  
2) Why Cross Browser Testing?  
A web application can be opened in any web browser by the end user, So we need to ensure that the web application will work as expected in all popular browsers.  
  
3) Popular Web Browsers  
a) Google Chrome: It was released in 2008, its market share approximately 68%  
b) Mozilla Firefox: It was released in 2004, its market share approximately 19%  
  
c) Internet Explorer: It was released in 1995, its market share approximately 6.5%  
  
4) Working with different Browsers  
  
> Selenium WebDriver supports Browser compatibility tests on almost every popular browser, including Chrome, Firefox, IE, Opera and Safari.  
> The WebDriver API drives the web browser as the real user would drive it.  
> By default, Firefox driver comes with selenium-serverstanalone.jar library added.  
> For Chrome, IE, Safari, Opera, there are libraries that need to be instantiated externally.  
  
5) How to conduct Cross Browser Testing using Selenium WebDriver?  
  
a) Element Locators - Same for all Browsers.  
b) WebDriver Methods/Commands -Same for all Browsers.  
c) Programming features (Java/C#/Python/Perl/Ruby/PHP) - Same for all Browsers.  
d) JUnit / TestNG Annotations - Same for all Browsers.  
e) Browser Driver - various from one browser to another.  
  
Note: For Mozilla Firefox, just create the driver, For other browsers, libraries that need to be instantiated externally.  
------------------------  
f) Inspect Elements -  
  
For Mozilla Firefox -Built in feature Page Inspector,  
(Install Firebug and Firepath)  
For Chrome and IE - Built in Developer tools  
  
6) Create Browser Drivers  
   
(For Google Chrome, IE and Other Browsers, download Browser drivers and set   
path in Selenium Test Scripts)  
a) Mozilla Firefox Browser:  
WebDriver driverName = new FirefoxDriver();  
b) Google Chrome  
//Instantiate Chrome Browser driver  
  
System.setproperty("webdriver.chrome.driver", "driver .exe file path");  
WebDriver driverName = new ChromeDriver();  
  
c) IE Browser driver  
  
System.setproperty("webdriver.ie.driver", "driver .exe file path");  
WebDriver driverName = new InternetExplorerDriver();  
  
7) Create a Test Case and Execute using Mozilla Firefox, Chrome and IE Browsers.  
  
**Test Case:** Verify Launch Application (Google) functionality in Firefox, Chrome and IE Browsers.  
  
**Test Steps:**  
   
i) Launch the Browser  
ii) Navigate to https://www.google.com url  
  
**Verification point:**  
Capture the page Title (Actual) and Compare with Expected.  
  
**Expected Page Title:** Google  
-------------------------------  
**a) Test Case for Mozilla Firefox Browser**  
  
WebDriver driver = new FirefoxDriver();  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println("Google Application Launched - Passed");  
}  
else {  
System.out.println("Google Application Not Launched -Failed");      
}  
driver.close();  
------------------------------  
**b) Test Case for Google Chrome Browser**  
   
System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");  
WebDriver driver = new ChromeDriver();  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println("Google Application Launched - Passed");  
}  
else {  
System.out.println("Google Application Not Launched -Failed");      
}  
driver.close();  
------------------------------  
**c) Test Case for internet Explorer Browser**  
  
System.setProperty("webdriver.ie.driver", "E:\\IEDriverServer.exe");  
WebDriver driver = new InternetExplorerDriver();  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println("Google Application Launched - Passed");  
}  
else {  
System.out.println("Google Application Not Launched -Failed");      
}  
driver.close();  
------------------------------  
8) Create a Test Case and Execute using Mozilla Firefox, Chrome and IE Browsers Continuously.  
  
public class TestCase1 {  
public static WebDriver driver;  
public static int browser;  
public static String BrowserName;  
  
public static void main(String[] args) {  
  
for (browser = 1; browser <= 3; browser++){  
if (browser == 1) {  
driver = new FirefoxDriver();  
BrowserName = "Mozilla Firefox Browser: ";  
}  
else if (browser == 2) {  
System.setProperty("webdriver.chrome.driver", "E:\\chromedriver.exe");  
driver = new ChromeDriver();  
BrowserName = "Google Chrome Browser: ";  
}  
else if (browser == 3){  
System.setProperty("webdriver.ie.driver", "E:\\IEDriverServer.exe");  
driver = new InternetExplorerDriver();  
BrowserName = "Internet Explorer Browser: ";  
}  
driver.get("https://www.google.com");  
  
String PageTitle = driver.getTitle();  
  
if (PageTitle.equals("Google")){  
System.out.println(BrowserName + " - Google Application Launched - Passed");  
}  
else {  
System.out.println(BrowserName + " - Google Application Not Launched -  
  
Failed");      
}  
driver.close();  
}  
}  
}  
------------------------

Created on – 07/05/2018

Revision Completed on – 13/04/2020