**Selenium Understanding Document**

**Selenium Sample Commands:**

**Command                                               Description**  
**driver.get("http://www.google.com");**  To open an application  
**driver.findElement(By.id("passwd-id"));**  Finding Element using Id  
**driver.findElement(By.name("passwd"));**  Finding Element using Name  
**driver.findElement(By.xpath("//input[@id=’passwd-id’]"));**  Finding Element using Xpath  
**element.sendKeys("some text");**  To type some data  
**element.clear();**  clear the contents of a text ﬁeld or textarea  
**driver.findElement(By.xpath("//select"));**  Selecting the value  
**select.findElements(By.tagName("option"));**  Selecting the value  
**select.deselectAll();**  This will deselect all OPTIONs from the ﬁrst SELECT on the page  
**select.selectByVisibleText("Edam");**  select the OPTION withthe displayed text of “Edam”  
**findElement(By.id("submit")).click();**  To click on Any button/Link  
**driver.switchTo().window("windowName");**  Moving from one window to another window  
**driver.switchTo().frame("frameName");**  swing from frame to frame (or into iframes)  
**driver.switchTo().frame("frameName.0.child");**  to access subframes by separating the path with a dot, and you can specify the frame by its index too.  
**driver.switchTo().alert();**  Handling Alerts  
**driver.navigate().to("http://www.example.com");**  To Navigate Particular URL  
**driver.navigate().forward();**  To Navigate Forward  
**driver.navigate().back();**  To Navigate Backword  
**driver.close()**  Closes the current window  
**driver.quit()**   Quits the driver and closes every associated window.  
**driver.switch\_to\_alert()**  Switches focus to an alert on the page.  
**driver.refresh()**  Refreshes the current page.  
**driver.implicitly\_wait(30)**  Amount of time to wait  
**driver.set\_script\_timeout(30)**  The amount of time to wait  
**driver.get\_screenshot\_as\_file('/Screenshots/foo.png')**  The full path you wish to save your screenshot to  
**driver.get\_screenshot\_as\_base64()**  Gets the screenshot of the current window as a base64 encoded string which is useful in embedded images in HTML

**Simple Launch Firefox browser:**

WebDriver webDriver = new FirefoxDriver();

// Setting the browser size

webDriver.manage().window().setSize(new Dimension(1024, 768));

// Go to wikipedia

webDriver.navigate().to("https://en.wikipedia.org/wiki/Main\_Page");

// Type in the search-field: "WebDriver"

webDriver.findElement(By.id("searchInput")).sendKeys("WebDriver");

// submitting the search query

webDriver.findElement(By.id("searchInput")).submit();

// Test if Wikipedia redirects to the correct article:// "Selenium (software)"

String textFound = webDriver.findElement(By.cssSelector("h1")).getText();

if (textFound.contains("Selenium (software)")) {

System.out.println("Test passes!");

} else {

System.out.println("Test fails!");

}

// Waiting a little bit before closing

Thread.sleep(7000);

// Closing the browser and webdriver

webDriver.close();

webDriver.quit();

}}

**NAVIGATION COMMANDS:**

1. **Browser Back and Forward (NAVIGATION)**

Steps to implement Browser back and forward through Selenium Web Driver

1. Create Driver for any Browser(Mozilla)

2. Go to the URL

3. Navigate to some page in website.

4. Use Selenium code to Navigate Back to Main Page.

**CODE:**

driver.navigate().back();

driver.navigate().forward();

**Example**

WebDriver driver =new FirefoxDriver();

driver.get("http://seleniumhq.org/");

driver.findElement(By.linkText("Download")).click();

Thread.sleep(3000);            //delay

driver.navigate().back();

driver.navigate().forward();

**2.Single selection dropdown list**

**WebElement element = driver.findElement(By.name("selectedCustomer"));  
Select dd= new Select(element);  
List allOptions= dd.getOptions();**  
//To go through the list, we can use an Iterator.   
//Iterator should be of the same type as the List  
//which is WebElement in this case.   
  
**Iterator it = allOptions.iterator();**  
//Using while loop, we can iterate till the List has   
//a next WebElement [hasNext() is true]  
//number of items in the list  
**System.out.println(allOptions.size());**  
  
**while(it.hasNext()){**//When you say it.next(), it points to a particular  
//WebElement in the List.  
**WebElement el = it.next();** //Check for the required element by Text and click it  
**if(el.getText().equals("mango")){  
  System.out.println(el.getAttribute("value"));  
   el.click();   
 }**}

**WebElement customerdd =driver.findElement(By.name("customerProject.shownCustomer"));**

**//convert the element to select object**

Select cust = new Select(**customerdd**);

**cust.selectByIndex(1);**                                       //**Select by** **Index**

Thread.sleep(3000);

**cust.selectByValue("2");                                   //Select by Value**

Thread.sleep(3000);

**cust.selectByVisibleText("mango");                //Select by Visible Text**

**Multiple select**

WebElement userdd = driver.findElement(By.name("users"));  
**Select usr = new Select(userdd);  
usr.selectByIndex(0);                     //Select by Index(From Start location)  
usr.selectByIndex(2);                     //Select by index(To End Location)**

**Deselect All:**

//You can deselect the options  
usr.**deselectAll**();                                          //Deselect ALL selected elements  
usr.deselectByIndex(0);                              //Deselect By using Index  
usr.deselectByValue(value);                       //Deselect By using Value  
usr.deselectByVisibleText(text);                 //Deselect By using Text  
-----------------------------------------------------------------------------------------------------------------------------

**IFRAMES - How to handle Frames in Web Driver**

|  |
| --- |
| ChromeOptions options = new ChromeOptions();  options.addArguments("test-type");  options.addArguments("start-maximized");  options.addArguments("--js-flags=--expose-gc");  options.addArguments("--enable-precise-memory-info");  options.addArguments("--disable-popup-blocking");  options.addArguments("--disable-default-apps");  options.addArguments("test-type=browser");  options.addArguments("disable-infobars");  driver = new ChromeDriver(options);  driver.manage().window().maximize();  driver.get("http://timesofindia.indiatimes.com/");  driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  List forms= driver.findElements(By.tagName("iframe")); //Frame List  System.out.println(forms.size());  for(int i=0;i<forms.size();i++)  {  System.out.println(forms.get(i).equals(driver.getPageSource().contains("src")));  } |

**To Perform actions in frames:**

WebElement ifr = driver.findElement(By.xpath("//iframe[@src='/poll.cms']"));

**driver.switchTo().frame(ifr);**                             **//Switch to iFrame**driver.findElement(By.id("test")).sendKeys("8");  //Perform Action in iFrame

**Steps to switch to particular iFrame by index through Selenium Web Driver.**

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
**List forms= driver.findElements(By.tagName("iframe"));**System.out.println(forms.size());  
**driver.switchTo().frame(0);**driver.findElement(By.id("clicktripad")).click();

**Steps to iterate through the Window Handles**

1. Create Driver for any Browser(Mozilla)

2. Go to the URL  
3. Collect Window Handles through Set  
4. Create an iterator to iterate through Window Handles.  
5. At First iterator will not be pointing to any Window Handle, only First increment Points to First Window Handle, Second increment Points to second iterator.  
Set windowHandles = driver.getWindowHandles();  
Iterator it = windowHandles.iterator();  
while(it.hasNext())  
{  
 System.out.println(it.next());

}

**When two browsers are opened and Web Driver need to shift the control from Parent Window to Child Window.**

1. Create Driver for any Browser (Mozilla)

2. Go to the URL  
3. Collect Window Handles through Set  
4. Create an iterator to iterate through Window Handles.  
5. Increment the iterator and store the Window Handle as Parent.  
6. Increment the iterator and store next Window Handle as Child.  
7. Switch to Child Browser using Child Window Handle.  
Set windowHandles = driver.getWindowHandles();  
Iterator it = windowHandles.iterator();  
  
String parentBrowser= it.next();  
String childBrowser = it.next();  
driver.switchTo().window(childBrowser);

**Close the pop up window and switch to parent window**

Set windowHandles = driver.getWindowHandles();  
Iterator it = windowHandles.iterator();  
String parentBrowser= it.next();  
String childBrowser = it.next();  
driver.switchTo().window(childBrowser);   
Thread.sleep(3000);  
driver.close(); //close the current window(Child Browser)  
driver.switchTo().window(parentBrowser); //Switch to Parent Browser

**Calendar PopUp - 1**

Normal Calendar (current month) Popup can be handled in the following way.

/\*IRCTC calendar\*/  
driver.findElement(By.id("calendar\_icon1")).click();   
driver.findElement(By.xpath("//div[@id='CalendarControl']/table[tbody[tr[td[text()='October 2012']]]]/descendant::a[text()='5']")).click();

**/\*makemytrip calendar\*/**driver.get("http://www.makemytrip.com/");  
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
driver.findElement(By.id("deptDateRtripimgExact")).click(); //find Calendar  
driver.manage().timeouts().implicitlyWait(0, TimeUnit.SECONDS);  
boolean flag=true;  
while(flag){  
  try {  
 WebElement el = driver.findElement(By.xpath("//div[contains(@class,'ui-datepicker-group') and descendant::span[text()='March']]/descendant::a[text()='5']")); // Required future date  
 if(el !=null)   //Check if the required date element is found or not  
       {  
 el.click(); // if required Date is found, then click  the date  
 flag=false;  
 }  
    }   
catch (Exception e) { //Catches exception if no element found  
try {  
 Thread.sleep(500);  
 driver.findElement(By.xpath("//a[@title='Next']")).click(); //Click on next month  
 }  
catch (InterruptedException e1)   
      {  
 // TODO Auto-generated catch block  
  e1.printStackTrace();  
      }  
**Drop Down Menu:**

WebElement parentMenu = driver.findElement(By.linkText("Tourist Trains"));

Actions act = new Actions(driver); // Create an Action object  
//move to the parent menu item

act.moveToElement(parentMenu).build().perform();

Thread.sleep(3000);   //wait till the child items are displayed

driver.findElement(By.linkText("Bharat Tirth")).click();

### Context Click (Right Click)

WebElement parentMenu = driver.findElement(By.linkText("Tourist Trains"));  
Actions act = new Actions(driver); //Create Action object for Driver  
act.contextClick(parentMenu).build().perform(); //Context Click  
act.sendKeys(Keys.ARROW\_RIGHT).build().perform();  
Thread.sleep(1000);  
act.sendKeys(Keys.ARROW\_DOWN).build().perform();  
Thread.sleep(1000);  
act.sendKeys(Keys.ENTER).build().perform();

**Use JavaScript to perform some actions**

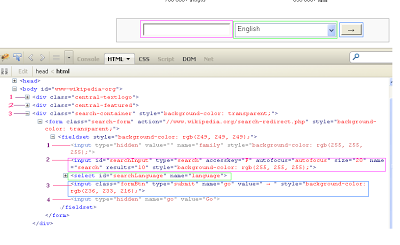
JavascriptExecutor js = (JavascriptExecutor) driver;  
String jsCmd = "document.getElementsByName('city')[0].value='ban'";  
js.executeScript(jsCmd);

**Use Alerts**

WebDriver driver = new FirefoxDriver();  
driver.get("http://www.2shared.com/");  
driver.manage().timeouts().implicitlyWait(3,TimeUnit.MINUTES);  
  
Alert alert = driver.switchTo().alert();  
alert.accept();  
//or  
alert.dismiss();

**Xpath**

Xpath in XML document shows the direction of element location through nodes and attributes. Let we try to understand how to identify Xpath of element with examples.

[](http://2.bp.blogspot.com/-9T8oQ-2SPbA/UbfoKtQdYFI/AAAAAAAAAGo/fsixYo0R5ss/s1600/Xpath.PNG)

Above given image is taken from http://www.wikipedia.org/. Look into the image there are three fields 1. Input text box 2. select drop down and 3. input button. And bellow of those fields there is expansion of relative XML nodes through firebug. As you see in image, you can use "id=searchInput" or "name=search" to identify input text box to type something in to it as bellow given example.

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| type | id=searchInput | ID Example |

or

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| type | name=search | Name Example |

**Xpath Tutorials**

Now if you want to identify same element (input textbox) with xpath then you can use any of the bellow given syntax in to the target column with type command in above example.

**Locating element using Xpath with Examples for input text box**

**1. Identifying Xpath using full path of XML**

**xpath=//body/div[3]/form/fieldset/input[2]**  //// Here //body is the main root node, /div[3] describes the 3rd div child node of parent node body, /form describes the child node form of parent node div[3], /fieldset describes the child node fieldset of parent node form, /input[2] describes the 2nd input child node of parent node fieldset.

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| type | xpath=//body/div[3]/form/fieldset/input[2] | Xpath Example1 |

**2. Writting Xpath using last()**

**xpath=//body/div[3]/form/fieldset/input[last()-2]**  //// Here /input[last()-2] describes the 3rd upper input node(input[2]) from last input node.  
**xpath=//body/div[3]/form/fieldset/\*[last()-3]**//// Here /\*[last()-3] describes the 4th upper  node(input[2]) from last node.

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| type | xpath=//body/div[3]/form/fieldset/input[last()-2] | Xpath Example2 |

**3. Xpath locator using @ and attribute**

**xpath=//body/div[3]/form/fieldset/input[@type='search']**   //// Here /input[@type='search'] describes the input node having attribute type='search'.

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| type | xpath=//body/div[3]/form/fieldset/input[@type='search'] | Xpath Example3 |

**4. Xpath expression using @ and attribute**

**xpath=//body/div[3]/form/fieldset/input[@accesskey='F']**   //// Here /input[@accesskey='F'] describes the input node having attribute @accesskey='F'. Another way of same is as bellow.

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| type | xpath=//body/div[3]/form/fieldset/input[@accesskey='F'] | Xpath Example4 |

**5. Xpath syntax using @ and attribute**

**xpath=//input[@accesskey='F']**  //// Here //input[@accesskey='F'] describes the input node having attribute @accesskey='F'. Try it by using it in above example.

**6. Xpath example using @ and attribute**

**xpath=//input[@type='search']**   //// Here /input[@type='search'] describes the input node having attribute type='search'. Try it by using it in above example.

**7. XML Xpath using /descendant:: keyword**

**xpath=//div[@class='search-container']/descendant::input[@accesskey='F']**   //// Here i have used descendant in between. In this case i have described only starting node div with attribute class='search-container' and final node input with accesskey='F' attribute. So not need to describe in between nodes. Try it by using it in above example.

**8. Xpath query example using contains keyword**

**xpath=//input[contains(@id, "searchInput")]**   ////Here i have used contains keyword to identify id attribute with text "searchInput". Try it by using it in above example.

**9. xpath using and with attributes**

**xpath=//input[contains(@id, "searchInput") and contains(@accesskey,"F")]**   ////In this example, It will look at two attributes in input node. Try it by using it in above example.

**10. XML xpath value value using position()**

**xpath=//div[@class='search-container']/descendant::input[position()=2]**  ////This xpath will select input node which is on number 2 position and it is for input text box as shown in image. Try it by using it in above example.

**11. Using starts-with keyword**  
**xpath=//input[starts-with(@type, "s")]    ////**In this example, It will find input node with attribute is 'type' and its value is starting with 's' (here it will get type = 'search').

**12. Using OR (|) condition with xpath**  
**xpath=//input[@accesskey='F'] | //input[@id='searchInput']**  
**xpath=//input[@accesskey='F' or @id='searchInput']**  //// In both these example, it will find input text box with accesskey='F' or @id='searchInput'. If any one found then it will locate it. Very useful when elements appears alternatively.  
  
**13. Using wildcard \* with to finding element xpath**  
**xpath=//\*[@accesskey='F']**  
  
**14. Finding nth child element of parent**  
**xpath=//body/\*[3]/form/fieldset/\*[2]**    ////This xpath is for search text box. Here, /\*[3] describes the 3rd child element of body which is div[3]. Same way \*[2] describes the 2nd child element of fieldset which is input[2]  
  
All above examples are for input text box. Now let me write Xpath for drop down.

**Xpath Examples for drop down**

**1. xpath=//body/div[3]/form/fieldset/select**

**2. xpath=//body/div[3]/form/fieldset/select[last()]**

**3. xpath=//body/div[3]/form/fieldset/select[@id='searchLanguage']**

**4. xpath=//body/div[3]/form/fieldset/select[@name='language']**

**5. xpath=//div[@class='search-container']/descendant::select[@name='language']**

**6. xpath=//select[contains(@id, "searchLanguage")]**

**7. xpath=//div[@class='search-container']/descendant::select[position()=1]**  
**8. xpath=//body/div[3]/form/fieldset/select[count(\*)>1]**

| New Test | | |
| --- | --- | --- |
| **Command** | **Target** | **Value** |
| open | http://www.wikipedia.org/ |  |
| select | xpath=//div[@class='search-container']/descendant::select[position()=1] | label=English |

**Other Xpath Example**

**1. Finding xpath for target link 'url'**  
**//a[@href='//meta.wikimedia.org/wiki/List\_of\_Wikipedias']**////This xpath example will find link with given URL (//meta.wikimedia.org/wiki/List\_of\_Wikipedias) on the page.  
  
**2. Finding xpath of element with no child**  
**xpath=//img[count(\*)=0]**////This xpath is for wikipedia text logo which is display on top of the page. This xpath will find that image element which have not any child element. Here image node is last and it has not any child element.  
  
**xpath=//div[2]/descendant::img[count(\*)=0]   ////**This xpath is for wikipedia logo image which is display under logo text.

**We can count the number of links present in the page. We can also print the link text of each Web link.**

List allLinks= driver.findElements(By.xpath("//a"));  
//display the count of links in the page  
System.out.println(allLinks.size());  
//display the text for each link on the page  
for(int i=0;i< allLinks.size();i++)  
{ //display href for each link  
 System.out.println(allLinks.get(i).getAttribute("href"));  
 //display text for each link  
 System.out.println(allLinks.get(i).getText());  
 allLinks.get(i).click();  
}

**Proxy Settings:**

1. Import Selenium.Proxy

2. Create a Profile object for Firefox

3. Create a string variable with value.

4. Create a Proxy object.

5. Set the values through proxy.

6. Set the proxy preference to proxy object using profile object.

7. Pass the profile object to Firefox Driver.

import org.openqa.Selenium.Proxy  
FirefoxProfile profile = new FirefoxProfile();  
String PROXY = "xx.xx.xx.xx:xx";  
Proxy proxy = new Proxy();  
proxy.HttpProxy=PROXY;  
proxy.FtpProxy=PROXY;  
proxy.SslProxy=PROXY;  
profile.SetProxyPreferences(proxy);  
FirefoxDriver driver = new FirefoxDriver(profile);

**File Upload**

WebDriver driver = new FirefoxDriver();  
driver.get("http://www.2shared.com/");  
String FilePath = "C:\\Users\\abc\\Desktop\\test.xml";  
driver.findElement(By.id("upField")).sendKeys(FilePath);  
driver.findElement(By.xpath("//input[@type='image']")).click();

**File Download**

FirefoxProfile Prof = new FirefoxProfile();  
Prof.setPreference("browser.download.dir", "D:\\java prj");  
Prof.setPreference("browser.download.folderList", 2);  
Prof.setPreference("browser.helperApps.neverAsk.saveToDisk","application/zip");  
   
WebDriver driver = new FirefoxDriver(Prof);  
driver.get("http://seleniumhq.org/download/");  
driver.manage().timeouts().implicitlyWait(3,TimeUnit.MINUTES);  
driver.findElement(By.xpath("//a[@name='client-drivers']/table/tbody/tr[1]/td[4]/a")).click();

**Web Driver with Excel**import java.io.FileInputStream;  
import jxl.Sheet;  
import jxl.Workbook;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.testng.annotations.AfterClass;  
import org.testng.annotations.BeforeClass;  
import org.testng.annotations.Test;  
public class WebDriverExcel {  
private WebDriver driver;  
@BeforeClass  
public void Startup(){  
driver = new FirefoxDriver();  
}  
@Test (description="login")  
public void Login() throws Exception{  
FileInputStream fi=new FileInputStream("D:\\Selenium\\Test.xls");  
Workbook w=Workbook.getWorkbook(fi);  
Sheet s=w.getSheet(0);  
driver.get("http://127.0.0.1/login");  
try  
{  
for (int i = 1; i < s.getRows(); i++)  
{  
//Read data from excel sheet  
String s1 = s.getCell(0,i).getContents();  
String s2 = s.getCell(1,i).getContents();  
driver.findElement(By.name("txtUserName")).sendKeys(s1);  
driver.findElement(By.name("txtPassword")).sendKeys(s2);  
driver.findElement(By.name("Submit")).click();  
Thread.sleep(3000);  
if(driver.getTitle().equals("OrangeHRM"))  
{  
System.out.println("admin page Displayed successfully");  
}  
else  
{  
System.out.println("Do not Displayed");  
}  
driver.findElement(By.linkText("Logout"));  
}  
}  
catch(Exception e)  
{  
System.out.println(e);  
}  
}  
@AfterClass  
public void teardown(){  
driver.quit();  
}  
}

**Sample DB connection Snippet:**

import java.sql.\*;

import javax.sql.\*;

public class dbconnection

{

public static void main(String args[])

{

String email;

String dbUrl = "jdbc:mysql://localhost:3306/test";  //This URL is based on your IP address

String username="username"; //Default username is root

String password="password"; //Default password is root

String dbClass = "com.mysql.jdbc.Driver";

String query = "Select email from users where user\_id = 1;";

try

{

Class.forName(dbClass);

Connection con = DriverManager.getConnection (dbUrl,username,password);

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(query);

while (rs.next())

{

dbtime = rs.getString(1);

System.out.println(email);

} //end while

con.close();

} //end try

catch(ClassNotFoundException e)

{

e.printStackTrace();

}

catch(SQLException e)

{

e.printStackTrace();

}

}  //end main

}  //end class

**Sample Practices 🡪Gmail automation**

|  |
| --- |
| System.*setProperty*("webdriver.chrome.driver","C://Users/sangee/Downloads/chromedriver\_win32 (1)/chromedriver.exe");  ChromeOptions options = **new** ChromeOptions();  options.addArguments("test-type");  options.addArguments("start-maximized");  options.addArguments("--js-flags=--expose-gc");  options.addArguments("--enable-precise-memory-info");  options.addArguments("--disable-popup-blocking");  options.addArguments("--disable-default-apps");  options.addArguments("test-type=browser");  options.addArguments("disable-infobars");  *driver* = **new** ChromeDriver(options);  *driver*.manage().window().maximize();  *driver*.get("https://accounts.google.com/ServiceLogin?");  // gmail login  *driver*.findElement(By.*id*("identifierId")).sendKeys("\*\*\*\*\*\*\*username\*\*\*\*");  *driver*.findElement(By.*xpath*("//\*[@id='identifierNext']/content/span")).click();  Thread.*sleep*(1000);  *driver*.findElement(By.*name*("password")).sendKeys("\*\*\*\*\*\*\*\*\*\*\*");  *driver*.findElement(By.*xpath*("//\*[@id='passwordNext']/content/span")).click();  System.***out***.println("Logged in successfully"); |

**Webdriver interview questions:**

**1.  Difference between Absolute path & Relative path.**

Absolute path will start with root path (/) and Relative path will from current path (//)

**Absolute xPath :** /html/body/div[3]/div[2]/div[2]/div[2]/div[2]/div[2]/div[2]/div/div[4]/div[1]/div/div[@id='main']/div[@id='Blog1']/div[1]/div[1]/div/div[1]/div/h3/a  
  
**Relative xPath :** //h3/a[text()='Working on New Window']

**2.  Tell me some TestNG Annotations.**

@Test,@Parameters,@Listeners,@BeforeSuite,@AfterSuite,@BeforeTest,@AfterTest,  
@DataProvider,@BeforeGroups,@AfterGroups,@BeforeClass,@AfterClass,  
@BeforeMethod,@AfterMethod,@Factory

<http://testng.org/doc/documentation-main.html#annotations>

**3.  What are desiredcapabilities?**

Desired Capabilities help to set properties for the Web Driver. A typical use case would be to  
set the path for the Firefox Driver if your local installation doesn't correspond to the default  
settings.

<https://code.google.com/p/selenium/wiki/DesiredCapabilities>

**4.  Difference between Selenium RC and Selenium Web driver.**

 Difference between Selenium RC and Selenium Web driver.

|  |  |
| --- | --- |
| **Selenium RC** | **Selenium Web driver** |
| Selenium RC’s architecture is way more complicated. | Web Driver’s architecture is simpler than Selenium RC’s. |
| Selenium RC is slower since it uses a JavaScript program called Selenium Core.This Selenium Core is the one that directly controls the browser, not you. | Web Driver is faster than Selenium RC since it speaks directly to the browser uses the browser’s own engine to control it. |
| Selenium Core, just like other JavaScript codes, can access disabled elements. | Web Driver interacts with page elements in a more realistic way. |
| Selenium RC’s API is more matured but contains redundancies and often confusing commands. | Web Driver’s API is simpler than Selenium RC’s. It does not contain redundant and confusing commands. |
| Selenium RC cannot support the headless HtmlUnit browser. It needs a real, visible browser to operate on. | Web Driver can support the headless HtmlUnit browser. |
| Selenium RC Has Built-In Test Result Generator. Selenium RC automatically generates an HTML file of test results. | Web Driver has no built-in command that automatically generates a Test Results File. |
| Selenium RC needs the help of the RC Server in order to do so. | web Driver directly talks to the browser |
| Selenium RC can support new browsers | It cannot readily support new browsers |

**5.  Difference between Web driver listener and  TestNG Listener.**

TestNG and Web driver Listener have different interfaces to implement and call them. They both  
modify respective behaviour. You can use Listeners in Annotation. Below 2 URL gives the  
 detailed  list of listener and their interfaces.

<http://testng.org/doc/documentation-main.html#testng-listeners>

[http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/events/](http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/events/AbstractWebDriverEventListener.html)  
[AbstractWebDriverEventListener.html](http://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/events/AbstractWebDriverEventListener.html)

**6. Describe your framework.**

Refer automation link

**7.  Which is the best way to locate an element?**

Finding elements by ID is usually going to be the fastest option, because at its root, it eventually calls down to document.getElementById(), which is optimized by many browsers.

Finding elements by XPath is useful for finding elements using very complex selectors, and is the most flexible selection strategy, but it has the potential to be very slow, particularly in IE. In IE 6, 7, or 8, finding by XPath can be an order of magnitude slower than doing the same in Firefox. IE provides no native XPath-over-HTML solution, so the project must use a JavaScript XPath implementation, and the JavaScript engine in legacy versions of IE really is that much slower.

If you have a need to find an element using a complex selector, I usually recommend using CSS Selectors, if possible. It's not quite as flexible as XPath, but will cover many of the same cases, without exhibiting the extreme performance penalty on IE that XPath can.

**8.  Why we refer Firefox driver to the web driver  inheritance.**

web Driver driver = new FireFoxDriver();

WebDriver is an interface which contain several abstract methods such as get(...), findElamentBy(...) etc.

We simply create reference of web Driver and we can assign objects (Firefox driver, CromeDriver, IEDriver, Andriod driver etc) to it.

Ex :

WebDriver driver = new FireFoxDriver();-----------(1)

If we are using (1) we can do the same thing by using

FireFoxDriver driver = new FireFoxDriver();---------(2)

We can use (1) and (2) for same purpose but if we want to switch to another browser in same program

then again we have to create the object of other class as for example

CromeDriver driver = new CromeDriver();.

creating object of several class is not good. So we create the reference of WebDriver and

we assign the objects of another class as for example

WebDriver driver; // it is created only one time in the program

driver = new FireFoxDriver();// any where in the program

driver = new CromeDriver(); // any where in the program

**9.  What are the features of TestNG?**

TestNG is a testing framework designed to simplify a broad range of testing needs, from  
unit testing (testing a class in isolation of the others) to integration testing (testing entire  
systems made of several classes, several packages and even several external frameworks,  
 such as application servers). You can use test suite,annotations, automatically generation  
 of report and much more.

**10.   What is the difference between thread.Sleep()  and selenium. Set Speed ("2000")?**

If the application is taking time to load the page then we use selenium.waitforpageload(" "). This command is doesn’t wait upto the given time whenever the page load is completed.

If the application is taking time to refresh the page, then we use Thread. Sleep ( ).it is a standard wait it simply wait to the given time.

**selenium.setSpeed**

 1. Takes a single argument in string format

 Ex: selenium.setSpeed("2000") - will wait for 2 seconds

 2. Runs each command in after setSpeed delay by the number of milliseconds mentioned in set Speed.

**thread.sleep**

 1. Takes a single argument in integer format

 ex: thread. Sleep(2000) - will wait for 2 seconds

 2. Waits for only once at the command given at sleep.

**11.   In what situation selenium finding element get fails?**

-->Element loading issue

-->Dynamic id of web element

**12.   What is the difference between "GET" and "NAVIGATE" to open a web page in selenium web driver?**

Get method will get a page to load or get page source or get text that's all whereas navigate  
 will guide  through the history like refresh, back, forward.For example if we want to move  
forward and do some functionality and back to the home page this can be achieved  
through navigate() only. driver.get will wait  till the whole page gets loaded and driver.navigate  
will just redirect to that page and will not wait

**13.   Please tell me the difference b/w implicitly Wait and Explicit wait**.

Implicit Wait sets internally a timeout that will be used for all consecutive Web Element searches. It will try lookup the element again and again for the specified amount of time before throwing a NoSuchElementException if the element could not have been found. It does only this and can't be forced into anything else - it waits for elements to show up.

Explicit Wait or just Wait is a one-timer used by you for a particular search. It is more extendible in the means that you can set it up to wait for any condition you might like. Usually, you can use some of the prebuilt Expected Conditions to wait for elements to become clickable, visible, invisible, etc., or just write your own condition that suits your needs.

**14.   How we can retrieve the dynamically changing Ids?**

When we login Facebook the login label's id  
changes dynamically thus resulting in failure.

We have a hierarchy of locators and Facebook Is dynamic in nature,so we are not able to  
use "id" for  identification for after that we have remaining 7 locator's for that :2. xpath ()..  
3. name..4. css.. 5. link text.. 6. partiallinktext...7.tag name. so u can use any one for  
 identifying it. Most probably u can use "xpath" or  "css-locator" and if there r tag then  
link text or partial-link text. it depend on u . But we never use id's in Ajax application  
 because it’s not possible.

**15.What is the difference between driver.Close()  and driver.Quit () method?**

Close() - It is used to close the browser or page currently which is having the focus.

Quit() - It is used to shut down the web driver instance or destroy the web driver instance  
(Close all the windows)

**16.   How to scroll web element?--not browser—**

FirefoxProfile profile=new FirefoxProfile();

profile.setEnableNativeEvents(true);

WebDriver driver=new FirefoxDriver(profile);

driver.navigate("http://jqueryui.com/draggable/");

Thread.sleep(6000L);

WebElement element=driver.findElement(By.xpath("//div[@id='draggable']"));

Actions actn=new Actions(driver);

actn.dragAndDropBy(element, 50, 50).build().perform();

}

**17.   What is the basic use of Firefox profiles and how can we use them using selenium?**

A profile in Firefox is a collection of bookmarks, browser settings, extensions, passwords,  
and history; in short, all of your personal settings.

We use them to change user agent, changing default download directory, changing versions etc.

<http://code.google.com/p/selenium/wiki/FirefoxDriver>

**18.   Customize the name of file going to be downloaded?**

You have to download AUTO IT.exe file and has to be install

and later you have create .au3 file (in this file you have to specify the commands in  
 VB script  like your file  name, where have to save, it will be easy may be 3 or 4 steps )  
using AUTOIT...then right click the .au3 file you have to compile ....after that you will  
get the .exe file with the name of .au3 file ..In eclipse you will give the code like this

<----processbuildder .exe="" .start="" au3="" file="" of="" path="" processbuilder="" ps="new" the="">

**19.   How to handle internationalisation through  web driver?**

FirefoxProfile profile = new FirefoxProfile();

profile.set Preference("intl.accept\_languages","jp");

Web driver driver = new FirefoxDriver(profile); driver.get(google.com) will open google in  
Japanese Lang

**20.   How to overcome same origin policy through web driver?**

Proxy server.

DesiredCapabilities capability=new DesiredCapabilities.firefox();

capability.setCapability(CapabilityType.PROXY,"your desire proxy")

WebDriver driver=new FirefoxDriver(capability);

**21. How to put text in Facebook search box using  selenium web driver.**

driver.findElement(By.xpath("//div[contains(@class, '\_586i')]")).sendKeys("abc");

**22. Difference between flex and flash application**.

In flash there is no code just based on creativity(design) we will complete the  
work(time consuming process) whereas flex contain some small functions  
which is integrated with mxml,PHP..(no tool is there to develop

 flex we want to use the properties of css and style sheet)

**23. What is Error Collector in TestNG? What is its use?**

This class allows the collection of errors during the process of retrieving the  
test data for the test method parameters

[http://testngdatabind.sourceforge.net/apidocs/net/sf/testng/databinding/core/error/](http://testngdatabind.sourceforge.net/apidocs/net/sf/testng/databinding/core/error/ErrorCollector.html)

[ErrorCollector.html](http://testngdatabind.sourceforge.net/apidocs/net/sf/testng/databinding/core/error/ErrorCollector.html)

**24. How can we get the font size, font color, font type used  for a particular text on a web page using Selenium web driver?**

driver.findelement(By.Xpath("Xpath ").getcssvalue("font-size);

driver.findelement(By.Xpath("Xpath ").getcssvalue("font-colour);

driver.findelement(By.Xpath("Xpath ").getcssvalue("font-type);

driver.findelement(By.Xpath("Xpath ").getcssvalue("background-colour);

**25. How to run tests in multiple browser parallel? Is there  any other option other than selenium grid?**

**26. How to prepare Customized html Report using  TestNG  in hybrid framework.**

Below are the 3 ways:

• Junit: with the help of ANT.

• TestNG: using inbuilt default.html to get the HTML report. Also XST reports from ANT,

Selenium, TestNG combination.

• Using our own customized reports using XSL jar for converting XML content to HTML.

**27. “What’s the hierarchy of TestNG annotations?  Explain me about annotation hierarchy & execution  order?**

**Please find hierarchy below:**

1.      org.testng.annotations.Parameters (implements java.lang.annotation.Annotation)

2.      org.testng.annotations.Listeners (implements java.lang.annotation.Annotation)

3.      org.testng.annotations.Test (implements java.lang.annotation.Annotation)

4.      org.testng.annotations.AfterMethod (implements java.lang.annotation.Annotation)

5.      org.testng.annotations.BeforeTest (implements java.lang.annotation.Annotation)

6.      org.testng.annotations.BeforeMethod (implements java.lang.annotation.Annotation)

7.      org.testng.annotations.Optional (implements java.lang.annotation.Annotation)

8.      org.testng.annotations.AfterTest (implements java.lang.annotation.Annotation)

9.      org.testng.annotations.Guice (implements java.lang.annotation.Annotation)

10.  org.testng.annotations.BeforeGroups (implements java.lang.annotation.Annotation)

11.  org.testng.annotations.ExpectedExceptions (implements java.lang.annotation.Annotation)

12.  org.testng.annotations.TestInstance (implements java.lang.annotation.Annotation)

13.  org.testng.annotations.NoInjection (implements java.lang.annotation.Annotation)

14.  org.testng.annotations.AfterSuite (implements java.lang.annotation.Annotation)

15.  org.testng.annotations.AfterClass (implements java.lang.annotation.Annotation)

16.  org.testng.annotations.AfterGroups (implements java.lang.annotation.Annotation)

17.  org.testng.annotations.DataProvider (implements java.lang.annotation.Annotation)

18.  org.testng.annotations.BeforeSuite (implements java.lang.annotation.Annotation)

19.  org.testng.annotations.BeforeClass (implements java.lang.annotation.Annotation)

20.  org.testng.annotations.Factory (implements java.lang.annotation.Annotation)

21.  org.testng.annotations.Configuration (implements java.lang.annotation.Annotation)

22.  org.testng.annotations.ObjectFactory (implements java.lang.annotation.Annotation)

Useful link:  
<http://www.guru99.com/all-about-testng-and-selenium.html>

**28. How the TestNG interacts with Selenium Core?**

Explain me steps and internal architecture?"

<http://www.guru99.com/all-about-testng-and-selenium.html>

**29. Is it possible test web services using selenium?**

Using Jmeter we can test how one website is talking to each other means time taken to  
send data, feeds, messages from one website to other website. Jmeter does a nice job  
of doubling for performance and api tests.

**30. How to refresh a page without using context click?**

1.Using sendKeys.Keys method

2.Using navigate.refresh() method

3.Using navigate.refresh() method

4.Using get() method

5.Using sendKeys() method

1.Using sendKeys.Keys method

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

driver.findElement(By.id("firstname-placeholder")).sendKeys(Keys.F5);

2.Using navigate.refresh() method

driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-interview-questions.html");

driver.navigate().refresh();

3.Using navigate.to() method

driver.get("http://ruchi-myseleniumblog.blogspot.in/2014/01/selenium-hybrid-framework-using.html");

driver.navigate().to(driver.getCurrentUrl());

4.Using get() method

driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/basic-core-java-interview-questions.html");

driver.get(driver.getCurrentUrl());

5.Using sendKeys() method

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

driver.findElement(By.id("firstname-placeholder")).sendKeys("\uE035");

**31. Can u send a code for printing in selenium?**

There are two cases:

**Case1.**Any hyperlink/button on a web page, n clicking that link/button a print dialog box  
opens. (Performing an action on web page)

**Case2.**or do u want to open print dialog box within ur own script, not by performing any  
action on web page.

So If Case 1: just a call for WebElement.click() event will work to open it.

If Case 2: Call a Printer Job object (Use Awt API).

For code: Google it.

<http://code.google.com/p/selenium/issues/detail?id=1815>

**32. How to find broken images in a page using  Selenium Web driver.**

1. Get xpath and then using tag name; get all the links in the page

2. Click on each and every link in the page

3. In the target page title, look for 404/500 error.

**33. How to handle Ajax popup window?**

By using getWindowHandles() and obj.switchTo.window(windowid) we can handle popups using  
explicit wait and driver.swtchT0.window("name") commands for your requirements.

**34. How to handle auto complete box in web driver?**

By typing in box and capturing list elements

driver.findElement(By.id("your searchBox")).sendKeys("your partial keyword");

Thread.sleep(3000);

List listItems = driver.findElements(By.xpath("your list item locator"));

listItems.get(0).click();

driver.findElement(By.id("your searchButton")).click();

**35. How to get the name of browser using Web Driver?**

public class JsExecute

{

WebDriver driver;

JavascriptExecutor js;

@Before

public void setUp() throws Exception

{

driver=new FirefoxDriver();

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

}

@Test

public void test()

{

JavascriptExecutor js = (JavascriptExecutor) driver;

System.out.println(js.executeScript("return navigator.appCodeName"));

}}

OR

String s = (String) ((JavascriptExecutor) driver).executeScript("return navigator.userAgent;");

System.out.println("Browser name : " + s);

**36. How to handle colors in web driver?**

Use getCssValue(arg0) function to get the colors by sending 'color' string as an argument.

Example

String col = driver.findElement(By.id(locator)).getCssValue("color");

**37. How to pass parameters from testng.xml into  test case.**

package programs;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.firefox.FirefoxDriver;  
import org.openqa.selenium.htmlunit.HtmlUnitDriver;  
import org.openqa.selenium.ie.InternetExplorerDriver;  
import org.testng.annotations.BeforeTest;  
import org.testng.annotations.Parameters;  
import org.testng.annotations.Test;  
  
public class Parallelexecution {  
  
 private WebDriver driver = null;  
  
 @BeforeTest  
 @Parameters({ "BROWSER" })  
 public void setup(String BROWSER) {  
  System.out.println("Browser: " + BROWSER);  
  
  if (BROWSER.equals("FF")) {  
   System.out.println("Firefox Browser is selected");  
   driver = new FirefoxDriver();  
  } else if (BROWSER.equals("IE")) {  
   System.out.println("Internet Explorer Browser is selected");  
   driver = new InternetExplorerDriver();  
  } else if (BROWSER.equals("HU")) {  
   System.out.println("Html Unit Browser is selected");  
   driver = new HtmlUnitDriver();  
  } else if (BROWSER.equals("CH")) {  
   System.out.println("Google chrome Browser is selected");  
   driver = new ChromeDriver();  
  }  
 }  
  
 @Test  
 public void testParallel() throws Exception {  
  driver.get("http://ruchi-myseleniumblog.blogspot.in/2013/12/100-selenium-interview-questions.html");  
  
 }  
}

## above sample program BROWSER is a variable which value would be passed from TestNG.xml and TestNG.xml and it will run the test multiple time each time BROWSER value would be set with different browser name and test will check the BROWSER value and decide which browser test will run.

**38. How to get text from captcha image??**

driver.findElement(By.xpath(".//\*[@id='SkipCaptcha']")).click();

String attr = ie.findElement(By.xpath(".//\*[@id='SkipCaptcha']")).getAttribute("value");

System.out.println("The value of the attribute 'Name' is " + attr);

**39. Is there a way to click hidden LINK in web driver?**

String Block1 = driver.findElement(By.id("element ID"));

JavascriptExecutor js1=(JavascriptExecutor)driver;

js1.executeScript("$("+Block1+").css({'display':'block'});");

**40. What Class Extends Web Driver?**

AndroidDriver, ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver,  
InternetExplorerDriver, IPhoneDriver, PhantomJSDriver, RemoteWebDriver, SafariDriver

**41. What are the APIs that support Web Driver?**

API are  nothing but collection of all selenium commands for Locating UI Elements  
(WebElements),Fetching a Page,User Input  etc…

**42. How to disable cookies in browser.**

Using deleteAllVisibleCookies() in selenium

**43. "We have heard about frameworks well it can be  
 broadly classified into these TDD, BDD and ATDD frameworks .What’s the Difference?"**

For Details please see: <http://assertselenium.com/2012/11/05/difference-between-tdd-bdd-atdd/>

**44. How to change user agent in Firefox by selenium  web driver.**

FirefoxProfile profile = new FirefoxProfile();

profile.setPreference("general.useragent.override", "some UA string");

Web Driver driver = new FirefoxDriver(profile);

**45. What is Selenese?**

Selenese is HTML language based command, which is used in Selenium IDE.

**46.Differences between QTP and selenium.**

1) Selenium generates a proxy while starting browser. QTP does not

2) QTP uses only Vb script. Selenium is available in many languages

3) QTP is paid and selenium is free.

4) You can run script from a particular line in QTP but in selenium, you cannot.

5) Selenium works on all browsers. QTP only works on IE, mozilla. Support from chrome has been introduced lately.

6) QTP is more organized and user friendly

7) Selenium requires more technical skills

8) QTP can also be used on desktop based applications but selenium cannot be used

**47. What is the MOST challenging test problem in my career in Automation?**

In my career

Changing XPATHS' between testing server and production server-by keeping generic xpath

Keep separate property files for production and UAT

automating flash apps

Mobile Automation

**48. “Suppose developer changed the existing image to new image with same xpath. Is test case pass or fail?"**

Pass

**49. How to handle network latency using selenium?**

Using driver.manage.pageLoadingtime for network latency

**50. How does u handle dynamic elements without using  xpath (with example?)**

By using classname or css.

**51. What are the different types of driver implementation?**

AndroidDriver, AndroidWebDriver, ChromeDriver, EventFiringWebDriver, FirefoxDriver, HtmlUnitDriver, InternetExplorerDriver, IPhoneDriver, IPhoneSimulatorDriver, RemoteWebDriver, SafariDriver, WebDriverBackedSelenium

**52. Code for Opening Firefox browser?**

Webdriver driver=new FireFoxdriver();

**53. Which repository you have used to store the test  scripts?**

I have created scripts in excel file and store them in Test cases folder under src .

**54. How to work with radio button in web driver?**

We can select the value from the drop down by using 3 methods.

selectByVisibleText - select by the text displayed in drop down

selectByIndex  - select by index of option in drop down

selectByValue  - select by value of option in drop down



WebElement e = driver.findElement(By.id("44"));

Select selectElement=new Select(e);

// both of the below statements will select first option in the weblist

selectElement.selectByVisibleText("xyz");

selectElement.selectByValue("1");

**55. How to work with dynamic web table?**

You can get the total number of tags within a tag by giving the xpath of the  
  element by using  this function -

List ele = driver.findElements(By.xpath("Xpath of the table"));

Now you can use a for each loop to loop through each of the tags in the above list  
and then read each  value by using getText() method.

**56. Detail about TestNG Test Output folder.**

It is the directory where reports are generated. Every time tests run in a suite, TestNG  
creates index.html and  other files in the output directory.

**57. In frame if no frame Id as well as no frame  name then which attribute I should consider  throughout our script.**

You can go like this.....driver.findElements(By.xpath("//iframe"))...

Then it will return List of frames then switch to each and every frame and search for  
 the locator which you want then break the loop

**58. What is object repository?**

It is collection of object names their properties, attributes and their values .It maye be  
 excel, XML,property file or text file

**59. TestNG vs. Junit?**

Advantages of TestNG over Junit

In Junit we have to declare @BeforeClass and @AfterClass which is a constraint where as in TestNG there is no constraint like this.

-->Additional Levels of setUp/tearDown level are available in TestNG like @Before/AfterSuite,@Before/AfterTest and @Before/AfterGroup

No Need to extend any class in TestNG.

There is no method name constraint in TestNG as in Junit. You can give any name to the test methods in TestNG

-->In TestNG we can tell the test that one method is dependent on another method where as in Junit this is not possible. In Junit each test is independent of another test.

Grouping of testcases is available in TestNG where as the same is not available in Junit.

-->Execution can be done based on Groups. For ex. If you have defined many cases and segregated them by defining 2 groups as Sanity and Regression. Then if you only want to execute the “Sanity” cases then just tell TestNG to execute the “Sanity” and TestNG will automatically execute the cases belonging to the “Sanity” group.

Also using TestNG your selenium test case execution can be done in parallel.

For more Details Go to : <http://www.mkyong.com/unittest/junit-4-vs-testng-comparison/>

**60. What is the difference between @before method and @beforeclass.**

In JUnit4 @Before is used to execute set of preconditions before executing a test.  
 For example, if there is a need to open some application and create a user before  
executing a test, then this annotation can be used for that method.  Method that is  
marked with @Before will be executed before executing every test in the class.

If a JUnit test case class contains lot of tests which all together need a method  
which sets up a precondition  and that needs to be executed before executing the  
Test Case class then we can utilise “@BeforeClass” annotation.

**61. What are the different Parameters for @Test annotation?**

Parameters are keywords that modify the annotation’s function.

For more details Go to: <http://testng.org/doc/documentation-main.html#parameters>

**62. Can we run group of test cases using TestNG?**

Test cases in group in Selenium using TestNG will be executed with the below options.

 If you want to execute the test cases based on one of the group like regression test or smoke test

 @Test(groups = {"regressiontest", "smoketest"})

 For more details please see: <http://testng.org/doc/documentation-main.html#test-groups>

**63. Differences between Selenium web driver, IDE and RC?**

<http://qtpselenium.com/selenium-tutorial/difference-between-ide-rc-webdriver/>

**64. How to highlight an object like qtp/uft does through selenium and java?**

### How to highlight an object with selenium and java

How to highlight an object like qtp/uft does through selenium and java?

public void highlightElement(WebDriver driver, WebElement element) {

for (int i = 0; i < 2; i++)

{

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "color: yellow; border: 2px solid yellow;");

js.executeScript("arguments[0].setAttribute('style', arguments[1]);", element, "");

}}

Call the highlightElement method and pass webdriver and WebElement which you want to highlight as arguments.

**65. What are the different assertions in SIDE?**

**Assertions** are like Assessors, but they verify that the state of the application conforms  
 to what is expected. Examples include "make sure the page title is X" and "verify that this  
 check box is checked".

All Selenium Assertions can be used in 3 modes: "assert", "verify", and "waitFor".  
  
 For example, you can "assertText", "verifyText" and "waitForText". When an "assert" fails, the test is aborted. When a "verify" fails, the test will continue execution, logging the failure. This allows a single "assert" to ensure that the application is on the correct page, followed by a bunch of "verify" assertions to test form field values, labels, etc.  
  
"waitFor" commands wait for some condition to become true (which can be useful for testing Ajax applications). They will succeed immediately if the condition is already true. However, they will fail and halt the test if the condition does not become true within the current timeout setting (see the **setTimeout** action below).

<http://selenium-tutorial.blogspot.in/2013/03/selenium-webdriver-assertions.html>

**66. How to store a value which is text box using  web driver?**

driver.findElement(By.id("your Textbox")).sendKeys("your keyword");

**67. How to handle alerts and confirmation boxes. Confirmation boxes and Alerts are handled in same way in selenium.**

var alert = driver.switchTo().alert();

alert.dismiss();  //Click Cancel or Close window operation

alert.accept();   //Click OK

Handle Confirmation boxes via JavaScript,

driver.executeScript("window.confirm = function(message){return true;};");

**68. How to mouse hover on an element?**

Actions action = new Actions(webdriver);

WebElement we = webdriver.findElement(By.xpath("html/body/div[13]/ul/li[4]/a"));

action.moveToElement(we).moveToElement(webdriver.findElement(By.xpath("/expression-here"))).click().build().perform();

**69. How to switch between the windows?**

private void handlingMultipleWindows(String windowTitle) {

            Set windows = driver.getWindowHandles();

            for (String window : windows) {

                driver.switchTo().window(window);

                if (driver.getTitle().contains(windowTitle)) {   return;   }     }     }

**70. How to switch between frames?**

WebDriver's [driver.switchTo().frame()](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html) method takes one of the three possible arguments:

[A number.](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html#frame%28int%29)

Select a frame by its (zero-based) index. That is, if a page has three frames, the first frame would be at index "0", the second at index "1" and the third at index "2". Once the frame has been selected, all subsequent calls on the WebDriver interface are made to that frame.

[A name or ID.](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html#frame%28java.lang.String%29)

Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.

[A previously found WebElement.](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.TargetLocator.html#frame%28org.openqa.selenium.WebElement%29)

Select a frame using its previously located WebElement.

Get the frame by it's id/name or locate it by [driver.findElement()](http://selenium.googlecode.com/svn/trunk/docs/api/java/org/openqa/selenium/WebDriver.html#findElement%28org.openqa.selenium.By%29) and you'll be good.

**71. What is actions class in web driver?**

Actions class with web Driver help is Sliding element, Resizing an Element, Drag & Drop,

hovering a mouse, especially in a case when dealing with mouse over menus.

**Dragging & Dropping an Element:**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

public class testDragandDrop {

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

   WebDriver driver = new FirefoxDriver();

  driver.get("http://jqueryui.com/resources/demos/droppable/default.html");

  WebElement draggable = driver.findElement(By.xpath("//\*[@id='draggable']"));

  WebElement droppable = driver.findElement(By.xpath("//\*[@id='droppable']"));

  Actions action = new Actions(driver);

action.dragAndDrop(draggable, droppable).perform();

  }

}

**Sliding an Element:**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

public class testSlider {

  /\*\*

  \* @param args

  \* @throws InterruptedException

  \*/

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

   WebDriver driver = new FirefoxDriver();

  driver.get("http://jqueryui.com/resources/demos/slider/default.html");

  WebElement slider = driver.findElement(By.xpath("//\*[@id='slider']/a"));

  Actions action = new Actions(driver);

  Thread.sleep(3000);

  action.dragAndDropBy(slider, 90, 0).perform();

  }

}

**Re-sizing an Element:**

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.interactions.Actions;

public class testResizable {

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

   WebDriver driver = new FirefoxDriver();

  driver.get("http://jqueryui.com/resources/demos/resizable/default.html");

    WebElement resize = driver.findElement(By.xpath("//\*[@id='resizable']/div[3]"));

  Actions action = new Actions(driver);

  action.dragAndDropBy(resize, 400, 200).perform();

    }

}

**72. Difference between the selenium1.0 and selenium 2.0?**

Selenium 1 = Selenium Remote Control.

Selenium 2 = Selenium Web driver, which combines elements of Selenium 1 and Web driver.

**73. Difference between find element () and findelements ()?**

**findElement() :**

Find the first element within the current page using the given "locating mechanism".

Returns a single WebElement.

**findElements() :**

Find all elements within the current page using the given "locating mechanism".

Returns List of Web Elements.

**findElement() :**

Find the first element within the current page using the given "locating mechanism".

Returns a single WebElement.

Syntax: WebElement findElement(By by)

Ex:

driver.get("http://ruchi-myseleniumblog.blogspot.in/");

WebElement widget = driver.findElement(By .xpath(".//\*[@id='BlogArchive1\_ArchiveList']"));

widget.click();

**findElements() :**

Find all elements within the current page using the given "locating mechanism".

Returns List of WebElements.

Syntax:  
        WebElement ullist = driver.findElement(By.className("posts"));

  List posts = ullist.findElements(By.tagName("li"));

  System.out.println("List of Posts are Below");

  for (int i = 0; i < posts.size(); i++) {

   String post = posts.get(i).findElement(By.tagName("a")).getText();

   System.out.println(post);

  }

**74. How to take the screen shots in seelnium2.0?**

//store screenshots  
 public static void captureScreenShot(String filePath) {  
  File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  
    try {  
   FileUtils.copyFile(scrFile, new File(filePath));  
  } catch (IOException e) {  
   // TODO Auto-generated catch block

e.printStackTrace();

}

}

**75. What is the default time for selenium Ide and webdriver?**

Default timeout in selenium ide is 30 seconds.

For web driver go to below URL:

<http://assertselenium.com/2013/01/29/webdriver-wait-commands/>

**76. Write down scenarios which we can't automate?**

Barcode Reader, Captcha etc.

**77. In TestNG I have some test's  Test1-Test2-Test3-Test4-Test5I want to run my execution order is Test5-Test1-Test3-Test2-Test4.How do you set the execution order can you explain for that?**

Use priority parameter in @test annotation or TestNG annotations.

**78. Differences between jxl and ApachePOI.**

jxl does not support  XLSX files

jxl exerts less load on memory as compared to ApachePOI

jxl doesn't support rich text formatting while ApachePOI does.

jxl has not been maintained properly while ApachePOI is more up to date.

Sample code on Apache POI is easily available as compare to jxl.

### 79.How to ZIP files in Selenium

// Sample Function to make zip of reports  
 public static void zip(String filepath){  
  try  
  {  
   File inputFolder=new File('Mention file path her");  
   File outputFolder=new File("Reports.zip");  
   ZipOutputStream out = new ZipOutputStream(new BufferedOutputStream(new FileOutputStream(outputFolder)));  
   BufferedInputStream in = null;  
   byte[] data  = new byte[1000];  
   String files[] = inputFolder.list();  
   for (int j=0; j    {  
    in = new BufferedInputStream(new FileInputStream  
    (inputFolder.getPath() + "/" + files[j]), 1000);   
    out.putNextEntry(new ZipEntry(files[j]));  
    int totalcount;  
    while((totalcount= in.read(data,0,1000)) != -1)  
    {  
     out.write(data, 0, totalcount);  
    }  
    out.closeEntry();  
  }  
  out.flush();  
  out.close();    
}  
  catch(Exception e)  
  {  
  e.printStackTrace();  
           return "Fail - " + e.getMessage();  
  }  
 }

**80. What is default port no?**

4444

**81. If Default port no is busy how to change port no?**

We can use any port number which is valid.. First create an object to remote control configuration.  
Use 'setPort' method and provide valid port number(4545,5555,5655, etc).. There after attach this  
remote control configuration object to selenium server..i.e

RemoteControlConfiguration r= new RemoteControlConfiguration();

r.setPort(4567);

SeleniumServer s= new SeleniumServer(r);

**82. Does Selenium support https protocols?**

Yes

**83. Majorly asked test scenario with framework in Interviews?**

Majorly asked are:

·         Login for Gmail scenario

·         Goggle search and finding no of results

·         Downloading a file and save it

·         Checking mails and deleting them

·         Do shopping in flipkart.com

**84. Selenium support mobile applications?**

No, it is browser automation tool, it only automates Websites opening in mobile browser, and mobile APPs

 can't be automated.

**85. What is wraps Driver?**

For casting selenium instance to selenium2 (webdriver). wraps driver is used.

For more details.

[http://selenium-junit4-runner.btmatthews.com/apidocs/com/btmatthews/selenium](http://selenium-junit4-runner.btmatthews.com/apidocs/com/btmatthews/selenium/junit4/runner/WrappedDriverFactory.html)  
[/junit4/runner/WrappedDriverFactory.html](http://selenium-junit4-runner.btmatthews.com/apidocs/com/btmatthews/selenium/junit4/runner/WrappedDriverFactory.html)

**86. Can you explain Junit Annotation? If there are 1000 test cases. 500 test cases are executed. How  will you execute the rest of the test cases by using annotation?"**

The annotations generated with JUnit 4 tests in Selenium are:

 1. @Before public void method() - Will perform the method() before each test. This method  
 can prepare the  test

 2. @Test public void method() - Annotation @Test identifies that this method is a test  
method.environment,e.g. read input data, initialize the class)

 3. @After public void method() - Test method must start with test@Before - this annotation  
 is used for  executing a method before

**87. Difference between assert and verify in selenium  web driver.**

When an “assert” fails, the test will be aborted. Assert is best used when the  
 check value has to pass for  the test to be able to continue to run log in.

Where if a “verify” fails, the test will continue executing and logging the failure.  
Verify is best used to  check non critical things. Like the presence of a  
 headline element.

**88. "I want to find the location of ""b"" in the below  code, how can I find out without using xpath, name,  id, csslocator, index.**

a

b

c

    ·      driver.findElement(By.xpath("//\*[contains(text(),'b')]")).click(); or

//div/button[contains(text(),'b']

**89. How to do Applet testing using selenium?**

Please see below URLs:

<http://docs.codehaus.org/display/FEST/Selenium>

<https://code.google.com/p/festselenium/>

**90.  Name 5 different exceptions you had in selenium web driver and mention what instance you got it and how do you resolve it?**

WebDriverException

NoAlertPresentException

NoSuchWindowException

NoSuchElementException

TimeoutException

• WebDriverException

 WebDriver Exception comes when we try to perform any action on the non-existing

driver.

WebDriver driver = new InternetExplorerDriver();

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

driver.close();

driver.quit();

• NoAlertPresentException

When we try to perform an action i.e., either accept() or dismiss() which is not required

at a required place; gives us this exception.

try{

driver.switchTo().alert().accept();

}

catch (NoAlertPresentException E){

E.printStackTrace();

}

• NoSuchWindowException

 When we try to switch to an window which is not present gives us this exception:

WebDriver driver = new InternetExplorerDriver();

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

driver.switchTo().window("Yup\_Fail");

driver.close();

In the above snippet, line 3 throws us an exception, as we are trying to switch to an

window that is not present.• NoSuchFrameException

• Similar to Window exception, Frame exception mainly comes during switching between

the frames.

WebDriver driver = new InternetExplorerDriver();

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

driver.switchTo().frame("F\_fail");

driver.close();

In the above snippet, line 3 throws us an exception, as we are trying to switch to an

frame that is not present.

• NoSuchElementException

 This exception is thrown when we WebDriver doesn’t find the web-element in the DOM.

WebDriver driver = new InternetExplorerDriver();

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

driver.findElement(By.name("fake")).click();

• TimeoutException

 Thrown when a command does not complete in enough time.

All the above exceptions were handled using try catch exceptions.

**91. How do you manage the code versions in your project?**

Using SVN or other versioning tools

**92. Latest version of Firefox and selenium in market and the version on which you are testing  which you are testing.**

FF Latest version till Dec,2013 for windows7,64 bit :26.0.I use FF 25.0.1 (ur ans. may differ)

Selenium web driver latest version till dec,2013- 2.39.0 I use selenium 2.37 see latest at

<http://www.seleniumhq.org/download/>

**93. How to know all the methods supported in web driver and its syntax.**

In Org.openqa.selenium package, web driver interface has all the main methods that can

be used in Selenium Web driver

[HTTP://docs.seleniumhq.org/docs/03\_webdriver.jsp](http://docs.seleniumhq.org/docs/03_webdriver.jsp)

**94. How do you create html test report from your test script?**

• I would see below 3 ways:

• Junit: with the help of ANT.

• TestNG: using inbuilt default.html to get the HTML report. Also XLST reports from ANT,

Selenium, TestNG combination.

• Using our own customized reports using XSL jar for converting XML content to HTML.

**95. List the browsers, OS supported by the Selenium**

Windows Linux Mac

IE Y NA NA

FF Y Y Y

Safari Y N Y

Opera Y Y Y

Chrome Y Y Y

**96. Can you explain Selenium Mobile Automation?**

<https://code.google.com/p/selenium/wiki/AndroidDriver>  
<http://manojhans.blogspot.in/2013/08/native-android-apps-automation-with.html>

**97. What mobile devices it may Support?**

Selenium Web driver supports all the mobile devices operating on Android, IOS operating Systems

Android – for phones and tablets (devices & emulators)

iOS for phones (devices & emulators) and for tablets (devices & emulators)

**98. What is the difference between single and  double slash  in Xpath?**

/

1.It starts selection from the document node

2. It Allows you to create 'absolute' path expressions

3. e.g “/html/body/p” matches all the paragraph elements

 //

1. It starts selection matching anywhere in the document

2. It Allows you to create 'relative' path expressions

3. e.g“//p” matches all the paragraph elements

**99. What are the test types supported by Selenium?**

Selenium supports UI and functional testing. As well it can support performance testing

for reasonable load using selenium grid.

**100. In what all case we have to go for “JavaScript executor”.**

Consider FB main page after you login. When u scrolls down, the updates get loaded. To  
 handle this activity, there is no selenium command. So you can go for javascript to set  
the scroll down value like driver.executeScript("window.scrollBy(0,200)", "");

**Queries:**

1. How to handle **web tables** in selenium

2. **How to use functions in xpath in selenium**

**3. How to identify the hidden elements in selenium**

**4. How to identify dynamic element in selenium**

**5. How to run Scripts in a Remote Machine**

**6. Handling Mouse hover**