**Note: practise with code**

**what are limitations in selenium web driver**

**ex: theory only**

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

**installing/configure selenium**

**ex: write down steps**

1. Download java and install then set the environment variables.
2. Download eclipse and install.
3. Create a workspace in eclipse.
4. Create a project 🡪 package 🡪 class
5. Download selenium libraries (Jar files)
6. Configure jar files into the project Right click on the project then select properties.
7. In the following window select java build path and then select add jar external files.
8. Browse for the jar files and add.

**what are different ways of locating elements in selenium**

**ex : write code for all locators**

By.id:

p.findElement(By.*id*("q")).sendKeys("vmware");

By.className:

p.findElement(By.className("gsc-search-button")).click();

By.click:

x.findElement(By.name("Login")).click();

By.tagname:

x.findElement(By.tagName(“html tag name”))

By Css selector:

x.findElement(By.cssSelector("#username")).sendKeys("swathi");

By Link:

x.findElement(By.Link(“link text”));

By xpath:

f.findElement(By.xpath("//a[@value='GOI']")).click();

**which is fastest way to identify elements in web page?**

**ex: theory only**

css selectors

**what is absolute path and relative path in xpath**

**ex: write code**

Absolute path: It uses the complete path from root to desired element.  
f.findElement(By.xpath("//\*[@id='dropdownGroup1']/div/ul[2]/li[4]/a")).click();

Relative Path: We can write by referencing any element.

f.findElement(By.xpath("//a[@value='GOI']")).click();

**different types of waits or synchronization in selenium webdriver**

**ex: write code**

Conditional

Unconditional

**how to save screen shots using selenium webdriver**

**Ex: write code**

import java.io.IOException;

import org.openqa.selenium.OutputType;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.chrome.ChromeDriver;

public class screenshot {

public static void main(String[] args) throws IOException

{

System.setProperty("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver.exe");

WebDriver p = new ChromeDriver();

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

File scrFile = ((TakesScreenshot)p).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(scrFile, new File("c:\\Downloads\\screenshot.png"));

}

}

**how to handle multiple windows in selenium webdriver**

**Ex: write code**

import java.util.List;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class MultipleWindowsHandle {

WebDriver driver;

public void setup() throws Exception {

driver=new FirefoxDriver();

String URL="http://www.seleniummaster.com";

driver.get(URL);

driver.manage().window().maximize();

}

public void test() throws Exception {

driver.findElement(By.xpath("//img[@alt='SeleniumMasterLogo']")).click();

String Parent\_Window = driver.getWindowHandle();

for (String Child\_Window : driver.getWindowHandles())

{

driver.switchTo().window(Child\_Window);

driver.findElement(By.*id*("dropdown\_txt")).click();

List dropdownitems=driver.findElements(By.*xpath*("//div[@id='DropDownitems']//div"));

int dropdownitems\_Size=dropdownitems.size();

System.**out**.println("Dropdown item size is:"+dropdownitems\_Size);

((WebElement) dropdownitems.get(1)).click();

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

}

driver.switchTo().window(Parent\_Window);

driver.findElement(By.*className*("btn\_style")).click();

}

public void close()

{

driver.quit();

}

}

**how to lanuch webpage using chrome driver**

**Ex: write code**

System.setProperty("webdriver.chrome.driver", "C:\\SeleniumDrivers\\chromedriver.exe");

WebDriver p = new ChromeDriver();

**what is desired capabilities in selenium webdriver**

**Ex: write code**

importorg.openqa.selenium.WebDriver;

importorg.openqa.selenium.ie.InternetExplorerDriver;

importorg.openqa.selenium.remote.DesiredCapabilities;

public class IEtestforDesiredCapabilities

{

public static void main(String[] args)

{

DesiredCapabilities capabilities = DesiredCapabilities.internetExplorer();

capabilities.setCapability(CapabilityType.BROWSER\_NAME, "IE");

capabilities.setCapability(InternetExplorerDriver.

INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,**true**);

System.*setProperty*("webdriver.ie.driver", "C:\\IEDriverServer.exe");

WebDriver driver = newInternetExplorerDriver(capabilities);

driver.manage().window().maximize();

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

driver.quit();

}

}

**how to set language while opening website**

**Ex: write code**

FirefoxProfile profile = **new** FirefoxProfile();

profile.setPreference("intl.accept\_languages", "de");

WebDriver driver = **new** FirefoxDriver(profile);

**how to handle windows based popups (upload and dropdown)**

**Ex: write code**

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import org.openqa.selenium.Alert;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class DemoWebAlert

{

WebDriver driver;

public DemoWebAlert()

{

}

public void setUp()

{

driver=new FirefoxDriver();

driver.get("file:///F:/Work/Selenium/Testing-Presentation/DemoWebPopup.htm");

driver.manage().window().maximize();

}

public void testWebAlert() throws InterruptedException {

driver.findElement(By.*xpath*("//button[contains(text(),'Try it')]")).click();

Thread.*sleep*(5000);

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

alert.accept();

driver.findElement(By.*xpath*("//button[contains(text(),'Try it')]")).click();

Thread.*sleep*(5000);

driver.switchTo().alert().dismiss();

driver.findElement(By.*xpath*("//button[contains(text(),'Try it')]")).click();

Thread.*sleep*(5000);

System.***out***.println(driver.switchTo().alert().getText());

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

}

public void tearDown()

{

driver.quit();

}

}

**write code to verify any application login page is working or not**

**(u should write code to use textbox, button click events)**

**Ex: write code**

**how to select items from dropdown/select box**

**Ex: write code**

public class checkBoxSel

{

public static void main(String[] args)

{

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

EventFiringWebDriver dr = null ;

dr = new EventFiringWebDriver(driver);

dr.get("http://www.google.co.in/");

dr.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

dr.findElement(By.linkText("Gmail")).click() ;

Select sel = **new** Select(driver.findElement(By.tagName("select")));

sel.selectByValue("fil");

}

}

**how to know if checkbox is checked or not in webpage**

**ex: write code**

if($('#isAgeSelected').attr('checked'))

{

$("#txtAge").show();

}

else

{

$("#txtAge").hide();

}

**tell me code to pass values from parent window to child window**

**Ex: write code**

function newWindow(){

var newWindow;

if(!newWindow||newWindow.closed)

{

newWindow =

window.open('popup','addwin','width=500,height=250,status=yes,scrollbars=yes,resizable=yes');

var tmp=newWindow.document;

setTimeout('newWindow.document.test.text1.value= 'objf.form\_parent.text1.value',50);

tmp.close();

}

}

**write code to find out if all links are working or not**

**Ex: write code**

import static org.junit.Assert.\*;

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.ProtocolException;

import java.net.URL;

import javax.net.ssl.SSLException;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

import org.junit.Test;

public cla**ss** BrokenLinks

{

String pageUrl = "http://www.google.com";

public void testBrokenLinks()

throws IOException, InterruptedException

{

Document page = getPage(pageUrl);

for (Element link : getLinksFromPage(page))

{

String url = getHrefValueFromLink(link);

if (isUrlValid(url))

*assertTrue*(getPageResponseCode(url) < 400);

}

}

private Document getPage(String url) throws IOException

{

return Jsoup.connect(pageUrl).get();

}

private Elements getLinksFromPage(Document page)

{

return page.select("a");

}

private String getHrefValueFromLink(Element link)

{

return link.attr("href");

}

private Boolean isUrlValid(String href)

{

return href.indexOf("http") >= 0;

}

private int getPageResponseCode(String url) throws IOException

{

int responseCode = 0;

try {

HttpURLConnection httpConnection = createHttpConnection(url);

httpConnection.connect();

responseCode = httpConnection.getResponseCode();

}

catch (ProtocolException | SSLException exception) {

System.***out***.println("could not open " + url);

}

return responseCode;

}

private HttpURLConnection createHttpConnection(String href) **throws** IOException

{

URL url = new URL(href);

HttpURLConnection huc = (HttpURLConnection)url.openConnection();

huc.setRequestMethod("GET");

return huc;

}

}

**write code on how to use javascriptexecutor?**

**Ex: write code**

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

public class JavaScriptExecuter

{

public static void main(String[] args)

{

WebDriver driver = new FirefoxDriver();

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

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

driver.manage().window().maximize();

JavascriptExecutor js = (JavascriptExecutor)driver;

String sDomain = js.executeScript("return document.domain;").toString();

System.out.println("Domain = "+sDomain);

String sURL = js.executeScript("return document.URL;").toString();

System.out.println("URL = "+sURL);

String sTitle = js.executeScript("return document.title;").toString();

System.out.println("Title = "+sTitle);

js.executeScript("window.scrollBy(0,200)");

System.out.println("Successfully did the vertical scroll by 200px");

}

}

**difference between assert and verify?**

**Ex: write code**

When an “assert” fails, the test will be aborted.

Where if a “verify” fails, the test will continue executing and logging the failure.

package testWordpress;

import org.testng.Assert;

import org.testng.annotations.Test;

public class AssertionsTest

{

public void testMultiply()

{

System.out.println(&quot;Before Error &quot;);

Assert.assertEquals(21, multiply(10, 5));

System.out.println(&quot;After Error &quot;);

}

public int multiply(int x, int y)

{

return x / y;

}

}

package testWordpress;

import org.testng.Assert;

import org.testng.annotations.Test;

public class AssertionsTest

{

public void testMultiply()

{

System.out.println(&quot;Before Error &quot;);

try

{

Assert.assertEquals(21, multiply(10, 5));

}

catch(Throwable t)

{

System.out.println(&quot;After Error &quot;);

}

}

public int multiply(int x, int y) {

return x / y;

}

}

**difference between driver.close and driver.quit methods?**

**Ex: write code**

**Driver.close()**

Close the browser window that the driver has focus of

String homeWindow = driver.getWindowHandle();

Set<String> allWindows = driver.getWindowHandles()

Iterator<String> windowIterator = allWindows.iterator();

while(windowIterator.hasNext())

{

String childWindow = windowIterator.next();

if (homeWindow.equals(childWindow)){

driver.switchTo().window(childWindow);

driver.close();

}

**Driver.quit()**

Calls dispose

public IWebDriver Driver;

public void SetupTest()

{

Driver = WebDriverFactory.GetDriver();

}

public void TearDown()

{

if (Driver != null)

Driver.Quit();

}

**common exceptions in selenium?**

**Ex: write code**

|  |
| --- |
| try { |

|  |
| --- |
| br = new BufferedReader(new FileReader("Data")); |

|  |
| --- |
| } catch(IOException ie) |
| {  ie.printStackTrace();  } |
|  |

|  |  |
| --- | --- |
|  |  |

**how to handle Ajax calls in selenium?**

**Ex: write code**

package Ajax;

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.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class AjaxExample

{

private String URL = "http://demos.telerik.com/aspnet-ajax/

ajax/examples/loadingpanel/explicitshowhide/defaultcs.aspx";

WebDriver driver;

WebDriverWait wait;

public void setUp()

{

driver=new FirefoxDriver();

driver.manage().window().maximize();

driver.navigate().to(URL);

}

public void test\_AjaxExample()

{

By container = By.*cssSelector*(".demo-container");

wait = new WebDriverWait(driver, 5);

wait.until(ExpectedConditions.presenceOfElementLocated(container));

WebElement noDatesTextElement = driver.findElement(By.*xpath*("//div[@class='RadAjaxPanel']/span"));

String textBeforeAjaxCall = noDatesTextElement.getText().trim();

driver.findElement(By.linkText("1")).click();

By loader = By.className("raDiv");

wait.until(ExpectedConditions.invisibilityOfElementLocated(loader));

WebElement selectedDatesTextElement = driver.findElement(By.*xpath*("//div[@class='RadAjaxPanel']/span"));

wait.until(ExpectedConditions.*visibilityOf*(selectedDatesTextElement));

String textAfterAjaxCall = selectedDatesTextElement.getText().trim();

Assert.assertNotEquals(textBeforeAjaxCall, textAfterAjaxCall);

String expectedTextAfterAjaxCall = "Thursday, January 01, 2015";

Assert.assertEquals(textAfterAjaxCall, expectedTextAfterAjaxCall);

}

}

we have webtable, need to click on second row from table.

when we click on child will be populated. first column in primary column for each row.

tell me steps to verify child form has proper data or not

Ex:

**How to assign the value to textbox other than sendkeys method?**

public class type

{

 public static void setattribute(WebElement element,String attributename,String Value)

{

WrapsDriver wrappedElement=(WrapsDriver)element;

JavascriptExecuter driver =(JavaScriptExecutor)wrappedElement.getWrapedDriver;

driver.executeScript(arguments[0].setAttribute(arguments[1],arguments[2]",element,

                                    attributeName,value);

}

**Selenium grid, how to execute scripts on multiple browser**

package myPackage;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.concurrent.TimeUnit;

 import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Parameters;

 public class BaseClass

{

    public static ThreadLocal<RemoteWebDriver> dr = new ThreadLocal<RemoteWebDriver>();

    @Parameters("myBrowser")

    public void beforeClass(String myBrowser) throws MalformedURLException{

 RemoteWebDriver driver = null;

if(myBrowser.equals("chrome"))

{

            DesiredCapabilities capability = new DesiredCapabilities().chrome();

            capability.setBrowserName("chrome");

            capability.setPlatform(Platform.WINDOWS);

            driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

        }

        else if(myBrowser.equals("firefox")){

            DesiredCapabilities capability = new DesiredCapabilities().firefox();

            capability.setBrowserName("firefox");

            capability.setPlatform(Platform.WINDOWS);

            driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capability);

        }

        setWebDriver(driver);

getDriver().manage().window().maximize();

        getDriver().manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS)

 }

public WebDriver getDriver() {

        return dr.get();

    }

  public void setWebDriver(RemoteWebDriver driver) {

        dr.set(driver);

    }

    public void afterClass()

{

        getDriver().quit();

        dr.set(null);

    }

 }

**TestParallel.java**

package myPackage;

import java.net.MalformedURLException;

import java.net.URL;

import org.openqa.selenium.By;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

 public class TestParallel extends BaseClass

{

    public void test\_01() throws InterruptedException, MalformedURLException

{

        try

{

            getDriver().get("http://www.w3schools.com/");

 getDriver().findElement(By.xpath("html/body/div[2]/div/a[4]")).click();

            Thread.sleep(10000);

            getDriver().findElement(By.xpath("//\*[@id='gsc-i-id1']")).sendKeys("test");

            Thread.sleep(5000);

}

        catch(Exception e){

            System.out.println(e);

        }

    }

}