**SELENIUM QUESTIONS**

1. **What are limitations in selenium web driver?**
2. Detailed test reports cannot be generated.

No centralized maintenance of objects/elements.

1. **Installing/configure selenium?**
2. **What are different ways of locating elements in selenium?**
3. By ID:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.id("element id")) |

By CLASS:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.className("element class")) |

By NAME:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.name("element name")) |

By TAGNAME:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.tagName("element html tag name")) |

By CSS Selector:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.cssSelector("css selector")) |

By Link:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.link("link text")) |

By XPath:

|  |  |
| --- | --- |
|  | in Java: driver.findElement(By.xpath("xpath expression"))  By PartialLinkText  In Java: driver.findElement(By.partialLinkText(“String Link Text”)) |

1. **Which is fastest way to identify elements in web page?**
2. IDs are the safest, fastest locator option and should always be your first choice

IDs should be unique in every page according to W3C website

Even if the DOM changes, if the ID is still there, then WebDriver can still locate it

Always try and get extra IDs added into the code, this makes testers life easier

Fastest locator as it uses the document.getElementById() javascript command which is optimized by many browsers.

1. **What is absolute path and relative path in Xpath?**

|  |  |
| --- | --- |
|  | 1. **Absolute Xpath**: It uses Complete path from the Root Element to the desire element.   Eg:/html/body/div[5]/div[2]/div/div[2]/div[2]/h2[1]  **Relative Xpath**: You can simply start by referencing the element you want and go from there.  Always Relative Xpaths are preferred as they are not the complete paths from the Root element. (//html//body) ..Beacuse in future any of the webelement when added/Removed then Absolute Xpath changes. So Always use Relative Xpaths in your Automation.  Eg:-.//\*[@id='answers']/h2[1]/a[1] |

1. **different types of waits or synchronization in selenium web driver**
2. Synchronization/ Waits can be achieved in many different ways :

Sleep method of Thread class

Page Load timeout

Script timeout

Implicit Wait

Explicit Wait

**sleep() command :**

The command halts the execution of the script for specified amount of milliseconds.

Example :

WebDriver driver = new FirefoxDriver();driver.get(“http://seleniumatfingertips.wordpress.com”);driver.manage().window().maximize();

WebElement homeLink = driver.findElement(By.linkText(“Home”));

homeLink.click();

try{

           Thread.sleep(5000);

} catch(Exception e){

            System.out.print(e);

}

This method is discouraged as even if the page/ web element is loaded, it instructs the browser to wait unnecessarily for the fixed amount of time.

**Page Load timeout :**

We can set the amount of time to wait for a page load to complete before throwing an error.

Example :

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

Once added in the script, the WebDriver instance waits for 20 seconds for every page to get loaded before throwing an exception. If the page is not loaded in 20 seconds of time, then it throws TimedOutException at run time.

**Script timeout :**

We can set the amount of time to wait for an asynchronous script to finish execution before throwing any error.

Example :

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

Once added in the script, the WebDriver instance waits for 20 seconds for every asynchronous script to get executed on the web page before throwing an exception.

**Implicit Wait :**

Implicit wait is a mechanism, which is used to notify WebDriver instance to wait for specific time, if any element is readily not available on the webpage.

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

The above line once added at the beginning of the automation script, the WebDriver will wait for 10 SECONDS for any element on the web page to be loaded before throwing any exception.

Once set, the timing will be set for the life of the WebDriver instance until and unless it is changed again. Meaning, using implicit wait command, we can say the WebDriver instance to wait for specified amount time before throwing NoSuchElementException, if it does not find any element on the web page immediately.

The synchronization mechanism is not associated with any particular web element, so it will be applicable for all the elements that our script interacts with, it is not highly appreciated.

**Explicit Wait :**

Explicit wait is a technique used to advise the WebDriver instance to stand by the execution till the time the certain condition is met or the maximum wait time is elapsed.

WebElement element = driver.findElement(By.linkText(“Home”));WebDriverWait wait = new WebDriverWait(driver, 20);

wait.until(ExpectedConditions.visibilityOf(element));

WebDriverWait wait = new WebDriverWait(driver, 20);

The above line sets maximum time the WebDriver instance will wait for the condition to become true.

wait.until(ExpectedConditions.visibilityOf(element));

The above line asks WebDriver instance to wait until the element is visible on the web page.

1. **How to save screen shots using selenium web driver?**
2. Driver.findElement(by.Xpath(“---“).sendkeys(“Learn Automation”);

TakeScreenShot ts=(TakeScreenShot)driver;// type casting

File source= ts.getScreenShotAs(OutputType.File);// saving screenshot as file

FileUtils.copyfile(Source, newfile(“path”);

System.out.println(“screenshot captured”);

1. **How to handle multiple windows in selenium web driver?**
2. public static void main(String[] args) {

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

WebDriver driver = new FirefoxDriver();

driver.get("<https://www.facebook.com/>");

driver.findElement(By.cssSelector("#terms-link")).click();

System.out.println(driver.getTitle());

    Set<String>ids=driver.getWindowHandles();

    Iterator<String> it=ids.iterator();

    String parentid = it.next();

    String childid = it.next();

    driver.switchTo().window(childid);

    System.out.println(driver.getTitle());

}

1. **How to launch webpage using chrome driver?**
2. System.setProperty("webdriver.chrome.driver","pathofchromedriver\\chromedriver.exe");
3. **What is desired capabilities in selenium web driver?**
4. public java.lang.String getBrowserName();
5. public void setBrowserName(java.lang.String browserName)
6. public java.lang.String getVersion()
7. public void setVersion(java.lang.String version)
8. public Platform getPlatform()
9. public Platform getPlatform()
10. The getCapability method of the DesiredCapabilities class can be used to get the capability that is in use currently in the system.
11. public java.lang.Object getCapability(java.lang.String capabilityName)
12. setCapabilityMethod

The setCapability() method of the Desired Capabilities class can be used to set the device name, platform version, platform name, absolute path of the app under test (the .apk file of the app(Android) under test), app Activity (in Android) and appPackage(java).

"setCapability method" in[Java](http://www.guru99.com/java-tutorial.html)has the below declarations:

setCapability : public void setCapability(java.lang.String capabilityName,boolean value)

setCapability :public void setCapability(java.lang.String capabilityName,java.lang.String value)

setCapability :public void setCapability(java.lang.String capabilityName,Platform value)

setCapability :public void setCapability(java.lang.String key,java.lang.Object

value.

1. **How to set language while opening website?**

System.setProperty(“webdriver.chrome.driver”,”D:/DollarArchive/chromedriver.exe”);  
ChromeOptions options = new ChromeOptions();  
options.addArguments(“–lang= sl”);  
ChromeDriver driver = new ChromeDriver(options);  
driver.get(“[http://google.co.in&#8221](#8221););

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

**popups:-**

**Set <String> set1=driver.getWindowHandles();  
Iterator <String> win1=set1.iterator();  
String parent=win1.next();  
String child=win1.next();  
driver.switchTo().window(child);**

**Dropdown:-**

**webElement month\_dropdown= driver.findElement(by.id(“month”).click();**

**select month\_dd= new select(month\_dropdown);**

**month\_dd.selectByIndex(3);**

**month\_dd.selectByValue(10);**

**month\_dd.selectByVisibleText(march);**

1. **Write code to verify any application login page is working or not?**

public void login() {

WebDriver driver = new FirefoxDriver();

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

WebElement id = driver.findElement(By.name("username"));

WebElement pass = driver.findElement(By.name("password"));

WebElementbutton= driver.findElement(By.xpath("/html/body/div/div/div[2]/div/form/p[3]/input"));

id.sendKeys("tuser991@yahoo.co.in");

pass.sendKeys("abc123");

button.submit();

1. **How to select items from dropdown/select box?**

**webElement month\_dropdown= driver.findElement(by.id(“month”).click();**

**select month\_dd= new select(month\_dropdown);**

**month\_dd.selectByIndex(3);**

**month\_dd.selectByValue(10);**

**month\_dd.selectByVisibleText(march);**

1. **How to know if checkbox is checked or not in webpage?**

Boolean x= driver.findElement(by.id(“myid”)).isselected();

If(x== false)

Driver.findElement(by.id(“myid”).click();

Boolean x= driver.findElement(by.id(“myid”)).isselected();

If(x== true)

Driver.findElement(by.id(“myid”).click();

1. **tell me code to pass values from parent window to child window**
2. **Write code to find out if all links are working or not?**

List<WebElement> links=driver.findElements(By.tagName("a"));

System.out.println("Total links are "+links.size());

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

{

WebElement ele= links.get(i);

String url=ele.getAttribute("href");

verifyLinkActive(url);

}

}

public static void verifyLinkActive(String linkUrl)

{

 try

        {

 URL url = new URL(linkUrl);

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

           httpURLConnect.setConnectTimeout(3000);

           httpURLConnect.connect();

           if(httpURLConnect.getResponseCode()==200)

           {

    System.out.println(linkUrl+" - "+httpURLConnect.getResponseMessage());

           }

if(httpURLConnect.getResponseCode()==HttpURLConnection.HTTP\_NOT\_FOUND)

           {

System.out.println(linkUrl+" - "+httpURLConnect.getResponseMessage() + " - "+ HttpURLConnection.HTTP\_NOT\_FOUND);

            }

        } catch (Exception e) {

        }

1. **Write code on how to use JavaScript executor?**

JavaScriptExecutor js=((JavaScriptExecutor)driver);

Js.executeScript(“document.getElementById(“Email”).value=’mounikakarlapudi@gmail.com’);

Js.executeScript(“document.getElementById(“Next”).click();

1. **Difference between assert and verify?**

**Assert: -** When an "assert" command fails then test execution will be aborted.

* Assert is best used when the check value has to pass for the test to be able to continue to run. Like a log in.

assertTrue(isElementPresent(By.cssSelector("div.bbMargin")));

* **Verify: -** When a "verify" command fails then test will continue executing and logging the failure.
* Verify is best used to check non critical things. Like hte presence of a headline element.

|  |  |
| --- | --- |
|  | if(driver.getPageSource().contains("Text - Testing with mounika"))  {  System.out.println("Text is Present");  }  else  {  System.out.println("Text is not Present");  } |

1. **Difference between driver.close and driver.quit methods?**
2. Driver.close();// it closes the current page

Driver.quit();// it closes entire browser

1. **Common exceptions in selenium?**

**ElementNotVisibleException**: Although an element is present in the DOM, it is not visible (cannot be interacted with). E.g. Hidden Elements – defined in HTML using type=”hidden”.

**ElementNotSelectableException**: Although an element is present in the DOM, it may be disabled (cannot be clicked/selected).

**InvalidSelectorException**: Selector used to find an element does not return a WebElement. Say XPath expression is used which is either syntactically invalid or does not select WebElement.

**NoSuchElementException**: WebDriver is unable to identify the elements during run time, i.e. FindBy method can’t find the element.

**NoSuchFrameException**: WebDriver is switching to an invalid frame, which is not available.

**NoAlertPresentException**: WebDriver is switching to an invalid alert, which is not available.

**NoSuchWindowException**: WebDriver is switching to an invalid window, which is not available.

**StaleElementReferenceException**: The referenced element is no longer present on the DOM page (reference to an element is now Stale). E.g. The Element belongs to a different frame than the current one OR the user has navigated away to another page.

**SessionNotFoundException**: The WebDriver is performing the action immediately after ‘quitting’ the browser.

**TimeoutException**: The command did not complete in enough time. E.g. the element didn’t display in the specified time. Encountered when working with waits.

**WebDriverException**: The WebDriver is performing the action immediately after ‘closing’ the browser.

1. **How to handle Ajax calls in selenium?**
2. **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**

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

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

public class WithoutSendKeys {

public static void main(String[] args) {

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

WebDriver driver=new ChromeDriver();

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

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

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

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

((JavascriptExecutor)driver).executeAsyncScript("argument[0].value='mounikakarlapudi24'",send);

}

}

1. **Selenium grid, how to execute scripts on multiple browser**
2. **Write code for drag/drop in selenium?**
3. Action.dragAndDrop.perform();

Thread.sleep(3000);

Driver.quit();

1. **Write code for right click in selenium?**

action.contextClick(drag);

1. **Write code for scroll to specific element**

**1.Using javascript Executor:**

public class ScrollTestCase {

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

// load browser

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

WebDriver driver=new ChromeDriver();

// maximize browser

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

// Open Application

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

// Wait for 5 second

Thread.sleep(5000);

// This will scroll page 400 pixel vertical

((JavascriptExecutor)driver).executeScript("scroll(0,400)");

2. **scrolldown into view**

// Create instance of Javascript executor

JavascriptExecutor je = (JavascriptExecutor) driver;

//Identify the WebElement which will appear after scrolling down

WebElement element = driver.findElement(By.xpath(".//\*[@id='mCSB\_3\_container']/p[3]"));

// now execute query which actually will scroll until that element is not appeared on page.

je.executeScript("arguments[0].scrollIntoView(true);",element);