## Selenium :

**Q1) Diff^n Betn implicit & explicit wait**

**Implicit wait :** It is a type of wait which waits for a specified time while locating an element before throwing NoSuchElement Exception. By default selenium tries to find elements immediately when required without any wait, so it is good to use implicit wait This wait is applied to all the elements of the current driver instance

```
driver.manage().timeouts().implicitlyWait(5,TimeUnit.SECONDS);
```

**Explicit wait :** It is a type of wait which is applied to a particular web element until the expected condition specified is met.

```
Webdriver_wait wait = new WebDriverWait(driver,10);
WebElement element = wait.until(ExpectedConditions.elementsToBeClicked(By.id("elementId")));
```

**Q3) Diff^n - getWindowHandle & getWindowHandles**

**getWindowHandle() :** It is used to get the address of the current browser where the control is and return type is string

**getWindowHandles() :** It is used to get the address of all the open browser and its return type is Iterator<String>

**Q6) Diff^n - getAttribute() and getText()**

**getAttribute() :** This method is useful to read elements attribute value like id, name, type

**getText() :** This method is used to read text from element or alert

**Q8) Diff^n - Absolute xpath & Relative xpath.**

**Absolute xpath :** is the full path starting from root node and ends with desired dependant elements nodes. It will start using single forward slash

**Relative xpath :** Instead of starting from root node, Relative xpath starts from anywhere in betn node or current element node (last node of element). It will start using double forward slash (//)

**Q2) What is Fluent wait**

**Fluent wait :** It is a type of wait in which we can also specify polling interval (intervals after which driver will try to find the element) along with the maximum timeout value

```
Wait wait = new FluentWait(driver)
.withTimeout(20, SECONDS)
.pollingEvery(5, SECONDS)
.ignoring(NoSuchElementException.class);
WebElement textBox = wait.until(new Function
{
public WebElement apply(WebDriver driver)
{return driver.findElement(By.id("textBox
}});
```

**Q4) Diff^n findElement and findElements()**

**findElement() :** It returns only one web ele. It will find the first matching element. It will show NoSuchElement exception.

**findElements() :** It returns list of web elem. It will find all the matching elements It will not show any exception, returns empty list.

**Q5) Diff^n - driver.get() & driver.navigate()**

**driver.get() :** This method generally used for open URL. It will wait till the whole pages gets loaded

**driver.navigate() :** This method is generally used for navigate to URL, navigate back, navigate forward, refresh the page.

**Q7) Diff^n / and //**

**Single slash (/) :** Start execution selection from the document node (Root path). It allows you to create 'absolute' path expressions.

**Double slash (//) :** Start selection anywhere in the document (current path). It enables to create 'relative' path expressions

**Q8) Diff^n Betn close and quit command**

**driver.close() :** Used to close the current browser having focus

**driver.quit() :** used to close all the browser instances

**Q9) Diff^n - xpath and css selector.**

Using xpaths we can traverse up in the doc ie we can move to parent element.

Using CSS selector we can only move down-

**Q10) How to handle alerts in Selenium**

-> In order to accept or dismiss an alert box the alert class is used. This requires first switching to the alert box and then using accept() or dismiss() command as the case may be

```
Alert alert = driver.switchTo().alert();
// To accept the alert alert.accept();
Alert alert = driver.switchTo().alert();
// To cancel the alert box alert.dismiss();
```

**Q12) How do you drag and drop in Selenium**

-> Using Action class

```
Action act = new Actions(driver);
Jd")); act.dragandDrop(sourceElement, target
Element).build().perform();
```

**Q14) How to handle frames ?**

```
driver.switchTo().frame("classFrame");
S.o.P(driver.findElement(By.linkText("Tree")).
getText());
driver.findElement(By.linkText("Tree")).click();
driver.switchTo().defaultContent();
```

**Q16) How to handle Scroll Bars ?**

```
JavascriptExecutor jsx = (JavascriptExecutor)
driver;
jsx.executeScript("window.scrollBy(0,4500)
"") // scroll down
Thread.sleep(3000);
jsx.executeScript("window.scrollBy(4500,0)
"") // scroll up
```

**Q18) How to handle Links ?**

```
driver.findElement(By.xpath("//a[text()=
'SIGN-ON']")).click();
driver.findElement("By.partialLinkText("REG")).
click();
driver.findElement("By.linkText("Home")).click();
```

**Q20) Commonly used TestNG annotations**

- @Test
- @BeforeMethod
- @AfterMethod
- @BeforeClass
- @AfterClass
- @BeforeTest
- @AfterTest
- @BeforeSuite
- @AfterSuite

**Q21) Common assertions provided by testNG**

- assertEquals
- assertNotEquals
- assertFalse
- assertTrue
- assertNotNull
- Fail

**Q22)**

-> Using In TestNG

@Test(priority=1)

**Q11) How to handle dropdowns in Selenium**

Using select class -

```
Select countriesDropDown = new Select(driver
findElement(By.id("countries")));
dropdown.selectByVisibleText("India");
//or using index of the option starting from
dropdown.selectByIndex(1); // or using
Value attribute
dropdown.selectByValue("Ind");
```

**Q13) How to scroll down a page using Javascript in Selenium.**

-> by using window.scrollBy() function. eg.

```
((JavascriptExecutor)driver).executeScript(
"window.scrollBy(0,500)");
```

**Q15) How to handle Iframes ?**

```
List<WebElement> FrameList = driver.find
Elements(By.xpath("//iframe"));
Int numofFrames = FrameList.size();
S.o.P("No of Frames:" + numofFrames);
S.o.P(driver.switchTo().frame(3).getTitle());
```

~~Q17) How to Assert ?~~

**Q17) How to capture screenshot ?**

We can use selenium webdriver "TakesScreenshot" method to capture screenshot. Java class file will be used to store screenshot in your systems local drive

```
file srcfile = ((TakeScreenshot)driver).get
ScreenshotAs(outputType.FILE);
FileUtils.copyFile(srcfile, new File("E:\SELF
AUTOMATN\Webdriver_auto\screenshots\Booked
Details.png"));
```

**Q19) What is action class in WebDriver**

-> Action class is used to control the actions of mouse

Method name of action class to build up actions chain is "build()".perform()

```
WebElement UserName = driver.findElements
(By.name("User Name"));
Actions builder = new Actions(driver);
```

**Q22) How can we set priority of test cases in TestNG ?**

priority parameter in @Test annotation TestNG we can define priority of testcases

Q23) What is TestNG?

TestNG (NG for Next Generation) is a testing framework that can be integrated with selenium or any other automation tool to provide multiple capabilities like assertions, reporting, parallel test execution etc

Advantages -

1) It provides diffn assertions that helps in checking the expected & actual results
2) provides parallel execution of test methods
3) We can define dependancy of one test method over other in TestNG.
4) Assign priority to test methods in seleni.
5) Allows grouping of test methods into test groups

Q24) Methods supported alert object

1) Accept(): It performs the click operation on ok button which is available on alert
2) Dismiss: It won't click on the cancel button, It closes the alert.
3) getText: In order to capture text content available on alert we can use this method
4) sendKeys: We can enter the text content

Q25) Methods supported by driver object

1) get(): This method opens the specified URL in the current browser window.
driver.get("www.google.com");
2) getcurrenturl(): This method is used to get string URL of the current web page loaded in the opened browser
driver.getcurrenturl();
3) getpage source(): This method is used to get the page source code of the current loaded web page.
driver.getpagesource();
4) gettitle(): This method is used to get the title of the current web page.
driver.gettitle();

List - Types of methods in driver object

1) get()
2) getcurrentURL()
3) getpagesource()
4) getText()
5) getTagName()
6) findElement()
7) findElements()
8) getattribute()
9) getcssvalue()
10) getsize()
11) getwindowHandle()
12) getwindowHandles()
13) close()
14) quit()
15) manage()

Q26) List out the methods supported by the Webelement object

1) click()
2) sendkeys()
3) submit()
4) isDisplayed()
5) isSelected()
6) isEnabled()
7) getlocation()
8) clear()
9) getText()
10) getAttribute()

Q27) List out the methods supported by the action class object

→ Action class is an ability provided by Selenium for handling keyboard and mouse events.

Methods under keyboard action class:

1) sendkeys() - used to send series of keys
2) keydown() - perform keypress without release
3) keyUp() - perform keyrelease

Methods under MouseActions:

1) click()
2) doubleclick() - perform double click on ele.
3) clickandhold() - Long click on mouse
4) Dragand drop() - Drag from one point & drop to another
5) moveToElement() - shifts the mousepointer to center of ele
6) contextclick() - perform right click on

Q28) List Exception classes available

1) NoSuchElement Exception
2) NoSuchWindow
3) NoSuchFrame
4) NoAlertPresent
5) Invalid Selector
6) ElementNotVisible
7) ElemenNotselectable
8) TimeOut
9) NoSuchsession E.

Q29) Describe How to execute Javascript

JavaScript Executor is an interface that helps to execute javascript through S.webdr
It provides two methods

1) execute script
2) execute AsyncScript (To run JS on the selected window or current page)

Selenium supports JS executor. There is no need for extra plugin or add on, we just need to import.

[org.openqa.selenium.JavascriptExecutor]

Syntax:-

```
JavascriptExecutor js=(JavascriptExecutor)
                        driver;
js.executescript(script, Arguments);
```

Script: This is the JS that need to execute
Arguments: It is the arguments to script. It's optional

i) import the package
ii) create reference
iii) Call Javascript Executor Methods.

Q31) How to Handle the Frames?

In order to work or Identify GUI elements in a frame by using seleni approach we have to switch to that respective frame

→ We can handle frames in an appl

1) By using Index of the frame.
SwitchTo().frame(frame Number)
2) By using name of the frame
SwitchTo().frame(frame name)
3) By using webelement of the frame
SwitchTo().frame(frame webelement)

Q30) Explain Page Object Model of writing testscripts & advantages

The page object model eliminates the FindElement method. The POM works based on the principle of encapsulation. Here the webelements have declared or privated based on the getter method we are returning the webelement.

In POM the webelements have derive based on the ID and name attribute of the element in the DOM structure. Suppose in the HTML DOM structure if any elements has attribute ID or name based on it respective values we can create an webElement.

If the element in the HTML Dom's structure if they don't have ID and attribute the element we can be created based on @FindBy annotations This can be achieved by "pagefactor. initElement(obrowser.this);

initElement is a static method, it accepts driver object of a current class as parameter based on this execution only the above mentioned approach we can create the webelement

Advantages:

i) It provides reusability of webElement
ii) It reduces the code duplication and improves test maintenance.
iii) Make code readable.
iv) Code becomes less and optimized

Q32) How to select the items from the dropdown?

1) Select by index(int index)
2) Select by value(String value)
3) Select by visibleText(String Text)

1) Select by index: This method selects the dropdown option by its indexno. We provide an integer value as the index no as an argument
Syntax - selectByIndex(int args)

2) select by value: by its value we provide string value as an argument
SelectByvalue(string arg)

3) Select by visibleText: by its text
selectBy
SelectByvisibleText(String arg)
se.selectByvisibleText("white");

Q33) How do you display the items available in DropDown

using getoptions()

select class provides the following methods to get the optn of dropdown

1) getoptions()
2) get firstselected option()
3) getselectedoptions()

1) getoptions(): used to get all the options in a dropdown or multi select box.

syntax: getoptions(); List<webele>

---

Q34) What is log4j? What are the logger options are available?

Log4j is used to track the logs.

Log4j is a tool to help the programmer output log stmts to a variety of op targets

Methods & descrn

i) public void debug(object message)
It prints messages with the level Level.DEBUG

ii) public void error(object message)
Level.ERROR

iii) public void fatal(object message)
Level.FATAL

iv) public void info(object message)
Level.INFO

v) public void warn(object message)
Level.WARN

vi) public void trace(object message)
Level.TRACE

Log4j has 3 main components

1) Loggers: Responsible for capturing logging informan

2) appenders: Responsible for publishing Logging infon to various preferred destn

3) Layouts: Res. for formatting logging info in diff styles

Log4j is an API. It is a Apache proj
the main purpose is to track the logs during the execn of testscripts

Q35) How to Handle popup browsers?

By default selenium does not support child browser or tabbed browsers which are opened from the current object ses

In order to handle child browsers Selenium supports the following two methods.

1) getwindowHandle()
2) getwindowHandles()

1) getwindowHandle(): It returns windaw handle of parent browser in string repren

2) getwindowHandles(): It returns window handler of the both parent and child browser in set() of string representation

---

Q36) Diffn - Selenium and testng.

Selenium is a testing framework to test specially the UI of the applin how it behaves on browser.

TestNG is a testing framework to test the unit, functional, integran testings of the applin

---

Q37) Diffn - Selenium & Cucumber

| Selenium | Cucumber |
|---|---|
| - It is an auton tool for web apps | - It is an auton tool for behaviour-drive development. |
| - It executes UI tests | - It does acceptance T. |
| - script orean is complex | - more simple. |

---

Q38) Cucumber Annotations

1) @Given - Describes the pre-requisite for the test to be executed
2) When - Defines the trigger point for any test scenario execution
3) Then - Holds the expected result for the test to be executed
4) And - provides logical AND conn betn any two statements. Can be used in conjunction with GIVEN, WHEN & THEN statements
5) But - It signifies logical OR conn betn any two stmts.
6) Scenario - Details about the scenario under the test needs to be captured after the keyword "scenario".

---

Concepts of OOPS in Selenium Auton Framework:

1) Abstraction 2) Interface 3) Inheritance
4) Polymorphism → method overloading & riding
5) Encapsulation

Other selenium auton FW concepts

1) Webelement 2) WebDriver 3) findBy
4) findelement.

1) Abstraction: is the methodology of hiding the implen of internal details and showing the functionality to the users. In POM design pattern, we write locators such as id, name, xpath etc and methods in page class we utilize these locators in tests but we can't see the Implin of the methods Literally we hide the implin of locators from the tests

2) Interface: — WebDriver driver=new FirefoxDriver();
WebDriver itself is an interface. It means