-MR-

---

1. What is Selenium..?

- It is a Open source tool for the Automation testing
- It is used to do the Automation testing for Web based application
- It supports different programming language like Java, Python C#
- It supports different web browsers like Google Chrome, Opera-mini, Microsoft Edge
- It can run at different Operating system like Window, Linux, Mac

---

2. How to download the chrome driver..?

- Search chrome driver for selenium in the google chrome
- Download the **chromedriver.exe** based our current chrome version
- To know the current version of **chromedriver.exe**, Setting--> About Chrome --> Version name detail will available

---

3. What is driver..?

- Browser is technically called as driver in selenium
- Web browser is technically called as WebDriver in selenium
- Chrome browser is technically called as ChromeDriver in selenium

---

4. How to launch the Browser..?

```
System.setProperty("webdriver.chrome.driver","chromedriver.exe");

WebDriver driver=new ChromeDriver();
```

---

5. How to maximize the browser…?

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

---

6. How to minimize the browser…?

```
driver.manage().window().window().minimize();
```

---

7. How to close the launched browser…?

We can close the launched browser by using close().

```
driver.close();
```

-MR-

---

8. What is sleep()..?

The sleep() is used to stop the execution of programme for the particular time.

```
Thread.sleep(4000);
```

- We should pass the time duration as argument to the sleep method.
- We have to call the sleep() method from Thread class
- The sleep() causes the **InterruptedException** exception
- We should manually throws the Exception in the method name, Else we can handle it by using try-catch

---

3. What is get()..?

- The get() is used to hit the link in the browser.
- We should pass the link value as argument in the method as String value.

```
driver.get("www.google.com");
```

---

4. What is findElement()..?

- The findElement() method is used to find the single Web Element in the HTML webpage.
- We should pass the locators to the findElement() as a argument.

```
driver.findElement(By.locator("locator-value"));

driver.findElement(By.xpath("//*[@id=username]"))
```

---

5. What is findElements()..?

- The findElements() method is used to find the multiple Web Element in the HTML webpage.
- We should pass the locators to the findElement() as a argument.

```
driver.findElement(By.locator("locator-value"));

driver.findElements(By.xpath("//*[@id=username]"))
```

-MR-

---

6. What are the different between findElement() and findElements()..?

findElement() method:

- This command is used to access any single element on the web page
- It will return the object of the first matching element of the specified locator
- It will throw NoSuchElementException when it fails to identify the element

findElements() method:

- This command is used to uniquely identify the list of web elements within the web page.
- The usage of this method is very limited
- If the element doesn't exist on the page then, then it will return value with an empty list

---

7. What are the locators..?

- The locator is used to locate the web element in the HTML web page.
- The locator is classified into different types
    1. By ID
    2. By Name
    3. By Tag Name
    4. By Class name
    5. By Xpath

1. By ID: driver.findElement(By.id("JournalDev"));

2. By Name: driver.findElement(By.name("JournalDev"));

3. By Tag Name: driver.findElement(By.tagName("JournalDev"));

4. By Class Name: driver.findElement(By.className("JournalDev"));

5. By Xpath: driver.findElement(By.xpath("JournalDev"));

-MR-

---

8. What is the Xpath..?

- Xpath is used to locate an WebElement in the HTML Web page
- Xpath is classified into 2 different types
    1. Relative xpath
    2. Absolute xpath

1. Absolute xpath:

- The simplest XPath locator example in Selenium is to provide the absolute path of an element in the DOM structure.
- It is a xpath from Root of the HTML page.
- It is mentioned by using single slash(\)

Example: /html/body/div[1]/div/div[2]/header/div/div[2]/a/img

2. Relative xpath:

- The releative xpath locator is start from middle of the html page.
- It is mentioned by using double slash(\\)

Example: //*[@id="block-perfecto-main-menu"]/ul/li[6]/a

---

9. What are the way to get the Relative xpath..?

We can get the relative xpath by using different ways

a. Basic xpath
b. Contains()
c. Using OR, AND
d. Starts-with
e. Text()

a. Basic xpath:

This is the basic method of getting the xpath from the web page

Xpath=//tag[@attribute='value'];

XPath=//div[@name='username']

b. Contains():

The contains() is used to find the element when the text is partially matched with the element's text.

Xpath=//tag[contains(@attribute='value')];

XPath=//div[contains(@id,'Name')]

c. Using OR, AND Condition:

- In OR expression, two condition are used
- Any one of the condition should be true or may be both condition will true

Xpath=//tag[@attribute1='value1' or @attribute2='value2'];

-MR-

XPath=//div[@id='FirstName' or @name='LastName']

- In AND expression, two condition are used
- Both of two condition should be true

Xpath=//tag[@attribute1='value1' and @attribute2='value2'];

XPath=//*[@id='FirstName' and @name='FirstName']

d. Starts-with():

The Starts-with() is used to find the element when the text of the element is starts with some text.

Xpath=//tag[starts-with(@attribute='startvalue')];

XPath=//input[starts-with(@name,'pass')]

e. text():

The text() is used to locate an elements based on exact text of a web element

Xpath=//tag[text()='textValue')];

XPath=//input[text()='username']

---

10. What is click()..?

The click() method is used to click the web element in the HTML Webpage.

WebElement loginButton = driver.findElement(By.xpath("//*[@name=login]"));

loginButton.click();

---

11. What is sendKeys()..?

- The sendKeys() method is used to enter any value to the text field in the HTML Webpage.
- We need to pass the value as an argument to the send keys method by String value.

WebElement username= driver.findElement(By.xpath("//*[@name='username']"));

username.sendKeys("administrator");

---

12. What is clear()..?

- The clear() is used to clear the existing text in the mentioned text field.

WebElement username= driver.findElement(By.xpath("//*[@name='username']"));

username.clear();

-MR-

---

13. How to choose the option from drop-down box…?

a) Find the element and save it in the WebElement variable

b) Create an object for the select class

c) Pass the WebElement variable to the Select class

d) Call selectByVisibleText() to select the option

e) Pass the option value as String to the selectByVisibleText().

Steps:

```
WebElement   userGroup = driver.findElement(By.xpath("//*[@name='usergroup']"));

Select userGroupSelect = new Select(userGroup );

userGroupSelect.selectByVisibleText("Administrator");
```

---

14. What are the Waits and it's types…?

- The Waits are command in the Selenium
- During automated testing of websites, issues may occur due to variations in time lag for loading the web elements.
- Wait command help observe and troubleshoot these issues.
- When a page loads on a browser, various web elements on it with may load at different time intervals.
- Wait commands direct a test script to pause for a certain time before throwing an Exception

---

15. What are the types of Wait..?

The wait is classified into 3 different types

- Implicit Wait
- Explicit Wait
- Fluent Wait

-MR-

---

16. What is implicit Wait..?

- The implicit wait will wait for the particular time to load the Web element.
- We should pass the time detail.
- Implicit Wait directs the Selenium WebDriver to wait for a certain measure of time before throwing an exception.
- Once this time is set, WebDriver will wait for the element before the exception occurs.
- It's default wait time is 0 seconds
- We can declare the implicit wait for overall action in a same class

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

---

17. What is Explicit Wait:

- By using the Explicit Wait command, the WebDriver is directed to wait until a certain condition occurs before proceeding with executing the code.
- Setting Explicit Wait is important in cases where there are certain elements that naturally take more time to load.
- If one sets an implicit wait command, then the browser will wait for the same time frame before loading every web element. This causes an unnecessary delay in executing the test script.
- In order to declare explicit wait, one has to use "**ExpectedConditions**".

Expected Conditions Example:

- alertIsPresent()
- elementSelectionStateToBe()
- elementToBeClickable()
- elementToBeSelected()
- frameToBeAvaliableAndSwitchToIt()
- invisibilityOfTheElementLocated()
- presenceOfElementLocated()
- textToBePresentInElement()

Syntax:

```
WebDriverWait wait = new WebDriverWait(driver,30)

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

wait.until(ExpectedConditions.visibilityOfElementLocated(menu));
```

-MR-

---

18. What is Fluent Wait:

- Fluent Wait in Selenium marks the maximum amount of time for Selenium WebDriver to wait for a certain condition (web element) becomes visible.
- It also defines how frequently WebDriver will check if the condition appears before throwing the "Exception".
- While using Fluent Wait, it is possible to set a default polling period as needed. The user can configure the wait to ignore any exceptions during the polling period.

Syntax:

```
Wait wait = new FluentWait(driver)
.withTimeout(timeout, SECONDS)
.pollingEvery(timeout, SECONDS)
.ignoring(Exception.class);
```

---

19. How to upload the file using selenium…?

- We can upload the file by using **sendKeys()** method.

Steps:

1. Declare the file path as a String value
2. Find the element and store it in a variable
3. Use **sendkeys**() method to upload the file
4. Pass the file location to the **sendKeys**()

```
String fileName="D:\\Users\\photo.png";
WebElement fileUpload = driver.findElement(By.id('users'));
fileUpload.sendKeys(fileName);
```

---

20. How to refresh the page…?

- We can refresh the page using navigate() and refresh() in selenium.

```
driver.navigate().refresh();
```

-MR-

---

21. How to Hit the link…?

We can hit the link by using 2 ways

- get()
- navigate().to();

1. get():

driver.get("www.google.com");

2. navigate().to():

driver.navigate().to("www.google.com")

---

22. What is the purpose of quit()…?

The quit() method is used to close the all launched browser at a same time.

driver.quit();

---

23. How to get the title of the launched page..?

- We can get the title of the page by using **getTitle()** method.
- It will return the title of the page in the String format.

String titleOfThePage = driver.getTitle();

---

24. How to get the URL of the launched page..?

- We can get the URL of the current page by using **getCurrentURL()** method.
- It will return the URL of the current page in the String format.

String URLOfThePage= driver.getCurrentURL();

-MR-

---

25. How to get the Unique ID of the individual tab..?

- Every tab has Unique id in the selenium.
- We can get the Unique id by using **getWindowHandle()** method.
- It will return the Unique id of the tab in the String format.

```
String uniqueID = driver.getWindowHandler();
```

---

26. How to get the Unique ID of the multiple tab..?

- Every tab has Unique id in the Selenium.
- We can get the Unique id of the all launched tab/browser by using **getWindowHandles()** method.
- It will return the Unique ID of the all tab in the Set format.
- We should store it in the Set datatype.

```
Set uniqueID = driver.getWindowHandles();
```

---

27. How to switch the control from one tab to another tab..?

- If we open the multiple tab in the single browser in the same time, we can not control other than default tab.
- So, we can control other than default tab by using switchTo() method.
- We should pass the unique id of the tab which we want to switch the control.

```
Set uniqueID = driver.getWindowHandles();
String secondTab = uniqueID.get(1);
driver.switchTo(secondTab);
```

---

28. What is an Active Element..?

- The element which is currently focused in the launched page.

Example:

If we hit the google search link (https://www.google.co.in/), the search text field is an active element.

---

29. How to handle the active element of the current launched webpage..?

- We can handle the active element of the webpage by using switchTo() method.

```
driver.switchTo().activeElement().sendKeys("abc");
```

-MR-

---

30. What is Page Object Model..?

- Page Object Model, also known as POM.
- POM is a design pattern in Selenium that creates an object repository for storing all web elements
- It is useful in reducing code duplication and improves test case maintenance.
- In Page Object Model, consider each web page of an application as a class file. Each class file will contain only corresponding web page elements. Using these elements, testers can perform operations on the website under test.

Advantages of POM:

- Helps with easy maintenance
- Helps with reusing code
- Readability and Reliability of scripts:

---

31. How to find the element of the webpage in Page Object Model…?

- We can find the element of the webpage by using **@FindBy** annotation.

```
class Login
 {
      @FindBy(xpath="//*[@id='username']")
      public static WebElement userName;

      @FindBy(xpath="//*[@id=pwd]")
      public static WebElement password;
 }
```

---

32. How to access the WebElement from POM class..?

- We can access the WebElement from P0M class by using PageFactory class.

```
PageFactory.initElements(driver, Login.class);
Login.userName.sendKeys("administrator")
Login.password.sendKeys("123")
```

-MR-

---

33. How to take the screenshot…?

- We can take the screenshot by using FileHandler class

Step Code:

- Create an object for TakesScreenshot class
- Covert driver into TakesScreenshot class by Upcasting
- Get the screenshot output as File type
- Store the Output file into File datatype of variable as source file
- Create an object for define the destination file path.
- Copy the source file into destination file by using copy method in FileHandler class.

```
TakesScreenshot screenshot = (TakesScreenshot)driver;
File source =  screenshot.getScreenshotAs(OutputType.FILE);
File destination =  new File("screen.png");
FileHandler.copy(source , destination );
```

---

34. How to perform drag and drop in selenium…?

- We can drag and drop the WebElement by using Action class
- We should pass the source and destination webElement to the dragAndDrop() method.

Step Code:

```
WebElement from = driver.findElement(By.xpat(“//*[@id=’name’]”));
WebElement to = driver.findElement(By.xpat(“//*[@id=id]”));
Action action = new Action(driver);
action.dragAndDrop(from, to).build().perform();
```

-MR-

---

35. How to perfom scrolling action in Selenium..?

- We can perform scrolling action by using **JavascriptExecutor** class.

a) Scroll down Verticall by using pixels:

```
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.scrollBy(0,1000)");
```

b) Scroll down the web page by the visibility of the element:

```
JavascriptExecutor js = (JavascriptExecutor) driver;
WebElement submit = driver.findElement(By.xpath("//*[@id='name']"));
js.executeScript("arguments[0].scrollIntoView();", submit );
```

c) Scroll down to bottom of the web page:

```
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.scrollTo(0, document.body.scrollHeight)");
```

---

36. How to get value of an attribute from the Webpage..?

- We can get the value of an attribute by using **getAttribute()** method.
- It will return the value in String format.

** Your answer **

Code Steps:

```
WebElement username = driver.findElement(By.xpath("//*[@id='username']"));
String indexValue = username.getAttribute('index');
```

---

37. How to double click the element in the Webpage..?

- We can double click the element by using Action class.

Code Steps:

- Fine the element and store it to WebElement variable
- Create an Object for the class named Action.
- Pass the driver to the Action class
- Call doubleClick() method from the object of the Action class and pass the web element.
- Call the perform() method to the object

```
WebElement submitButton = driver.findElement(By.xpath("//*[@id='name']"));
Action action = new Action(driver);
action.doubleClick(submitButton).perform();
```