

# Selenium Interview Questions |

## Page 3

---

Question 21:

How can we submit a form using Selenium?

You can submit form in Selenium using below code snippet

```
driver.findElement(By.id("id")).submit();
```

### Selenium

[Start Discussion](#)

Question 22:

What is difference between findElement() & findElements()?

**findElement():**

1. It finds the element on the current page as per the given element locator mechanism.
2. It returns only the single element.

**Syntax: WebElement element = webdriver.findElement(By.id("item\_id"));**

1. If the element not found on the page then it will throw NoSuchElementException.

**findElements():**

1. It finds all the element on the current page as per the given element locator mechanism.
2. It returns the list of elements.

**Syntax: List<WebElement> element = webdriver.findElements(By.id("item\_id"));**

1. If any element not found on current page as per the given element locator mechanism, It will return empty list.

### Selenium

[Start Discussion](#)

### Question 23:

What is the difference between click() & submit() in Selenium?

`Click()` & `Submit()` both are used to click button in web page.

Selenium Web Driver provides the special method `submit()` to submit any form. `Submit()` method works same as clicking on submit button.

**Submit():** We can use `submit()` method for only submit form. There are some restrictions that are given below:-

1. Element's `type="submit"` & button should be inside `<form>` tag, then only `submit()` method will work.
2. If element's `type="button"`, then `submit()` will not work.
3. If button is outside of `<form>` tag, then `submit()` will not work.

**Click():** We can use `click()` method to click on any button. There is no restrictions for click.

1. Element's `type="button"` or `type="submit"`, then click method will work for both type.
2. If button is inside the `<form>` tag or outside the `<form>` tag, then click method will work.

## Selenium

[Start Discussion](#)

### Question 24:

What are different types of annotation which are useful?

Different types of annotations which are useful are listed below-

1. Test
2. Before
3. After
4. Ignore
5. BeforeClass
6. AfterClass
7. RunWith

## Selenium

[Start Discussion](#)

### Question 25:

What are the different types of waits available in WebDriver?

Selenium WebDriver provides the two types of wait:

1. Implicit Wait
2. Explicit Wait

**Implicit Wait:** Implicit provides the facility to poll the DOM for a certain amount of time when trying to find the element if they are not immediately. Default time setting is 0.

**Syntax:**

```
WebDriver driver = new FirefoxDriver();

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

driver.get("https://www.modernpathshala.com/");

WebElement element = driver.findElement(By.id("Element_Id"));
```

**Explicit Wait:** An explicit waits is code you define to wait for a certain condition to occur before proceeding further in the code. The worst case of this is `time.sleep()`, which sets the condition to an exact time period to wait.

**Syntax:**

```
WebDriver driver = new FirefoxDriver();

driver.get("https://www.modernpathshala.com/");

WebElement myDynamicElement = (new WebDriverWait(driver, 10)).until(ExpectedConditions.presenceOfElementLocated(By.id("myDynamicElement")));
```

## Selenium

[Start Discussion](#)

### Question 26:

What is difference between `driver.close()` and `driver.quit()`?

**`driver.close()`:**

It is used to close the browser or page currently which is having the focus.

**`driver.quit()`:**

It is used to shut down the web driver instance or destroy the web driver instance means close all the browser windows. Basically it calls the `driver.dispose` method which in turn close s all the browser windows and ends the web driver session gracefully.

## Selenium

[Start Discussion](#)

### Question 27:

#### How to capture screenshot in WebDriver?

We can take the screenshot in WebDriver using this command.

```
File sourceFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);  
FileUtils.copyFile(sourceFile, new File("D:\\my_screenshot.png"));
```

#### Selenium

[Start Discussion](#)

### Question 28:

#### What is the TextNG?

TextNG is a testing framework for Java programming language and designed to simplify a broad range of testing needs, from unit testing to integration testing (testing entire system made of several classes and packages). In other words, TextNG is designed to cover all categories of testing like unit, end-to-end, functional, integration etc. where **NG** means **Next Generation**. It is an open source framework and inspired by JUnit. It is similar to Junit and designed to be better than Junit.

#### Selenium

[Start Discussion](#)