

Selenium Java

Interview Questions And Answers



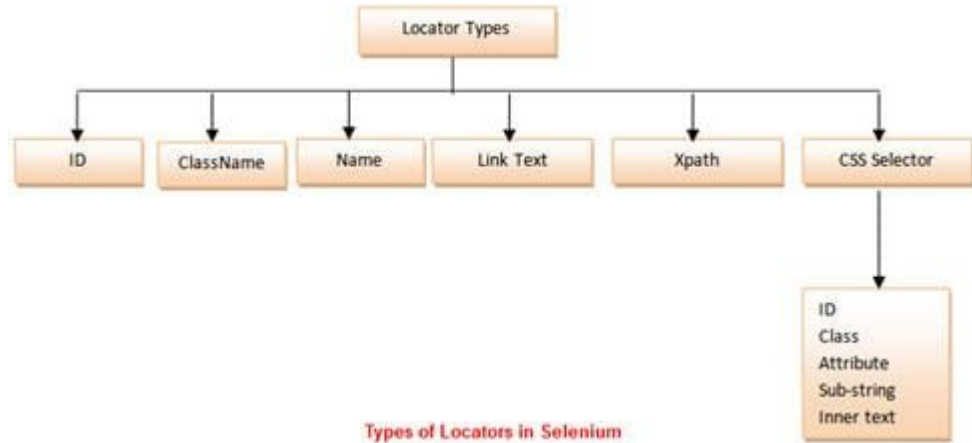
Pramod Dutta
Sr. SDET

Cracking Automation Tester Interview Series : **Part #2**



Question #1 : How will you find an element using Selenium?

- ID
- Name
- Tag
- Attribute
- CSS
- Linktext
- PartialLink Text
- Xpath etc



<https://www.guru99.com/using-contains-sibling-ancestor-to-find-element-in-selenium.html>

Question #2 : What is the difference between verify and assert commands?

Assert: In other words, the test will be terminated at the point where the check fails.

Verify: In verification, all the commands are going to run guaranteed even if any of the tests fail.

Assert command	Verify command
When an "assert" command fails then test execution will be aborted.	When a "verify" command fails then test will continue executing and logging the failure.
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.	Verify is best used to check non critical things. Like the presence of a headline element.

Question #3 : Explain what are the JUnits annotation linked with Selenium?

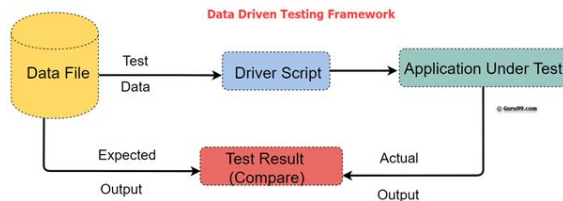
@Before public void method() – It will perform the method () before each test, this method can prepare the test

@Test public void method() – Annotations @Test identifies that this method is a test method environment

@After public void method()- To execute a method before this annotation is used, test method must start with test@Before

Question #4 : Explain what is DataDriven framework and Keyword driven?

DDT : The test data is separated and kept outside the Test Scripts, while Test Case logic resides in Test Scripts.



Keyword Driven : The functionality of the application under test is documented in a table as well as step by step instructions for each test.

Testcase Name Keywords Object Name Object Type can be xpath, name,css etc. Value for textbox, area, url etc

TestCase	Keyword	Object	ObjectType	value
Reset Login In Application	GOTOURL			url
	SETTEXT	username	name	Demo
	SETTEXT	password	name	testPassword
	CLICK	resetButton	name	
Login In Application	GOTOURL			url
	SETTEXT	username	name	Demo
	SETTEXT	password	name	testPassword
	CLICK	loginButton	name	

Excel Sheet For Keyword Driven Test

Question #5 : How you can switch between frames?

To switch between frames webdrivers [`driver.switchTo().frame()`] method takes one of the three possible arguments

A number: It selects the number by its (zero-based) index

A number or ID: Select a frame by its name or ID

Previously found WebElement: Using its previously located WebElement select a frame

```
package SampleTest;

import java.util.concurrent.TimeUnit;

public class program {

    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\abc\\Desktop\\Server\\ChromeDriver.exe");
        System.setProperty("webdriver.ie.driver", "C:\\Users\\abc\\Desktop\\Server\\IEDriverServer.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.toolsqa.com/iframe-practice-page/");

        driver.switchTo().fr
    }
}
```



Question #6 : How you can perform double click ?

```
Actions act = new Actions(driver);  
act.doubleClick(webelement);
```

```
Actions act = new Actions(driver);  
act.|
```

- build() : Action - Actions
- click() : Actions - Actions
- click(WebElement target) : Actions - Actions
- clickAndHold() : Actions - Actions
- clickAndHold(WebElement target) : Actions - Actions
- contextClick() : Actions - Actions
- contextClick(WebElement target) : Actions - Actions
- doubleClick() : Actions - Actions
- doubleClick(WebElement target) : Actions - Actions
- dragAndDrop(WebElement source, WebElement target) :
- dragAndDropBy(WebElement source, int xOffset, int yOffset) :

Question #7 : How to read a JavaScript variable in Selenium WebDriver?

```
JavascriptExecutor js = (JavascriptExecutor)driver;  
//Click on button using JS  
js.executeScript("arguments[0].click();", button);
```


Question #8 : How to wait for Element to visible by a AJAX call?

```
WebDriverWait wait = new WebDriverWait(driver, waitTime);  
wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
```

Question #9 : What is Page Object Model (POM) ?

Page Object Model is a **design pattern for creating an object repository** for web UI elements.

Each web page in the application is required to have its own corresponding page class.

The page class is thus responsible for finding the WebElements in that page and then perform operations on those web elements.

Question #10 : What is Page Factory?

Page Factory class in Selenium is an extension to the Page Object Design pattern. It is used to initialize the elements of the page object or instantiate the page objects itself.

Annotations in Page Factory are like this:

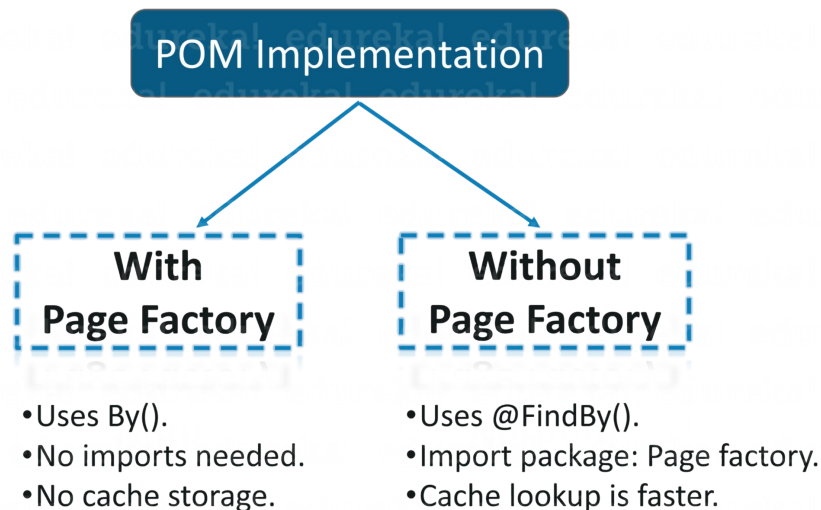
```
@FindBy(id = "userName")  
WebElement txt_UserName;
```

```
PageFactory.initElements(driver, Login.class);
```

Question #11 : What is the difference between Page Object Model and Page Factory?

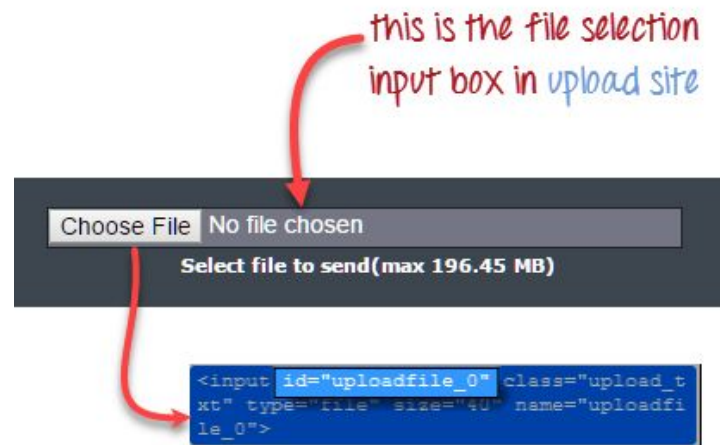
Page Object Model is a design pattern to create an Object Repository for web UI elements.

Page Factory is a built-in class in Selenium for maintaining object repository.



Question #12 : How to upload a file in Selenium WebDriver?

```
driver.get(baseUrl);  
WebElement uploadElement =  
driver.findElement(By.id("uploadfile_0"));  
uploadElement.sendKeys("D:\\\\newhtml.txt");
```



Don't Forget to Subscribe!



TheTestingAcademy

SUBSCRIBE

