



# **Selenium Interview Question & Answer – Part 2**

---

**Learn Selenium Automation from Scratch**

**Selenium Interview Question &  
Answer – Part 2 (Set of 25 Q&A)**

**Trainer – Haradhan Pal**



# Links

---

My YouTube Channel Link:

[https://www.youtube.com/c/HaradhanAutomationLibrary?sub\\_confirmation=1](https://www.youtube.com/c/HaradhanAutomationLibrary?sub_confirmation=1)

Telegram Group:

[https://t.me/joinchat/Bv\\_U1ZHC-5thNmFI](https://t.me/joinchat/Bv_U1ZHC-5thNmFI)

Facebook Page:

<https://www.facebook.com/haradhanautomationlibrary>

Facebook Group:

<https://www.facebook.com/groups/995593230934871>

LinkedIn Profile:

<https://www.linkedin.com/in/haradhan-pal-181a8326/>

---

# Agenda

---

## **Selenium WebDriver Interview Question & Answer – 26-50 (Part 2, Set of 25 Q&A)**

---

# Q26: What are the API commands we can use for a Browser using Selenium Web Driver?

---

Navigate to the required given web page - **get()**

`driverObject.get("URL");` Example: `driver.get("https://www.google.co.in");`

Close the focused Browser only - **close()**

`driverObject.close();` Example: `driver.close();`

Close all Browsers which are opened by WebDriver during test execution - **quit()**

`driverObject.quit();` Example: `driver.quit();`

Navigate from one URL to another URL - **navigate().to()**

`driverObject.navigate().to("URL");`

Navigate back to previous URL - **navigate().back()**

`driverObject.navigate().back();`

To move single item forward in the Browser history - **navigate().forward()**

`driverObject.navigate().forward();`

Refresh the content of the current web page - **navigate().refresh()**

`driverObject.navigate().refresh();`

To maximize the focused Browser - **manage().window().maximize()**

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

---

## Q27: What are the different Webpage operation available in Selenium WebDriver?

---

- To return title of the current Browser - **getTitle()**  
Syntax: `String variable = driver.getTitle();`  
Example: `driver.get("http://www.rediff.com/");`  
`String pageTitle = driver.getTitle();`  
`System.out.println(pageTitle);`
  - To return HTML page source of the current focused Browser - **getPageSource()**  
Syntax: `String stringName = driver.getPageSource();`  
Example: `driver.get("https://www.facebook.com/");`  
`String source = driver.getPageSource();`  
`System.out.println(source);`
  - To return current URL of the focused Browser - **getCurrentUrl()**  
Syntax: `String stringName = driver.getCurrentUrl();`  
Example: `driver.get("https://www.google.co.in");`  
`String url = driver.getCurrentUrl();`  
`System.out.println(url);`
-

## Q28: What are the different WebEdit operation available in Selenium WebDriver?

---

- ❑ To enter a value in the editable text box - **sendKeys()**

Syntax:

```
driver.findElement(By.ElementLocator("value")).sendKeys("Test Data");
```

- ❑ To clear already entered value in the editable text box - **clear()**

Syntax:

```
driver.findElement(By.ElementLocator("value")).clear();
```

---

## Q29: What are the different operation available for a link in Selenium WebDriver?

---

- ❑ To click a Link - **click()**

Syntax: `driver.findElement(By.ElementLocator("value")).click;`

- ❑ To check whether the Element is in enabled status or not - **isEnabled()**

Syntax: `driver.findElement(By.ElementLocator("value")).isEnabled();`

- ❑ To check if the Element is displayed or not - **isDisplayed()**

Syntax: `driver.findElement(By.ElementLocator("value")).isDisplayed();`

---

## Q30: What are the different operation available for a Check box in Selenium WebDriver?

---

- To select or unselect a Checkbox - click()

Syntax: `driver.findElement(By.ElementLocator("value")).click;`

- To verify whether the Checkbox is Displayed or not - isDisplayed()

Example: `boolean a = driver.findElement(By.ElementLocator("value")).isDisplayed();`  
`System.out.println(a);`

- To verify whether the Checkbox is in Enabled status or not - isEnabled()

Example: `boolean b = driver.findElement(By.ElementLocator("value")).isEnabled();`  
`System.out.println(b);`

- To verify whether the Checkbox is Selected or not - isSelected()

Example: `boolean c = driver.findElement(By.ElementLocator("value")).isSelected();`  
`System.out.println(c);`

---



## **Q31: What is Frame?**

---

Frame is a web page which is embedded in another web page or an HTML document embedded inside another HTML document. The Frame is often used to insert content from another source, such as an advertisement, into a Web page. The `<frame>` tag specifies a frame.

---

## **Q32: How you can identify in case any Frame is available in a webpage?**

---

Open the application using Internet Explorer or Chrome browser and then click on F12 button from keyboard. Press Ctrl + F and try to search with word "iframe". If you are able to find some search result which means some Frame is available in the webpage.

---

## Q33: How you can handle Frame using Selenium WebDriver?

---

To Switch between Frames you have to use the driver's `switchTo().frame` command. We can use the `switchTo().frame()` in below mentioned ways:

- `switchTo.frame(int frameNumber)`: Pass the frame index and driver will switch to that frame.
  - `switchTo.frame(string frameNameOrId)`: Pass the frame element Name or ID and driver will switch to that frame.
-

# Q34: How you can handle Dropdown list using Selenium WebDriver?

---

Selenium already provides Select class that has some predefined method for handling dropdowns which help a lot while working with Dropdown. To handle Drop Down and Multi Select List in Selenium we use the following types of Select Methods.

## **Types of Select Methods:**

- i. selectByVisibleText Method - selects an option by its displayed text
- ii. selectByIndex Method - selects an option by its index
- iii. selectByValue Method - selects an option by the value of its "value" attribute

## **Types of DeSelect Methods:**

- i. deselectByVisibleText Method - deselects an option by its displayed text
- ii. deselectByIndex Method - deselects an option by its index
- iii. deselectByValue Method - deselects an option by the value of its "value" attribute
- iv. deselectAll Method - deselects all previously selected options

Syntax: `Select year = new Select(driver.findElement(By.id("year")));`  
`year.selectByValue("15");`  
`year.selectByVisibleText("2020");`

---

## Q35: What is alert and how you can handle alert using Selenium WebDriver?

---

Alert is a small message box which displays on-screen notification to give the user some kind of information or ask for permission to perform certain kind of operation. It may be also used for warning purpose. Alert interface provides the below few methods which are widely used in Selenium Webdriver.

1) void dismiss() // To click on the 'Cancel' button of the alert.

Example: `driver.switchTo().alert().dismiss();`

2) void accept() // To click on the 'OK' button of the alert.

Example: `driver.switchTo().alert().accept();`

3) String getText() // To capture the alert message.

Example: `driver.switchTo().alert().getText();`

4) void sendKeys(String stringToSend) // To send some data to alert box.

Example: `driver.switchTo().alert().sendKeys("Text");`

---

## Q36: What is the difference between Implicit wait and Explicit wait?

---

- ❑ Implicit Wait: Implicit waits are used to provide a default waiting time between each consecutive test step/command across the entire test script. Thus, the subsequent test step would only execute when that time has elapsed after executing the previous test step/command.

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

- ❑ Explicit Wait : Explicit wait is used to pause the execution until the time a particular condition is met or the maximum time has elapsed. Explicit waits are applied for a particular instance only.

Syntax: `WebDriverWait wait = new WebDriverWait(driver, timeout);  
element = wait.until(ExpectedConditions.elementToBeClickable(locator));  
element.click();`

---

## Q37: What is fluent wait?

---

The importance of fluent wait is to tell WebDriver to wait for a condition, as well as the **frequency** with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

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

---

## Q38: What are few important Expected Conditions we can use in Explicit Wait?

---

- ☐ alertIsPresent()
  - ☐ elementToBeClickable()
  - ☐ elementToBeSelected()
  - ☐ presenceOfAllElementsLocatedBy()
  - ☐ presenceOfElementLocated()
  - ☐ textToBePresentInElement()
  - ☐ textToBePresentInElementLocated()
  - ☐ textToBePresentInElementValue()
  - ☐ titleIs()
  - ☐ titleContains()
  - ☐ visibilityOf()
  - ☐ visibilityOfAllElements()
  - ☐ visibilityOfAllElementsLocatedBy()
  - ☐ visibilityOfElementLocated()
-



## Q39: What are the common exceptions you had in Selenium web driver?

---

- ❑ **ElementNotVisibleException:** Although an element is present in the DOM, it is not visible, might be hidden.
  - ❑ **ElementNotSelectableException:** Although an element is present in the DOM, it may be disabled (cannot be clicked/selected).
  - ❑ **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.
  - ❑ **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.
-

## Q40: How to upload a file using Selenium WebDriver?

---

Uploading files in WebDriver is done by simply using the `sendKeys()` method on the file-select input field to enter the path to the file to be uploaded.

Syntax: `WebElement browse`  
`=driver.findElement(By.name("uploadfile"));`  
`browse.sendKeys("D:\\Test\\Test.txt");`

---

## Q41: What is Xpath and why we use it?

---

- ❑ XPath is defined as XML path. It is a syntax or language for finding any element on the web page using XML path expression. XPath is used to find the location of any element on a webpage using HTML DOM structure.
  - ❑ Xpath are used to locate elements which does not contain any id, class or name.
  - ❑ Xpath Symbols:
    - ➔ `//tagname[@attribute-name='value1']`
    - ➔ `//*[ @attribute-name='value1']`
-

## Q42: What is CSS Selector?

---

- ❑ CSS Selectors are string patterns used to identify an element based on a combination of HTML tag, id, class, and attributes. Locating by CSS Selector is the most common locating strategy of advanced Selenium users because it can access even those elements that have no ID or name.
  - ❑ CSS selector is used by browsers to target specific HTML elements and apply styles. For web browser automation, Selenium webdriver uses CSS selectors to locate and perform an action on an object instead of applying a style.
  - ❑ Syntax: `css=<HTML tag><[attribute=Value of attribute]>;`  
Example: `input[type=submit]`
  - ❑ Using CSS selectors to locate elements has some benefits:
    - ❖ It is faster
    - ❖ More readable
    - ❖ And used more often
-

## Q43: What is the difference between Xpath and CSS Selector?

---

- ❑ Using XPath we can traverse both forward and backward whereas CSS selector only moves forward.
  - ❑ CSS Selector is Faster as compared to Xpath
  - ❑ CSS Selector is easy to write as compared to Xpath
  - ❑ CSS Selector does not allow to create a selector based on text content whereas Xpath allows
-

## Q44: How to select any particular text using the Selenium Webdriver?

---

`driver.get("/");` //URL need to passed

```
WebElement itemtext =  
driver.findElement(By.xpath("//*[contains(text(),'Sele  
nium webdriver')]"));
```

```
Actions action= new Actions(driver);  
action.doubleClick(itemtext).build().perform();
```

---

## Q45: How to perform drag and drop action In Selenium WebDriver?

---

```
Actions action = new Actions(driver);
```

```
action.dragAndDrop(source_locator,  
target_locator).build().perform();
```

## **Q46: How to enter value in a text field without calling the sendKeys()?**

---

```
JavascriptExecutor js =  
(JavascriptExecutor)webdriver;
```

```
js.executeScript("document.getElementById('UserID').  
value='test@gmail.com'");
```

```
JS.executeScript("document.getElementById('Passwor  
d').value='Test123'");
```

---



## **Q47: How to launch the Chrome browser?**

---

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

```
WebDriver driver = new ChromeDriver();
```

---

#### **Q48: How to launch any url using Internet Explorer browser?**

---

```
System.setProperty("webdriver.ie.driver","  
iedriver.exe file path");
```

```
WebDriver driver = new InternetExplorerDriver();
```

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

---

## Q49: What is the difference between single and double slash in X-path?

---

### **Single slash '/' :**

Single slash ( / ) start selection from the document node

It allows you to create 'absolute' path expressions

### **Double Slash '// ' :**

Double slash ( // ) start selection matching anywhere in the document

It enables to create 'relative' path expressions

---

## Q50: How to perform double click in Selenium WebDriver?

---

```
Actions action = new Actions(driver);  
action.doubleClick(webElement);
```