SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

1. What is Selenium?

Selenium is an open-source automated testing framework used for web applications across different browsers and platforms. It allows testers to write test scripts in various programming languages like Java, Python, C#, etc.

2. What are the components of Selenium?

The main components are:

- Selenium IDE (Integrated Development Environment)

- Selenium RC (Remote Control) – Deprecated

- Selenium WebDriver

- Selenium Grid

3. What is WebDriver?

WebDriver is a component of Selenium that provides a platform to automate web application testing by interacting directly with the browser. It uses browser-specific drivers to run tests and doesn't rely on a JavaScript engine, unlike Selenium RC.

4. What are the Advantages & Disadvantages of Selenium WebDriver?

Advantages:

- Supports multiple browsers (Chrome, Firefox, etc.)

- Supports multiple programming languages

- No need for a specific server to execute tests

Disadvantages:

- No built-in reporting feature

- Limited support for handling images, CAPTCHA, etc.

- Doesn't support desktop applications, only web-based.

5. WebDriver Architecture in Selenium 3 & Selenium 4?

In Selenium 3, WebDriver interacts directly with the browser using browser-specific drivers, making HTTP requests. In Selenium 4, the WebDriver follows the W3C Standard, improving browser interaction by reducing compatibility issues and relying on native browser support.

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

6. Communication between the WebDriver and Browser

The WebDriver sends commands (like opening a URL or finding an element) to the browser through the browser's driver, which converts the commands into HTTP requests. The browser processes these requests and sends back the results.

7. What is the architecture of Selenium WebDriver?

WebDriver has a layered architecture where the client library sends commands to the WebDriver API. These commands are converted into HTTP requests and sent to the browser driver, which communicates with the actual browser.

8. How to Launch Browsers in WebDriver?

Example to launch Chrome browser:

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

9. How to Open URL?

Example to open a URL:

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

10. How to capture the Title of the page?

Example:

```
String title = driver.getTitle();
```

11. How to capture the URL of the page?

Example:

```
String currentUrl = driver.getCurrentUrl();
```

12. How to capture the page source of the Page?

Example:

```
String pageSource = driver.getPageSource();
```

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

13. What is WebDriverManager?

WebDriverManager is a library that automates the management of browser drivers for Selenium WebDriver. It allows you to avoid manually downloading and setting up drivers.

14. How to Check WebElement is Displayed, Enabled, and Selected?

isDisplayed checks if an element is visible. isEnabled checks if an element is enabled for user interaction. isSelected checks if a checkbox, radio button, or similar element is selected.

Example:

```
WebElement element = driver.findElement(By.id("elementId"));
boolean displayed = element.isDisplayed();
boolean enabled = element.isEnabled();
boolean selected = element.isSelected();
```

15. What are the Conditional Methods in Selenium WebDriver?

isDisplayed, isEnabled, and isSelected are commonly used to verify the visibility, interactivity, and selection state of elements.

16. How to Navigate Back & Forward?

Navigate back:

```
driver.navigate().back();
```

Navigate forward:

```
driver.navigate().forward();
```

17. How to Refresh Page?

Example:

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

18. What is the difference between get() & navigate.to()?

get() waits until the page is completely loaded, while navigate.to() can be used for back and forward navigation and doesn't wait for the entire page to load.

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

19. What is the difference between findElement() and findElements() in Selenium WebDriver?

findElement returns a single WebElement and throws NoSuchElementException if the element is not found. findElements returns a list of WebElements (can be empty if no elements are found).

20. How To Provide & Clear text from Input Box?

Provide text:

```
driver.findElement(By.id("inputBox")).sendKeys("text");
```

Clear text:

```
driver.findElement(By.id("inputBox")).clear();
```

21. How to capture Text from Input Box?

Example:

```
String text = driver.findElement(By.id("inputBox")).getAttribute("value");
```

22. What is the difference between getText() and getAttribute('value')?

getText retrieves the visible inner text of an element, while getAttribute('value') retrieves the value of the "value" attribute, typically used for input fields where text is entered.

23. How to handle Dropdown in Selenium WebDriver?

Dropdowns in Selenium WebDriver can be handled using the Select class. The Select class provides methods to select or deselect options in a dropdown.

Example to select by visible text:

```
WebElement dropdown = driver.findElement(By.id("dropdownId"));
Select select = new Select(dropdown);
select.selectByVisibleText("Option");
```

24. What is the use of Select class?

The Select class in Selenium WebDriver is used to interact with dropdowns. It provides methods to select options by visible text, index, or value, and also allows deselecting options in multi-select dropdowns.

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

25. How to handle Multiple Dropdowns in a page using Generic Method?

You can create a generic method that accepts the WebElement of the dropdown and the option to select (by text, index, or value). This method can then be reused for different dropdowns.

Example:

```
public void selectDropdownOption(WebElement dropdown, String option)
{
   Select select = new Select(dropdown);
   select.selectByVisibleText(option);
}
```

26. How to Handle Bootstrap Drop Down in Selenium WebDriver?

Bootstrap dropdowns are not handled using the Select class since they do not use the <select> tag. You can handle them by clicking on the dropdown element to reveal the options and then clicking on the desired option using XPath or CSS selectors.

Example:

```
driver.findElement(By.id("dropdownButton")).click();
driver.findElement(By.xpath("//a[text()='Option']")).click();
```

27. How to handle JQuery Dropdown in Selenium WebDriver?

JQuery dropdowns can be handled by first clicking on the dropdown to open it, and then selecting an option using its XPath or CSS selector.

Example:

```
driver.findElement(By.id("dropdownButton")).click();
driver.findElement(By.xpath("//li[text()='Option']")).click();
```

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

28. How to Check Drop Down Options are Sorted?

To check if dropdown options are sorted, retrieve all the options from the dropdown, store them in a list, and compare the list with a sorted version of it.

29. How to Handle Auto Suggest Drop Down in Selenium WebDriver?

Google Search:

To handle auto-suggest dropdowns like in Google search, send the keys to the search box and wait for the suggestions to load. Then, select the desired option from the list.

Example:

```
driver.findElement(By.name("q")).sendKeys("Selenium");
List<WebElement> suggestions = driver.findElements(By.xpath("//ul[@role='listbox']//li"));
for (WebElement suggestion : suggestions) {
    if (suggestion.getText().equals("Selenium WebDriver")) {
        suggestion.click();
        break;
    }
}
```

Bing Search:

Bing search works similarly to Google. You send the keys to the input box and capture the suggestions.

Example:

```
driver.findElement(By.name("q")).sendKeys("Selenium");
List<WebElement> suggestions = driver.findElements(By.xpath("//ul[@role='listbox']//li"));
for (WebElement suggestion : suggestions) {
    if (suggestion.getText().equals("Selenium WebDriver")) {
        suggestion.click();
        break;
    }
}
```

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

30. How to Handle Autocomplete Google Places Drop Down?

Google Places Autocomplete dropdown can be handled similarly to auto-suggest dropdowns by sending the keys and then selecting the desired place from the suggestions list.

Example:

```
driver.findElement(By.id("placesInput")).sendKeys("New York");
List<WebElement> suggestions = driver.findElements(By.xpath("//ul[@role='listbox']//li"));
for (WebElement suggestion : suggestions) {
    if (suggestion.getText().contains("New York, USA")) {
        suggestion.click();
        break;
    }
}
```

31. How To Select specific Check box?

To select a specific checkbox, locate it using its XPath, CSS, or another locator strategy and click on it.

Example:

```
driver.findElement(By.id("checkboxId")).click();
```

32. How to select all the checkboxes?

To select all checkboxes, use a locator that matches all the checkboxes and iterate through the list of elements, clicking on each one.

Example:

```
List<WebElement> checkboxes = driver.findElements(By.xpath("//input[@type='checkbox']"));
for (WebElement checkbox : checkboxes) {
    checkbox.click();
}
```

33. How to select multiple checkboxes by choice?

You can select multiple checkboxes by finding their locators and clicking on the specific checkboxes you want.

Example:

```
driver.findElement(By.id("checkbox1")).click();
driver.findElement(By.id("checkbox2")).click();
```

SELENIUM INTERVIEW QUESTIONS AND ANSWERS (PART 01)

34. How to select last N checkboxes?

You can use the findElements method to retrieve all checkboxes and then select the last N elements by iterating over the list.

Example:

```
List<WebElement> checkboxes = driver.findElements(By.xpath("//input[@type='checkbox']"));
for (int i = checkboxes.size() - N; i < checkboxes.size(); i++) {
    checkboxes.get(i).click();
}
```

35. How to select first N checkboxes?

Similar to selecting the last N checkboxes, but this time you iterate through the first N elements.

```
List<WebElement> checkboxes = driver.findElements(By.xpath("//input[@type='checkbox']"));
for (int i = 0; i < N; i++) {
    checkboxes.get(i).click();
}
```
