Selenium Interview Questions

Reference https:/www.javatpoint.com/selenium-interview-questions

A list of frequently asked **Selenium interview questions** and answers are given below.

Basic Level - Selenium Interview Questions

1) What is test automation or automation testing?

Automation testing enables the use of specialized tools to automate the execution of manually designed test cases without any human intervention. Automation testing tools can access the test data, controls the execution of tests and compares the actual result against the expected result. Consequently, generating detailed test reports of the system under test.

---

2) What are the advantages of automation testing?

- Automation testing supports both functional and performance test on an application.
- It supports the execution of repeated test cases.
- It facilitates parallel execution.
- It aids in testing a large test matrix.
- It improves accuracy because there are no chances of human errors.
- It saves time and money.

---

3) Name some of the commonly used Automation Testing tools that are used for Functional Automation.

- Quick Test Professional, provided by HP.
- Rational Robot, provided by IBM.
- Coded UI, provided by Microsoft.
- Selenium, open source.
- Auto It, Open Source.

---

4) Name some of the commonly used Automation Testing tools that are used for Non-Functional Automation.

- Load Runner, provided by Hp.
- JMeter, provided by Apache.
- Burp Suite, provided by PortSwigger.
- Acunetix, provided by Acunetix.

5) What is Selenium?

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite. **Jason Huggins** developed Selenium in 2004 as an internal tool at **Thought Works**. Selenium supports automation across different browsers, platforms, and programming languages.

6) What are the different components of Selenium?

Selenium is not just a single tool but a suite of software's, each having a different approach to support automation testing. It comprises of four major components which include:

1. Selenium Integrated Development Environment (IDE)
2. Selenium Remote Control (Now Deprecated)
3. WebDriver
4. Selenium Grid

7) List out the names of programming languages, browsers and operating systems that are supported by Selenium.

Selenium supports various operating systems, browsers and programming languages. Following is the list:

- **Programming Languages**: C#, Java, Python, PHP, Ruby, Perl, JavaScript.
- **Operating Systems**: Android, iOS, Windows, Linux, Mac, Solaris.

- **Browsers**: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.

8) What are the significant changes/upgrades in various Selenium versions?

Selenium v1.0:

- Version 1.0 was the initial release of Selenium.
- It included three tools: Selenium IDE, Selenium RC, and Selenium Grid.

Selenium v2.0:

- Selenium WebDriver was introduced replacing Selenium RC in version "2.0".
- With the onset of WebDriver, RC got deprecated and moved to the legacy package.

Selenium v3:

- The latest release Selenium 3 has new added features and functionalities.
- It includes Selenium IDE, Selenium WebDriver, and Selenium Grid.

9) List some of the test types that are supported by Selenium.

Selenium supports two types of testing:

- Functional Testing
- Regression Testing
- Sanity Testing
- Smoke Testing
- Responsive Testing
- Cross Browser Testing
- UI testing (black box)
- Integration Testing

10) What is Selenium IDE?

Selenium IDE is implemented as Firefox extension which provides record and playback functionality on test scripts. It allows testers to export recorded scripts in many languages like HTML, Java, Ruby, RSpec, Python, C#, JUnit and TestNG.

Selenium IDE has limited scope, and the generated test scripts are not very robust, and portable.

11) What do you mean by Selenese?

Selenium commands, also known as "Selenese" are the set of commands used in Selenium that run your tests. For example, command - open (URL); launches the desired URL in the specified browser and it accept both relative and absolute URLs.

A sequence of Selenium commands (Selenese) together is known as a test script.

12) What are the different ways of locating a web element in Selenium?

In Selenium, web elements are identified and located with the help of Locators. Locators specify a target location which uniquely defines the web element in the context of a web application. Thus, to identify web elements accurately and precisely we have different types of locators in Selenium:

- ID
- ClassName
- Name
- TagName
- LinkText
- PartialLinkText
- Xpath
- CSS Selector
- DOM

13) How many types of WebDriver API's are available in Selenium?

The list of WebDriver API's which are used to automate browser include:

- AndroidDriver
- ChromeDriver
- EventFiringWebDriver
- FirefoxDriver
- HtmlUnitDriver
- InternetExplorerDriver
- iPhoneDriver
- iPhoneSimulatorDriver
- RemoteWebDriver

14) List out some of the Automation tools which could be integrated with Selenium to achieve continuous testing.

Selenium can be used to automate functional tests and can be integrated with automation test tools such as **Maven, Jenkins, &Docker** to achieve continuous testing. It can also be integrated with tools such as **TestNG, &JUnit** for managing test cases and generating reports.

15) What do you mean by the assertion in Selenium?

The assertion is used as a verification point. It verifies that the state of the application conforms to what is expected. The types of assertion are "assert", "verify" and "waitFor".

16) Explain the difference between assert and verify commands?

Assert: Assert command checks if the given condition is true or false. If the condition is true, the program control will execute the next phase of testing, and if the condition is false, execution will stop, and nothing will be executed.

Verify: Verify command also checks if the given condition is true or false. It doesn't halt program execution, i.e., any failure during verification would not stop the execution, and all the test phases would be executed.

17) What do you mean by XPath?

- XPath is a language used for locating nodes in XML documents.
- XPath can be used as a substitute when you don't have a suitable id or name attribute for the element you want to locate.
- XPath provides locating strategies like:
  - XPath Absolute
  - XPath Attributes

18) Explain XPath Absolute and XPath attributes.

XPath Absolute:

- XPath Absolute enables users to mention the complete XPath location from the root HTML tag to the specific elements.
- Syntax: //html/body/tag1[index]/tag2[index]/.../tagN[index]
- Example: //html/body/div[2]/div/div[2]/div/div/div/fieldset/form/div[1]/input[1]

XPath Attributes:

- XPath Attributes is always recommended when you don't have a suitable id or name attribute for the element you want to locate.
- Syntax: //htmltag[@attribute1='value1' and @attribute2='value2']
- Example: //input[@id='passwd' and @placeholder='password']

19) What is the difference between "/" and "//" in XPath?

Single Slash "/": Single slash is used to create XPath with absolute path.

Double Slash "//": Double slash is used to create XPath with the relative path.

20) What are the different types of annotations which are used in Selenium?

JUnit annotations which can be used are:

- Test
- Before
- After
- Ignore
- BeforeClass
- AfterClass
- RunWith

21) What are the WebDriver supported Mobile Testing Drivers?

WebDriver supported "mobile testing drivers" are:

- AndroidDriver
- IphoneDriver
- OperaMobileDriver

22) What are the popular programming languages supported by Selenium WebDriver to write Test Cases?

- JAVA
- PHP
- Python
- C#
- Ruby
- Perl

23) What is the difference between type keys and type commands?

TypeKeys() will trigger JavaScript event in most of the cases whereas .type() won't.

---

24) What is the difference between "type" and "typeAndWait" command?

"type" command is used to type keyboard key values into the text box of software web application. It can also be used for selecting values of combo box whereas "typeAndWait" command is used when your typing is completed and software web page start reloading. This command will wait for software application page to reload. If there is no page reload event on typing, you have to use a simple "type" command.

---

25) What is the difference between findElement() and findElements()?

findElement(): It is used to find the first element within the current page using the given "locating mechanism". It returns a single WebElement.

findElements(): It uses the given "locating mechanism" to find all the elements within the current page. It returns a list of web elements.

---

26) What is the wait? How many types of waits in selenium?

Selenium Webdriver introduces the concept of waits for the AJAX-based application. There are two types of waits:

1. Implicit Wait
2. Explicit Wait

---

27) What is the main disadvantage of implicit wait?

The main disadvantage of implicit wait is that it slows down test performance.

Another disadvantage of implicit wait is:

Suppose, you set the waiting limit to be 10 seconds, and the elements appear in the DOM in 11 seconds, your tests will be failed because you told it to wait a maximum of 10 seconds.

28) What is Selenium Grid?

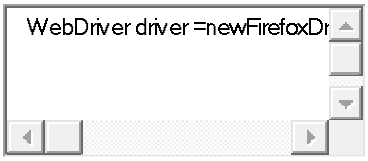
Selenium Grid facilitates you to distribute your tests on multiple machines and all of them at the same time. So, you can execute tests on Internet Explorer on Windows and Safari on Mac machine using the same text script. It reduces the time of test execution and provides quick feedback.

Advance Level - Selenium Interview Questions

29) How can we launch different browsers in Selenium WebDriver?

We have to create an instance of a driver of that particular browser.

```
WebDriver driver =newFirefoxDriver();
```

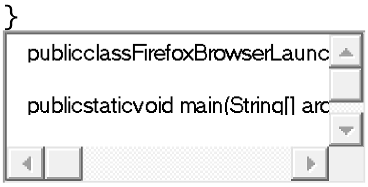


```
WebDriver driver =newFirefoxDr
```

Here, "WebDriver" is an interface, and we are creating a reference variable "driver" of type WebDriver, instantiated using "FireFoxDriver" class.

30) Write a code snippet to launch Firefox browser in WebDriver.

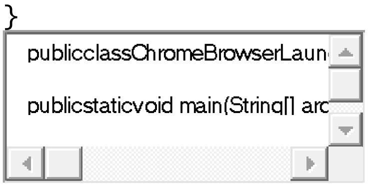
```
publicclassFirefoxBrowserLaunchDemo {

publicstaticvoid main(String[] args) {

//Creating a driver object referencing WebDriver interface
WebDriver driver;

//Setting webdriver.gecko.driver property
System.setProperty("webdriver.gecko.driver", pathToGeckoDriver + "\\geckodriver.exe");

//Instantiating driver object and launching browser
driver = newFirefoxDriver();

//Using get() method to open a webpage
driver.get("http://javatpoint.com");

//Closing the browser
driver.quit();

    }

}
```



```
publicclassFirefoxBrowserLaunc

publicstaticvoid main(String[] arc
```

31) Write a code snippet to launch Chrome browser in WebDriver.

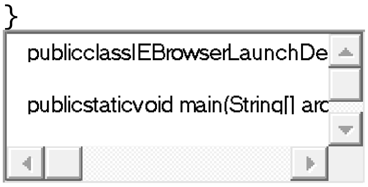
```
publicclassChromeBrowserLaunchDemo {

publicstaticvoid main(String[] args) {

//Creating a driver object referencing WebDriver interface
WebDriver driver;

//Setting the webdriver.chrome.driver property to its executable's location
System.setProperty("webdriver.chrome.driver", "/lib/chromeDriver/chromedriver.exe");

//Instantiating driver object
driver = newChromeDriver();

//Using get() method to open a webpage
driver.get("http://javatpoint.com");

//Closing the browser
driver.quit();

    }

}
```



publicclassChromeBrowserLaun
publicstaticvoid main(String[] arc

32) Write a code snippet to launch Internet Explorer browser in WebDriver.

```
publicclassIEBrowserLaunchDemo {

publicstaticvoid main(String[] args) {

//Creating a driver object referencing WebDriver interface
WebDriver driver;

//Setting the webdriver.ie.driver property to its executable's location
System.setProperty("webdriver.ie.driver", "/lib/IEDriverServer/IEDriverServer.exe");


//Instantiating driver object
driver = newInternetExplorerDriver();

//Using get() method to open a webpage
driver.get("http://javatpoint.com");

//Closing the browser
driver.quit();

    }

}
```

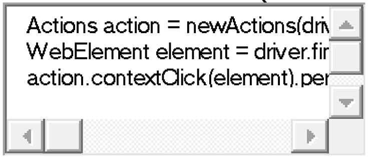


publicclassIEBrowserLaunchDe
publicstaticvoid main(String[] arc

33) Write a code snippet to perform right-click an element in WebDriver.

We will use **Action class** to generate user event like right-click an element in WebDriver.

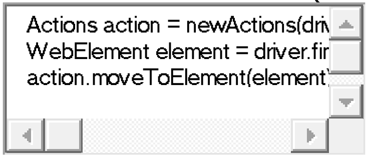
```
Actions action = newActions(driver);
WebElement element = driver.findElement(By.id("elementId"));
action.contextClick(element).perform();
```



```
Actions action = newActions(driv
WebElement element = driver.fir
action.contextClick(element).per
```

34) Write a code snippet to perform mouse hover in WebDriver.

```
Actions action = newActions(driver);
WebElement element = driver.findElement(By.id("elementId"));
action.moveToElement(element).perform();
```



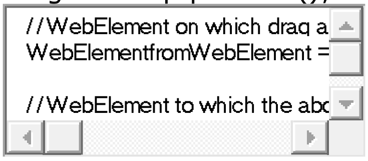
```
Actions action = newActions(driv
WebElement element = driver.fir
action.moveToElement(element)
```

35) How do you perform drag and drop operation in WebDriver?

Code snippet to perform drag and drop operation:

```
//WebElement on which drag and drop operation needs to be performed
WebElementfromWebElement = driver.findElement(By Locator of fromWebElement);

//WebElement to which the above object is dropped
WebElementtoWebElement = driver.findElement(By Locator of toWebElement);

//Creating object of Actions class to build composite actions
Actions builder = newActions(driver);

//Building a drag and drop action
```

```
Action dragAndDrop = builder.clickAndHold(fromWebElement)
             .moveToElement(toWebElement)
             .release(toWebElement)
         .build();

//Performing the drag and drop action
dragAndDrop.perform();
```



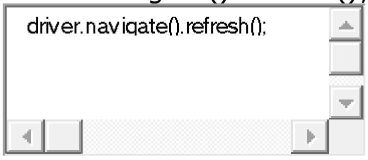
```
//WebElement on which drag a
WebElementfromWebElement =

//WebElement to which the abc
```

36) What are the different methods to refresh a web page in WebDriver?

There are multiple ways of refreshing a page in Webdriver.

1. Using driver.navigate command -

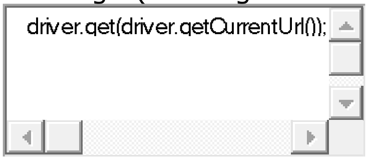
```
driver.navigate().refresh();
```



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

2. Using driver.getCurrentUrl() with driver.get() command -

```
driver.get(driver.getCurrentUrl());
```



```
driver.get(driver.getCurrentUrl());
```

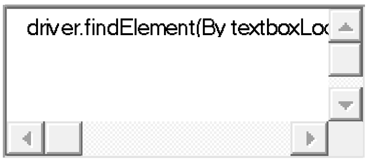
3. Using driver.getCurrentUrl() with driver.navigate() command -

```
driver.navigate().to(driver.getCurrentUrl());
```

```
driver.navigate().to(driver.getCur
```

4. Pressing an F5 key on any textbox using the sendKeys command -

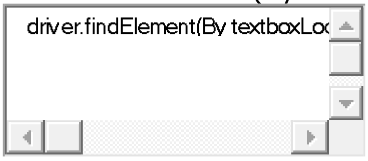
```
driver.findElement(By textboxLocator).sendKeys(Keys.F5);
```



```
driver.findElement(By textboxLoc
```

5. Passing ascii value of the F5 key, i.e., "\uE035" using the sendKeys command -

```
driver.findElement(By textboxLocator).sendKeys("\uE035");
```

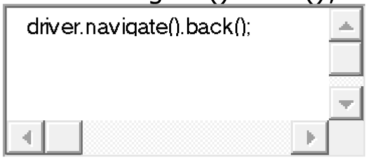


```
driver.findElement(By textboxLoc
```

37) Write a code snippet to navigate back and forward in browser history?

Navigate back in browser history:

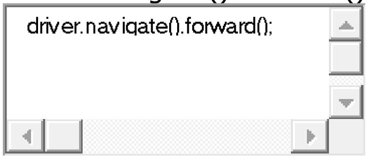
```
driver.navigate().back();
```



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

Navigate forward in browser history:

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



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

38) How to invoke an application in WebDriver?

```
driver.get("url"); or
driver.navigate().to("url");
```