Selenium Interview Questions-Answers

1. **Why should Selenium be selected as a test tool?**

Selenium :

1. is free and open source
2. have a large user base and helping communities
3. have cross Browser compatibility (Firefox, Chrome, Internet Explorer, Safari etc.)
4. have great platform compatibility (Windows, Mac OS, Linux etc.)
5. supports multiple programming languages (Java, C#, Ruby, Python, Pearl etc.)
6. has fresh and regular repository developments
7. supports distributed testing
8. **What is Selenium? What are the different Selenium components?**

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web-based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.

Selenium is not just a single tool or a utility, rather a package of several testing tools and for the same reason, it is referred to as a Suite. Each of these tools is designed to cater different testing and test environment requirements.

The suite package constitutes of the following sets of tools:

* [**Selenium Integrated Development Environment (IDE)**](http://www.softwaretestinghelp.com/selenium-ide-download-and-installation-selenium-tutorial-2/) – Selenium IDE is a record and playback tool. It is distributed as a Firefox Plugin.
* **Selenium Remote Control (RC)** – Selenium RC is a server that allows a user to create test scripts in the desired programming language. It also allows executing test scripts within the large spectrum of browsers.
* [**Selenium WebDriver**](http://www.softwaretestinghelp.com/selenium-webdriver-selenium-tutorial-8/) – WebDriver is a different tool altogether that has various advantages over Selenium RC. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
* [**Selenium Grid**](http://www.softwaretestinghelp.com/selenium-grid-selenium-tutorial-29/) – Selenium Grid is used to distribute your test execution on multiple platforms and environments concurrently.

1. **What are the limitations of Selenium?**

Following are the limitations of Selenium:

* Selenium supports testing of only web-based applications
* Mobile applications cannot be tested using Selenium
* Captcha and Barcode readers cannot be tested using Selenium
* Reports can only be generated using third-party tools like TestNG or JUnit.
* As Selenium is a free tool, thus there is no ready vendor support though the user can find numerous helping communities.
* The user is expected to possess prior programming language knowledge.

1. **When should I use Selenium IDE?**

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. Selenium IDE is an ideal tool for a naïve user.

1. **What is Selenese?**

Selenese is the language which is used to write test scripts in Selenium IDE.

1. **What are the different types of locators in Selenium?**

The locator can be termed as an address that identifies a web element uniquely within the webpage. Thus, to identify web elements accurately and precisely we have [different types of locators in Selenium](http://www.softwaretestinghelp.com/using-selenium-xpath-and-other-locators-selenium-tutorial-5/):

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

1. **When should I use Selenium Grid?**

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably.

1. **What do we mean by Selenium 1 and Selenium 2?**

Selenium RC and WebDriver, in a combination are popularly known as Selenium 2. Selenium RC alone is also referred as Selenium 1.

**9. Which is the latest Selenium tool?**

WebDriver

**10. How do I launch the browser using WebDriver?**

The following syntax can be used to launch Browser:  
 WebDriver driver = **new** FirefoxDriver();  
 WebDriver driver = **new** ChromeDriver();  
 WebDriver driver = **new** InternetExplorerDriver();

**11. What is the difference between “/” and “//” in Xpath?**

**Single Slash “/” –**Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document node/start node.

**Double Slash “//” –** Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

**12. What is Same origin policy and how it can be handled?**

The problem of same origin policy disallows to access the DOM of a document from an origin that is different from the origin we are trying to access the document.

Origin is a sequential combination of scheme, host and port of the URL. For example, for a URL http://www.softwaretestinghelp.com/resources/, the origin is a combination of http, softwaretestinghelp.com, 80 correspondingly.

Thus the Selenium Core (JavaScript Program) cannot access the elements from an origin that is different from where it was launched. For Example, if I have launched the JavaScript Program from “http://www.softwaretestinghelp.com”, then I would be able to access the pages within the same domain such as “http://www.softwaretestinghelp.com/resources” or “http://www.softwaretestinghelp.com/istqb-free-updates/”. The other domains like google.com, seleniumhq.org would no more be accessible.

So, In order to handle same origin policy, Selenium Remote Control was introduced.

**13. How can you find if an element in displayed on the screen?**

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()
2. isSelected()
3. isEnabled()

**Syntax:**

**isDisplayed():**  
***boolean****buttonPresence = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isSelected():**  
***boolean****buttonSelected = driver.findElement(By.id(“gbqfba”)).isDisplayed();*

**isEnabled():**  
***boolean****searchIconEnabled = driver.findElement(By.id(“gbqfb”)).isEnabled();*

**14. How can we get a text of a web element?**

Get command is used to retrieve the inner text of the specified web element. The command doesn’t require any parameter but returns a string value. It is also one of the extensively used commands for verification of messages, labels, errors etc displayed on the web pages.

**Syntax:**  
String Text = driver.findElement(By.id(“Text”)).getText();

**15. How to select value in a dropdown?**

Value in the drop down can be selected using WebDriver’s Select class.

**Syntax:**

**selectByValue:**  
Select selectByValue = **new** Select(driver.findElement(By.id(“SelectID\_One”)));  
selectByValue.selectByValue(“greenvalue”);

**selectByVisibleText:**  
Select selectByVisibleText = **new** Select (driver.findElement(By.id(“SelectID\_Two”)));  
selectByVisibleText.selectByVisibleText(“Lime”);

**selectByIndex:**  
Select selectByIndex = **new** Select(driver.findElement(By.id(“SelectID\_Three”)));  
selectByIndex.selectByIndex(2);

**16. What are the different types of navigation commands?**

Following are the [navigation commands](http://www.softwaretestinghelp.com/selenium-webdriver-waits-selenium-tutorial-15/):  
**navigate().back()** – The above command requires no parameters and takes back the user to the previous webpage in the web browser’s history.

**Sample code:**  
driver.navigate().back();

**navigate().forward()** – This command lets the user to navigate to the next web page with reference to the browser’s history.

**Sample code:**  
driver.navigate().forward();

**navigate().refresh()** – This command lets the user to refresh the current web page there by reloading all the web elements.

**Sample code:**  
driver.navigate().refresh();

**navigate().to()** – This command lets the user to launch a new web browser window and navigate to the specified URL.

**Sample code:**  
driver.navigate().to(“https://google.com”);

**17. How to click on a hyper link using linkText?**

driver.findElement(By.linkText(“Google”)).click();

The command finds the element using link text and then click on that element and thus the user would be re-directed to the corresponding page.

The above mentioned link can also be accessed by using the following command.

driver.findElement(By.partialLinkText(“Google”)).click();

The above command find the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element with the specified substring and then clicks on it.

**18. How to**[**handle frame in WebDriver**](http://www.softwaretestinghelp.com/selenium-tutorial-18/)**?**

An inline frame acronym as iframe is used to insert another document with in the current HTML document or simply a web page into a web page by enabling nesting.

**Select iframe by id**  
driver.switchTo().frame(“ID of the frame“);

**Locating iframe using tagName**  
driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));

**Locating iframe using index**

**frame(index)**  
driver.switchTo().frame(0);

**frame(Name of Frame)**  
driver.switchTo().frame(“name of the frame”);

**frame(WebElement element)**  
**Select Parent Window**  
driver.switchTo().defaultContent();

**19. When do we use findElement() and findElements()?**

**findElement():**findElement() is used to find the first element in the current web page matching to the specified locator value. Take a note that only first matching element would be fetched.

**Syntax:**

WebElement element = driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));

**findElements():**findElements() is used to find all the elements in the current web page matching to the specified locator value. Take a note that all the matching elements would be fetched and stored in the list of WebElements.

**Syntax:**  
List <WebElement> elementList = driver.findElements(By.xpath(“//div[@id=’example’]//ul//li”));

**20. How to find more than one web element in the list?**

At times, we may come across elements of same type like multiple hyperlinks, images etc arranged in an ordered or unordered list. Thus, it makes absolute sense to deal with such elements by a single piece of code and this can be done using WebElement List.

**Sample Code:**

// Storing the list

List &lt;WebElement&gt; elementList = driver.findElements(By.xpath("//div[@id='example']//ul//li"));

// Fetching the size of the list

int listSize = elementList.size();

for (int i=0; i&lt;listSize; i++)

{

// Clicking on each service provider link

serviceProviderLinks.get(i).click();

// Navigating back to the previous page that stores link to service providers

driver.navigate().back();

}

**21. What is the difference between driver.close() and driver.quit command?**

**close()**: WebDriver’s close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does is return any value.

**quit()**: Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does is return any value.