

1) What is Selenium Webdriver?

Selenium WebDriver is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It directly communicates with the browser and controls it. Selenium WebDriver supports various programming languages like – Java, C#, PHP, Python, Perl, Ruby. and Javascript.

2) What is Selenium Grid and when do we go for it?

Selenium Grid is used to run tests on different machines against different browsers in parallel. We use Selenium Grid in the following scenarios:

- Execute your test on different operating systems
- Execute your tests on different versions of same browser
- Execute your tests on multiple browsers
- Execute your tests in parallel and multiple threads

3) What are the advantages of Selenium Grid?

Below are the benefits of Selenium Grid:

- Selenium Grid gives the flexibility to distribute your test cases for execution.
- Reduces batch processing time.
- Can perform multi-browser testing.
- Can perform multi-OS testing.

4) What is a Hub in Selenium Grid?

Hub is the central point to the entire GRID Architecture which receives all requests. There is only one hub in the selenium grid. Hub distributes the test cases across each node.

5) What is a Node in Selenium Grid?

There can be multiple nodes in Grid. Tests will run in nodes. Each node communicates with the Hub and performs test assigned to it.

6) What are the types of WebDriver API's that are supported/available in Selenium?

Selenium Webdriver supports most of the browser driver APIs like Chrome, Firefox, Internet Explorer, Safari and PhantomJS

7) Which WebDriver implementation claims to be the fastest?

HTML UnitDriver is the most light weight and fastest implementation headless browser for of WebDriver. It is based on HtmlUnit. It is known as Headless Browser Driver. It is same as Chrome, IE, or FireFox driver, but it does not have GUI so one cannot see the test execution on screen.

8) What are the open source frameworks supported by Selenium WebDriver?

Some of the popular open source frameworks supported by Webdriver are:

-TestNG

-JUnit

-Cucumber

-Robot Framework

-Appium

-Protractor

9) What is the difference between Soft Assert and Hard Assert in Selenium?

Hard Assert throws an *AssertionException* immediately when an assert statement fails and test suite continues with next *@Test*. It marks method as fail if assert condition gets failed and the remaining statements inside the method will be aborted.

Soft Assert collects errors during *@Test*. Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

10) What are the verification points available in Selenium?

Different types of verification points in Selenium are:

To check element is present

```
if(driver.findElements(By.XPath("value")).size()!=0) {  
  
System.out.println("Element is present");  
  
}else{  
  
System.out.println("Element is absent");  
  
}
```

```
}
```

To check element is visible

```
if(driver.findElement(By.Id("submit")).isDisplayed()){  
  
System.out.println("Element is visible");  
  
}else{  
  
System.out.println("Element is visible");  
  
}
```

To check element is enabled

```
if(driver.findElement(By.Id("submit")).isEnabled()){  
  
System.out.println("Element is enabled");  
  
}else{  
  
System.out.println("Element is disabled");  
  
}
```

To check text is present

```
if(driver.getPageSource().contains("Text")){  
  
System.out.println("Text is present");  
  
}else{  
  
System.out.println("Text is not present");  
  
}
```

11) Why do we create a reference variable 'driver' of type WebDriver and what is the purpose of its creation?

We create an instance of the WebDriver interface and cast it to different browser class using the reference variable 'driver'. Then we can use different methods of the web driver interface like get(), getTitle(), close(), etc...to write automation code.

12) What are the different types of exceptions you have faced in Selenium WebDriver?

Different types of exceptions in Selenium are:

– NoSuchElementException

-NoSuchWindowException

-NoSuchFrameException

-NoAlertPresentException

-ElementNotVisibleException

-ElementNotSelectableException

-TimeoutException

13) How to login into any site if it is showing an authentication pop-up for Username and Password?

To work with Basic Authentication pop-up (which is a browser dialogue window), you just need to send the user name and password along with the application URL.

Syntax:

```
driver.get("http://admin:admin@yoururl.com");
```

14) What is implicit wait in Selenium WebDriver?

The implicit wait will tell the WebDriver to wait a certain amount of time before it throws a “No Such Element Exception.” The default setting of implicit wait is zero. Once you set the time, the web driver will wait for that particular amount of time before throwing an exception.

Syntax:

```
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
```

15) What is WebDriver Wait in Selenium WebDriver?

Explicit waits are a concept from the dynamic wait, which waits dynamically for specific conditions. It can be implemented by the WebDriverWait class.

Syntax:

```
WebDriverWait wait = new WebDriverWait(driver, 10);  
WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("button")));
```

16) What is Fluent Wait in Selenium WebDriver?

Each FluentWait instance defines the maximum amount of time to wait for a condition, as well as the frequency with which to check the condition. Furthermore, the user may configure the wait to ignore specific types of exceptions whilst waiting, such as NoSuchElementExceptions when searching for an element on the page.

Syntax:

```
Wait wait = new FluentWait(driver)

.withTimeout(30, SECONDS)

.pollingEvery(5, SECONDS)

.ignoring(NoSuchElementException.class);

WebElement element = wait.until(new Function() {

public WebElement apply(WebDriver driver) {

return driver.findElement(By.id("element"));

}}

})
```

17) How to input text into the text box fields without calling the sendKeys()?

We can use Javascript action to enter the value in text box.

Syntax:

```
JavascriptExecutor executor = (JavascriptExecutor)driver;

executor.executeScript("document.getElementById('textbox_id').value='new value'");
```

18) How to clear the text inside the text box fields using Selenium WebDriver?

Syntax:

```
driver.findElement(By.Id("textbox_id")).clear();
```

19) How to get an attribute value of an element using Selenium WebDriver?

```
driver.findElement(By.Id("button_id")).getAttribute("text");
```

20) How to press Enter key on text box in Selenium WebDriver?

```
driver.findElement(By.Id("button_id")).sendKeys(keys.ENTER);
```

21) How to pause a test execution for 5 seconds at a specific point?

We can pause test execution for 5 seconds by using the wait command.

Syntax:

```
driver.wait(5);
```

22) Is Selenium Server needed to run Selenium WebDriver scripts?

In case of Selenium WebDriver, it does not require to start Selenium Server for executing test scripts. Selenium WebDriver makes the calls between browser & automation script.

23) What happens if we run this command `driver.get("www.google.com");`?

It will load a new web page in the current browser window with the website url set to "www.google.com" . This is done using an http get operation, and the method will block until the load is complete.

24) What is an alternative to `driver.get()` method to open a URL using Selenium WebDriver?

We can use `driver.navigate().To("URL")` method to open a URL.

25) What is the difference between `driver.get("URL")` and `driver.navigate().to("URL")` commands?

`driver.get()` is used to navigate particular URL(website) and wait till page load.

`driver.navigate()` is used to navigate to particular URL and does not wait to page load. It maintains browser history or cookies to navigate back or forward.