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

**driver.close()**

The driver.close() command is used to close the current browser window having focus. In case there is only one browser open then calling driver.close() quits the whole browser session.

**Usability**It is best to use driver.close() when we are dealing with multiple browser tabs or windows e.g. when we click on a link that opens another tab. In this case after performing required action in the new tab, if we want to close the tab we can call the driver.close() method.

## driver.quit()

The driver.quit() is used to quit the whole browser session along with all the associated browser windows, tabs and pop-ups.

**Usability**  
It is best to use driver.quit() when we no longer want to interact with the driver object along with any associated window, tab or pop-up. Generally, it is one of the last statements of the automation scripts. In case, we are working with Selenium with TestNG or JUnit, we call driver.quit() in the @AfterSuite method of our suite. Thus, closing it at the end of the whole suite.

2. What are the different types of waits available in WebDriver?

## Implicit Wait:

Selenium Web Driver has borrowed the idea of implicit waits from Watir.The implicit wait will tell to the web driver to wait for certain amount of time before it throws a "No Such Element Exception". The default setting is 0. Once we set the time, web driver will wait for that time before throwing an exception.

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

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

## Explicit Wait:

The explicit wait is used to tell the Web Driver to wait for certain conditions (**Expected Conditions**) or the maximum time exceeded before throwing an "**ElementNotVisibleException**" exception.

Once we declare explicit wait we have to use "**ExpectedCondtions**" or we can configure how frequently we want to check the condition using **Fluent Wait**. These days while implementing we are using **Thread.Sleep()**generally it is not recommended to use

Syntax: WebDriverWait wait = new WebDriverWait(WebDriverRefrence,TimeOut);

Eg: driver = new ChromeDriver();

WebDriverWait wait=new WebDriverWait(driver, 20);

## Fluent Wait:

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

**Frequency:**Setting up a repeat cycle with the time frame to verify/check the condition at the regular interval of time

Syntax: Wait wait = new FluentWait(WebDriver reference)

.withTimeout(Duration.ofSeconds(SECONDS))

.pollingEvery(Duration.ofSeconds(SECONDS))

.ignoring(Exception.class);

Eg: Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)

.withTimeout(30, TimeUnit.SECONDS)

.pollingEvery(5, TimeUnit.SECONDS)

.ignoring(NoSuchElementException.class);

3. How to type in a textbox using Selenium?

WebElement username = driver.findElement(By.id(“username”));

username.sendKeys(“Suganthi Ravi”);

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

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

* isDisplayed();
* isSelected();
* isEnabled();

5. How can we get a text of a web element?

String buttonText = driver.findElement(By.cssSelector("div.success > button")).**getText()**;

String fullText = driver.findElement(By.cssSelector("div.success")).**getText()**;

6. How to select value in a dropdown?

**Step 1:** Import the "Select" package.

import org.openqa.selenium.support.ui.Select;

**Step 2:** Declare the drop-down element as an instance of the Select class. In the example below, we named this instance as "drpCountry".

Select drpCountry = new Select (driver.findElement(By.name(“country”)));

**Step 3:** We can now start controlling "drpCountry" by using any of the available Select methods. The sample code below will select the option "ANTARCTICA."

drpCountry.selectByVisibleText(“ANTARCTICA”);

## Select Methods:

The following are the most common methods used on drop down list.

|  |  |
| --- | --- |
| **Method** | **Description** |
| **selectByVisibleText()**and **deselectByVisibleText()** *Example:* | * Selects/deselects the option that displays the text matching the parameter. * **Parameter**: The exactly displayed text of a particular option |
| **selectByValue()** and **deselectByValue()** *Example:* | * Selects/deselects the option whose "value" attribute matches the specified parameter. * **Parameter**: value of the "value" attribute * Remember that not all drop-down options have the same text and "value", like in the example below. |
| **selectByIndex()** and **deselectByIndex()** *Example:* | * Selects/deselects the option at the given index. * **Parameter**: the index of the option to be selected. |
| **isMultiple()** *Example:* | * Returns TRUE if the drop-down element allows multiple selections at a time; FALSE if otherwise. * **No parameters needed** |
| **deselectAll()** *Example:* | * Clears all selected entries. This is only valid when the drop-down element supports multiple selections. * **No parameters needed** |

7. What are the different types of navigation commands?

* + driver.navigate().to("www.javatpoint.com");
  + driver.navigate().forward();
  + driver.navigate().back();
  + driver.navigate().refresh();

8. How to click on a hyper link using linkText?

driver.findElement(By.linkText("click here")).click();

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

To Switch between iFrames we have to use the driver’s switchTo().frame command. We can use the switchTo().frame() in three 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.
* switchTo.frame(WebElement frameElement): Pass the frame web element and driver will switch to that frame.

10. When do we use findElement() and findElements()?

Selenium WebDriver defines two methods for identifying the elements, they are findElement and findElements . findElement: This command is used to uniquely identify a web element within the web page. findElements: This command is used to uniquely identify the list of web elements within the web page.