Selenium Questions

- **What is difference between assert and verify commands?**

Assert: Assert command checks whether the given condition is true or false. Let’s say we assert whether the given element is present on the web page or not. If the condition is true then the program control will execute the next test step but if the condition is false, the execution would stop and no further test would be executed.

Verify: Verify command also checks whether the given condition is true or false. Irrespective of the condition being true or false, the program execution doesn’t halts i.e. any failure during verification would not stop the execution and all the test steps would be executed.

- **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();
```

- **What are the different types of waits available in WebDriver?**

1. Implicit Wait
2. Explicit Wait

Implicit Wait: Implicit waits are used to provide a default waiting time (say 30 seconds) between each consecutive test step/command across the entire test script. Thus, subsequent test step would only execute when the 30 seconds have elapsed after executing the previous test step/command.

Syntax

```
import java.util.concurrent.TimeUnit
drv.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

drawbacks – It gives rise to the test script execution time as each of the command would be ceased to wait for a stipulated amount of time before resuming the execution.

Explicit Wait: Explicit waits are used to halt the execution till the time a particular condition is met or the maximum time has elapsed. Unlike Implicit waits, explicit waits are applied for a particular instance only.

// explicit wait – to wait for the compose button to be click-able

```
import org.openqa.selenium.support.ui.ExpectedConditions
import org.openqa.selenium.support.ui.WebDriverWait
```

```
WebDriverWait wait = new WebDriverWait(drv,30);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(“”)));
```

Example:

```
wait.until(ExpectedConditions.elementToBeClickable(By.xpath(“//div[contains(text(),’COMPOSE’)]”)));
```

```
wait.until(ExpectedConditions.textToBePresentInElement(By.xpath(“//div[@id= ‘forgotPass’”), “text to be found”));
```

```
wait.until(ExpectedConditions.alertIsPresent()) !=null);
```

```
wait.until(ExpectedConditions.titleIs(“gmail”));
```

```
wait.until(ExpectedConditions.frameToBeAvailableAndSwitchToIt(By.id(“newframe”)));
```

- **How can we get a text of a web element?Syntax:**
  String Text = driver.findElement(By.id(“Text”)).getText();

- **How to select value in a dropdown?**

WebDriver’s Select class.

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);
```

- **How to handle frame in WebDriver?**

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));
```

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();
```

- **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.

```
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.

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

- **How can we handle web based pop up?**
    - void dismiss() – The accept() method clicks on the “Cancel” button as soon as the pop up window appears.
    - void accept() – The accept() method clicks on the “Ok” button as soon as the pop up window appears.
    - String getText() – The getText() method returns the text displayed on the alert box.

void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

Syntax:

```
Import org.openqa.selenium.Alert
// accepting javascript alert
            Alert alert = driver.switchTo().alert();
           alert.accept();
```

- **How can we handle Window pop up?**

// pressing keys with the help of keyPress and keyRelease events

import java.awt.Robot
rb.keyPress(KeyEvent.VK_D);
rb.keyRelease(KeyEvent.VK_D);

- **How to assert title of the web page?***//verify the title of the web page assertTrue("The title of the window is incorrect.",driver.getTitle().equals("Title of the page"));*
- **How to mouse hover on a web element using WebDriver?**`Actions actions=new Actions(driver);`
- **How to retrieve css properties of an element?**

Syntax:
driver.findElement(By.id("id")).getCssValue("name of css attribute");
driver.findElement(By.id("id")).getCssValue("font-size")

- **How to capture screenshot in WebDriver?**

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("C:\\CaptureScreenshot\\google.jpg"));

- What do you mean by 'Firefox driver is upcasted to Webdriver interface?

Webdriver is an Interface and FirefoxDriver is a class which has implemented Webdriver Interface. Firefox driver is upcasted to Webdriver interface means

```
Webdriver driver = new FirefoxDriver();
```

We do not follow the practice of writing something like this:

```
FirefoxDriver driver = new FirefoxDriver();
```

- **Dynamic Xpath if attribute value change**

Example – 'username_123' changed to 'username_234' but 'username' always remained constant.

driver.findElement(By.xpath("//[contains(@id,'username')]")).sendKeys("username");*
driver.findElement(By.xpath("//[starts-with(@id,'user')]")).sendKeys("username");*