# **Challenges and limitations of Selenium WebDriver:**

As we all know Selenium WebDriver is a tool that automates the browser to mimic real user actions on the web. Selenium is a free open-source testing tool. Some of the challenges with Selenium WebDriver are as follows

1. We cannot test the windows application
2. We cannot test mobile apps
3. Limited reporting
4. Handling dynamic Elements
5. Handling page load
6. Handling popup windows
7. Handling captcha

# **Difference between Assert and Verify in Selenium**

**Assert:** In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

**Verify:** In simple words, there won’t be any halt in the test execution even though the verify condition is true or false.

# **Types of waits available in Selenium WebDriver**

In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

* Implicit Waits
* Explicit Waits
* Fluent Waits

## **Implicit Waits**

Implicit waits tell to the WebDriver to wait for certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception. The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

## **Explicit Waits**

It is applied on certain element with defined *expected condition* and *time*. This wait is only applied to the specified element. This wait can also throw exception when element is not found.

The following are the Expected Conditions that can be used in Explicit Wait

* alertIsPresent()
* elementSelectionStateToBe()
* elementToBeClickable()
* elementToBeSelected()
* frameToBeAvaliableAndSwitchToIt()
* invisibilityOfTheElementLocated()
* invisibilityOfElementWithText()
* presenceOfAllElementsLocatedBy()
* presenceOfElementLocated()
* textToBePresentInElement()
* textToBePresentInElementLocated()
* textToBePresentInElementValue()
* titleIs()
* titleContains()
* visibilityOf()
* visibilityOfAllElements()
* visibilityOfAllElementsLocatedBy()
* visibilityOfElementLocated()

## **Fluent Waits**

FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an “ElementNotVisibleException” exception.

To say in effortless manner, it tries to find the web element repeatedly at regular intervals of time until the timeout or till the object gets found.

# **delete cookies in Selenium**

* To delete cookies we use *deleteAllCookies()* method

# **refresh a browser using Selenium WebDriver**

There are multiple ways to refresh a page in selenium

* Using driver.navigate().refresh() command as mentioned in the question 45
* Using driver.get(“URL”) on the current URL or using driver.getCurrentUrl()
* Using driver.navigate().to(“URL”) on the current URL or driver.navigate().to(driver.getCurrentUrl());
* Using sendKeys(Keys.F5) on any textbox on the webpage

# **getWindowHandle() and getWindowHandles()**

driver.getWindowHandle() – It returns a handle of the current page (a unique identifier)  
driver.getWindowHandles() – It returns a set of handles of the all the pages available.

### **driver.close() and driver.quit()?**

Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

*driver.close():* To close current WebDriver instance  
*driver.quit():* To close all the opened WebDriver instances

## 

**Selenium Python Tricks**

# **How to get auto suggest locator**

Simply select sources tab on the debugger, perform your search and let the suggestions pop-up.

Once you see the suggestions press F8. This action will freeze the web page and you can easily inspect them. To resume the page again you can use F8 to continue. **Refer to the attached image for more info.**