Difference between assert and verify

Both Assert and Verify commands are used to find whether a given input is present or not on the webpage. There are some difference between Assert and Verify in Selenium.

**Assert command in selenium:**

When an “assert” command fails, the test execution will be aborted. So when the Assertion fails, all the test steps after that line of code are skipped. The solution to overcoming this issue is to use a try-catch block. We use the Assertion in the try catch block

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 command in selenium:**

When a “verify” command fails, the test will continue executing and logging the failure. Mostly, the Verify command is used to check non-critical things. In such cases where we move forward even though the end result of the check value is failed.

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

**Note:** In TestNG, we use only Assert Statements.

**How To Use Soft Assert In TestNG**

Before knowing what is Soft Assert, first let’s see what is an Assert and what is the disadvantage in using Assert and why we are moving to Soft Assert.

There are two types of Assert:

1. Hard Assert
2. Soft Assert

When an assert fails the test script stops execution unless handled in some form. We call general assert as Hard Assert

**Hard Assert** – Hard Assert throws an *AssertException* immediately when an assert statement fails and test suite continues with next *@Test*

The disadvantage of Hard Assert – It marks method as fail if assert condition gets failed and the remaining statements inside the method will be aborted.

To overcome this we need to use Soft Assert. Let’s see what is Soft Assert.

**Soft Assert** – 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.

If there is any exception and you want to throw it then you need to use *assertAll()* method as a last statement in the @Test and test suite again continue with next *@Test* as it is.

Example with out assetAll

**package** SoftwareTestingMaterial;

**import** org.testng.Assert;

**import** org.testng.annotations.Test;

**import** org.testng.asserts.SoftAssert;

**public** **class** Assertions {

@Test

**public** **void** softAssert() {

SoftAssert softAssertion=**new** SoftAssert();

System.***out***.println("soft assertion method was stared");

softAssertion.assertTrue(**false**);

System.***out***.println("soft assertion method was failed");

}

@Test

**public** **void** hardAssert() {

System.***out***.println("hard assertion method was stared");

Assert.*assertTrue*(**false**);

System.***out***.println("hard assertion method was failed");

}

}

Output:

hard assertion method was stared

soft assertion method was stared

soft assertion method was failed

FAILED: hardAssert

Default suite

Total tests run: 2, Failures: 1, Skips: 0

Conclusion:

Hard Assert:

hard assertion method was failed—is aborted, since hard assert got failed, it will abort the next step.

Failures: 1—is hard assert

SoftAssert:

soft assertion method was stared

soft assertion method was failed

Even though soft assert failed next step was executed.