ASSERTIONS

IN SELENIUM

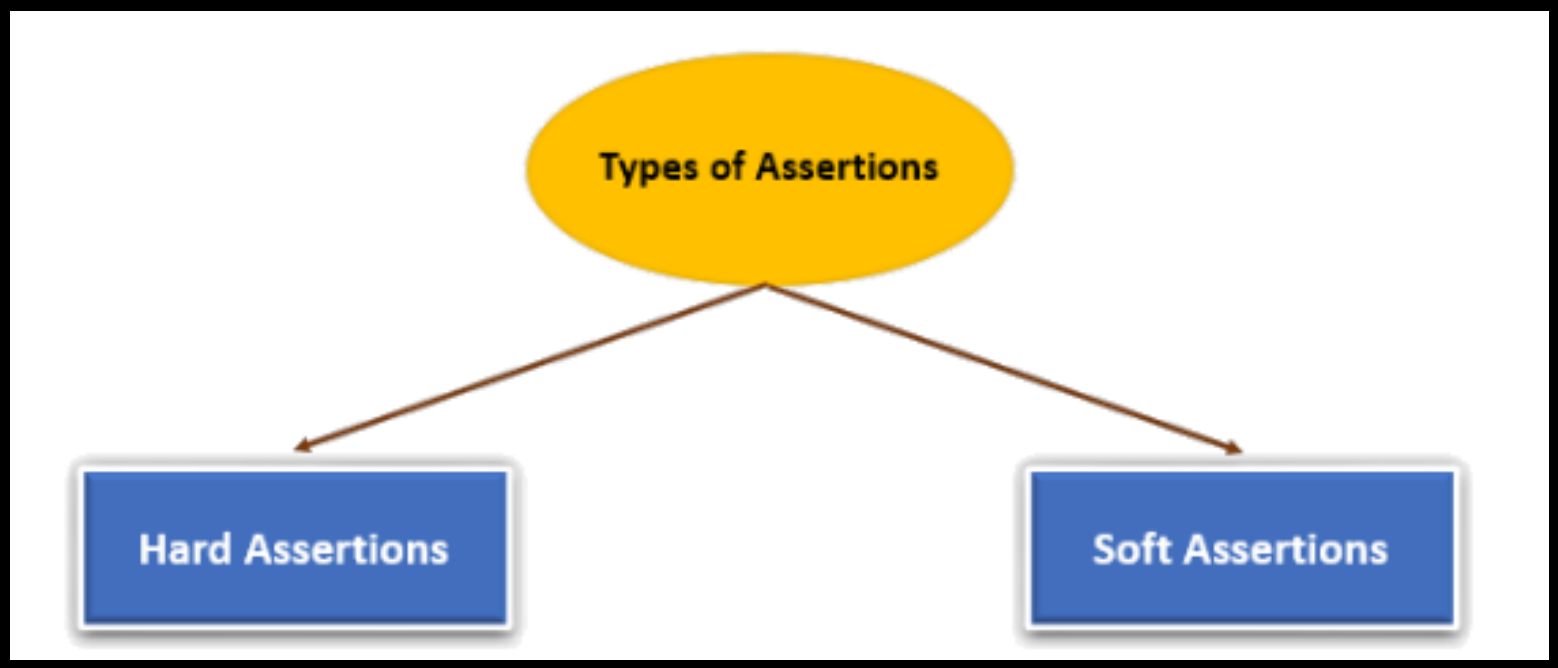
Definition:

The word Assertion mean -" A statement that you strongly believe is true".

"Assertion is used to compare the actual result of an application with the expected result".Generally assertions verifies whether the application is same or not when we check with our expectation.

Types Of Assertions:

- Hard Assertions
- Soft Assertions (Verify Methods)
- Wait for method is also consider as a type of assertion



Hard Assert

- A hard assertion does not continue with execution until the assertion condition is True.
- Hard assertions usually throw an Assertion Error whenever an assertion condition fails.
- The test case will be immediately marked as Failed when a hard assertion condition fails.
- By default, Assert in Selenium WebDriver are Hard Asserts.

Soft Assert

- A soft assertion continues with test execution even if the assertion condition fails.
- Soft Assertion does not throw any error when the assertion condition fails but continues with the next step of the test case.

Assertion Methods:

- assertEquals
- assertNotEquals
- assertTrue
- assertFalse
- assertNull
- assertNotNull
- assertAll()

assertEquals:

- This is used to compare expected and actual values in selenium webdriver. Whenever the expected and actual values are same, the assertion passes with no exception. But, if the actual and expected values are not just same, the assert fails with an exception and the test is marked as “failed”. The suite continues to run with the next @Test annotation(if any).

```
Assert.assertEquals(actual,expected);
```

assertNotEquals:

- assertNotEquals is just opposite to the functioning of assertEquals assertion. Whenever the expected and actual values matches, the assertion fails with an exception and marks the test-case as “failed”. The particular testcase is aborted and execution continuous with the next @Test annotation.

```
Assert.assertNotEquals(actual,expected,
                Message);
```

assertTrue:

- When we are dealing with Boolean conditions, we should use assertTrue. This assertion returns true if the applied condition passes. If the condition is false/fails, this assertion skips the current method from execution.

```
Assert.assertTrue(condition);
```

assertFalse:

- Assert.assertFalse checks the Boolean value returned by a condition is false or not. When a condition value is true, the assertion aborts the method by an exception. This is basically opposite to assertTrue.

```
Assert.assertFalse(condition);
```

assertNull:

- This assertion checks for a object, if it is null or not. When an object is ‘null’ the assertion returns an exception resulting in aborting the test.

```
Assert.assertNull(object);
```

assertNotNull:

- Assert.assertNotNull is vice-versa of assertNull. When a object has some value, the assertion aborts the method with an exception.

```
Assert.assertNotNull(object);
```

assertAll:

When using assertAll(), it allows you to verify all the assertions made in a test case, and it collects all the assertion errors in a single test case, instead of failing the test immediately when an assertion fails. This allows you to see all the assertion errors in one go, which can be helpful for troubleshooting and debugging.

```
SoftAssert softAssert = new SoftAssert();
softAssert.assertAll();
```