| S.No. | Selenium Remote Webdriver | Selenium Webdriver |
| --- | --- | --- |
| 1. | It is a class which implements a Webdriver interface. | It is an interface and to use this, org.openqa.selenium.\* package is required. |
| 2. | It has some extra methods which are used for the implementation of the class. | It has a lesser number of methods as compared to Remote Webdriver, used for the interface implementation. |
| 3. | In this, the object is used to manage browser which is present in the grid. | In this, many browsers are managed by the Webdriver object. |
| 4. | Remote Webdriver is a part of Webdriver. | It is not part of any Webdriver. |
| 5. | It provides methods like startSession(), getSessionId(), etc. | It provides methods like quit(), get(), etc. |
| 6. | To test this, the remote machine is required. | It can be tested on the local machine. |
| 7. | It does not depend on the OS. | It depends on the OS. |

ASSERTIONS IN SELENIUM([https://www.softwaretestinghelp.com/assertions-in-selenium/)-->refer](https://www.softwaretestinghelp.com/assertions-in-selenium/)--%3erefer) the link for various assertion in testng

Assertions are used for validating a test case and helps us understand if a test case has passed or failed.

The assertion is considered to be met if the actual result of an application matches with that of the expected result.

We consider a test as successful if it runs without any exception.

**Two types of  assertions in**[**Selenium**](https://www.softwaretestinghelp.com/selenium-tutorial-1/)**:**

* Hard Assertions
* Soft Assertions

**Hard Assertions**

* A hard assertion does not continue with execution until the assertion condition is met.
* Hard assertions usually throw an Assertion Error whenever an assertion condition has not been met. The test case will be immediately marked as Failed when a hard assertion condition fails.
* Eg:when you want to verify if you have logged in correctly and fail the test if you haven’t made a successful login, as there is no point in proceeding further if the pre-condition(login) itself fails.

**Soft Assertions**

* A soft assertion continues with the next step of the test execution even if the assertion condition is not met.
* Soft Assertions are the type of assertions that do not throw an exception automatically when an assertion fails unless it is asked for. This is useful if you are doing multiple validations in a form, out of which only a few validations directly have an impact on deciding the test case status.
* Here, we use a class called SoftAssert and the method assertAll() is called to throw all exceptions caught during execution. When softAssert is used, it performs assertion and if an exception is found, its not thrown immediately, rather it continues until we call the method assertAll() to throw all exceptions caught.
* It is wise to use different objects of ‘SoftAssert’ class for each test case.

### When To Use Hard And Soft Assertion?

If you need to execute all the steps of a test case to be executed even after an assertion fails, and you also want to report assertion exception, then opt for using Soft Assertions. Using Soft Assertions in your test scripts is a good practice and an effective way of handling your test execution

If you want your test case execution to proceed only after an assertion is passed (**For Example,** To Verify valid login and only then execute the other steps), then use Hard Assertions.