**Assertions in Junit**

Assertion is used to check whether the condition is true or false

Below are some assert methods

**Boolean:** **If you want to test the Boolean conditions (true or false).**

assertTrue(condition)

assertFalse(condition)

Here, the condition is a boolean value.

### Identical: If you want to check the initial value of an object/variable, you have the following methods:

assertNull(object)

assertNotNull(object)

Here, the object is a[Java](https://www.guru99.com/java-tutorial.html)object, for e.g**.** assertNull(actual);

### Null object: If you want to check whether the objects are identical (i.e. comparing two references to the same java object) or different, follow the below methods:

assertSame(expected, actual), It will return true if expected == actual

assertNotSame(expected, actual)

### Assert Equals: If you want to test the equality of two objects, you have the following methods:

assertEquals(expected, actual)

It will return true if:expected.equals( actual )returns true.

### Assert Array Equals :

assertArrayEquals(expected, actual)

The above method must be used if the arrays have the same length for each valid value for **i** as shown below:

assertEquals(expected[i],actual[i])

assertArrayEquals(expected[i],actual[i])

### Fail Message:

If you want to throw any assertion error, you have fail() that always results in a fail verdict.

Fail(message);

You can have the assertion method with an additional stringparameter as the first parameter. This string will be appended in the failure message if the assertion fails. E.g**.**fail( message )can be written as:

assertEquals (message, expected, actual)

let us write a simple program to explain assertions in Junit

public class TestAssertions {

@Test

public void testAssertions() {

//test data

String str1 = new String ("abc");

String str2 = new String ("abc");

String str3 = null;

String str4 = "abc";

String str5 = "abc";

int val1 = 5;

int val2 = 6;

String[] expectedArray = {"one", "two", "three"};

String[] resultArray = {"one", "two", "three"};

//Check that two objects are equal

assertEquals(str1, str2);

//Check that a condition is true

assertTrue (val1 < val2);

//Check that a condition is false

assertFalse(val1 > val2);

//Check that an object isn't null

assertNotNull(str1);

//Check that an object is null

assertNull(str3);

//Check if two object references point to the same object

assertSame(str4,str5);

//Check if two object references not point to the same object

assertNotSame(str1,str3);

//Check whether two arrays are equal to each other.

assertArrayEquals(expectedArray, resultArray);

}

}