

JUnit

Method Summery

`void #assertEquals([message], expected, actual)`
Asserts that two values are equal.

`void #assertFalse([message], boolean_condition)`
Asserts that a condition is false.

`void #assertFloatEquals([message], expectedFloat, actualFloat, tolerance)`
Asserts that two floating point values are equal to within a given tolerance.

`void #assertMatches([message], regularExpression, actualString)`
Asserts that a regular expression matches a string.

`void #assertNotNull([message], object)`
Asserts that an object is not null.

`void #assertNotSame([message], notExpected, actual)`
Asserts that two values are not the same.

`void #assertNotUndefined([message], definedObject)`
Asserts that an object is not undefined.

`void #assertNull([message], nullValue)`
Asserts that an object is null.

`void #assertSame([message], expected, actual)`
Asserts that two values are the same.

`void #assertTrue([message], boolean_condition)`
Asserts that a condition is true.

`void #assertUndefined([message], undefinedValue)`
Asserts that an object is undefined.

`void #fail([message], [instanceOfCallStack], [userMessage])`
Fails a test.

Method Details

`assertEquals`

`void assertEquals([message], expected, actual)`
Asserts that two values are equal. `AssertionFailedError` is thrown if the actual value does not match the regular expression.

Parameters

`[message]` – The test description/message.
`expected` – the expected value.
`actual` – the actual value.

Returns

`void`

Sample

```
// Asserts that two values are equal. AssertionFailedError is thrown if the actual value does not match the
regular expression.
jsunit.assertEquals("Solution name test", "someSolution", application.getSolutionName());
jsunit.assertEquals("Simple math test", 2, 1 + 1);
```

`assertFalse`

`void assertFalse([message], boolean_condition)`
Asserts that a condition is false. `AssertionFailedError` is thrown if the evaluation was not false.

Parameters

`[message]` – The test description/message.
`boolean_condition` – the actual value.

Returns

`void`

Sample

```
// Asserts that a condition is false. AssertionFailedError is thrown if the evaluation was not false.
jsunit.assertFalse("False test", application.isLastPrintPreviewPrinted());
```

`assertFloatEquals`

`void assertFloatEquals([message], expectedFloat, actualFloat, tolerance)`
Asserts that two floating point values are equal to within a given tolerance. `AssertionFailedError` is thrown if the expected value is not within the tolerance of the actual one.

Parameters

[message] – The test description/message.

expectedFloat – the expected value.

actualFloat – the actual value.

tolerance – float tolerance when comparing.

Returns

void

Sample

```
// Asserts that two floating point values are equal to within a given tolerance. AssertionError is thrown
if the expected value is not within the tolerance of the actual one.
jsunit.assertFloatEquals("Float equals test", 3.12, 3.121, 0.0015);
```

assertMatches

void **assertMatches**([message], regularExpression, actualString)

Asserts that a regular expression matches a string. AssertionError is thrown if the expected value is not the actual one.

Parameters

[message] – The test description/message.

regularExpression – the regular expression used for matching.

actualString – the actual value to be matched.

Returns

void

Sample

```
// Asserts that a regular expression matches a string. AssertionError is thrown if the expected value is
not the actual one.
jsunit.assertMatches("Match test", new RegExp("gr(a|e)y"), "gray");
```

assertNotNull

void **assertNotNull**([message], object)

Asserts that an object is not null. AssertionError is thrown if the object is not null.

Parameters

[message] – The test description/message.

object – the actual value.

Returns

void

Sample

```
// Asserts that an object is not null. AssertionError is thrown if the object is not null.
var a; // this is undefined, not null
jsunit.assertNotNull("Not null test", a);
```

assertNotSame

void **assertNotSame**([message], notExpected, actual)

Asserts that two values are not the same. AssertionError is thrown if the expected value is the actual one.

Parameters

[message] – The test description/message.

notExpected – the value that is not expected.

actual – the actual value.

Returns

void

Sample

```
// Asserts that two values are not the same. AssertionError is thrown if the expected value is the actual
one.
var a = new Date(1990, 1, 1);
var b = new Date(1990, 1, 1);
jsunit.assertNotSame("Not same test", a, b);
jsunit.assertEquals("But equals", a, b);
```

assertNotUndefined

void **assertNotUndefined**([message], definedObject)

Asserts that an object is not undefined. AssertionError is thrown if the object is undefined.

Parameters

[message] – The test description/message.
definedObject – the actual value.

Returns

void

Sample

```
// Asserts that an object is not undefined. AssertionError is thrown if the object is undefined.
var a = 0;
jsunit.assertNotUndefined("Not undefined test", a);
```

assertNull

void **assertNull**([message], nullValue)

Asserts that an object is null. AssertionError is thrown if the object is not null.

Parameters

[message] – The test description/message.
nullValue – the actual value.

Returns

void

Sample

```
// Asserts that an object is null. AssertionError is thrown if the object is not null.
jsunit.assertNull("Null test", null);
```

assertSame

void **assertSame**([message], expected, actual)

Asserts that two values are the same. AssertionError is thrown if the expected value is not the actual one.

Parameters

[message] – The test description/message.
expected – the expected value.
actual – the actual value.

Returns

void

Sample

```
// Asserts that two values are the same. AssertionError is thrown if the expected value is not the actual
one.
var a = new Date(1990, 1, 1);
var b = a;
jsunit.assertSame("Same test", a, b);
```

assertTrue

void **assertTrue**([message], boolean_condition)

Asserts that a condition is true. AssertionError is thrown if the evaluation was not true.

Parameters

[message] – The test description/message.
boolean_condition – the actual value.

Returns

void

Sample

```
// Asserts that a condition is true. AssertionError is thrown if the evaluation was not true.
jsunit.assertTrue("True test", application.isLastPrintPreviewPrinted());
```

assertUndefined

void **assertUndefined**([message], undefinedValue)

Asserts that an object is undefined. AssertionError is thrown if the object is defined.

Parameters

[message] – The test description/message.
undefinedValue – the actual value.

Returns

void

Sample

```
// Asserts that an object is undefined. AssertionError is thrown if the object is defined.  
jsunit.assertUndefined("Undefined test", thisIsUndefined);
```

fail

void **fail**([message], [instanceOfCallStack], [userMessage])

Fails a test. AssertionError is always thrown.

Parameters

[message] – The test description/message. This is usually the only parameter specified when calling this method.

[instanceOfCallStack] – an internal JSUnit call stack. Use null for this if you want to get to the next optional parameter. Usually not specified.

[userMessage] – an user message. Usually not specified.

Returns

void

Sample

```
// Fails a test. AssertionError is always thrown.  
jsunit.fail("Fail test");  
jsunit.fail("test", null, "Fail"); // 2nd param is not used in Servoy, params 3 and 1 get merged to form a  
message. The result is the same as in the line above.
```