

## Celestra Assert plugin cheatsheet – v5.8.1 – <https://github.com/Serrin/Celestra/>

Plugin properties	Description
<code>assert(condition [, message   error]);</code>	<code>celestra.assert(condition [, message   error]);</code>
<code>assert.strict = true</code>	Default value is true. If this property is true, then always the strict equality is used in the <code>assert.equal();</code> , <code>assert.notEqual();</code> , <code>assert.deepEqual();</code> , <code>assert.notDeepEqual();</code>
<code>assert.deepEqual(value1, value2 [, message   error]);</code>	<code>celestra.deepStrictEqual(value1, value2 [, message   error]);</code> or <code>celestra.assertDeepEqual(value1, value2 [, message   error]);</code>
<code>assert.doesNotMatch(string, regexp[, message   error]);</code>	<code>celestra.assertDoesNotMatch(string, regexp[, message   error]);</code>
<code>assert.equal(value1, value2 [, message   error]);</code>	<code>celestra.assertStrictEqual(value1, value2 [, message   error]);</code> or <code>celestra.assertEqual(value1, value2 [, message   error]);</code>
<code>assert.fail(message OR error);</code>	<code>celestra.assertFail(message OR error);</code>
<code>assert instanceof (value, constructor [, message   error]);</code>	<code>celestra.assertInstanceOf(value, constructor [, message   error]);</code>
<code>assert.isFalse(condition [, message   error]);</code>	<code>celestra.assertFalse(condition [, message   error]);</code>
<code>assert.isNil(value [, message   error]);</code>	<code>celestra.assertIsNil(value [, message   error]);</code>
<code>assert.isNotNil(value [, message   error]);</code>	<code>celestra.assertIsNotNil(value [, message   error]);</code>
<code>assert.isTrue(condition [, message   error]);</code>	<code>celestra.assertTrue(condition [, message   error]);</code>
<code>assert.match(string, regexp[, message   error]);</code>	<code>celestra.assertMatch(string, regexp[, message   error]);</code>
<code>assert.notDeepEqual(value1, value2 [, message   error]);</code>	<code>celestra.assertNotDeepStrictEqual(value1, value2 [, message   error]);</code> or <code>celestra.assertNotDeepEqual(value1, value2 [, message   error]);</code>
<code>assert.notEqual(value1, value2 [, message   error]);</code>	<code>celestra.assertNotStrictEqual(value1, value2 [, message   error]);</code> or <code>celestra.assertNotEqual(value1, value2 [, message   error]);</code>
<code>assert.notInstanceOf(value, constructor [, message   error]);</code>	<code>celestra.assertNotInstanceOf(value, constructor [, message   error]);</code>
<code>assert.notStrictEqual(value1, value2 [, message   error]);</code>	<code>celestra.assertNotStrictEqual(value1, value2 [, message   error]);</code>
<code>assert.notTypeOf(value, type [, message   error]);</code>	<code>celestra.assertNotTypeOf(value, type [, message   error]);</code>
<code>assert.ok(condition [, message   error]);</code>	<code>celestra.assert(condition [, message   error]);</code>
<code>assert.strictEqual(value1, value2 [, message   error]);</code>	<code>celestra.assertStrictEqual(value1, value2 [, message   error]);</code>
<code>assert.Throws(callback [, message   error]);</code>	<code>celestra.assertThrows(callback [, message   error]);</code>
<code>assert.typeOf(value, type [, message   error]);</code>	<code>celestra.assertTypeOf(value, type [, message   error]);</code>