Celestra Assert plugin cheatsheet – v5.8.0 – https://github.com/Serrin/Celestra/

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