

The list below summarises the methods available, along with their signature and purpose:

`deepEqual(value, expected[, message])`: A recursive, strict comparison that works on all the JavaScript types. The assertion passes if `value` and `expected` are identical in terms of properties, values, and they have the same prototype;

`equal(value, expected[, message])`: Verify the `value` provided is equal the `expected` parameter using a non-strict comparison (`==`).

`notDeepEqual(value, expected[, message])`: Same as `deepEqual()` but tests for inequality;

`notEqual(value, expected[, message])`: Same as `equal()` but tests for inequality;

`propEqual(value, expected[, message])`: A strict comparison of the properties and values of an object. The assertion passes if all the properties and the values are identical;

`strictEqual(value, expected[, message])`: Verify the `value` provided is equal to the `expected` parameter using a strict comparison (`===`);

`notPropEqual(value, expected[, message])`: Same as `propEqual()` but tests for inequality;

`notStrictEqual(value, expected[, message])`: Same as `strictEqual()` but tests for inequality;

`ok(value[, message])`: An assertion that passes if the first argument is truthy;

`throws(function [, expected ] [, message ])`: Test if a callback throws an exception, and optionally compare the thrown error;

The meaning of the parameters accepted by these methods is described below:

`value`: The value returned by a function, a method, or stored in a variable that has to be verified;

`expected`: The value to test against. In case of the `throws()` method, this can be an "Error Object (instance), Error Function (constructor), a RegExp that matches (or partially matches) the String representation, or a callback Function that must return true to pass the assertion check";

`message`: An optional string describing the assertion;

`function`: The function to execute that should return an Error;