

Global Environment

| | |
|---|---|
| <code>afterEach (fn)</code> | Called once after each test |
| <code>beforeEach (fn)</code> | Called once before each test |
| <code>afterAll (fn)</code> | Called only once after all the tests in describe finish |
| <code>beforeAll (fn)</code> | Called only once before all the tests in describe |
| <code>describe (name, fn)</code> | Test suite describer |
| <code>expect (value)</code> | Starts an expectation chain |
| <code>expect.extend (matchers)</code> | Adds more matchers to expect |
| <code>expect.<asymmetric-match>()</code> | Used as value to match to |
| <code>it (name, fn)</code> | Test describer (another alias for it is test) |
| <code>it.only (name, fn)</code> | Skip all tests after this one |
| <code>it.skip (name, fn)</code> | Skip this test |
| <code>fit (name, fn)</code> | Runs only this test |
| <code>jest</code> | The jest object |
| <code>require.requireActual (moduleName)</code> | Returns the actual module instead of a mock |
| <code>require.requireMock (moduleName)</code> | Returns a mock module instead of the actual module |
| <code>xdescribe/xit/xtest</code> | Makes the test pending |
| <code>fdescribe/fit</code> | Makes the test/suite the only one running |

Writing assertions with expect

| | |
|--------------------------------------|--|
| <code>.not</code> | Expects the opposite of the rest of the expectation. |
| <code>.toBe (value)</code> | Strict equality (===) |
| <code>.toEqual (value)</code> | Recursively checks the equality of all fields |
| <code>.toMatch (regexp)</code> | |
| <code>.toMatchObject (object)</code> | Matches a subset of the properties of an object |

Writing assertions with expect (cont)

| | |
|--|--|
| <code>.toMatchSnapshot ()</code> | Matches the most recent snapshot (or creates one) |
| <code>.toBeDefined ()</code> | |
| <code>.toBeUndefined ()</code> | |
| <code>.toBeNull ()</code> | |
| <code>.toBeTruthy ()</code> | |
| <code>.toBeFalsy ()</code> | |
| <code>.toBeGreaterThan (number)</code> | |
| <code>.toBeGreaterThanOrEqual (number)</code> | |
| <code>.toBeLessThan (number)</code> | |
| <code>.toBeLessThanOrEqual (number)</code> | |
| <code>.toBeCloseTo (number, numDigits)</code> | |
| <code>.toBeNaN ()</code> | |
| <code>.toBeInstanceOf (Class)</code> | |
| <code>.toContain (item)</code> | Strict equality (===) |
| <code>.toContainEqual (item)</code> | Recursively checks the equality of all fields |
| <code>.toHaveLength (number)</code> | |
| <code>.toThrow ()</code> | |
| <code>.toThrowError (error)</code> | error can be a string, RegExp or Class |
| <code>.toThrowErrorMatch (snapshot)</code> | Will expect an error while matching (the snapshot will be the error's message) |
| <code>.toHaveBeenCalled ()</code> | |
| <code>.toHaveBeenCalledTimes (number)</code> | |
| <code>.toHaveBeenCalledWith (arg1, arg2, ...)</code> | |
| <code>.toHaveBeenCalledLastCalledWith (arg1, arg2, ...)</code> | |

Mock functions

| | |
|--|--|
| <code>jest.fn()</code> | Creates a mock function |
| <code>mockFn.mock.calls</code> | An array that represents all calls that have been made into this mock function |
| <code>mockFn.mock.instances</code> | An array that contains all the object instances |
| <code>mockFn.mockClear()</code> | Resets all information stored in the above arrays. |
| <code>mockFn.mockReset()</code> | Resets all information stored in the mock |
| <code>mockFn.mockImplementation(fn)</code> | Sets a function that should be used as the implementation of the mock |
| <code>mockFn.mockImplementationOnce(fn)</code> | Sets a function that should be used as the implementation of the mock for one call |
| <code>mockFn.mockReturnThis()</code> | Suger for using <code>jest.fn(() => this)</code> |
| <code>mockFn.mockReturnValueOnce(value)</code> | |

The jest object

| | |
|--|---|
| <code>jest.resetAllMocks()</code> | Resets the state of all mocks |
| <code>jest.clearAllTimers()</code> | Removes any pending timers from the timer system |
| <code>jest.disableAutomock()</code> | Disables automatic mocking in the module loader |
| <code>jest.enableAutomock()</code> | Enables automatic mocking in the module loader |
| <code>jest.fn(? implementation)</code> | Returns a new, unused mock function |
| <code>jest.isMockFunction(fn)</code> | Determines if the given function is a mocked function |

The jest object (cont)

| | |
|--|---|
| <code>jest.genMockFromModule(moduleName)</code> | Generate a mocked version of the module for you |
| <code>jest.mock(moduleName, ?factory, ?options)</code> | Mocks a module with an auto-mocked version |
| <code>jest.resetModules()</code> | Resets the module registry - the cache of all required modules |
| <code>jest.runAllTicks()</code> | Exhausts the micro -task queue |
| <code>jest.runAllTimers()</code> | Exhausts the macro -task queue |
| <code>jest.runAllImmediate()</code> | Exhausts all tasks queued by <code>setImmediate()</code> |
| <code>jest.runTimersToTime(msToRun)</code> | Executes only the macro task queue |
| <code>jest.runOnlyPendingTimers()</code> | Executes only the macro-tasks that are currently pending |
| <code>jest.setMock(moduleName, moduleExports)</code> | Explicitly supplies the mock object that the module system should return for the specified module |
| <code>jest.unmock(moduleName)</code> | Indicates that the module system should never return a mocked version of the specified module |
| <code>jest.useFakeTimers()</code> | Instructs Jest to use fake versions of the standard timer functions |
| <code>jest.useRealTimers()</code> | Instructs Jest to use the real versions of the standard timer functions |

