search

Fork me on GitHub

Jasmine

Behavior-Driven JavaScript

GET STARTED DOCS SUPPORT RELEASES GITHUB

Namespaces

async-matchers
jasmine
matchers
Spy#calls

Classes

Clock
Env
jsApiReporter
Spy

Interfaces

Configuration
Reporter
SpyStrategy

Global

afterAll
afterEach
beforeAll
beforeEach
describe
expect
expectAsync
fail
fdescribe
fit
it
pending
spyOn
spyOnAllFunctions

Namespace: matchers

matchers

Matchers that come with Jasmine out of the box.

Members

`not :matchers`

Invert the matcher following this `expect`

Type:

matchers

Since: 1.3.0

Example

```
expect(something).not.toBe(true);
```

Methods

`nothing()`

`expect` nothing explicitly.

Since: 2.8.0

Since: 1.3.0

Example

```
expect(thing).toBe(realThing);
```

toBeCloseTo(expected, precision*opt*)

`expect` the actual value to be within a specified precision of the expected value.

Parameters:

Name	Type	Attributes	Default	Description
`expected`	Object			The expected value to compare against.
`precision`	Number	<optional>	2	The number of decimal points to check.

Since: 1.3.0

Example

```
expect(number).toBeCloseTo(42.2, 3);
```

toBeDefined()

`expect` the actual value to be defined. (Not `undefined`)

Since: 1.3.0

Example

```
expect(result).toBeDefined();
```

`expect` the actual value to be falsy

Since: 2.0.0

Example

```
expect(result).toBeFalsy();
```

`toBeGreaterThan(expected)`

`expect` the actual value to be greater than the expected value.

Parameters:

Name	Type	Description
`expected`	Number	The value to compare against.

Since: 2.0.0

Example

```
expect(result).toBeGreaterThan(3);
```

`toBeGreaterThanOrEqual(expected)`

`expect` the actual value to be greater than or equal to the expected value.

Parameters:

Name	Type	Description
`expected`	Number	The expected value to compare against.

Since: 2.0.0

Example

```
expect(result).toBeGreaterThanOrEqual(25);
expect('foo').toBeInstanceOf(String);
expect(3).toBeInstanceOf(Number);
expect(new Error()).toBeInstanceOf(Error);
```

`toBeLessThan(expected)`

`expect` the actual value to be less than the expected value.

Parameters:

Name	Type	Description
`expected`	Number	The expected value to compare against.

Since: 2.0.0

Example

```
expect(result).toBeLessThan(0);
```

`toBeLessThanOrEqual(expected)`

`expect` the actual value to be less than or equal to the expected value.

Parameters:

Name	Type	Description
`expected`	Number	The expected value to compare against.

Since: 2.0.0

Example

```
expect(result).toBeLessThanOrEqual(123);
```

`toBeNaN()`

Since: 2.6.0

Example

```
expect(thing).toBeNegativeInfinity();
```

`toBeNull()`

`expect` the actual value to be `null`.

Since: 1.3.0

Example

```
expect(result).toBeNull();
```

`toBePositiveInfinity()`

`expect` the actual value to be `Infinity` (infinity).

Since: 2.6.0

Example

```
expect(thing).toBePositiveInfinity();
```

`toBeTrue()`

`expect` the actual value to be `true`.

Since: 3.5.0

Example

```
expect(result).toBeTrue();
```

`expect` the actual value to be `undefined`.

> Since: 1.3.0

Example

```
expect(result).toBeUndefined():
```

`toContain(expected)`

`expect` the actual value to contain a specific value.

Parameters:

Name	Type	Description
`expected`	Object	The value to look for.

> Since: 2.0.0

Example

```
expect(array).toContain(anElement);
expect(string).toContain(substring);
```

`toEqual(expected)`

`expect` the actual value to be equal to the expected, using deep equality comparison.

Parameters:

Name	Type	Description
`expected`	Object	Expected value

> Since: 1.3.0

Example

`toHaveBeenCalledBefore(expected)`

`expect` the actual value (a `Spy`) to have been called before another `Spy`.

Parameters:

Name	Type	Description
`expected`	Spy	`Spy` that should have been called after the `actual` `Spy`.

Since: 2.6.0

Example

```
expect(mySpy).toHaveBeenCalledBefore(otherSpy);
```

`toHaveBeenCalledTimes(expected)`

`expect` the actual (a `Spy`) to have been called the specified number of times.

Parameters:

Name	Type	Description
`expected`	Number	The number of invocations to look for.

Since: 2.4.0

Example

```
expect(mySpy).toHaveBeenCalledTimes(3);
```

`toHaveBeenCalledWith()`

`expect` the actual (a `Spy`) to have been called with particular arguments at least once.

Parameters:

toHaveClass(expected)

`expect` the actual value to be a DOM element that has the expected class

Parameters:

Name	Type	Description
`expected`	Object	The class name to test for

Since: 3.0.0

Example

```
var el = document.createElement('div');
el.className = 'foo bar baz';
expect(el).toHaveClass('bar');
```

toMatch(expected)

`expect` the actual value to match a regular expression

Parameters:

Name	Type	Description
`expected`	RegExp | String	Value to look for in the string.

Since: 1.3.0

Example

```
expect("my string").toMatch(/string$/);
expect("other string").toMatch("her");
```

toThrow(expected*opt*)

`expect` a function to `throw` something.

Parameters:

Name	Type	Attributes	Description

search

Fork me on GitHub

Jasmine

Behavior-Driven JavaScript

toThrowError(expected*opt*, message*opt*)

`expect` a function to `throw` an `Error`.

Parameters:

Name	Type	Attributes	Description
`expected`	Error	<optional>	`Error` constructor the object that was thrown needs to be an instance of. If not provided, `Error` will be used.
`message`	RegExp | String	<optional>	The message that should be set on the thrown `Error`

Since: 2.0.0

Example

```
  expect(function() { return 'things'; }).toThrowError(MyCusto
  expect(function() { return 'things'; }).toThrowError(MyCusto
  expect(function() { return 'stuff'; }).toThrowError(MyCustom
  expect(function() { return 'other'; }).toThrowError(/foo/);
  expect(function() { return 'other'; }).toThrowError();
```

toThrowMatching(predicate)

`expect` a function to `throw` something matching a predicate.

Parameters:

Name	Type	Description
`predicate`	function	A function that takes the thrown exception as its parameter and returns true if it matches.

Since: 3.0.0

Since: 3.3.0

Returns:

Type matchers

Documentation generated by JSDoc 3.6.3 on Mon Sep 23 2019 16:24:45 GMT-0700 (Pacific Daylight Time)