Jasmine matchers

- toBe():

Expects the actual value to be equal (===) to the expected value.

- toEqual():

Similar to the toBe method but it uses deep equality comparison.

- toBeFalsy():

Expects the actual value to be falsy.

```
describe('test.js', function() {
    it('is true', function() {
        expect(true).toBe(true);
    });

    it('has same keys and values', function() {
        expect(obj1).toEqual(obj2);
    });

    it('is falsy', function() {
        expect(0).toBeFalsy();
    });
});
```

Jasmine matchers

- toBeTruthy():

Expects the actual value to be truthy.

- toBeNull():

Expects the actual value to be null.

- toBeDefined():

Expects the actual value to be defined.

```
describe('test.js', function() {
        it('is truthy', function() {
                expect({}).toBeTruthy();
        });

        it('is null', function() {
                expect(result).toBeNull();
        });

        it('is defined', function() {
                expect(result).toBeDefined();
        });
});
```

Jasmine matchers

- toContain():

Expects the actual value to contain a specific value.

- not:

This is a chain method that can be used with all matchers.

- toBeNaN():

Expects the actual value to be NaN (Not a Number).

```
describe('test.js', function() {
	it('contains 1', function() {
		var array = [1, 2, 3];
		expect(array).toContain(1);
	});

	it('is not true', function() {
		expect(true).not.toBe(false);
	});

	it('is NaN', function() {
		expect('4'*5).toBeNaN();
	});
});
```

Jasmine matchers

- toThrow():

Expects a function to throw something.

- toThrowError():

Expects a function to throw an error.

- toMatch():

Expects the actual value to match a regular expression.

```
describe('test.js', function() {
        it('throws', function() {
                const err = function(message) {
                        throw new Error(message);
                };
                expect(err).toThrow();
        });

        it('throws an error', function() {
                const err = function(message) {
                        throw new Error();
                };
                expect(err).toThrowError();
        });

        it('matches regex', function() {
                const STR = 'my string';
                expect(STR).toMatch(/string$/);
        });
});
```