JASMINE 2 CHEAT SHEET

daveceddia.com
@dceddia

BASICS

A Test

```
describe("A test", function() {
  it("should pass", function() {
    var passing = true;
    expect(passing).toBe(true);
  });
);
```

An expectation

```
expect(greatness).toBe(true);
```

Before and After

```
describe("before & after", function() {
  beforeEach(function() {
    // set up mocks, spies, etc.
  });

  afterEach(function() {
    // clean up
  });
});
```

MATCHERS

```
expect(obj).toBe(null);
```
objects must be *the same object*

```
expect(obj).toEqual({id: 7});
expect(msg).toMatch(/abc/);
```
Tests a regular expression.

```
expect(obj).toBeDefined();
expect(obj).toBeUndefined();
expect('a').toBeTruthy();
expect(obj).toBeFalsy();
```
Falsy values: **false**, **0**, **""**, **null**, **undefined**, **NaN**

```
expect(arr).toContain();
```
Find an item in an array.

```
expect(7).toBeLessThan(42);
expect(42).toBeGreaterThan(7);
expect(1.2).toBeCloseTo(1.23,
1);
```
2nd arg is decimal precision. 0 rounds.

```
expect(fn).toThrow();
```
Passes if `fn` throws an exception.

```
expect(obj).toThrowError();
expect(obj).toBeNull();
```

SPIES

Create a spy

```
spyOn(obj, 'method');
jasmine.createSpy('optional name');
jasmine.createSpyObj('name',
    ['fn1', 'fn2', ...]);
```

Modify spy behavior

```
obj.method.and.callThrough();
obj.method.and.returnValue(val);
obj.method.and.callFake(fn);
obj.method.and.throwError(err);
obj.method.and.stub();
```

Count/verify calls

```
obj.method.calls.any();
obj.method.calls.count();
obj.method.calls.reset();
```

Inspect calls

```
obj.method.calls.first();
obj.method.calls.mostRecent();
obj.method.calls.all();
```

Call description object

```
{
 object: {...}, // 'this' object
 args: []       // the arguments passed
}
```