Testing Javascript with Jasmine

unit testing

independent checking for proper operation of the *smallest testable* part of an application

- Independent: isolated from the rest of the system
- Smallest testable – means you have to approach the all of code development from this prospective
- Should be repeatable

mocking

technique where dependency and its behavior is imitated (or faked)

- Can be done by the developer or a mocking library

Jasmine

Steps for Jasmine-based Tests

- ✧ Download and unzip Jasmine standalone into a directory
  - https://github.com/jasmine/jasmine/releases
- ✧ Erase everything in src and spec directories
- ✧ Place your application code into the src directory
- ✧ Place test code or a spec into the spec directory
- ✧ Update SpecRunner.html
  - Replace references to erased src and spec files with yours
- ✧ Start Browser-sync (or whatever you use for local server) and go to the http://…/SpecRunner.html to have your specs run

Writing a Spec (test)

```
describe("My Function", function() {
   var initValue;

   beforeEach(function() {
      initValue = "someVal";
   });

   it("should not return true", function() {
      var result = someFunc(initValue);
      expect(result).not.toBe(true);
   });
});
```

Summary

- ✧ Unit testing is an essential process in any software dev
  - Planning for unit testing changes how you write code (for the better)
- ✧ Mocks are used to fake the dependencies of target code
- ✧ describe(“message”, function () {}) is to used to group tests together
- ✧ beforeEach(function() {}) is used to initialize state before running each test
- ✧ it(“message”, function() { actual test }) is used to run the actual test code