GFT

JavaScript Testing

GFT INTERNAL TRAINING

INNOVATE. TRANSFORM. DELIVER.

Unit testing

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

Good Unit Test Principles FIRST

- **F**ast
- **I**ndependent
- **R**epeatable
- **S**elf-validating
- **T**imely

http://agileinaflash.blogspot.com.es/2009/02/first.html

Unit testing benefits

- bug-free code ready for deliver
- enable safe refactoring
- error origin is easier to spot
- problems are found early in the development cycle
- document the code
- code to be tested is written with better practices

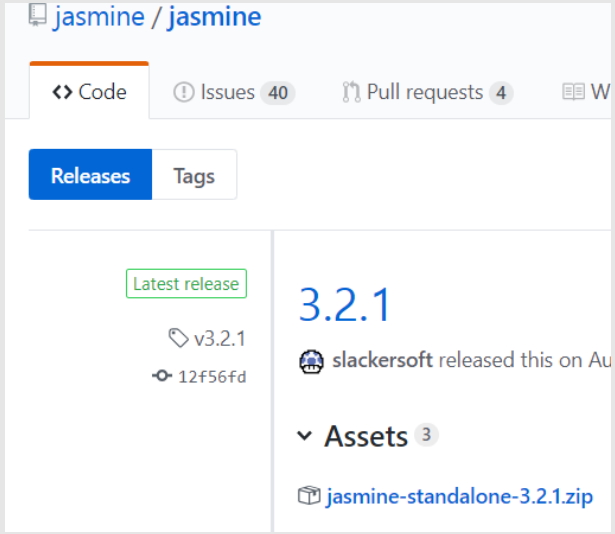
http://www.qa-systems.com/fileadmin/user_upload/resources/White_Papers/Why_Bother_to_Unit_Test.pdf

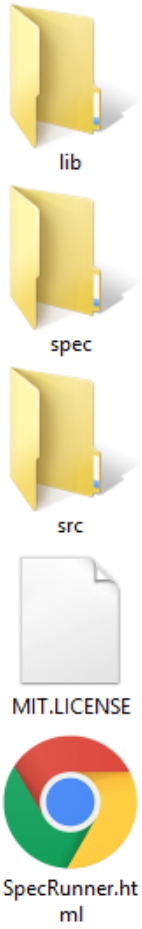
Testing JS with Jasmine

Jasmine

Behavior-Driven Javascript

https://jasmine.github.io/

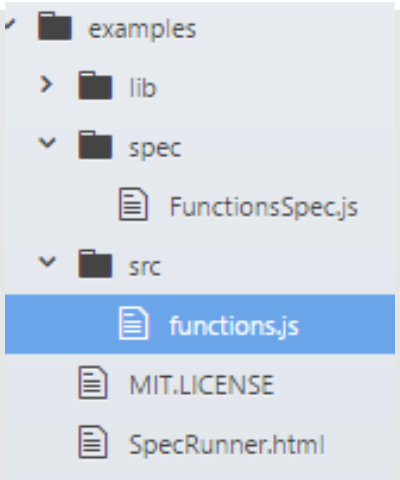




- jasmine code
- specification (tests)
- source (code to test)

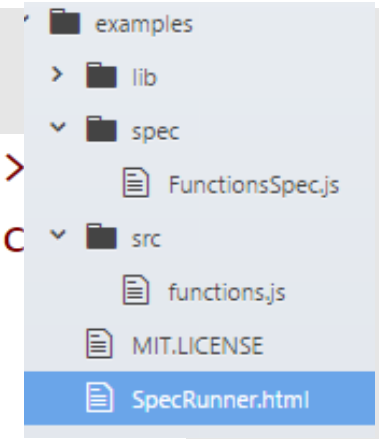
Code to Test

```
function isEven(number) {
  return !(number % 2);
}
function isMultipleOf7(number){
  return !(numero % 7);
}
function isOdd(number) {
  //does not work right
  return !(number % 2);
}
```



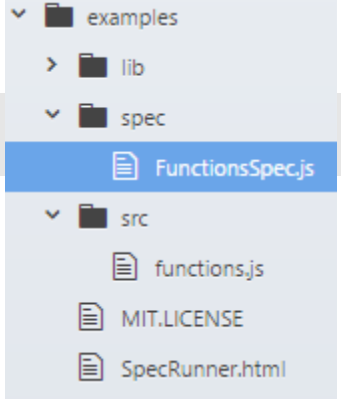
Test runner

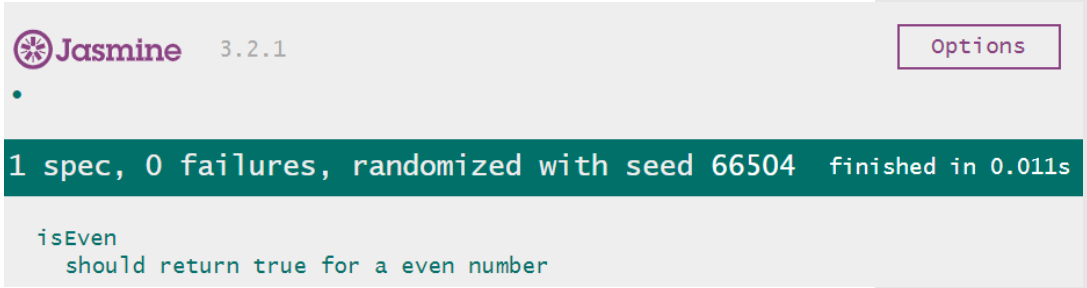
```
<script src="lib/jasmine-3.2.1/jasmine.js"></script>
<script src="lib/jasmine-3.2.1/jasmine-html.js"></sc
<script src="lib/jasmine-3.2.1/boot.js"></script>

<!-- include source files here... -->
<script src="src/functions.js"></script>

<!-- include spec files here... -->
<script src="spec/FunctionsSpec.js"></script>
```



Test example

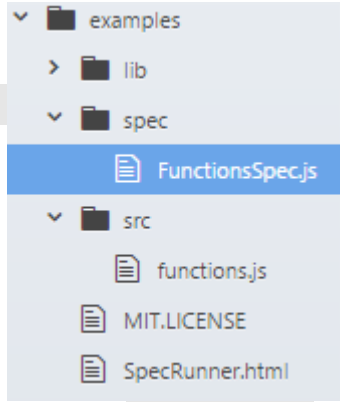
```
describe("isEven", function() {
  var evenNumber, oddNumber;
  //executes before all test
  beforeEach(function() {
    evenNumber = 6;
    oddNumber = 7;
  });
  //test
  it("should return true for a even number", function() {
    expect(isEven(evenNumber)).toBe(true);
  });
});
```





Test example

```
//test
it("should return true for a even number", function() {
  expect(isEven(evenNumber)).toEqual(true);
});
//test
it("should not return true for an odd number", function() {
  expect(isEven(oddNumber)).not.toEqual(true);
});
//test
it("should return a boolean", function() {
  expect(typeof(isEven(evenNumber))).toEqual("boolean");
});
```




```
3 specs, 0 failures, randomized wit

  isEven
    should return a boolean
    should return true for a even number
    should not return true for an odd number
```

In projects

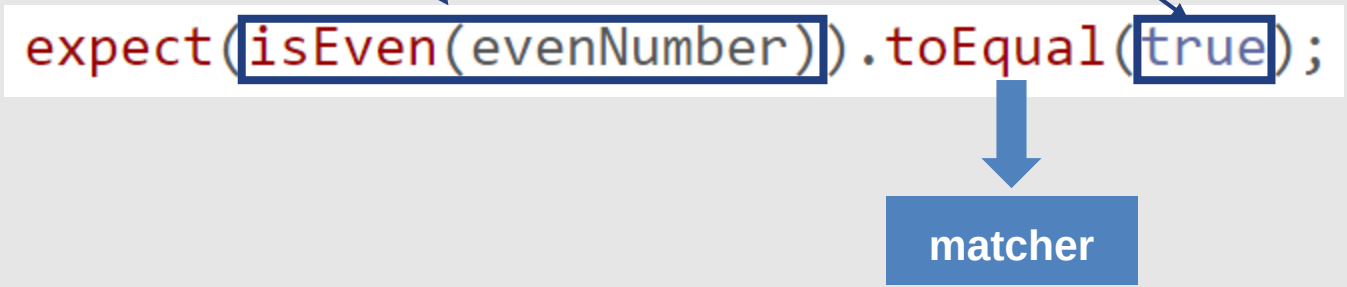
- SpecRunner.html is not needed
- Developers or testers write specs (tests)
- Tools used for automated test
  - karma → test runner
  - phantomJS → headless testing (without GUI)
  - webpack → module bundler

```
//Spec Suite or Test Suite -->
//  where the behavior is defined with tests
describe("title (what is being tested)", function() {
  //may be some code
  //test or spec
  it("title of the test or spec", function() {
     //test code
  });
  //may be more tests
});
```

describe can be nested

Expectations

- are built with the function **expect** which takes **the actual value**
- the expect function is chained with a **matcher** function, which takes **the expected value**



Matchers

- matcher implements a boolean comparison between actual value and expected value
- reports to Jasmine the result and Jasmine will then pass or fail the spec
- any matcher can evaluate to a negative assertion by chaining the call to expect with a not before the matcher

```
expect(isEven(oddNumber)).not.toEqual(true);
```

Matchers included in Jasmine

- toEqual() → ==
- toBe() → ===
- toMatch()
- toBeNull()
- toBeTruthy()
- .....

https://jasmine.github.io/api/3.2/matchers.html

Custom Matchers

There is also the ability to write custom matchers for when a project's domain calls for specific assertions that are not included in Jasmine

https://jasmine.github.io/tutorials/custom_matcher

JS Testing frameworks

- **Jasmine** (oldest, default in angular)

- **Jest** (inspired by Jasmine, great performance, made by facebook)

https://jestjs.io/

- **Chai** (assertion library, usually used with Mocha)

https://www.chaijs.com/

- **Mocha** (test framework, usually used with Chai)

https://mochajs.org/

Learning resources

- Clean code JS

  https://github.com/ryanmcdermott/clean-code-javascript

- Jasmine official documentation

  https://jasmine.github.io/pages/docs_home.html

- Overview of JS testing tools

https://medium.com/welldone-software/an-overview-of-javascript-testing-in-2018-f68950900bc3

Shaping the future of digital business

GFT Internal Technical Training

Jordi Alemany

jordi.alemany@gft.com

+34 935 639474

Eduardo García Ibaseta

eduardo.garcia-ibaseta@gft.com
+34 935 639476

GFT IT Consulting, S.L.
Av. Alcalde Barnils, 69-71
08174 Sant Cugat del Vallès (BARCELONA)