**Testing With Jasmine**

**Testing is writing code to test your own code!**

**Benefits of testing:**

* Different cases
* Different edge cases
* Manually testing software is tedious and error prone
  + In a small program we an manually check all pieces of functionality before pushing our code
  + As programs get larger, it becomes impossible to catch everything that might break
* Act of writing tests is a good way to clarify the expectations of a function
  + We have to specify what the legal input/output should be
* Reading tests is a great way to understand a new code base
* Testing is a ore skill for professional developers

**Testing Tools**

**Jasmine:**

* Each language has multiple testing libraries (e.g) python, JS, Ruby)
* Jasmine is a popular javascript testing framework
* Jest (another popular JavaScript Testing library) is commonly used with Node and React and provides multiple layers on top of jasmine
* Once you learn how to test in one frame =work, you can apply the same knowledge to others

**Testing process:**

* Testing allows us to automate the tedious work of making sure the new lines of code don’t break pre-existing code
* In order to work with jasmine in the browser we need to import the library using a script tag

**Naming convention:**

* Make the test file the same name as original document with .test.js

**Unit Testing**

* It is a function that accepts two arguments
* The first argument is the name used to refer to the test
* The second argument is a callback function that executes the test (function from non .test js file)
* It (“should do ‘x’ ”, call back function () {

Expect(someValue).someMatcher(…);

Expect(functionfromOGFile(value).toEqual(expectedOutput);

});

**Matchers**

**see jasmine\_testing test taxes and index**

* **toEqual** will compare values of objects / arrays, doesn’t have to be array to array or object to object
* **.toEqual(obj)** checks that the value is the same
* **.toBe(obj)** is the same object (different lists with same values doesn’t match)
* **.toContain** does object/array contain this item?
* **.not** add before matcher to invert
* Full list of matchers: www.jasmine.github.io/apo/edge/matchers.html
* .toEqual(obj) deep equality - does not care about reference, it cares about values. Arrays with the same value will pass, even if they are not the same array
* .toBe() is ===

return […new Set(values)] 🡪 gives back an array

the Set(values) will return an object with the values and adding the array brackets and … will spread the object and output as an array

**Describe Tests**

* describe (“user form tests”, function () {

it (“should submit a user on submitUser()” function () {  
submit user

expect(users.length).toEqual(1);

});

});

Chartjs.org

Insights.stackoverflow.com/survey/2019

Similar to jasmine:

Mochajs.org is very similar to

jestjs.io

cypress.io

cucumberjs.io

protractortest.org

chaijs.com

avajs

nightwatchjs.org

nightmarejs.org