TDD with Mocha & Chai

Module 1 Week 3

Notes Repo: https://github.com/C-Shi/lhl-flex-lecture

Learning Objectives

Automate Testing

JavaScript Modules

Test Runner and Assertion Library

Mocha and Chai

Why Test Driven Development (TDD)

Manual Testing

Common Practice

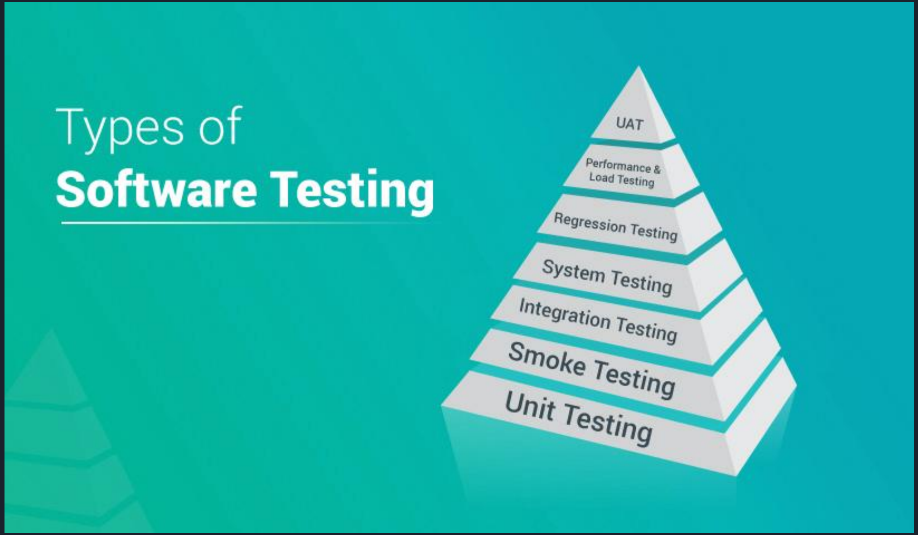
1. Use console.log() to print out value
2. Debug Tool / Breakpoint
3. Manual Browser Test

Challenge

1. Repetitive work
2. Human error

Automated Testing

1. Automated Testing is using a software (programming code) to test against another piece of software (programming code)
2. Automate a human-driven manual process of validating a software
3. Improve code quality
4. SAVE YOUR TIME



Node.js Module

1. Modules are blocks of encapsulated code that communicates with an external application
2. module is an object that is globally available
3. Developer can require or export a module

```
// log module in any node.js file
console.log(module);

// require a module
const fs = require('fs');

// export a module
module.exports = {} // it can be anything
```

Why we need to know module for testing?

Automated Testing tool

Assertion Library

1. Is the tool to verify that things are correct
2. Help simplify your testing logic | NO IF AND IF AND ANOTHER IF ….
3. NodeJS assert library, Chai

```
function sum(x, y) {
  return x + y;
}

// using assertion library
assert.equal(sum(1, 2), 3);

// without assertion library
const result = sum(1, 2);
if(result !== 3) {
  throw new Error(`Expected ${result} to be equal to 3`);
}
```

Automated Testing tool

Test Runner

1. Is the execution unit that allows you to run the test
2. Create an isolate environment for your test to run against a specific chunk of code
3. Allows all tests to be run in a batch process
4. Mocha, Karma, Jasmine, etc

```
describe('test calculator library, () => {
 it('should return 2 if calling add with 1 and 1', () => {
   // test 1 + 1 = 2
 });

 it('should return 0 if calling subtract with 1 and 1', () => {
   // test 1 - 1 = 0
 })
});
```

Introduce Mocha

1. Help organize and execute test code
2. Install package with *npm i mocha* or *npm i mocha —save-dev*
3. Look for documentation at https://mochajs.org/
4. Add test command to package.json

```
// describe is used to target a specific feature you want to test against
describe('test sum function', function() {
  // it is used to specify a specific test case
  it('should provide sum of 6 given [1, 2, 3]', function() {

  })

  it('should not provide sum of 6 given [1, 2]', function() {

  })
})
```

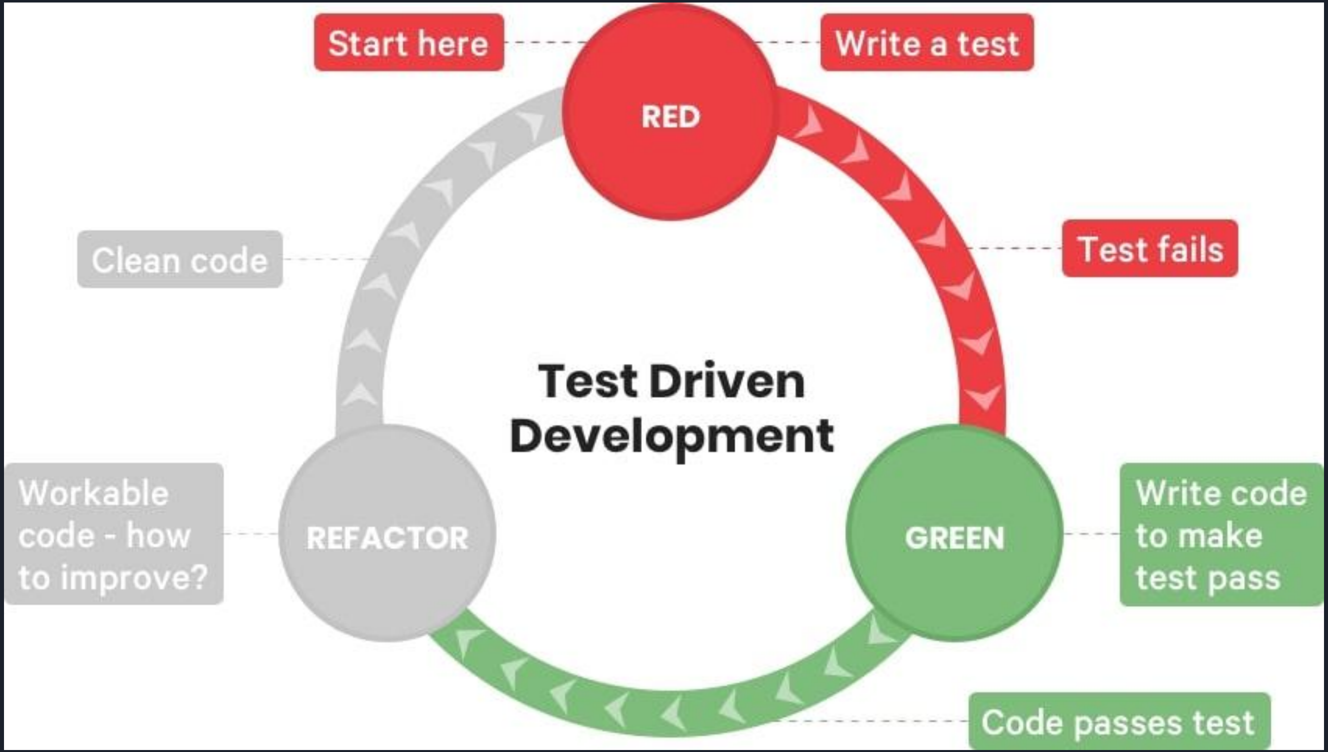
Introduce Chai

1. More powerful, human readable assertion library compared to native assert module
2. Provide chainable method for easy testing
3. Install package with *npm i chai* or *npm i chai –save-dev*
4. Look for documentation at https://www.chaijs.com/ or cheatsheet at https://devhints.io/chai

```
// assert group
assert.isTrue(300 > 100);
assert.isArray('abc'.split(''));

// expect group
expect('Hello World')
  .to.be.a('string')
  .to.include('Hello')
```

Test Driven Development


Start here
Write a test
RED
Test fails
Clean code
Test Driven Development
Workable code - how to improve?
REFACTOR
GREEN
Write code to make test pass
Code passes test