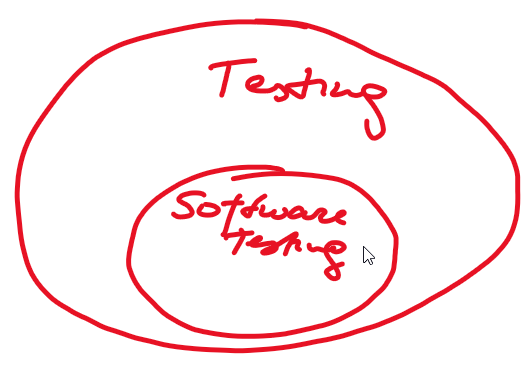
# Testing

## What is testing?

Process to improve quality of software by finding bugs or possible improvements. Ref: <https://en.wikipedia.org/wiki/Software_testing> . At the core of all testing is the comparison. When we test we compare the actual software behavior / code / other artifacts (log files, data produced, etc) with expected behavior / code / etc.



When working on a new checkout page the design put the “enter coupon code” input field at very visible location. Then it was noticed that there are many abandoned shopping carts after the new checkout page went live. How was this conclusion - by comparing the old shopping cart to the new one (comparing the data). A/B testing showed liked the new shopping cart itselft (design, usability, etc). After some more experimentation with the design and more data gathering it was concluded that this was due to the “enter coupon code” input field being very visible and people abandoning shopping carts while looking for the coupons. This was all based on “comparison”.

There are many conceptions of testing, one of them is described by James Bach: <https://www.youtube.com/watch?v=ILkT_HV9DVU>

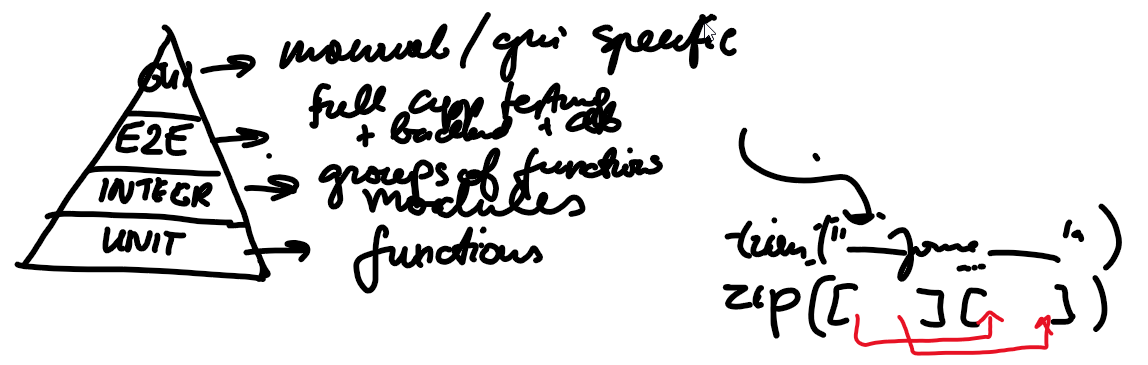
What is testing: <https://docs.google.com/presentation/d/1YjQJYdSzfFYjHxL6qC0peaIfoYZbAFMJJZs3V8eQAAk/edit?usp=sharing>

Test levels and test types: <https://docs.google.com/presentation/d/1YUALQpytOOb5pI8MurR3l-BwksBwxci8MLZXtfJ_-mw/edit?usp=sharing>

Unit tests are almost always written by developers. Integration tests can be wirten by both testers and developers. System tests - usually testers in companies, but for personal projects developers can also do that.

## Test levels

Ref: <https://docs.google.com/presentation/d/1YUALQpytOOb5pI8MurR3l-BwksBwxci8MLZXtfJ_-mw/edit?usp=sharing>

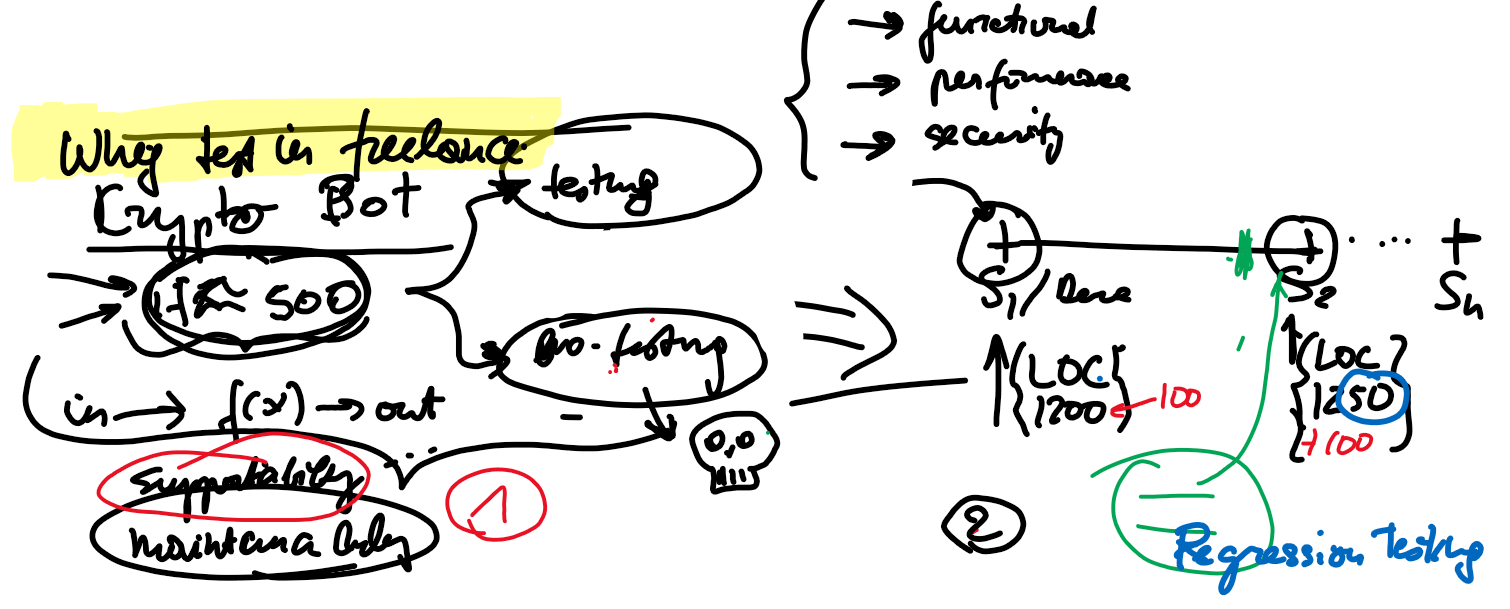


## Why do we test code?

Ref: <https://slides.javascript.en.sdacademy.pro/slides/Testing/testing.html#/>

Ref: <http://homeworks.javascript.en.sdacademy.pro/>

One interesting question: why should we test as freelancers - maintainability (more for the owner of the projects / the client) and regressions before demo.



# What’s important?

JS has many testing frameworks, which should you choose. This is dictated by the framework you use. React → Jest, React Testing Framework, Cypress. Angular → Karma + Jasmin (low level testing - functions / unit / components), Protractor / Cypress (e2e / gui / “as client would use it”).

# Karma

Exported to Angular doc: <https://docs.google.com/document/d/1i-crirEt4mAHWoAVVk6Xo4TnFo4pOuSCkBm5KI3xS_0/edit#heading=h.3dre8tcg1lvx>

# Jest

It has good documentation: <https://jestjs.io/docs/getting-started>

Ref: <https://slides.javascript.en.sdacademy.pro/slides/Testing/jest.html#/>

Ref: <https://www.youtube.com/watch?v=NHMIn723hQY> (starts from snapshot testing)

# Mocha

* A test framework. Commonly uses chai as an assertion library.
* Also has good documentation: <https://mochajs.org/>
* Ref: <https://semaphoreci.com/community/tutorials/getting-started-with-node-js-and-mocha>
* Ref: <https://slides.javascript.en.sdacademy.pro/slides/Testing/mocha.html#/>
* See this screenshot to setup a simple project:
  + *npm init*
  + *config:*

{

"name": "mochademo",

"version": "1.0.0",

"description": "",

"type": "module",

"main": "index.js",

"scripts": {

"test": "mocha || true"

},

"author": "",

"license": "ISC",

"devDependencies": {

"mocha": "^9.0.2"

}

}

* + *npm install mocha*
  + *add tests:*

*// add.spec.js*

import { equal } from "assert";

import { add } from "../app/add.js";

*// ... test the export*

*// console.log(add(1, 2));*

**describe**("Testing add", () => {

**it**("should add numbers correctly", () => {

**equal**(**add**(2, 3), 5);

});

**it**("should add numbers correctly", () => {

**equal**(**add**(-2, -3), -6);

});

});

*// test.js*

import assert from "assert";

**describe**("Array", function () {

**describe**("#indexOf()", function () {

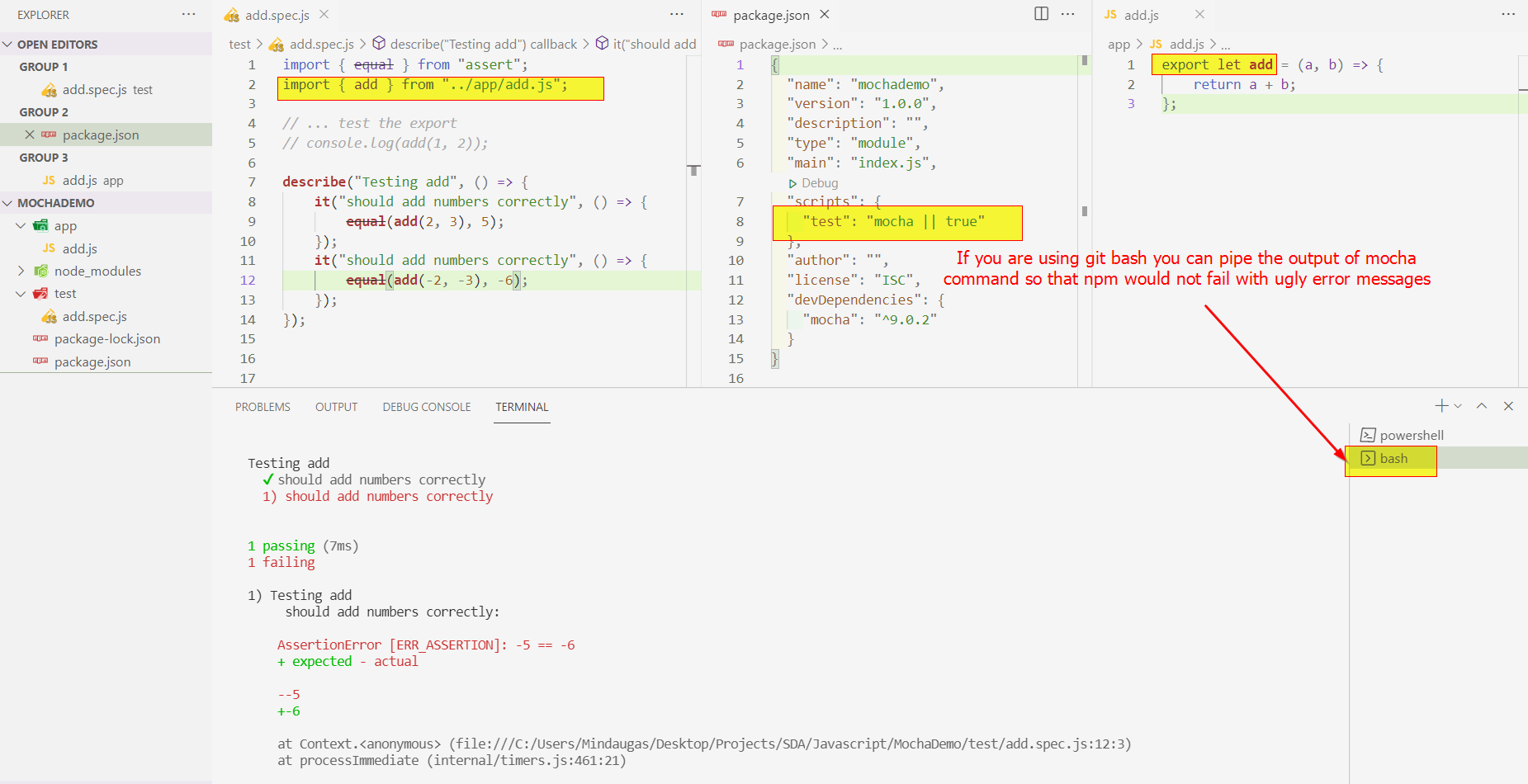
**it**("should return -1 when the value is not present", function () {

**assert**.**equal**([1, 2, 3].**indexOf**(4), -1);

});

});

});



# Cypress

Ref: <https://docs.google.com/presentation/d/1mU-RUA85me_3KVDKW4dEWqrl2TXiJiRrTZyQ6TSj7lQ/edit?usp=sharing>

Simple test

**describe**('My First Test', () => {

**it**('Hello world!', () => {

**expect**(true).to.**equal**(true)

})

})

More complex test:

**describe**('Test home page', () => {

**it**('Visits homepage successfully', () => {

*// given*

let url = 'http://127.0.0.1:5500/app'

*// when / then*

cy.**visit**(url)

})

**it**('Homepage contains the greeting message', () => {

*// given*

let url = 'http://127.0.0.1:5500/app'

*// when*

cy.**visit**(url)

*// then*

cy.**contains**('Hello world!')

})

**it**('Enters the name and presses the button, receiving the hello message', () => {

*// given*

let url = 'http://127.0.0.1:5500/app'

*// when*

cy.**visit**(url)

cy.**get**('#name\_input').**type**('Mindaugai')

cy.**get**('button').**click**()

*// then*

cy.**contains**('Labas Mindaugai!')

})

})

For a more complex app:

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8" />

<meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

<title>Document</title>

</head>

<body>

<p>Hello world!</p>

<br />

<br />

<input *type*="text" *id*="name\_input" />

<button>Click me!</button>

<br />

<div>

<div>A</div>

</div>

</body>

<script>

let input = document.**getElementById**("name\_input");

let button = document.**getElementsByTagName**("button")[0];

button.**addEventListener**("click", () => {

let value = document.**createTextNode**("Labas " + input.value + "!");

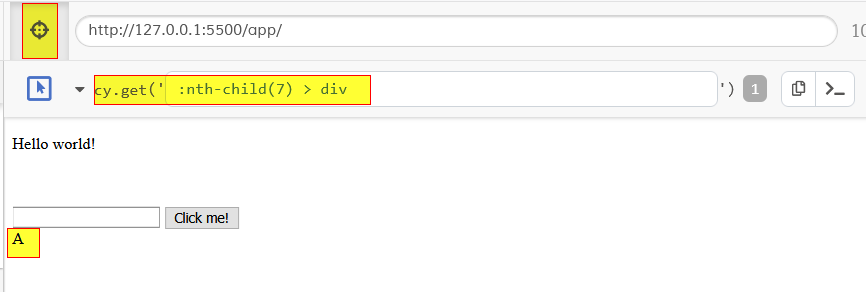
document.body.**appendChild**(value);

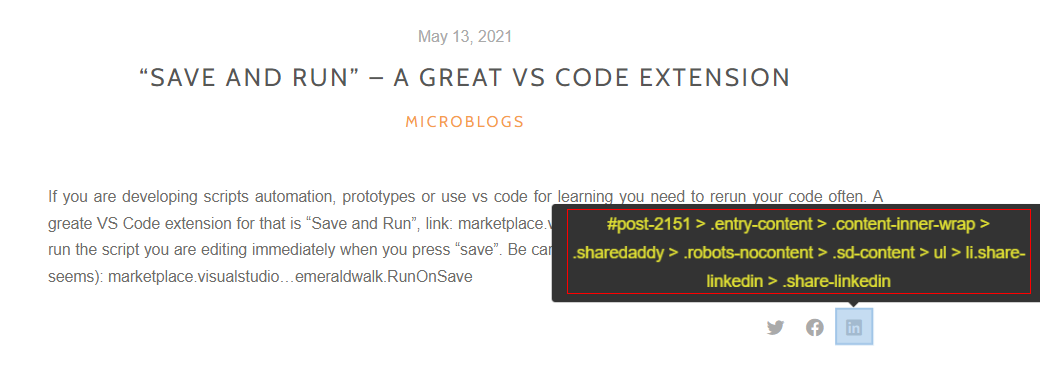
});

</script>

</html>

Selector playground:





Ref: <https://slides.javascript.en.sdacademy.pro/slides/Testing/cypress.html#/>

Ref: <https://docs.google.com/presentation/d/1mU-RUA85me_3KVDKW4dEWqrl2TXiJiRrTZyQ6TSj7lQ/edit?usp=sharing>