1. from command line

create a 'testing-demo' directory and cd into it (NO SPACES)

2. open vs code from the testing-demo folder

3. create two files

functions.js

functions.test.js

4. from terminal

npm init -y

5. npm i --save-dev jest (only installs the library in the dev environment)

6. in package.json

"scripts: {"test": "jest"}

YOU WILL NEED TO DELETE THE DEFAULT CONTENT OF THE "scripts" property

7. inside functions.test.js

test("should return I am {age}", () => {

expect().toEqual()

})

8. at the top of the files

const { stateAge } = require('./functions')

9. edit the test

expect(stateAge(37)).toEqual("I am 37.")

10. inside of functions.js

module.exports = {

stateAge: (age) => {

return `I am ${age}`

}

}

11. in terminal

npm run test

12. write a new test (YOU DONT NORMALLY WANT TO CHAIN YOUR TESTS INTO ONE TEST, INSTEAD YOULL WANT TO USE A DESCRIBE BLOCK FOR THAT PURPOSE)

test("Should return user created status", () => {

expect(addUser('Joey', 21)).toEqual('User created')

expect(addUser('Chandler', 15)).toEqual('User not old enough')

})

13. run the test again

14. const { stateAge, addUser } = require('./functions')

15. inside of functions.js create a new property

module.exports = {

stateAge: (age) => {

return `I am ${age}`

},

addUser: (name, age) => {

let user = {name, age}

if (user.age >= 21) {

return 'User created'

} else {

return 'User not old enough'

}

}

}

16. back to the test files MODIFY EXISTING TEST

test("Should return user object or status", () => {

expect(addUser('Joey', 21)).toEqual({

name: 'Joey',

age: 21

})

expect(addUser('Chandler', 15)).toEqual('User not old enough')

})

17. modify the actual funciton to return the object

18. test("Should add a new color to the colors array and return the new array", ()=>{

expect(addColor('Pink').toContain('pink')

})

19. const { stateAge, addUser, addColor } = require('./functions')

20.

let colors = ['Red', 'Green', 'Blue']

module.exports = {

stateAge: (age) => {

return `I am ${age}`

},

addUser: (name, age) => {

let user = {name, age}

if (user.age >= 21) {

return 'User created'

} else {

return 'User not old enough'

}

},

addColor: (color) {

colors.push(color)

}

}

21. TEST IT, IT SHOULD FAIL

22. MODIFY FUNCTION TO HAVE

Colors.push(color.toLowerCase())