

- 
- <https://www.linkedin.com/in/namratabafna/>



---

Tests and their importance

---

Types of tests

---

Test using Jest

---

Setting up Jest

---

Jest vs mocha

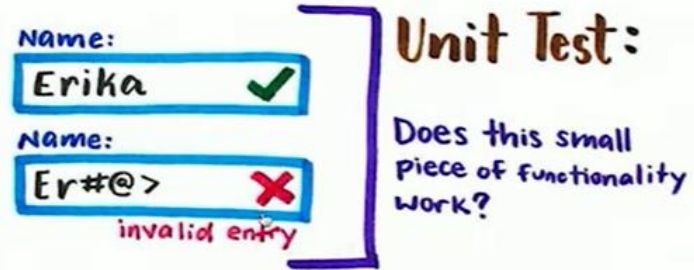
---

Challenge

---

Solution

# What is a Unit Test?



[twitter.com/Una/status/850451564527591424](https://twitter.com/Una/status/850451564527591424)

# What is an Integration Test?

The diagram illustrates an integration test for a component. It features a large purple bracket on the right side that groups the elements on the left. The elements include two UI components, two test assertions, and a definition of an integration test.

**UI Components:**

- A button labeled "Select droplet" with a blue checkmark icon.
- A dropdown menu labeled "Select droplet" with a blue arrow icon. The dropdown is open, showing two options: "droplet-1" and "droplet-2". The "droplet-2" option is highlighted with an orange border.

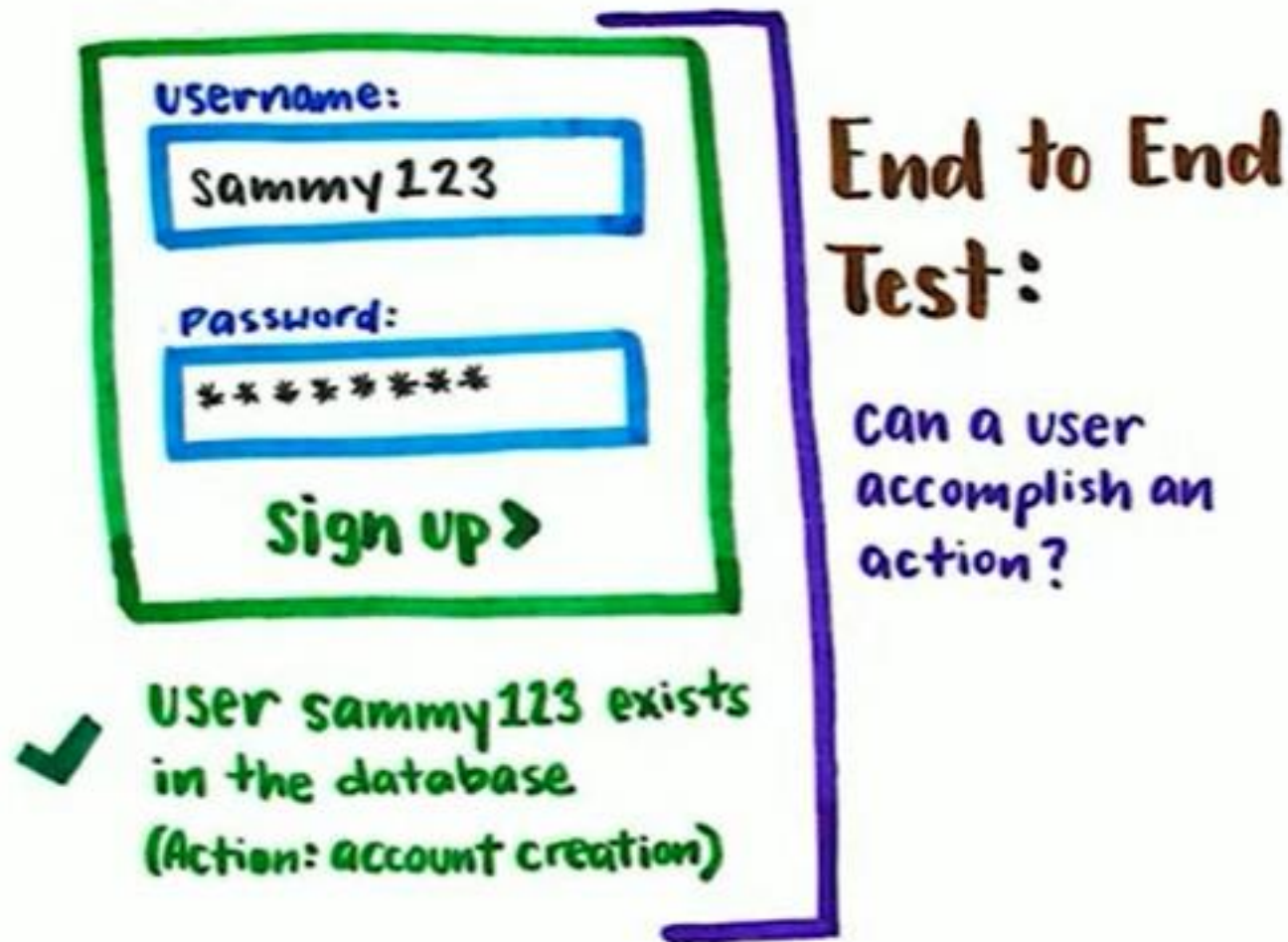
**Test Assertions:**

- ✓ Droplet list populates in dropdown
- ✓ Response shows droplet-2 details when selected

**Integration Test:**

Does the component integrate into the system?

# What is an End to End Test?



# Where do we focus our time?





# Watch feature

```
• it works

expect(received).toBe(expected)

Expected value to be (using ===):
  3
Received:
  1

    at Object.<anonymous>.test (__tests__/sum.js:4:18)

✗ it works (6ms)

Test Suites: 1 failed, 1 total
Tests:       1 failed, 1 total
Snapshots:   0 total
Time:        0.44s, estimated 1s
Ran all test suites.

Watch Usage
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press q to quit watch mode.
> Press Enter to trigger a test run.
```

▼ setup-jest-math

> \_\_tests\_\_

> coverage

> node\_modules

package.json

subtract.js

sum.js

yarn.lock

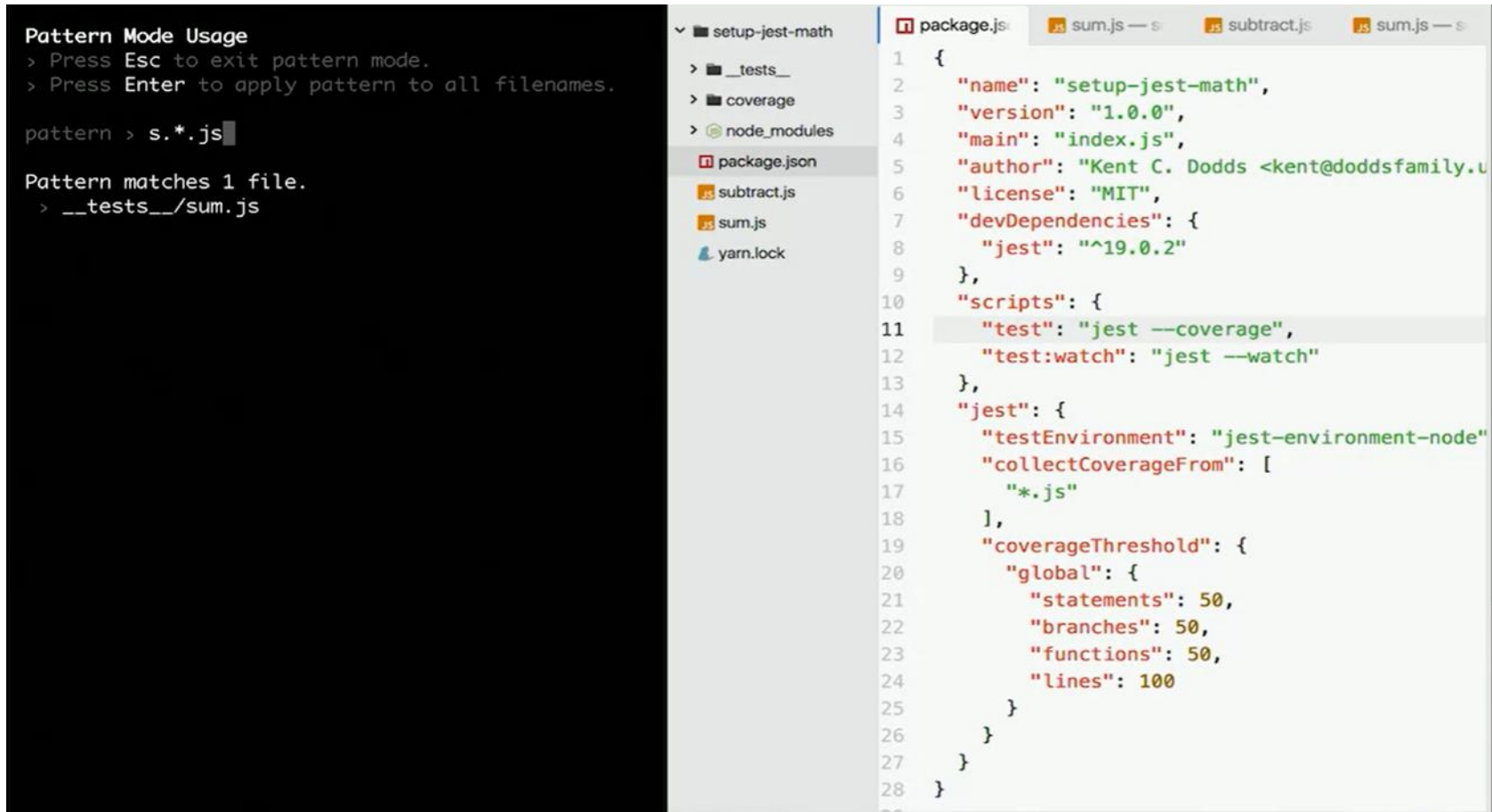
```
1 {
2   "name": "setup-jest-math",
3   "version": "1.0.0",
4   "main": "index.js",
5   "author": "Kent C. Dodds <kent@doddsfamily.u",
6   "license": "MIT",
7   "devDependencies": {
8     "jest": "^19.0.2"
9   },
10  "scripts": {
11    "test": "jest --coverage",
12    "test:watch": "jest --watch"
13  },
14  "jest": {
15    "testEnvironment": "jest-environment-node"
16    "collectCoverageFrom": [
17      "*.js"
18    ],
19    "coverageThreshold": {
20      "global": {
21        "statements": 50,
22        "branches": 50,
23        "functions": 50
```

# Pattern

```
Pattern Mode Usage
> Press Esc to exit pattern mode.
> Press Enter to apply pattern to all filenames.

pattern > s.*.js

Pattern matches 1 file.
> __tests__/sum.js
```

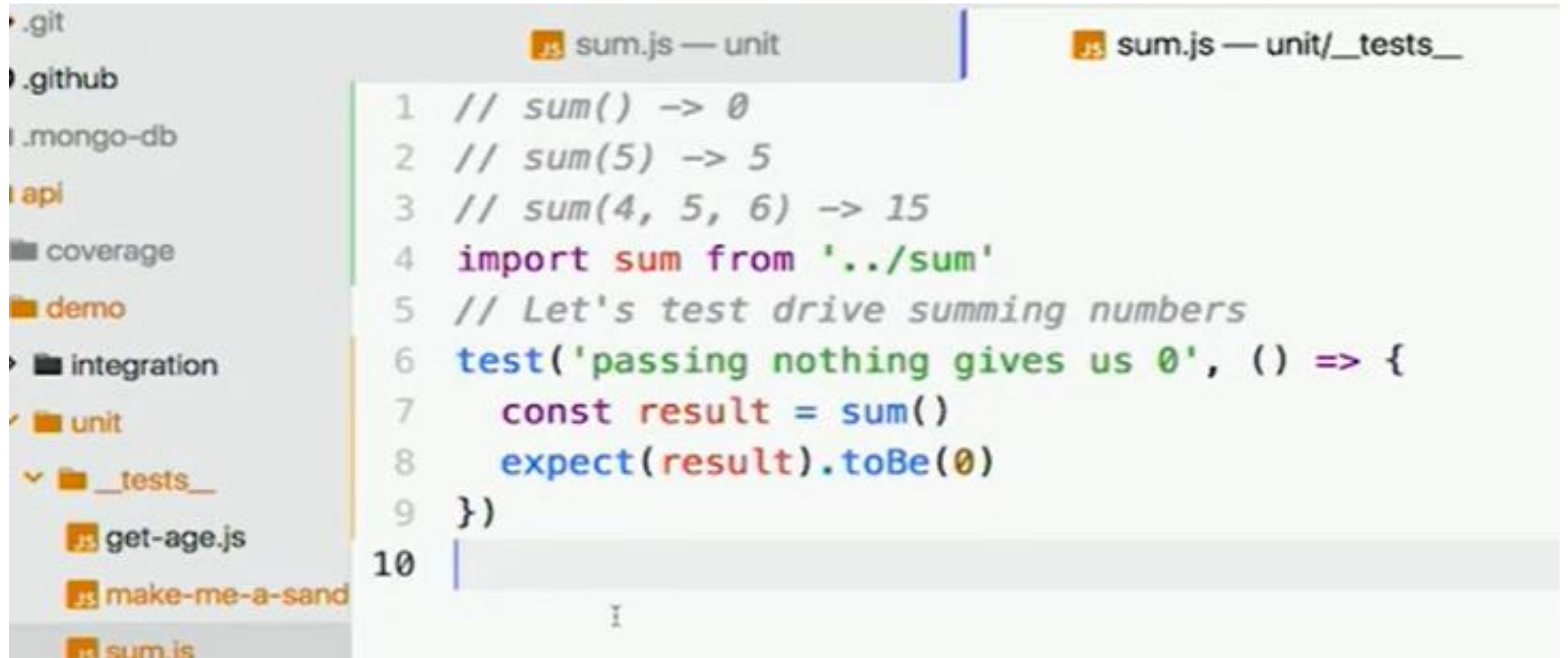


The screenshot shows a code editor interface. On the left is a file explorer for a project named 'setup-jest-math'. It contains a folder '\_\_tests\_\_' and files 'coverage', 'node\_modules', 'package.json', 'subtract.js', 'sum.js', and 'yarn.lock'. The 'package.json' file is selected. On the right is a terminal window showing the 'Pattern Mode Usage' instructions. The main editor area displays the contents of 'package.json', which is a JSON object with fields for 'name', 'version', 'main', 'author', 'license', 'devDependencies', 'scripts', and 'jest'. The 'scripts' field contains 'test' and 'test:watch'. The 'jest' field contains 'testEnvironment', 'collectCoverageFrom', 'coverageThreshold', and 'global'.

```
1 {
2   "name": "setup-jest-math",
3   "version": "1.0.0",
4   "main": "index.js",
5   "author": "Kent C. Dodds <kent@doddsfamily.com>",
6   "license": "MIT",
7   "devDependencies": {
8     "jest": "^19.0.2"
9   },
10  "scripts": {
11    "test": "jest --coverage",
12    "test:watch": "jest --watch"
13  },
14  "jest": {
15    "testEnvironment": "jest-environment-node",
16    "collectCoverageFrom": [
17      "*.js"
18    ],
19    "coverageThreshold": {
20      "global": {
21        "statements": 50,
22        "branches": 50,
23        "functions": 50,
24        "lines": 100
25      }
26    }
27  }
28 }
```



# TDD



The screenshot shows a code editor with a file explorer on the left and two open files. The file explorer lists a project structure including .git, .github, .mongo-db, api, coverage, demo, integration, unit, and \_\_tests\_\_. The unit folder is expanded, showing \_\_tests\_\_, get-age.js, make-me-a-sand, and sum.js. The active file is 'sum.js — unit', which contains the following code:

```
1 // sum() -> 0
2 // sum(5) -> 5
3 // sum(4, 5, 6) -> 15
4 import sum from '../sum'
5 // Let's test drive summing numbers
6 test('passing nothing gives us 0', () => {
7   const result = sum()
8   expect(result).toBe(0)
9 })
```

The cursor is at line 10. A second tab, 'sum.js — unit/\_\_tests\_\_', is also open but empty.

> .github

> .mongo-db

▼ api

> coverage

▼ demo

> integration

▼ unit

▼ \_\_tests\_\_

get-age.js

make-me-a-sand

sum.js

get-age.js

jest.config.json

make-me-a-sandw

sum.js

> dist

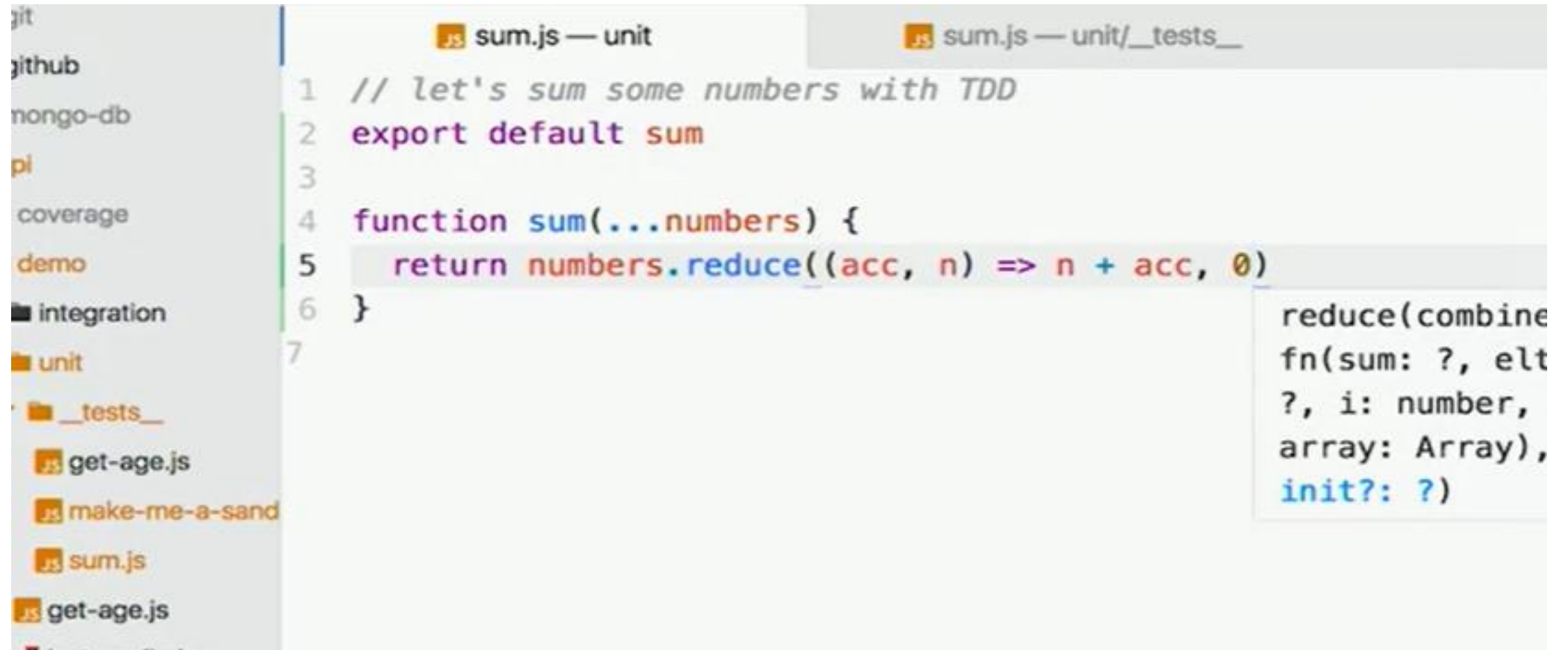
> node\_modules

> other

> scripts

```
1 // sum() -> 0
2 // sum(5) -> 5
3 // sum(4, 5, 6) -> 15
4 import sum from '../sum'
5 // Let's test drive summing numbers
6 test('passing nothing gives us 0', () => {
7   const result = sum()
8   expect(result).toBe(0)
9 })
10
11 test('passing a single number returns that number', () => {
12   const result = sum(5)
13   expect(result).toBe(5)
14 })
15
16 test('passing any number of numbers returns their sum', () => {
17   const result = sum(4, 5, 6)
18   expect(result).toBe(15)
19 })
20
```

# Code wrt TDD



```
sum.js — unit
1 // let's sum some numbers with TDD
2 export default sum
3
4 function sum(...numbers) {
5   return numbers.reduce((acc, n) => n + acc, 0)
6 }
7
```

reduce(combine  
fn(sum: ?, elt  
?, i: number,  
array: Array),  
init?: ?)