Next Journey in the Javascript

1 What we covered in the courses

- Javascript essentials: Variable, Primitive data types, operators
- Array and Objects
- Functions: basic functions, recursive functions, nested functions, Anonymous functions
- Classes: constructor, methods, inheritance, prototype
- Asynchronous programming: Callback, Promise, async/await
- DOM Event Handling
- DOM events: keyboard events, mouse events, drag and drop events, form events
- Web APIs: DOM manipulation, fetch API, timer API

2 What you should continue in your JS developer journey

Goal: develop large scale applications using Javascript

Learn more about Javascript ES6 features

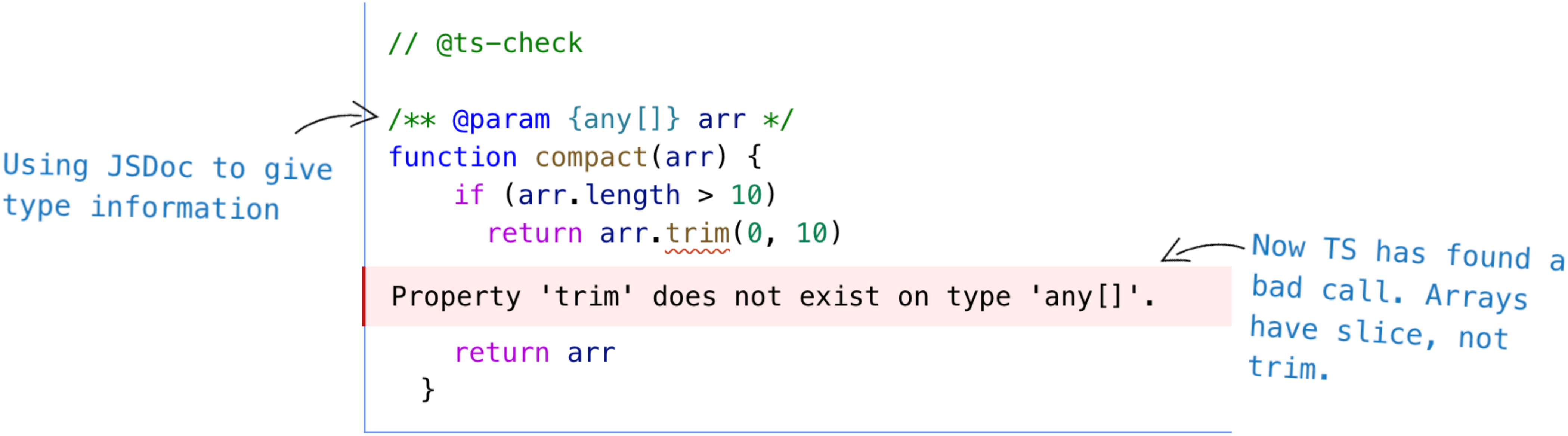
- Modules feature: decompose your code into modules
- Iterator and Generator: improve performance and reduce memory usage for large collections.
- JS Standard Library: Set and Map, JSON related API, URL API, Fetch API, etc.
- … and more

JS Frameworks and Libraries

- Angular (Google) : a platform for building mobile and desktop web applications
- Vue: a progressive framework for building user interfaces
- React (Facebook) : a library for building user interfaces

Typed Javascript

- TypeScript: a superset of JavaScript that adds optional static typing
- Check your code in the editor and compile it to JavaScript



Using JSDoc to give type information

```
// @ts-check

/** @param {any[]} arr */
function compact(arr) {
    if (arr.length > 10)
      return arr.trim(0, 10)
Property 'trim' does not exist on type 'any[]'.
    return arr
  }
```

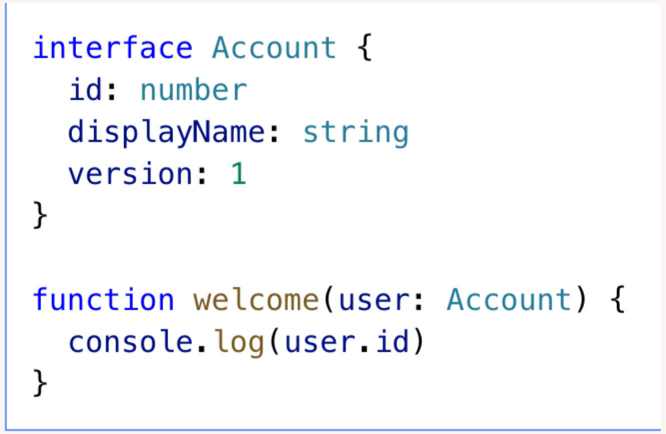
Now TS has found a bad call. Arrays have slice, not trim.

JavaScript with JSDoc

Describe Your Data

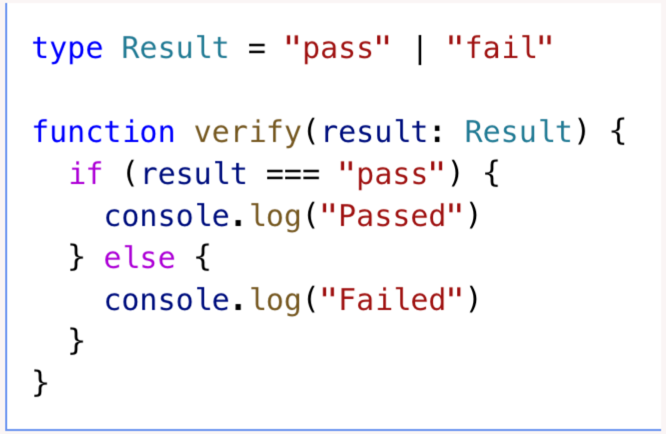
Describe the shape of objects and functions in your code.

Making it possible to see **documentation and issues in your editor**.



```
interface Account {
  id: number
  displayName: string
  version: 1
}

function welcome(user: Account) {
  console.log(user.id)
}
```

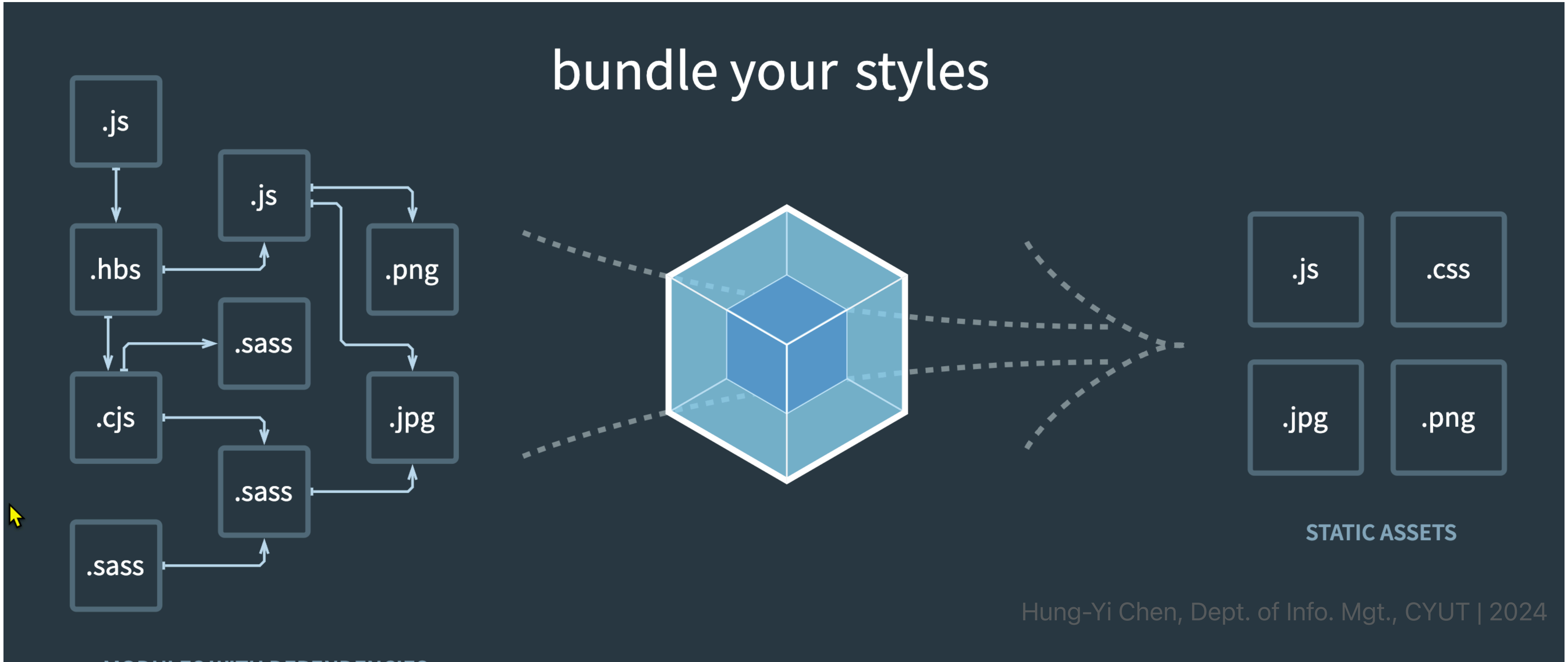


```
type Result = "pass" | "fail"

function verify(result: Result) {
  if (result === "pass") {
    console.log("Passed")
  } else {
    console.log("Failed")
  }
}
```

Code Bundling

- Bundling the small modules into a single file for production
- Tools such as Webpack, Rollup, Parcel

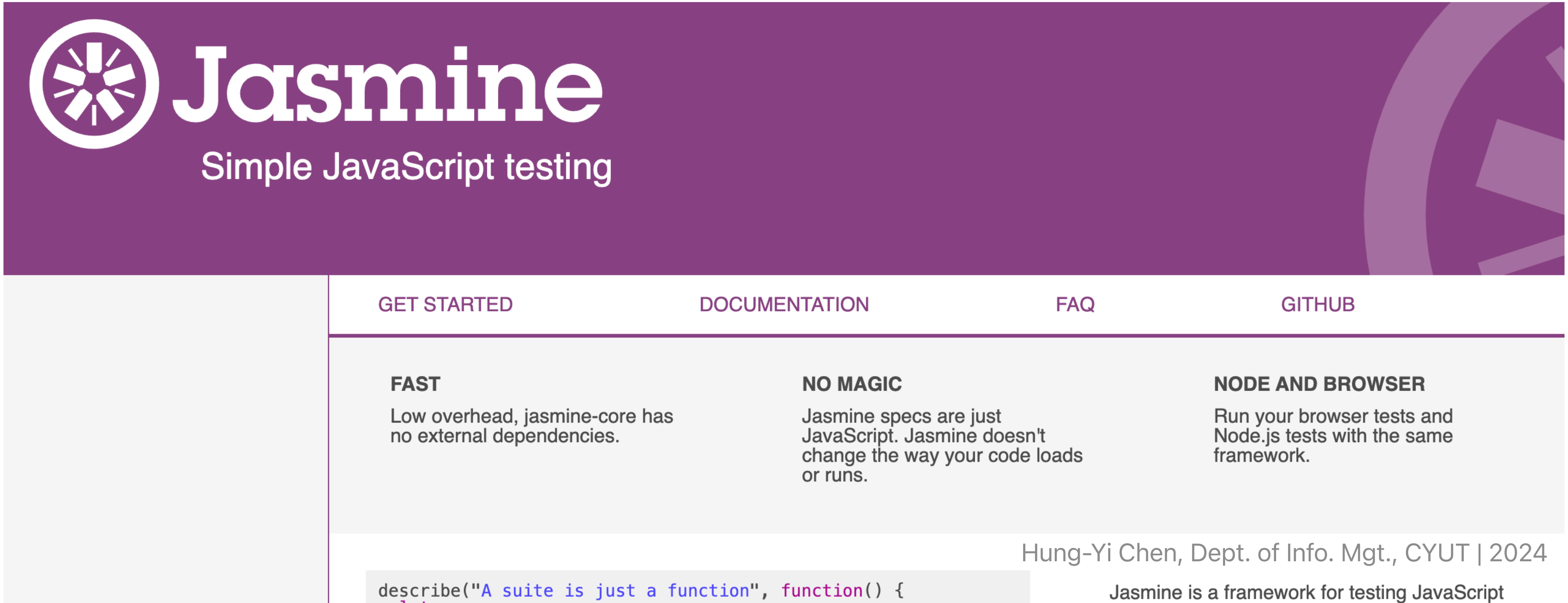


Package Management

- NPM: Node Package Manager
- Yarn: a faster and more secure package manager

Unit Testing

- Test your code to ensure it works as expected so that you can integrate it into your application with confidence
- Tools: Jest, Mocha, Jasmine



Jasmine

Simple JavaScript testing

FAST

Low overhead, jasmine-core has no external dependencies.

NO MAGIC

Jasmine specs are just JavaScript. Jasmine doesn't change the way your code loads or runs.

NODE AND BROWSER

Run your browser tests and Node.js tests with the same framework.

```
describe("A suite is just a function", function() {
```

Jasmine is a framework for testing JavaScript

Server-side Javascript

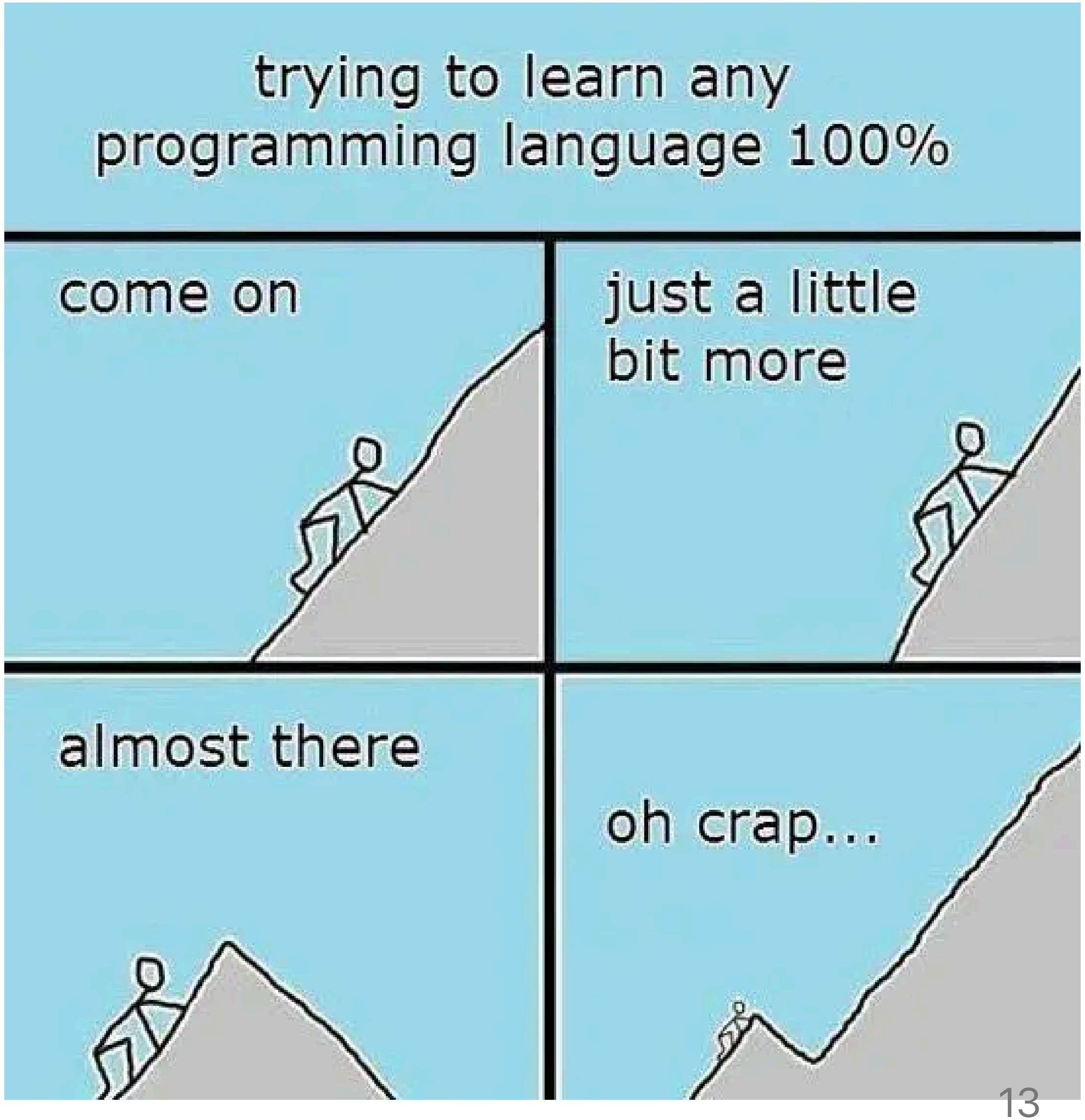
- Node.js: a JavaScript runtime built on Chrome's V8 JavaScript engine
- Express: a fast, unopinionated, minimalist web framework for Node.js

Version Control

- Git: a distributed version control system
- GitHub: a web-based Git repository hosting service

3 Keep learning and coding!

Learning and Master a programming language is not a sprint, it's a marathon.



4 Good luck and have fun!

God bless you! 🙏

"Hold on to instruction, do not let it go; guard it well, for it is your life." Proverbs 4:13 (NIV)