Node.js: Getting Started

INTRODUCTION

Samer Buna
SOFTWARE ARCHITECT - AGILELABS
@samerbuna | samer.dev

This Course

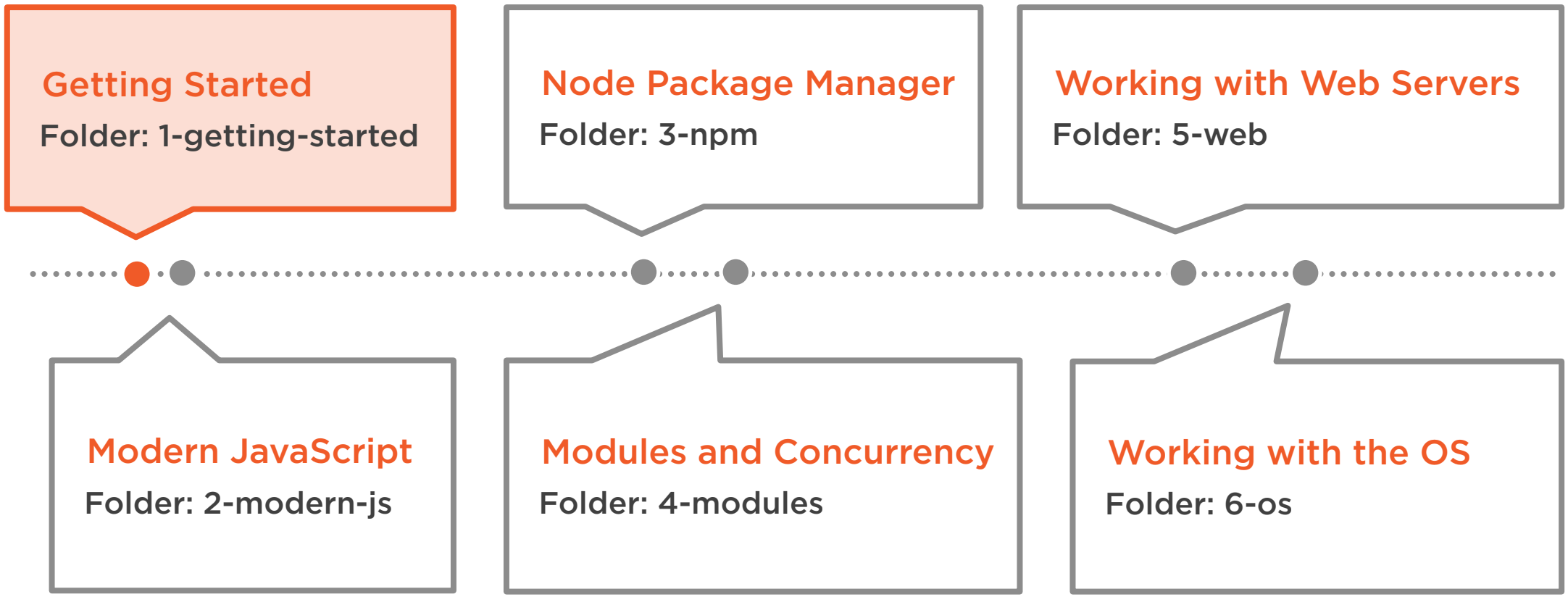
Beginner level

No assumptions about Node knowledge

Some assumptions about

- Basic programming concepts
- JavaScript

This Course


Getting Started
Folder: 1-getting-started
Node Package Manager
Folder: 3-npm
Working with Web Servers
Folder: 5-web
Modern JavaScript
Folder: 2-modern-js
Modules and Concurrency
Folder: 4-modules
Working with the OS
Folder: 6-os

Not Covered

Addons

Buffers

Streams

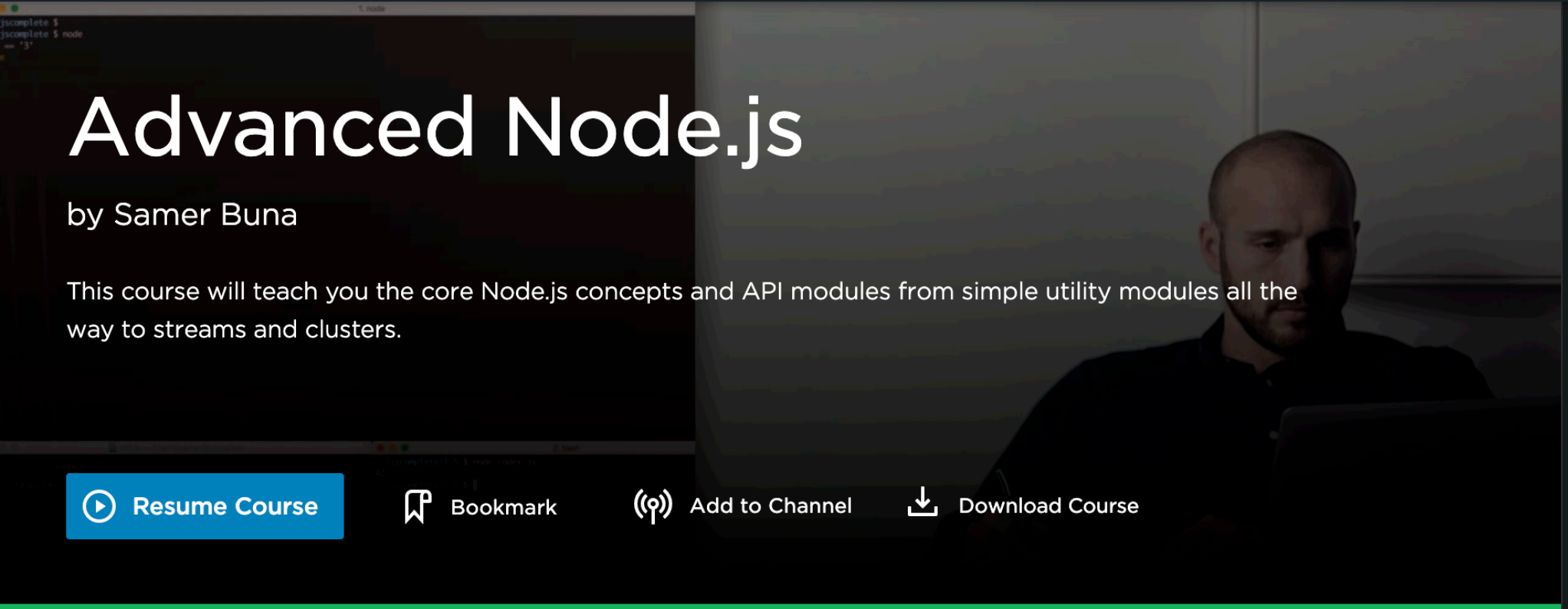
Modules

- crypto
- zlib
- dns
- net/dgram
- ...

Advanced Node.js

by Samer Buna

This course will teach you the core Node.js concepts and API modules from simple utility modules all the way to streams and clusters.



Resume Course | Bookmark | Add to Channel | Download Course

Table of contents | Description | Transcript | Exercise files | Discussion | Learning Check | Related Courses

Expand All

Course author

Samer Buna

Samer Buna is a polyglot coder with years of practical experience in designing, implementing, and testing software, including web and mobile applications development, API design, functional...

Course info

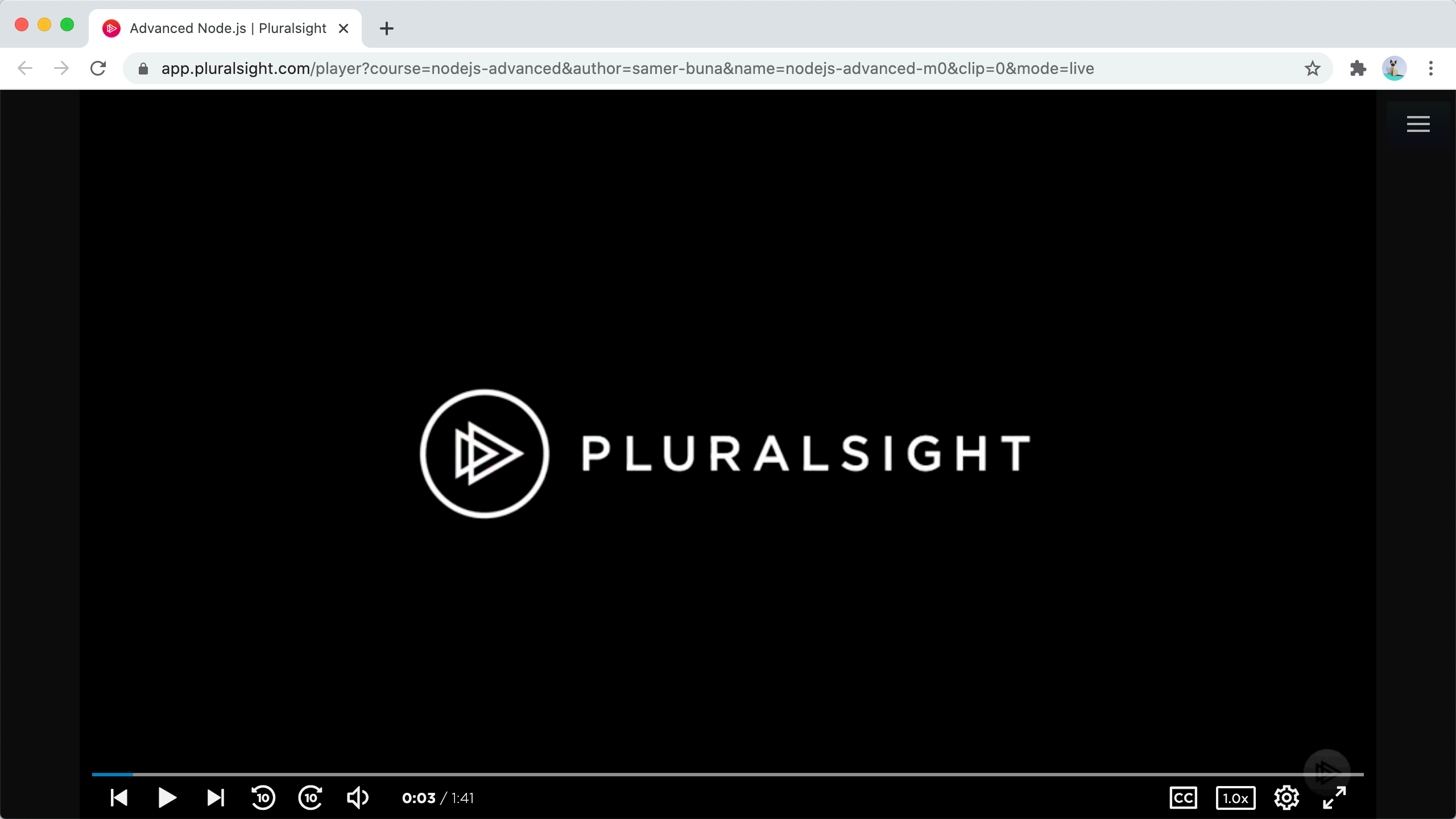
Level	Advanced
Rating	★★★★★ (345)
My rating	★★★★★
Duration	3h 45m

samer.dev/adv-nodejs

There will be problems

Mac

Windows


Advanced Node.js | Pluralsight
app.pluralsight.com/player?course=nodejs-advanced&author=samer-buna&name=nodejs-advanced-m0&clip=0&mode=live
PLURALSIGHT
0:03 / 1:41
CC
1.0x

What is Node.js?

Node.js

JavaScript on your backend servers...

~~Node.js~~ VM (V8/Chakra)

JavaScript on your backend servers...

Node.js

A wrapper around a VM like V8

Node.js

A wrapper around V8 with built-in modules providing rich features through easy-to-use asynchronous APIs

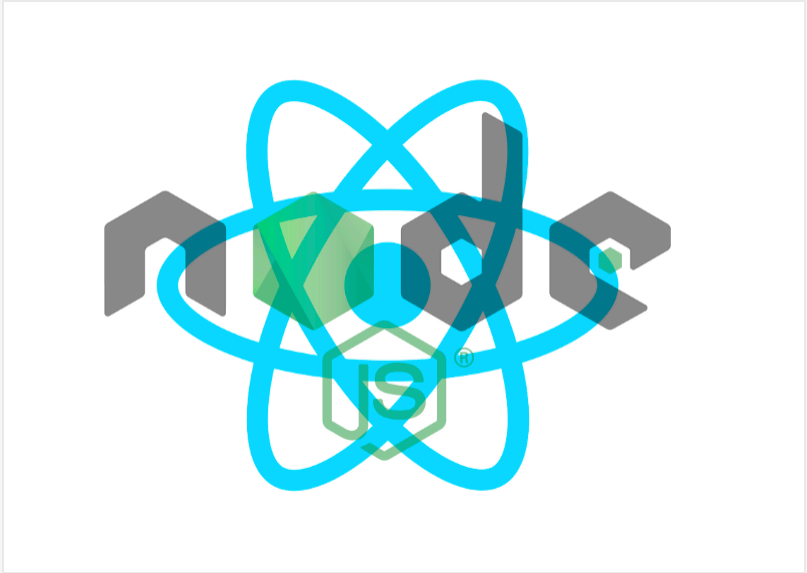
Why Node?

Pros

- Wrapper around V8 (execute JavaScript)
- Built-in modules (fs, http, crypto, zip, ...)
- Asynchronous APIs (no threads)
- C++ addons
- Debugger and other utilities
- **NPM**
- **Module dependency manager**

Why React Developers LOVE Node

Samer Buna
Updated September 2020 . 4 min read



Some people mistakenly assume that Node is required in order to use React. **It is not.** You don't need Node to run a React project. You don't even need a browser.

Full-stack!
One language everywhere

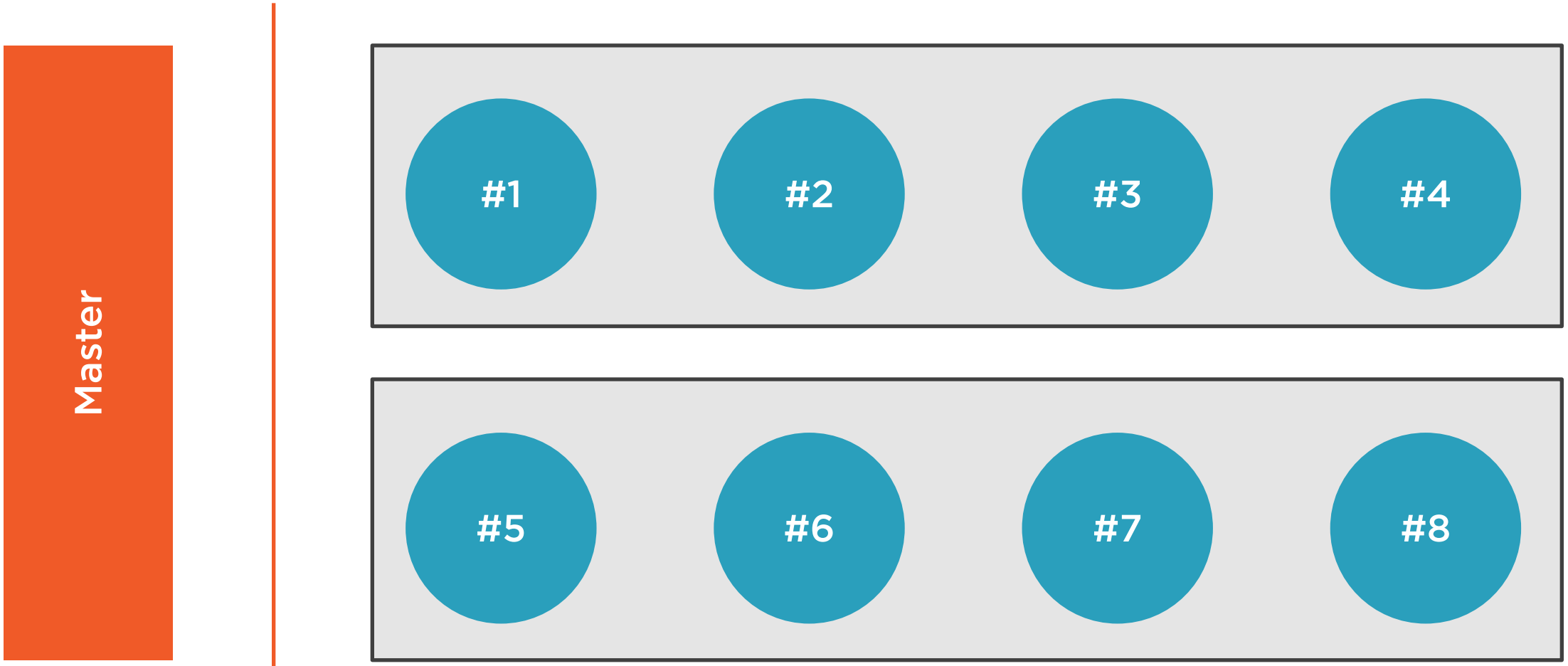
Cons

Different way of thinking

Picking good options

Small packages

Why Node is named Node



What exactly is a CALLBACK?

Callback

A function that Node will “call back” at a later point in the time of a program

A callback is just a function

```
function cb(data) {
  // do something with data
}

someAsyncMethod(cb);
```

Asynchronous Programming as Seen at Starbucks

Samer Buna Follow
Mar 21, 2016 · 3 min read ★

Make Me a Latte, Starbucks!

```
function Samer(readyLatte) {
  // drink readyLatte..
}

starbucks.makeMeALatte(Samer);
```

I promise you a chick

```
const egg = chicken.makeChick(); // It's a promise!


egg
  .then(chick => raiseChick()) // Success outcome
  .catch(badEgg => throw badEgg) // Fail outcome
```

Hard Coding Concepts Explained with Simple Real-life Analogies

How to explain coding concepts like streams, promises, linting, and declarative programming to a 5-year-old

Samer Buna Follow
Nov 8, 2017 · 15 min read

I love thinking about coding concepts by comparing them to familiar things we know in life. There are so many analogies out there about coding concepts. Some of them are good while others are confusing, mainly because they focus on partial aspects of a concept while ignoring many others. This article will summarize some of the analogies that I think best fit a few coding concepts in complete ways.

What You Get When You Install Node

MODULE