

Node.js: Getting Started

INTRODUCTION



Samer Buna

CHIEF ENGINEER AT AGILELABS.COM

@samerbuna | samerbuna.com



What is Node.js?



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

Table of contents

Description

Transcript

Exercise files

Discussion

Learning Check

Recommended

This course is part of: Node.js Path

Expand all

Course Overview

✓ 1m 41s ▾

Node != JavaScript

✓ 1h 2m 24s ▾

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 ★★★★ (232)

My rating ★★★★

Duration 3h 45m

Released 16 Feb 2017

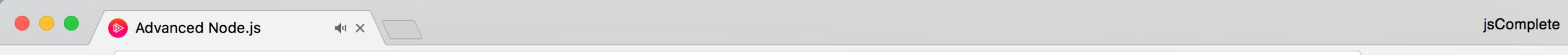
Share course

There will be problems

Mac

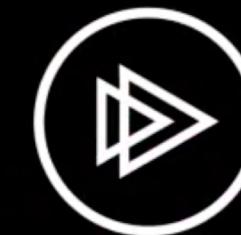
Windows





Advanced Node.js

Course Overview : Course Overview



PLURALSIGHT

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?



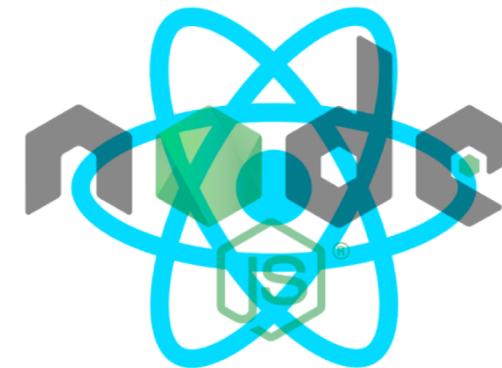
Pros



- 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



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.

React gives you a language to **describe a user interface (UI)**. That interface could be the controls in your car, a modern fridge screen, or your microwave buttons. Basically, it could be any **interface** where a **user** needs to read and interact with an intelligent device.

React was initially made for the web user interface and it is most popular there. This is why we have the **ReactDOM** library that works with a browser's DOM. However, React today is also popular for non-Web interfaces like iOS, Android, virtual reality, and even command line interfaces.

Full-stack!
One language everywhere



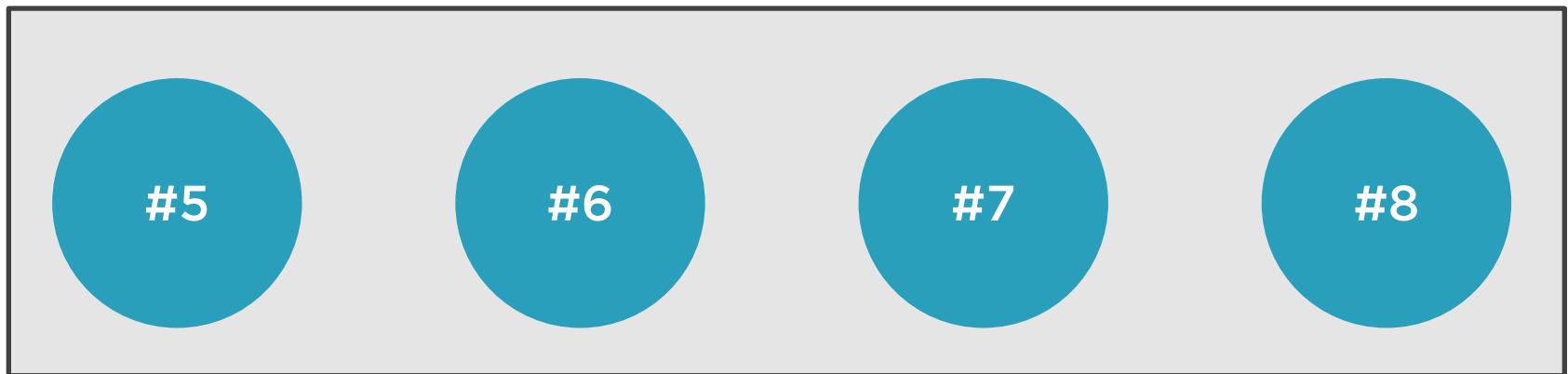
Cons



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);
```





jsComplete

[Follow](#)[Sign in](#)[HOME](#)[EDGECODERS.COM](#)[JSCOMPLETE.COM](#)

★ Member preview



Samer Buna

[Follow](#)

Author for Pluralsight, O'Reilly, Manning, and LinkedIn Learning. Curator of jsComplete.com

Mar 21, 2016 · 3 min read

Asynchronous Programming as Seen at Starbucks



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
```





Follow



Sign in

HOME

DEV

LEARN TO CODE FOR FREE



Samer Buna

Follow

Author for Pluralsight, O'Reilly, Manning, and LinkedIn Learning. Curator of jsComplete.com

Nov 8, 2017 · 15 min read

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

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

