Intro to Node

It's just not that mysterious. ¯_(ツ)_/¯

tl;dr

- Node is an interpreter that uses JavaScript as its language
- It can do tons of tasks, including serving things over http
- To use node, you ...

1. Install it from nodejs.org
2. Write a text file in JavaScript
3. Then "node theScriptFile.js"

Disclaimer

- Node takes more than an hour to learn!
- Our goal is for an understanding of node and how it runs JavaScript
- Please sign up for our Node course!

Node is a script interpreter

- Just like ...
- bash/ksh runs Unix shell commands
- .bat files runs DOS commands
- Submitting JCL on a mainframe
- perl runs perl commands
- node runs JavaScript commands

History

- In 2009, Ryan Dahl downloaded Google Chrome's source code and extracted the JavaScript interpreter known as V8
- He made it run from the command line
- He added capabilities that will never be in a browser but you might need in a scripting language
- Knowing the current directory (__dirname), Reading from command line, Reading/writing files and streams
- He removed things only a browser would care about...
- DOM interfaces (b/c there is no DOM)

Why node?

- It can do anything you'd want a scripting language to do ...
- Automate SysAdmin tasks
- ...
- But it is so much faster and well-designed that you can ...
- Web server
- Print server
- ...

How is it better designed?

- It is not multi-threaded. Killing old processes and spooling up new processes takes a lot of CPU and memory.
- JavaScript uses an event loop which continuously runs on a single thread.
- Much more on this later (time permitting)
- Instead of spawning threads, all processes define a callback function which will be put on the main thread when it runs out of things to do.
- If a process is super cpu-intensive, this isn't the best architecture. But if it involves waiting on things (like file system or threads or listeners) it rocks!
- It is unopinionated so you can use it with any number of libraries
- Usually installed via npm
- Usually included with require('libraryName');

How is it worse?

- Since everything runs on the main thread, if it throws, the whole system crashes. Use try/catches!
- If you use blocking I/O calls, the thread is taken up by that process and nothing else can run. (Don't do that!)
- Some devs hate callback hell. (Learn and use promises and async/await).

How to use node

1. Install it
2. Write a script
3. Execute that script

1. To install node

- Go to NodeJS.org
- Click on the big green button.
- Follow the prompts.

- After that, most utilities and libraries will be installed via npm.
- npm is installed with node

2. Write a script

- Use any text editor to create JavaScript commands and save them to a file

greet.js

```
let name = argv[1];
console.log("Hello " + name);
```

webserver.js

```
let express = require('express');
let app = express();

let homeDir = "/webroot";
app.use(express.static(homeDir));

app.listen(5000, () => {
  console.log("Ready on port 5000");
});
```

3. Run it

```
$ node greet.js Sanjay
Hello Sanjay
$ node greet.js Olivia
Hello Olivia
$ node server.js
Ready on port 5000
```

- Node doesn't have to be any more complex than that.

tl;dr

- Node is an interpreter that uses JavaScript as its language
- It can do tons of tasks, including serving things over http
- To use node, you ...

1. Install it from nodejs.org
2. Write a text file in JavaScript
3. Then "node theScriptFile.js"