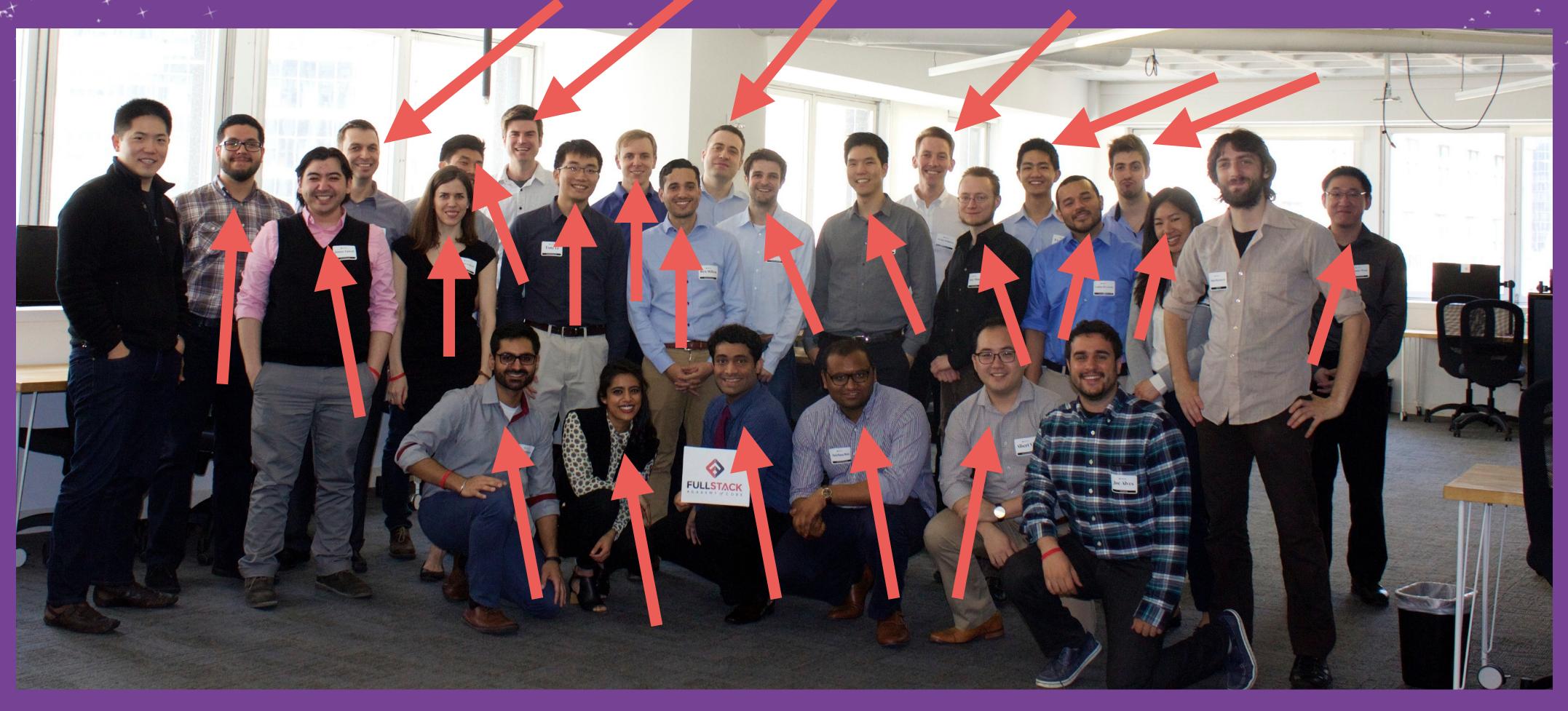
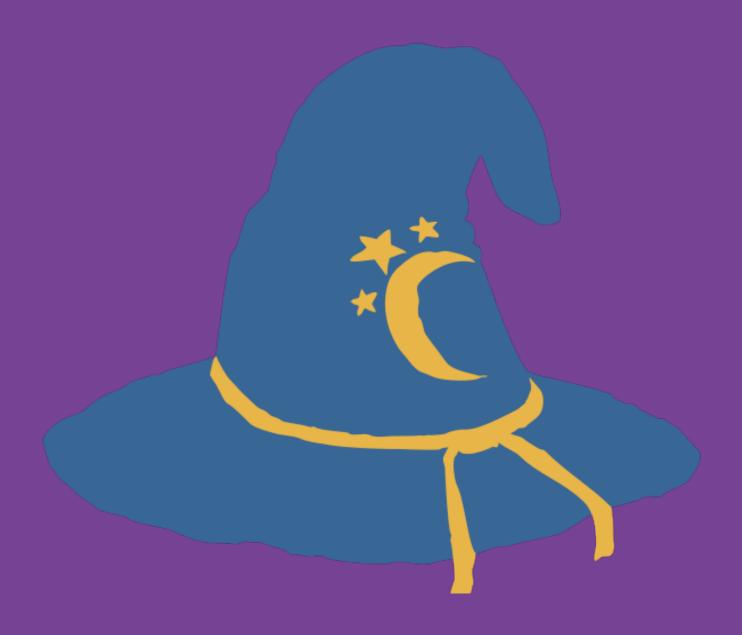
Hello.











Browserify

"I built Browserify because I wanted to keep writing node"

- James Halliday (@substack)



Allows for the use of the CommonJS module system in the browser

 Allows for the use of the CommonJS module system in the browser

· Shims node core libraries for the browser*

· Allows for the use of the CommonJS module system in the browser

· Shims node core libraries for the browser*

 Allows for node_modules to be used in the browser*

Getting Started

CLI: npm install -g browserify

Or

your favorite build process tool

node core modules

- require('url') work sanely with url paths!
- · require('streams') .pipe in the browser!
- require('crypto') hashing and encryption!
- · require('events') event emitters!
- new Buffer(0) Ulnt8Arrays and ArrayBuffer!

```
fs.readfie()
```

```
require('os').platform()
===
```

'browser'

http.createServer()

npm modules

transforms



**** *****



1.3.14 instantaneous-browserification (2015-02-24)

Features

• CommonJS: - angular modules are now packaged for npm with helpful exports

npm install angular

where angular and browserify clash

Two module systems

where angular and browserify clash

Two module systems

Passing application value around is weird

where angular and browserify clash

- Two module systems
- Passing application value around is weird
- Many requires are used to "run" the file, rather than work with an export

Synchronous (technically pre-processed)

- · Synchronous (technically pre-processed)
- Your development process is your build process

- · Synchronous (technically pre-processed)
- Your development process is your build process
- · Reusable server-side and client-side code

- · Synchronous (technically pre-processed)
- Your development process is your build process
- · Reusable server-side and client-side code

Ease of transformations

- · Synchronous (technically pre-processed)
- Your development process is your build process
- · Reusable server-side and client-side code
- Ease of transformations

Transferrable to future exploration of technologies

Thanks.