The New York
Code + Design
Academy

INTRODUCTION TO

node JS®

Agenda

- Quick Intro
- Node.js: The Beginning
- What Is Node.js?
- Why Use Node.js?
- Installing Node.js

`Node.js is a platform for building applications`

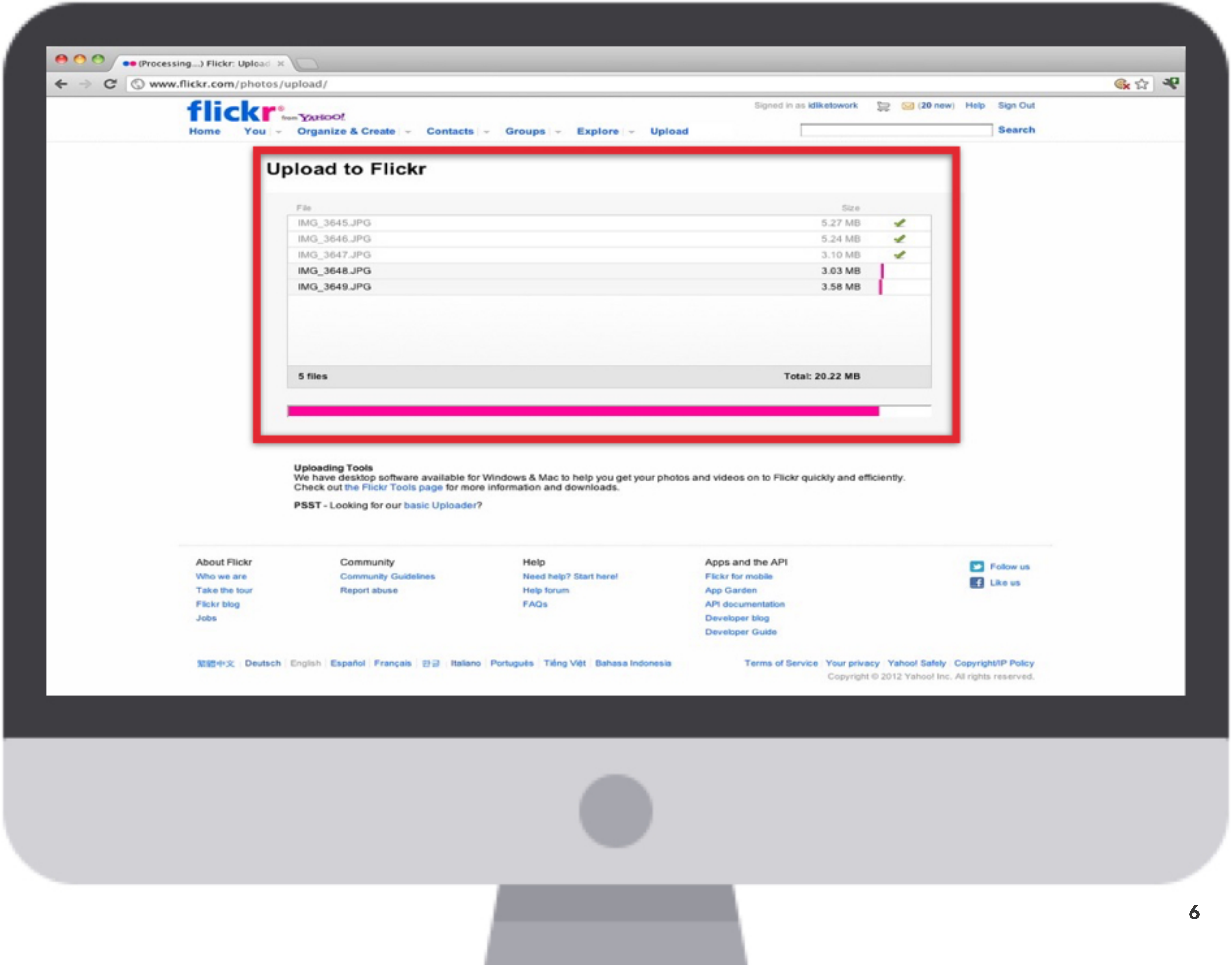
What makes Node.js so special?

- It allows you to run JavaScript outside of the browser.
    - Which means you can build desktop apps, too!
- It's really fast but..the reason for that is a little complicated.
- Let's give it a little context by understanding how and why Node was created.

NODE.JS THE BEGINNING

Ryan Dahl

- Creator of Node.js
- He first presented Node.js at a JavaScript Conference in 2009
- He was inspired after watching a Flickr demo



Upload to Flickr

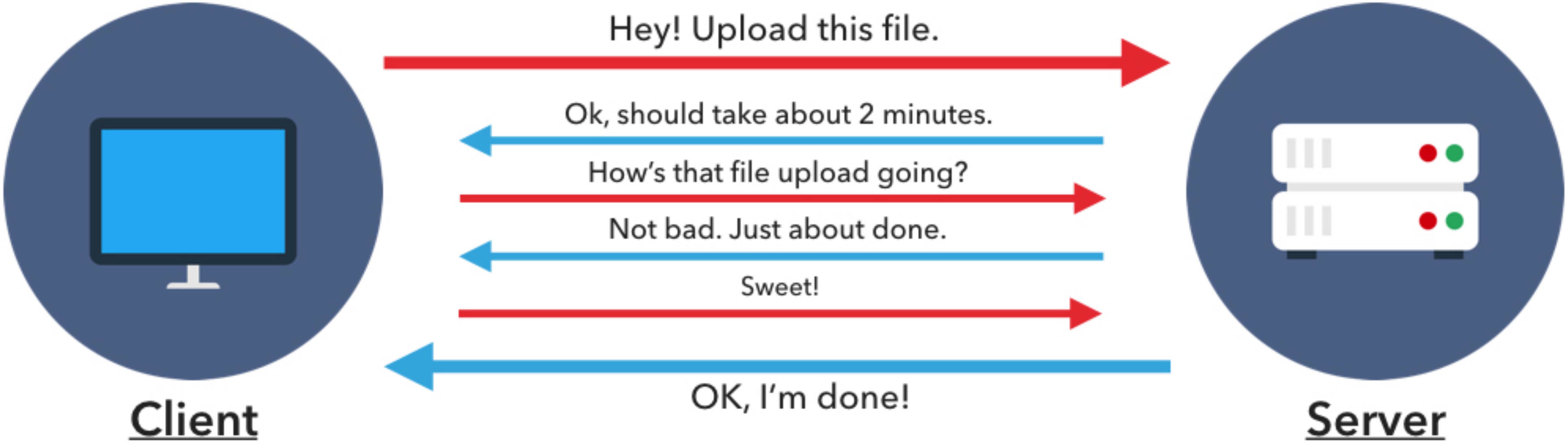
File	Size	
IMG_3645.JPG	5.27 MB	✓
IMG_3646.JPG	5.24 MB	✓
IMG_3647.JPG	3.10 MB	✓
IMG_3648.JPG	3.03 MB	
IMG_3649.JPG	3.58 MB	
5 files	Total: 20.22 MB	

Uploading Tools
We have desktop software available for Windows & Mac to help you get your photos and videos on to Flickr quickly and efficiently.
Check out the Flickr Tools page for more information and downloads.

PSST - Looking for our basic Uploader?

Copyright © 2012 Yahoo! Inc. All rights reserved.

NON- BLOCKING



Hey! Upload this file.

Ok, should take about 2 minutes.

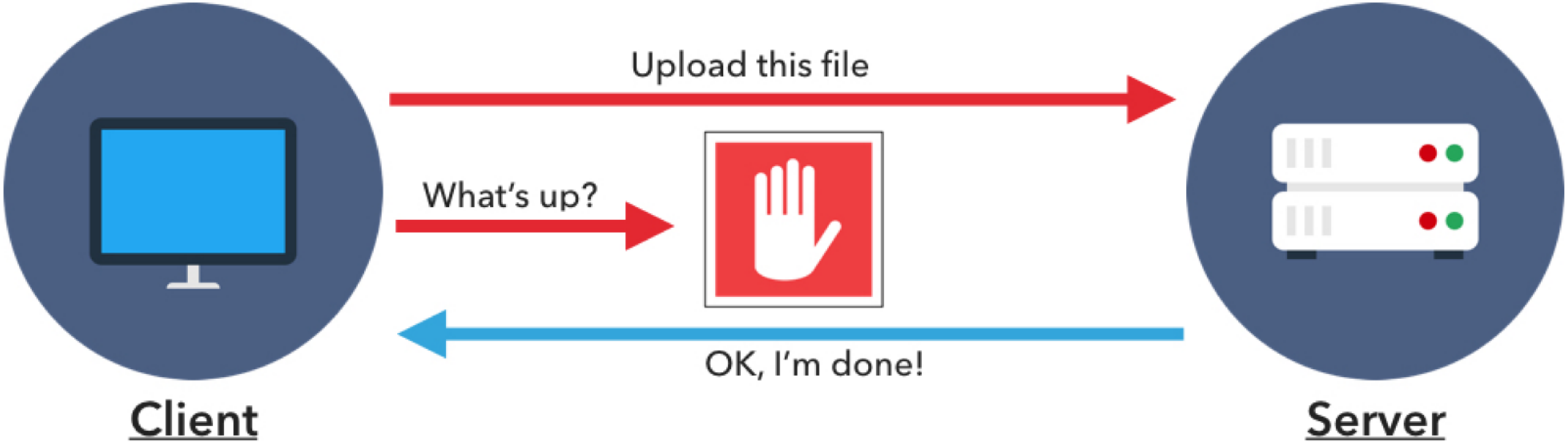
How’s that file upload going?

Not bad. Just about done.

Sweet!

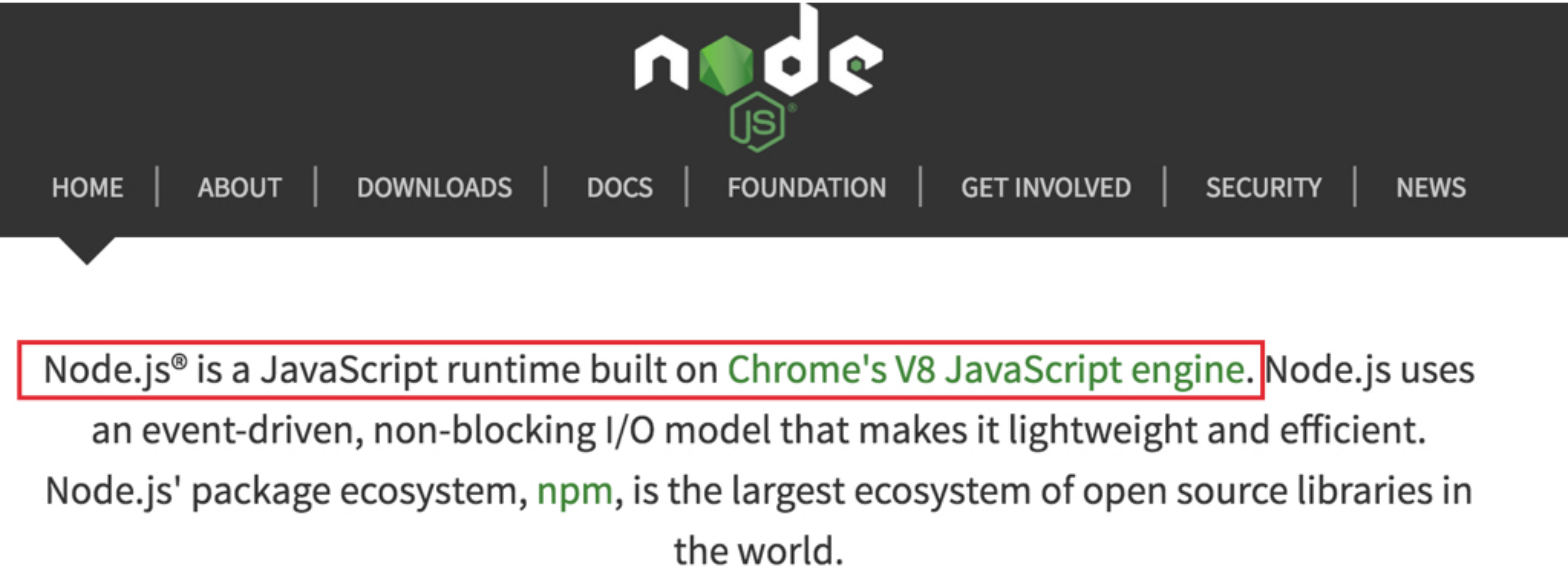
OK, I’m done!

BLOCKING



WHAT IS NODE.JS?

https://nodejs.org



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

```
Node.js is a JavaScript runtime built on Chrome's V8
JavaScript engine.
```

What is a JavaScript Engine?

- A program that compiles JavaScript code into native machine code.
- Sometimes referred to as a compiler.
- Example engines include:
  - V8 (Google)
  - SpiderMonkey (Mozilla)
  - JavaScriptCore (Apple)
  - Chakra (Microsoft)

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.

What is a JavaScript Runtime?

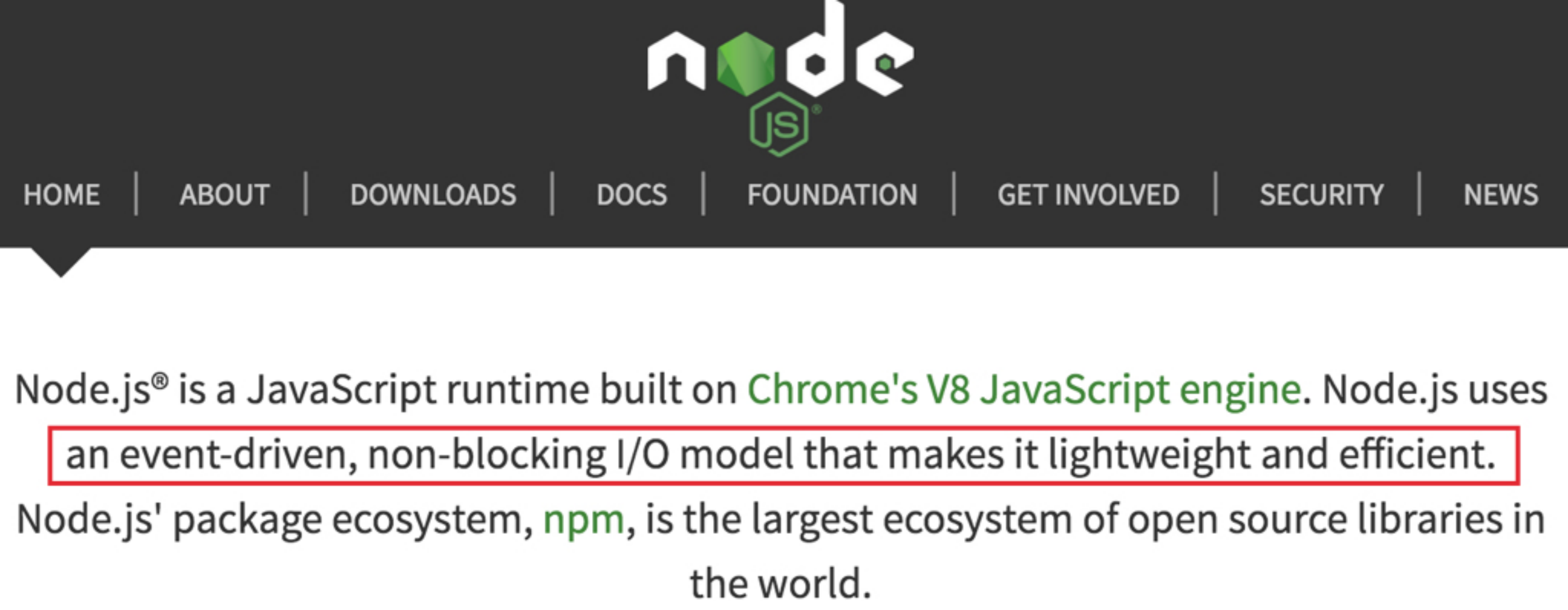
- A library used by the JavaScript Engine to implement functions during runtime aka execution of a program.
- These libraries often include functions for input/output and memory management.
- Example runtimes include:
  - Node.js
  - Browsers

Google Chrome

- Uses a **client-side** JavaScript Runtime
- The Runtime is part of the browser
- Handles tasks, such as:

Node.js

- Is a **server-side** JavaScript Runtime
- Is installed on a computer or server
- Handles tasks, such as:
  - Database queries
  - File I/O Requests



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js' package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Node is **Event-Driven**

Remember the JavaScript Event-Handling lecture?

- When this **event** happens, do this **action**.
    - *Example:* When a user `clicks` this button, `display` this menu.
    - This is considered a *Client-Side* event.

Node is **Event-Driven**

- Some common server-side events, include;
  - `connect`
  - `abort`
  - `open`
  - `close`
- *Example*: When this file is open, `append` the date.

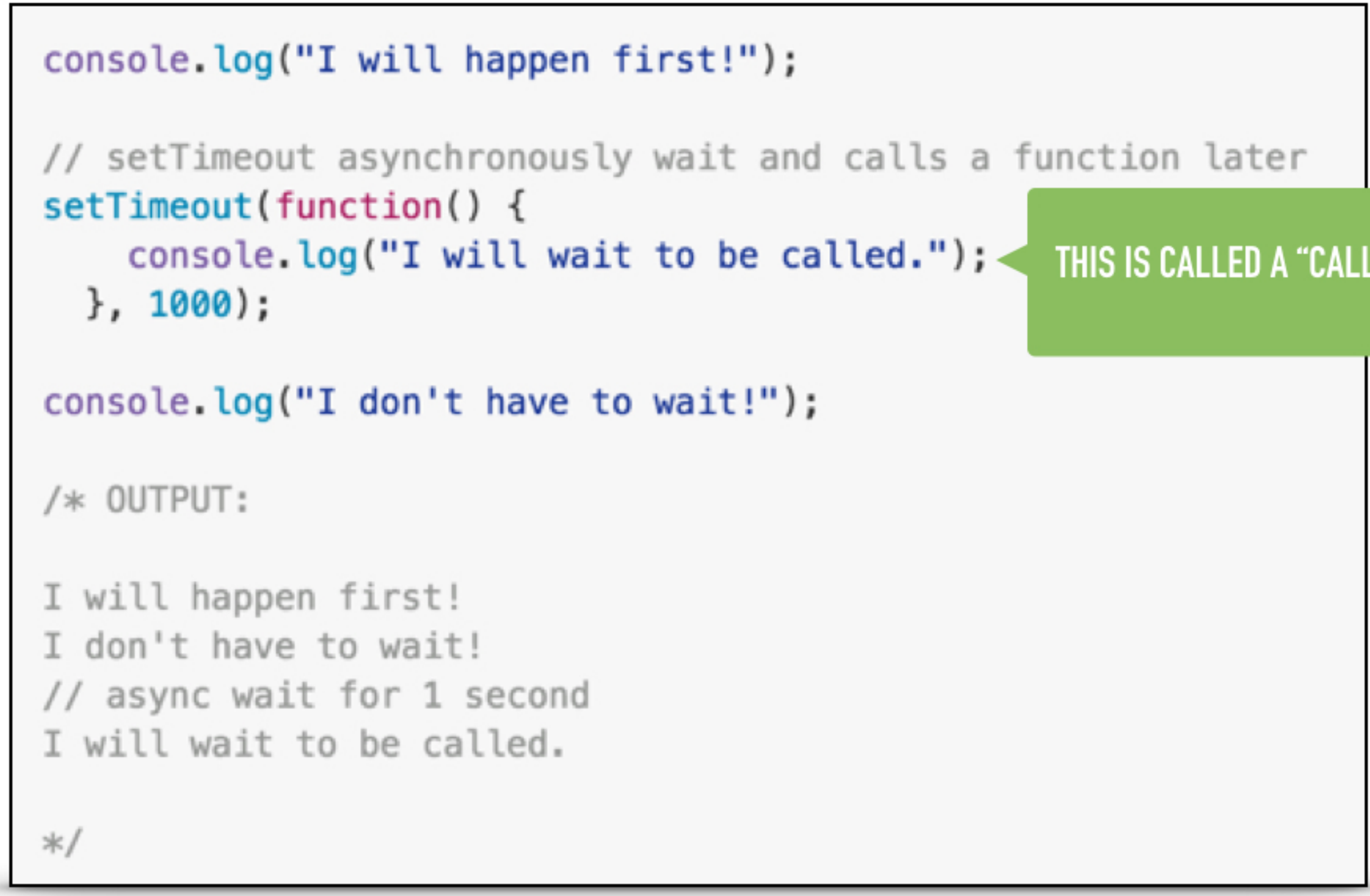
Node is **Event-Driven**

- Node is always **listening** for new `events`
    - Kind of like a nosy neighbor

Node is **Event-Driven**

- When Node recognizes an `event`, it sends the relating action off to process, then creates a `callback`.
  - A **`callback`** is just that, Node calls back that action, so it can answer another event.
  - *Example:* When this file is `open`, `append` the date…**brb**…Ok, now `close` the file.

Asynchronous Code



```
console.log("I will happen first!");

// setTimeout asynchronously wait and calls a function later
setTimeout(function() {
    console.log("I will wait to be called.");
  }, 1000);

console.log("I don't have to wait!");

/* OUTPUT:

I will happen first!
I don't have to wait!
// async wait for 1 second
I will wait to be called.

*/
```

THIS IS CALLED A "CALLBACK"

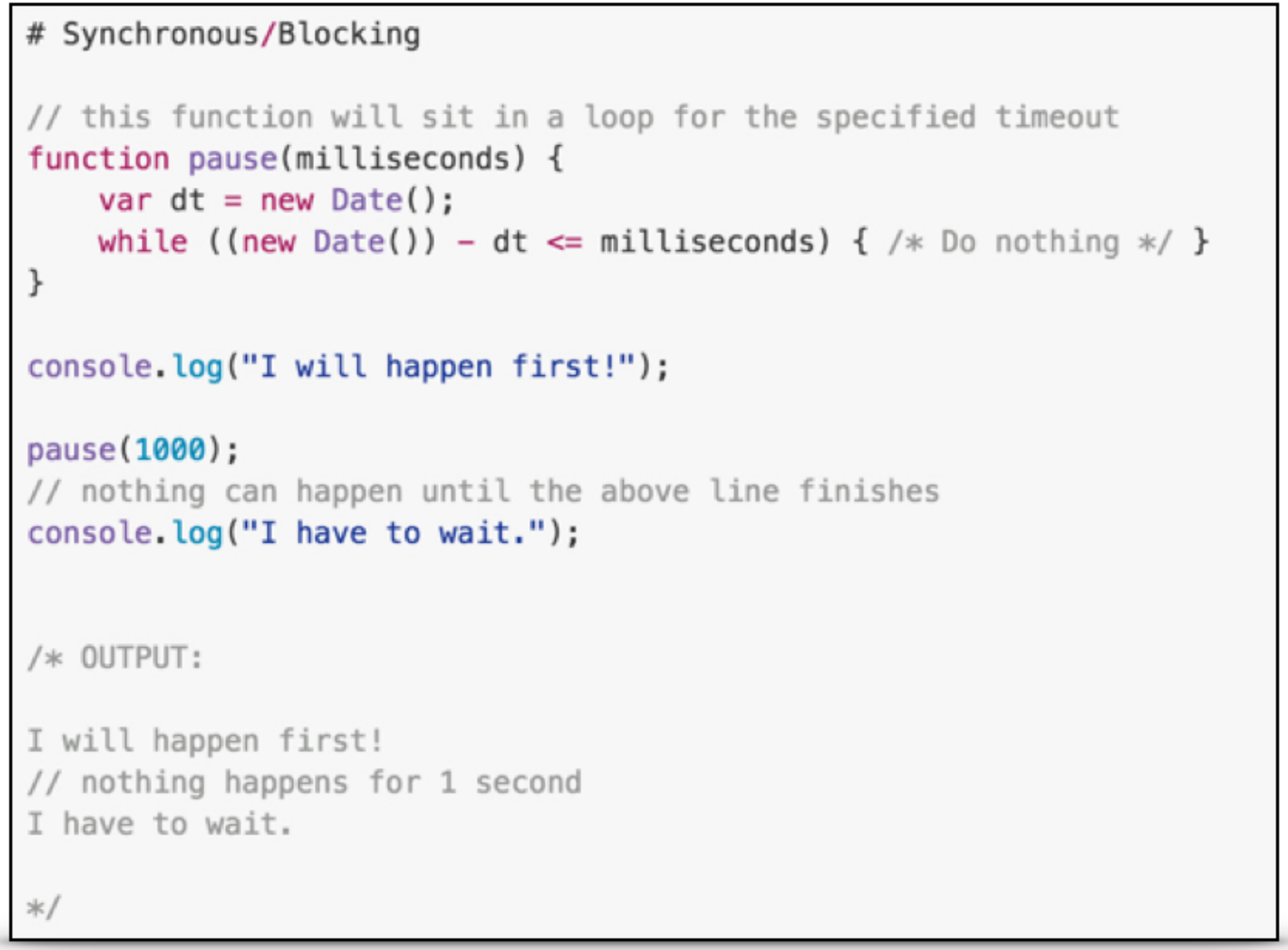
Node is **Non-Blocking**

- Non-Blocking operations are sometimes referred to as **Asynchronous** operations
- Other code will execute while Node waits for the asynchronous operation to complete

vs **Blocking**

- Blocking operations are sometimes referred to as **Synchronous** operations
- No other code can execute until the synchronous operation completes
- If the operation is slow, this can be an issue

Synchronous Code



```
# Synchronous/Blocking

// this function will sit in a loop for the specified timeout
function pause(milliseconds) {
    var dt = new Date();
    while ((new Date()) - dt <= milliseconds) { /* Do nothing */ }
}

console.log("I will happen first!");

pause(1000);
// nothing can happen until the above line finishes
console.log("I have to wait.");


/* OUTPUT:

I will happen first!
// nothing happens for 1 second
I have to wait.

*/
```

Node is Single-Threaded

- A thread is a single computer process
- Node's main Event Loop runs in a single thread
- Events and Callbacks are queued in the order they are received

Event Loop Example

- A web request is received
- Node executes the handler for that request
- The handler initiates a database query, with a callback
- Node is free and able to handle other requests
- The database query ends, and Node is notified (event)
- Node adds the callback to the queue
- Node executes the handler after processing any events before it in the queue

WHY USE NODE.JS?

On It's Own, Node is **NOT**...

- a web framework
  - so, it's not a JavaScript version of Ruby on Rails

HOWEVER, when **paired** with…

- web frameworks, such as **Express** or **AngularJS**
  - you can create full-fledged web applications

You can ALSO **create**...

- desktop applications
- video games
- chat rooms
- remote controls for toy cars 😮

INSTALLING NODE.JS

Installation Steps

1. Go to https://nodejs.org
2. Click the **LTS** download button
3. Open Installer
4. Follow prompts to complete the installation

Basic Node Terminal Commands

- **`node`**
  - opens an interactive shell where you can execute JavaScript code
- **`node file_name.js`**
  - executes JavaScript code that is in a file
- **`node -v`**
  - displays the version of Node installed on your computer

Exercise #1: Hello Node!

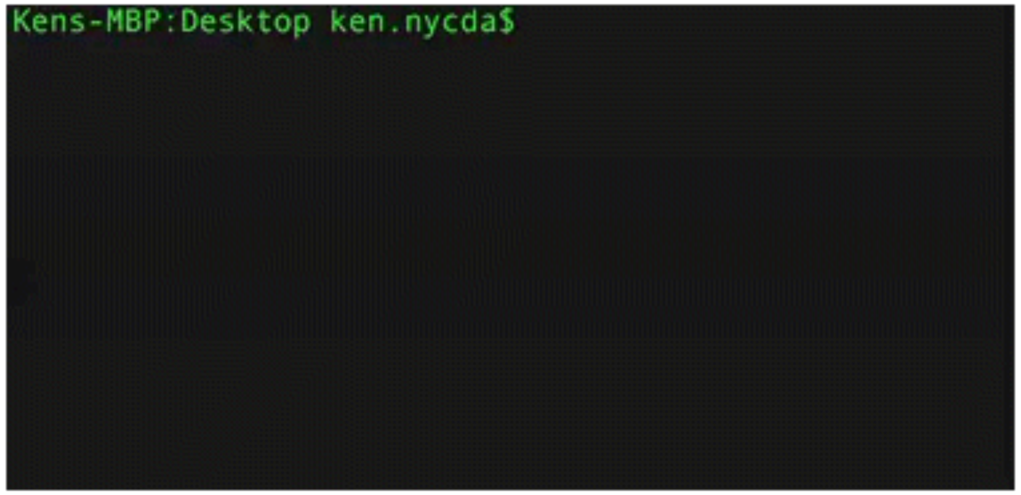
- Using the Node interactive shell, output **`Hello Node!`** in the terminal

- Save a **`Hello Node!`** script in a file and execute that file in the Terminal

Homework: Descending String Interval in the Terminal

Write a program that will countdown from 10. Instead of using actual numbers for the countdown, have the output display a string consisting of asterisk marks.

For example, if the countdown is at 5, there should be five asterisk marks *****.



```
Kens-MBP:Desktop ken.nycda$
```