Day 30

Know Thy Node

Rutgers Coding Bootcamp | April 4, 2016

Project Week

Instructor’s Feedback

Seriously, mind-blown.

A Little Perspective

Last time I said this you barely knew what a "div" was.

Specifically…

Things I've noticed people doing *incredibly* well:

- Stunning front-end (Seriously, you guys are better than me at this point)
- Amazing execution of challenging concepts (no one took the easy road)
- Fantastic explanations of your code and technology
- And basically everything.

Seriously.

Put the video + GitHub code on your LinkedIn profiles

Next Steps…

Close out your projects well!

- Create README’s for your code on GitHub
- Use a custom domain URL
- Create default, working “test cases”
- Consider writing a blog article / video that builds from scratch
- Fork / Star other people’s code

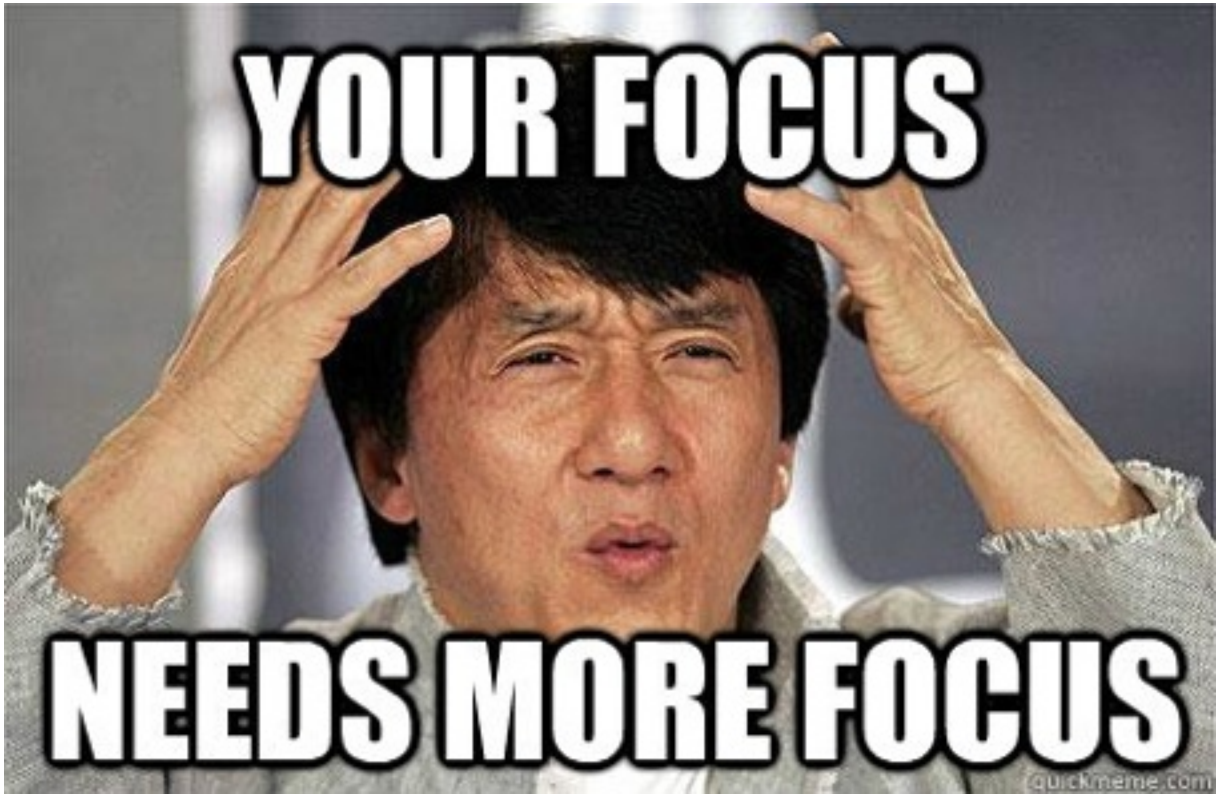
And seriously.

Put the video + GitHub code on your LinkedIn profiles

Assessments!

The Mystery of “Backend”

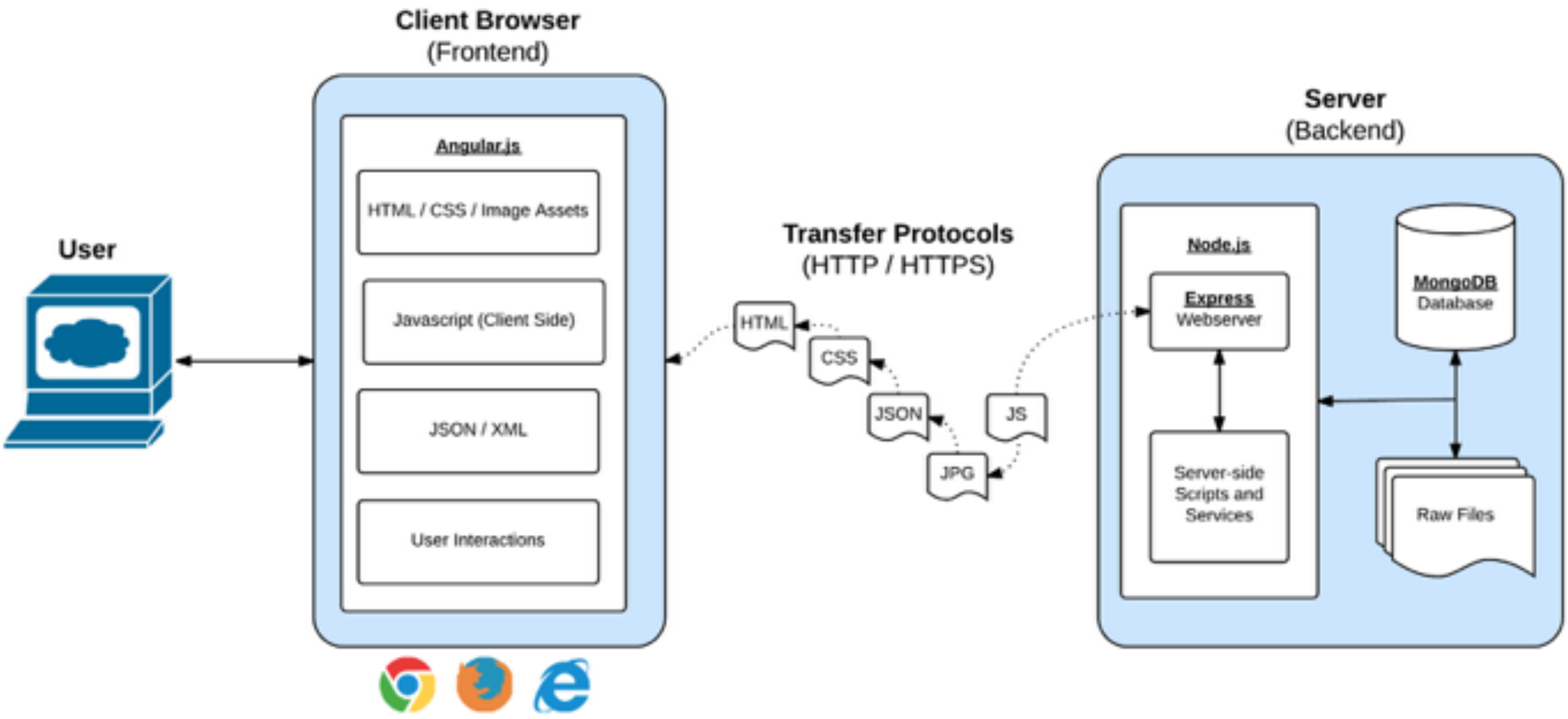
FOCUS!



This next stuff is important!

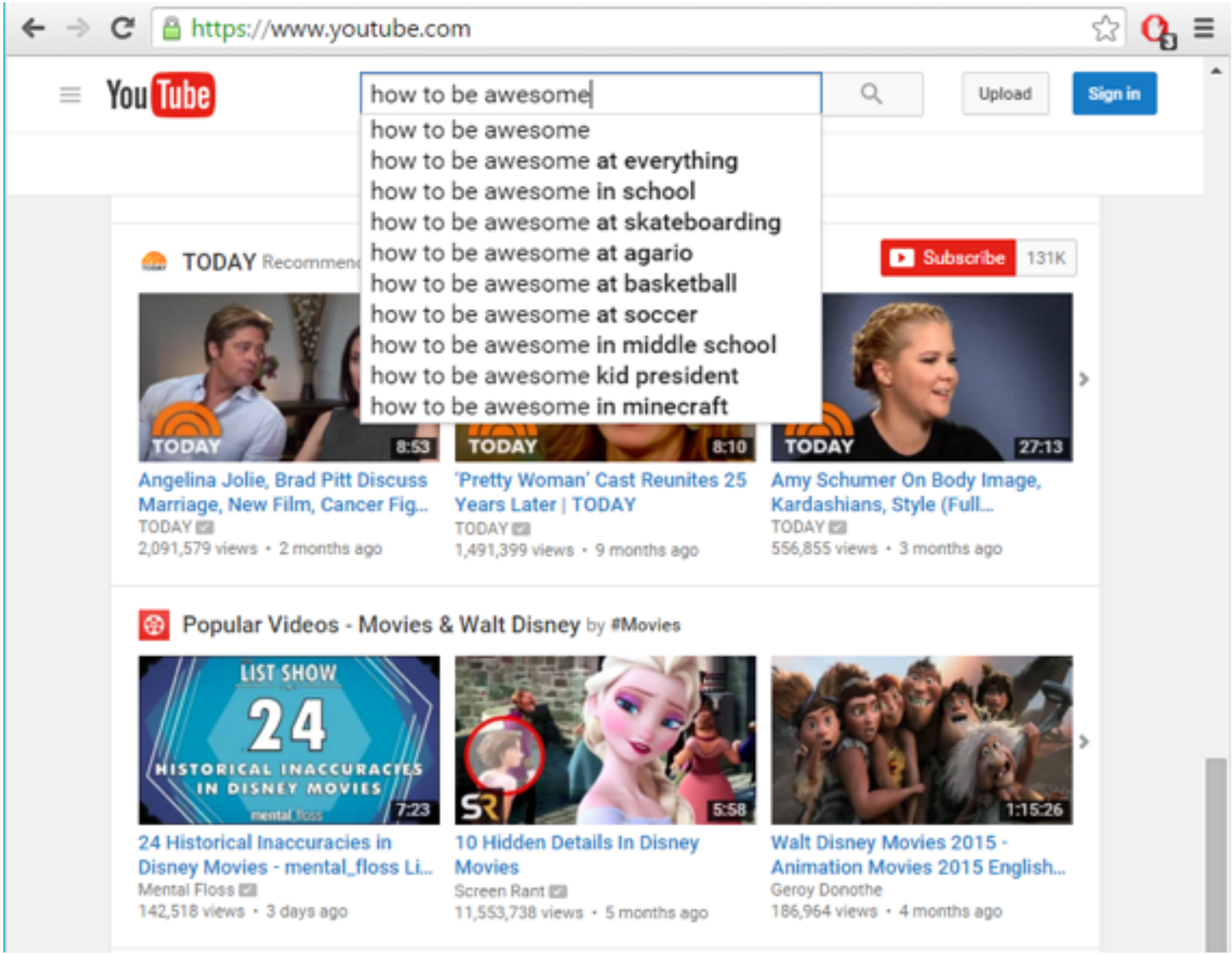
Full-Stack Development?

Full-Stack Development



- In modern **web applications** there is a constant back-and-forth communication between the visuals displayed on the user's browser (**frontend)** and the data and logic stored on the server (**backend).**

The "Magic" of YouTube



https://www.youtube.com

how to be awesome

- how to be awesome
- how to be awesome at everything
- how to be awesome in school
- how to be awesome at skateboarding
- how to be awesome at agario
- how to be awesome at basketball
- how to be awesome at soccer
- how to be awesome in middle school
- how to be awesome kid president
- how to be awesome in minecraft

Angelina Jolie, Brad Pitt Discuss Marriage, New Film, Cancer Fig...
TODAY
2,091,579 views • 2 months ago

'Pretty Woman' Cast Reunites 25 Years Later | TODAY
TODAY
1,491,399 views • 9 months ago

Amy Schumer On Body Image, Kardashians, Style (Full...
TODAY
556,855 views • 3 months ago

Popular Videos - Movies & Walt Disney by #Movies

24 Historical Inaccuracies in Disney Movies - mental_floss Li...
Mental Floss
142,518 views • 3 days ago

10 Hidden Details In Disney Movies
Screen Rant
11,553,738 views • 5 months ago

Walt Disney Movies 2015 - Animation Movies 2015 English...
Geroy Donothe
186,964 views • 4 months ago

Key Question

Examples of “Server-Side” Code?

Server-Side Code in Action!

- API that parse URL parameters to provide selective JSONs
- Firebase methods that provide a timestamp back to users
- Clicking an invoice that provides a PDF report
- Image processing software that takes an image applies a filter, then saves the new version
- Google providing “results” relevant to your searches on other sites.

Critical Question

What is a “server”?

Definition of “Server”

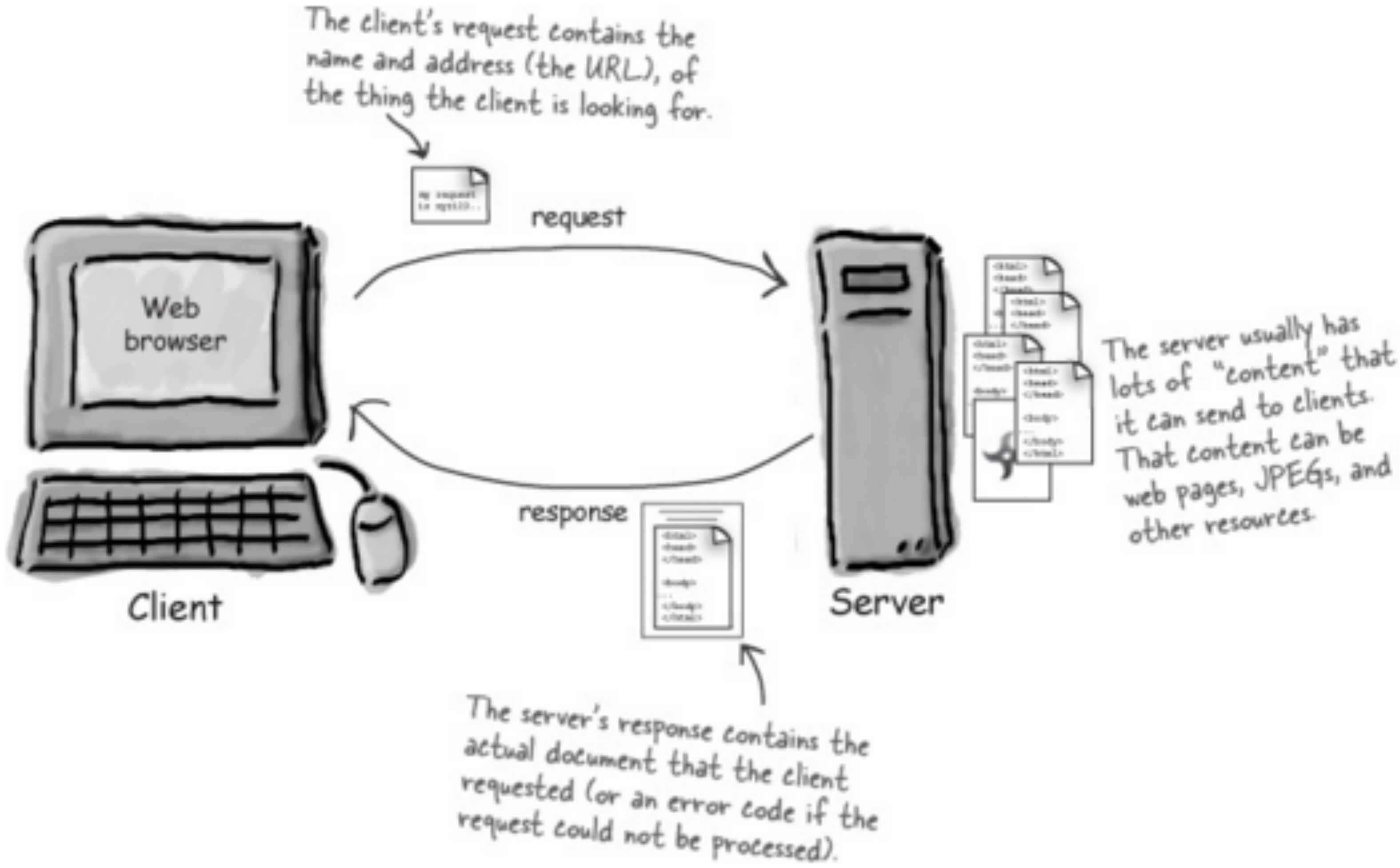
A web server takes a client request and gives something back

“A web browser lets a user request a resource. The web server gets the request, finds the resource, and returns something to the user. Sometimes the resource is an *HTML* page. Sometimes it's a picture. Or a sound file. Or even a PDF document. Doesn't matter--the client asks for the thing (resource) [or action] and the server sends it back.”

“... When we say "server", we mean either the physical (hardware) or the web server application (software) [that actually runs the server commands]”

Kathy Sierra, Author of Head First Servlets and JSP

Server Definition


The client's request contains the name and address (the URL), of the thing the client is looking for.
request
Web browser
Client
response
Server
The server usually has lots of "content" that it can send to clients. That content can be web pages, JPEGs, and other resources.
The server's response contains the actual document that the client requested (or an error code if the request could not be processed).

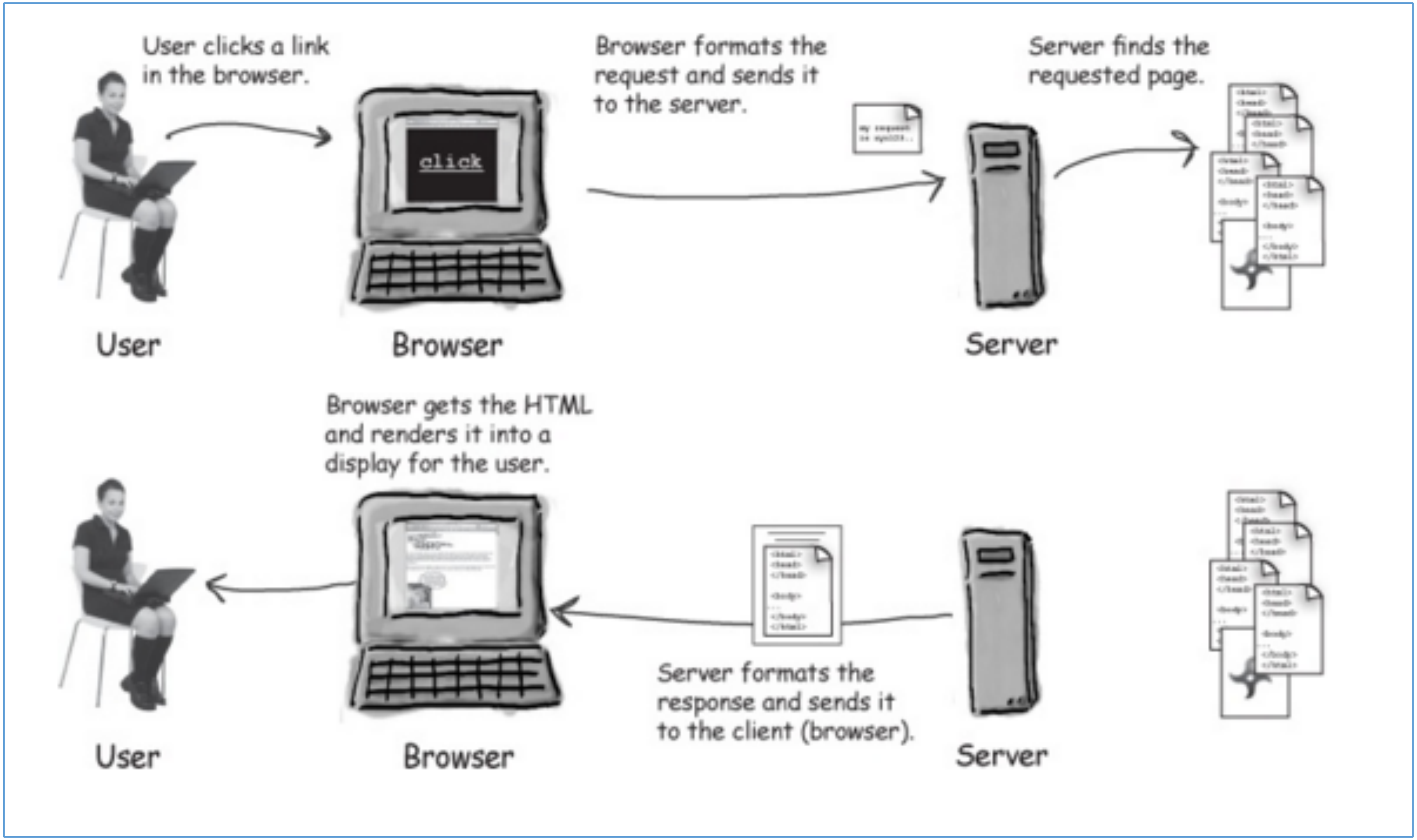
Definition of “Web Client”

Web client lets the user request something on the server and then shows the result (response) of the server.

When we talk about client, though, we usually mean both (or either) the human user and the browser application. The browser is the piece of the software that knows how to communicate with the server. The browser's other big job is interpreting the HTML code [sent by the server] and rendering the web page to the user.

Kathy Sierra, Author of Head First Servlets and JSP

Web Client Definition



User clicks a link in the browser.
Browser formats the request and sends it to the server.
Server finds the requested page.
click
User
Browser
Server
Browser gets the HTML and renders it into a display for the user.
Server formats the response and sends it to the client (browser).
User
Browser
Server

> YOUR TURN!!

Assignment

Talk to the person next to you and re-explain to one another the following terms:

- Server
- Web Client
- Request
- Response

Aight. Relax.

Yay!
You made it through the ultra important stuff!

You can go back to Facebook now

(But…. Not really, the next stuff is still important)

Intro to Node.JS

Key Question

So what is NodeJS?

Definition of "NodeJS"

Node.js is an open-source, cross-platform JavaScript runtime environment designed to be run outside of browsers.

It is a general utility that can be used for a variety of purposes including asset compilation, scripting, monitoring, and **most notably as the basis for web servers**

Our made-up definition of Node. Yay for sounding intelligent!

> YOUR TURN!!

Assignment

Take a few moments to research 5 companies that actively use NodeJS in production.

Why Use NodeJS as a Server?

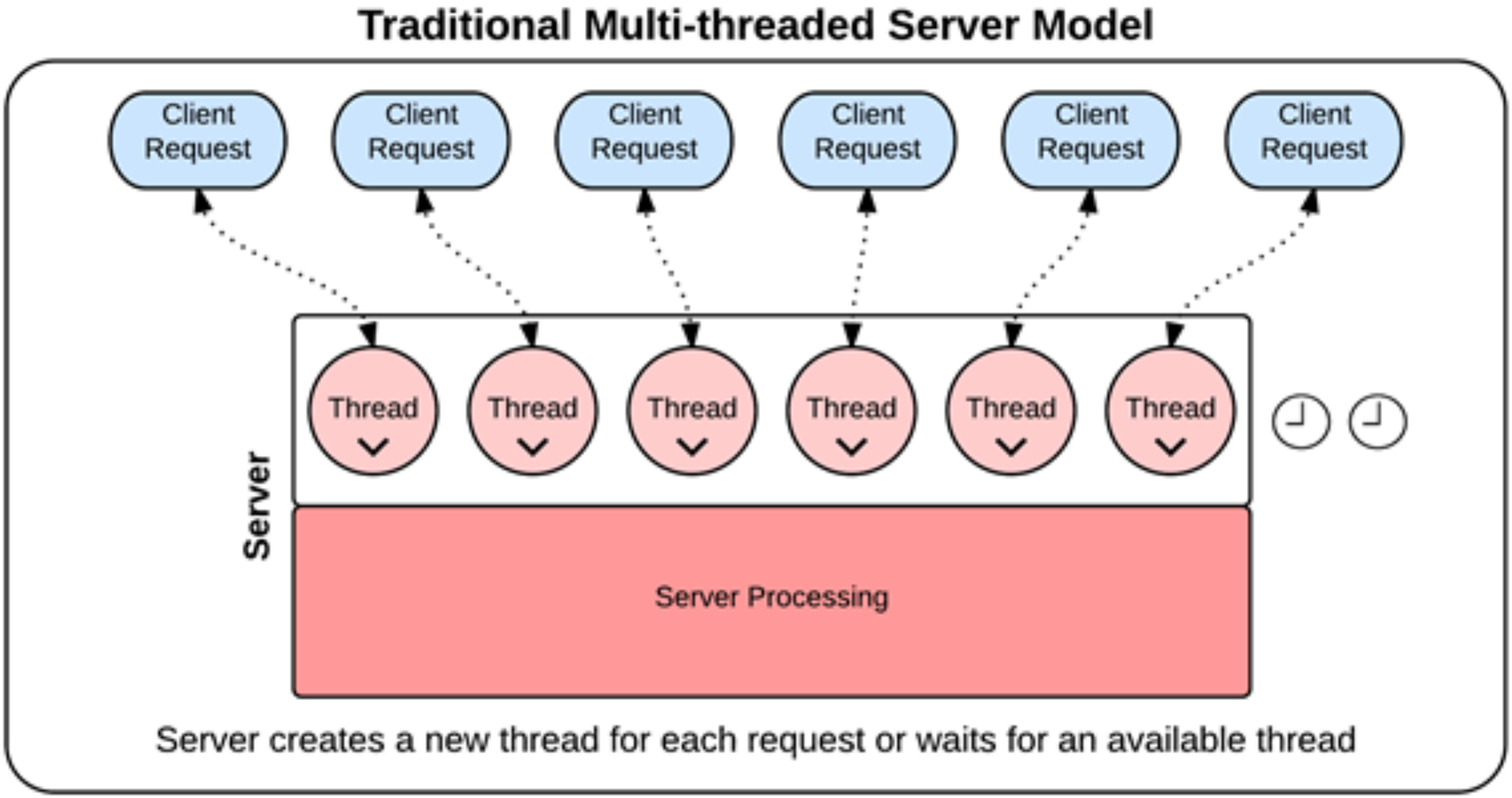
- **It re-uses Javascript** – meaning a front-end Javascript developer can also build an entire server themselves

- **It's easily extendable.** Numerous plugins exist to expand the capabilities of Node

- **Fast-implementation,** which allows for the creation of an entire working server with only a few lines of code.

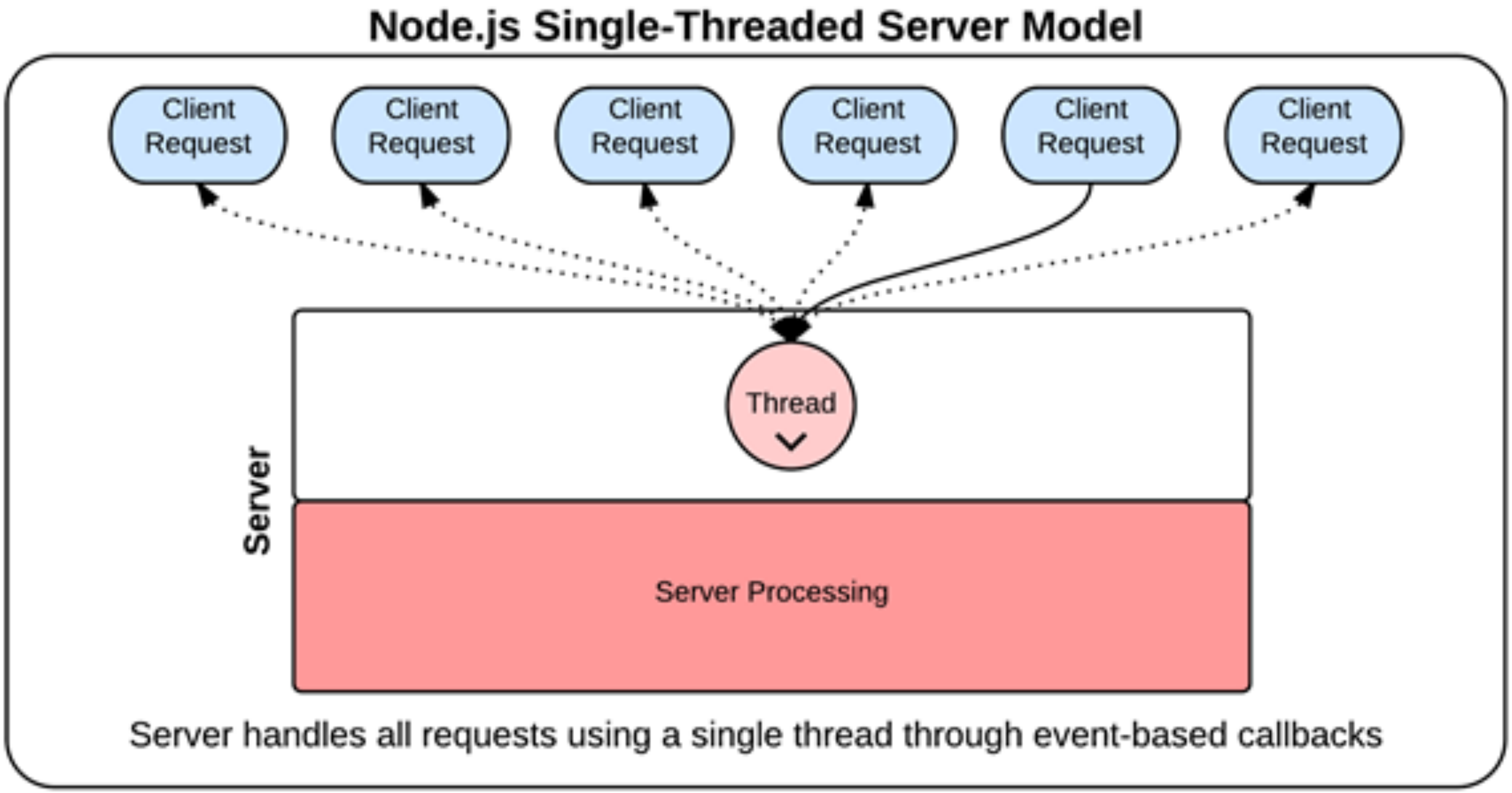
- **Single-Threaded Asycncronous Model** – meaning it can handle multiple requests simultaneously and not get bottlenecked.

Syncrononous Threading



In traditional synchronous threading, <u>each request requires its own thread</u>. No other request can pass through that thread until complete. Since there is a limited pool of threads, <u>this can create bottlenecks.</u>

Asynchronous Threading (Node Way)


Node.js Single-Threaded Server Model
Client Request
Client Request
Client Request
Client Request
Client Request
Client Request
Thread
Server
Server Processing
Server handles all requests using a single thread through event-based callbacks

In Node-based asynchronous threading, a single thread is used throughout. Each thread is "put to the side" using callbacks and responded to when ready. Because of this, there is <u>no limit on the number of requests</u> that can be responded to and there is <u>no bottleneck</u>.

Coding Time!

Homework!