



< WEB DEVELOPMENT />

[WWW.HACKYALE.COM](http://WWW.HACKYALE.COM)



< WEB DEVELOPMENT />

week\_0  
DIVING IN!

# WELCOME TO FRONT END DEVELOPMENT

## The Agenda

- Introductions
- What is HackYale?
- To the code!



## INTRODUCTIONS

LET'S GROW TO LOVE ONE ANOTHER

# INTRODUCTIONS

## Dani Dickstein

- First time teaching Front End Development
- Sophomore, Saybrook, Computer Science Major
- Loves Computer Science because it's our only super power.

# INTRODUCTIONS

## Alex Reinking

- Taught Front End Development last Fall
- Long-time programmer
- Sophomore, Trumbull, Math/Computer Science Double Major
- Worked on Euterpea with Master Paul Hudak last summer
- Crippling addition to Frosted Flakes

# INTRODUCTIONS

Where are we?

- The Center for Engineering, Innovation and Design
- Must be a member! ([ceid.yale.edu](http://ceid.yale.edu))
- Be respectful, etc.

# INTRODUCTIONS

Who are we?

- Budding programmers, all-star designers, clueless freshmen
- Coding is more fun with friends!



WHAT IS THIS HACKYALE?  
PROGRAMMING + HACKING CULTURE

# HACKYALE

---

Practical, not as theoretical / academic as  
a Yale CS class

- Zero -> prototype
  - Not training CTOs
  - Preparing you with the tools to train  
yourself to do whatever you want
-

# WHY HACKYALE?

---

Good ideas + good developers =  
good tech companies

- Yale  $\supset$  many students with good ideas
  - Yale  $\nexists$  many students who can implement those ideas
-

*“But it’s not that probable that I’m going to make an awesome dynamic web app while I take a classes and go abroad and I double major and I try to set up an internship for next summer/job after college and I stress about my love life and I don’t call my friends from home enough and I worry about whether or not those ants in the corner of my common room are breeding!”*

*“But it’s not that probable that I’m going to make  
an awesome dynamic web app while I take a  
classes and go abroad and I double major and I  
try to set up an internship for next summer/job  
after college and I stress about my love life and I  
don’t call my friends from home enough and I  
worry about whether or not those ants in the  
corner of my common room are breeding!”*

## Fair Point



GIVE WHAT YOU CAN

AND MAKE TIME TO GIVE A LOT

# COURSE OBJECTIVES

- Learn how to build pretty, static websites.
- Give you a jumping off point to learn more and build dynamic sites, maybe even this semester.
- For designers, another canvas on which to showcase your craft.

# BE YOUR OWN MOTIVATOR

We can't make you successful developers

We can equip you with a kernel of knowledge and key resources with which to make yourselves successful developers



# HACKING CULTURE

THE “DO-IT-YOURSELF” ATTITUDE

# ANOTHER POSSIBLE MOTIVATOR?

HackYale opens a network for finding cool tech jobs

- Brewster, Google, Art.sy, Microsoft, Palantir, Panorama Education, RedFin, Twitter, SeeClickFix...



TO THE COMPUTERS!

ABOUT TIME!

## Quick note: Mac vs. PC

All examples and screenshots are performed and taken on Macs. However, everything we do in this class should be pretty much the same for those of you on PC.

# GETTING STARTED

- Create a folder called “hackyale” on your Desktop
- In it, create a folder called “week\_0”
- Open Sublime and open that folder
- Create index.html
- Type “Hello World” and save it
- Open index.html in Google Chrome (and set as default behavior)

# VERSION CONTROL with Git

TIME TO GET LEARNED

# WHAT IS VERSION CONTROL?

In a nutshell, version control is a really advanced tool for saving your progress.

# WHAT'S WRONG WITH FILE → SAVE?

Imagine you are working on a super important essay, like your application to Yale.

So you write one essay and that's draft1.docx. And you want to be able to go back to previous versions, so you have draft2.docx and draft3.docx.

# WHAT'S WRONG WITH FILE → SAVE?

And then you are like “Wow, this is terrible. I need to start over.”

So you get draft1b.docx and draft2b.docx.

And then you like one paragraph from draft3.docx and two other paragraphs from draft2b.docx and you merge them together and get draft4.docx.

# WHAT'S WRONG WITH FILE → SAVE?

And then you send it to a couple classmates to edit, and maybe your English teacher, and they all give you totally different revisions. That's draft5a.docx, draft5b.docx, draft5c.docx.

Where was that one paragraph you wrote back in September? What was draft2b.docx again?

All of a sudden your organized approach becomes a complete mess!

# WHAT'S WRONG WITH FILE → SAVE?

But you're not dealing with a 600 word essay.  
You are dealing with hundreds of lines of code.

In a professional environment, millions of lines  
of code.

So we need a better system. We need Version  
Control.

# How DOES GIT WORK?

In your coding project (your folder) you create a new Git **repository**, which will automatically watch your project and track all the changes in content.

Every time you want to “save,” you **commit** your changes to Git.

# GIT BRANCHES

All of your work in Git is done on a **branch**, which is like a version of your project.

You can create multiple branches and flip between them so you can compare code in different versions.

Moving from one branch to another is called **checking out**.

# MERGING TO MASTER

The main branch is called the **master branch**. You should only put code on the master branch when it's fully tested and ready.

Usually all your work will be done in a separate branch, and when the code in that branch is ready, it will be **merged** into the master branch.

# PURPOSE OF GIT

Git is a bit more than a tool to save your progress.

**Main purpose:** Efficient and Easy Collaboration

# GITHUB

GitHub is a website that hosts Git repositories.

After hosting your repository on GitHub, you can push any changes that you make locally so that the repository on GitHub will be updated.

If you are collaborating on the project with others, they can pull down your changes from GitHub, and when they are ready, push their own.



**WELCOME TO  
THE FRONT-END**

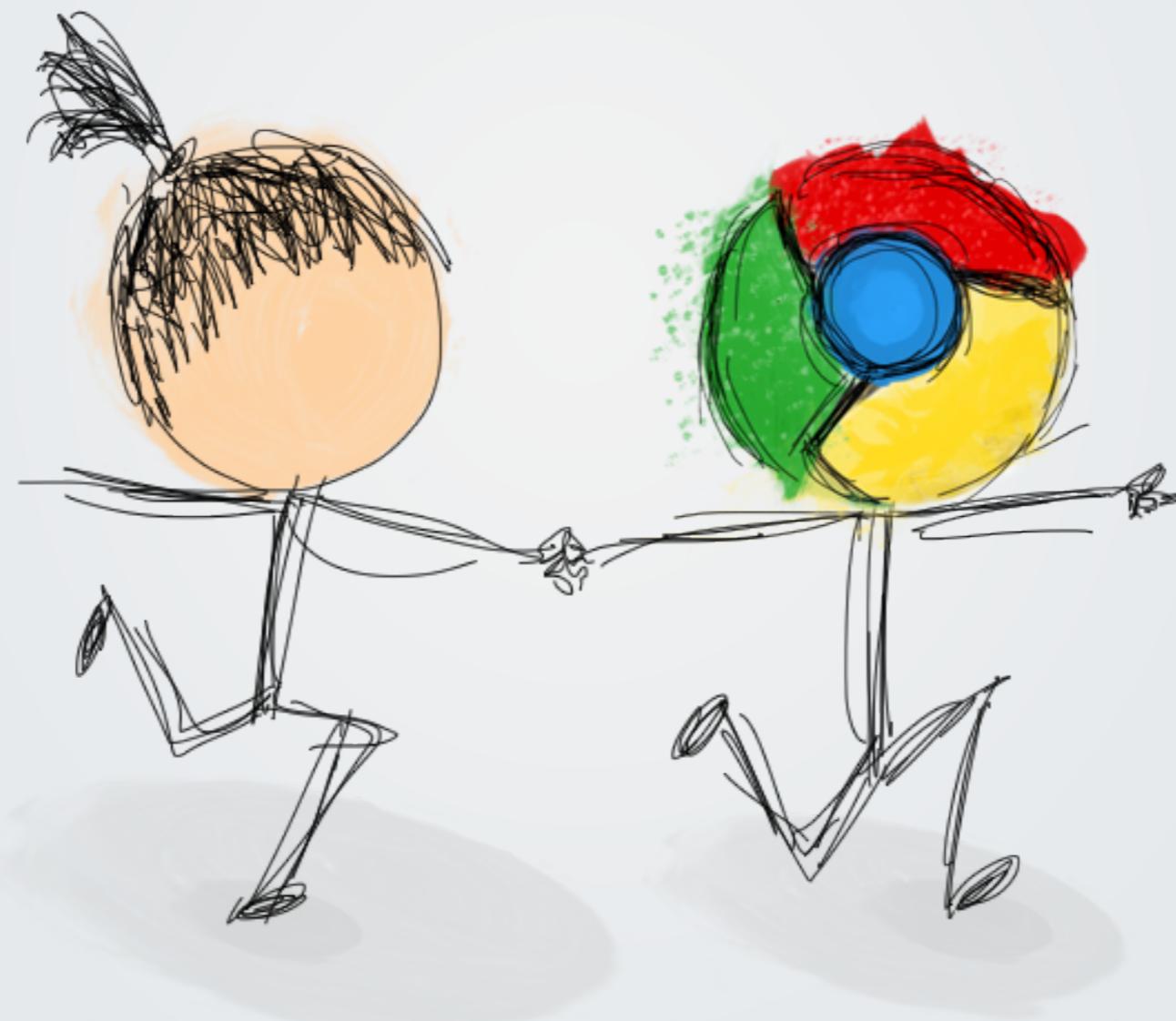


## KEY CONCEPTS

THINGS YOU WANT TO BE AWAKE FOR

# KEY CONCEPT 0

# GOOGLE IS YOUR FRIEND



# KEY CONCEPT 0

80% of web development is knowing where to look

Most common answer = Google

- Things to Google:
  - Error messages (when something goes wrong)
  - Syntax (how to say things in code)
  - How to do basically anything.
  - Think of whatever question you would ask an expert, and ask it to Google first.

*WHAT DO WE DO  
WHEN WE  
ENCOUNTER A  
PROBLEM WE CAN'T  
IMMEDIATELY  
SOLVE?*





**GOOGLE IS YOUR FRIEND**

**BUT WE ARE HERE FOR YOU TOO!**

**WE'LL GO FAST, BUT THIS IS AN  
INTERACTIVE CLASS.**

**PLEASE STOP US WHENEVER YOU HAVE A  
QUESTION.**

# KEY CONCEPT 1

There are two parts to learning

- Concept
  - What you can do
- Syntax and implementation
  - How to do it

# WE WILL TEACH YOU CONCEPTS

But the implementation is on you!

- The fastest way to learn is practice, practice, practice
  - Making mistakes helps a lot, too
- Please, please, please, follow along examples in class
  - If you're bored, think of how you can make it better
  - If you don't get it, ask for help!

# KEY CONCEPT 2

Code is meant for humans to read

- *Extremely* important to be clear and concise
- Rely on conventions
  - Underscores instead of spaces, start with lower case...
- Use comments!
  - Comments are lines that are ignored by the computer, so you can tell your fellow programmers how your code works.
  - For others, and for your future self



**ANATOMY OF A WEBPAGE**  
HEAD, SHOULDERS, KNEES AND FOOTER

# FRONT-END DEVELOPMENT

An interaction between three “languages”

- HTML - HyperText Markup Language
- CSS - Cascading Style Sheets
- JavaScript - A programming language, NOT related to Java

*To be a successful developer, you'll need to learn all three*



IMAGINE FOR A SEC...

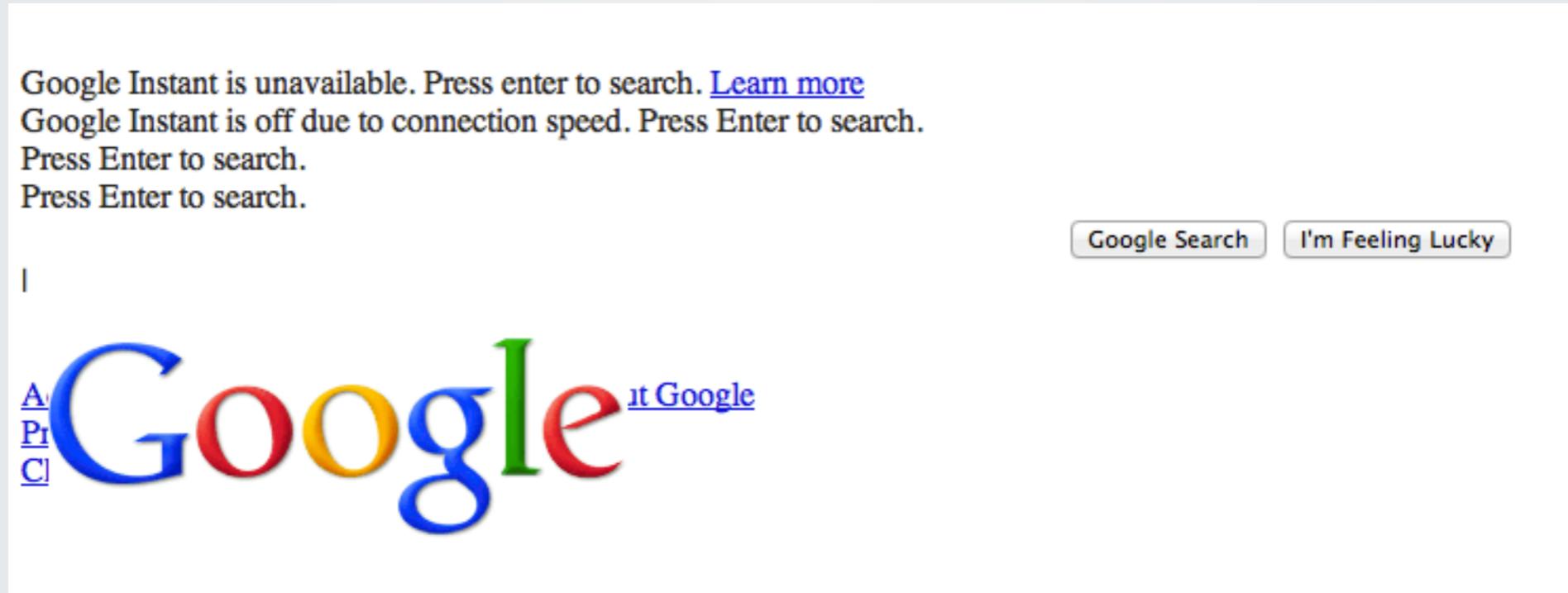
THAT A WEBSITE IS A HUMAN BODY

# HTML

## The “bones”

- The “content” of the Internet
- Builds the layout, structure and connections
- Easy to learn, easy to master

# THE HTML OF GOOGLE'S HOMEPAGE



In a week, you'll be able to make this!

# CSS

The “skin” or “physical features”

- The “style” of the Internet
- Defines how HTML elements look
  - Width, height, color, position...
- Easy to learn, difficult to master
- The web designer’s best friend

# JAVASCRIPT

## The “muscles”

- The “interaction” or “animation” of the Internet
- Makes HTML elements interact with one another
  - And with other pages!
- Tougher to learn, and unfortunately, tough to master



# HTML: THE CONCEPT

SOME BOARD WORK REQUIRED



# HTML: IMPLEMENTATION

TYPE ALONG!

# Coding in HTML

To program the nested structure of HTML, we use *tags*

- Tags are just 1-4 letters that refer to something
  - Ex. “p” for “paragraph”; “h1” for “header 1”
- To differentiate tags from plain text, we enclose them in brackets
  - <p>, <h1>
  - This *opens* a tag
  - To close a tag, do </h1>

# MORE ABOUT TAGS

---

Tags delineate content blocks

- `<h1> I'm inside a tag! </h1>`
  - Tags have “properties,” and these properties are then passed on to the content within the tags
  - Mr. `<tag>` says “abide by my laws until further notice”
  - Then Ms. `</tag>` says “further notice”
-

# TAGS CAN BE NESTED

---

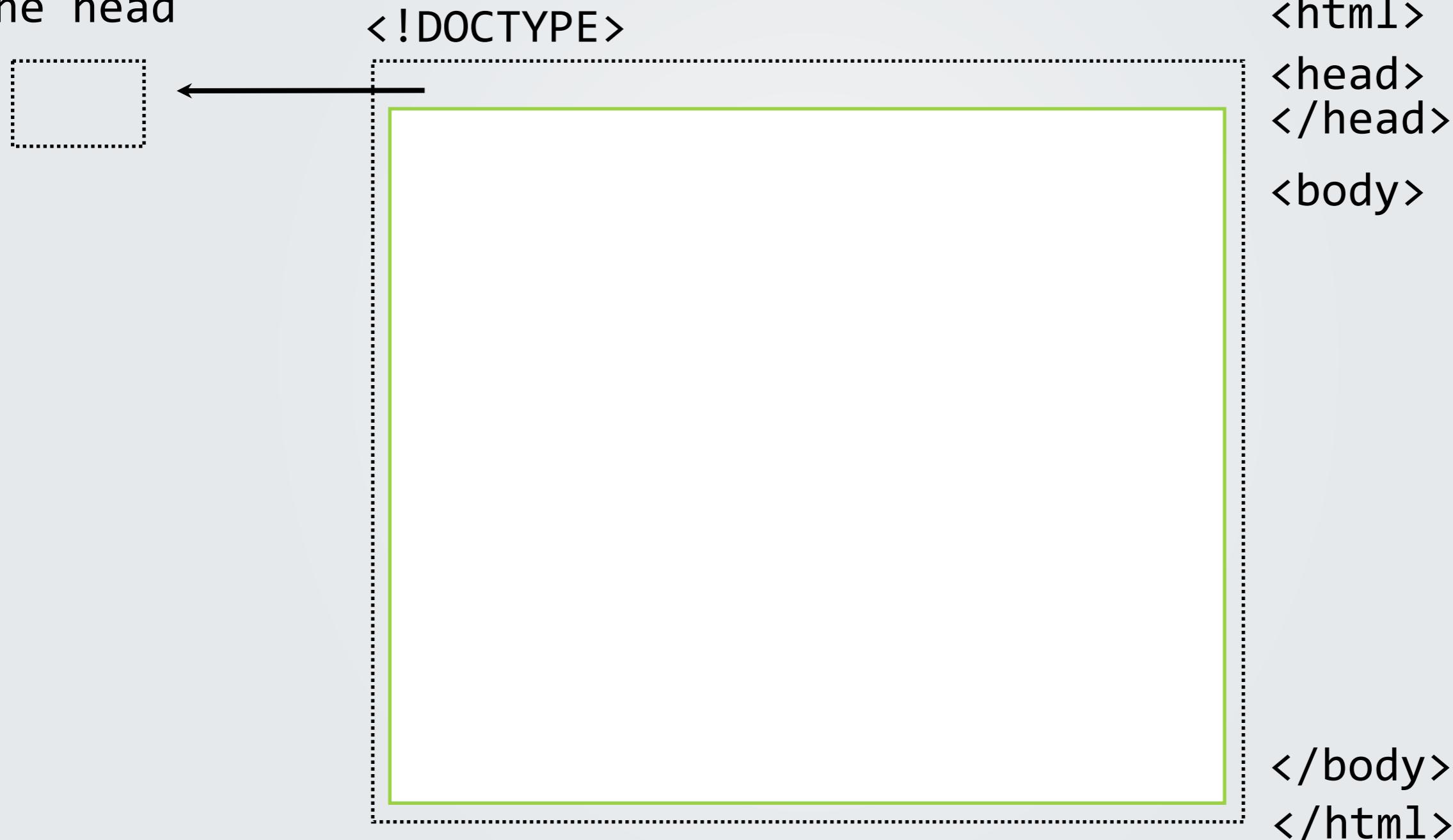
```
<div>  
  <h1>A sick header</h1>  
  
  <div>  
    <p>My sweet paragraph</p>  
  
  </div>  
</div>
```

---

This is how we can create boxes within boxes

# HTML DOCUMENT STRUCTURE

The head



# SOME COMMON TAGS

<div> (“division”) the content blocks of HTML (an empty shell)

- <div>This content will be in an div block!</div>

<a> (“anchor”) links

- <a href="<http://www.hackyale.com>">Home</a>

<img> images

- 

<p> paragraphs

- <p>This text will be in a nice paragraph</p>

# THE INTERNET

THE LONGEST LINE EVER

The Internet is really, really great...

[http://www.youtube.com/watch?  
v=7\\_LPdttKXPc](http://www.youtube.com/watch?v=7_LPdttKXPc)



THE VIDEO WAS GREAT

THANKS, AARON TITUS

HOWEVER, WE'D LIKE TO GET A BIT  
MORE TECHNICAL AND ADD THE  
CONTEXT OF WEB DEVELOPMENT

# TERMINOLOGY

## Client-server model

- Client == (you and your) browser
- Server == machine sending (or “serving”) you the data and files you request

HOST ~== “server”

- “to host” (code, files, applications) ~== “to serve”

LOCAL == hosted on the machine in question

REMOTE == hosted on a different machine



# REQUEST-RESPONSE CYCLE

(1) Client (browser) makes a “request”

- REQUEST == textual message whose syntax and semantics are defined by HyperText Transfer Protocol (*HTTP*)
- Think of a protocol as a “language”

(2) Server issues a “response”

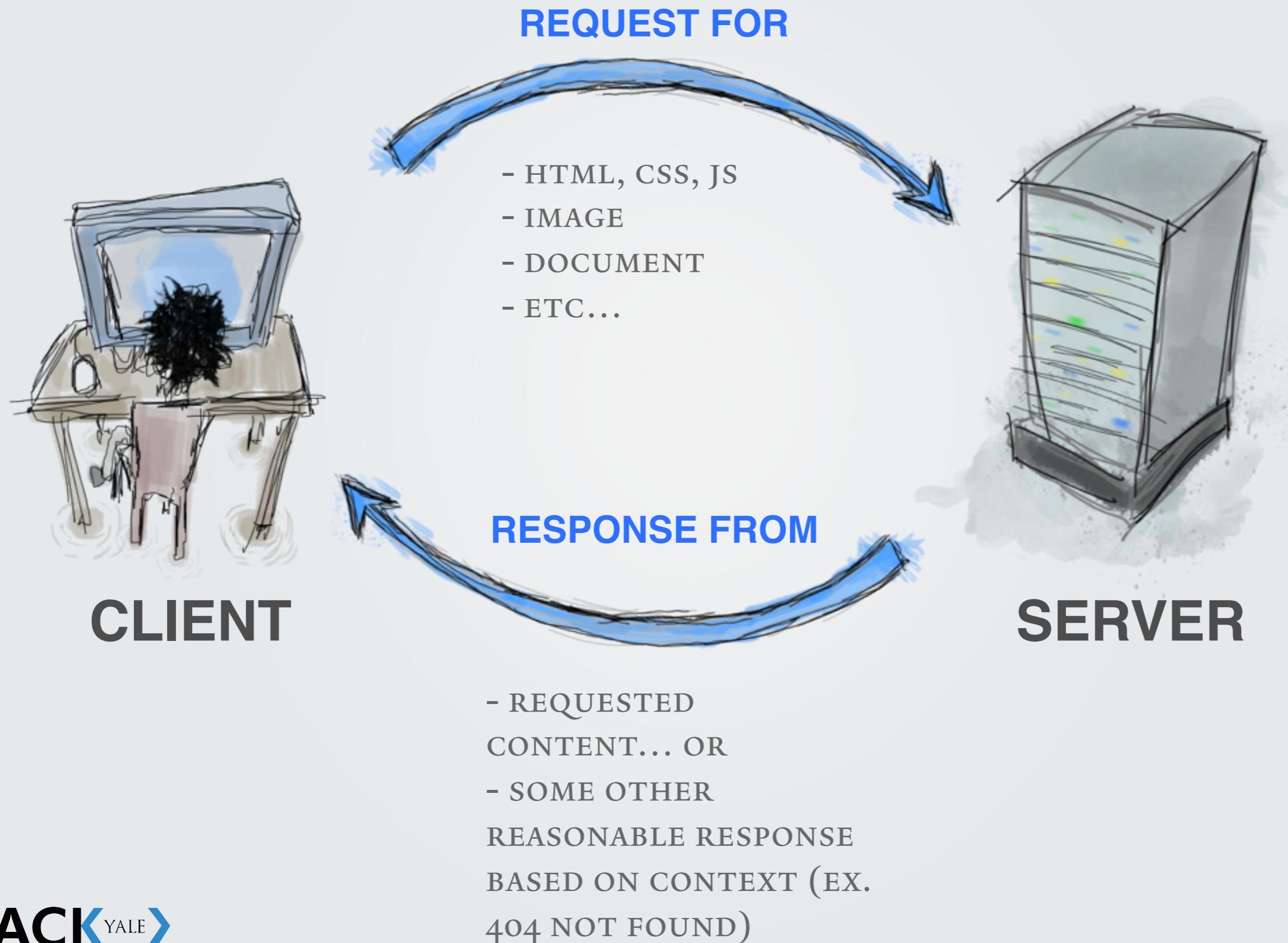
- RESPONSE == textual message defined by HTTP
- Contains status code. Ex: 404 (“*Not Found*”), 200 (“*Okay*”), 500 (“*Internal Server Error*”)

(3) Cycle repeats itself

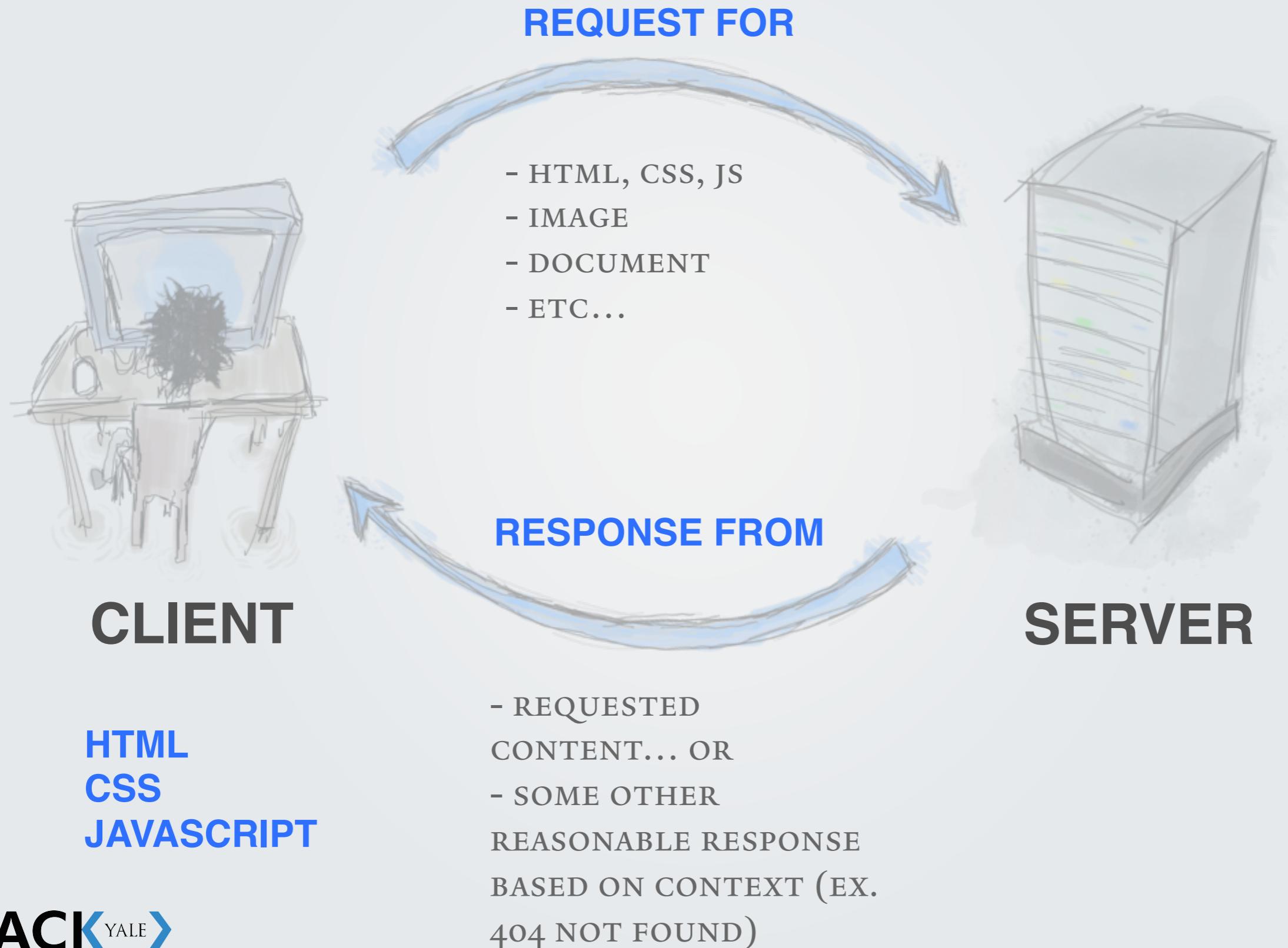


KEY TECHNOLOGIES  
WEB DEVELOPMENT 101

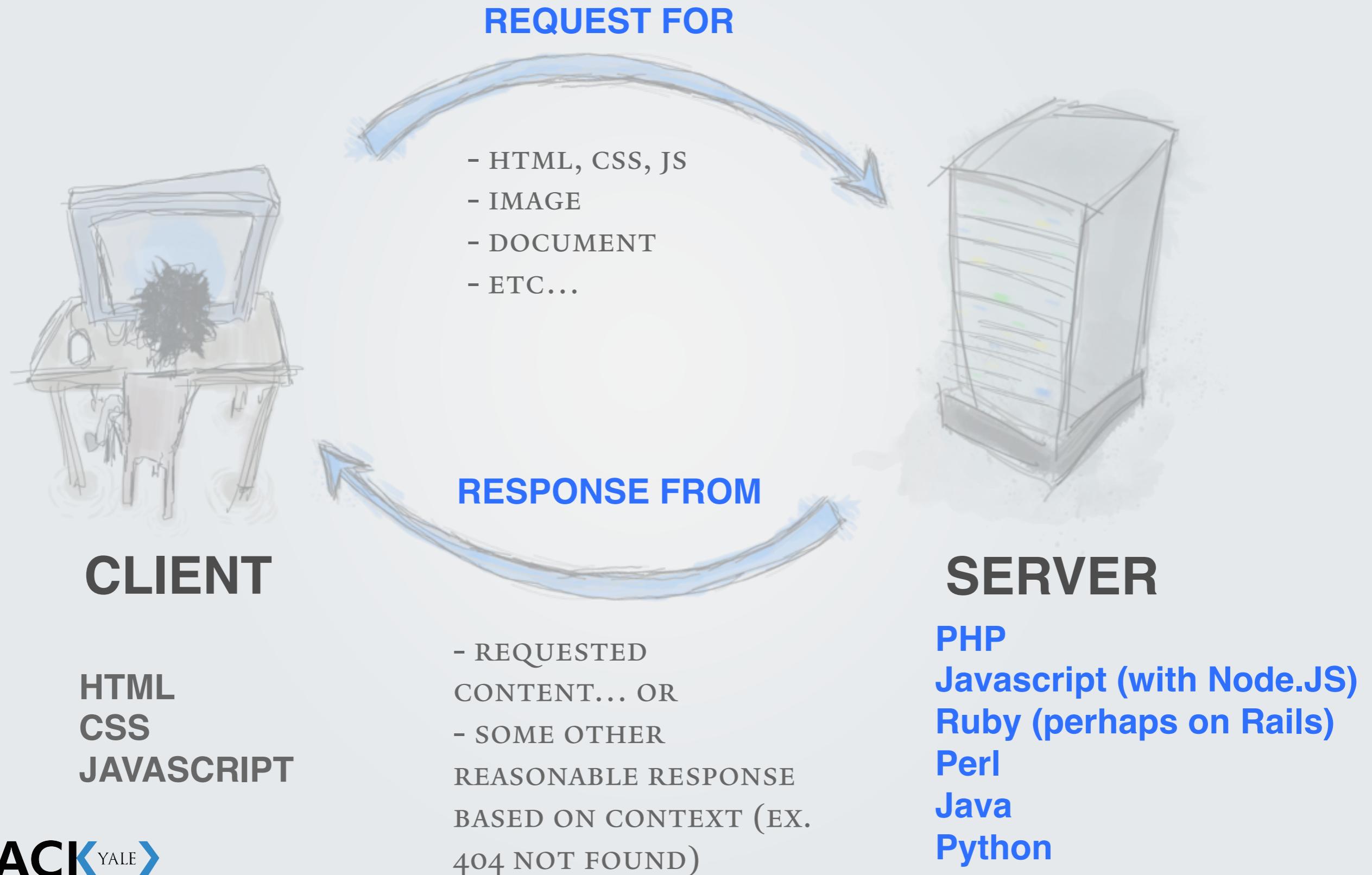
# THE CLIENT-SERVER MODEL



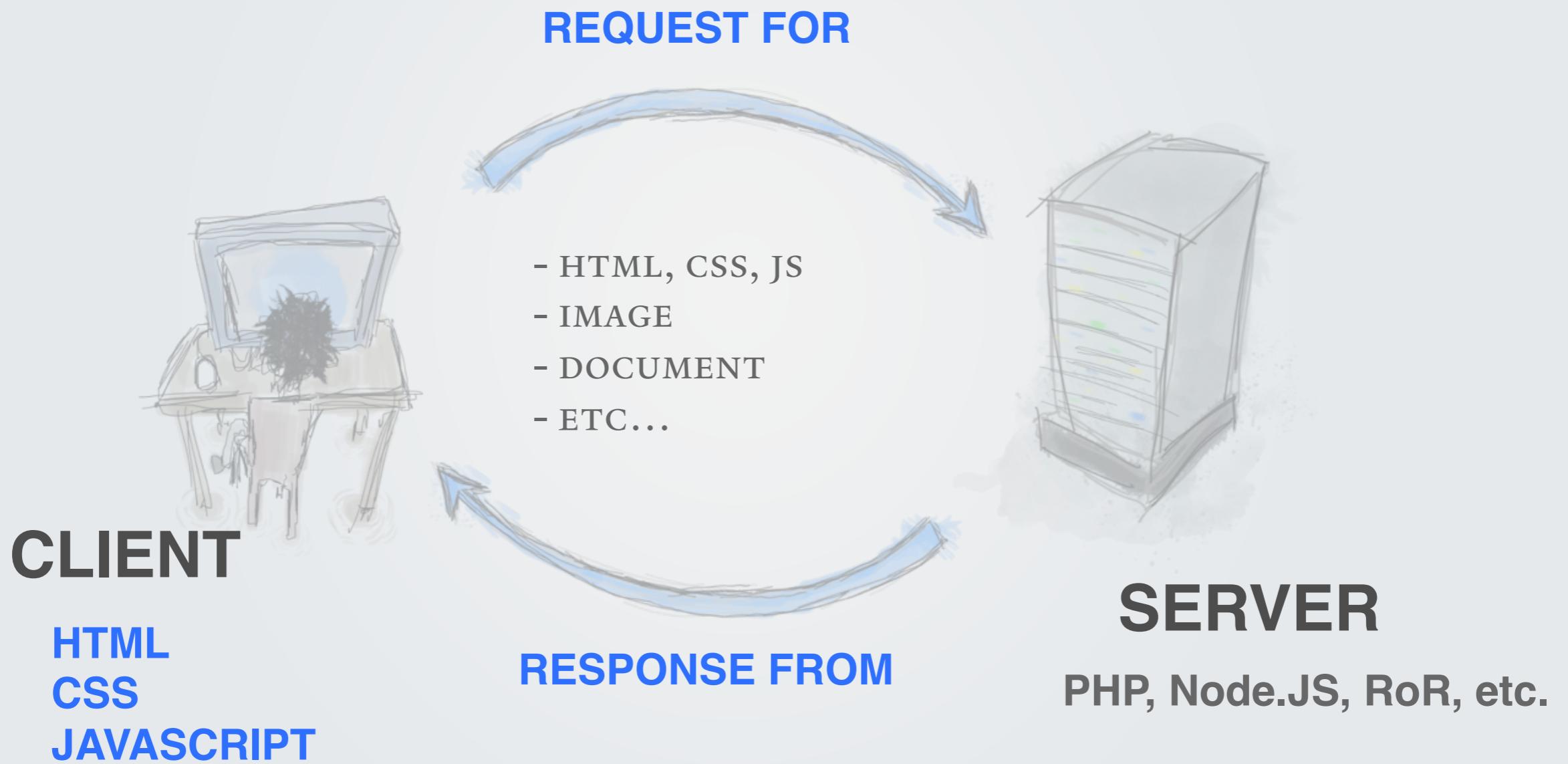
# ON THE CLIENT SIDE



# ON THE SERVER SIDE



# THE FRONT-END IS HOW TO DISPLAY CONTENT FROM THE SERVER



# NEXT TIME

1. More Cool Things you can do in HTML
2. How to Dress your Webpage in Style
3. More Concrete Examples and Exercises!

# ASSIGNMENT 0

Time to practice what you've learned today!

- Make a simple profile website (yep, linked pages!)
- 2 pages: a biography and a page of some of your favorite sites
- Don't worry about style – we will cover this in the next lecture.

Also, do:

- Learn Command Line the Hard Way (  
<http://cli.learncodethehardway.org/book/>)
- Practical Guide to HTML (<http://learn.shayhowe.com/>)

# SUBMITTING HOMEWORK

This course uses <http://lore.com> – there you will be able to find some resources and be able to submit assignments.

The Course Code is: [7J8B98](#)

You can also find all the assignment details, slides, and code samples on HackYale's GitHub wiki page:

<https://github.com/hackyale/HackYale/wiki>

# OFFICE HOURS

Dani Dickstein

1:00pm – 2:00pm Friday, CEID M105-106

Alex Reinking

3:30pm – 4:30pm Friday, CEID M105-106

We will also answer any questions by email,  
and can open more office hours on request.



QUESTIONS EVEN GOOGLE CAN'T ANSWER?

[TEAM@HACKYALE.COM](mailto:TEAM@HACKYALE.COM)

[WWW.HACKYALE.COM](http://WWW.HACKYALE.COM)