CSCI-4208: Developing Advanced Web Applications

Week 1: Lecture 1

Overview

- Syllabus
- Course Philosophy
- Web Application Overview

- **Unit 1: Frontend Development**
- Web Client: Browser - Crash Course
- HTML - Crash Course

Syllabus

Use Syllabus for this Section

Course Philosophy

What is CSCI-4208: Developing Advanced Web Applications?

An opinionated, hopefully frustration-free crash course in full stack app development

Course Philosophy

Opinionated:

- There are many ways to do things on the web: can't learn them all at once!
- We'll cover what I think you need to know to effectively get started making web apps

Hopefully frustration-free:

- We will go slowly through the essential concepts and speed through the obvious stuff
- You may fill in "obvious stuff" knowledge gaps via MDN or W3Schools

Course Philosophy

Goals:

If you never take another web programming class again, you will leave CSCI-4208 with the following skills:

- Create **attractive, responsive full stack web apps** that work on phones.
- Have the **vocabulary and background knowledge** to understand technical writing/discussions about the web (e.g. web API documentation; job interviews; random blog posts)
- Have the **foundation** to further pursue areas of full stack programming that you're interested in. Expand your knowledge by trying new tools and libraries

Course Philosophy

Non-Goals:

CSCI-4208 is **not** a class that will turn you into a senior frontend/backend developer.

- Nor is any class; software takes years of experience to develop expertise.

CSCI-4208 is **not** a class that will teach you all there is to know about web programming.

- For example, we will **not** teach how to support old browsers, legacy devices, etc.

Course Philosophy

Disclaimer:

CSCI-4208 is a newer offering, meaning:

- **Everything is subject to change.** Including everything I've just told you and everything I'm about to tell you.

- **There will be all the mistakes of a newer course!**
  - Bugs in homework
  - Awkward lectures
  - Things that are too hard / too easy

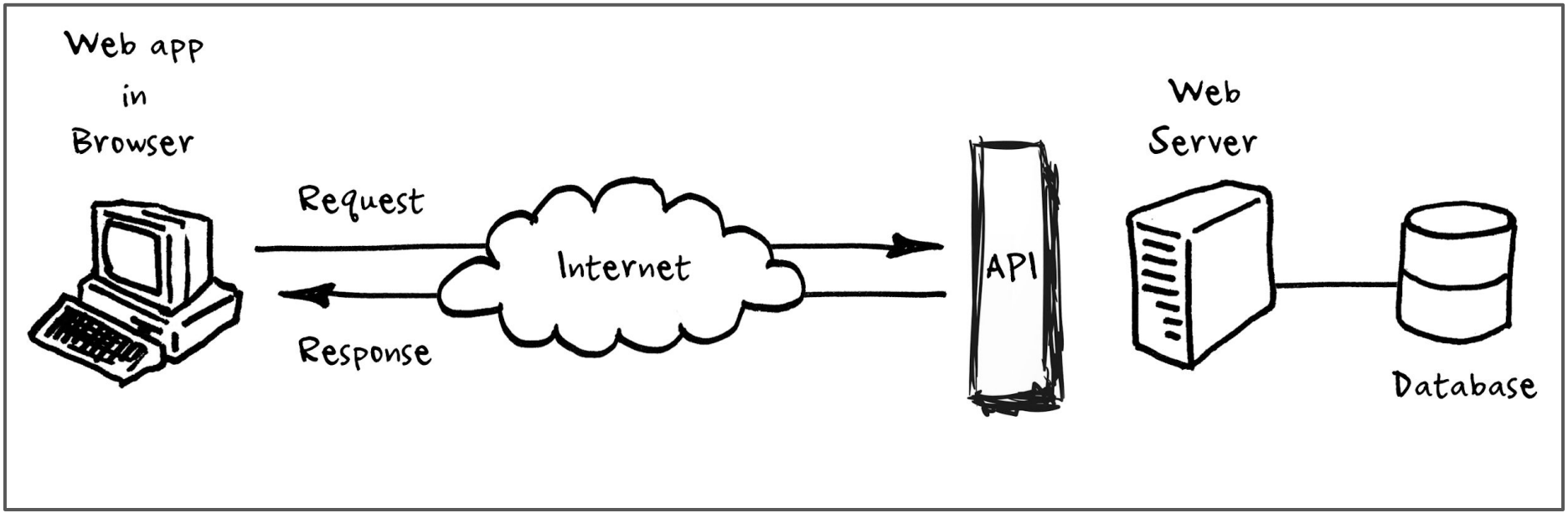
Please be patient with me! I'll also solicit your constructive feedback.

Course Philosophy

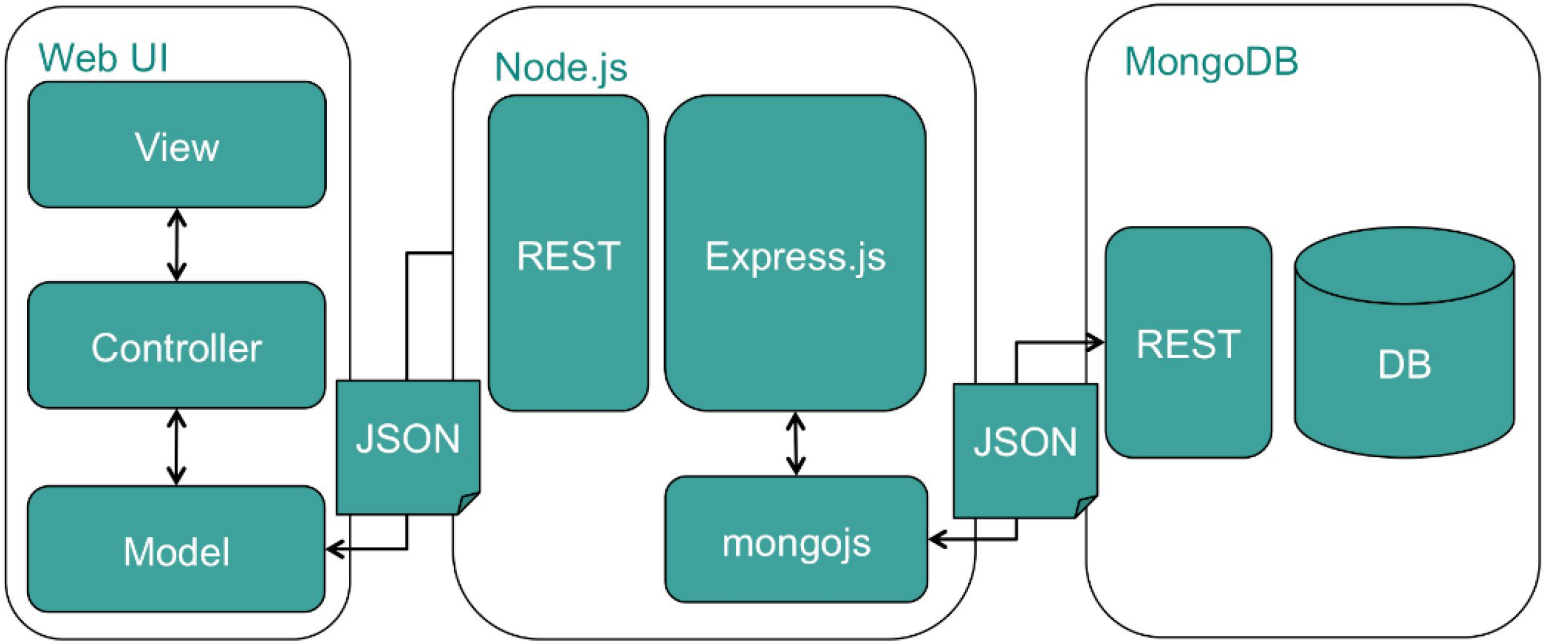
Honor Code:

- **DON’T** blindly copy other people's CSCI-4208 solutions.
- **OK** to look at other website's code for inspiration
- **OK** to look at StackOverflow / Google / etc for help
- **OK** to share a webpage you made in CSCI-4208 to show off the webpage itself

Web Application Overview



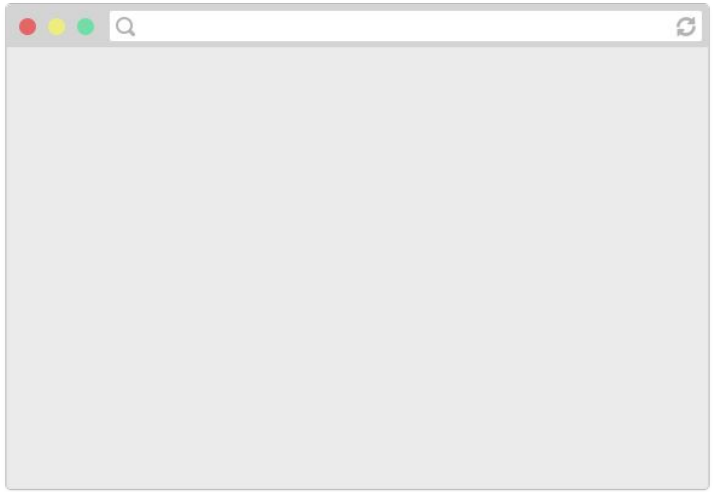
Web Application Overview



Unit 1: Frontend Development

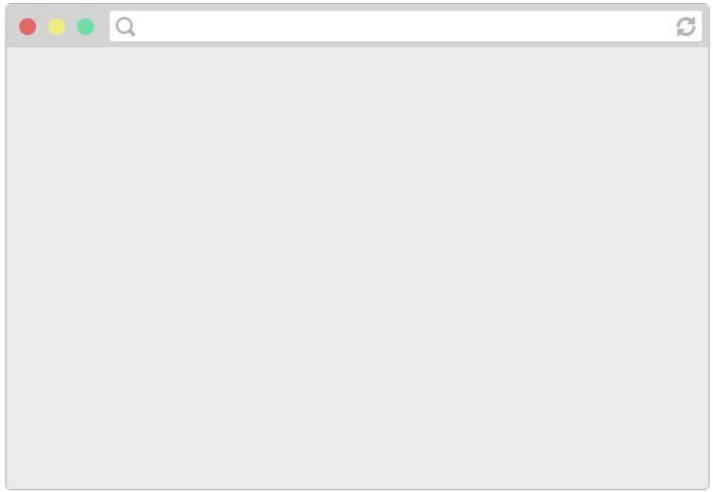
- Web Client: Browser - Crash Course
- HTML - Crash Course

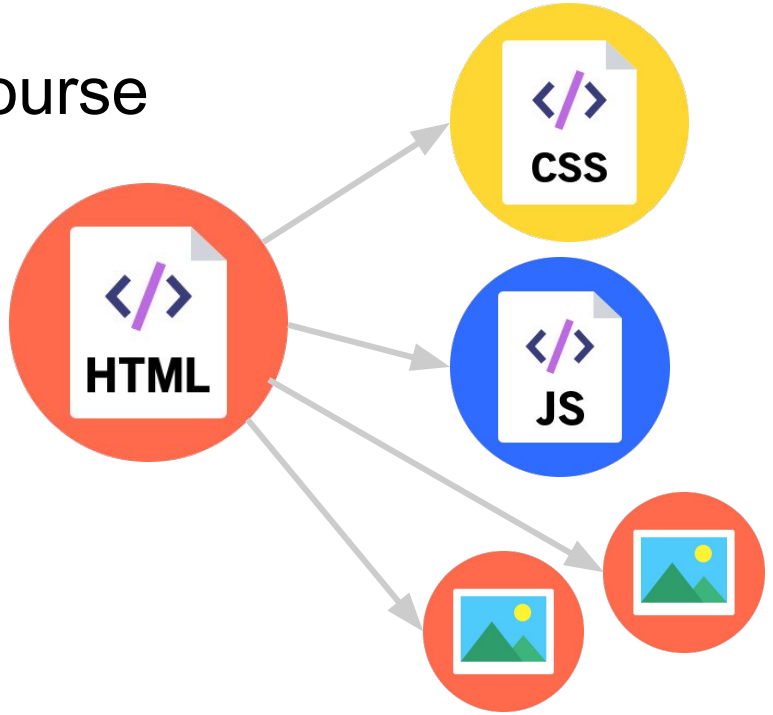
Web Clients: Browser - Crash Course



Web pages are written in a markup language called **HTML**, so browsers display a web page by reading and interpreting its HTML.

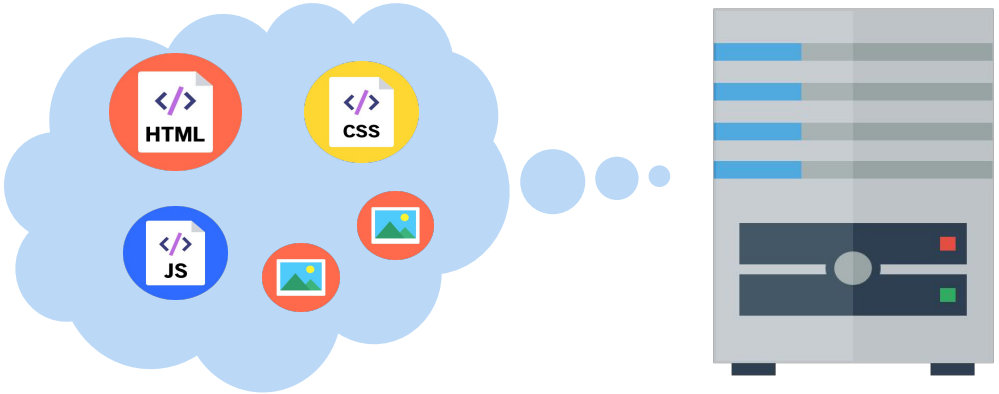
Web Clients: Browser - Crash Course





The HTML file might link to other resources, like images, videos, as well as **JavaScript** and **CSS** (stylesheet) files, which the browser then also loads.

Web Clients: Browser - Crash Course

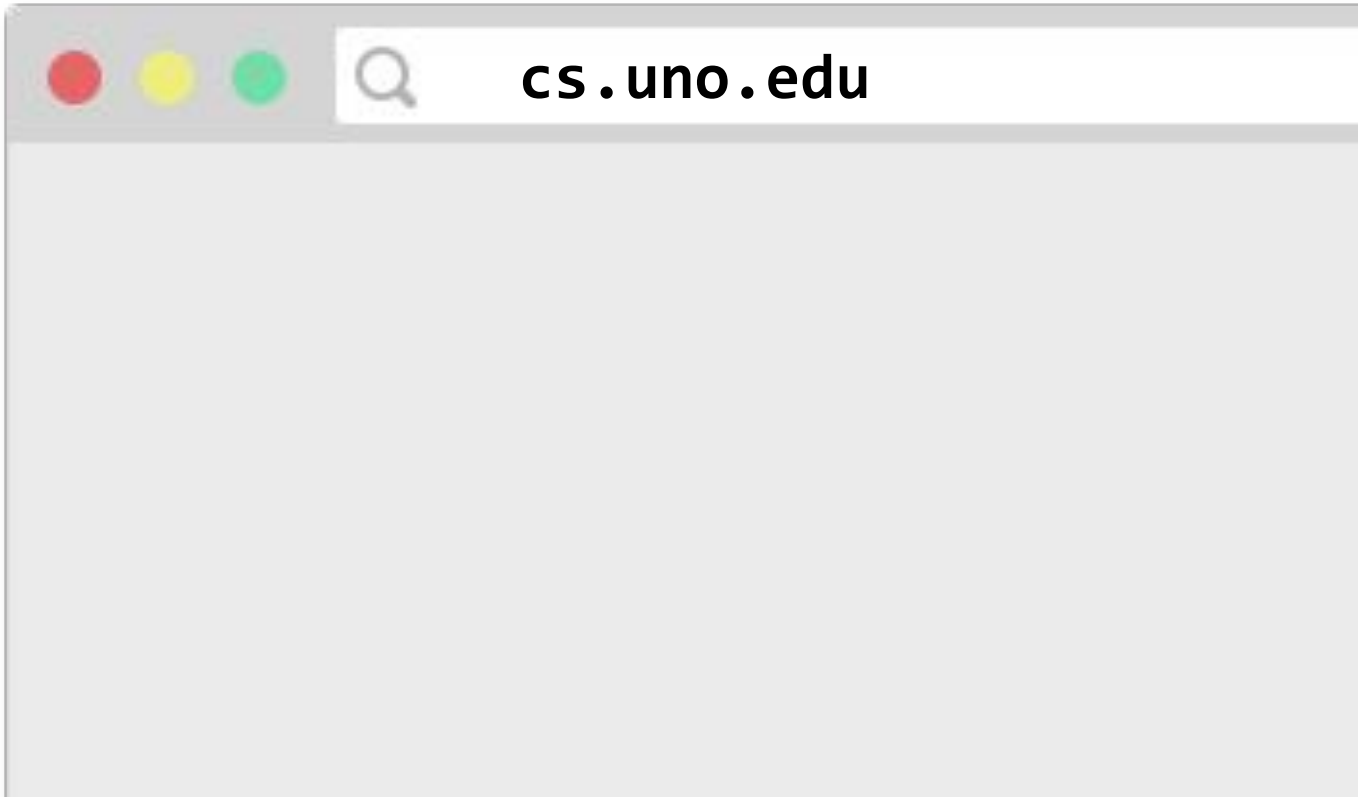


A **web server** is a program running on a computer that delivers web pages in response to requests.

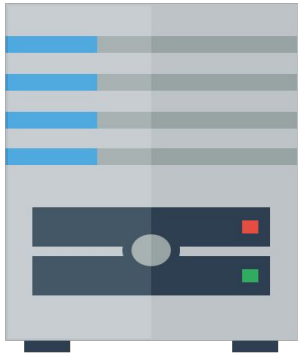
It either stores or generates the web page returned.

Web Clients: Browser - Crash Course

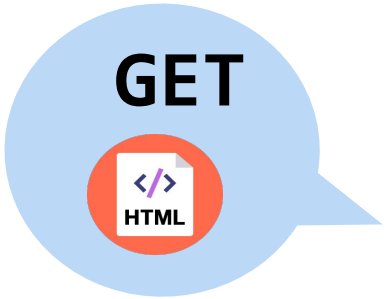
1. You type in a URL, which is the address of the HTML file on the internet.



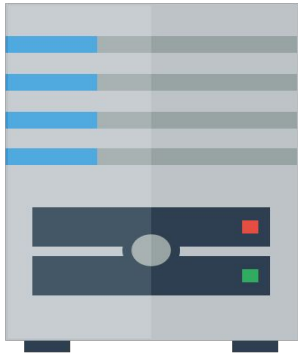
Web Clients: Browser - Crash Course



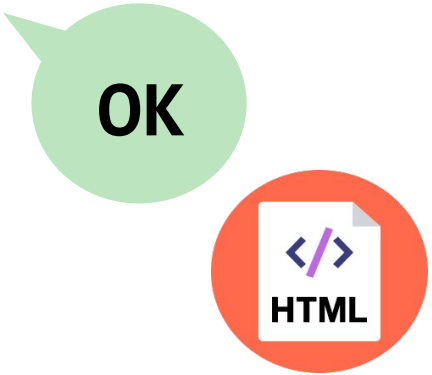
2. The browser asks the web server that hosts the document to send it.



Web Clients: Browser - Crash Course

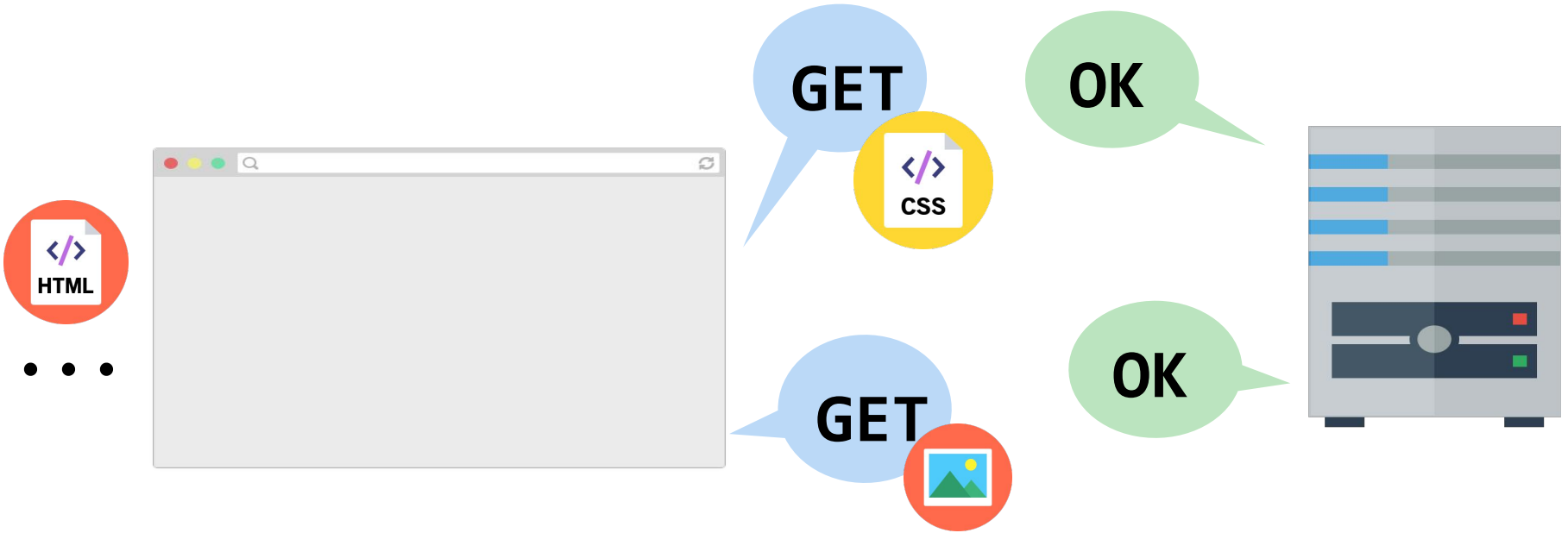


3. The web server responds to the browser with HTML file that was requested.



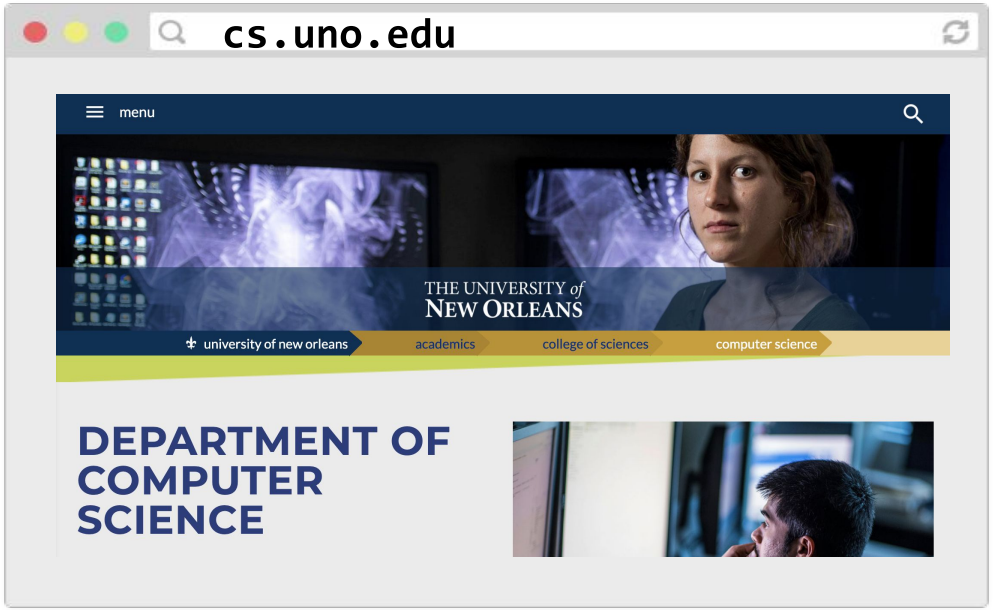
Web Clients: Browser - Crash Course

4. The browser reads the HTML, sees the embedded resources and asks the server for those as well.



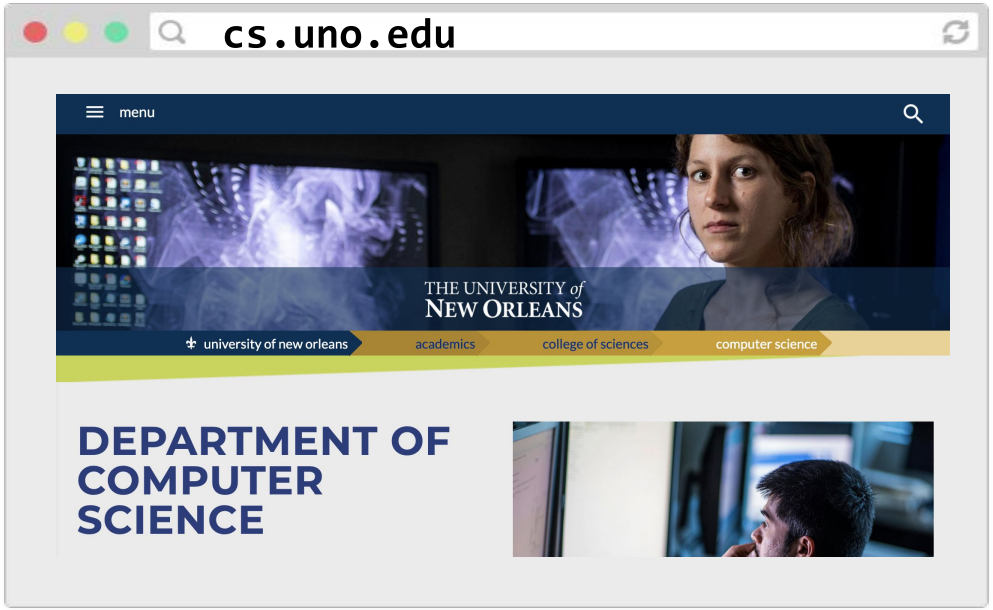
Web Clients: Browser - Crash Course

5. The web page is loaded when all the resources are fetched and displayed.



Web Clients: Browser - Crash Course

(That was obviously very hand-wavy. We'll get more detailed when we talk about servers later in the semester.)



Web Clients: Browser - Crash Course

- HTML Parser & Viewer
- CSS Parser
- HTTP, File, Websocket communications
- JavaScript Runtime Environment
- Browser API: Local Storage, Cookies, Canvas, etc.

Browsers are Write once, Run Everywhere.
Similar to JVM, but even better and more ubiquitous.
Because, it's not only portable logic but also portable GUI too!

HTML - Crash Course

- What is HTML
- Basic HTML Page Structure
- HTML elements
- HTML attributes
- Block-level vs Inline-level
- Live coding

HTML - Crash Course

What is HTML?

HTML (**H**yper**t**ext **M**arkup **L**anguage)

- Describes the **content** and **structure** of a web page; not a programming language.
- Made up of building blocks called **elements**.

```
<p>
  HTML is <em>awesome!!!</em>
  <img src="puppy.png" >
</p>
```

HTML - Crash Course

Basic HTML page structure

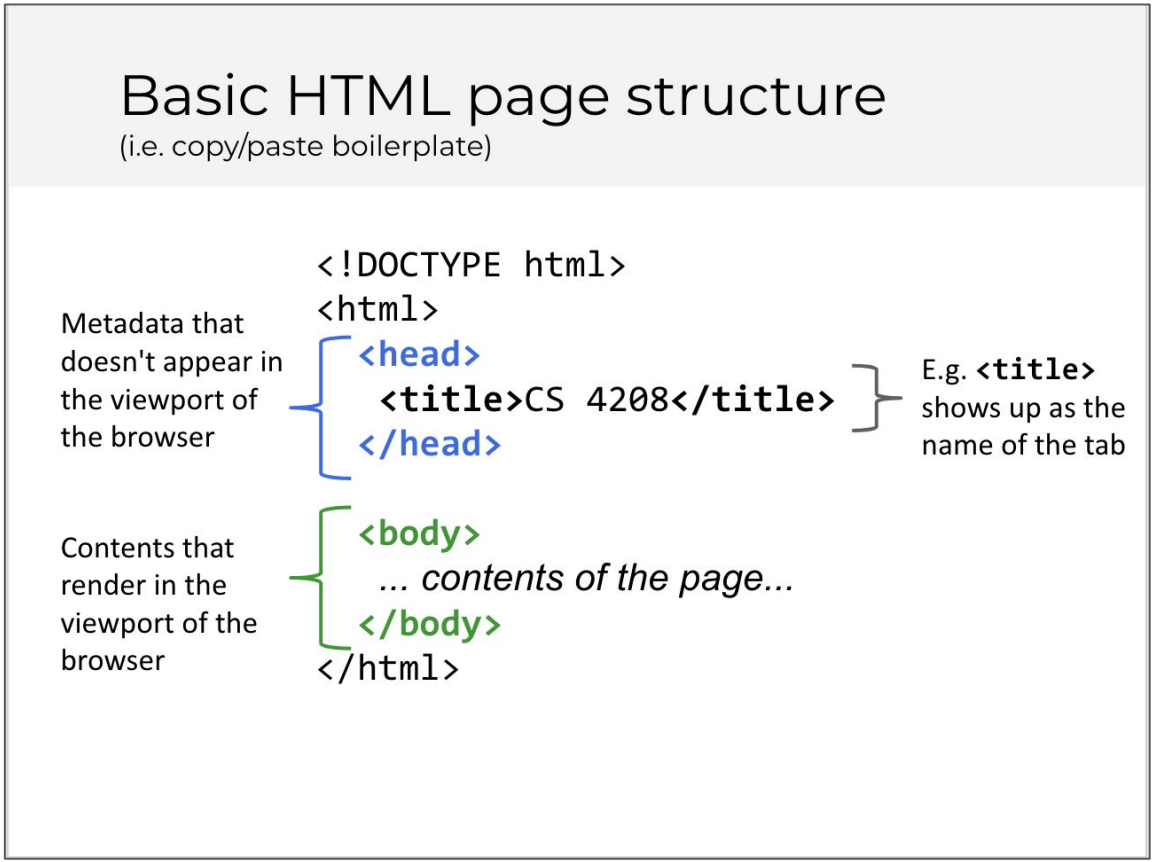
(i.e. copy/paste boilerplate)

```
<!DOCTYPE html>
<html>
  <head>
   <title>CS 4208</title>
  </head>

  <body>
   ... contents of the page...
  </body>
</html>
```

Saved in a *filename*`.html` file.

HTML - Crash Course



HTML - Crash Course

HTML elements

```
<p>
  HTML is <em>awesome!!!</em>
  <img src="puppy.png" />
</p>
```

- An element usually has start and ending tags (** ** and ** **)
  - **content**: stuff in between start and end tags
- An element can be self-closing (**img**)
- An element can have attributes (**src="puppy.jpg"**)
- Elements can contain other elements (**p** contains **em** and **img**)

HTML - Crash Course

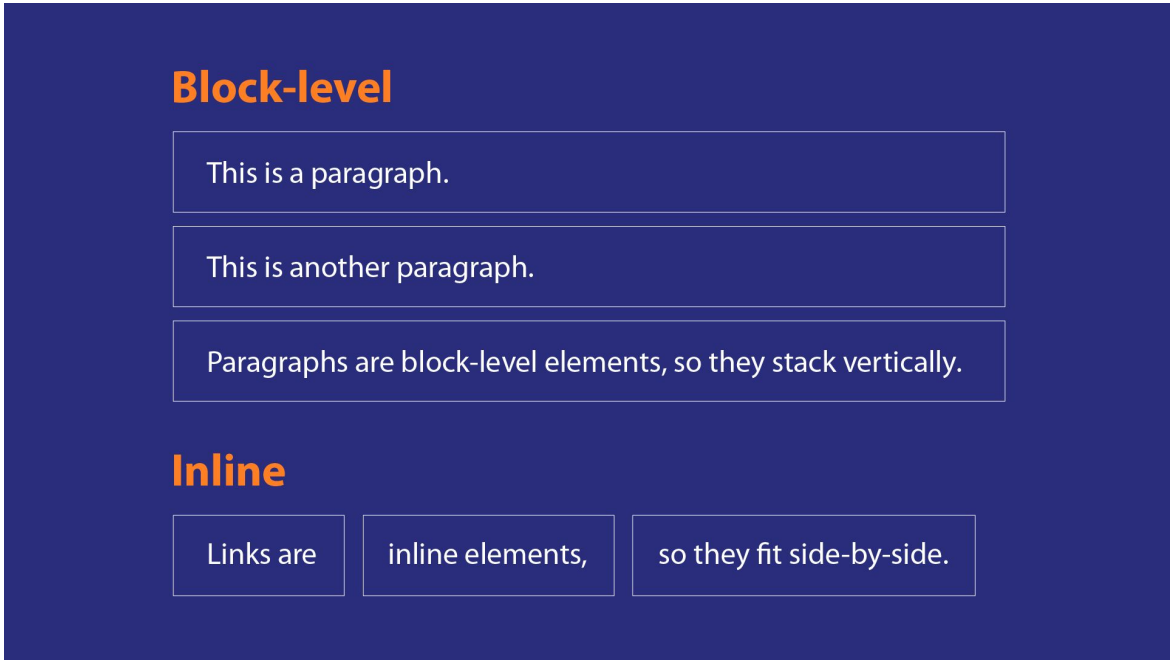
Some HTML elements

(to place within **<body>**)

Top-level heading **h1, h2, … h6**	** **Moby Dick** **
Paragraph	** **Call me Ishmael.** **
Line break	since feeling is first** ** who pays any attention
Image	**<img src="cover.png" >**
Link	****click here!****
Strong (bold)	****Be BOLD****
Emphasis (italic)	He's my ****brother**** and all

HTML - Crash Course

Block-level vs Inline-level



The End.

Let's write some HTML

Coming Soon... to a Moodle near you!

Lab 1	HTML 'Interactive Story' Lab
Homework 1	Design a *'Interactive Story'* with HTML