

The Web: HTML, CSS, and more!

TF: Larry Zhang

HBS CS50 Spring 2016

Friday, April 15, 2016, 4-5 PM – HBS Hawes 201

Agenda

- ▶ HTML
- ▶ CSS
- ▶ Bootstrap
- ▶ And more!

HTML: What is it?

- ▶ Everything you see on the internet through your web browser
- ▶ Stands for Hypertext Markup Language
- ▶ It's markup, i.e. it describes the webpage! It doesn't construct it
- ▶ HTML describes semantic documents, it's not a programming language



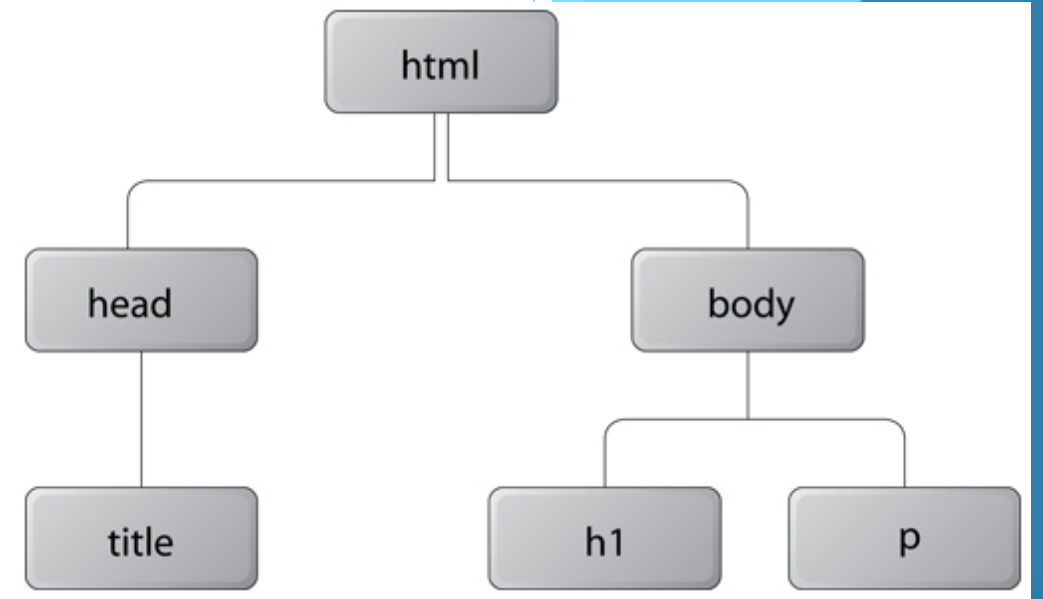
Google

Google Search

I'm Feeling Lucky

HTML: How to use it

- ▶ HTML is inherently a tree, known as the Document Object Model (DOM)
- ▶ Structure is defined by blocks, or tags
 - ▶ Content goes in between
 - ▶ `<div>Look I'm a webpage</div>`
- ▶ You can nest tags to form “branches” on the tree
- ▶ Include text, pictures, video!
- ▶ You can comment HTML like you can in C, except you need `<!-- -->` instead
- ▶ HTML is smart! You can mess it up e.g. forget to include a closing tag, and HTML will still often work as intended



HTML: Tag types

- ▶ `<html>`: Begins and ends the HTML document
 - ▶ `<head>`, `<body>`, `<footer>`: The three main divisions of the page!
 - ▶ `<head>`: Put scripts, metadata, and your title here
 - ▶ `<body>`: Put your page content here
 - ▶ `<footer>`: Put more scripts, metadata here
-

- ▶ `<div>`: Divides the page into sections
-

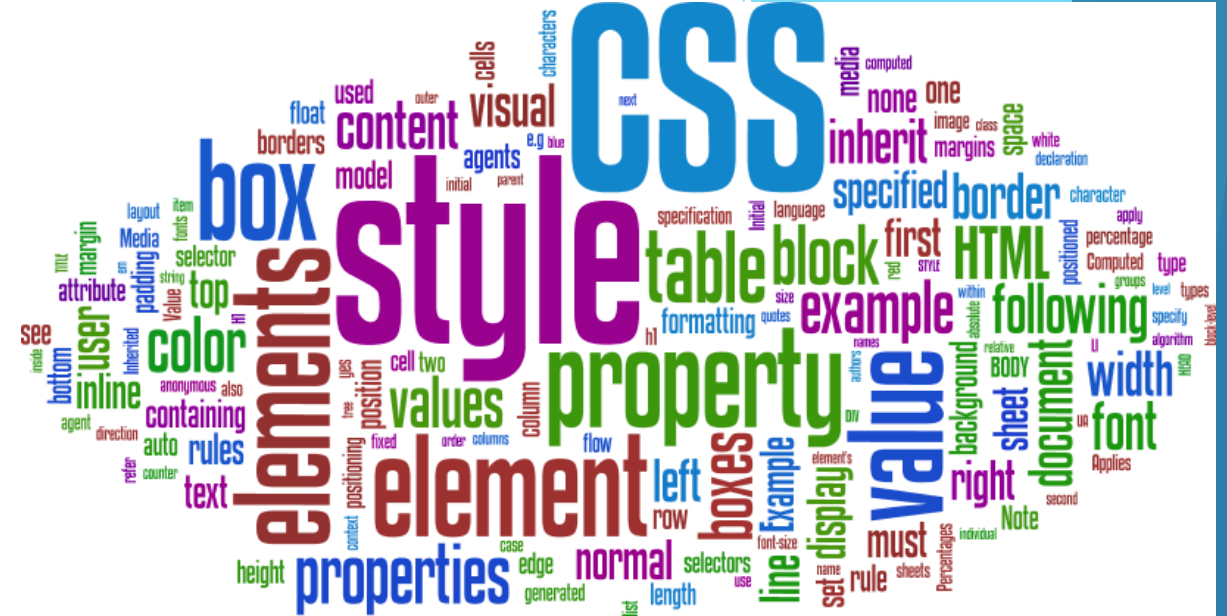
- ▶ `<h1>`, `<h2>`, ..., `<h6>`: Headers! h1 is **HUGE**, h6 is small

- ▶ ``: includes images!
- ▶ `<p>`: Text (lit. paragraph!)
- ▶ `</html>`
- ▶ And more!



Next!

CSS: What is it?



- ▶ If HTML is the doc...
- ▶ *CSS is what makes it pretty.*
- ▶ CSS = Cascading stylesheets
- ▶ You can use lots of stylesheets at the same time! Styles with highest priority cascade and override the others => cascading!
- ▶ Priorities:
 - ▶ 1. *Inline*: You can define styles inside the block declaration! E.g. `<div style=".....">`
 - ▶ 2. *Blocked*: You can define styles using `<style>` (preferably inside `<head>`)! E.g. `<style>...</style>`
 - ▶ 3. *External file*: You can also just put them into an external stylesheet file (preferred)
 - ▶ !important – you can add this to any line of the stylesheet to make that take top priority (subject to the three above when multiple !important's come into conflict)

CSS: How to use it

- ▶ Styles consist of three things:

- ▶ [selector] {
 [property]: [value],

}

- ▶ You can select things in a variety of ways:

- ▶ Block type: you can select specific block types, like **h1**!
- ▶ Classes: You can assign a **class** to a block, like so: <div class="sector"></div>
 - ▶ Essentially grouping certain blocks to be similar!
- ▶ IDs: You can assign an **id** to a block, like so: <div id="unique"></div>
 - ▶ Identify a specific block



CSS: What you can do

Change:	Example declaration
text color	color: blue;
font size	font-size: 20px;
width, height	width: 400px;
text underline/bolding/italicizing	font-style: bold;
background color/image	background-image: url('/bg.jpg');
position	left: 50%;
position type	position: fixed;
margin/padding	margin: 0px 10px 0px 10px;

and more!

This can get a little
tedious though...

Enter: Bootstrap!

- ▶ Originated as a Twitter framework for unifying design aesthetics internally
- ▶ Now, it's used in millions of websites to make constructing them much easier
- ▶ Still being improved! Bootstrap 4 was just released last year!
- ▶ Bootstrap: a free, open-source front-end library for creating websites and web apps! What this means:
 - ▶ Free: duh.
 - ▶ Open-source: the code of the library itself is easily accessible and modified
 - ▶ Front-end: for constructing user-viewed webpages, as opposed to organizing them (more on this at the end of seminar if we wish)
 - ▶ Basically, this takes all the work out of styling websites (and actually does much more as well)



Bootstrap: Capabilities

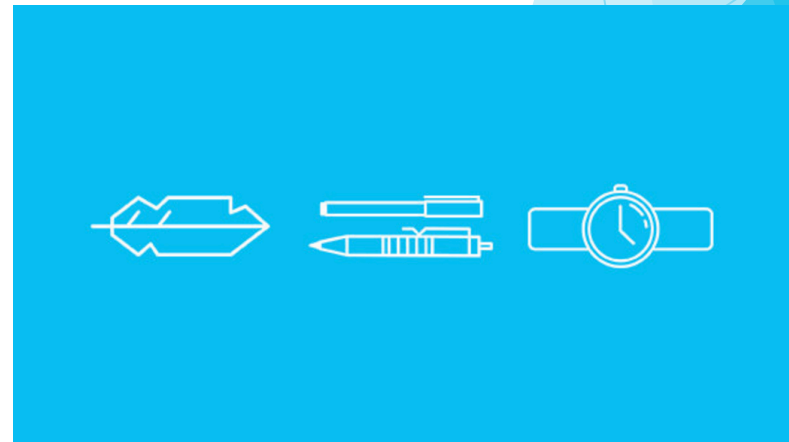
- ▶ Page organization by columns
- ▶ Menus and navigation lists
- ▶ Advanced buttons and button features
- ▶ Pagination (separating data into chunks for viewing)
- ▶ Labels and forms
- ▶ Various typographic improvements
- ▶ Progress bars
- ▶ Warning messages
- ▶ And more!

Bootstrap Examples

<http://expo.getbootstrap.com/>

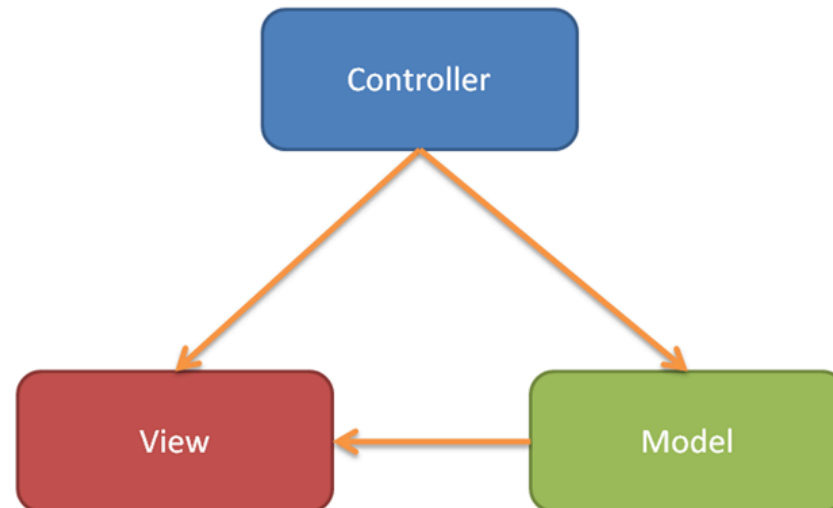
More Front-End Libraries

- ▶ Foundation 5
- ▶ InK
- ▶ Pure
- ▶ YUI
- ▶ Skeleton
- ▶ Semantic UI
- ▶ Toast



Frameworks

- ▶ More extensive than libraries, and usually also support back-end
- ▶ Front-end = HTML, CSS and Js
- ▶ Back-end = SQL, PHP, etc.
- ▶ Back-end allows us to dynamically change what shows up in our website using data and the MVC architecture!



Framework Examples

Grouped by language

Ruby

Ruby on Rails



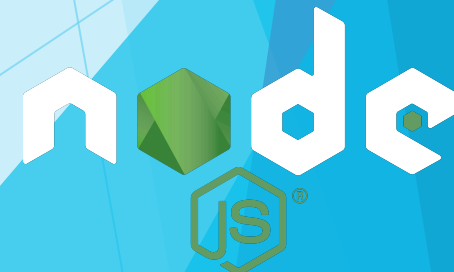
Javascript

Meteor.js
Angular.js
Node.js



Python

Django
Flask



Frameworks: Capabilities

- ▶ In most of these frameworks, a *single command in your terminal* is enough to get you a complete website!
 - ▶ Data models ready
 - ▶ MVC architecture implemented
 - ▶ Oftentimes, user accounts are also done or take no more than another line of code
- ▶ Meteor.js: `meteor create [appname]`
- ▶ Ruby on Rails: `rails generate app [appname]`
- ▶ Django: `django-admin startproject [appname]`

Thank you!!