Intro to Web Development

Upsilon Pi Epsilon
UConn Chapter

What’s the point?

Web Development is:

- Ubiquitous
- Popular
- Powerful
- Versatile
- Marketable

Scope

What we’ll cover today:

HTML
Java-
Script
CSS

Structure

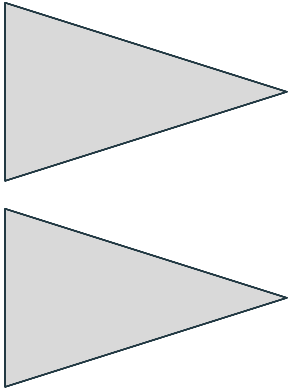
- **HTML**

- **CSS**

- **JavaScript**

Advanced Structure (not covered today)

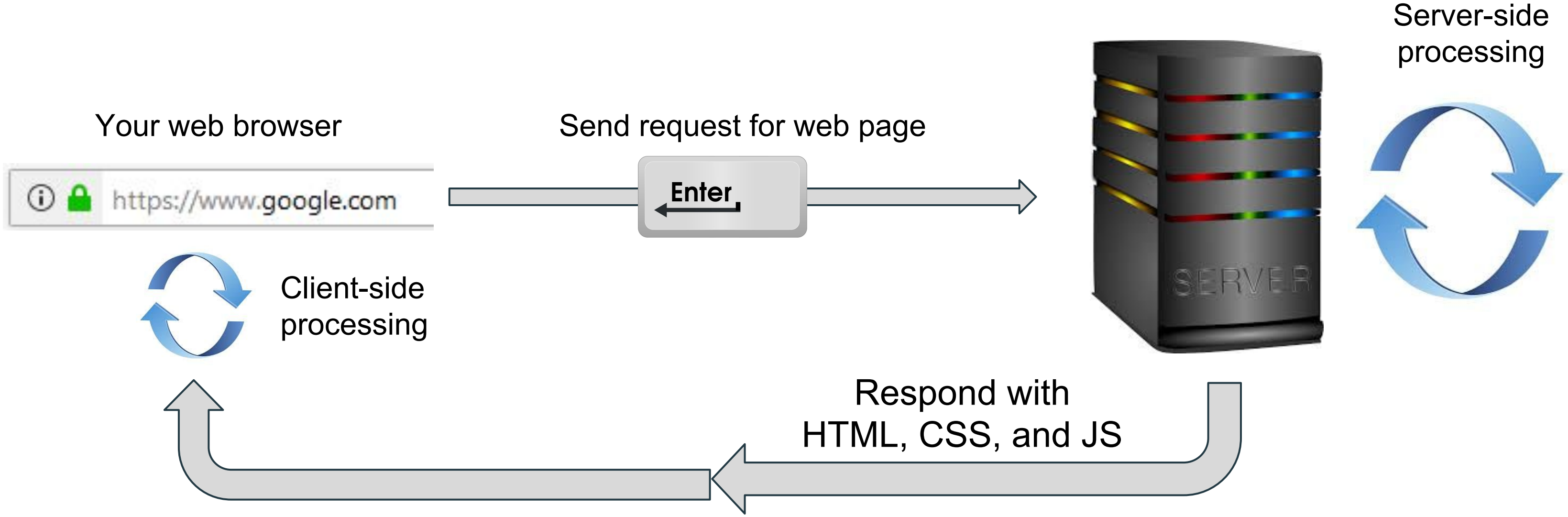
- Front end frameworks
  - AngularJS
  - React
- Back end frameworks
  - Flask
  - Django
- **Many** more



Allows you to have a more dynamic user experience

Allows you to have more core functionality, use different languages and their libraries

What happens when you load a website?



Your web browser

Send request for web page

Server-side processing

Client-side processing

Respond with HTML, CSS, and JS

How do I run a website without a server?

- It’s trivial to start a server on your computer to host a site for testing
- However, this won’t be necessary today
  - We will not be using any back end processing
  - Any modern web browser understands HTML, CSS, and JS
  - You can open an HTML file directly with your browser if only using these languages

Let’s get started!

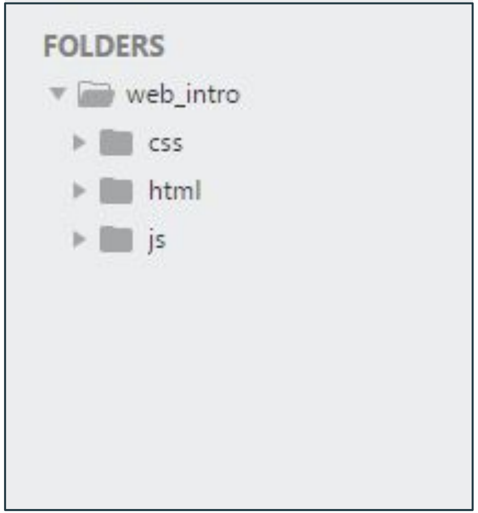
Get a good editor

- To make a website, all you need is a good text editor
- Unless you have one you strongly prefer:
  - Download Sublime for the purposes of this workshop
  - https://www.sublimetext.com/
- It will recognize the language of different files and offer specialized highlighting for each one

Setup project structure

- Make a ‘web_intro’ folder
- In that, make ‘html’, ‘css’, and ‘js’ directories
- Create:
    - ‘index.html’ in html
    - ‘global.css’ in css
    - ’functions.js’ in js

(These names aren’t special)

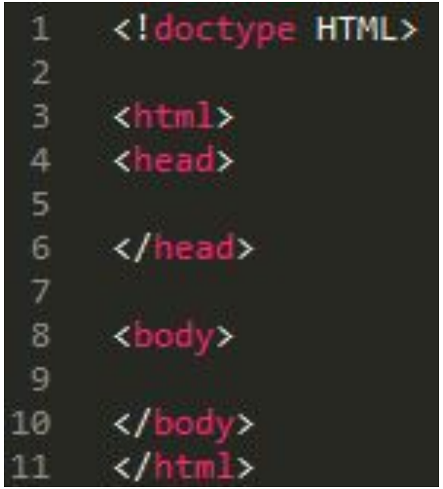


Part 1: HTML - Hypertext Markup Language

Html provides the structure (“bones”) for the application

Head, Body, and Elements

- These are the most basic components of a webpage
- In html, every element is enclosed in tags.
  - Elements start with <NAME> and end with </NAME>
  - Some can self close
- First two lines tells your browser to read html
- Head and body are 2 special elements
  - Every page has 1 head and 1 body
- Body: Everything shown to the user
- Head: Metadata (title, scripts, fonts, etc)



```html
<!doctype HTML>

<html>
<head>

</head>

<body>

</body>
</html>
```

Important tags (click for examples)

- <head>
- <body>
- <div>
  - Most basic container element
  - Think of it as a box that you should put everything else inside
  - Use liberally
- <p>
  - Paragraph
  - Common option to hold basic text
- <h1>, <h2>,..., <h6>
  - Headers (big text) from biggest to smallest
- <img>
- <a>
  - Used for hyperlinks
- <ul>
  - Creates a list of elements
  - <li> creates each element in the list
- <table>
  - Creates a table of items
  - <tr> creates each row in the table
  - <th> creates each header cell
  - <td> creates each datum cell
- <button>
  - Creates a button
  - Used often with an onclick() attribute

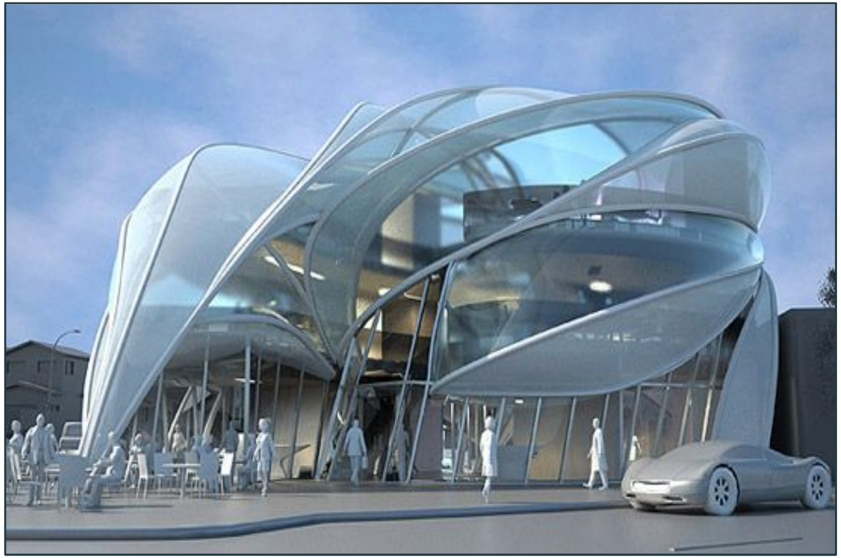
Part 2: CSS - Cascading Style Sheets

CSS makes things look

~~the way you want them to~~

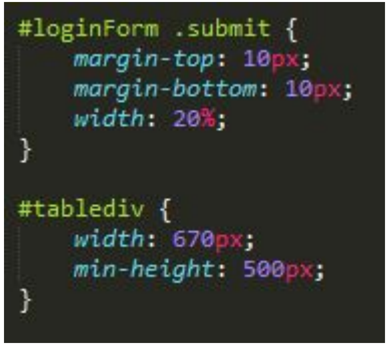
different

good



Format

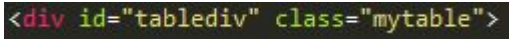
- The basics of CSS are very simple
  - Refer to an element in HTML by tag, ID, class name, etc.
  - Apply a property with *name: value*



```
#loginForm .submit {
    margin-top: 10px;
    margin-bottom: 10px;
    width: 20%;
}

#tablediv {
    width: 670px;
    min-height: 500px;
}
```

- Example of identifying an element:



```
<div id="tablediv" class="mytable">
```

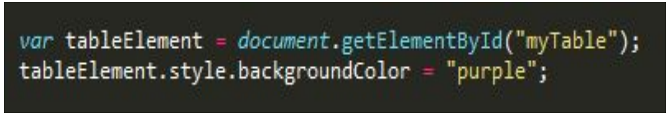
  - You could refer to this using these selectors:
    - div (would also apply the rules to all other divs)
    - #tablediv
    - .mytable (would also apply the rules to all other elements with class “mytable”)
- IDs are unique to one element, classes can consist of multiple elements

Part 3: JavaScript

JavaScript makes things do things

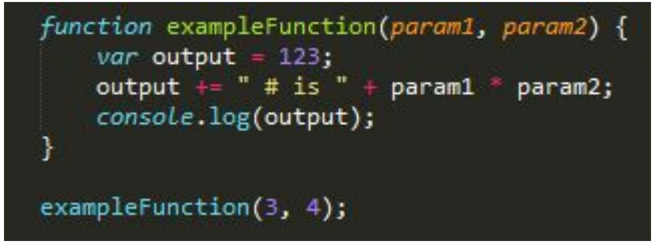
Details

- Natively understood by web browsers
- Usable to put functionality on the webpage
  - The user’s browser assumes the burden of executing the code
  - This is why JavaScript plugins can slow down your computer
- Can directly modify elements in the DOM (compiled HTML, essentially)



```
var tableElement = document.getElementById("myTable");
tableElement.style.backgroundColor = "purple";
```

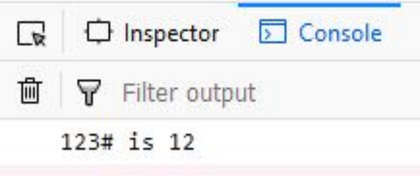
Weakly typed



```
function exampleFunction(param1, param2) {
    var output = 123;
    output += " # is " + param1 * param2;
    console.log(output);
}

exampleFunction(3, 4);
```

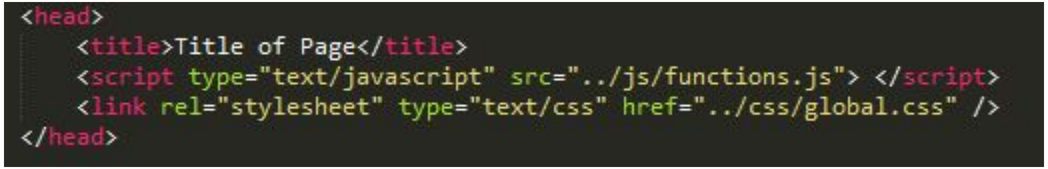
Prints to the browser console



```
123# is 12
```

How to combine HTML/JS/CSS?!

Very easily!



```
<head>
    <title>Title of Page</title>
    <script type="text/javascript" src="../js/functions.js"> </script>
    <link rel="stylesheet" type="text/css" href="../css/global.css" />
</head>
```

You can also write JS and CSS right in the HTML with the correct tags, but it is cleaner to keep them separate

Hands-on time!

- Do your best to recreate the example website that is about to be on the screen
- Google is your best friend! Especially for things like:
  - HTML elements
  - CSS selectors
  - CSS properties/values
  - JS examples
- Ask us any questions
- Refer back to this presentation