Stand Up!

Project Post-Mortem

Check-in Time

- What did you learn from projects?
- What was different than you expected?
- What are some benefits of TDD in practice?
- What topics do you understand but you feel like you struggle to use?
- What topics are you still struggling with conceptually?

Breathe

Stay Seated & Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes. We’ll start back in 10 minutes.

Intro to Front-End

Intro to Front-End

GETTING DATA TO USERS

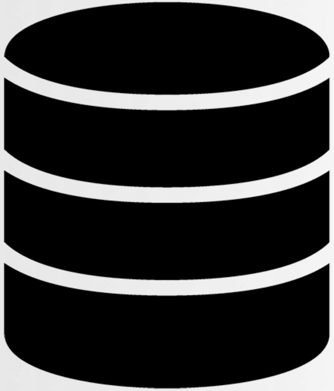
Notebooks Ready? It’s time for a mini lecture.

This might look familiar...

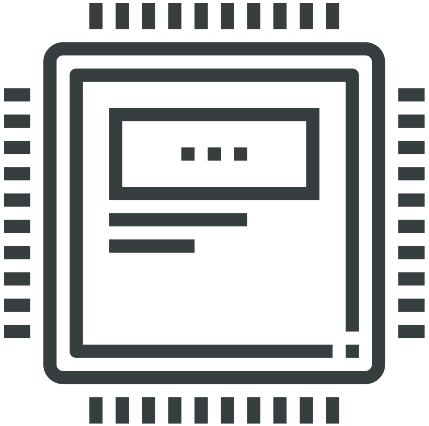
How the Web Works

ANATOMY OF A WEB APPLICATION

Web applications are composed of 3 major components



Database



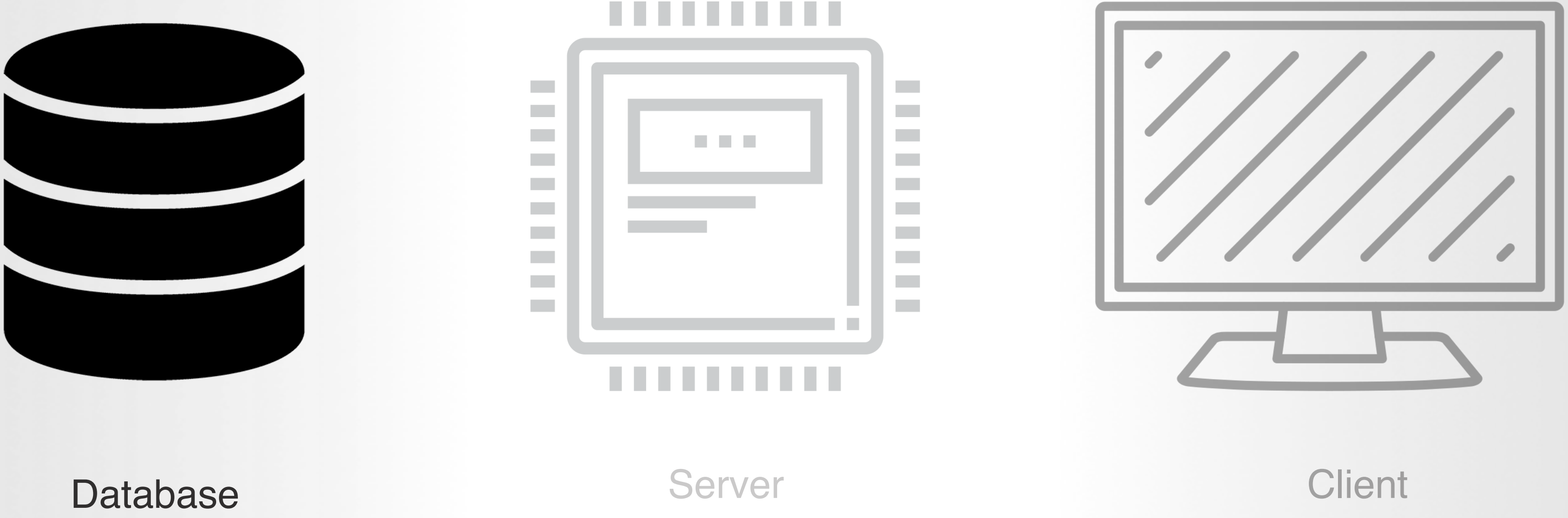
Server

Client

How the Web Works

DATABASES

Databases store data. On a social media site, this might be your username, password, friends, birthday, posts, etc.



Database

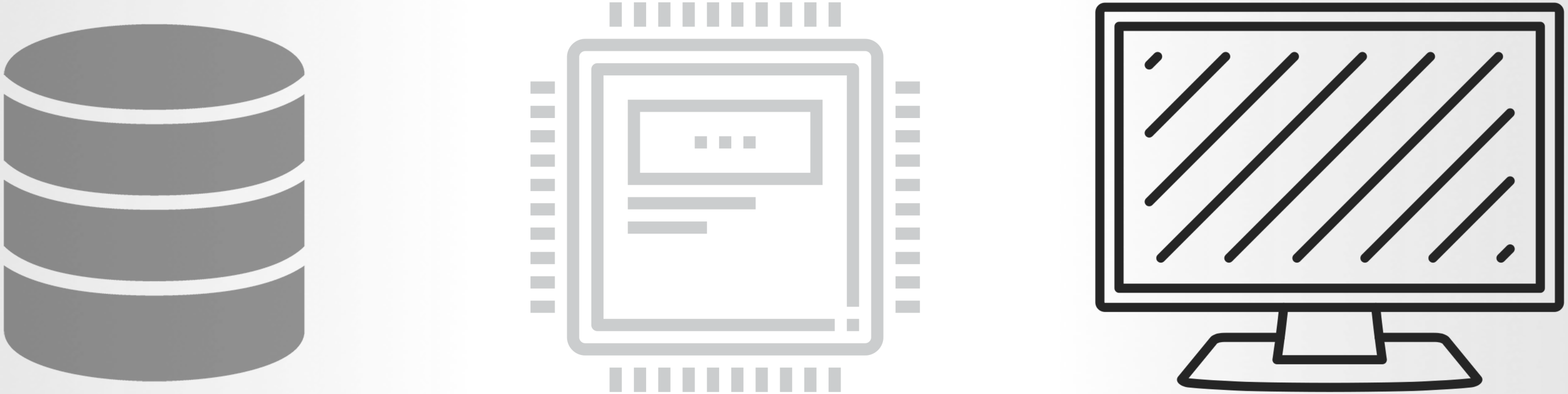
Server

Client

How the Web Works

CLIENTS

Clients are the computers rendering the web app. Like your computer. My computer. The computer of that person sitting next to you.



Database

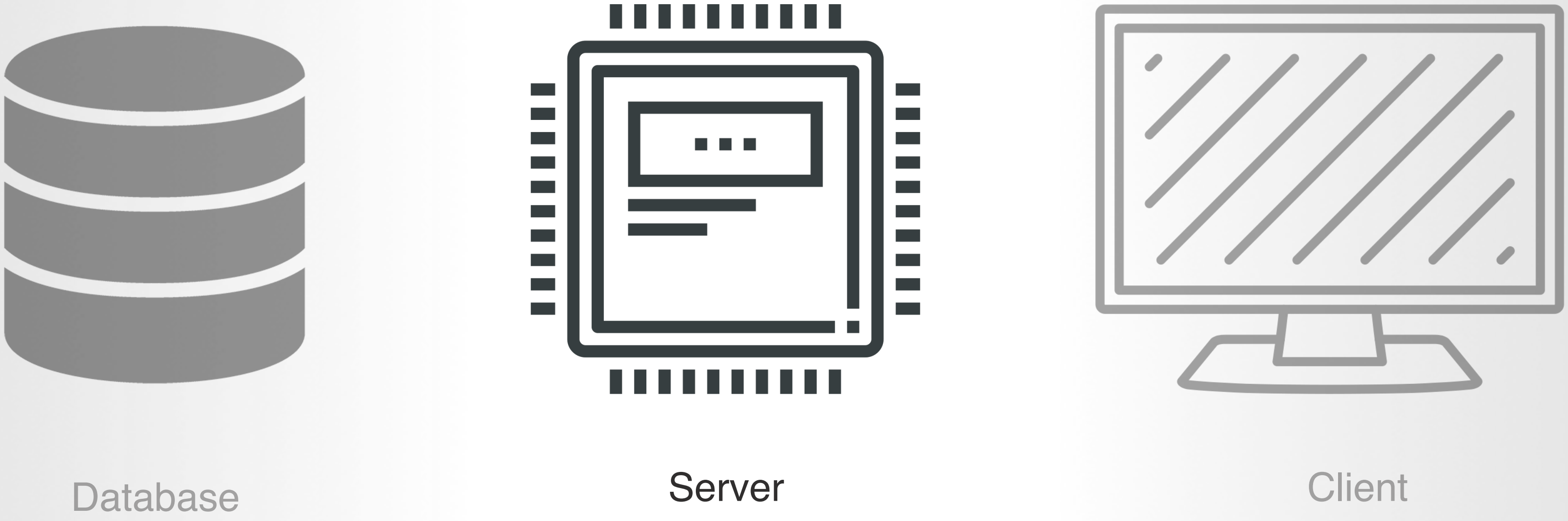
Server

Client

How the Web Works

SERVERS

Servers manage the flow of communication between the client and the server. Servers are how the data from the database gets to page you are viewing.



Database

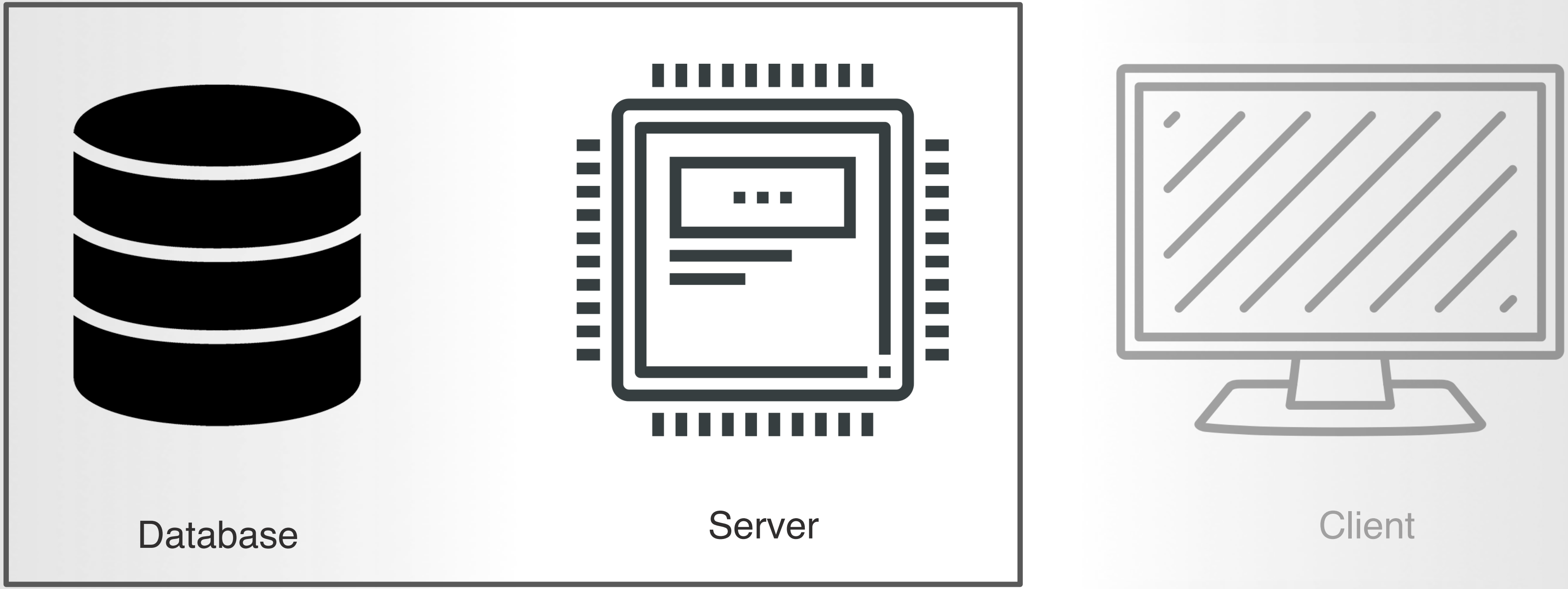
Server

Client

How the Web Works

ANATOMY OF A WEB APPLICATION

Thus far we’ve focused on the server and database:



Database

Server

Client

How the Web Works

ANATOMY OF A WEB APPLICATION

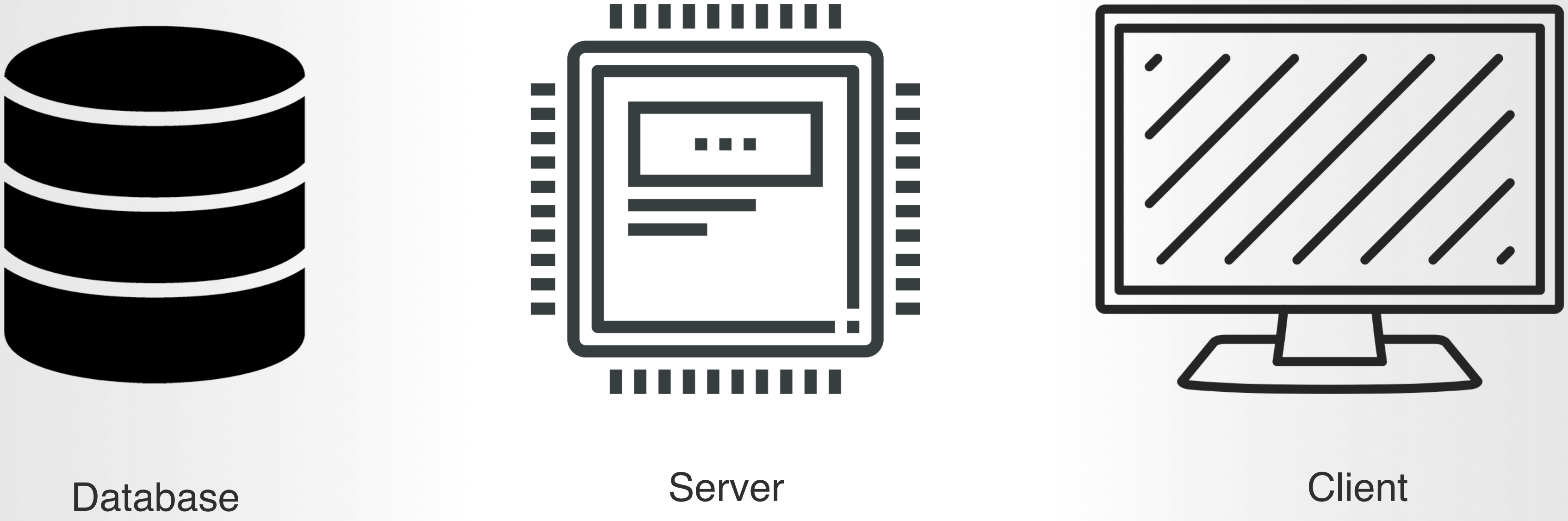
It’s time to shift our focus to the front-end of web applications

Client

How the Web Works

ANATOMY OF A WEB APPLICATION

What language belongs to which component?



Database

Server

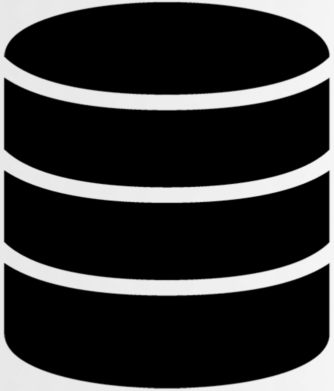
Client

How the Web Works

ANATOMY OF A WEB APPLICATION

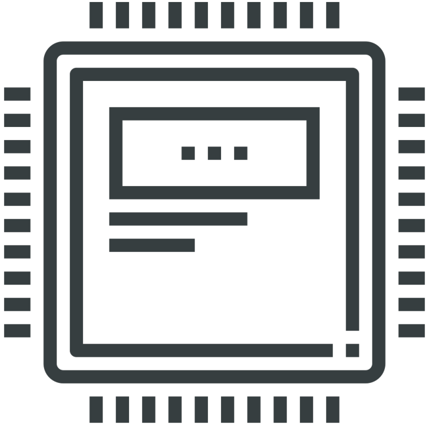
What language belongs to which component?

SQL



Database

Java



Server

JavaScript, HTML, CSS

Client

Intro to Front-End

GETTING DATA TO USERS

For the next few weeks, we’ll be focusing heavily on HTML, CSS, and JavaScript.

Intro to Front-End

GETTING DATA TO USERS

A preview of terms:

HTML - Hypertext Markup Language determines what appears on a website. It’s the content, the words, the images.

CSS - Cascading Style Sheets determines how content appears on the page. It’s the color, the positioning, the size, and the overall styling.

JavaScript - The programming language of front-end development. This is the language we use to manipulate the data coming from the API and get it onto the page.

Angular - Angular is a JavaScript framework. It helps you organize JavaScript projects in a logical, scalable way and provides a lot of built in functionality that makes your application more efficient.

Intro to Front-End

GETTING DATA TO USERS

Suppose we had the following DTO. What would a response from our GET route look like?

```
public class Member {
    private Integer id;
    private String name;
    private Integer age;
    private String bio;

    //getters and setters omitted for brevity
}
```

Intro to Front-End

GETTING DATA TO USERS

If we sent back an ArrayList with 2 Member objects, it might look something like this in JSON.

```
[
   {
       "id": 2,
       "name": "Sue",
       "age": 27,
       "bio": "I like Reiki and beach yoga. I love my cat and all energies on all ephemeral
               planes."
   },
   {
       "id": 4,
       "name": "Simone",
       "age": 33,
       "bio": "I am a singer and travel extensively to perform. Looking for someone who can
              keep up with me."
   }
]
```

Intro to Front-End

GETTING DATA TO USERS

A web application is fundamentally about displaying data to users in a meaningful and easy to digest manner.

The web APIs that we spent the first half of the course building were all about getting data to and from the front-end. Now that we can build an API, let’s see how we can actually retrieve that data and display it in a browser.

Intro to Front-End

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

```
axios.get(‘/members')
    .then(function (response) {
        console.log(response);
    })
```

Intro to Front-End

GETTING DATA TO USERS

So we got the data into our browser, now let’s get it into our actual page.

Intro to Front-End

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>



axios.get(‘/members')
    .then(function (response) {
        document.write(response.data[0].name);
    })
```

Intro to Front-End

GETTING DATA TO USERS

It’s that easy. We’ll spend a lot of time next week on the basics of how to get that data from the API to the page. For now we just want you to know that it’s possible!

Let’s spend a little time exploring HTML, CSS, and JavaScript independently and seeing what’s possible, and then we’ll dive into learning HTML and CSS for the remainder of this module.

Don’t worry! We’ll bring all the API knowledge back into play soon. But this module is about simply understanding that we can get data on to the page and focusing on learning the ins and outs of how to present and style that data.

Front-End Practice

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Read through the provided code.
- Study the JavaScript, HTML, and CSS. It’s a different language than Java but see what you can intuit about JavaScript from your Java knowledge.
- Try changing some colors or sizes in the CSS file and see what happens.

15 minutes!

lunch.

Intro to HTML

Intro to HTML

GETTING STUFF ON THE PAGE

Notebooks Ready? It’s time for a mini lecture.

Intro to HTML

GETTING STUFF ON THE PAGE

HTML is used to make every website.

HTML is the language that determines what appears on webpages.

HTML is made up of opening and closing tags.

HTML is the skeleton of the webpage, we use elements to dictate where everything needs to be.

Intro to HTML

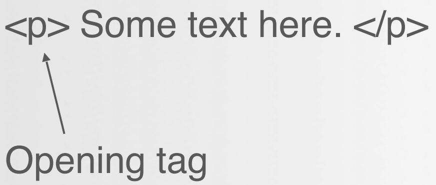
GETTING STUFF ON THE PAGE

```
<p> Some text here. </p>
```

The code above is an example of an HTML element. In the next slides, we’ll break the element down.

Intro to HTML

GETTING STUFF ON THE PAGE

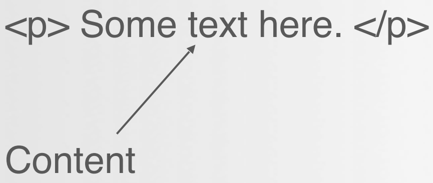


<p> Some text here. </p>

Opening tag

Intro to HTML

GETTING STUFF ON THE PAGE

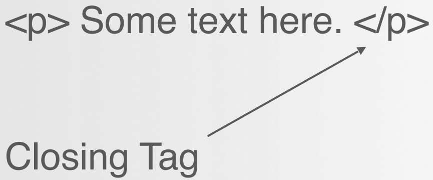


<p> Some text here. </p>

Content

Intro to HTML

GETTING STUFF ON THE PAGE



<p> Some text here. </p>

Closing Tag

Intro to HTML

GETTING STUFF ON THE PAGE

A couple more things of note:

1. A few tags are self-closing.

1. HTML elements can have attributes.

Attributes provide additional info about an element. They come in name/value pairs: name = “value” here.

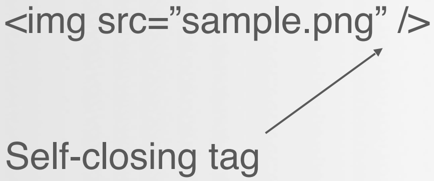
Intro to HTML

GETTING STUFF ON THE PAGE

```
<img src=”sample.png” />
```

Intro to HTML

GETTING STUFF ON THE PAGE



```
<img src=”sample.png” />
```

Self-closing tag

Intro to HTML

GETTING STUFF ON THE PAGE

```
<img src=”sample.png” />
```

Attribute

Intro to HTML

GETTING STUFF ON THE PAGE

We’ve seen demonstrations of the <p> and <img> elements, but what other elements do we have at our disposal?

- <h1>, <h2>, <h3>, <h4>, <h5>, <h6> are used to display headings
- <ul> is an unordered list, <ol> is an ordered list
- <li> is used to display list items
- <div> is used to create divisions in the page
- <a> is used as a link

Let’s take a look at how all of these are used in a web page

Intro to HTML

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
<body>
    <div>
        <h1>This is an example of a header</h1>
    </div>
    <div>
        <ul>
            <li>Lists are really cool</li>
            <li>Wouldn’t you agree?</li>
        </ul>
    </div>
</body>
```

HTML Practice

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create a basic HTML page with your own information that looks similar to the image that you received.

15 minutes!

Breathe

Stay Seated & Take 3 Deep Breaths.

RELAX.

Now take a short walk. Clear your head. After a few minutes break, quickly review your notes.
We’ll start back in 10 minutes.

Intro to CSS

Intro to CSS

MAKING IT PRETTY

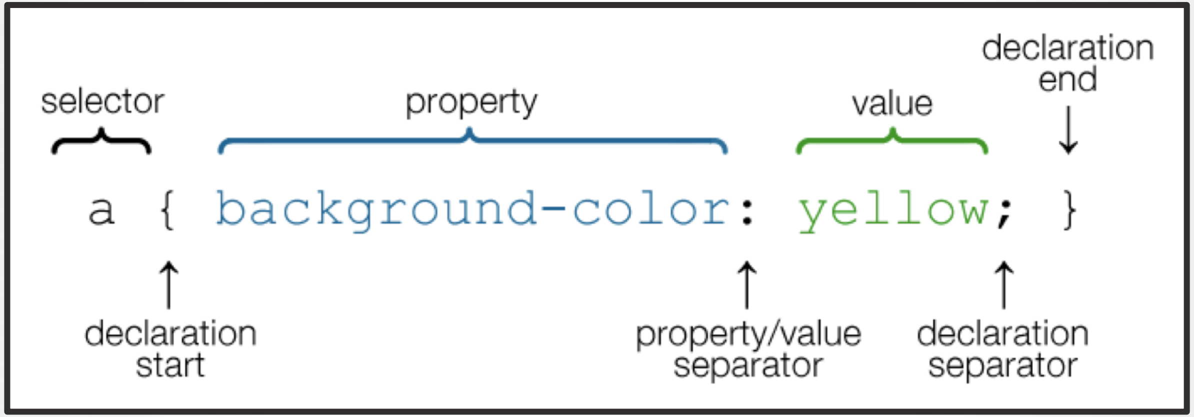
Notebooks Ready? It’s time for a mini lecture.

Intro to CSS

MAKING IT PRETTY

CSS works by hooking onto selectors added into HTML using classes and identifiers.

Once hooked, we apply styles to those HTML elements using CSS.



```
a { background-color: yellow; }
```

Intro to CSS

MAKING IT PRETTY

Frequently Used CSS:

Font / Color:

color: Sets color of text.

font-size: Sets size of the font.

font-style: Sets italics.

font-weight: Sets bold.

Background:

background-color: sets background color.

background-image: sets background image.

Alignment / Spacing:

padding (top/right/bottom/left): Adds space between element and its own border.

margin (top/right/bottom/left): Adds space between element and surrounding elements.

float: Forces elements to the sides, centers, or tops.

Intro to CSS

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
body {
    background-color: #F5DEB3;
    font-family: Arial, Helvetica, sans-serif;
}

h1 {
    text-align: center;
    color: maroon;
    font-size: 48px;
    text-decoration: underline;
}

p {
    color: limegreen;
}
```

Intro to CSS

MAKING IT PRETTY

Specifying Selectors:

- Id’s can be used to give an HTML element a unique identifier that you can use to stylize that specific element
- Classes can be used to assign HTML elements to a group. CSS can use these classes as a selector to style the elements of the group.

Intro to CSS

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

```
.orange {
    color: darkorange;
}

#fancy {
    font-family: "Brush Script MT", "Brush Script Std", cursive;
    color: deeppink;
}

#ugly {
    font-family: "Comic Sans MS”;
    color: olive;
}
```

CSS Practice

INDEPENDENT PRACTICE

It’s time to fly. Focus. Work hard. Ask for help when you need it.

Work in PAIRS to complete all of the goals below.

Goals:

- Create an HTML file and link to a CSS file
- Create a body element and add the following CSS
    - font-family of Arial, Helvetica, sans-serif.
    - background-color of white.
    - font color of #2e2e37.
- Create div element(s) and add the following CSS
    - background-color of #3bb3c7.
    - add a font color of white.
- Create element(s) with a footnote class and add the following CSS
    - font color of #5f6070.
    - add a font-size of 11px.

15 minutes!

CSS Specificity

MAKING IT PRETTY

What is specificity?

- Sometimes an element can have conflicting CSS properties, the browser must then decide which is most relevant to the element.
- You can think of CSS specificity as a hierarchy! Your CSS properties will overlap based on those hierarchies.

CSS Specificity

WATCH & LEARN

Close your laptop. Eyes on my screen. Pay attention.

HTML:

```
<h1 class="new-text">Specificity!<h1>
```

CSS:

```
.new-text {
     color: red;
}

h1 {
     color: blue;
}
```