

# J220

## Coding for Journalists

LECTURER

Yoli Martinez

PROMPTS

Sign into Poll Everywhere:

<https://pollev.com/yolim>

Open Sublime Text

start Zoom recording + captions

# Agenda

Announcements

Accessibility homework review

Importing CSS and CSS syntax

## **BREAK**

Fonts and **many** common CSS properties

Homework

# Announcements

NICAR Conference from March 2-5 🎉🎉

1. How to plan and set up coffee meetings with other attendees
  - a. Follow #NICAR23,
  - b. Join conference Slacks channels
  - c. Review schedule **NOW**
  - d. Wifi is always terrible, use old-school pen/paper
2. RSVP and attend the Berkeley Alumni meetup
  - a. March 3 @ 8pm
3. Bring back something you learned! It can be a new tool or ideas for collaboration
  - a. We'll set aside **30 mins** of next week's class for this skillshare

# Announcements

More javascript!

There's a lot of interest in learning, so instead of one week focused on javascript, we'll also use half of April 3 to get started

Class website is updated with the new schedule

# Announcements

Typo in meta tag in past slides

## **Wrong:**

```
<meta name="viewport" content="width=width-device,initial-scale=1">
```

## **Correct:**

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

# Homework Review: Meta Data


We expect all meta data to be on the page when turning in assignments that build a page:

```
<!DOCTYPE html>
<html class="html" lang="en">
  <head>
    <!-- what the browser needs -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>CSS Examples</title>
    <meta name="description" content="A page with coding samples">
  </head>
  <body>
    <!--seen by the user -->
  </body>
</html>
```

# Homework Review: Meta Data

We expect all meta data to be on the page when turning in assignments that build a whole page:

```
<!DOCTYPE html>
<html class="html" lang="en">
  <head>
    <!-- what the browser needs -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
    <title>CSS Examples</title>
    <meta name="description" content="A page with
  </head>
  <body>
    <!--seen by the user -->
  </body>
</html>
```



You **NEED**  
charset, viewport  
and most  
importantly title  
and description



# Homework Review: h3

A **LOT** of you used an h3 to get bold styling in the pull quote

Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

Vivamus fringilla purus sed porttitor pulvinar. Sed sodales, neque dictum suscipit ultricies, ex mauris molestie nulla, at interdum massa mauris et est. Integer ullamcorper sit amet diam eget lacinia. Fusce sit amet mauris ligula. Donec massa eros, vehicula quis rhoncus id, malesuada eget dolor

**"Vivamus fringilla purus sed porttitor pulvinar."**

Assume this is a pullquote

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus.

Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.

# Homework Review: h3

We haven't learned CSS yet, so this couldn't be styled properly

Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

Vivamus fringilla purus sed porttitor pulvinar. Sed sodales, neque dictum suscipit ultricies, ex mauris molestie nulla, at interdum massa mauris et est. Integer ullamcorper sit amet diam eget lacinia. Fusce sit amet mauris ligula. Donec massa eros, vehicula quis rhoncus id, malesuada eget dolor

**"Vivamus fringilla purus sed porttitor pulvinar."**

Assume this is a pullquote

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus.

Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.


# Homework Review: h3

You made a common accessibility mistake, using semantic HTML incorrectly to fit the style you want. Is this quote a heading? No.

Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

Vivamus fringilla purus sed porttitor pulvinar. Sed sodales, neque dictum suscipit ultricies, ex mauris molestie nulla, at interdum massa mauris et est. Integer ullamcorper sit amet diam eget lacinia. Fusce sit amet mauris ligula. Donec massa eros, vehicula quis rhoncus id, malesuada eget dolor

**"Vivamus fringilla purus sed porttitor pulvinar."**



Assume this is a pullquote

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus.

Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.


# Homework Review: h3

Instead of <h3>, which comes with automatic styling, correct use would be a <p> tag. Styling would be done later, with CSS

Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

Vivamus fringilla purus sed porttitor pulvinar. Sed sodales, neque dictum suscipit ultricies, ex mauris molestie nulla, at interdum massa mauris et est. Integer ullamcorper sit amet diam eget lacinia. Fusce sit amet mauris ligula. Donec massa eros, vehicula quis rhoncus id, malesuada eget dolor

**"Vivamus fringilla purus sed porttitor pulvinar."**



Assume this is a pullquote

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus.

Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.

# Homework Review: h3

Instead of <h3>, which comes with automatic styling, correct use would be a <p> tag. Styling would be done later, with CSS

Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu.

Vivamus fringilla purus sed porttitor pulvinar. Sed sodales, neque dictum suscipit ultricies, ex mauris molestie nulla, at interdum massa mauris et est. Integer ullamcorper sit amet diam eget lacinia. Fusce sit amet mauris ligula. Donec massa eros, vehicula quis rhoncus id, malesuada eget dolor

**"Vivamus fringilla purus sed porttitor pulvinar."**



Assume this is a pullquote

In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus.

Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim.



# Homework Review: main and article

In some submissions, you used `<main>` or `<article>`. It can be argued both can be correct, but their placement needs to include the whole page.

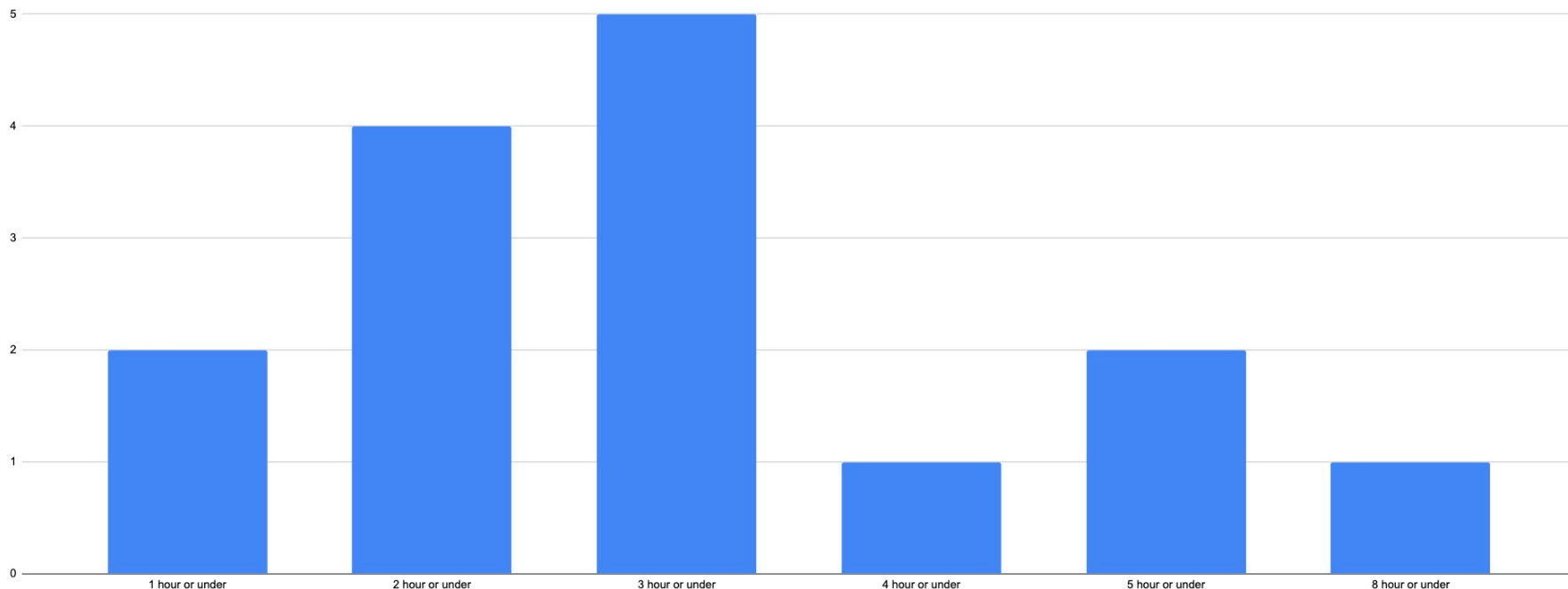
Which ever you use, they have to wrap the `<header>` and `<figure>`, these are part of the article or main part of the page.

Don't add `<main>` or `<article>` after your heading

# What questions do you have?

# How much time spent on J220 last week

Week 02-13: Number of students grouped by hours spent outside of lecture and office hours





# Importing CSS and CSS Syntax

# Importing CSS

CSS = **C**ascading **S**tyle **S**heets

We'll learn soon what the cascading means :)

# Importing CSS

Let's review our HTML document.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
  </head>
  <body>
    <main>
      <header>
        <h1>My first website</h1>
      </header>
      <p>Some text for the article.</p>
    </main>
  </body>
</html>
```

# Importing CSS

Let's review our HTML document.

- Stuff in **<head>** is read by the browser

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
  </head>
  <body>
    <main>
      <header>
        <h1>My first website</h1>
      </header>
      <p>Some text for the article.</p>
    </main>
  </body>
</html>
```

# Importing CSS

Let's review our HTML document.

- Stuff in `<head>` is read by the browser

- Everything in **`<body>`** is seen by the reader

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
  </head>
  <body>
    <main>
      <header>
        <h1>My first website</h1>
      </header>
      <p>Some text for the article.</p>
    </main>
  </body>
</html>
```

# Importing CSS

We add our CSS inside of **<head>** since the browser checks this area to figure out how to render the page.

Let's focus in...

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
  </head>
  <body>
    <main>
      <header>
        <h1>My first website</h1>
      </header>
      <p>Some text for the article.</p>
    </main>
  </body>
</html>
```

# Importing CSS

There are two ways of adding CSS to your file.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
  </head>
  <body>
    <!-- content on page -->
  </body>
</html>
```

## Importing CSS

There are two ways of adding CSS to your file.

1. You can write your CSS directly in your file by adding **<style>** tags inside of `<head>`. The CSS would be written in between the tags.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
    <style>
      /* CSS goes here */
    </style>
  </head>
  <body>
    <!-- content on page -->
  </body>
</html>
```



## Importing CSS

There are two ways of adding CSS to your file.

2. You can **link** your CSS file

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
    <link
      rel="stylesheet"
      href="./styles/main.css">
  </head>
  <body>
    <!-- content on page -->
  </body>
</html>
```

## Importing CSS

- a. The **“rel=”** attribute tells the browser that the linked file will add styling to the page

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
    <link
      rel="stylesheet"
      href="./styles/main.css">
  </head>
  <body>
    <!-- content on page -->
  </body>
</html>
```

## Importing CSS

b. “**href=**” lets the browser know where to find this file. Just like using href in an `<a>` link tag.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
    <link
      rel="stylesheet"
      href="./styles/main.css">
  </head>
  <body>
    <!-- content on page -->
  </body>
</html>
```

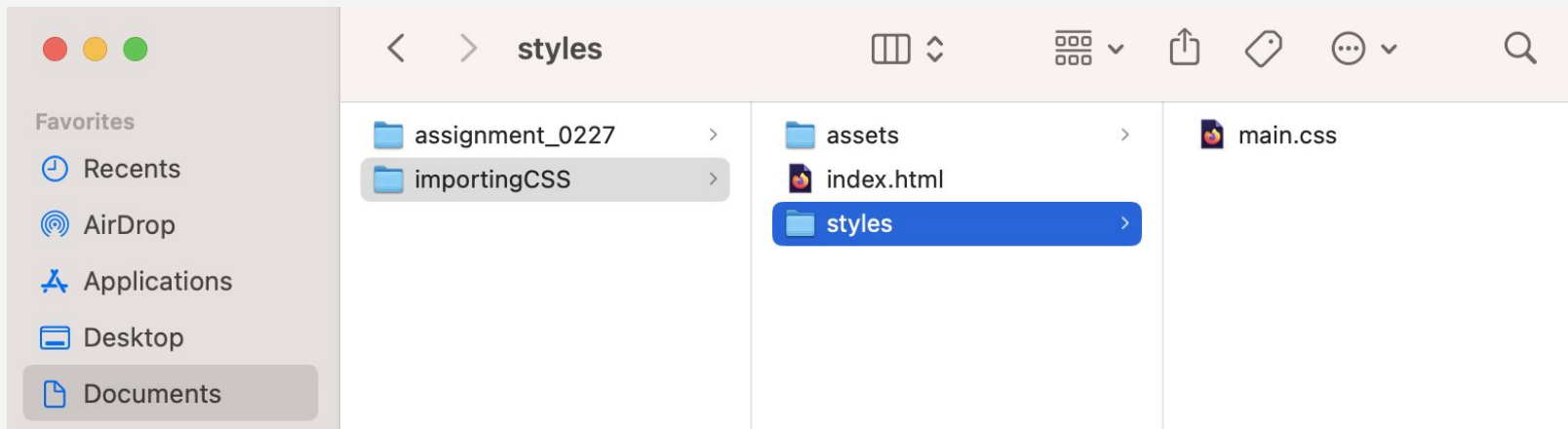
# Importing CSS

In this class we'll be using the link method.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- meta tags -->
    <link
      rel="stylesheet"
      href="./styles/main.css">
  </head>
  <body>
    <!-- content on page -->
  </body>
</html>
```

# Importing CSS

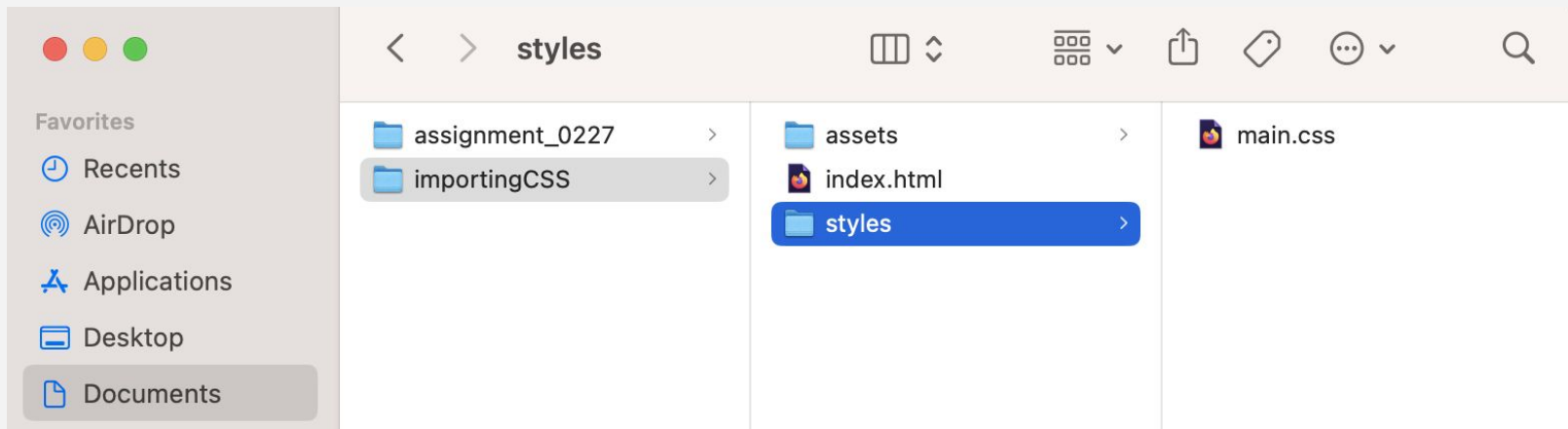
Just like you create an “assets” folder to store your images. You have to make a “styles” folder to hold your CSS. Notice the “.css” extension in “main.css”



# Importing CSS

To import CSS into the index.html file:

```
<link rel="stylesheet" href="./styles/main.css">
```



# What questions do you have?

# CSS Syntax

A quick review:

Anchor tags using  
ids for links

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```



# CSS Syntax

Use id to select a unique section of the page, allowing for in-page navigation.

If someone clicked on the Blog link, the page scroll down to the section with “id=blog”

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

Sidenote: Look at what happens to the url. The id hash gets added.

This is how people are able to share urls that link to specific parts of the page.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

A few rules for id:

They can only be used once. They are unique and if the same id is used multiple times, browser gets confused.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

Why does this  
matter for CSS?

Because you use id  
for styling too!

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

There are MANY ways of selecting an element to be styled in CSS.

# CSS Syntax

There are MANY ways of selecting an element to be styled in CSS.

Before writing the CSS, let's learn how these CSS Selectors look like on the the HTML page

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

These are the most  
common CSS  
Selectors

## 1. id

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

These are the most common CSS Selectors

## 1. id

Remember, an id can only be used **ONCE**

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```



# CSS Syntax

These are the most common CSS Selectors

1. id
2. class

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

These are the most common CSS Selectors

1. id
2. class

Use class when you need to select multiple items!

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

These are the most common CSS Selectors

1. id
2. class
3. HTML tag

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

This html page contains ids, classes, and many tags. With everything marked up properly, we can now start using CSS to style!

```
<!DOCTYPE html>
<html>
  <head>
    <!-- metadata goes here -->
  </head>
  <body>
    <nav>
      <ul>
        <li><a href="#blog">Blog</a></li>
        <li><a href="#resume">Resume</a></li>
      </ul>
    </nav>
    <header>
      <h1>Learning CSS</h1>
    </header>
    <section id="blog">
      <h2>Blog</h2>
      <!-- text goes here -->
    </section>
    <section class="colorSection" id="resume">
      <h2>Resume</h2>
      <!-- text goes here -->
    </section>
  </body>
</html>
```

# CSS Syntax

There is a specific way to grab each CSS selector in our main.css file:

1. id: use hash(#)

```
<section id="blog">  
  <h2>Blog</h2>  
  <!-- text goes here -->  
</section>
```

```
#id {  
  /* css properties go here*/  
}
```

# CSS Syntax

There is a specific way to grab each CSS selector in our main.css file:

2. class: use period (.)

```
<section class="colorSection">
  <h2>Resume</h2>
  <!-- text goes here -->
</section>
```

```
#id {
  /*  css properties go here*/
}

.colorSection {
  /*  css properties go here*/
}
```

# CSS Syntax

There is a specific way to grab each CSS selector in our main.css file:

3. tag: use name

```
<h2>Blog</h2>  
<!-- text goes here -->
```

```
#id {  
    /*    css properties go here*/  
}  
  
.colorSection {  
    /*    css properties go here*/  
}  
  
h2 {  
    /*    css properties go here*/  
}
```

# What questions do you have?



# CSS Syntax

CSS looks different from HTML.

```
#id {  
    /*    css properties go here*/  
}  
  
.colorSection {  
    /*    css properties go here*/  
}  
  
h2 {  
    /*    css properties go here*/  
}
```

# CSS Syntax

CSS looks different from HTML.

- Each selector has an open/close curly bracket

```
#id {  
    /*    css properties go here*/  
}  
  
.colorSection {  
    /*    css properties go here*/  
}  
  
h2 {  
    /*    css properties go here*/  
}
```

# CSS Syntax

CSS looks different from HTML.

- Each selector has an open/close curly bracket
- Comments, different from HTML's: `<!-- text -->`

```
#id {  
    /*    css properties go here*/  
}  
  
.colorSection {  
    /*    css properties go here*/  
}  
  
h2 {  
    /*    css properties go here*/  
}
```

# CSS Syntax

TIP:

Text editors let you easily add comments without having to remember the styling for different language:

**command+/**

CSS Comments:

```
/* css comments */
```

HTML Comments

```
<!-- html comments -->
```

# CSS Syntax

CSS looks different from HTML.

- Each selector has an open/close curly bracket
- Comments, different from HTML's: `<-- text -->`
- Indentation, two spaces

```
#id {  
  css properties go here*  
}
```

```
.colorSection {  
  css properties go here*  
}
```

```
h2 {  
  css properties go here*  
}
```

# CSS Syntax

Let's add some CSS properties!

property

:

value

;

```
#id {  
  /*  css properties go here*/  
}  
  
.colorSection {  
  /*  css properties go here*/  
}  
  
h2 {  
  /*  css properties go here*/  
}
```

# CSS Syntax

Let's add some CSS properties!

- color (for text)

property

:

value

;

```
#id {  
    /*    css properties go here*/  
}  
  
.colorSection {  
    color: white;  
}  
  
h2 {  
    color: yellow;  
}
```

# CSS Syntax

Let's add some CSS properties!

- color (for text)
- background-color

**property**

**:**

**value**

**;**

```
#id {  
  background-color: seagreen;  
}  
  
.colorSection {  
  color: white;  
  background-color: black;  
}  
  
h2 {  
  color: yellow;  
}
```



# CSS Syntax

Let's add some CSS properties!

- color (for text)
- background-color

property	:	value	;
----------	---	-------	---

semicolon separates  
different properties

```
#id {  
  background-color: seagreen;  
}  
  
.colorSection {  
  color: white;  
  background-color: black;  
}  
  
h2 {  
  color: yellow;  
}
```

# CSS Syntax

Review:

property

:

value

;

```
#id {  
  background-color: seagreen;  
}  
  
.colorSection {  
  color: white;  
  background-color: black;  
}  
  
h2 {  
  color: yellow;  
}
```

# CSS Syntax

Review:

property	:	value	;
----------	---	-------	---

```
#id {  
  background-color: seagreen;  
}  
  
.colorSection {  
  color: white;  
  background-color: black;  
}  
  
h2 {  
  color: yellow;  
}
```

# What questions do you have?

# Selecting CSS

By using class selectors in CSS you can simplify your styling.

Let's pretend the second `<p>` and the section both need a green background

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>A sentence to test using color in CSS.</p>
      <p class="useBackground">Another sentence using class.</p>
      <section class="useBackground">
        <h2>A new section title</h2>
        <p>A normal sentence inside of a section</p>
      </section>
    </main>
  </body>
</html>
```

# Selecting CSS

There could be an implementation that looks like this:

```
.useBackground {  
  background-color: green;  
}  
section {  
  background-color: green;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section>  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Selecting CSS

But by adding the class to the section, you simplify your CSS:

```
.useBackground {  
  background-color: green;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Selecting CSS

If you need all your headings to be the same color, you can select each:

```
h1 {  
  color: red;  
}  
h2 {  
  color: red;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```



# Selecting CSS

Or you can separate them with a comma, help reduce lines:

```
h1, h2 {  
  color: red;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Selecting CSS

People can get extremely creative when it comes to styling CSS selectors, especially once you start adding attributes

Read more:

[https://www.w3schools.com/cssref/css\\_selectors.php](https://www.w3schools.com/cssref/css_selectors.php)

# The Cascading in CSS

Important rules to remember:

1. CSS is read by the browser from top to bottom.
2. The most specific CSS selector creates the style
  - a. id > class > tag

# Cascading

Which background color will be used?

```
.useBackground {  
  background-color: blue  
}  
p {  
  color: red;  
}  
.useBackground {  
  background-color: gray;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Cascading

Which background color will be used?

```
.useBackground {  
  background-color: blue  
}  
p {  
  color: red;  
}  
.useBackground {  
  background-color: gray;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Cascading

What will be the background color for section?

```
p {  
  color: red;  
}  
.useBackground {  
  background-color: blue  
}  
section {  
  background-color: gray;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Cascading

What will be the background color for section?

```
p {  
  color: red;  
}  
.useBackground {  
  background-color: blue  
}  
section {  
  background-color: gray;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Cascading

What will be the background color for section?

```
#specialSection {  
  background-color: seagreen;  
}  
p {  
  color: red;  
}  
.useBackground {  
  background-color: blue  
}  
section {  
  background-color: gray;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground" id="specialSection">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```



# Cascading

What will be the background color for section?

```
#specialSection {  
  background-color: seagreen;  
}  
p {  
  color: red;  
}  
.useBackground {  
  background-color: blue  
}  
section {  
  background-color: gray;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>A sentence to test using color in CSS.</p>  
      <p class="useBackground">Another sentence using class.</p>  
      <section class="useBackground" id="specialSection">  
        <h2>A new section title</h2>  
        <p>A normal sentence inside of a section</p>  
      </section>  
    </main>  
  </body>  
</html>
```

# Break

Meet back in 15 minutes.

start Zoom recording + captions

# Screenshare

# CSS Properties

In these examples, we've been using the name of the color, but there are other, more ways of defining them.

```
#id {  
    color: red;  
}
```

# CSS Properties

As you type, the text editor can auto-populate a color.

Ex: darkseagreen, lavender, limegreen

Full list:

<https://developer.mozilla.org/en-US/docs/Web/CSS/named-color>

```
#id {  
  color: red;  
}
```

# CSS Properties

## 1. Named-color

```
#id {  
  color: red;  
}
```

# CSS Properties

1. Named-color
2. Hex color

```
#id {  
  color: #FF0000;  
}
```



# CSS Properties

1. Named-color
2. Hex color
3. RGB color

```
#id {  
    color: rgb(255, 0, 0);  
}
```

# CSS Properties

1. Named-color
2. Hex color
3. RGB color
4. HSL

```
#id {  
    color: hsl(0, 100%, 50%);  
}
```

# CSS Properties

1. Named-color
2. Hex color
3. RGB color
4. HSL

These are all the same color. What you used will depend on personal preference or the styles used by your news org.

```
#id {  
  color: red;  
}
```

```
#id {  
  color: #FF0000;  
}
```

```
#id {  
  color: rgb(255, 0, 0);  
}
```

```
#id {  
  color: hsl(0, 100%, 50%);  
}
```

# CSS Properties

You can use color in  
color,  
background-color,  
border, outlines, box  
and text shadows

We'll use only color  
and background-color

```
#id {  
  color: red;  
}
```

```
#id {  
  color: #FF0000;  
}
```

```
#id {  
  color: rgb(255, 0, 0);  
}
```

```
#id {  
  color: hsl(0, 100%, 50%);  
}
```

# CSS Properties

Adobe Color:

<https://color.adobe.com/create/color-contrast-analyzer>

Google a hex color, usually end up here:

<https://www.color-hex.com/>



# Quick Intro to Fonts

# Fonts

## Serif

Times New Roman

Garamond

Georgia

## Sans-serif

Verdana

Lato

Roboto

**Font: Serif**

Georgia



# Font: Serif



Georgia

Finishing strokes. Some claim it's easier to read in the body of articles. Think about the tone you're setting.

**Font: Sans-serif**

Roboto

# Font: Sans-serif



Roboto

Sans = “without” in French. These are fonts with no decorating features. Usually used for displays or titles

# CSS Properties

Not everyone has every font installed on their computers, with **font-family**, you can let the browser know what fonts to use and fall back on.

You need at least **two values**. Usually use three

```
h1 {  
  font-family:  
}
```

# CSS Properties

1. Your preferred font

```
h1 {  
    font-family: Georgia;  
}
```

# CSS Properties

1. Your preferred font
2. Your fallback

```
h1 {  
  font-family: Georgia, serif;  
}
```

# CSS Properties

1. Your preferred font
2. Your back-up

If a font name is more than one word, wrap it in quotes. You can use single (") or double (""), whichever you prefers

```
h1 {  
    font-family: "Times New Roman",  
    serif;  
}
```

# CSS Properties

Like the CSS color property, setting the size for fonts can be done in several ways.

1. pixels
2. rem
3. em
4. percent

```
p {  
  font-size:  
}
```



# CSS Properties

px:

A static way of setting a size. Good if you need to be very accurate, down to the pixel.

Cons: no resizing when page is zoomed in

```
p {  
  font-size: 16px;  
}
```

# CSS Properties

rem:

The browser adjusts size based on a font-size set in the <html> element.

This helps when zooming into page, mobile friendly

```
/*Base size is set in html*/  
html {  
    font-size: 16px;  
}  
p {  
    font-size: 1rem;  
}
```

# CSS Properties

em:

The font-size adjusts based on the size of the parent element.

```
/*Base size*/  
main {  
    font-size: 16px;  
}  
p {  
    font-size: 1rem;  
}
```

# CSS Properties

em:

If no base size is set, then a browser default is set. In most browser, a <p> is 16px

It can also adjust inherent sizes (like h1-h6)

```
/*Base size*/
main {
  font-size: 16px;
}
p {
  font-size: 1rem;
}
```

# CSS Properties

percent:

A size of 100% would be equivalent to 1em.

Percent sizing is more common with layout properties for spacing, not so much with fonts.

```
p {  
  font-size: 100%;  
}
```

# CSS Properties

For both accessibility and mobile-first production, we'll use **rem** for our font sizing.

```
/*Base size is set in html*/  
html {  
    font-size: 16px;  
}  
p {  
    font-size: 1rem;  
}
```

# What questions do you have?

Let's finally  
add some CSS! 🎉



# CSS Properties

Here is a simple page.  
We'll try to apply  
some common CSS  
properties to <p>

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>A sentence to test using color in CSS.</p>
      <p class="useBackground">Another sentence using
class.</p>
    </main>
  </body>
</html>
```

# CSS Properties

Here is a simple page.  
We'll try to apply  
some common CSS  
properties to <p>

- color
- font-family
- font-size

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>A sentence to test using color in CSS.</p>
      <p class="useBackground">Another sentence using
class.</p>
    </main>
  </body>
</html>
```

# CSS Properties

The `<p>` tag should use the Garamond font (a serif), be a size 16px (using rem) with the color blue.

How would you do this?

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>A sentence to test using color in CSS.</p>
      <p class="useBackground">Another sentence using
class.</p>
    </main>
  </body>
</html>
```

# Write CSS in your text editor, then paste to pollev



# CSS Properties

When using rem,  
remember to set the  
base size in html

```
/*Base size*/  
html {  
    font-size: 16px;  
}  
p {  
    font-family: Garamond, serif;  
    font-size: 1rem;  
    color: blue;  
}
```

# CSS Properties

Keep the CSS from before and add:

## **useBackground**

should have a background-color of black and font-size that's double of 16px.

How would you do this?

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>A sentence to test using color in CSS.</p>
      <p class="useBackground">Another sentence using
class.</p>
    </main>
  </body>
</html>
```

# Write CSS in your text editor, then paste to pollev



# CSS Properties

2rem = 32px

It automatically  
doubles the base-size.

If the base size were  
12px, 2rem would  
equal 24px.

```
/*Base size*/
html {
  font-size: 16px;
}
p {
  font-family: Garamond, serif;
  font-size: 1rem;
  color: blue;
}
.useBackground {
  background-color: black;
  font-size: 2rem;
}
```



# What questions do you have?

# CSS Properties

Quick review of more CSS properties for fonts

- font-weight
- font-style
- line-height

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
}
```

# CSS Properties

font-weight:

Sets the “weight” or boldness of a font. The CSS value can be a multiple of 100 from 100-900 or a select word: **lighter**, **normal**, **bold**, **bolder**

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700;  
}
```

# CSS Properties

Common conversions:

100 = thin

200

300 = light

400 = normal

500

600

700 = bold

800

900 = black, heavy

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700; /* renders as bold */  
}
```

# CSS Properties

Because of accessibility concerns, you need good reason to use font-weight 100 or 200, since these make it hard to read because of size and contrast

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700; /* renders as bold */  
}
```

# CSS Properties

Why use font-weight to make something bold instead of **<strong>**?

The tag has semantic meaning. It points to something that is important or serious. Not done for aesthetic reasons

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700;  
}
```

# CSS Properties

font-style:

Mostly used for one  
reason: **italics**

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700;  
  font-style: italics;  
}
```

# CSS Properties

font-style:

Other options include:  
normal, oblique,  
oblique with an angle

Avoid big sections of  
italic text, hard to read  
for people with  
Dyslexia

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700;  
  font-style: italics;  
}
```



# CSS Properties

line-height

The space between sentences.

Can use several CSS values: percent, em/rem or a number

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700;  
  font-style: italics;  
  line-height: 1.5;  
}
```

# CSS Properties

## line-height

The preferred implementation for both readability and making the page mobile-friendly is using a number.

Example: 1.5, which is multiplied by the font-size.

```
p {  
  font-family: Garamond, serif;  
  font-size: 1rem;  
  color: blue;  
  font-weight: 700;  
  font-style: italics;  
  line-height: 1.5;  
}
```

# What questions do you have?

# CSS Properties: Width

Just like with font, the width CSS value can be pixels, em, rem or percent.

For us, we'll be learning to set width to make pages a bit more mobile friendly.

# CSS Properties: Width

Imagine we have a very very long paragraph. In our set up now, the text reaches each edge of the page. It hard to read something that stretches across the page.

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget libero eget augue venenatis tempus mollis sed sem. Morbi in dictum tellus, eget elementum turpis. Sed commodo sapien eget imperdiet fringilla. Aenean pharetra quis lectus a pharetra. Maecenas felis ligula, feugiat at finibus vel, pharetra vel dolor. Maecenas rutrum mauris finibus, mollis odio ornare, accumsan lectus. Aenean erat sapien, sagittis non odio ultricies, pretium blandit diam. Suspendisse id elementum dui, et pretium tellus. Vestibulum blandit aliquet orci. Praesent consectetur porta quam ornare bibendum. Donec vel mollis nulla, at varius lacus. Nulla lacus neque, feugiat in lectus gravida, venenatis volutpat ex.

Aliquam condimentum sem a lorem volutpat, in sagittis lorem efficitur. Integer scelerisque ante tortor, ut viverra tortor mattis vel. Curabitur pretium mi id nulla tincidunt venenatis. Maecenas eu elementum metus. Suspendisse imperdiet, dui eu ornare imperdiet, nisl ligula laoreet purus, ut blandit enim orci in dolor. Maecenas nec viverra est. Aenean non porta enim. Mauris ligula leo, porttitor ac viverra sit amet, ornare id justo. Sed laoreet at elit ac viverra. Cras volutpat diam eget tellus cursus, vitae bibendum mauris vehicula. Donec ex augue, mattis eu mattis id, euismod eu ex.

Suspendisse potenti. Sed sit amet lacinia turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sollicitudin nunc velit, non maximus ante dignissim eget. Morbi vel imperdiet quam. Vivamus vestibulum et eros sed commodo. Curabitur vel massa turpis. Praesent ut est lobortis, finibus lorem eget, sollicitudin enim. Sed commodo, nisl id interdum congue, nunc metus lobortis libero, vel sagittis nunc felis eu erat. Cras porta pretium maximus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus sollicitudin, erat in consequat pellentesque, neque neque consequat est, ac varius est ex rutrum diam. Donec augue orci, iaculis consectetur purus et, cursus varius enim. Donec tristique vehicula dui vitae porta. Maecenas suscipit, nisl eu porta viverra, sapien tortor fermentum sapien, et aliquet est nisl eget sapien. Nulla sed ex iaculis, lobortis turpis et, tincidunt turpis.

# CSS Width

Using the width property, we can control the size an html element

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>Pretend there are many sentences here.</p>
      <p>Pretend there are many sentences here.</p>
    </main>
  </body>
</html>
```

# CSS Width

Using the width property, we can control the size an html element

```
main {  
  width: 90%;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>Pretend there are many sentences here.</p>  
      <p>Pretend there are many sentences here.</p>  
    </main>  
  </body>  
</html>
```

# CSS Properties: Width

Before:

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget libero eget augue venenatis tempus mollis sed sem. Morbi in dictum tellus, eget elementum turpis. Sed commodo sapien eget imperdiet fringilla. Aenean pharetra quis lectus a pharetra. Maecenas felis ligula, feugiat at finibus vel, pharetra vel dolor. Maecenas rutrum mauris finibus, mollis odio ornare, accumsan lectus. Aenean erat sapien, sagittis non odio ultricies, pretium blandit diam. Suspendisse id elementum dui, et pretium tellus. Vestibulum blandit aliquet orci. Praesent consectetur porta quam ornare bibendum. Donec vel mollis nulla, at varius lacus. Nulla lacus neque, feugiat in lectus gravida, venenatis volutpat ex.

Aliquam condimentum sem a lorem volutpat, in sagittis lorem efficitur. Integer scelerisque ante tortor, ut viverra tortor mattis vel. Curabitur pretium mi id nulla tincidunt venenatis. Maecenas eu elementum metus. Suspendisse imperdiet, dui eu ornare imperdiet, nisl ligula laoreet purus, ut blandit enim orci in dolor. Maecenas nec viverra est. Aenean non porta enim. Mauris ligula leo, porttitor ac viverra sit amet, ornare id justo. Sed laoreet at elit ac viverra. Cras volutpat diam eget tellus cursus, vitae bibendum mauris vehicula. Donec ex augue, mattis eu mattis id, euismod eu ex.

Suspendisse potenti. Sed sit amet lacinia turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sollicitudin nunc velit, non maximus ante dignissim eget. Morbi vel imperdiet quam. Vivamus vestibulum et eros sed commodo. Curabitur vel massa turpis. Praesent ut est lobortis, finibus lorem eget, sollicitudin enim. Sed commodo, nisl id interdum congue, nunc metus lobortis libero, vel sagittis nunc felis eu erat. Cras porta pretium maximus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus sollicitudin, erat in consequat pellentesque, neque neque consequat est, ac varius est ex rutrum diam. Donec augue orci, iaculis consectetur purus et, cursus varius enim. Donec tristique vehicula dui vitae porta. Maecenas suscipit, nisl eu porta viverra, sapien tortor fermentum sapien, et aliquet est nisl eget sapien. Nulla sed ex iaculis, lobortis turpis et, tincidunt turpis.

After:

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget libero eget augue venenatis tempus mollis sed sem. Morbi in dictum tellus, eget elementum turpis. Sed commodo sapien eget imperdiet fringilla. Aenean pharetra quis lectus a pharetra. Maecenas felis ligula, feugiat at finibus vel, pharetra vel dolor. Maecenas rutrum mauris finibus, mollis odio ornare, accumsan lectus. Aenean erat sapien, sagittis non odio ultricies, pretium blandit diam. Suspendisse id elementum dui, et pretium tellus. Vestibulum blandit aliquet orci. Praesent consectetur porta quam ornare bibendum. Donec vel mollis nulla, at varius lacus. Nulla lacus neque, feugiat in lectus gravida, venenatis volutpat ex.

Aliquam condimentum sem a lorem volutpat, in sagittis lorem efficitur. Integer scelerisque ante tortor, ut viverra tortor mattis vel. Curabitur pretium mi id nulla tincidunt venenatis. Maecenas eu elementum metus. Suspendisse imperdiet, dui eu ornare imperdiet, nisl ligula laoreet purus, ut blandit enim orci in dolor. Maecenas nec viverra est. Aenean non porta enim. Mauris ligula leo, porttitor ac viverra sit amet, ornare id justo. Sed laoreet at elit ac viverra. Cras volutpat diam eget tellus cursus, vitae bibendum mauris vehicula. Donec ex augue, mattis eu mattis id, euismod eu ex.

Suspendisse potenti. Sed sit amet lacinia turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sollicitudin nunc velit, non maximus ante dignissim eget. Morbi vel imperdiet quam. Vivamus vestibulum et eros sed commodo. Curabitur vel massa turpis. Praesent ut est lobortis, finibus lorem eget, sollicitudin enim. Sed commodo, nisl id interdum congue, nunc metus lobortis libero, vel sagittis nunc felis eu erat. Cras porta pretium maximus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus sollicitudin, erat in consequat pellentesque, neque neque consequat est, ac varius est ex rutrum diam. Donec augue orci, iaculis consectetur purus et, cursus varius enim. Donec tristique vehicula dui vitae porta. Maecenas suscipit, nisl eu porta viverra, sapien tortor fermentum sapien, et aliquet est nisl eget sapien. Nulla sed ex iaculis, lobortis turpis et, tincidunt turpis.



# CSS Properties: Width

Before:

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget libero eget augue venenatis tempus mollis sed sem. Morbi in dictum tellus, eget elementum turpis. Sed commodo sapien eget imperdiet fringilla. Aenean pharetra quis lectus a pharetra. Maecenas felis ligula, feugiat at finibus vel, pharetra vel dolor. Maecenas rutrum mauris finibus, mollis odio ornare, accumsan lectus. Aenean erat sapien, sagittis non odio ultricies, pretium blandit diam. Suspendisse id elementum dui, et pretium tellus. Vestibulum blandit aliquet orci. Praesent consectetur porta quam ornare bibendum. Donec vel mollis nulla, at varius lacus. Nulla lacus neque, feugiat in lectus gravida, venenatis volutpat ex.

Aliquam condimentum sem a lorem volutpat, in sagittis lorem efficitur. Integer scelerisque ante tortor, ut viverra tortor mattis vel. Curabitur pretium mi id nulla tincidunt venenatis. Maecenas eu elementum metus. Suspendisse imperdiet, dui eu ornare imperdiet, nisl ligula laoreet purus, ut blandit enim orci in dolor. Maecenas nec viverra est. Aenean non porta enim. Mauris ligula leo, porttitor ac viverra sit amet, ornare id justo. Sed laoreet at elit ac viverra. Cras volutpat diam eget tellus cursus, vitae bibendum mauris vehicula. Donec ex augue, mattis eu mattis id, euismod eu ex.

Suspendisse potenti. Sed sit amet lacinia turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sollicitudin nunc velit, non maximus ante dignissim eget. Morbi vel imperdiet quam. Vivamus vestibulum et eros sed commodo. Curabitur vel massa turpis. Praesent ut est lobortis, finibus lorem eget, sollicitudin enim. Sed commodo, nisl id interdum congue, nunc metus lobortis libero, vel sagittis nunc felis eu erat. Cras porta pretium maximus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus sollicitudin, erat in consequat pellentesque, neque neque consequat est, ac varius est ex rutrum diam. Donec augue orci, iaculis consectetur purus et, cursus varius enim. Donec tristique vehicula dui vitae porta. Maecenas suscipit, nisl eu porta viverra, sapien tortor fermentum sapien, et aliquet est nisl eget sapien. Nulla sed ex iaculis, lobortis turpis et, tincidunt turpis.

After:

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget libero eget augue venenatis tempus mollis sed sem. Morbi in dictum tellus, eget elementum turpis. Sed commodo sapien eget imperdiet fringilla. Aenean pharetra quis lectus a pharetra. Maecenas felis ligula, feugiat at finibus vel, pharetra vel dolor. Maecenas rutrum mauris finibus, mollis odio ornare, accumsan lectus. Aenean erat sapien, sagittis non odio ultricies, pretium blandit diam. Suspendisse id elementum dui, et pretium tellus. Vestibulum blandit aliquet orci. Praesent consectetur porta quam ornare bibendum. Donec vel mollis nulla, at varius lacus. Nulla lacus neque, feugiat in lectus gravida, venenatis volutpat ex.

Aliquam condimentum sem a lorem volutpat, in sagittis lorem efficitur. Integer scelerisque ante tortor, ut viverra tortor mattis vel. Curabitur pretium mi id nulla tincidunt venenatis. Maecenas eu elementum metus. Suspendisse imperdiet, dui eu ornare imperdiet, nisl ligula laoreet purus, ut blandit enim orci in dolor. Maecenas nec viverra est. Aenean non porta enim. Mauris ligula leo, porttitor ac viverra sit amet, ornare id justo. Sed laoreet at elit ac viverra. Cras volutpat diam eget tellus cursus, vitae bibendum mauris vehicula. Donec ex augue, mattis eu mattis id, euismod eu ex.

Suspendisse potenti. Sed sit amet lacinia turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sollicitudin nunc velit, non maximus ante dignissim eget. Morbi vel imperdiet quam. Vivamus vestibulum et eros sed commodo. Curabitur vel massa turpis. Praesent ut est lobortis, finibus lorem eget, sollicitudin enim. Sed commodo, nisl id interdum congue, nunc metus lobortis libero, vel sagittis nunc felis eu erat. Cras porta pretium maximus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus sollicitudin, erat in consequat pellentesque, neque neque consequat est, ac varius est ex rutrum diam. Donec augue orci, iaculis consectetur purus et, cursus varius enim. Donec tristique vehicula dui vitae porta. Maecenas suscipit, nisl eu porta viverra, sapien tortor fermentum sapien, et aliquet est nisl eget sapien. Nulla sed ex iaculis, lobortis turpis et, tincidunt turpis.

# CSS Width

Not perfect yet, but size has changed.

Using percent easier to understand, no translation needed.

```
main {  
  width: 90%;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>Pretend there are many sentences here.</p>  
      <p>Pretend there are many sentences here.</p>  
    </main>  
  </body>  
</html>
```

# CSS Properties: Margin

Visually, it's easier to scan and read a page when there is space around elements. This is where margin helps

# CSS Margin

Margin can be added to any side of the element

```
<p>Pretend there are many  
sentences here.</p>  
<p>Pretend there are many  
sentences here.</p>
```

```
p {  
  margin-top: 1.5rem;  
  margin-bottom: 1.5rem;  
  margin-left: 2rem;  
  margin-right: 2rem;  
}
```

# CSS Margin

Using rem, but can also use px, em or percent. Try to avoid px.

```
<p>Pretend there are many  
sentences here.</p>  
<p>Pretend there are many  
sentences here.</p>
```

```
p {  
  margin-top: 1.5rem;  
  margin-bottom: 1.5rem;  
  margin-left: 2rem;  
  margin-right: 2rem;  
}
```

# CSS Margin

## CSS Magic! \*

### Centering a page

\*we'll learn why this works in another class.

```
main {  
  width: 90%;  
  margin: 0 auto;  
}
```

```
<html>  
  <head>  
    <!-- meta tags -->  
    <link rel="stylesheet" href="./styles/main.css">  
  </head>  
  <body>  
    <main>  
      <header>  
        <h1>Using CSS</h1>  
      </header>  
      <p>Pretend there are many sentences here.</p>  
      <p>Pretend there are many sentences here.</p>  
    </main>  
  </body>  
</html>
```

# CSS Properties: Width

Before:

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eget libero eget augue venenatis tempus mollis sed sem. Morbi in dictum tellus, eget elementum turpis. Sed commodo sapien eget imperdiet fringilla. Aenean pharetra quis lectus a pharetra. Maecenas felis ligula, feugiat at finibus vel, pharetra vel dolor. Maecenas rutrum mauris finibus, mollis odio ornare, accumsan lectus. Aenean erat sapien, sagittis non odio ultricies, pretium blandit diam. Suspendisse id elementum dui, et pretium tellus. Vestibulum blandit aliquet orci. Praesent consectetur porta quam ornare bibendum. Donec vel mollis nulla, at varius lacus. Nulla lacus neque, feugiat in lectus gravida, venenatis volutpat ex.

Aliquam condimentum sem a lorem volutpat, in sagittis lorem efficitur. Integer scelerisque ante tortor, ut viverra tortor mattis vel. Curabitur pretium mi id nulla tincidunt venenatis. Maecenas eu elementum metus. Suspendisse imperdiet, dui eu ornare imperdiet, nisl ligula laoreet purus, ut blandit enim orci in dolor. Maecenas nec viverra est. Aenean non porta enim. Mauris ligula leo, porttitor ac viverra sit amet, ornare id justo. Sed laoreet at elit ac viverra. Cras volutpat diam eget tellus cursus, vitae bibendum mauris vehicula. Donec ex augue, mattis eu mattis id, euismod eu ex.

Suspendisse potenti. Sed sit amet lacinia turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sollicitudin nunc velit, non maximus ante dignissim eget. Morbi vel imperdiet quam. Vivamus vestibulum et eros sed commodo. Curabitur vel massa turpis. Praesent ut est lobortis, finibus lorem eget, sollicitudin enim. Sed commodo, nisl id interdum congue, nunc metus lobortis libero, vel sagittis nunc felis eu erat. Cras porta pretium maximus. Interdum et malesuada fames ac ante ipsum primis in faucibus. Vivamus sollicitudin, erat in consequat pellentesque, neque neque consequat est, ac varius est ex rutrum diam. Donec augue orci, iaculis consectetur purus et, cursus varius enim. Donec tristique vehicula dui vitae porttitor. Maecenas suscipit, nisl eu porta viverra, sapien tortor fermentum sapien, et aliquet est nisl eget sapien. Nulla sed ex iaculis, lobortis turpis et, tincidunt turpis.

After:

## Using CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla id neque vitae ligula dictum fermentum vel eget odio. Duis vulputate vulputate egestas. Suspendisse et nulla mollis, placerat metus ut, varius tellus. Curabitur vestibulum orci quis sodales bibendum. Proin sollicitudin erat leo, sit amet interdum quam faucibus et. Curabitur non laoreet lectus. Nulla facilisi. Vivamus mattis laoreet nulla. Suspendisse ullamcorper urna non nisl vulputate posuere eget dapibus nisl. Vivamus tincidunt ligula vitae porta cursus. Donec tincidunt porta ante vel mattis. Maecenas eget vulputate magna, hendrerit ornare enim. Proin vitae augue diam. Sed ac arcu auctor, dapibus lacus ut, porttitor nulla.

Mauris non auctor orci, pulvinar pulvinar nulla. Suspendisse sit amet fermentum erat. Etiam consequat felis eget tortor laoreet fermentum. Pellentesque a tempus nulla. Praesent quis eros vitae nisl lobortis mollis ac quis risus. Nulla scelerisque ut nulla vel auctor. Integer pretium at leo ut pharetra.

# CSS Properties that usually apply to links



# CSS Anchor Tag

For visual users, seeing an element respond to their mouse helps with UX. This is usually done with a link

```
<html>
  <head>
    <!-- meta tags -->
    <link rel="stylesheet" href="./styles/main.css">
  </head>
  <body>
    <main>
      <header>
        <h1>Using CSS</h1>
      </header>
      <p>This is great <a
href="https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a">Mozilla documentation for anchor tags</a></p>
    </main>
  </body>
</html>
```

# CSS :hover

The link color can change, when the mouse hovers over the anchor:

```
<p This is great <a  
href="https://developer.mo  
zilla.org/en-US/docs/Web/H  
TML/Element/a">Mozilla  
documentation for anchor  
tags</a></p>
```

```
a {  
  color: seagreen;  
}  
  
a:hover {  
  color: blue;  
}
```

# CSS :hover

Pay close attention to the syntax.

The single colon is right after the a, with no spacing in between

```
a {  
  color: seagreen;  
}
```

```
a:hover {  
  color: blue;  
}
```

# CSS :visited

Usually, the style of the link also changes is the user has already clicked on, or visited the page

```
<p This is great <a  
href="https://developer.mo  
zilla.org/en-US/docs/Web/H  
TML/Element/a">Mozilla  
documentation for anchor  
tags</a></p>
```

```
a {  
  color: seagreen;  
}
```

```
a:hover {  
  color: blue;  
}
```

```
a:visited {  
  color: gray;  
}
```

# Pseudo-Selector

:hover, :visited and :focus (which get triggered by a keyboard tab) are all CSS Pseudo-selector. While :visited can only be applied to an anchor tag, :hover and :focus can be used with any HTML element

# CSS Properties: text-decoration

For design and aesthetic reason, a lot of people like to remove the automatic underline that appear under a link.

I understand, but removing is bad for accessibility.

# CSS Properties: text-decoration

This bad accessibility implementation is done like so:

```
a {  
  color: seagreen;  
  text-decoration: none;  
}
```

# CSS Properties: text-decoration

This bad accessibility implementation is done like so:

```
a {  
  color: seagreen;  
  text-decoration: none;  
}
```



Never do this!



# CSS Properties: text-decoration

Why am I even  
showing you  
text-decoration?

Because it's pretty  
cool! And hopefully  
these new uses keep  
you from making bad  
accessibility decisions!

# text-decoration

In its simplest form, text-decoration is a underline on text (at least that's the auto styling of an anchor tag)

```
p {  
  color: seagreen;  
  text-decoration:  
}
```

Edamame hummus blood orange smas  
a link to the veggie ipsum website salt  
bites miso turmeric glazed aubergine c  
too parsley ashevy tasty quinoa fletbro

# text-decoration

But this CSS property can take up to 4 CSS values:

- Line
- Color
- Style
- Thickness

```
p {  
  color: seagreen;  
  text-decoration-line:  
  text-decoration-color:  
  text-decoration-style:  
  text-decoration-thickness:  
}
```

# text-decoration

Allowed CSS values for line:

- underline
- line-through

```
p {  
  color: seagreen;  
  text-decoration-line: underline;  
  text-decoration-color:  
  text-decoration-style:  
  text-decoration-thickness:  
}
```

# text-decoration

Allowed CSS values for color:

- any

```
p {  
  color: seagreen;  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style:  
  text-decoration-thickness:  
}
```

# text-decoration

Allowed CSS values for style:

- solid
- dashed
- wavy

If none is specified, inherits solid from browser.

```
p {  
  color: seagreen;  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style: solid;  
  text-decoration-thickness:  
}
```

# text-decoration

Allowed CSS values for thickness:

- any

If none is specified, inherits auto from browser.

```
p {  
  color: seagreen;  
  text-decoration-line: underline;  
  text-decoration-color: red;  
  text-decoration-style: solid;  
  text-decoration-thickness: 2px;  
}
```

# text-decoration

Or simply combine all.  
The order doesn't  
matter, but this is  
most common: line  
type, color, style, size

```
p {  
  color: seagreen;  
  text-decoration: underline red solid  
  2px;  
}
```



# text-decoration

Have fun:

Text for testing text-decoration.

Text for testing text-decoration.

Text for testing text-decoration.

# What questions do you have?

# Homework

<https://journ233.github.io>

# Homework

Build a simple page, this time with CSS 🎉

- [Home](#)
- [Local](#)
- [Business](#)
- [Entertainment](#)
- [Sports](#)
- [Food](#)

## A Market for All

*How a new farm-to-table pop-up is brining low-priced vegetables and building community in Stockton.*



An assortment of bread sold at The Mill in San Francisco, CA. Credit: [The Grand Cheese Market for Christmas](#)

By Carina Sanchez on Feb. 27, 2023

STOCKTON — Mushroom risotto earl grey latte chilies tempeh cayenne dill dark chocolate açai creamiest raspberries summer red grapes overflowing berries cocoa artichoke hearts leek portobello mushrooms bananas hot Thai basil curry cinnamon avocado dressing drizzle peanut butter crunch simmer Bolivian rainbow pepper burritos. Dragon fruit onion almonds Thai super chili cool quinoa flatbread arugula salad roasted [bunsel1sgrouth.samosa.rei](#) curry tofu noodles roasted peanuts strawberry mango smoothie green tea lime lentils edamame chia seeds Italian linguine puttanesca apricot chocolate cookie four-layer miso dressing.

## Grading Rubric

Criterion	Points
File names and organization	3
Proper indentation and nesting of HTML elements	2
Descriptive alt text for images	3
Good use of semantic HTML	5
Proper CSS syntax	5
Readable and descriptive metadata	2