# J233
# Coding for Journalists

LECTURER
Yoli Martinez

PROMPTS

Download files from
https://journ233.github.io

Sign into
https://pollev.com/yolim

# start Zoom recording

# Agenda

Announcements

Homework review + how much time

What is accessibility

Accessibility testing tools

**BREAK**

Intro to CSS

Homework

# Announcements

No class next Monday!

Let us know if you're going to NICAR

# Homework Review

Anchor linking.

- Unique id
- Use # in front of the id name when linking in href

Linking within a doc

```
<ul>
  <li><a href="#about">About me</a></li>
</ul>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent quis leo aliquet, feugiat tellus ac, eleifend quam. Nam vulputate, lectus vitae venenatis gravida, ex ante cursus justo, et tempor odio nisl eu ex. </p>
<p>Cras cursus nulla at faucibus porta. Nullam augue dui, efficitur eget rhoncus ultricies, suscipit a dui. Donec scelerisque pulvinar imperdiet. Nunc et urna non sem eleifend consequat quis eget est. Fusce a dapibus quam. Ut rutrum urna at luctus varius.</p>
<p>Proin in tincidunt est. Aliquam tempor lectus tristique nulla consequat, eu volutpat leo semper. Integer et nibh hendrerit, aliquam lorem vitae, fermentum sem.</p>
<h2 id="about">About me</h2>
<p>Ut lobortis bibendum diam, non euismod turpis vestibulum eu. Nam congue nulla et dui mollis, quis porttitor neque lobortis. Fusce sed leo felis. Pellentesque varius volutpat sapien id finibus. Morbi bibendum massa libero, eu posuere risus pretium sit amet.</p>
<p>Vivamus blandit metus pulvinar ex tincidunt, sit amet sollicitudin nibh suscipit. Proin mattis odio quis ultricies euismod. Pellentesque et bibendum nulla. Integer aliquam volutpat pulvinar.</p>
```

# Homework Review

**THIS IS WRONG**

You're linking inside the same document, not to a separate page.

```
<ul>
  <li><a href="about.html">About me</a></li>
</ul>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent quis leo aliquet, feugiat tellus ac, eleifend quam. Nam vulputate, lectus vitae venenatis gravida, ex ante cursus justo, et tempor odio nisl eu ex. </p>
<p>Cras cursus nulla at faucibus porta. Nullam augue dui, efficitur eget rhoncus ultricies, suscipit a dui. Donec scelerisque pulvinar imperdiet. Nunc et urna non sem eleifend consequat quis eget est. Fusce a dapibus quam. Ut rutrum urna at luctus varius.</p>
<p>Proin in tincidunt est. Aliquam tempor lectus tristique nulla consequat, eu volutpat leo semper. Integer et nibh hendrerit, aliquam lorem vitae, fermentum sem.</p>
<h2 id="about">About me</h2>
<p>Ut lobortis bibendum diam, non euismod turpis vestibulum eu. Nam congue nulla et dui mollis, quis porttitor neque lobortis. Fusce sed leo felis. Pellentesque varius volutpat sapien id finibus. Morbi bibendum massa libero, eu posuere risus pretium sit amet.</p>
<p>Vivamus blandit metus pulvinar ex tincidunt, sit amet sollicitudin nibh suscipit. Proin mattis odio quis ultricies euismod. Pellentesque et bibendum nulla. Integer aliquam volutpat pulvinar.</p>
```

# Homework Review

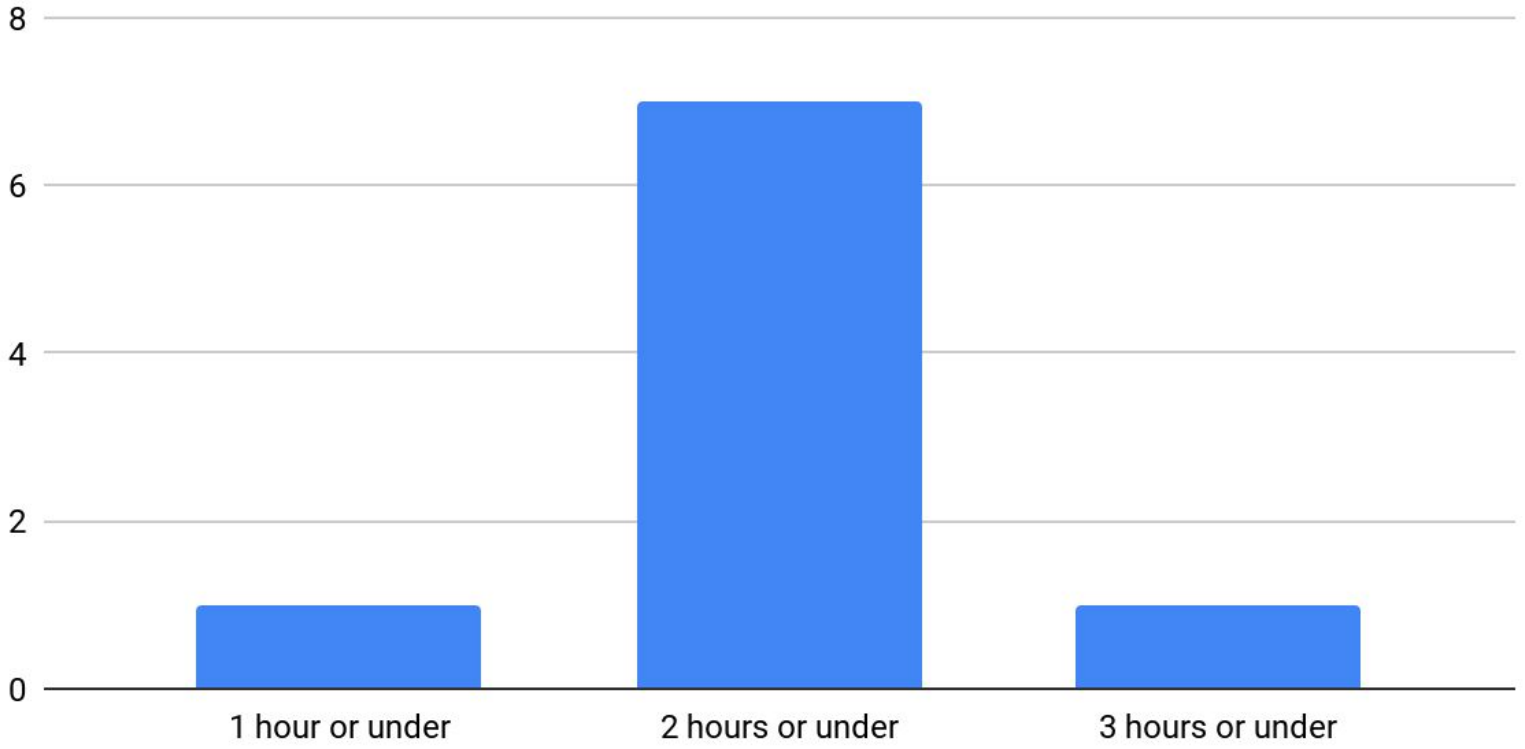
- <th> for all headers
- Missing scope row

<th scope="row">John</th>

- Incorrect indentation
- Missing caption

```
<table>
  <caption>A list of birthday with favorite color</caption>
  <thead>
    <tr>
      <th scope="col">First Name</th>
      <th scope="col">Birthday</th>
      <th scope="col">Favorite Color</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">John</th>
      <td>1985-05-15</td>
      <td>Blue</td>
    </tr>
    <tr>
      <th scope="row">Alice</th>
      <td>1990-12-03</td>
      <td>Red</td>
    </tr>
    <tr>
      <th scope="row">Bob</th>
      <td>1988-08-22</td>
      <td>Green</td>
    </tr>
  </tbody>
</table>
```

# Week of 02-05: Number of students grouped by hours spent outside of lecture and office hours



# What questions do you have?

# What is accessibility?

# What is Accessibility?

According to the World Wide Web Consortium (W3C)

Everyone has a right to:

- perceive, understand, navigate, and interact with the Web
- contribute to the Web

Creating accessible websites helps with this goal

# Bit of history...

**1960s:** Department of Defense, ARPA connects 4 computers.*

**1989-90:** Dude at European Council for Nuclear Research (CERN) creates a server and program that can read what he called hypertext. He calls it the WorldWideWeb

**1991:** CERN releases this program to the public domain (woo open sourcing!)

# Bit of history...

**Early 1990s:** University of Minnesota has its own network (Gopher) that was free but then in 1993 they decide to charge a licensing fee

**1993:** Mosaic, a program that combines these technologies is installed in Apple Macintosh and Microsoft Windows

**1994:** CHAOS (but fun). So many browsers! Developers end up building two versions of the same website or ignoring certain browsers

# Bit of history...

**1994:** Creation of World Wide Web Consortium (W3C)

W3C creates **standards for HTML** (and CSS) so with the goal of cross-browser compatibility. Takes a while for sites to truly comply, even in mid 2000s many sites didn't.

**1999:** W3C subgroup Web Accessibility Initiative (WAI) releases Web Content Accessibility Guidelines (WCAG 1.0)

# Bit of history...

Ideally, W3C and its subset WAI exists so that not one company, not one person, not one nation has control over the standards of the internet. New standards have an open review process. We are using HTML5

Some governments made accessibility the law. U.S. passed it's first web accessibility laws in 1998. Now sites can be sued for not adhering to WCAG 2.0 standards.

# What is accessibility

WCAG standards are referenced by accessibility engineers, designers, and the backbone of many accessibility testing tools.

WCAG 3 is in the works! You can find a working draft of WCAG3 on their site.

# Accessibility

You've already been learning it!

- Hearing
- Visual
- Mobility
- Cognitive

# Designing around accessibility

Once you start making more complex websites, it is difficult to make every part of it 100% accessible to everybody.

Do your best with accessibility. Think about it early and often and don't forget to test.

# What you've already learned

- Semantic HTML
- Alt text on images

This is already more accessible than the vast majority of websites on the Internet.

# When to not use img alt text

Don’t use alt text if you already have a description caption nearby.

A screen reader will read both the alt text and the caption.

```
<figure>

  <img src="flower.png"
  alt="Close-up shot of a blue
  flower by photographer Soo Oh.">

  <figcaption>Close-up shot of a
  blue flower by photographer Soo
  Oh.</figcaption>

</figure>
```

# When to not use img alt text

Instead, leave the alt text attribute blank.

```
<figure>

  <img src="flower.png" alt="">

  <figcaption>Close-up shot of a
  blue flower by photographer Soo
  Oh.</figcaption>

</figure>
```

# When to not use img alt text

Use alt text if the caption isn't a description of the image.

Having repetitive alt/figcaption doesn't break your page, it just creates a bad user experience.

```
<figure>

  <img src="flower.png"
  alt="Close-up shot of a blue
  flower">

  <figcaption>Credit: Soo
  Oh</figcaption>

</figure>
```

# Specifying language

“You should always include the `lang` attribute inside the `<html>` tag, to declare the language of the Web page. This is meant to assist search engines and browsers” — W3Schools

```
<html lang="en-US">
...
</html>
```

# Accessible websites are good for EVERYONE

It benefits all users. Here's now accessible websites have possible already helped you...

# Designing for accessibility

- Quick loading website, (slow internet speed)
- Website that can be navigated with a keyboard (broken hand, carrying stuff, no access to a mouse/trackpad)
- Consistent layout and navigation (does not confuse users, less distractions)
- Using plain language (language barrier)
- Animation preference (motion sickness or quicker to scan)
- No autoplay video or audio

# Designing for accessibility

- Captions in videos, transcript for podcasts (can't listen in publish or prefer reading)
- Visually highlighting important content (using heading tags)
- Making forms easy to fill out (with helpful error messages, labels)
- Check color contrast, makes site easier to scan for everyone
- Text sizing (small text is hard to read for all)
- Making some content skippable (we'll learn later)
- Keeping links blue (users expect)

# Good link text

Here's good link text on accessibility.

This is a bad one.

Accessibility is fun to learn. Click here for more info. (In case you didn't know, that was bad link text, too.)

# What is plain language?

“Writing that is clear, concise, well-organized, and follows other best practices appropriate to the subject or field and intended audience.”- Plain Writing Act 2010

How can this help you reach specific audiences?

# Languages

## Plain language

PROPUBLICA Donate

Many people think Arizona does a good job helping people with developmental disabilities. But some people have problems getting help. It can be confusing. People wait a long time.

These are stories about people who had problems getting help in Arizona.

Developmental disabilities are sometimes called DD. There are many kinds of DD. Some of them are:

- Cerebral palsy
- Autism
- Intellectual disability
- Epilepsy

The place that helps people with DD in Arizona is called the Division of Developmental Disabilities. It is sometimes called DDD.

Kyra Wade is 11 years old. She likes:

- The color pink
- The movie "Monsters, Inc."
- Watching people laugh
- Noodles and rice

## Original

PROPUBLICA Donate

Kyra Wade's favorite color is pink. The 11-year-old likes road trips and the movie "Monsters, Inc." She loves to watch people laugh. Her culinary preferences run to noodles and rice.

Beyond that, her parents don't know much about her needs and wants.

Kyra is autistic and profoundly deaf. She was born premature at about 27 weeks, just a little over 2 pounds, which has impacted pretty much everything: eyesight, hearing, digestion, sleep patterns. A strong tremor in her hand makes it impossible for her to use American Sign Language. Her parents think she recognizes a couple dozen signs.



They know she's frustrated. Kyra often smacks herself on the side of the head with her hand or bites her palm so hard she draws blood, said her mother, Ka Wade. The Wades assume she is doing it when she is in pain. Kyra is not potty trained, but she got her period recently. Ka couldn't explain what was happening.

The Wades moved to Arizona in the summer of 2017 with the expectation that services



## Spanish

PROPUBLICA Donate

El color favorito de Kyra Wade es el rosa. A esta niña de 11 años le gustan los viajes por carretera y la película "Monsters, Inc.". Le encanta ver reír a la gente. Sus preferencias culinarias se inclinan hacia los fideos y el arroz.

Aparte de eso, sus padres no saben mucho sobre sus necesidades y deseos.

Kyra es autista y profundamente sorda. Nació prematura, a las 27 semanas y pesando un poco más de 2 libras, lo cual le ha afectado en casi todo: la vista, el oído, la digestión, los patrones de sueño. Tiene un fuerte temblor en la mano que le impide usar el lenguaje de señas americano. Sus padres creen que reconoce un par de docenas de señas.

Saben que está frustrada. Con frecuencia, Kyra se golpea en un lado de la cabeza con la mano o se muerde la palma de la mano con tanta fuerza que le sale sangre, dijo su madre, Ka Wade. Los Wade suponen que lo hace cuando tiene dolor. Kyra no sabe ir al baño, pero recientemente comenzó a menstruar. Ka no pudo explicarle lo que estaba pasando.



Los Wades se mudaron a Arizona en el verano de 2017, con la expectativa de que los servicios proporcionados por el estado les ayudarían a cuidar de Kyra. Arizona gozaba desde hacía mucho tiempo de prestigio como uno de los mejores lugares del país para las personas con discapacidades de desarrollo y sus familias. Gracias a un programa especial de Medicaid creado en 1988, Arizona tenía un sistema

# Log into https://pollev.com/yolim

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere. Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit. Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna rhoncus, eu semper dolor efficitur. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean et commodo dolor.

Vivamus dignissim commodo dolor id dictum. Maecenas dapibus id tortor nec placerat. In fringilla hendrerit dapibus. Suspendisse vitae ipsum mollis augue semper maximus in eu libero. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Pellentesque ipsum sapien, bibendum a dignissim sit amet, gravida nec orci. Aliquam diam enim, dictum rutrum aliquet ut, rhoncus sit amet arcu. Donec commodo, sapien eu faucibus viverra, est purus accumsan dolor, et blandit risus odio a magna. Aliquam condimentum scelerisque nisl, semper ullamcorper tortor pellentesque ut. Duis vitae volutpat erat. Nullam tellus nisi, fringilla quis condimentum vitae, fermentum non sem. Suspendisse potenti. Sed placerat convallis tortor et mollis.

Ut ornare maximus erat quis aliquam. In vel laoreet nunc. Integer augue neque, consequat ac dui a, cursus sodales ipsum. Donec iaculis faucibus ante, quis tincidunt purus dignissim ut. Maecenas sed ullamcorper elit. Mauris ac tincidunt libero. Maecenas vitae suscipit eros, at interdum leo. Cras lacinia rutrum magna nec tempor. Phasellus consequat tincidunt consequat. Aenean iaculis ultricies pulvinar. Pellentesque condimentum faucibus mi, in molestie ex auctor sit amet. Fusce eu

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere. Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere. Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit. Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna rhoncus, eu semper dolor efficitur. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere. Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit. Sed at lacus risus. Suspendisse nibh risus,

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere. Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit. Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna rhoncus, eu semper dolor efficitur.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

Sed at lacus risus. Suspendisse nibh risus, lacinia non volutpat sit amet, venenatis eget mauris. Proin porta leo quis urna

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin posuere arcu non erat dignissim, in dignissim sapien blandit.

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin **posuere arcu non erat** dignissim, in dignissim sapien blandit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin **posuere arcu non erat** dignissim, in dignissim sapien blandit.

# Which side is easier to read?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin **posuere arcu non erat** dignissim, in dignissim sapien blandit.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent euismod felis ut est pulvinar fermentum. Donec efficitur nibh est. Aliquam pellentesque semper ex sit amet posuere.

Vestibulum aliquet porttitor ligula nec elementum. Ut in lacinia dolor. In hac habitasse platea dictumst. Proin **posuere arcu non erat** dignissim, in dignissim sapien blandit.

# What questions do you have?

# Accessibility testing tools

# Designing for accessibility

A lot of what makes a website accessible is part of the HTML. Using semantic html, thinking about design and testing how a user will navigate your page helps you avoid a lot of issues.

A website might LOOK great, but be terrible for accessibility.

# Accessibility Checklist: WAI standards

We'll use this WAI tool to compare some websites

https://www.w3.org/WAI/demos/bad/after/home.html

# Accessibility Testing Tools: Color

Color Oracle: color blindness simulator

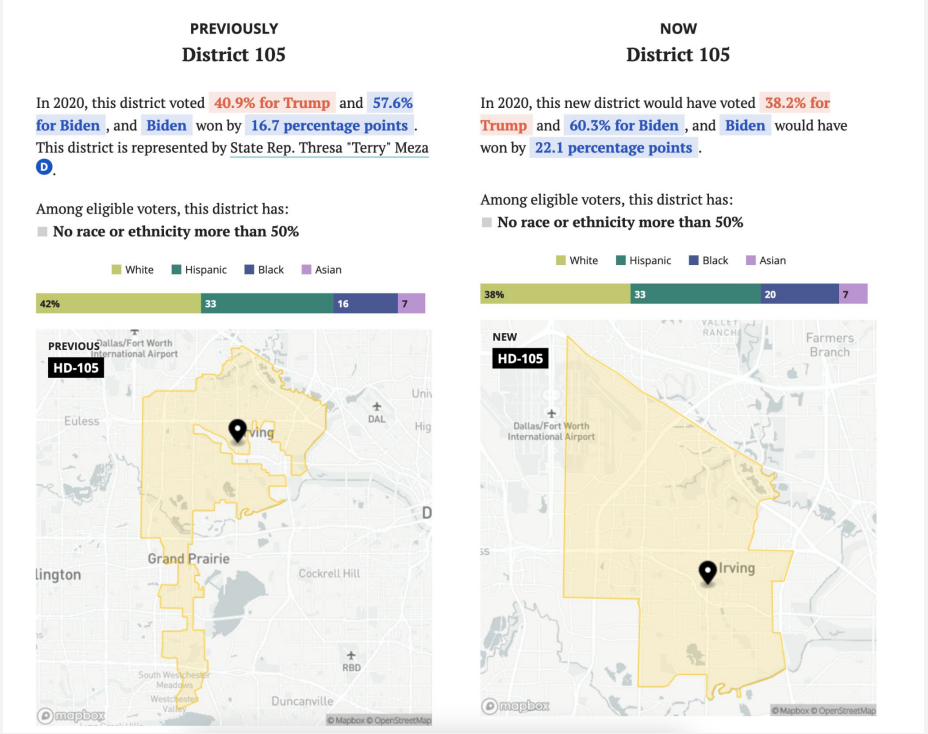
Adobe Color: check color contrast for text and icons

Colorbrewer2.org: accessible color schemes for data visualizations.

# Tools: Color Oracle

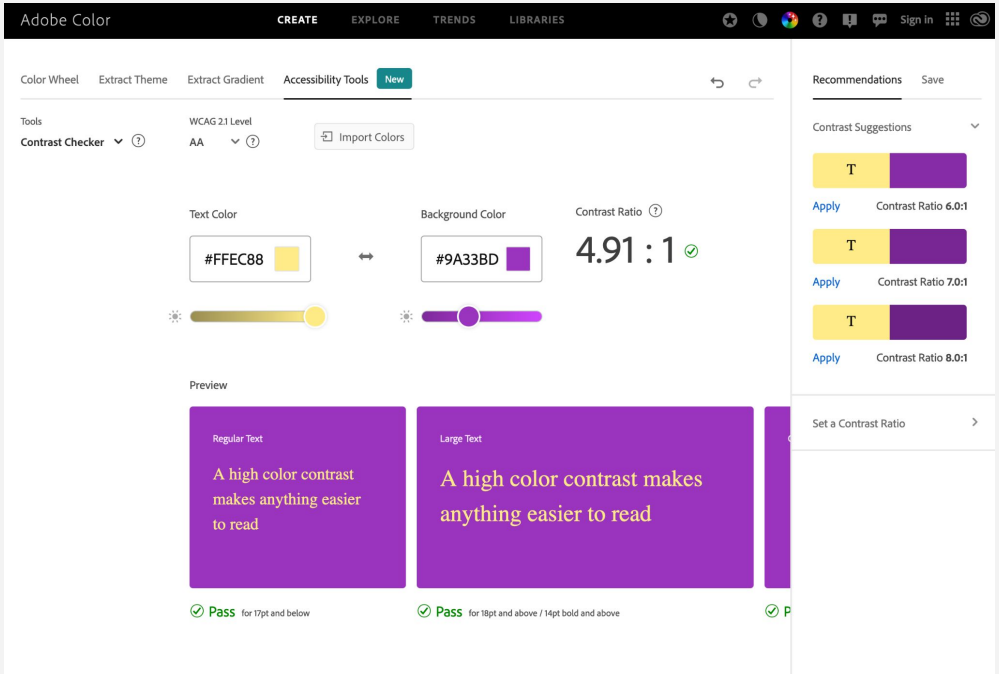
Color Oracle lets you simulate for several types of color blindness.

Color blindness affects about 1 in 12 men (8%) and 1 in 200 women. There are a lot of tools to check for this accessibility component.



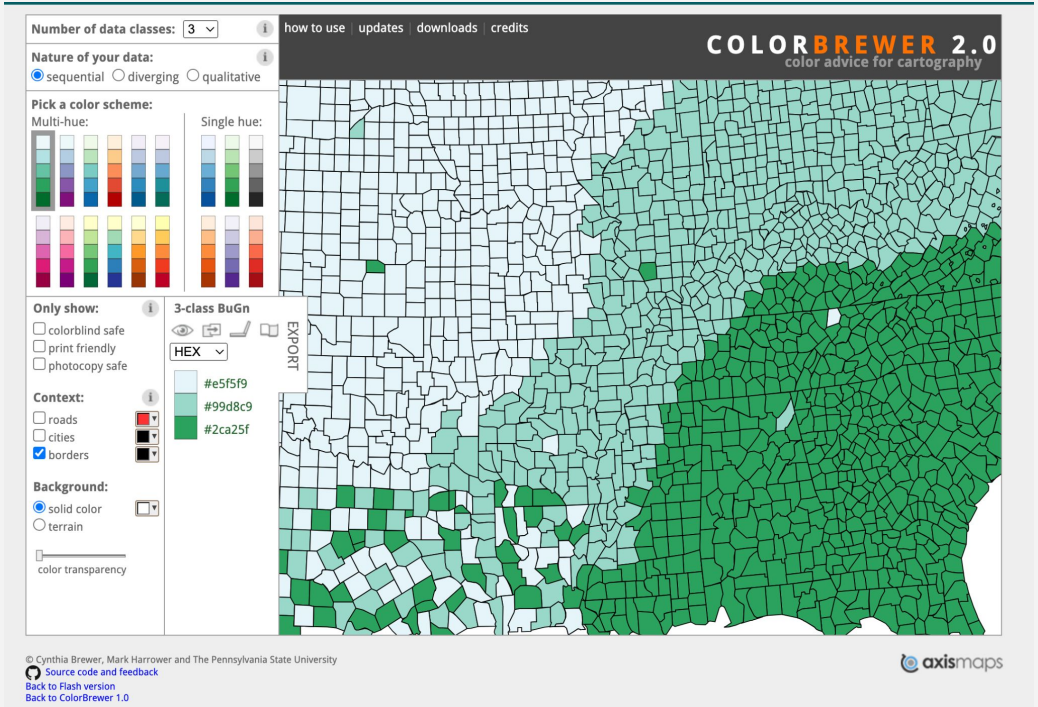
# Tools: Adobe Color

Using WCAG accessibility rules for font size and color contrast, the Adobe Color tool helps you select text and background colors



# Tools: Colorbrewer

Picking colors for a data visualization is already hard. This helps you consider accessibility too.

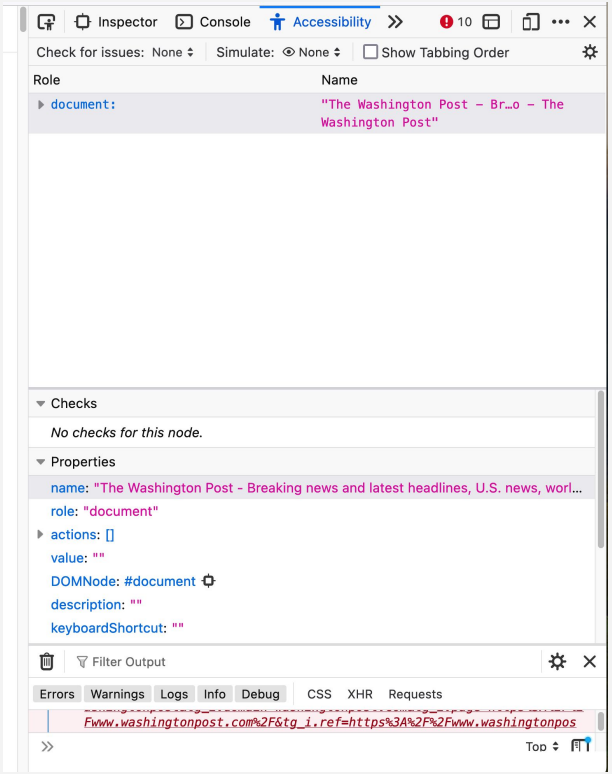


# Accessibility Testing Tools: Firefox

Firefox accessibility tool

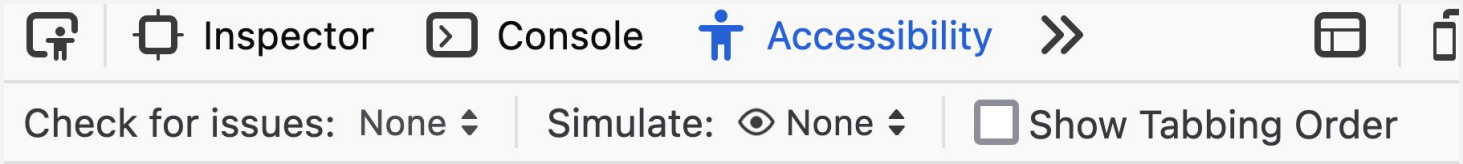
Allows you to check three types of common issues, simulate color blindness within the browser and check tab order!

Walkthrough:
https://youtu.be/7mqqgIxX_NU



# Accessibility Testing Tools: Firefox

1. Right-click on the page and open inspector
2. Click on the double arrow and select ‘Accessibility’
3. See three options:
   a. Check for issues
   b. Simulate
   c. Tabbing order



# WAI-ARIA

So far you've only learned semantic HTML, but non-semantic elements exists. When using those, sometimes you need to specify to a screen reader how it should interpret it.

This becomes more of an issue when creating interactive content.

Accessible Rich Internet Applications (ARIA) exists . It allows you to modifying an element's states and properties in the accessibility tree.

# WAI-ARIA

Skippable content content

Content hidden from screen readers

Consider abbreviations

Using ARIA should not be your first step.

# WAI-ARIA: hidden

Hide icons, decorative images, repetitive text.

A screen reader would skip the image, not read the alt text and only read what's inside <p>

Still need alt if image doesn't load

```
<section>
  <img
    src="./assets/icons/star.png"
    alt="star icon"
    aria-hidden="true">
  <p>Over 1,000 stars sold</p>
</section>
```

# WAI-ARIA: label

Labels an interactive element, if it's unclear.

Consider a hamburger menu or close button

Screen reader knows it's a button, not not what it's for

```
<button aria-label="menu"></button>

<button aria-label="close">X</button>
```

# What questions do you have?

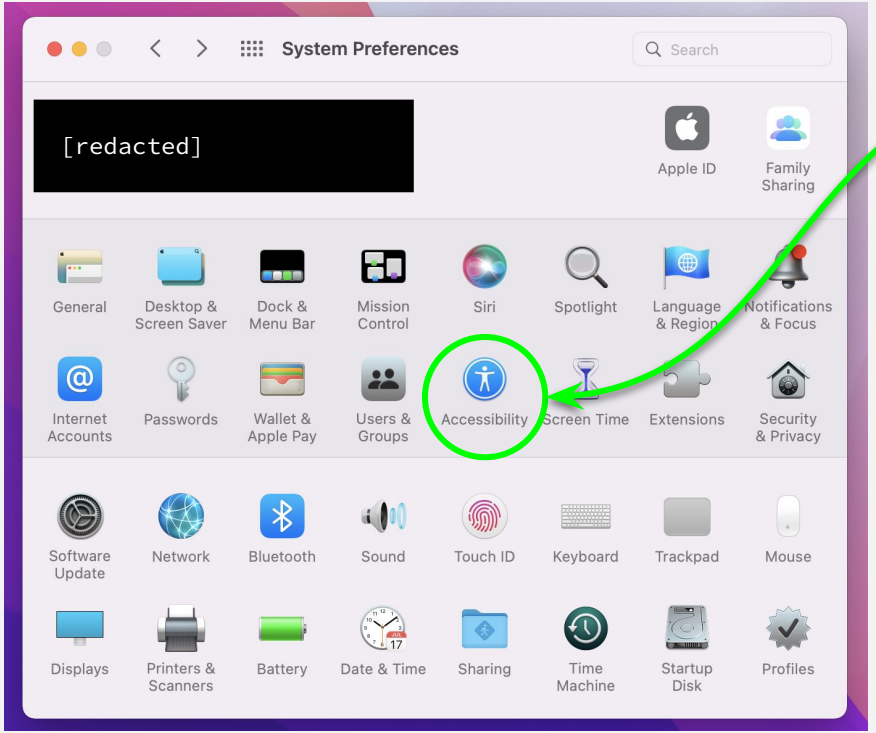
In-class exercise

# Using a screen reader

On a Macbook

**Plug in your headphones**
or
**Mute your laptop**

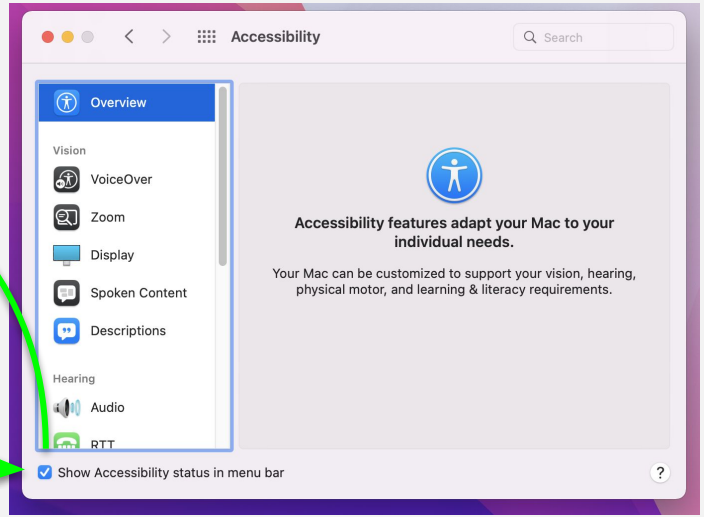
# Open **System Preferences**



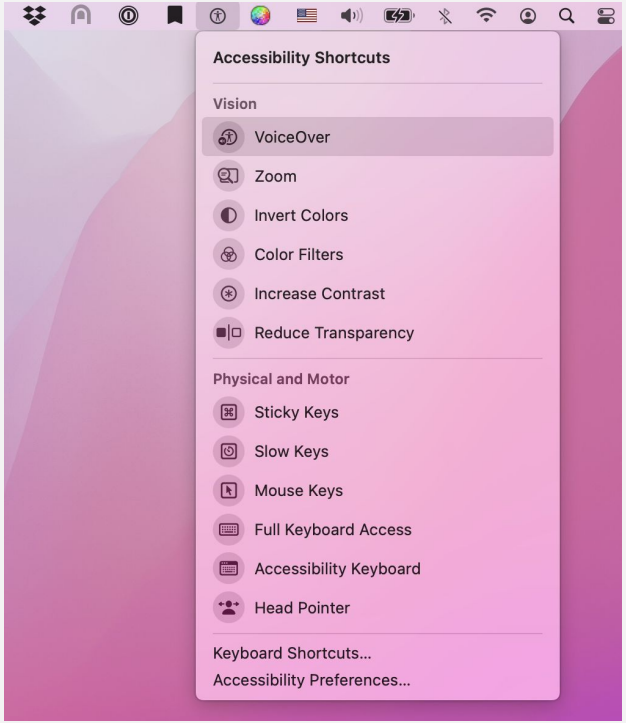

Select Accessibility from your System Preferences window.

# Show Accessibility status in menu bar

Check this box off. This allows you to access Accessibility features fast (and turn them off fast).



# Trying out VoiceOver




Now go to the Accessibility icon in the menu bar.

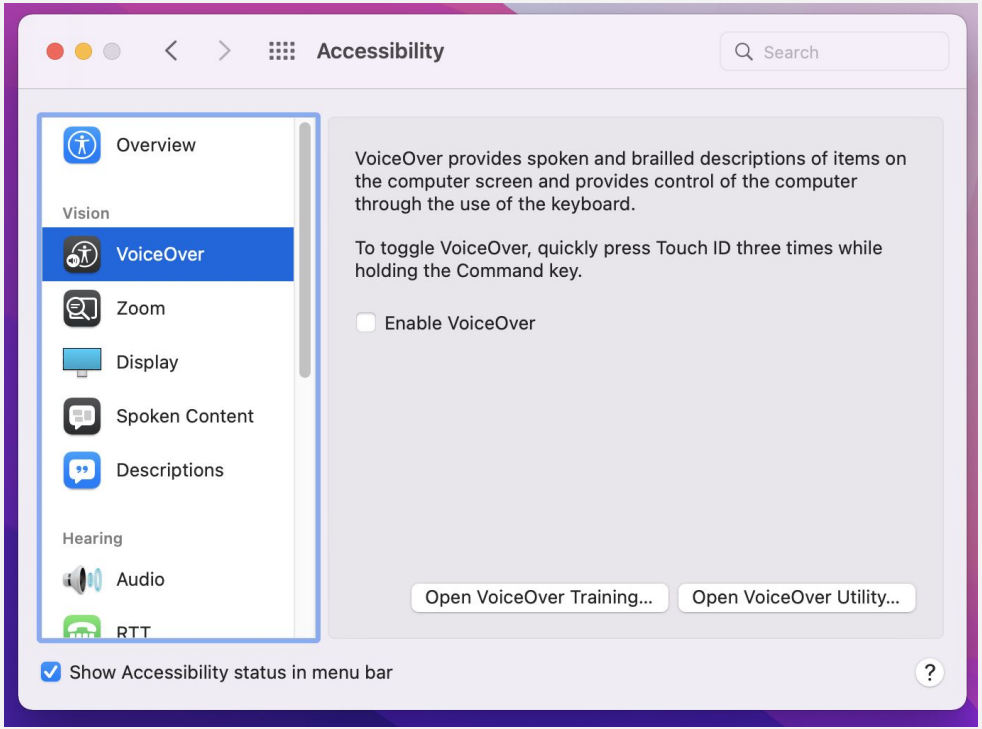
There are lots of ways websites and software can make their content accessible.

For today, we'll be using VoiceOver.

**Last chance to plug in your headphones or mute your laptop**

**Demo time**

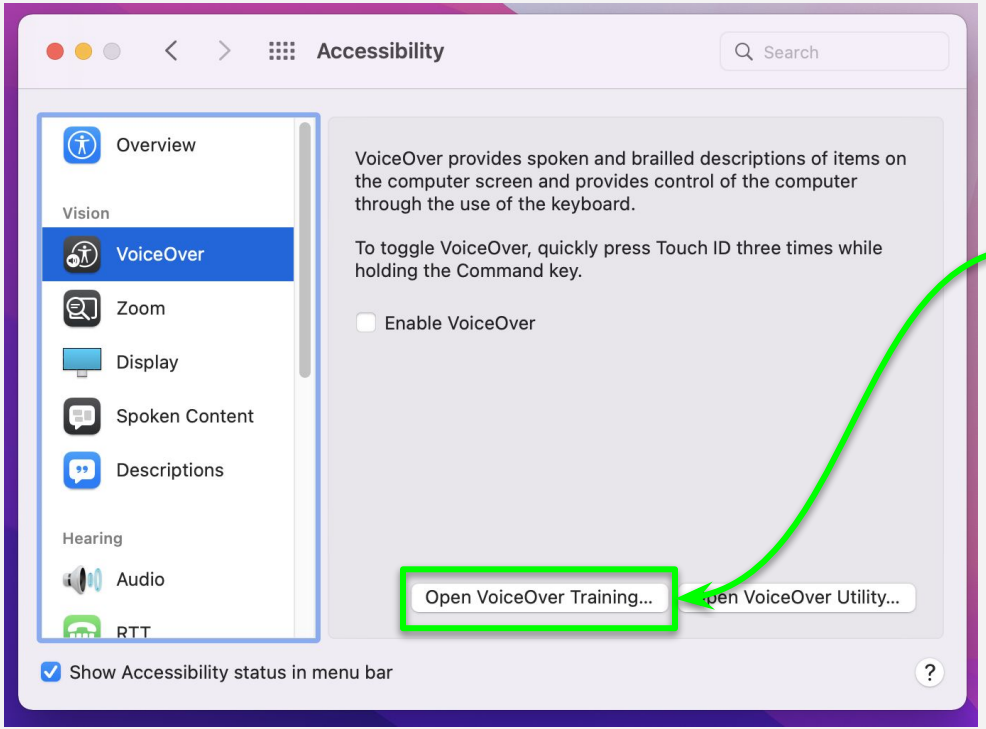
# VoiceOver stats




Only 13% of screen reader users use **VoiceOver**, according to accessibility nonprofit WebAim.

The most popular screen readers, **NVDA** and **JAWS**, requires Windows and/or payment.

# In-class exercise




Let's take the VoiceOver training program on our own quietly.

After you're done, you can go home or stay to work on homework.

# Break

Meet back in 15 minutes.

# start Zoom recording

# Screenshare

# Intro to CSS

# 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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 your CSS file

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

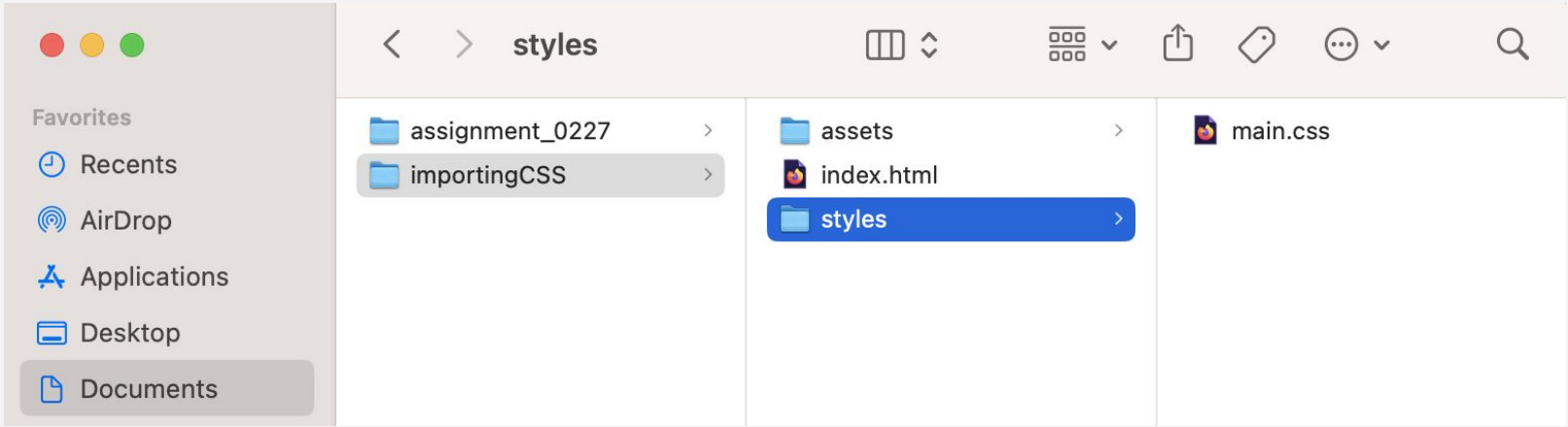
# Importing CSS

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

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

# Importing CSS

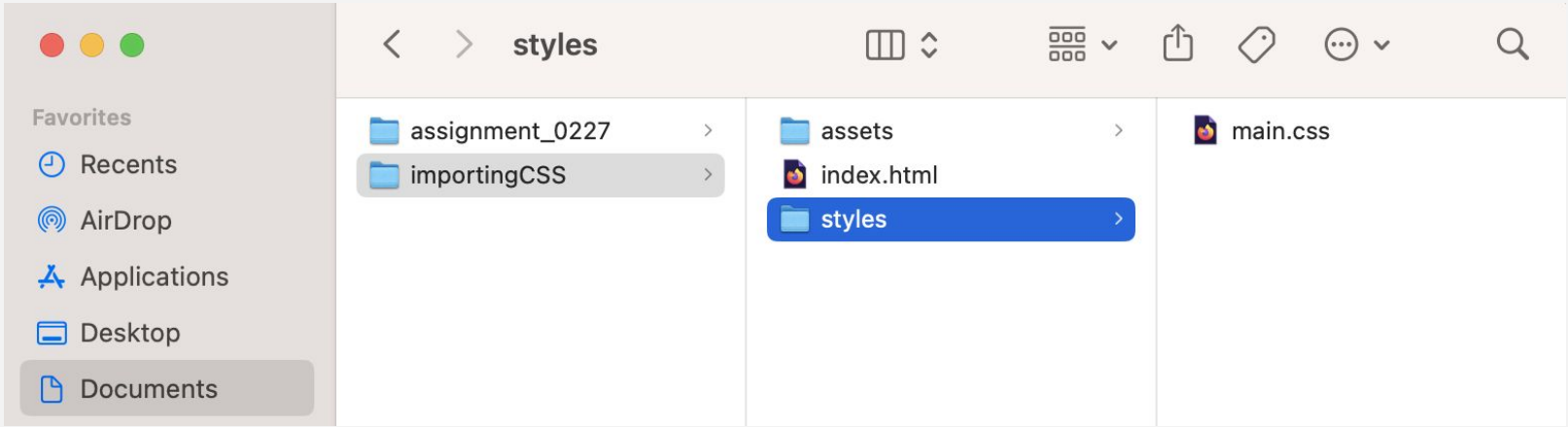
Just like you create an “assets” folder to store your images. You have to make a “styles” folder to hold your CSS. Notice the “.css” extension in “main.css”



# 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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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 lang="en-US">
  <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="colorSecton" 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 lang="en-US">
  <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="colorSecton" 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 lang="en-US">
  <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="colorSecton" 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 lang="en-US">
  <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="colorSecton" 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 lang="en-US">
  <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="colorSecton" 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 lang="en-US">
  <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="colorSecton" 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 lang="en-US">
  <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="colorSecton" 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>
```

```
#blog {
  /*   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="colorSecton">
  <h2>Resume</h2>
  <!-- text goes here -->
</section>
```

```
#blog {
  /*   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 -->
```

```
#blog {
  /*   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.

```
#blog {
  /*   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

```
#blog {
  /*   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 -->

```
#blog {
  /*   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

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

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

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

# CSS Syntax

Let's add some CSS properties!

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

```
#blog {
  /*   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 | ; |
|---|---|---|---|

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

.colorSection {
  color: white;
}

h2 {
  color: yellow;
}
```

# CSS Syntax

Let's add some CSS properties!

- color (for text)
- background-color

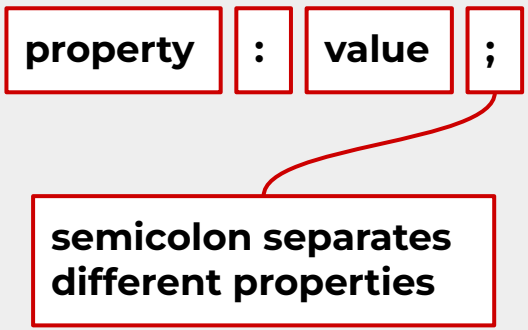
| property | : | value | ; |
|---|---|---|---|

```
#blog {
  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




```
#blog {
  background-color: seagreen;
}

.colorSection {
  color: white;
  background-color: black;
}

h2 {
  color: yellow;
}
```

# CSS Syntax

Review:

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

```
#blog {
  background-color: seagreen;
}

.colorSection {
  color: white;
  background-color: black;
}

h2 {
  color: yellow;
}
```

# CSS Syntax

Review:

| property | : | **value** | ; |
|---|---|---|---|

```
#blog {
  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 lang="en-US">
  <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:

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

```html
<html lang="en-US">
  <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:

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

```html
<html lang="en-US">
  <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:

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

```html
<html lang="en-US">
  <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:

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

```html
<html lang="en-US">
  <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

# 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 lang="en-US">
  <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 lang="en-US">
  <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?

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

```html
<html lang="en-US">
  <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?

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

```html
<html lang="en-US">
  <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?

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

```html
<html lang="en-US">
  <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?

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

```html
<html lang="en-US">
  <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>
```

# Homework

https://journ220.github.io

**Please help clean up:** close windows, return tables, etc.