

J220

Coding for Journalists

LECTURER
Soo Oh

PROMPTS

Preliminary lecture slides in
Slack

Zoom screenshare +
start Zoom recording

Agenda

Homework review + how much time

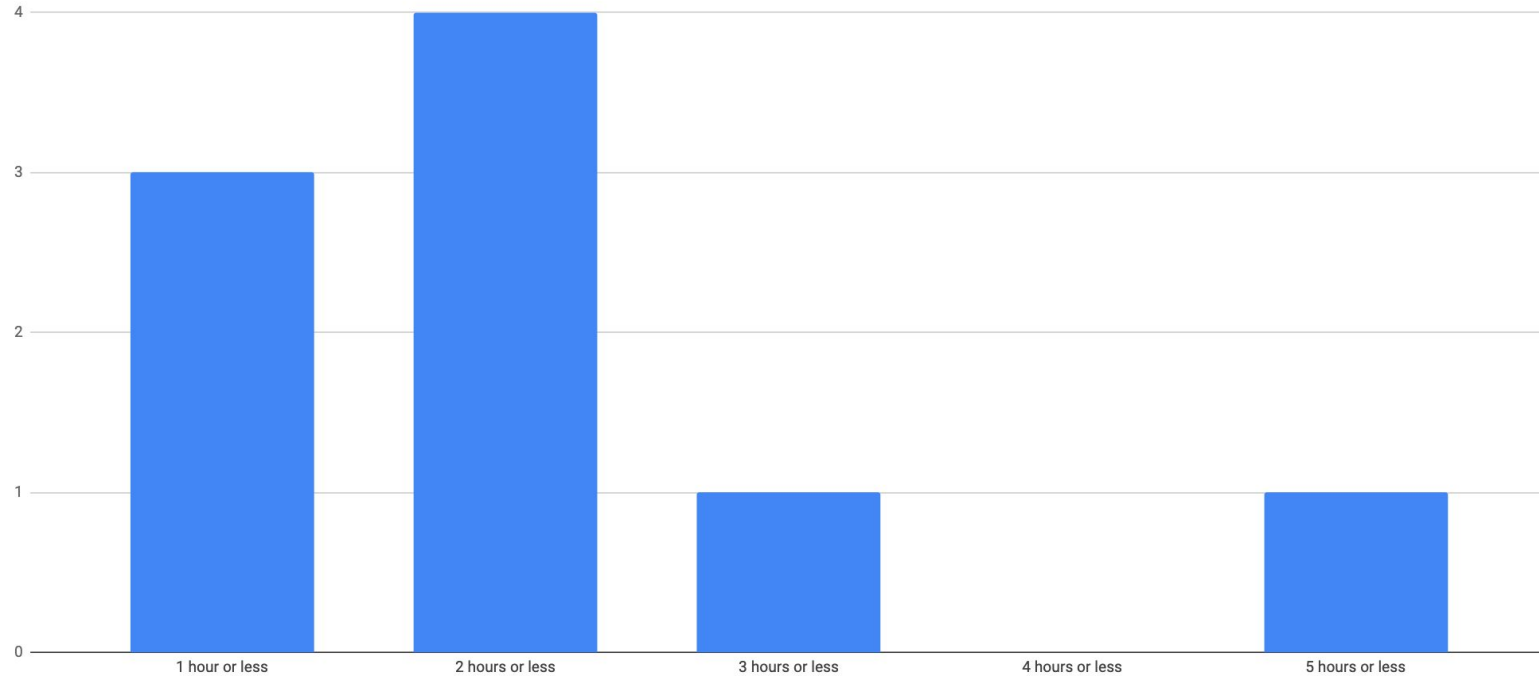
CSS

BREAK

Homework

How much time spent on J220

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



Review screen/projector
color/contrast for accessibility

Firefox accessibility

Screenshot of Firefox displaying the Washington Post article "Who is running for president in 2024 and who has dropped out" with the Accessibility pane open. The pane shows 15 contrast issues, including "Incumbent", "Four criminal indictments.", "Sources: Smart Politics and Post reporting", and others.

Who is running for president in 2024 and who has dropped out

Republicans
Haley Trump

Dropped out candidates
Burgum Christie DeSantis Elder Hurd Hutchinson Pence Ramaswamy Scott Suarez

Democrats
Biden Phillips
Incumbent

Dropped out candidates
Williamson

Others
Kennedy Stein

Accessibility pane:

- Role: **text leaf:** "Incumbent" @contrast
- Role: **text leaf:** "Four criminal indictments."
- Role: **text leaf:** "Sources: Smart Politics and Post reporting" @contrast
- Role: **text leaf:** "ended his bid for the Re... presidential nomination" @contrast
- Role: **text leaf:** "ended his bid for the Re... presidential nomination" @contrast
- Role: **text leaf:** "ended his bid for the Re... presidential nomination" @contrast
- Role: **text leaf:** "seized the spotlight" @contrast
- Role: **text leaf:** "raising the voting age"
- Role: **text leaf:** "suspended his campaign"
- Role: **text leaf:** "ended his long-shot 2024 presidential campaign" @contrast
- Role: **text leaf:** "is waging a long-shot bi_e Democratic nomination" @contrast
- Role: **text leaf:** "Amy B Wang" @contrast
- Role: **text leaf:** "Twitter"
- Role: **text leaf:** "Szu Yu Chen" @contrast

**Incognito
15 contrast
issues**

Screenshot of Firefox displaying the Washington Post article "Who is running for president in 2024 and who has dropped out" with the Accessibility pane open. The pane shows 10 contrast issues, including "Incumbent", "Hannah Knowles", "Amy B Wang", "Kati Perry", "Szu Yu Chen", "Sources: Smart Politics and Post reporting", "ended his bid for the Re... presidential nomination", "unsuccessfully sought to... Gov. Gavin Newsom (D)", and "Twitter".

Who is running for president in 2024 and who has dropped out

Republicans
Haley Trump

Dropped out candidates
Burgum Christie DeSantis Elder Hurd Hutchinson Pence Ramaswamy Scott Suarez

Democrats
Biden Phillips
Incumbent

Dropped out candidates
Williamson

Others
Kennedy Stein

Accessibility pane:

- Role: **text leaf:** "Incumbent" @contrast
- Role: **text leaf:** "Hannah Knowles" @contrast
- Role: **text leaf:** "Amy B Wang" @contrast
- Role: **text leaf:** "Kati Perry" @contrast
- Role: **text leaf:** "Szu Yu Chen" @contrast
- Role: **text leaf:** "Sources: Smart Politics and Post reporting" @contrast
- Role: **text leaf:** "ended his bid for the Re... presidential nomination" @contrast
- Role: **text leaf:** "unsuccessfully sought to... Gov. Gavin Newsom (D)" @contrast
- Role: **text leaf:** "Twitter" @contrast
- Role: **text leaf:** "Twitter" @contrast

**Logged in
10 contrast
issues**

Firefox accessibility

Other factors:

- Sidebars
- Ads
- Screen being used?

There are some things on your page you cannot control, but the tool helps you get a good overview.

CSS selectors

CSS Review

HTML

```
<h1>Headline</h1>
```

```
<p>Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Integer non nisl tellus.  
Praesent laoreet ut nisi vel  
volutpat. In nisi nulla,  
malesuada a diam non, euismod  
aliquam nibh.</p>
```

WEBSITE

Headline

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer non nisl tellus. Praesent
laoreet ut nisi vel volutpat. In
nisi nulla, malesuada a diam
non, euismod aliquam nibh.

CSS

```
h1 {  
  /* empty */  
}  
  
p {  
  /* empty */  
}
```

CSS Review

HTML

```
<h1>Headline</h1>
```

```
<p>Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Integer non nisl tellus.  
Praesent laoreet ut nisi vel  
volutpat. In nisi nulla,  
malesuada a diam non, euismod  
aliquam nibh.</p>
```

WEBSITE

Headline

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer non nisl tellus. Praesent
laoreet ut nisi vel volutpat. In
nisi nulla, malesuada a diam
non, euismod aliquam nibh.

CSS

```
h1 {  
  /* empty */  
}  
  
p {  
  color: steelblue;  
}
```

CSS Review

HTML

```
<h1>Headline</h1>
```

```
<p>Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Integer non nisl tellus.  
Praesent laoreet ut nisi vel  
volutpat. In nisi nulla,  
malesuada a diam non, euismod  
aliquam nibh.</p>
```

WEBSITE

Headline

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer non nisl tellus. Praesent
laoreet ut nisi vel volutpat. In
nisi nulla, malesuada a diam
non, euismod aliquam nibh.

CSS

```
h1 {  
  color: seagreen;  
}  
  
p {  
  color: steelblue;  
}
```

CSS Review

HTML

```
<h1>Headline</h1>
```

```
<p>Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.  
Integer non nisl tellus.  
Praesent laoreet ut nisi vel  
volutpat. In nisi nulla,  
malesuada a diam non, euismod  
aliquam nibh.</p>
```

WEBSITE

Headline

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Integer non nisl tellus. Praesent
laoreet ut nisi vel volutpat. In
nisi nulla, malesuada a diam
non, euismod aliquam nibh.

CSS

```
h1 {  
  color: seagreen;  
}  
  
p {  
  color: steelblue;  
}
```

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

We just reviewed type (or HTML tag) selectors.

class attributes are shared. You make up the **class** name; it doesn't matter for semantics or accessibility. We use **class** to define special styles.

id attributes must be unique to the element. You can't have more than one element with the same **id**.

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

Review: **class** and **id** selectors

```
<section class="intro">
  The following chapters are ...
</section>
<section class="chapter" id="chapter-1">
  ...
</section>
<section class="chapter" id="chapter-2">
  ...
</section>
<section class="chapter" id="chapter-3">
  ...
</section>
```

```
.chapter {
  color: red;
}
#chapter-1 {
  color: blue;
}
```

CSS Review

CSS syntax

```
.chapter {  
  color: red;  
}
```

Use a period before **class**

```
#chapter-1 {  
  color: blue;  
}
```

Use a hashtag before **id**

CSS Review

CSS syntax

```
.chapter {  
  color: red;  
}  
  
#chapter-1 {  
  color: blue;  
}
```

curly brackets
contain CSS
properties

CSS Review

CSS syntax

```
.chapter {  
  color: red;  
}  
  
#chapter-1 {  
  color: blue;  
}
```

indent for readability (2 spaces is typical)

CSS Review

CSS syntax

```
.chapter {  
  color: red;  
}  
  
#chapter-1 {  
  color: blue;  
}
```

property

:

value

;

CSS Review

CSS syntax

```
.chapter {  
  color: red;  
}  
  
#chapter-1 {  
  color: blue;  
  width: 60%;  
}
```

property : value ;

semicolon separates
different properties

CSS Review

CSS syntax

```
.properties {  
  color: blue;           /* or hex colors like #888888 */  
  width: 60%;           /* or pixel values like 100px */  
  font-family: serif; /* sans-serif, or Arial */  
  font-size: 12px;  
}
```

**how to write
comments in CSS**

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

We use **commas** to apply the same CSS to multiple elements. This is called a [selector list](#).

WEBSITE

Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer non nisl tellus. Praesent laoreet ut nisi vel volutpat. In nisi nulla, malesuada a diam non, euismod aliquam nibh.

CSS

```
h1, p {  
  color: seagreen;  
}
```

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

We use **commas** to apply the same CSS to multiple elements. This is called a [selector list](#).

WEBSITE

Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer non nisl tellus. Praesent laoreet ut nisi vel volutpat. In nisi nulla, malesuada a diam non, euismod aliquam nibh.

CSS

```
h1, p {  
  color: seagreen;  
}
```

```
/* separate lines are OK, too */  
  
h1,  
p {  
  color: seagreen;  
}
```

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

It's common to combine the **html** and **body** tags in CSS.

```
html, body {  
  /* empty */  
}
```

Why?

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

“It is not recommended to apply styles to the `<html>` element because they will be overridden by the `<body>` element styles and any other element in the document.

The only exception could be if you want to **declare the font styles** that will be inherited by all its descendant elements, **especially the font size**. This is because the `<html>` element selector as the root element, has the rem (root em unit) sizing of any element based on whatever font size set for the element (root element).”

We're going to learn about this today!

[Source](#)

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

“There is a weird thing in CSS where the **background-color** on `<body>` floods the whole viewport even if the metrics of the element itself don't cover that whole area. Unless the **background-color** gets set on the `<html>` element, then it doesn't.

If flooding is the goal, it can be smart to just set it on the `<html>` element to begin with.”

We haven't learned this yet, but this means, if you apply a width to `<body>`, which you really shouldn't do, then the color will expand past the width of the body. This might not make sense to you yet, but might later.

[Source](#)

CSS Selectors

type (tag), class, and
id selectors

html, body (and
using commas)

pseudo-selectors (for
links)

```
<html>
```

```
<body>
```

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

```
<html>
```

not visible to users

```
<body>
```

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

forms basis for rem sizing on page

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

<html>

not visible to users

<body>

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

forms basis for rem sizing on page

```
body {  
  background-color: thistle;  
  font-size: 16px;  
  /* OR */  
  font-size: 1rem;  
}
```

default font-size for page, visible to readers

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

<html>

<body>

combined

```
html, body {  
  background-color: thistle;  
  font-size: 16px;  
  /* good idea to add  
  font-family here too */  
}
```

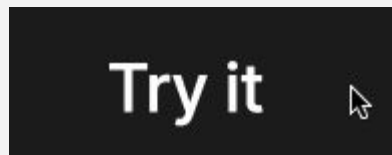
CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

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



CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

The link color can change, when the mouse hovers over the anchor. Below, a **blue** link turns **darkblue** on hover.

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

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

Pay attention to the syntax. There are no spaces between the *selector* (**a**) and the *pseudo-selector* (**:hover**).

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

: (colon, no spaces)
between **a** and
hover

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

Sometimes, the style of the link also changes to indicate the user has already clicked on or visited the page.

```
a {  
  color: blue;  
}  
  
a:hover {  
  color: darkblue;  
}  
  
a:visited {  
  color: gray;  
}
```

CSS Selectors

type (tag), class, and id selectors

html, body (and using commas)

pseudo-selectors (for links)

:hover, **:visited**, and **:focus** (which get triggered by a keyboard tab) are all pseudo-selectors.

While **:visited** can only be applied to a link (an `<a>` tag), **:hover** and **:focus** can be used with any HTML element.

CSS properties

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

We've talked about two CSS properties: **color** and **background-color**.

The property **color** applies to **text**.

WEBSITE

Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer non nisl tellus. Praesent laoreet ut nisi vel volutpat. In nisi nulla, malesuada a diam non, euismod aliquam nibh.

CSS

```
h1, p {  
  color: seagreen;  
}
```

CSS Properties

colors

font-family

font-size





font-weight

font-style

line-height

So far, we've used ["named" colors](#). There are ~160 named colors in CSS.

Hex values are 6-digit codes that represent an RGB color. The **RGB** color model is best used for digital screens, though **HSL** is also popular.

name	hex	chip
olive	#808000	
teal	#008080	
lime	#00FF00	
maroon	#800000	

You can find hex colors using [Adobe Color](#), [Google](#), or browser tools.

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

It doesn't matter whether you use named colors or hex colors.

WEBSITE

Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Integer non nisl tellus. Praesent laoreet ut nisi vel volutpat. In nisi nulla, malesuada a diam non, euismod aliquam nibh.

CSS

```
h1 {  
  color: olive; /* #808000 */  
}  
p {  
  color: #808000; /* olive */  
}
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

You can also use **rgb** or **rgba** colors. The fourth property in **rgba** allows you adjust opacity (or *alpha*).

WEBSITE

Headline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer non nisl tellus. Praesent laoreet ut nisi vel volutpat. In nisi nulla, malesuada a diam non, euismod aliquam nibh.

CSS

```
h1 {  
  color: rgb(128, 128, 0);  
}  
p {  
  color: rgba(128, 128, 0, 0.5);  
}
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

The property **background-color** applies to backgrounds. (I know this sounds obvious, but it can be confusing once you learn about CSS for font and text).

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

There are some known [web-safe fonts](#) that are installed on most computers.

- **Arial**
- Courier New
- **Georgia**
- Times New Roman
- **Verdana**

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

How are these fonts grouped up?

Arial

Georgia

Courier New

Verdana

Times New Roman

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

How are these fonts grouped up?

sans-serif

serif

monospace

Arial

Georgia

Courier New

Verdana

Times New Roman

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

Serif fonts: Finishing strokes. Some claim serif fonts are easier to read in the body of articles. Think about the tone you're setting.



Fancy

Georgia

```
html, body { font-family: Georgia, serif; }
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

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

Not Fancy



Arial

```
html, body { font-family: Arial, sans-serif; }
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

Monospace fonts: Every character is the same width.

Good for
code

Courier New

```
html, body { font-family: Courier New, monospace; }
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

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

You should use 2 to 3 values, in your preferred order.

```
p {  
  font-family: Helvetica, Arial, sans-serif;  
}
```

(It's encouraged to use quotation marks around [non-standard fonts](#).)

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

What about loading custom fonts? We'll learn that at a future lecture!

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

Setting the sizes for fonts (and other dimensions later) can be done in different ways:

- pixels
- rem
- em
- percent

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

Pixels: A static way of setting a size.

Good if you need to be very accurate, down to the pixel. Con: no resizing when page is zoomed in.

```
p {  
  font-family: Helvetica, Arial, sans-serif;  
  font-size: 16px;  
}
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

rem: The browser adjusts size based on a **font-size** set on the `<html>` element. This helps when zooming into page, mobile friendly.

```
html, body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 18px;
}
p {
  font-size: 1rem;
}
```

What size will `<p>` be on the page?

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

rem: The browser adjusts size based on a **font-size** set on the `<html>` element. This helps when zooming into page, mobile friendly.

```
html, body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 18px;
}
p {
  font-size: 1rem;
}
```

What size will `<p>` be on the page? **18px**

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

rem

```
html, body {  
  font-family: Helvetica, Arial, sans-serif;  
  font-size: 18px;  
}  
p {  
  font-size: 2rem;  
}
```

What size will `<p>` be on the page?

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

rem

```
html, body {  
  font-family: Helvetica, Arial, sans-serif;  
  font-size: 18px;  
}  
p {  
  font-size: 2rem;  
}
```

What size will `<p>` be on the page? **36px**

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

em: The size adjusts based on the **font-size** of the parent element. If no base **font-size** is set on the parent, then the browser sets default. In most browsers, `<p>` is **16px**.

```
html, body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 18px;
}
p {
  font-size: 1em;
}
```

What size will `<p>` be on the page?

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

em: The size adjusts based on the **font-size** of the parent element. If no base **font-size** is set on the parent, then the browser sets default. In most browsers, `<p>` is **16px**.

```
html, body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 18px;
}
p {
  font-size: 1em;
}
```

What size will `<p>` be on the page? **18px**

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

rem vs. em

HTML

```
<article>

  <p class="rem">Integer non
  nisl tellus. Praesent laoreet
  ut nisi vel volutpat.</p>

  <p class="em">In nisi nulla,
  malesuada a diam non, euismod
  aliquam nibh.</p>

</article>
```

CSS

```
html, body {
  font-size: 18px;
}
article {
  font-size: 20px;
}
.rem {
  font-size: 1rem;
}
.em {
  font-size: 1em;
}
```

What size will the `<p>`s be on the page?

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

rem vs. em

HTML

```
<article>  
  
  <p class="rem">Integer non  
  nisl tellus. Praesent laoreet  
  ut nisi vel volutpat.</p>  
  
  <p class="em">In nisi nulla,  
  malesuada a diam non, euismod  
  aliquam nibh.</p>  
  
</article>
```

18px

20px

CSS

```
html, body {  
  font-size: 18px;  
}  
article {  
  font-size: 20px;  
}  
.rem {  
  font-size: 1rem;  
}  
.em {  
  font-size: 1em;  
}
```

What size will the `<p>`s be on the page?

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

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.

```
html, body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 18px;
}

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

What size will `<p>` be on the page?

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

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.

```
html, body {
  font-family: Helvetica, Arial, sans-serif;
  font-size: 18px;
}

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

What size will `<p>` be on the page? **18px**

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

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

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

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

weight	description
100	thin
200	
300	light
400	normal
500	
600	
700	bold
800	
900	black, heavy

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

Why use **font-weight** to make something bold instead of `` or ``?

Those tags have semantic meaning. They point to something that is important or serious. They're not used for aesthetic reasons.

```
p.dek {  
  font-weight: bold;  
}
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

The property **font-style** is used to italicize text for aesthetic reasons. We don't use `` or `<i>`, for the same reason we don't use `` or `` just to style bold text.

Avoid big sections of italic text, which can be hard to read.

```
p.dek {  
  font-style: italic;  
}
```


CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

line-height: The space between lines of text.

```
p {  
  line-height: 1.2rem;  
}
```

Can use px, rem, em, percent, or a number (unitless).

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

The preferred implementation for both readability and making the page mobile-friendly is using a **unitless number**, which is multiplied by the **font-size**.

```
p {  
  line-height: 1.2;  
}
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

combining

So how do we combine this all?

```
html, body {  
  font: 18px/1.2 Garamond, 'Times New Roman', serif;  
}  
  
h1, h2, h3, h4, h5, h6 {  
  font-family: Helvetica, Arial, sans-serif;  
  font-weight: bold;  
}  
  
p {  
  font-size: 1rem;  
  line-height: 1.2;  
}  
  
p.dek {  
  font-style: italic;  
}
```

CSS Properties

colors

font-family

font-size

font-weight

font-style

line-height

combining

I combined values to **font** here:

```
html, body {  
  font: 18px/1.2 Garamond, 'Times New Roman', serif;  
}
```

Combine font values in a specific order:

```
font: font-style font-variant font-weight  
      font-size/line-height font-family
```

You can skip **font-style**, **font-variant**, and **line-height**.

What's **font-variant**? You can set it to **normal** or **small-caps**.

Break

Meet back in 15 minutes.

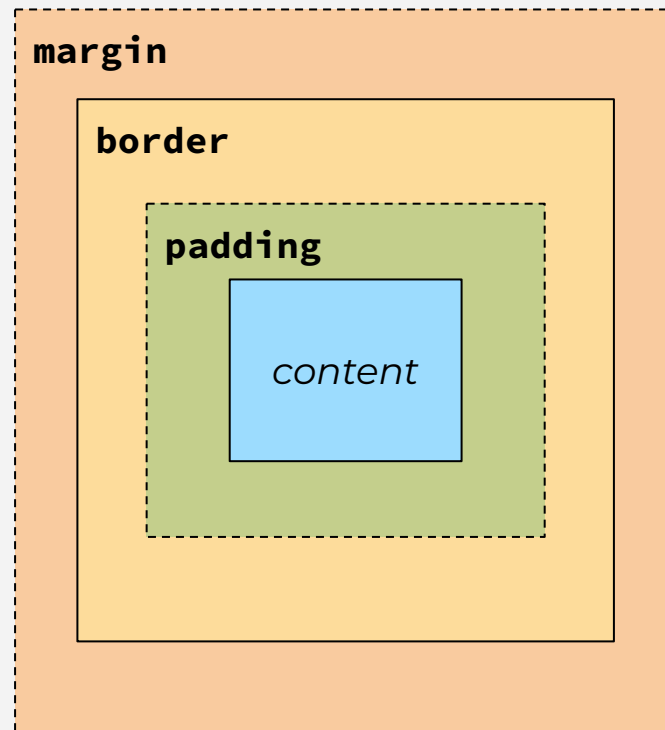
In-class activity

Box Model

Box Model

Let's talk about spacing!

This is the box model.

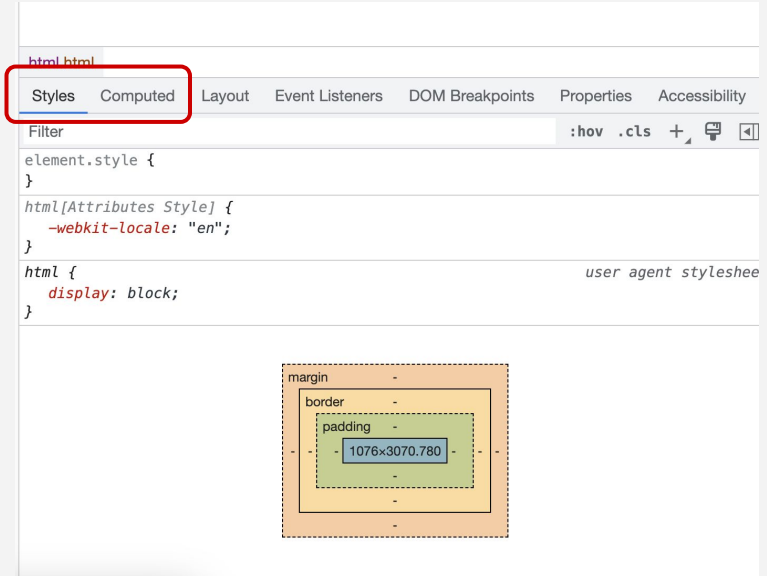


Opening browser development tools

You might have seen this in your browser.

In **Brave** or **Chrome**, right-click on a page and select “**Inspect**.”

Then click on the “**Styles**” or “**Computed**” tabs.

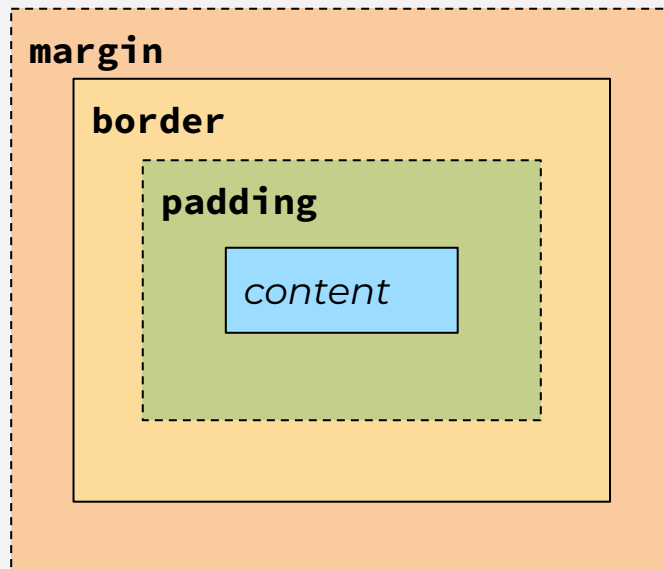


What makes up the box model

The CSS properties that makes up the box are:

- **margin**
- **border**
- **padding**

The *content* is what is found inside of the HTML element.



Example

Let's take a look at this `<aside>` element.

WEBSITE

onec ultrices tempus mi vitae bibem.

What do you think? [Email us](#) to let us know!

Nullam finibus commodo vulputate. Quisque quis tempus lacus. Quisque mi magna, accumsan sed pulvinar vitae, varius ac risus. Fusce elit lectus.

HTML

```
<aside>  
  
  <p>What do you think?  
  <a href="mailto:email@berkeley.edu">Email us</a> to let us know!</p>  
  
</aside>
```

Example

The stuff that's inside the `<aside>` tag is the **content**. That includes the `<p>` tag (which has its own box model).

WEBSITE

onec ultrices tempus mi vitae bibem.

What do you think? [Email us](#) to let us know!

Nullam finibus commodo vulputate. Quisque quis tempus lacus. Quisque mi magna, accumsan sed pulvinar vitae, varius ac risus. Fusce elit lectus.

HTML

```
<aside>
  <p>What do you think?
  <a href="mailto:email
  @berkeley.edu">Email
  us</a> to let us
  know!</p>
</aside>
```

Example

This `<aside>` tag has a thin black **border** (although here I've highlighted it orange).

WEBSITE

onec ultrices tempus mi vitae bibem.

What do you think? [Email us](#) to let us know!

Nullam finibus commodo vulputate. Quisque quis tempus lacus. Quisque mi magna, accumsan sed pulvinar vitae, varius ac risus. Fusce elit lectus.

CSS

```
aside {  
  border-width: 1px;  
  border-style: solid;  
  border-color: #000000;  
}
```

new CSS properties!

Example

The space inside of the **border** but between the *content* is the **padding**.

WEBSITE

onec ultrices tempus mi vitae bibem.

What do you think? [Email us](#) to let us know!

Nullam finibus commodo vulputate. Quisque quis tempus lacus. Quisque mi magna, accumsan sed pulvinar vitae, varius ac risus. Fusce elit lectus.

CSS

```
aside {  
  border-width: 1px;  
  border-style: solid;  
  border-color: #000000;  
  
  padding-top: 1rem;  
  padding-right: 0.66rem;  
  padding-bottom: 1rem;  
  padding-left: 0.66rem;  
}
```

Example

The space outside of the border between the `<aside>` tag and the above and below paragraphs is the **margin**.

WEBSITE

onec ultrices tempus mi vitae bibem.

What do you think? [Email us](#) to let us know!

Nullam finibus commodo vulputate.

Quisque quis tempus lacus. Quisque mi magna, accumsan sed pulvinar vitae, varius ac risus. Fusce elit lectus.

CSS

```
aside {  
  border-width: 1px;  
  border-style: solid;  
  border-color: #000000;  
  
  padding-top: 1rem;  
  padding-right: 0.66rem;  
  padding-bottom: 1rem;  
  padding-left: 0.66rem;  
  
  margin-top: 1rem;  
  margin-right: 0;  
  margin-bottom: 1rem;  
  margin-left: 0;  
}
```

Example

margin and **padding**

can seem very similar. The difference is more obvious when you add a **border** (even temporarily). They add spacing in different areas.

WEBSITE

onec ultrices tempus mi vitae bibem.

What do you think? [Email us](#) to let us know!

Nullam finibus commodo vulputate.

Quisque quis tempus lacus. Quisque mi magna, accumsan sed pulvinar vitae, varius ac risus. Fusce elit lectus.

CSS

```
aside {  
  border-width: 1px;  
  border-style: solid;  
  border-color: #000000;  
  
  padding-top: 1rem;  
  padding-right: 0.66rem;  
  padding-bottom: 1rem;  
  padding-left: 0.66rem;  
  
  margin-top: 1rem;  
  margin-right: 0;  
  margin-bottom: 1rem;  
  margin-left: 0;  
}
```


Combining property values

Let's take a closer look at the new CSS properties we just introduced.

```
aside {  
  border-width: 1px;  
  border-style: solid;  
  border-color: #000000;  
  
  padding-top: 1rem;  
  padding-right: 0.66rem;  
  padding-bottom: 1rem;  
  padding-left: 0.66rem;  
  
  margin-top: 1rem;  
  margin-right: 0;  
  margin-bottom: 1rem;  
  margin-left: 0;  
}
```

Combining property values

Just like with **font**, we can declare property values in one line.

Also, you don't have to declare a unit (like **px**, **rem**, or **em**) with the number **0**.

```
aside {  
  /* border will never work without at LEAST solid */  
  border: 1px solid #000000;  
  
  /* 3 different ways to declare padding */  
  /* 1) top right bottom left */  
  padding: 1rem 0.66rem 1rem 0.66rem;  
  
  /* 2) (top/bottom) (left/right) */  
  padding: 1rem 0.66rem;  
  
  /* 3) top (left/right) bottom */  
  padding: 1rem 0.66rem 1rem;  
  
  /* same goes for margin */  
  margin: 1rem 0 1rem 0;  
  margin: 1rem 0;  
  margin: 1rem 0 1rem;  
}
```

Default browser styling

When you use semantic HTML, you might find many elements already have a lot of **margin** and **padding** that you don't want.

Every browser has its own base stylesheet, but many of them are pretty similar.

Default browser styling

When you use semantic HTML, you might find many elements already have a lot of **margin** and **padding** that you don't want.

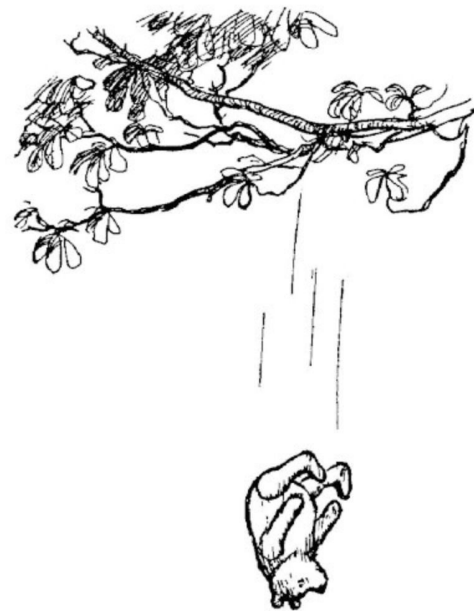
Every browser has its own base stylesheet, but many of them are pretty similar.

(Just FYI, default browser styling is different from **initial values**, which we'll talk about in a future class.)

Problems with default browser styling

Let's look at the [lecture examples](#) in our Feb. 5 lecture on HTML elements. You learned about the `<figure>` tag during lecture.

Figure



Winnie-the-Pooh falls out of a tree. (Ernest H. Shepard)

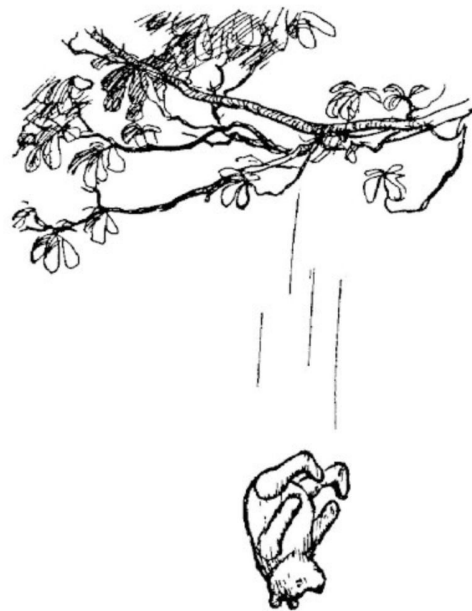
Problems with default browser styling

Even though the `<h2>` “**Figure**” is near the edge of the page, the `<figcaption>` is not.

edge

I don't know if this space is **padding** or **margin**, so I'm shading it **pink** for now

Figure



Winnie-the-Pooh falls out of a tree. (Ernest H. Shepard)

Problems with default browser styling

HTML

```
<section id="figures">
  <h2>Figure</h2>
  <figure>
    
    <figcaption>Winnie-the-Pooh
    falls out of a tree. (Ernest
    H. Shepard)</figcaption>
  </figure>
</section>
```

WEBSITE

Figure



Winnie-the-Pooh falls out of a tree. (Ernest H. Shepard)

When I inspect the elements in my browser, I notice that the `<h2>` has **margin**.

Problems with default browser styling

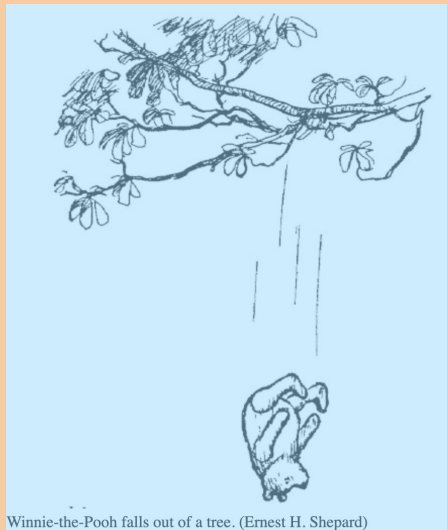
HTML

```
<section id="figures">
  <h2>Figure</h2>
  <figure>
    

    <figcaption>Winnie-the-Pooh
    falls out of a tree. (Ernest
    H. Shepard)</figcaption>
  </figure>
</section>
```

WEBSITE

Figure



Inspecting `<figure>` shows that it has some **margin** at the top and bottom, too, but *a*lso left and right.

Problems with default browser styling

HTML

```
<section id="figures">
  <h2>Figure</h2>

  <figure>
    

    <figcaption>Winnie-the-Pooh
    falls out of a tree. (Ernest
    H. Shepard)</figcaption>
  </figure>
</section>
```

WEBSITE

Figure



Winnie-the-Pooh falls out of a tree. (Ernest H. Shepard)

You may notice too, that even though both `<figure>` and `<h2>` have **margin** set for the same space, you don't see both of them, due to [margin collapse](#).

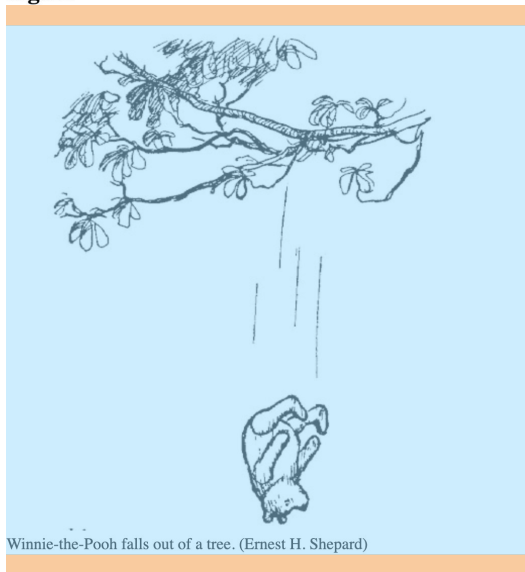
Problems with default browser styling

HTML

```
<section id="figures">
  <h2>Figure</h2>
  <figure>
    
    <figcaption>Winnie-the-Pooh
    falls out of a tree. (Ernest
    H. Shepard)</figcaption>
  </figure>
</section>
```

WEBSITE

Figure



CSS

```
/* If I want to make
figcaption flush with the
edge of the page, I have to
set the CSS on figure */
```

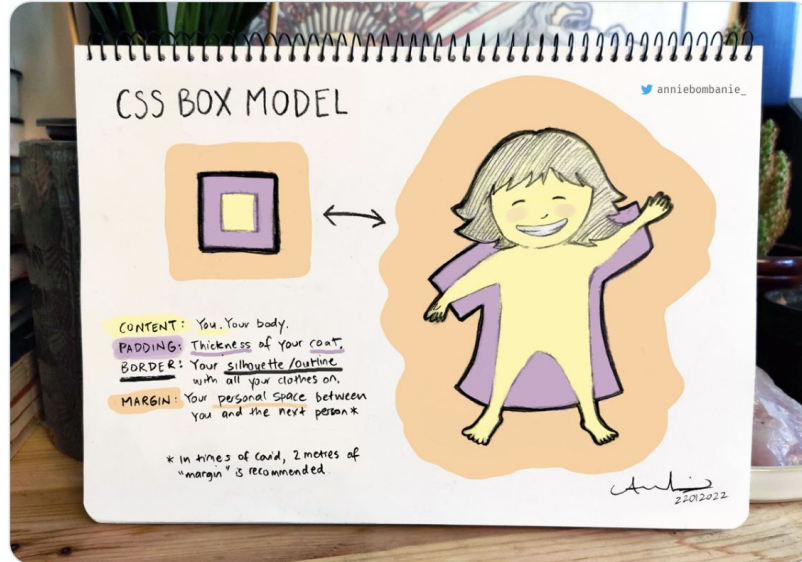
```
figure {
  margin-left: 0;
  margin-right: 0;
}
```



annie 🌈⚡
@anniebombanie_



CSS Box Model — an illustrated analogy.



5:10 AM · Jan 24, 2022



From [@anniebombanie](https://twitter.com/anniebombanie)

Homework

<https://journ220.github.io>

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