J220
Coding for Journalists

LECTURER
Soo Oh

PROMPTS

Download lecture files from
https://journ220.github.io

start Zoom recording + captions

Agenda

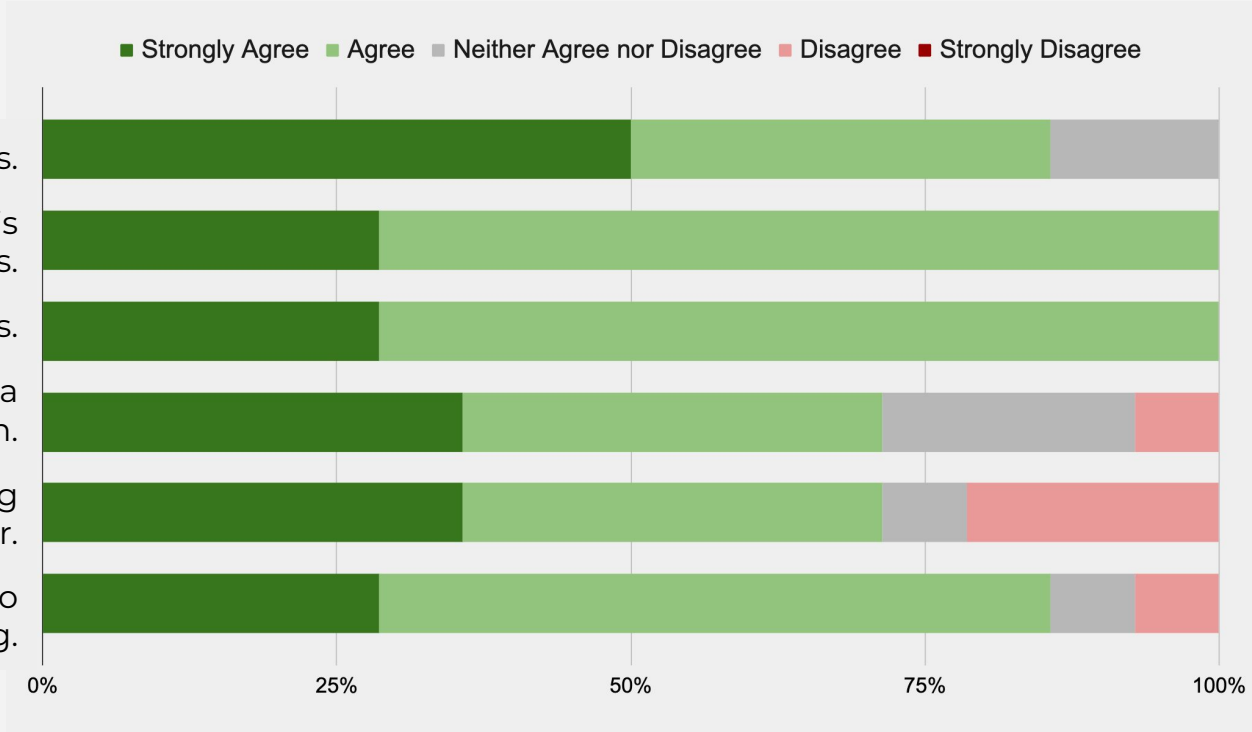
- **Announcements**
- **Midterm survey results**
- **Homework review:** Assignment 03-20 (+ “how much time”)
- **Lecture topic:** Responsive layouts
- **In-class activity I:** Mapping out a news site layout
- **Lecture topic:** HTML/CSS frameworks
- **This week’s homework + Previous final projects**
- **BREAK**
- **In-class activity II:** Wireframing

Announcements

- lecturers@journalism.berkeley.edu — email group for non-faculty lecturers (not sure if it'll bounce back if you're not on the list)
- Best to book office hours with the lecturer who corrected your homework. (You can usually tell by who left a comment — maybe bCourses doesn't show you? Slack Yoli and Soo if you want to make sure.)

Midterm survey results

Course assessment



What has been most helpful for your learning?

In-class activities (7)

Homework is practical/useful (5)

Reviewing slides/recording (5)

Homework review (2)

Real world examples

Code academy links

Accessibility focus

What has caused you the most difficulty in terms of learning?

Moving too fast (7)

Understanding concepts (6)

Too many tangents and hypotheticals (4)

HW including material not reviewed in class (2)

Class moving too slow (2)

Time constraints outside of class

What suggestion(s) can you make that would enhance your learning experience?

Speed of class (some concepts too fast, others too slow) (5)

More in-class activities (4)

More individual work (3)

More practical assignments

More visual aids

Career opportunities

What more are you hoping to learn?

Javascript (6)

How to build a webpage (5)

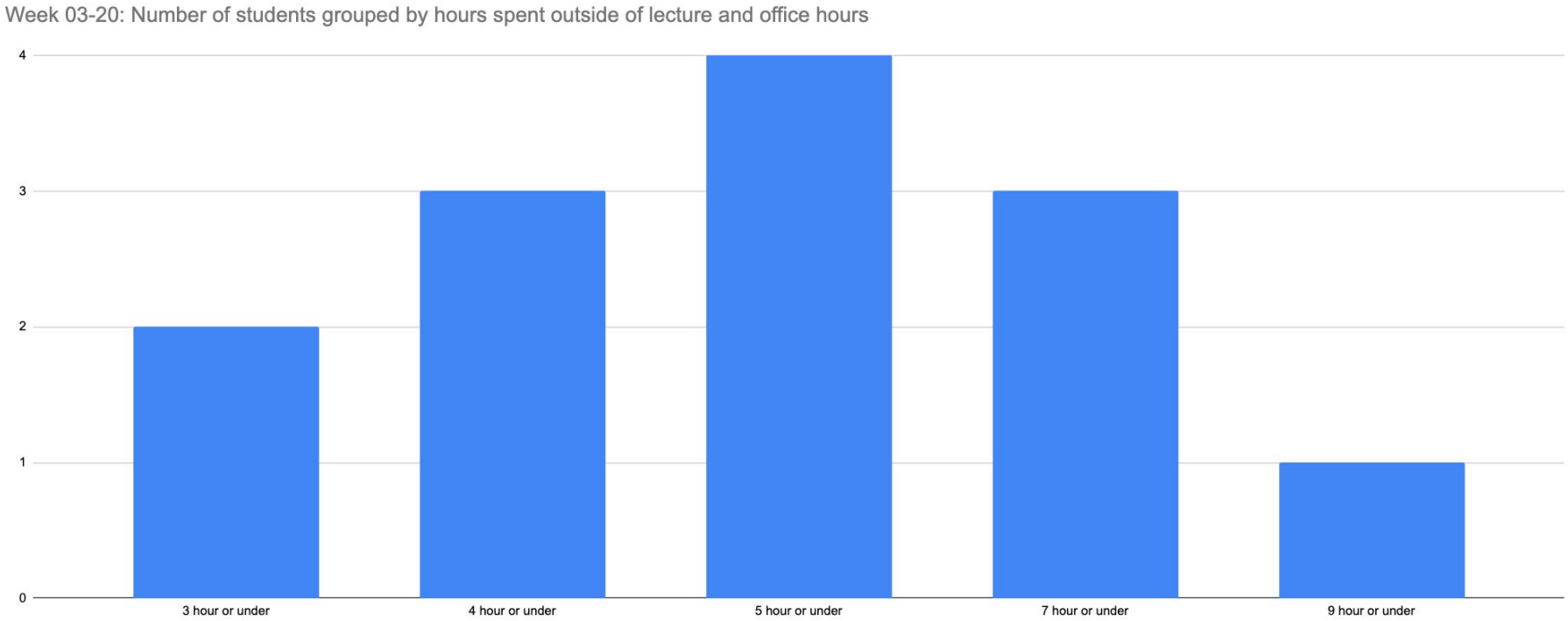
More CSS (3)

Build interactive pages (2)

Real world application of these skills

Homework review

How much time spent on J220 last week



Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files

```
<html>
```

```
<body>
```

Homework Review

`html, body {}`

`<html>`, `<head>`

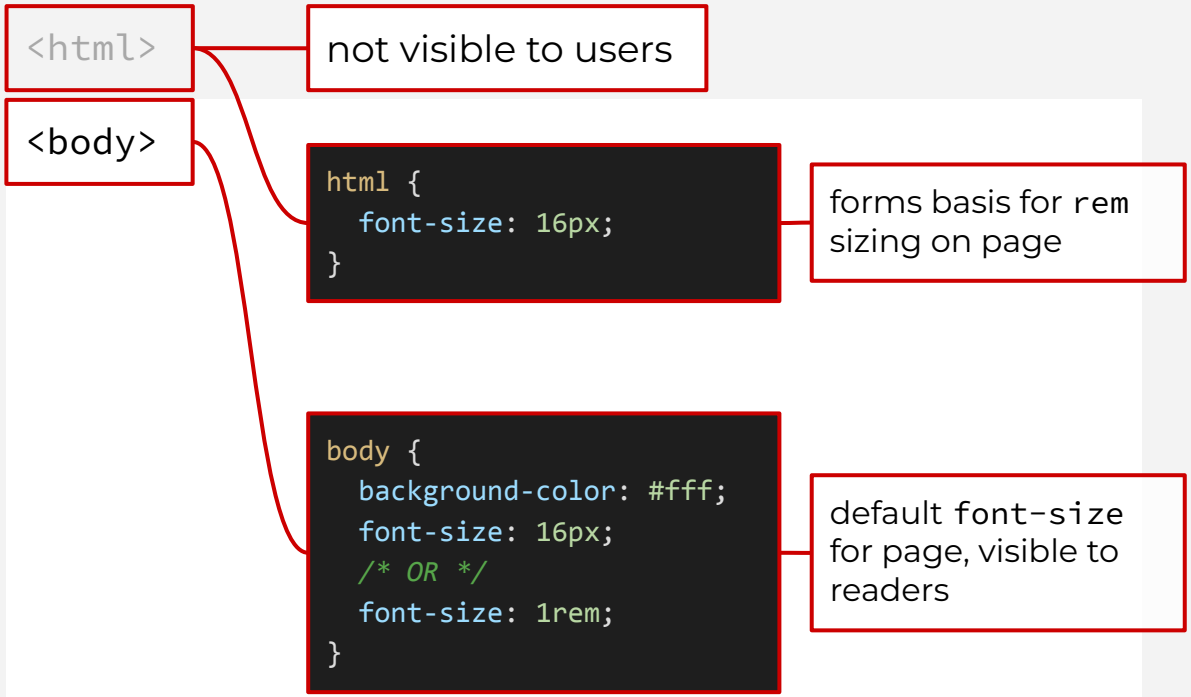
asset file names and organizing your files

Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files



Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files

Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files

"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)."

Source

Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files

“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.”

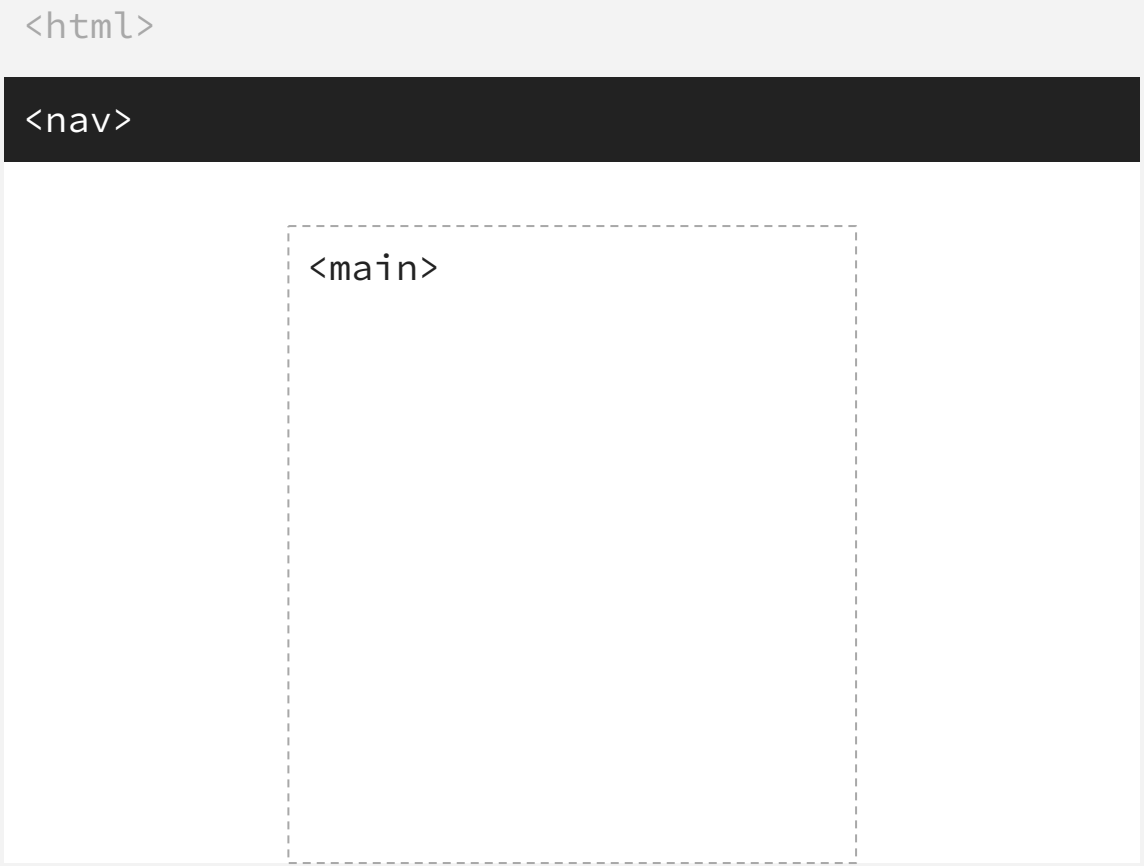
Source

Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files

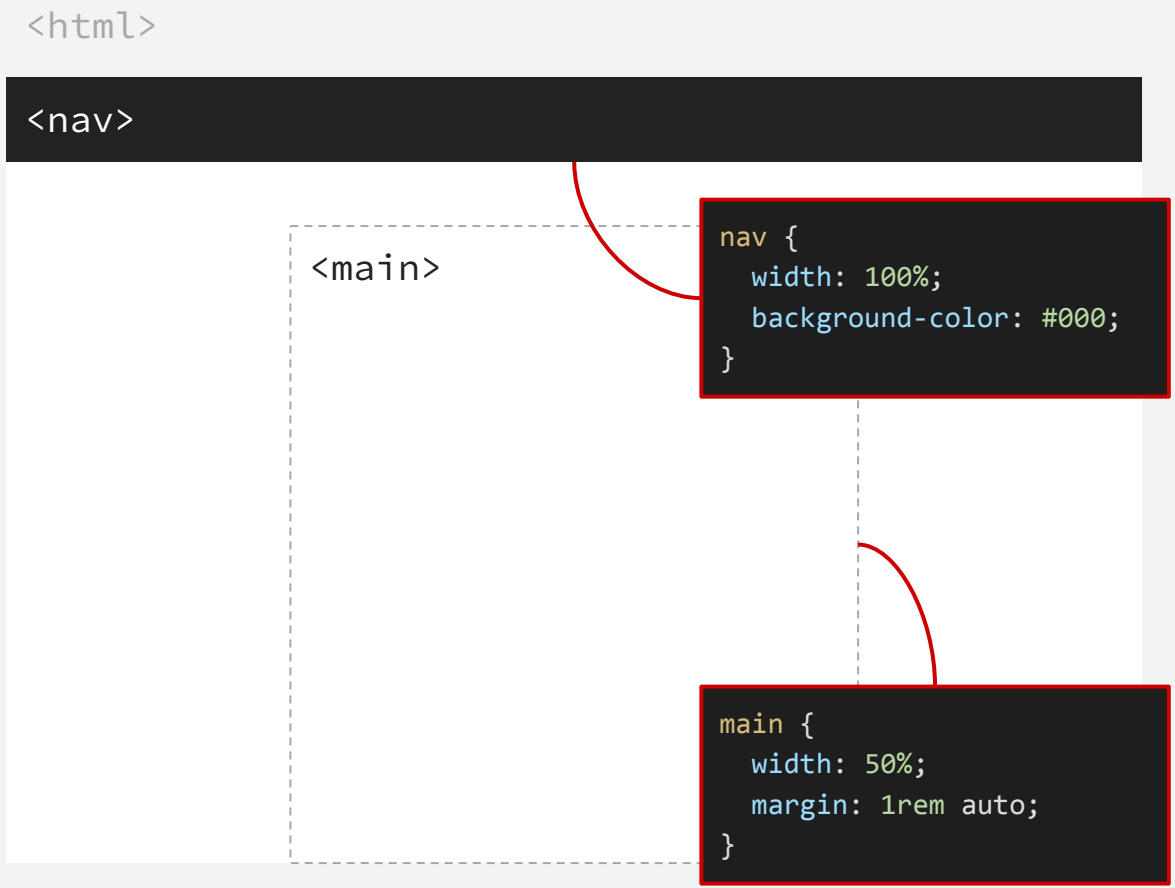


Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files

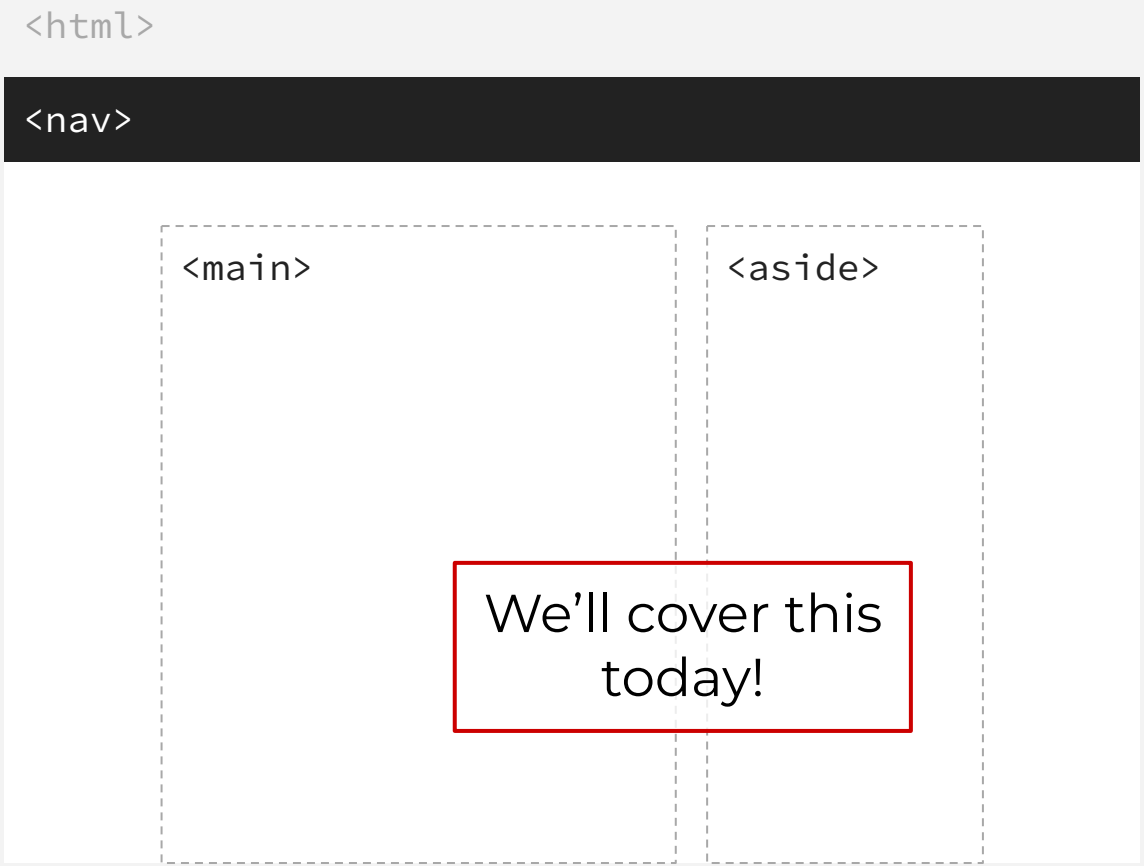


Homework Review

`html, body {}`

`<html>`, `<head>`

asset file names and organizing your files



Homework Review

`html, body {}`

`<html>, <head>`

asset file names and organizing your files

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="description" content="Farmers Market at Stockton">
    <title>A Market For All</title>

    <!-- Put your fonts and other CSS here -->
  </head>

...
```

Homework Review

html, body {}

<html>, <head>

asset file names and organizing your files

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="description" content="Farmers Market at Stockton">
    <title>A Market For All</title>

    <!-- Put your fonts and other CSS here -->
  </head>

...
```

Always needs to be first line in html file

Homework Review

`html, body {}`

`<html>, <head>`

asset file names and organizing your files

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="description" content="Farmers Market at Stockton">
    <title>A Market For All</title>

    <!-- Put your fonts and other CSS here -->
  </head>

...
```

always include natural language of page

Homework Review

html, body {}

<html>, <head>

asset file names and organizing your files

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="description" content="Farmers Market at Stockton">
    <title>A Market For All</title>

    <!-- Put your fonts and other CSS here -->
  </head>

...
```

<meta charset="utf-8"> should be the 1st line in **<head>** (even HTML comments should come after)

Homework Review

html, body {}

<html>, <head>

asset file names and organizing your files

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <meta name="description" content="Farmers Market at Stockton">
    <title>A Market For All</title>

    <!-- Put your fonts and other CSS here -->
  </head>

...
```

Import your Google fonts and add your CSS files *after* the meta information.

Homework Review

`html, body {}`

`<html>, <head>`

asset file names and organizing your files

- The main page in a folder is called **`index.html`** - browser will automatically look for it in any folder - this will not work when you are just viewing files from your computer
- For your file names, use **lowercase**, **remove spaces**, and **avoid most symbols** (except hyphens and underscores).

Responsive layouts

media queries

syntax

principles

more complex queries

What are media queries?

“Media queries allow you to apply CSS styles depending on a device's general type (such as print vs. screen) or other characteristics such as screen resolution or browser viewport width.”

Source

media queries

syntax

principles

more complex queries

```
h1 {
  font-size: 2em;
}

@media (min-width: 600px) {
  h1 {
    font-size: 3em;
  }
}
```

This is a very basic media query that changes the font-size on your site's headline based on your browser width.

media queries

syntax

principles

more complex queries

```
h1 {
  font-size: 2em;
}

@media (min-width: 600px) {
  h1 {
    font-size: 3em;
  }
}
```

This is a CSS **at-rule**. (It's just something to memorize.) Here, **`min-width`** defines the minimum width of the browser window. **Don't forget to use the parentheses!**

media queries

syntax

principles

more complex queries

```
h1 {
  font-size: 2em;
}

@media (min-width: 600px) {
  h1 {
    font-size: 3em;
  }
}
```

We define our **h1** font-size twice. First is our default value at **2em**. Then, nested within the media query, is another value at **3em**.

media queries

syntax

principles

more complex queries

```
h1 {
  font-size: 2em;
}

@media (min-width: 600px) {
  h1 {
    font-size: 3em;
  }
}
```

What this all means: The **<h1>** element will appear **2em**, until the screen reaches a minimum width of **600 pixels**. At 600+ pixels, the font-size becomes **3em**.

media queries

syntax

principles

more complex queries

```
h1 {
  font-size: 2em;
}

@media (min-width: 600px) {
  h1 {
    font-size: 3em;
  }
}
```

Just like **`min-width`**, there's also **`max-width`**. But **"mobile-first"** means that you're designing around **`min-width`**, not **`max-width`**.

media queries

syntax

principles

more complex queries

```
h1 {
  font-size: 2em;
}

@media (min-width: 600px) {
  h1 {
    font-size: 3em;
  }
}
```

Open `media_queries.html`

media queries

syntax

principles

more complex queries

```
.medium-only {
  display: none;
  color: orange;
}

/* For browser windows between 600 and 1199 pixels
wide */
@media (min-width: 400px) and (max-width: 1199px) {
  .medium-only {
    display: block;
  }
}
```

The **parentheses** and **and** keyword are important!

media queries

syntax

principles

more complex queries

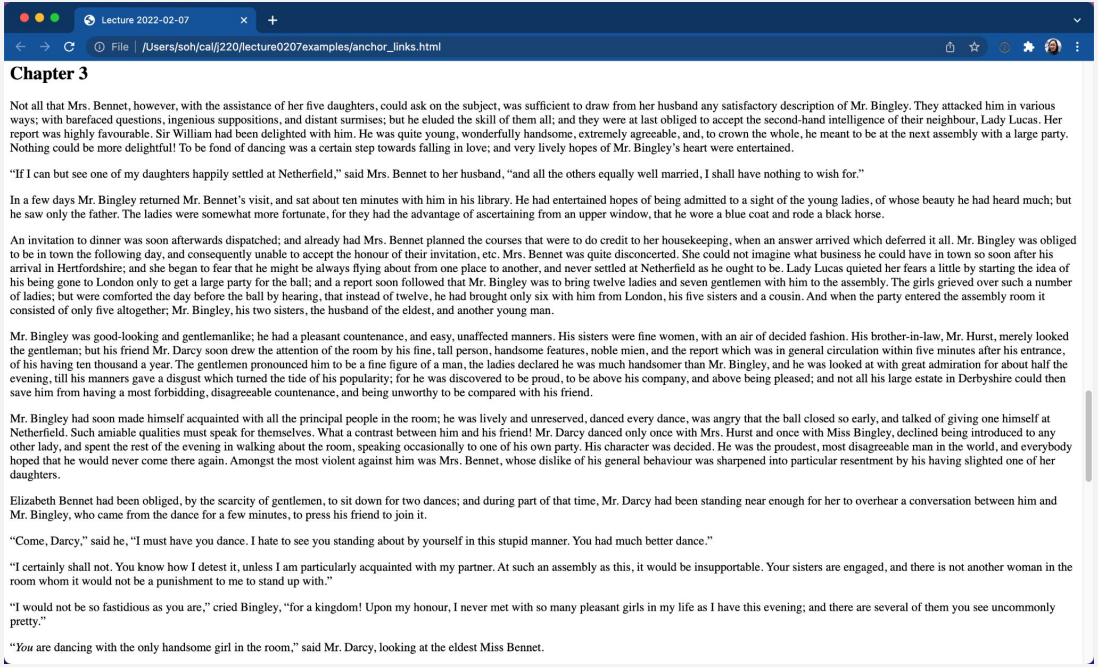
```
.medium-only {
  display: none;
  color: orange;
}

/* For browser windows between 600 and 1199 pixels
wide */
@media (min-width: 400px) and (max-width: 1199px) {
  .medium-only {
    display: block;
  }
}
```

Open `media_queries_complex.html`

What questions do you have?

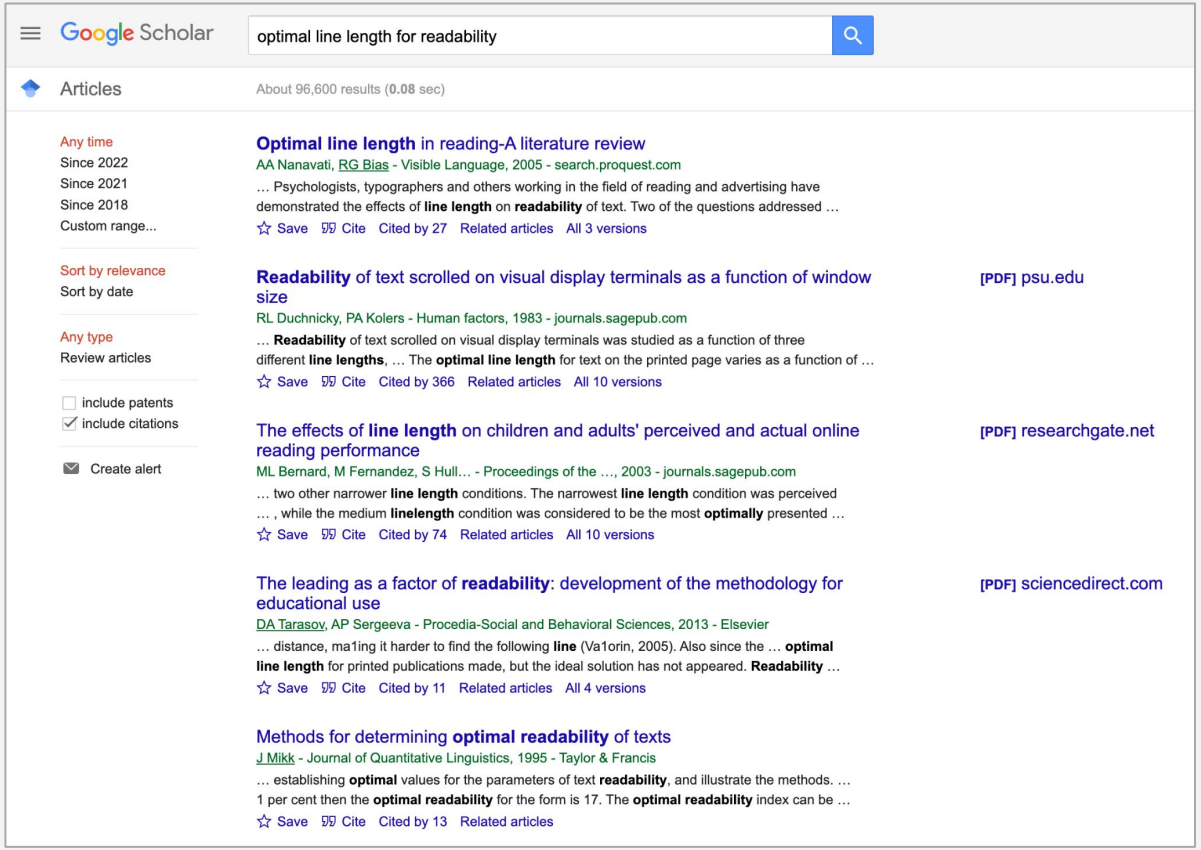
Responsive layouts: Thinking inside the box(es)...

Long text is hard to read

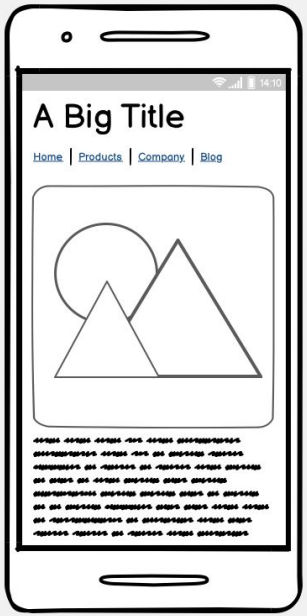


Screenshot of browser showing the text of Jane Austen's *Pride and Prejudice*.

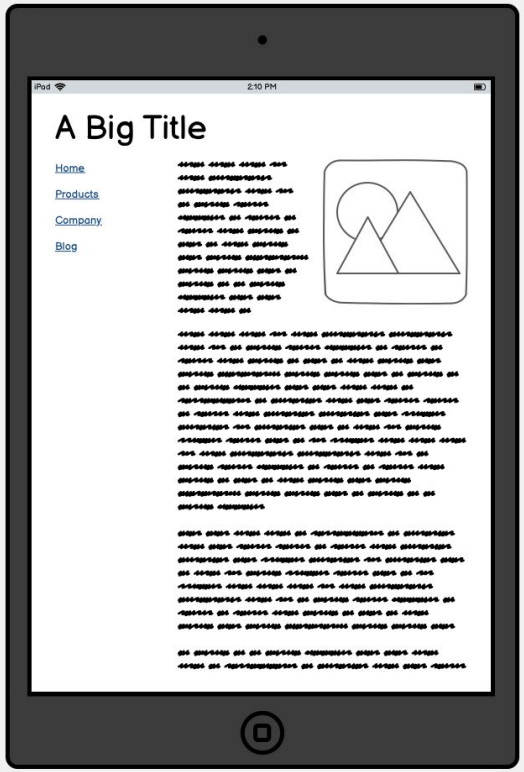
The way the text extends from edge to edge of the browser makes it hard to read.

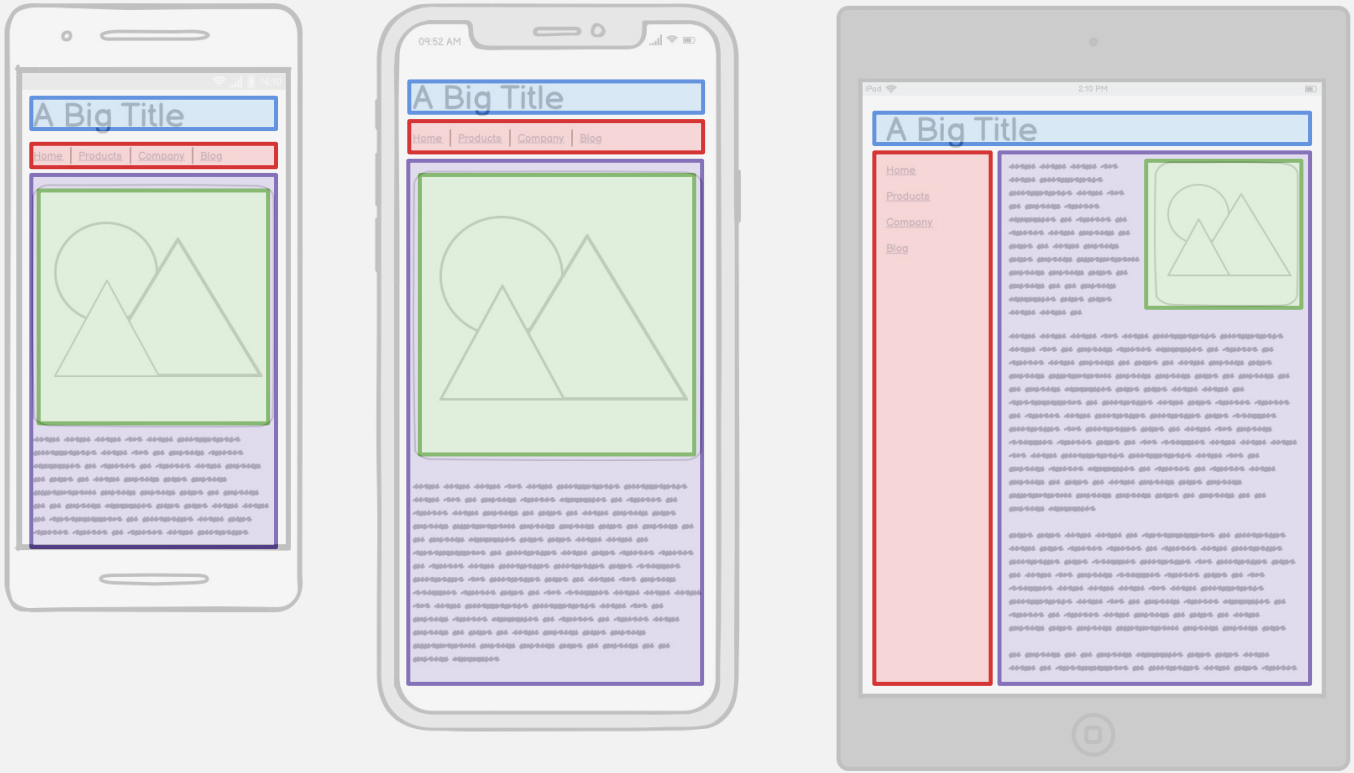


Screenshot of Google Scholar showing results for “optimal line length for readability”


A Big Title
Home | Products | Company | Blog

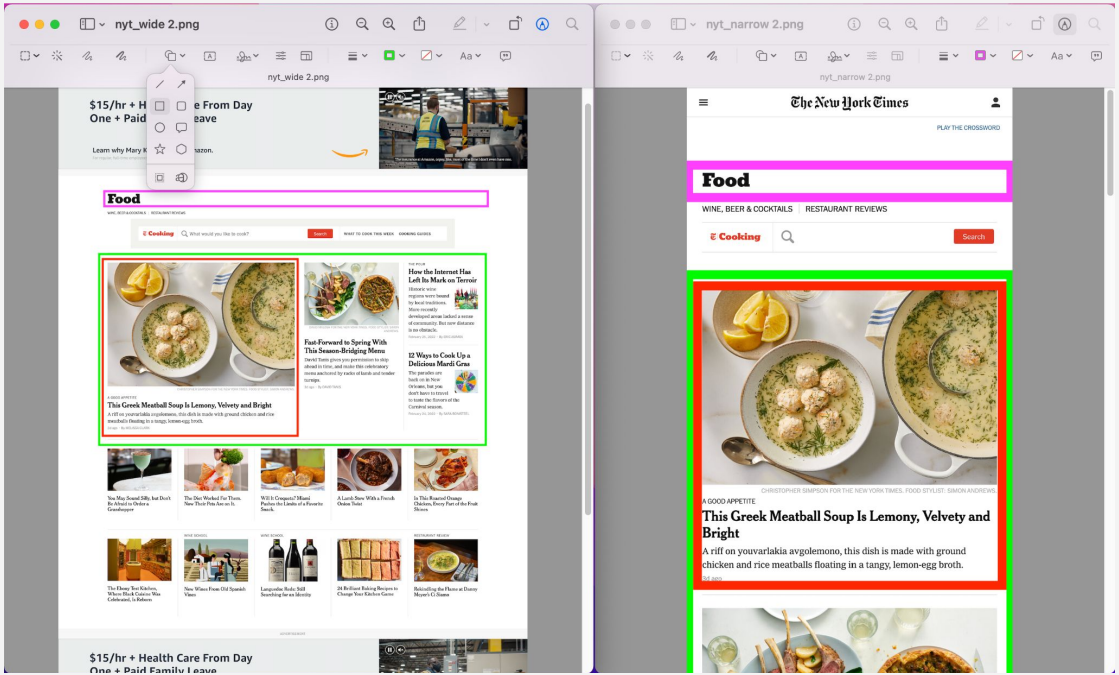

09:52 AM
A Big Title
Home | Products | Company | Blog


iPad
2:10 PM
A Big Title
Home
Products
Company
Blog


A Big Title
Home
Products
Company
Blog
09:52 AM
A Big Title
Home
Products
Company
Blog
iPad
2:10 PM
A Big Title
Home
Products
Company
Blog

In-class activity I: Layouts (Soo)

In-class activity



Look at the two screenshots from the New York Times Food section.

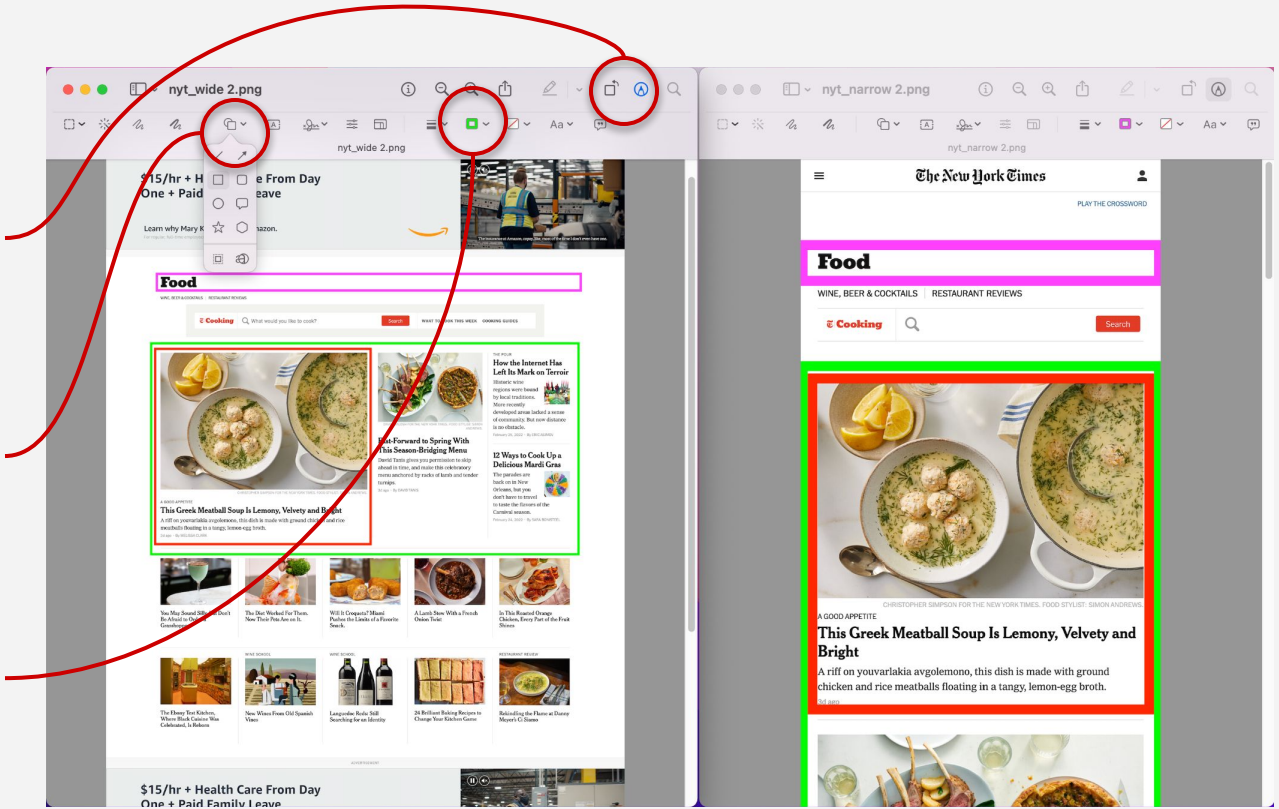
Open them up next to each other in the **Preview** app, and mark up the layout boxes using different colors.

How to use the Preview app to mark up images

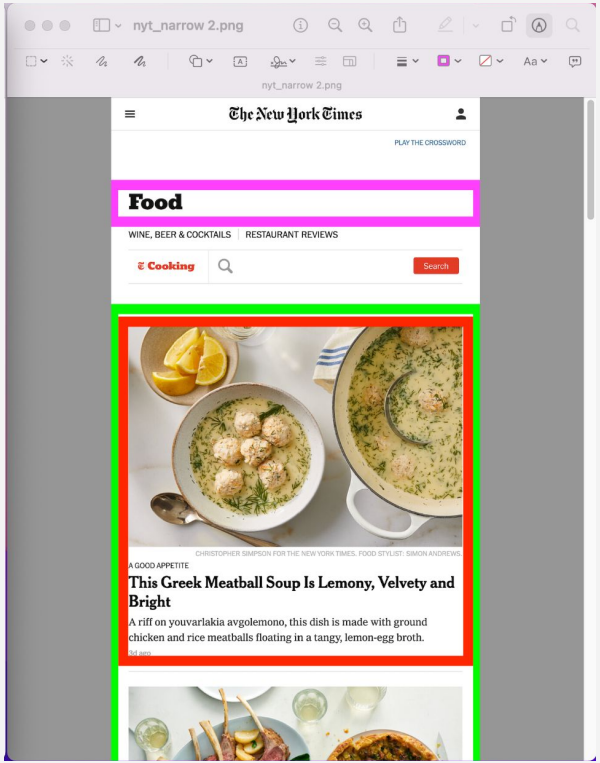
1. Click on this marker button

2. Click on shape to add shape

3. Change outline colors



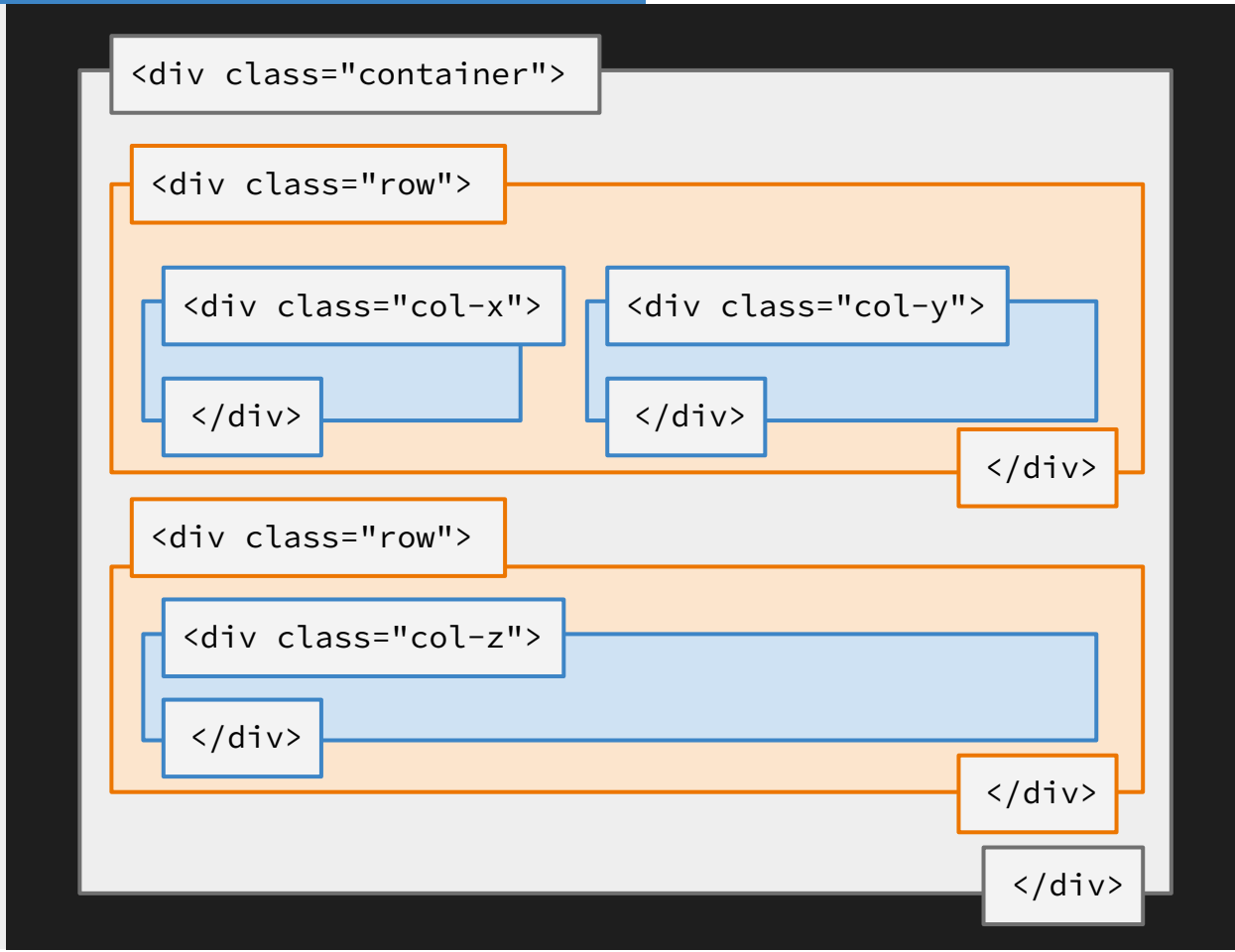
In-class activity: 15 minutes



HTML/CSS frameworks

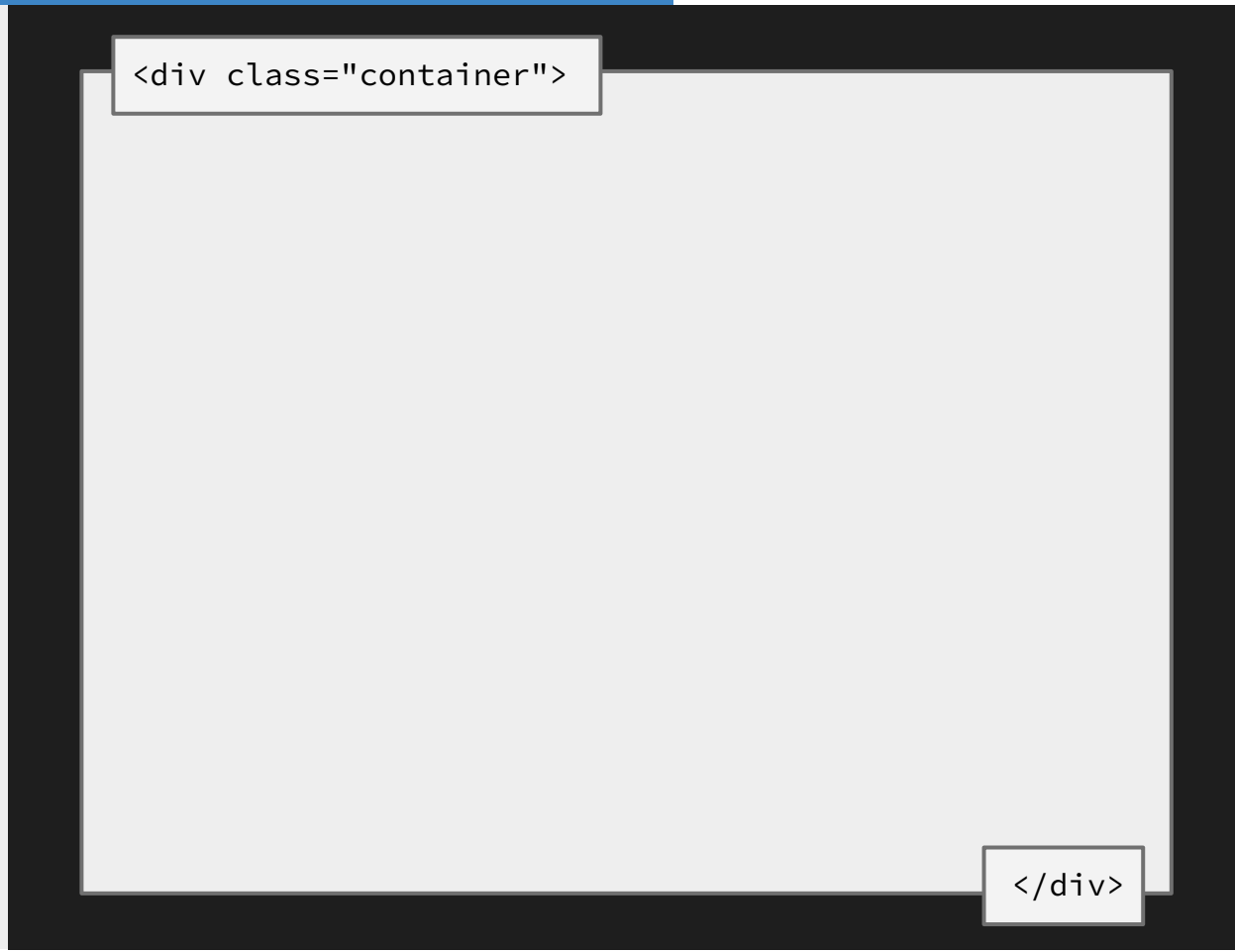
How HTML/CSS frameworks work

They all basically work like this.



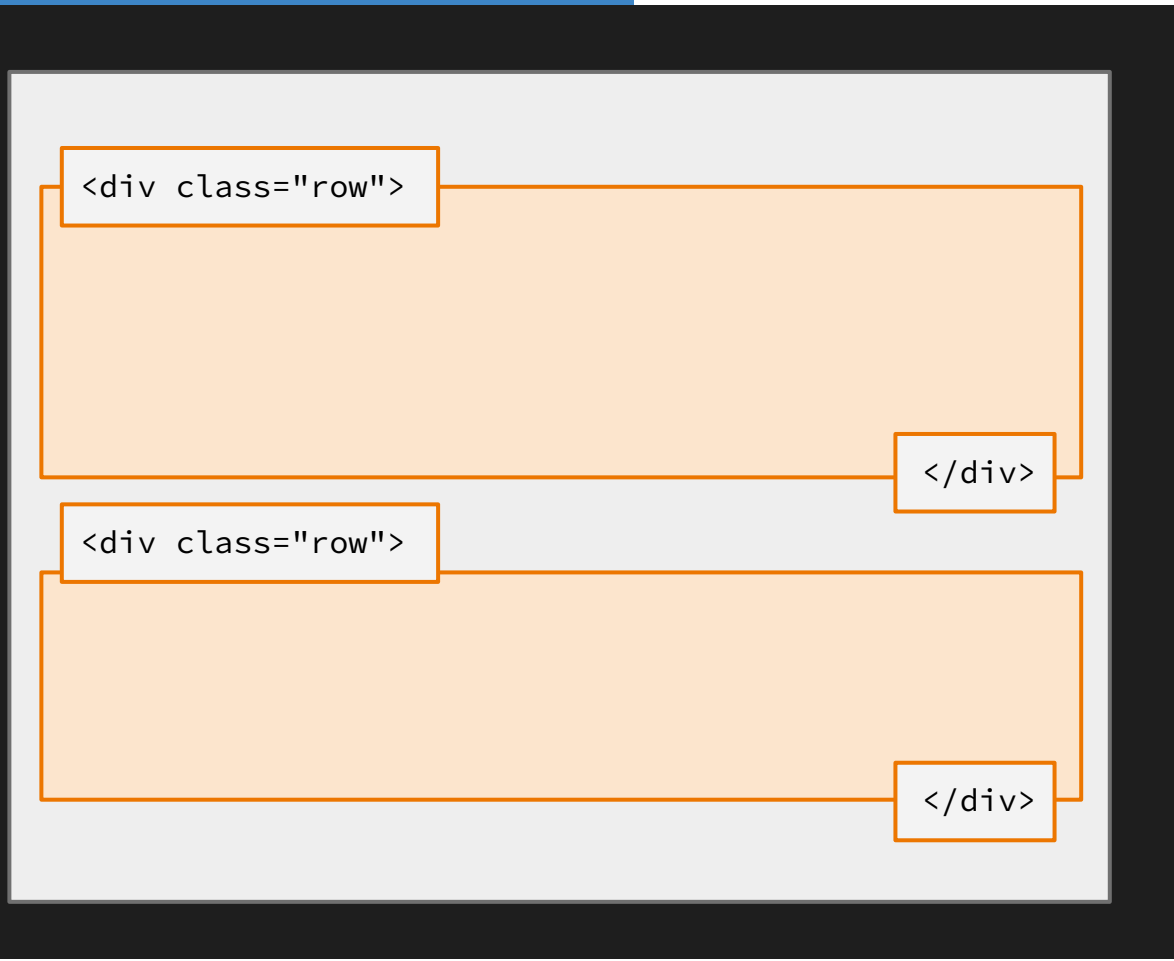
How HTML/CSS frameworks work

First, there's a big container. Usually it has some kind of **max-width** applied to it.



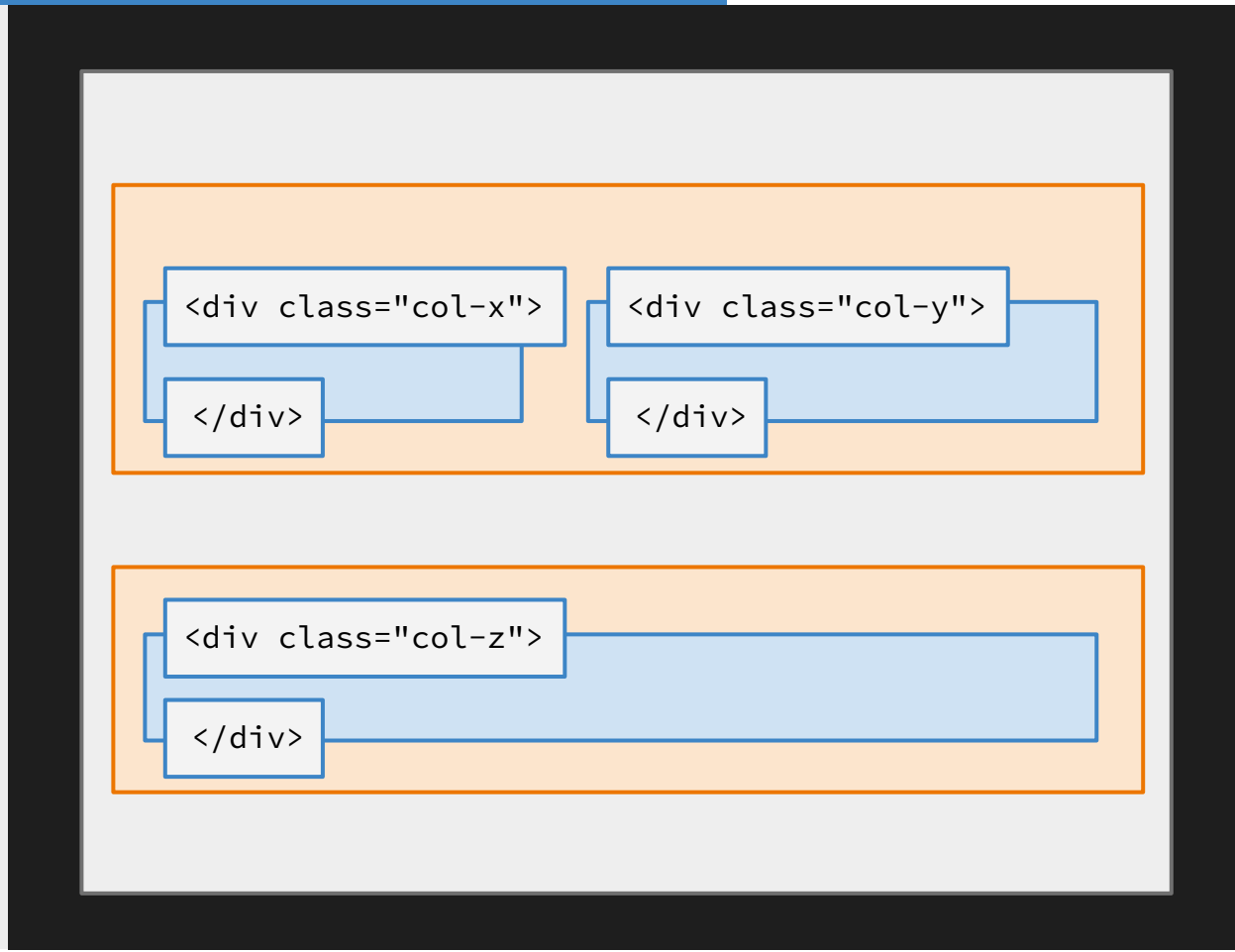
How HTML/CSS frameworks work

Within the container, you place rows to separate your content.



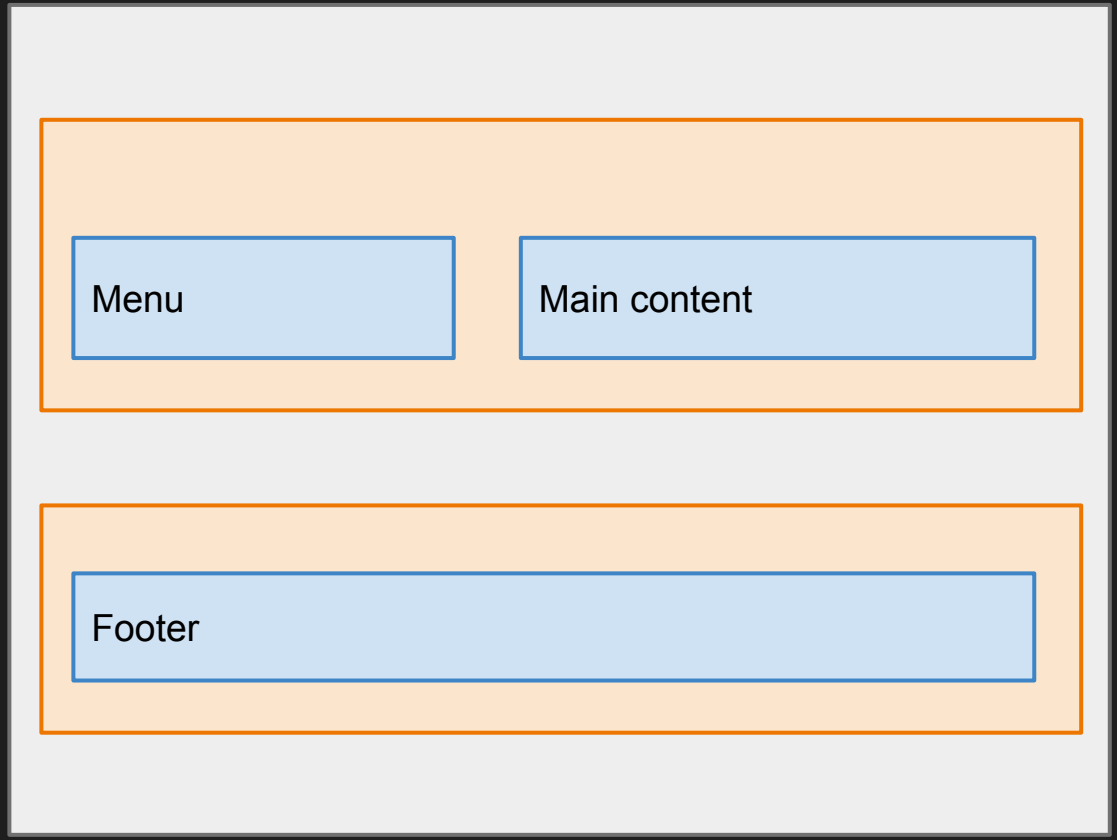
How HTML/CSS frameworks work

Within each row, you place columns to further separate your content.



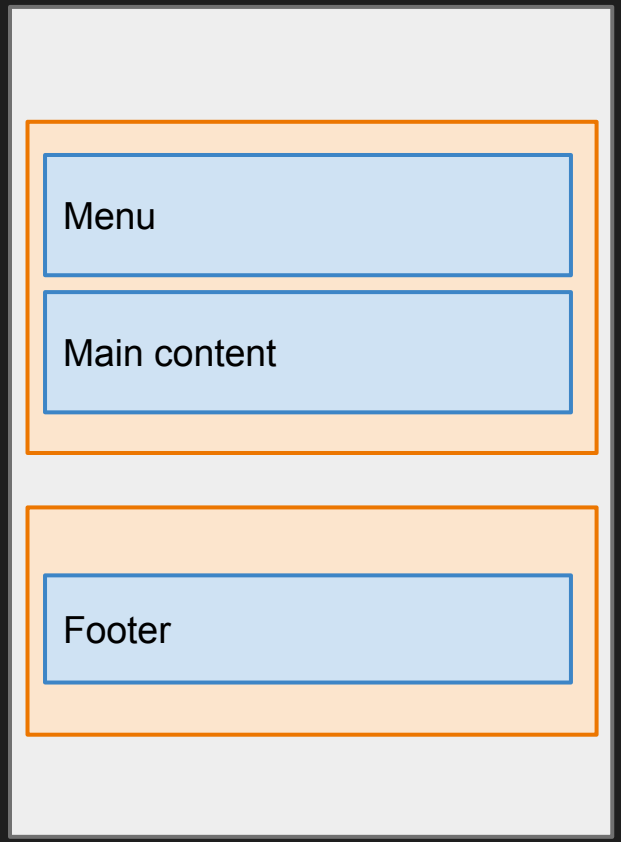
How HTML/CSS frameworks work

On a bigger screen, the page could look something like this.



How HTML/CSS frameworks work

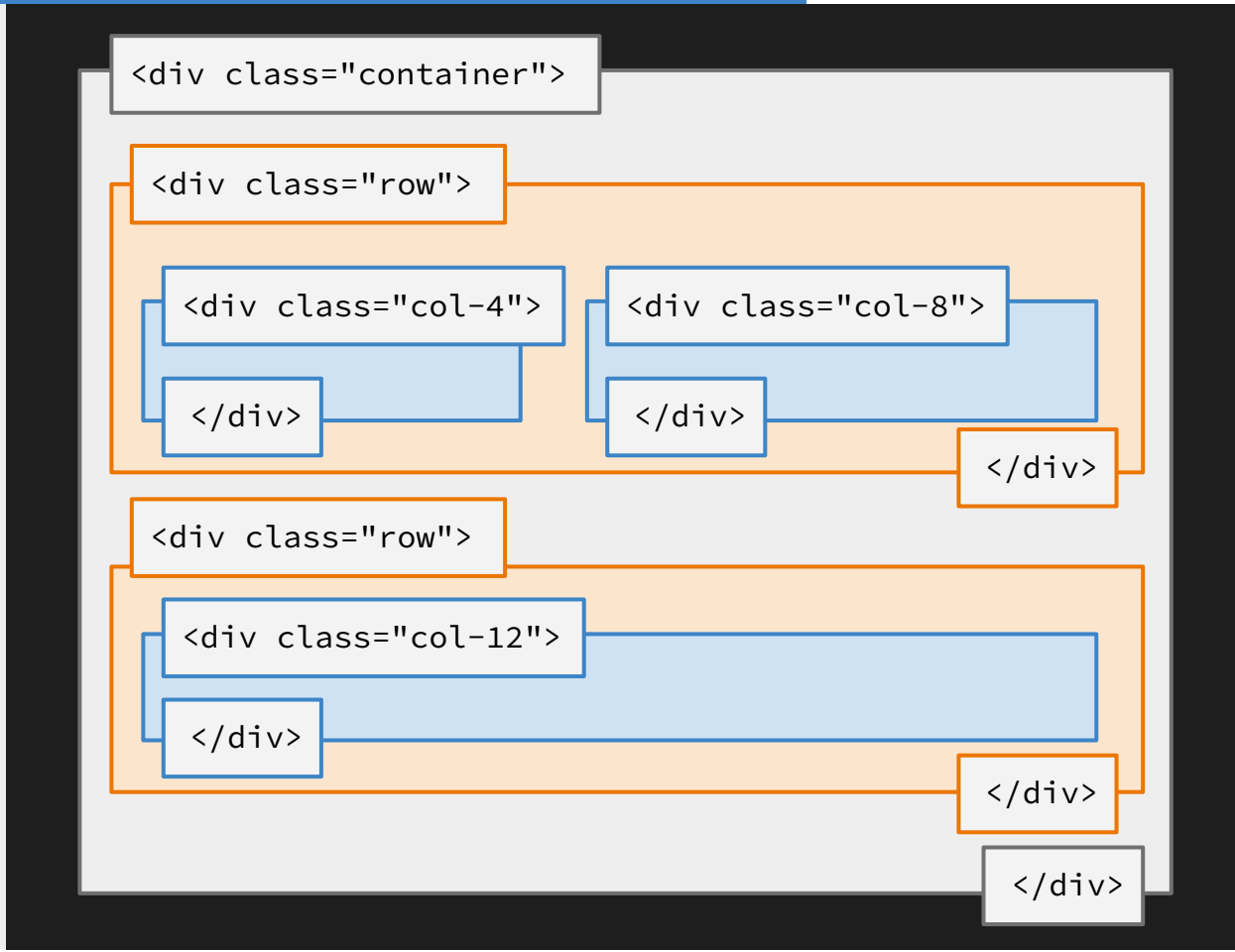
On a smaller screen, the page could look something like this.



How HTML/CSS frameworks work

Usually, frameworks come in 12-column layouts.

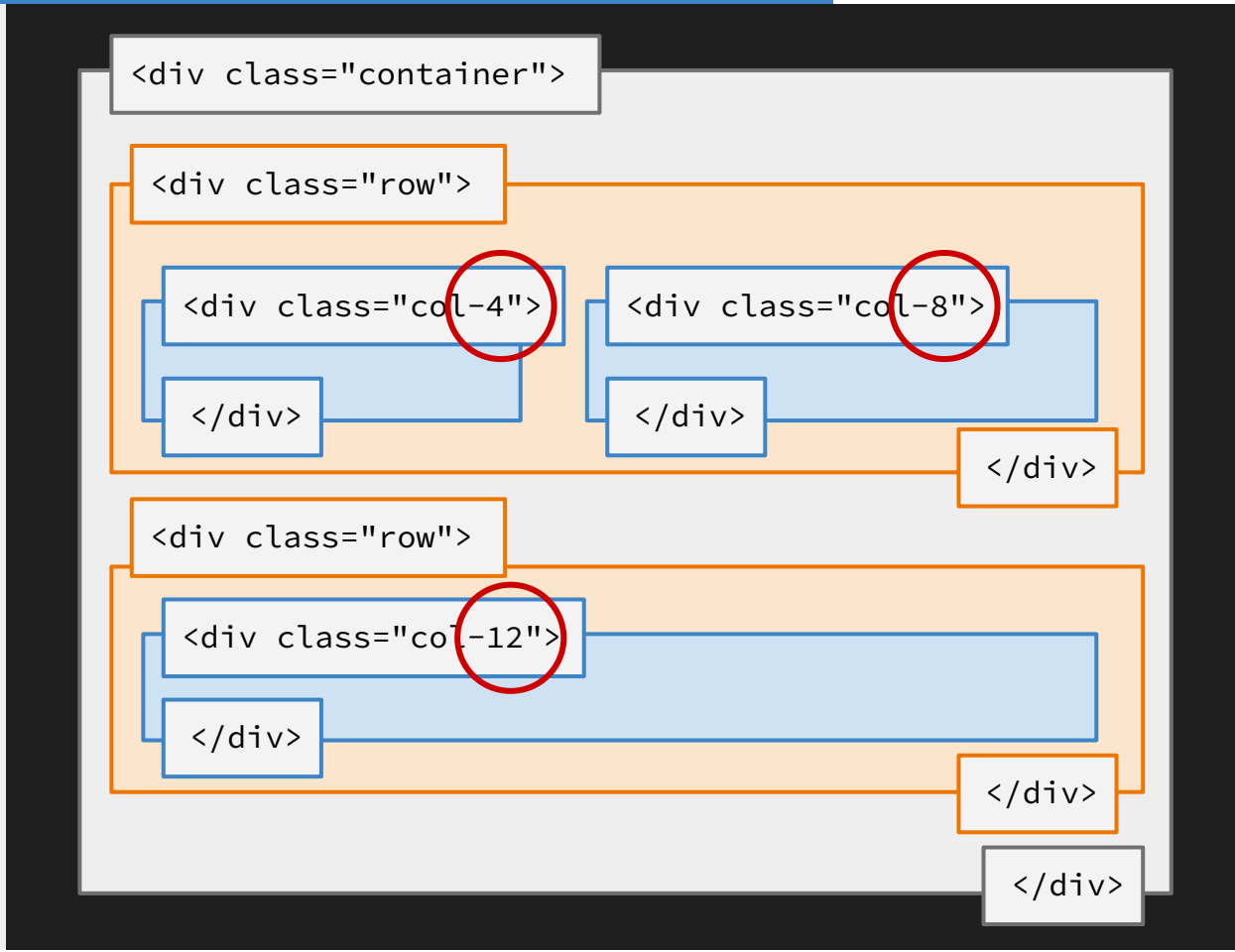
That means, the **columns** of each **row** need to add up to **12**.



How HTML/CSS frameworks work

Usually, frameworks come in 12-column layouts.

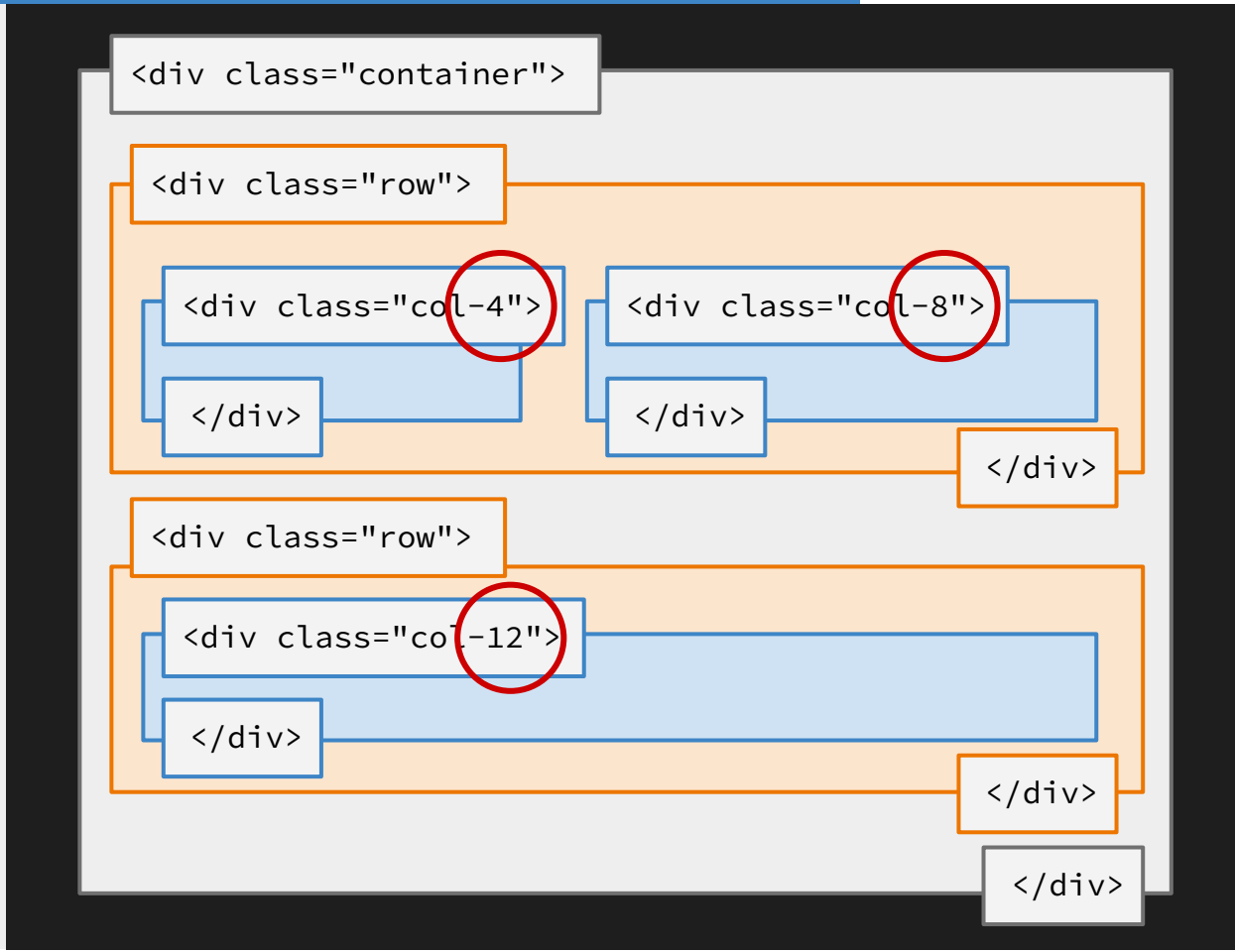
That means, the **columns** of each **row** need to add up to **12.**



How HTML/CSS frameworks work

Usually, frameworks come in 12-column layouts.

That means, the **columns** of each **row** need to add up to **12**.



Some tips

By default, western browsers display elements from **left to right** and **top to bottom**, just like how one would read in English. (You can change this with CSS for different languages.)

It's easiest to structure and order your HTML that way, too. (But `float` can be tricky.)

Open frameworks.html

What questions do you have?

More to explore

More to explore

`position`

multiple-column layout

grid layout, flexbox

There are other types of block arrangements we won't cover in class.

You can explore these on your own.

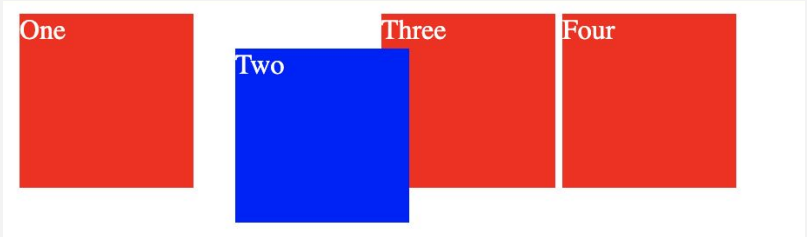
More to explore

`position`

multiple-column layout

grid layout, flexbox

The CSS property **`position`** allows you to fine-tune exactly how elements are laid out on your web page.



Yo

`media_queries.html` from the examples.

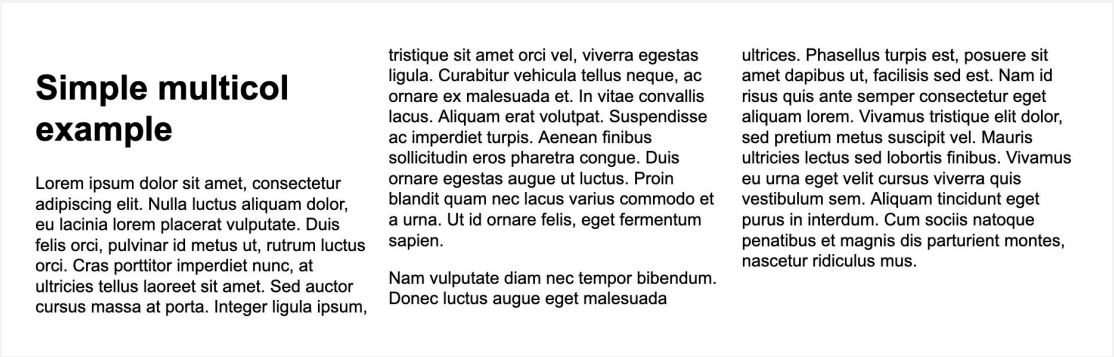
More to explore

position

multiple-column layout

grid layout, flexbox

The multiple-column layout lets you style elements like a print newspaper.



Simple multicol example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet. Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et. In vitae convallis lacus. Aliquam erat volutpat. Suspendisse ac imperdiet turpis. Aenean finibus sollicitudin eros pharetra congue. Duis ornare egestas augue ut luctus. Proin blandit quam nec lacus varius commodo et a urna. Ut id ornare felis, eget fermentum sapien.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

This can be bad to use because readers aren't used to reading like this on a website!

More to explore

position

multiple-column layout

grid layout, flexbox

[CSS grid layouts](#) are similar to [flexbox](#), but can be used in two dimensions. It's very similar to the HTML/CSS grid frameworks we discussed, except you don't need separate "row" elements.

Be careful: CSS Grid Layouts are a type of CSS method. But people also say "CSS grid layouts" when referring to styling a website in a grid format.

More to explore

position

multiple-column layout

grid layout, flexbox

Internet Explorer and some mobile browsers don't support **grid layouts** or **flexbox** very well.

Homework

https://journ220.github.io

Previous final projects

- https://zhongleqi.github.io/leqizhong_final_reresubmit/home.html
- https://jiyuntsai.github.io/project-ousd-library-closure/
- https://juliametraux.github.io/vasculitis/
- https://madtaub.github.io/photography-portfolio/
- https://oliviazhaoxu.github.io/
- https://mengyuan616.github.io/jacksonville_story/
- https://zhiwae97.github.io/Nature_CA/
- https://bscannestra.github.io/final/

Break

Meet back in 15 minutes.

Note to lecturers:
Pause recording and mute

start Zoom recording + captions

In-class activity II: Wireframes (Yoli)

A portfolio can be creative!

https://kazielmanawal.me/

https://lobenichou.com/

https://kwonjs.github.io/#about

https://daviddeloso.com/

https://carlaastudillo.com/

https://www.gurmanbhatia.com/

https://www.tylerjfisher.com/

https://nausheenhusain.github.io/index.html

http://stephaniestamm.com/

https://luciovilla.com/

END

https://journ220.github.io