

# J220

## Coding for Journalists

LECTURE 5

February 28, 2022

PROMPTS

Use iClicker.com to log your attendance.

start Zoom recording + captions

# Agenda

Log attendance via iClicker

Homework review

## **BREAK**

CSS flow

HTML frameworks

In-class activity

Homework

# Homework review

## Question

What is the difference between the `<title>` tag found nested in `<head>` and a headline wrapped in an `<h1>` inside of the `<body>` tag?

## Answer

- `<title>` tag is for the browser
- `<title>` shows up in the tab and in search results
- `<h1>` is seen by the reader on the page
- `<h1>` is the headline of the page
- `<title>` and `<h1>` can be the same text

## Which property is used to change text color?

- color
- foreground-color
- font-color
- text-color
- background-color

**Bonus question: Of these options, which are valid CSS properties?**

- color
- foreground-color
- font-color
- text-color
- background-color

What questions  
do you have?



# Differences between **id** and **class**

## HTML

```
<p id="first" class="intro">This is the first paragraph!</p>
<p class="intro">This is the 2nd paragraph with a "class" intro, but it's not the first paragraph.</p>
<p class="intro">This is the 3rd paragraph with a "class" intro, but it's not the first paragraph.</p>
```

## CSS

```
#first {
  font-size: 20px;
}
.intro {
  color: white;
}
```

**id** Only used **ONCE** in HTML. Used to style **ONE** element.

CSS selector is **#**

**class** Used to style **multiple** elements.

CSS selector is **.**

## Write a CSS style that changes all `<h3>` elements with the following values:

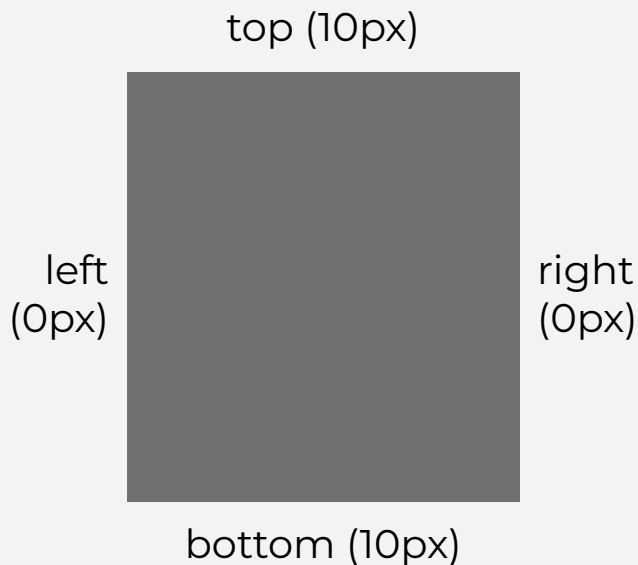
- the background of the elements should have the value of `#000000`
- the font color should be `#ffffff`
- the padding of the elements should be 10 pixels all around

```
h3 {  
  background-color: #000000;  
  color: #ffffff;  
  padding: 10px;  
}
```

No points taken off for indentation (it's weird in bCourses).

What questions  
do you have?

# margin vs. padding



Any of these are acceptable ( just choose one):

✓ `margin: 10px 0px 10px 0px;`  
*top right bottom left*

✓ `margin: 10px 0px;`  
*top+bottom left+right*

✓ `margin: 10px 0px 10px;`  
*top sides bottom*

✓ `margin-top: 10px; margin-bottom: 10px;`

These shortcuts  
also apply to  
**padding!**

# margin vs. padding

## Remember

**margin** can have values of **auto** and **negative numbers**

**padding** can only have values of zero or higher!

**Write a CSS style that changes all <p> elements with a class of "introduction" the following values:**

- the margins at the top and bottom of the elements are both 20 pixels
- the text size is 16 pixels
- the text font is Times New Roman (and has appropriate fallback options)
- the text color has a hex value of #111111
- the line-height is 20 pixels

```
p.introduction {  
  margin: 20px 0;  
  /*  
  ALSO OK:  
  margin: 20px 0 20px 0;  
  margin-top: 20px; margin-bottom: 20px;  
  */  
  font-size: 16px;  
  font-family: 'Times New Roman', serif;  
  color: #111111;  
  line-height: 20px;  
}
```

Extra credit half point if you got  
**p.introduction.**

What questions  
do you have?

# 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).
- One **assets** folder is fine. Sometimes, you'll separate out **css**, **js**, **images**, and other **assets**. The more complex the project, the more you'll end up splitting out folders.



# CSS flow

# First, download and unzip examples

<https://journ220.github.io> (Scroll to **Feb. 28**)

I'll also drop the link in Slack.

# CSS flow

## `<div>` and `<span>`

### `display`

block

inline

inline-block

none

### `float`

left

right

overflow

Introducing... **`<div>`** and **`<span>`**

These element tags have no semantic meaning. They're used solely to position and style elements on a page.

By default, browsers display **`<div>`** elements as “block” and **`<span>`** elements as “inline.”

What does that mean?

# CSS flow

`<div>` and `<span>`

**display**

block

inline

inline-block

none

**float**

left

right

overflow

A **display** value of **block** means the item takes up the entire width of a page and acts like a big box.

A **display** value of **inline** means the items act more like individual words. They can break across different lines.

A **display** value of **inline-block** is a combination of the two.

# CSS flow

`<div>` and `<span>`

**display**

block

inline

inline-block

none

**float**

left

right

overflow

A **display** value of **block** means the item takes up the entire width of a page and acts like a big box.

A **display** value of **inline** means the

**Open display.html**

They can break across different lines.

A **display** value of **inline-block** is a combination of the two.

What questions  
do you have?

# CSS flow

`<div>` and `<span>`

`display`

`block`

`inline`

`inline-block`

`none`

`float`

`left`

`right`

`overflow`

We learned about **float** at the last lecture. However, there are a couple issues with using **float** that you should be aware of.

When you float an element, you might create **unintended problems with the flow of the page**.

[Open float.html](#)

What questions  
do you have?



Ara, the good dog (📷 Soo Oh)

**Note to  
lecturers:**  
Pause  
recording

# Break

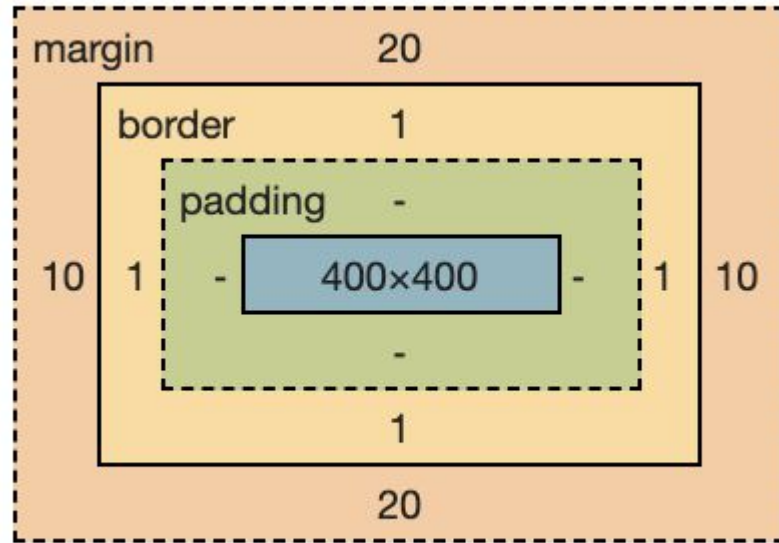
Meet back in 15 minutes.

start Zoom recording + captions

# Review of Box Model

Box model slides start on Slide 73 in previous lecture:

<https://journ220.github.io/assets/static/lecture02-14.pdf>



# 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 width of 600 pixels. At 600 pixels and wider, the font-size becomes **3em**.

# What questions do you have?



# 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

# What questions do you have?

# 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

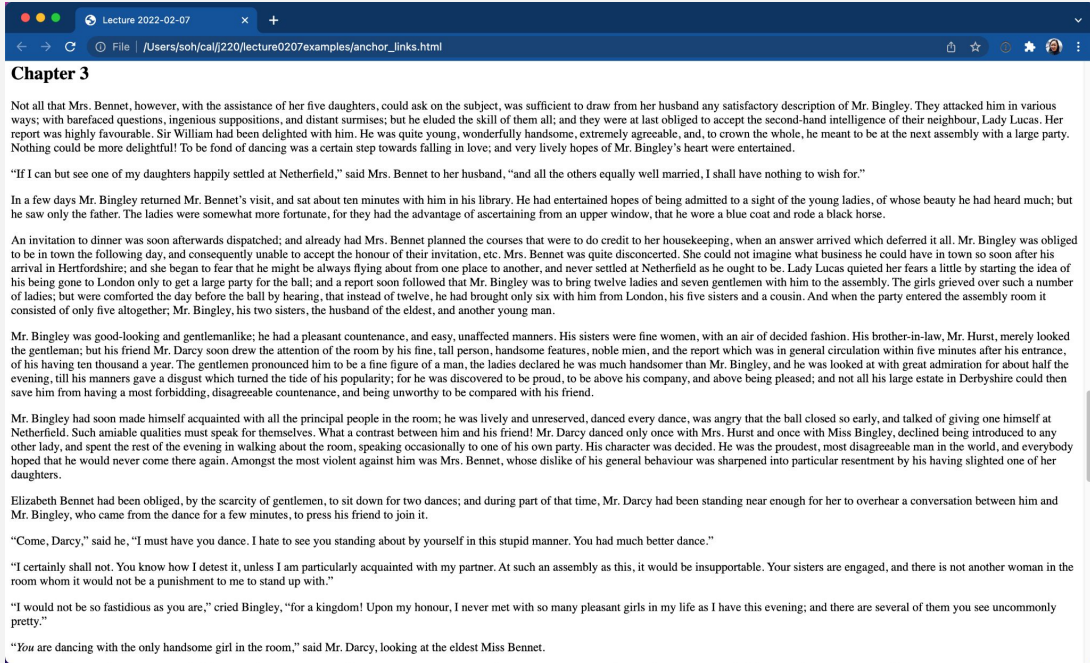
```
.medium-only {  
  display: none;  
  color: orange;  
}  
  
/* For browser windows between 600 and 1199 pixels  
wide */  
@media (min-width: 400px) and (max-width: 1199px) {  
  .medium-only {  
    display: block;  
  }  
}
```

Open `media_queries_complex.html`

# What questions do you have?

# CSS 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.



Google Scholar

optimal line length for readability

Articles About 96,600 results (0.08 sec)

Any time  
Since 2022  
Since 2021  
Since 2018  
Custom range...

Sort by relevance  
Sort by date

Any type  
Review articles

☐ include patents  
☒ include citations

☒ Create alert

**Optimal line length in reading-A literature review**  
AA Nanavati, RG Bias - Visible Language, 2005 - search.proquest.com  
... Psychologists, typographers and others working in the field of reading and advertising have demonstrated the effects of **line length** on **readability** of text. Two of the questions addressed ...  
☆ Save Cite Cited by 27 Related articles All 3 versions

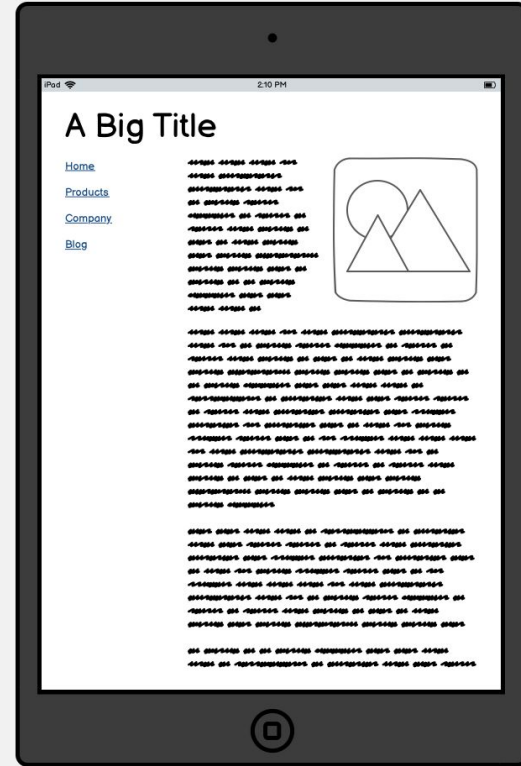
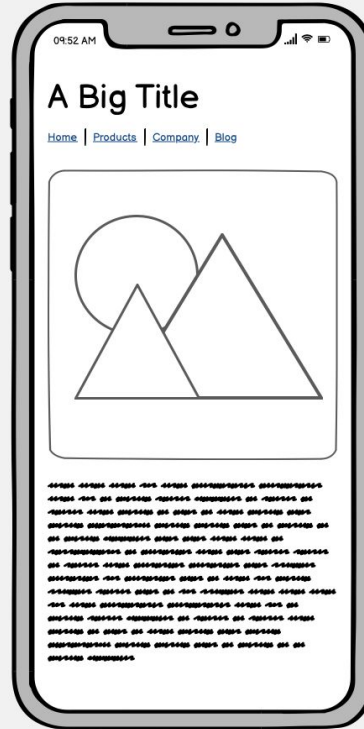
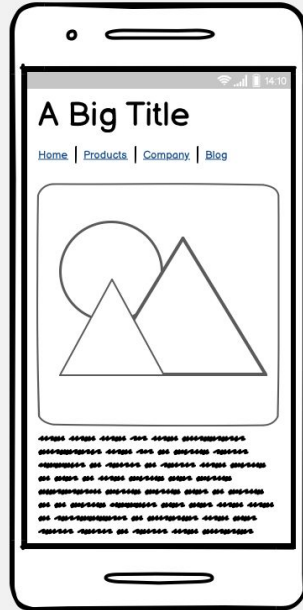
**Readability of text scrolled on visual display terminals as a function of window size**  
RL Duchnick, PA Kolers - Human factors, 1983 - journals.sagepub.com  
... **Readability** of text scrolled on visual display terminals was studied as a function of three different **line lengths**. ... The **optimal line length** for text on the printed page varies as a function of ...  
☆ Save Cite Cited by 366 Related articles All 10 versions

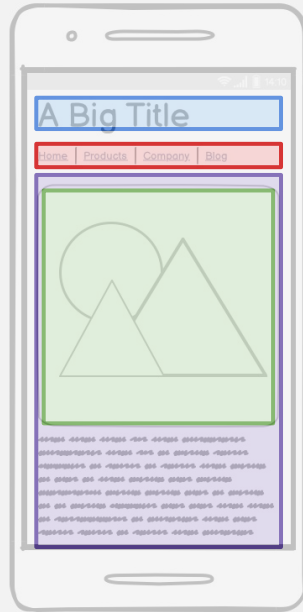
**The effects of line length on children and adults' perceived and actual online reading performance**  
ML Bernard, M Fernandez, S Hull... - Proceedings of the ..., 2003 - journals.sagepub.com  
... two other narrower **line length** conditions. The narrowest **line length** condition was perceived ... , while the medium **linelength** condition was considered to be the most **optimally** presented ...  
☆ Save Cite Cited by 74 Related articles All 10 versions

**The leading as a factor of readability: development of the methodology for educational use**  
DA Tarasov, AP Sergeeva - Procedia-Social and Behavioral Sciences, 2013 - Elsevier  
... distance, making it harder to find the following **line** (Va1orin, 2005). Also since the ... **optimal line length** for printed publications made, but the ideal solution has not appeared. **Readability** ...  
☆ Save Cite Cited by 11 Related articles All 4 versions

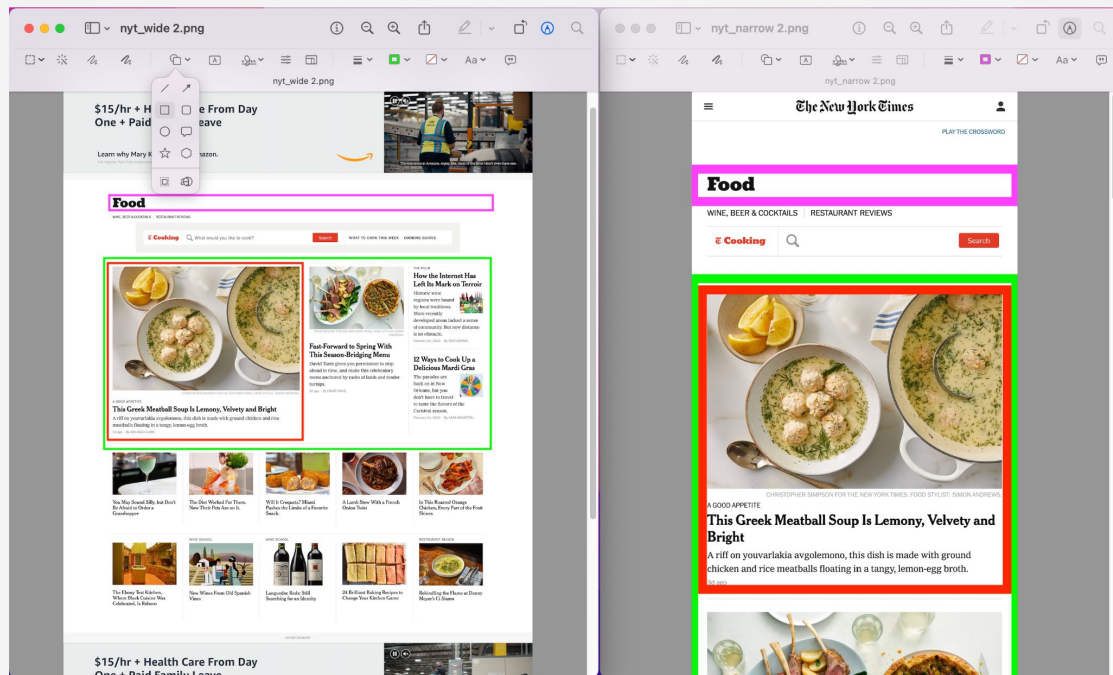
**Methods for determining optimal readability of texts**  
J Mikk - Journal of Quantitative Linguistics, 1995 - Taylor & Francis  
... establishing **optimal** values for the parameters of text **readability**, and illustrate the methods. ... 1 per cent then the **optimal readability** for the form is 17. The **optimal readability** index can be ...  
☆ Save Cite Cited by 13 Related articles

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





# In-class activity: Boxes



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

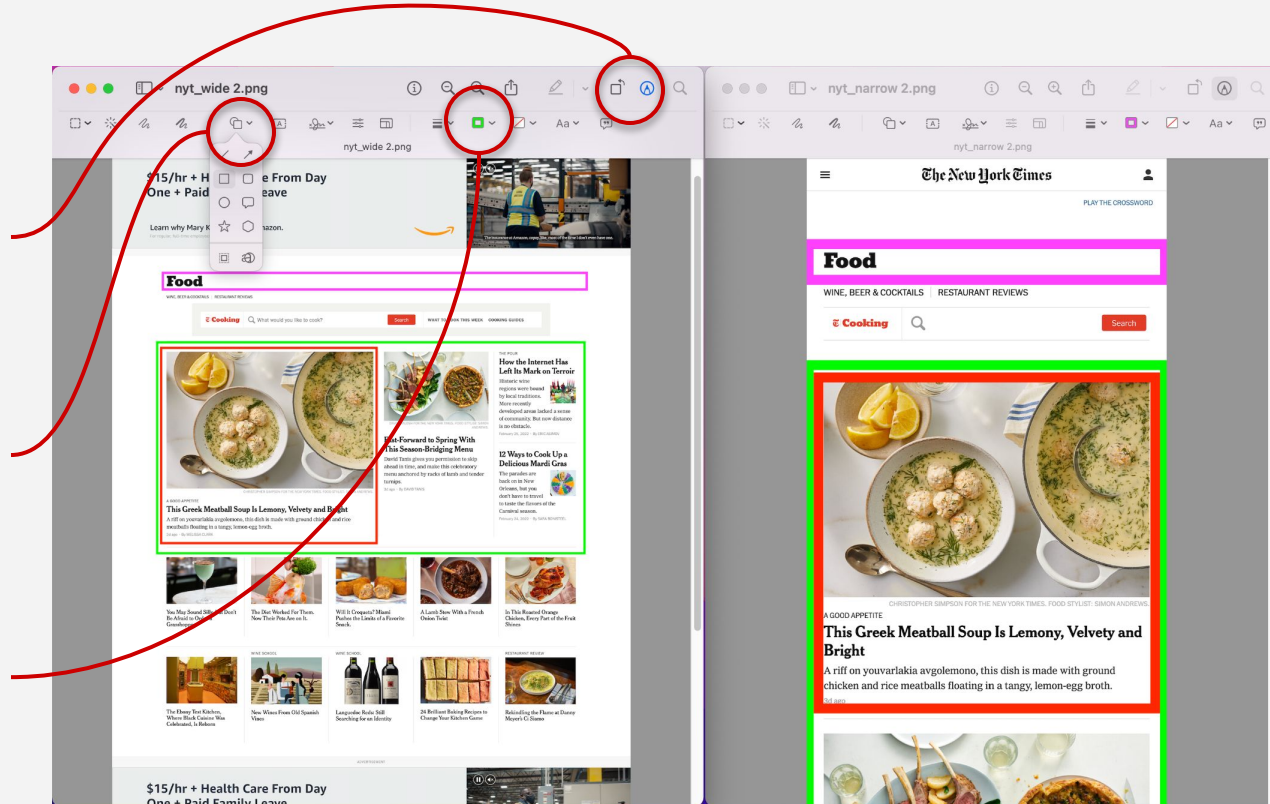
Open them up next to each other in the **Preview** app, and mark up the layout boxes using different colors.

# How to use the Preview app to mark up images

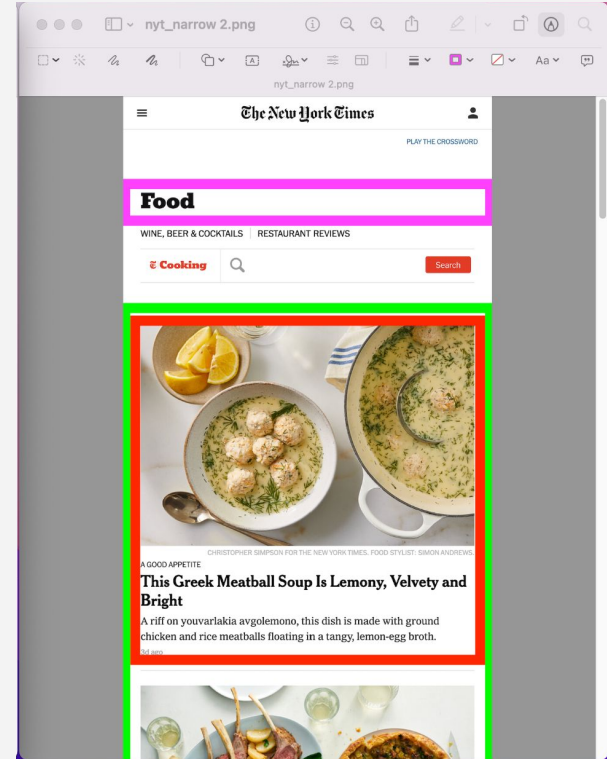
1. Click on this marker button

2. Click on shape to add shape

3. Change outline colors



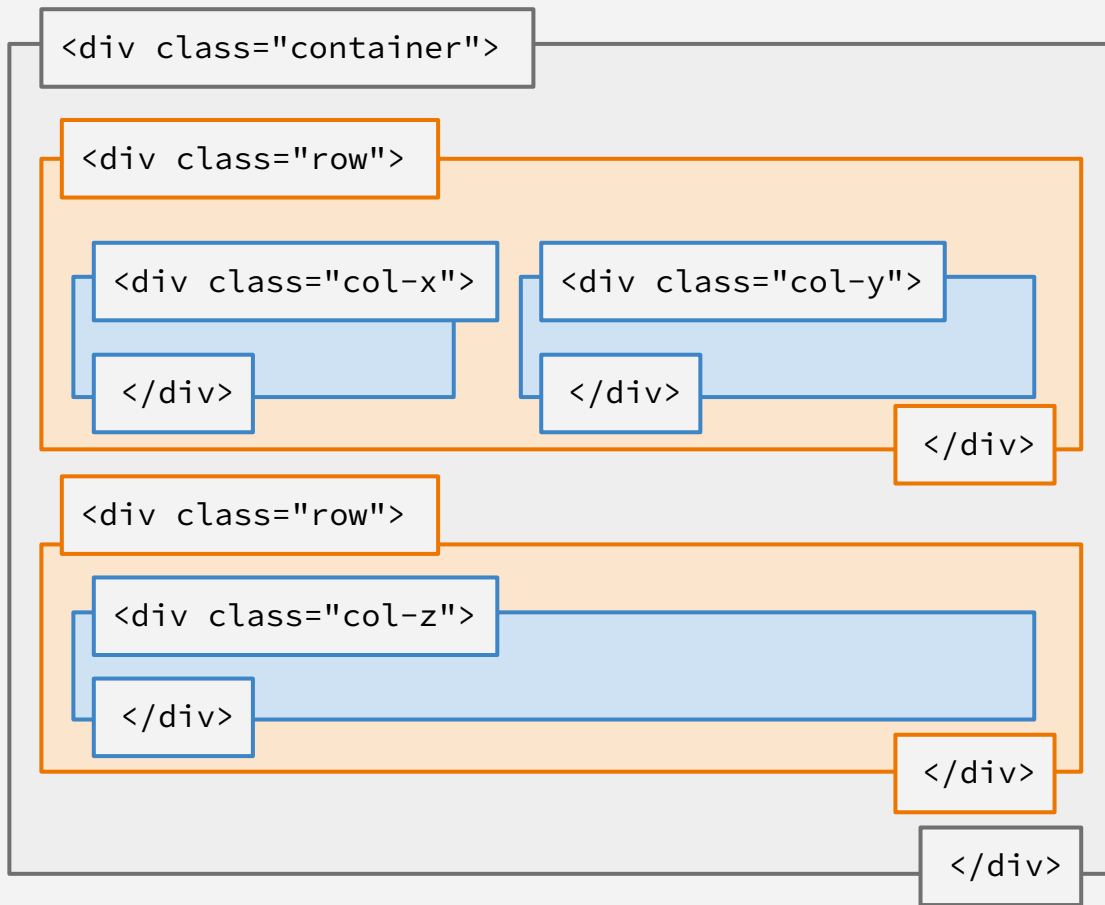
# In-class activity: 15 minutes



# HTML/CSS frameworks

# How HTML/CSS frameworks work

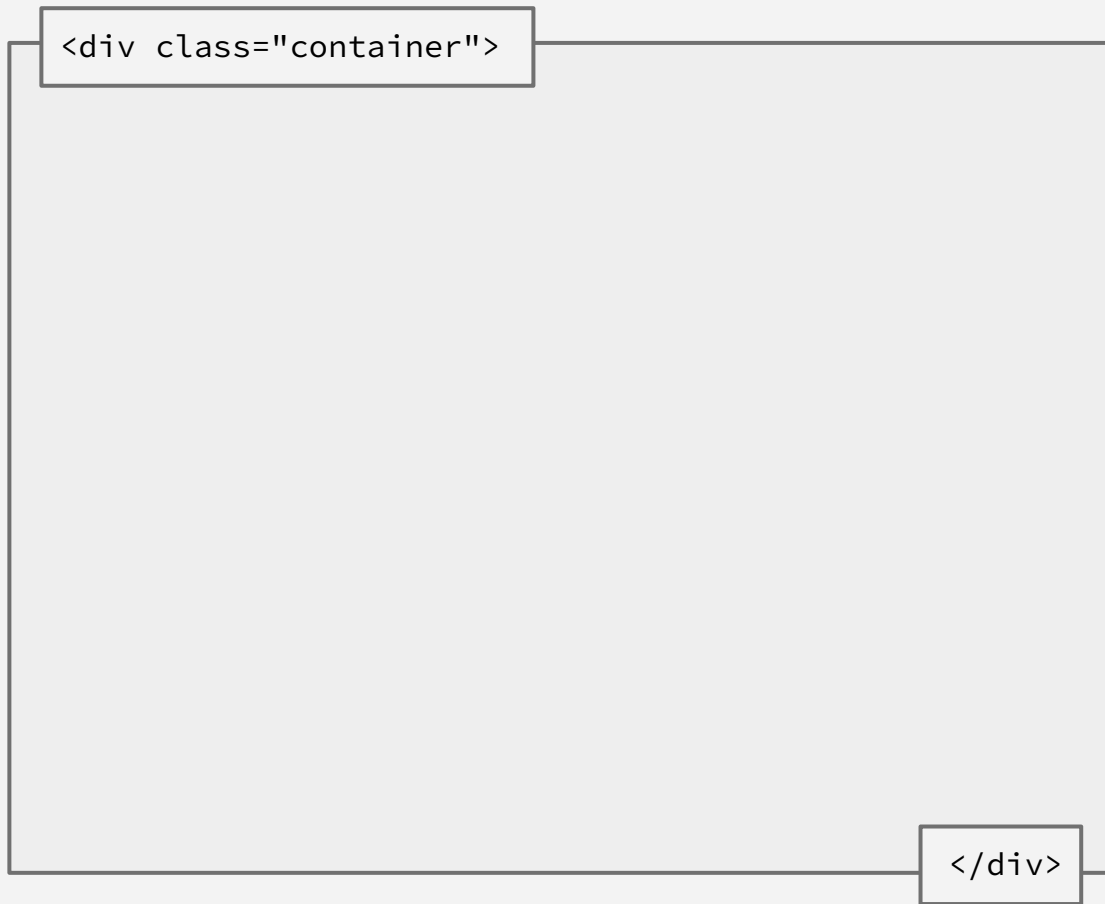
They all basically work like this.





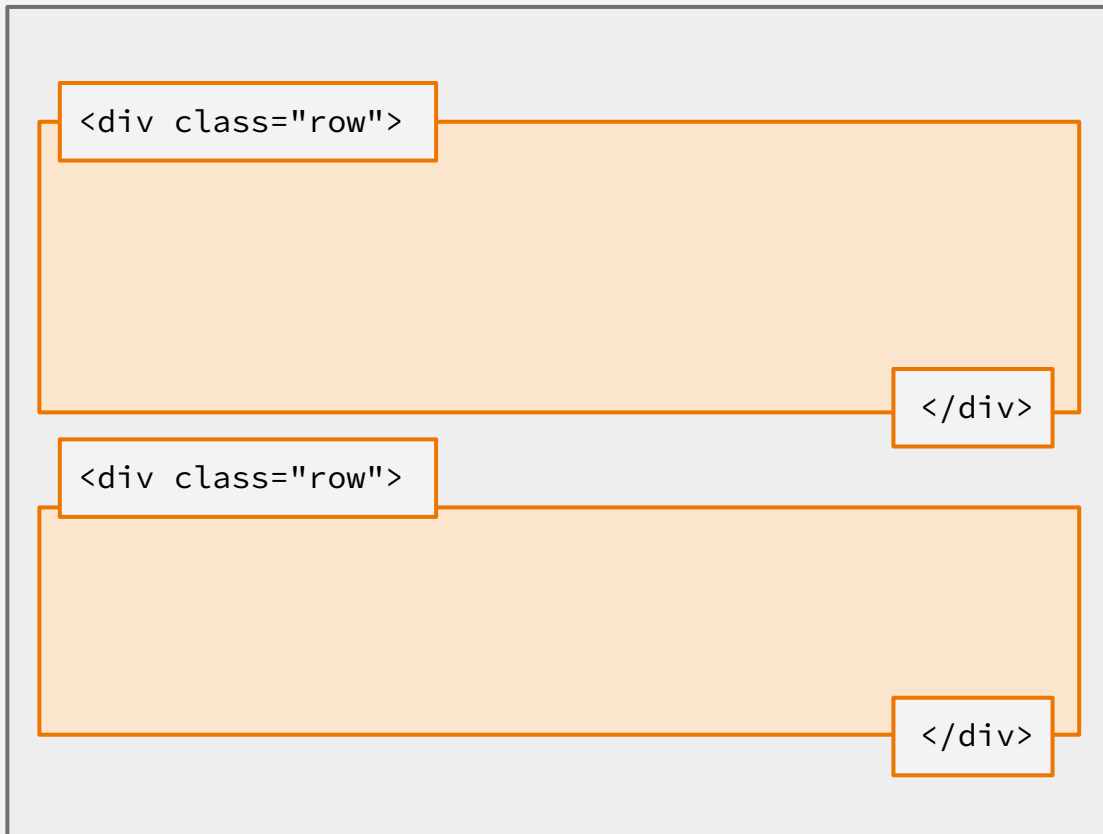
## How HTML/CSS frameworks work

First, there's a big container.



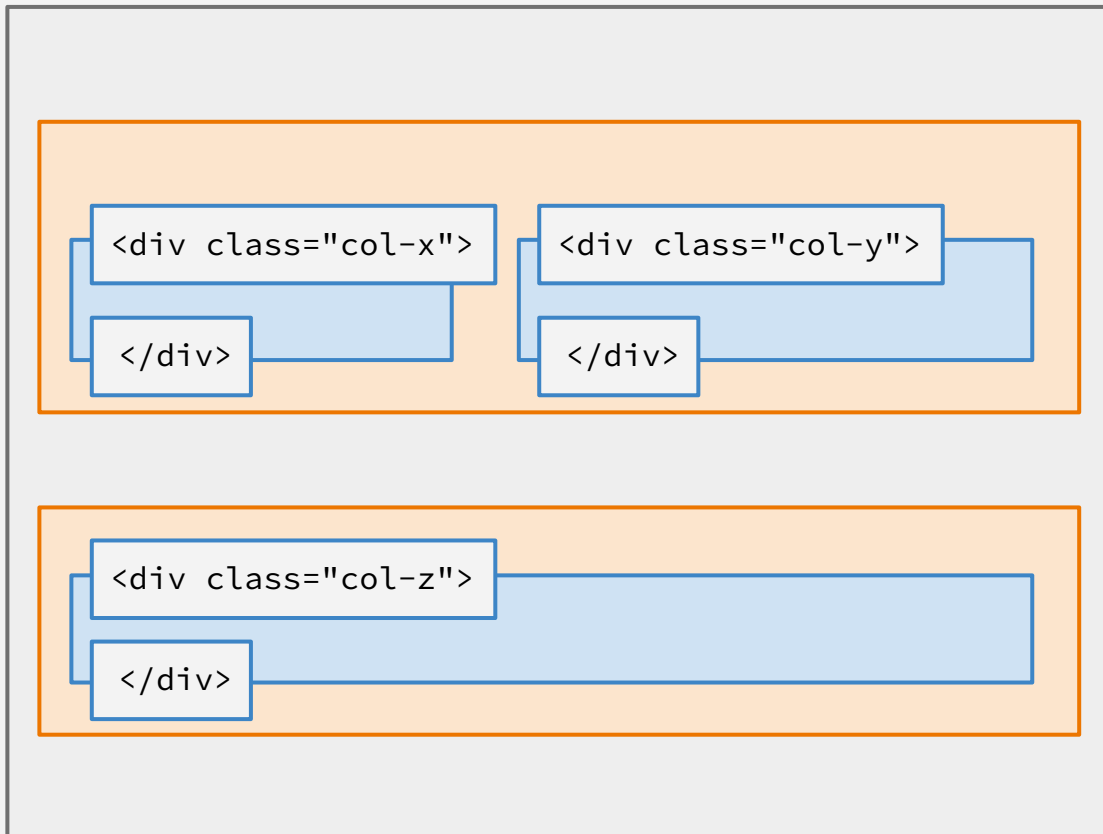
## How HTML/CSS frameworks work

Within the container, you place rows.



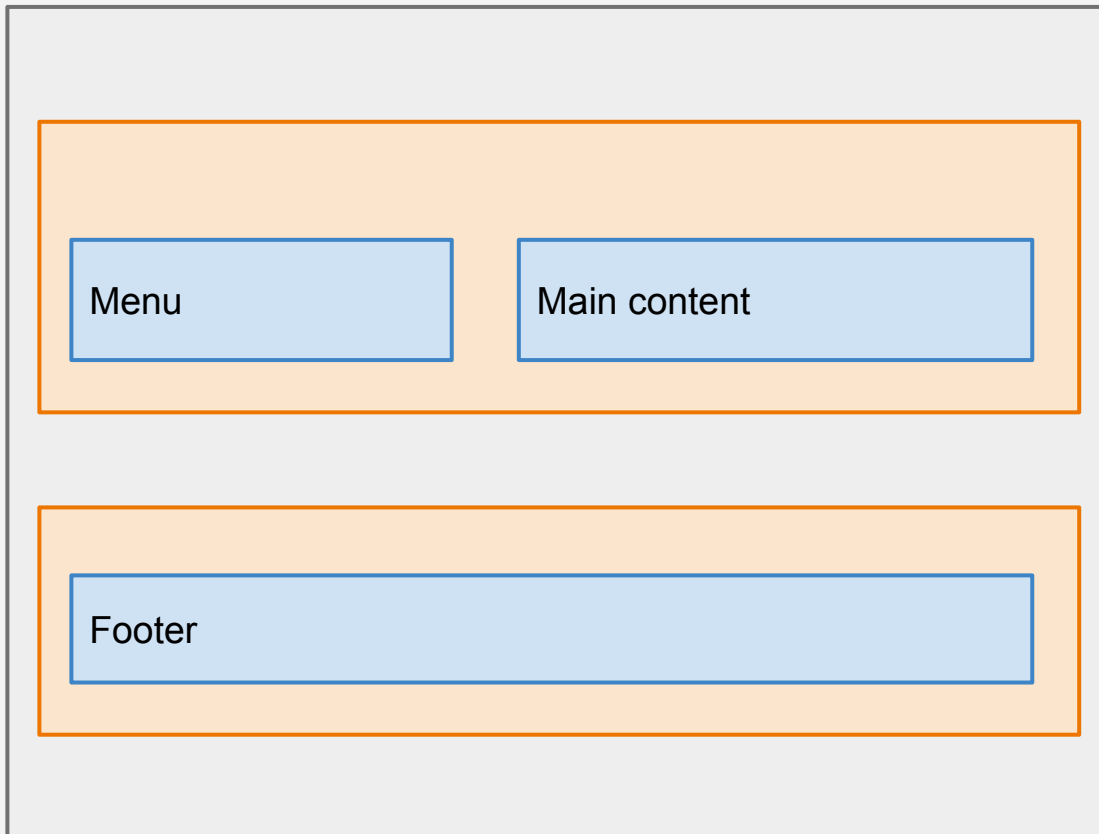
## How HTML/CSS frameworks work

Within each row, you place columns.



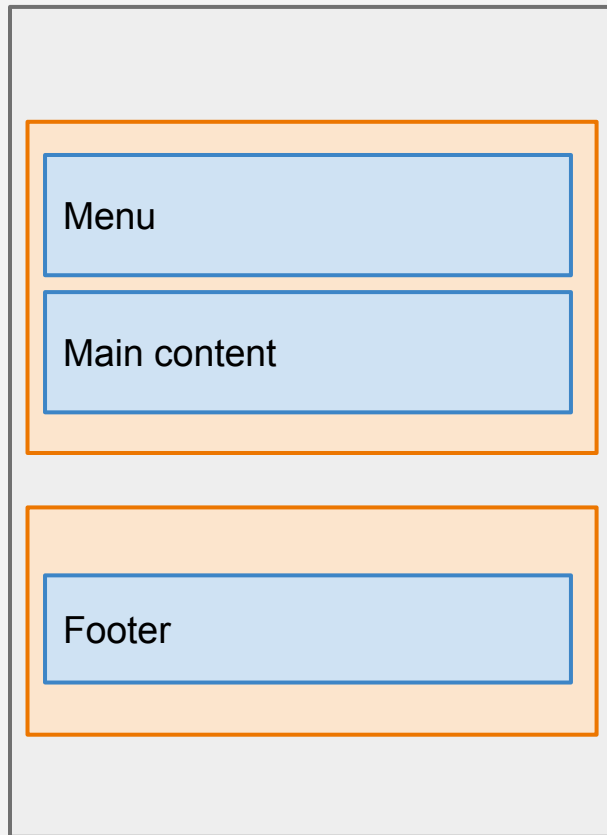
## How HTML/CSS frameworks work

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



## How HTML/CSS frameworks work

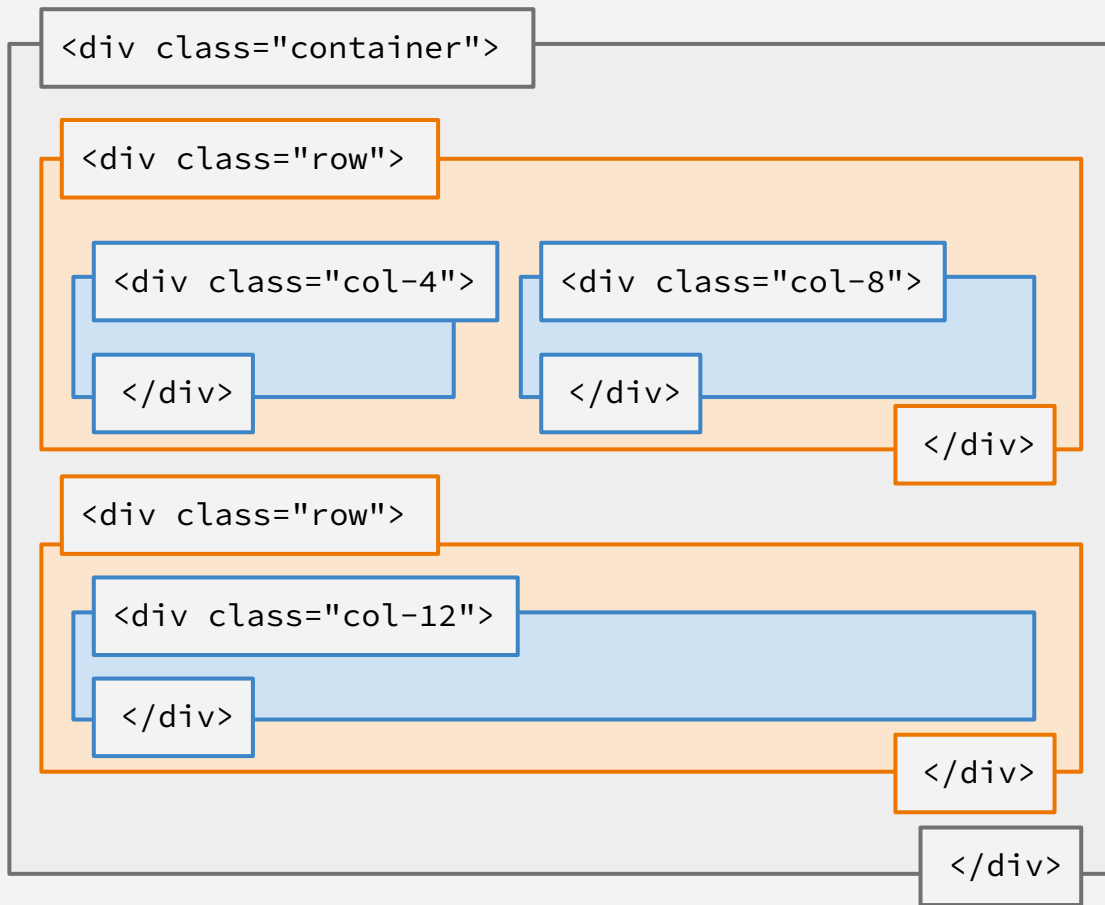
On a smaller screen, the page will look something like this.



## How HTML/CSS frameworks work

Usually, frameworks come in 12-column layouts.

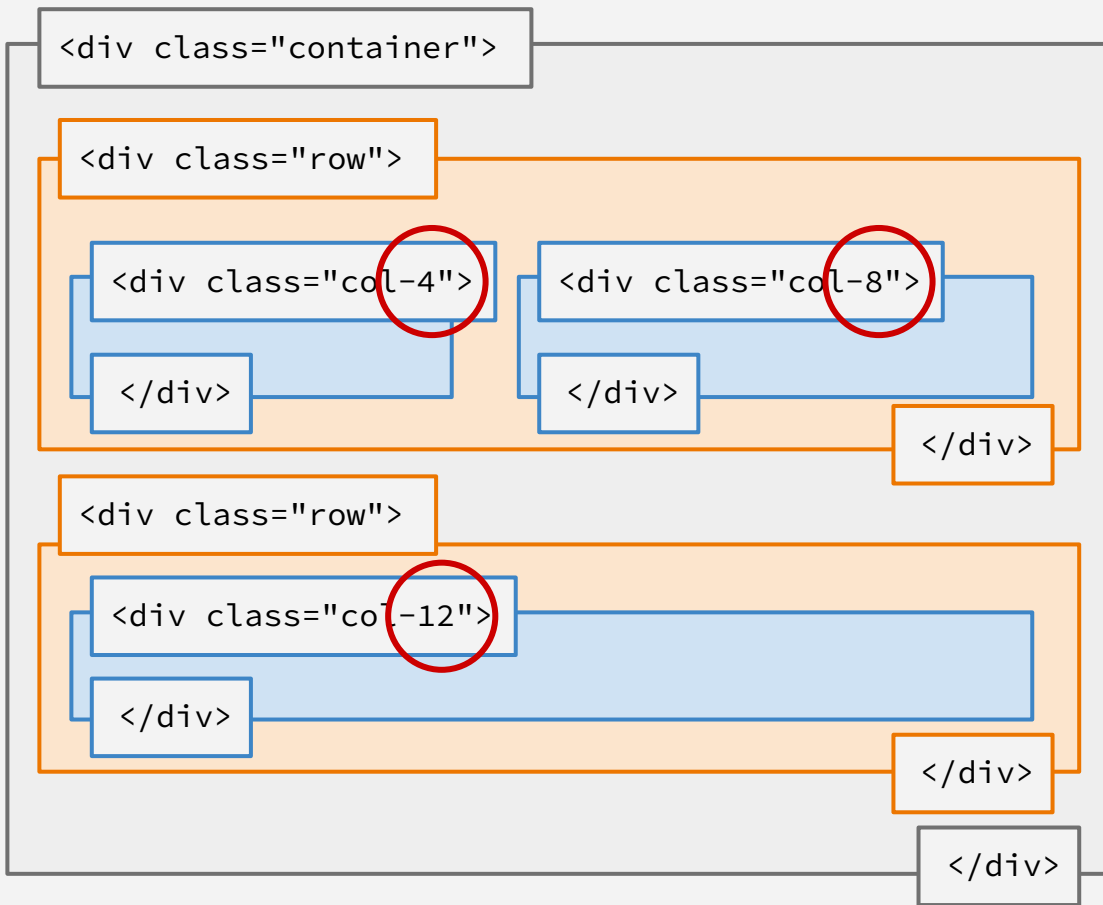
That means, the columns of each row need to add up to 12.



## How HTML/CSS frameworks work

Usually, frameworks come in 12-column layouts.

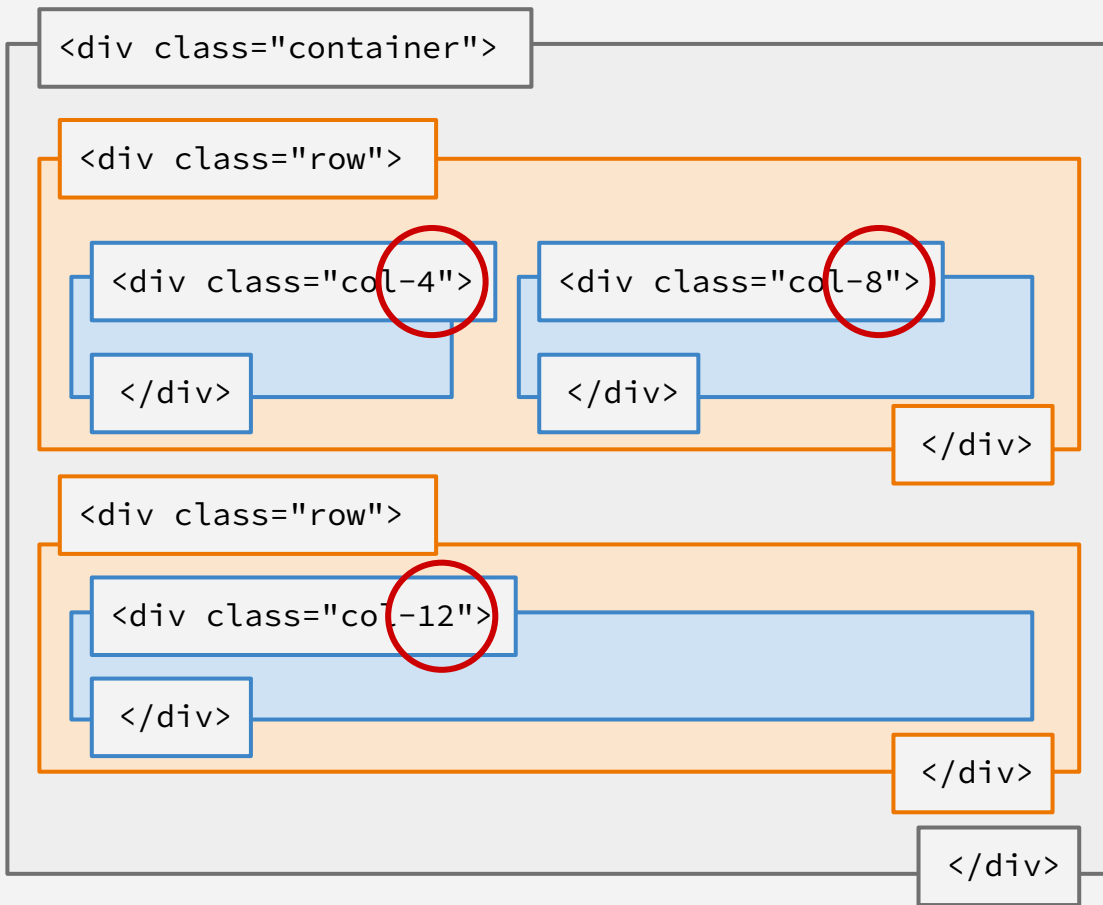
That means, the columns of each row need to add up to 12.



## 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.





# What questions do you have?

# 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

# More to explore

## More to explore

position

multiple-column  
layout

grid layout

There are other types of block arrangements we won't cover in class. You can explore these on your own.

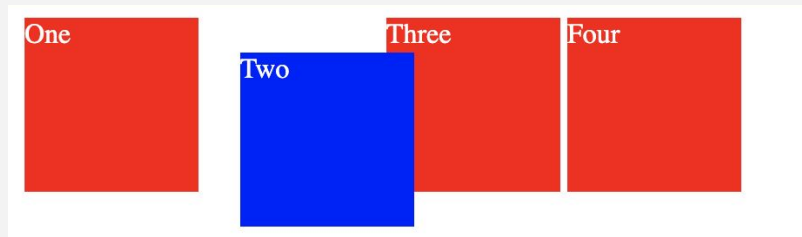
## More to explore

position

multiple-column  
layout

grid layout

The CSS property [position](#) allows you to fine-tune exactly how elements are laid out on your web page.



You can also see an example in **`media_queries.html`** from the examples.

# More to explore

position

multiple-column  
layout

grid layout

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.

## More to explore

position

multiple-column  
layout

grid layout

CSS Grid Layouts are similar to flexbox (we'll learn later), 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

Internet Explorer and some mobile browsers don't support **grid layouts** or **flexbox** very well. HTML/CSS frameworks solve for this problem by using **float**.



# In-class activity

Convert your site portfolio with the **simple-grid.css** file you downloaded from the examples. You'll copy **simple-grid.css** to your portfolio assets folder.

Before we start the activity, let's go over homework...

# Homework

# Final Project

In two weeks, we're going to start learning how to create wireframes. So you should make a decision on what kind of web page you want to make for your final project by March 14. You can redesign a website, or you can create a totally new website.

Sign up for office hours if you'd like to chat about your idea!

# Homework

**In Canvas:** You'll download a PDF of a page with a number of different asset files. You need to create the HTML and the CSS to recreate the PDF.

# In-class activity

Convert your site portfolio with the **simple-grid.css** file you downloaded from the examples. You'll copy **simple-grid.css** to your portfolio assets folder.

**Note to students:** Please help close the windows on your way out