Frameworks & CSS

GENERAL ASSEMBLY

Learning Objectives

- **Use** Materialize
- **Explain** the standards process with CSS
- **Identify** and **utilise** vendor prefixing
- **Use** CSS3 properties and values:
    - Box-shadows and text-shadows
    - Transitions
    - Animations

Agenda

- Materialize
- Standards in CSS
- Vendor Prefixing
- Box Shadows and Text Shadows
- Transitions
- Animations

Next Week

Review

GENERAL ASSEMBLY

Frameworks

GENERAL ASSEMBLY

What is a Framework?

- Again, someone else's code that we use to make our life easier
- Not always just CSS classes, it can include JS as well (for common things like sliders etc.)
- It changes the way we write code!
- They can be ***very*** difficult to overwrite
- They tend to be quite heavy (lots of stuff)

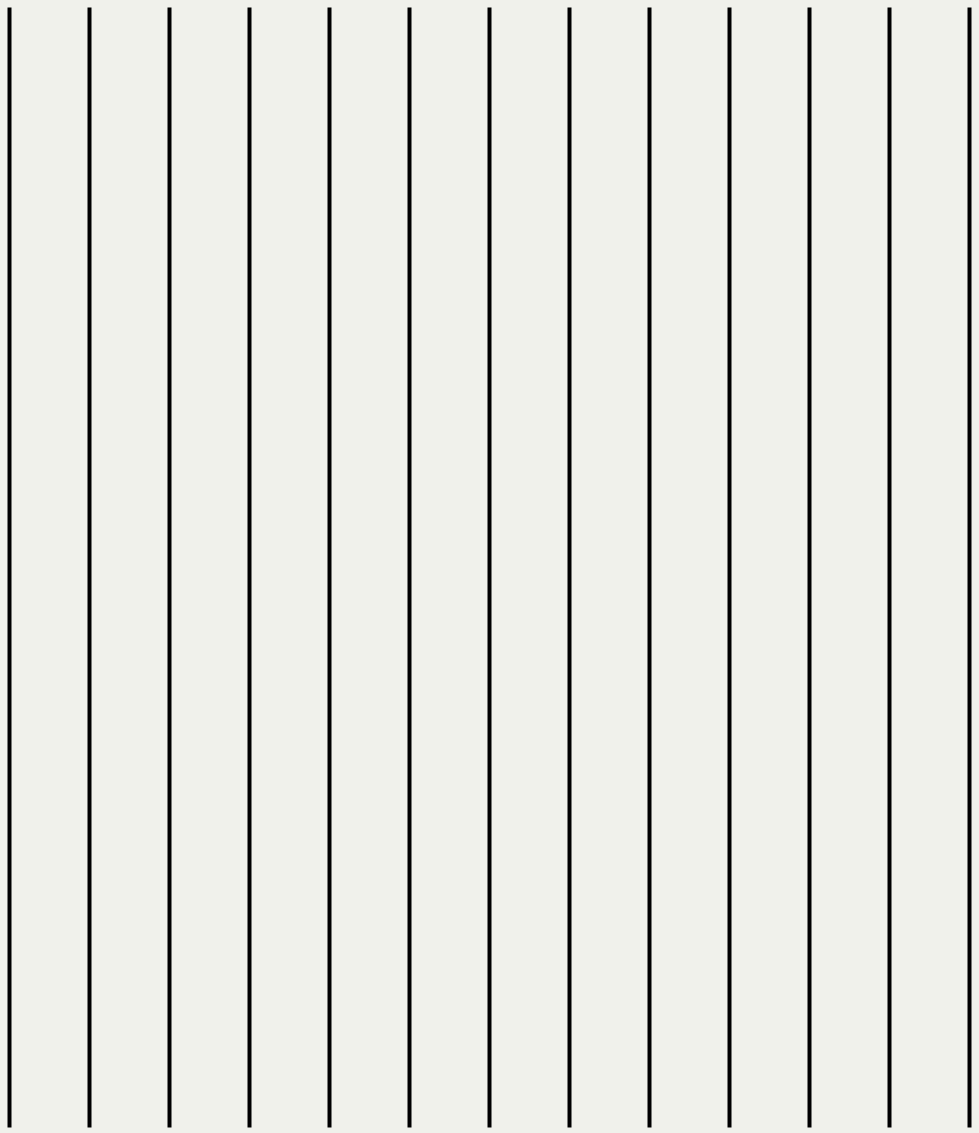
Frameworks

- Bootstrap
- Materialize
- Foundation

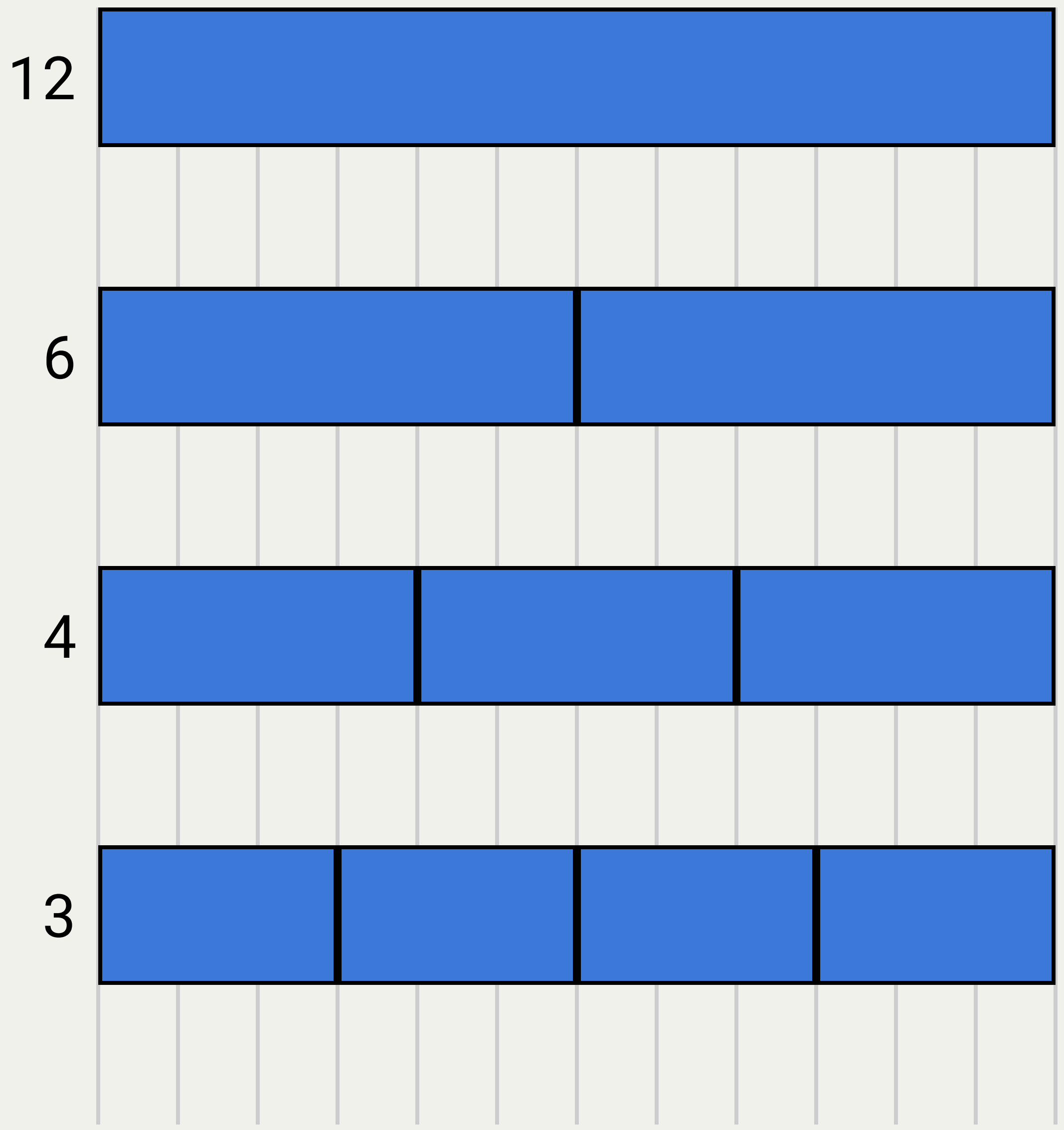
Grid System

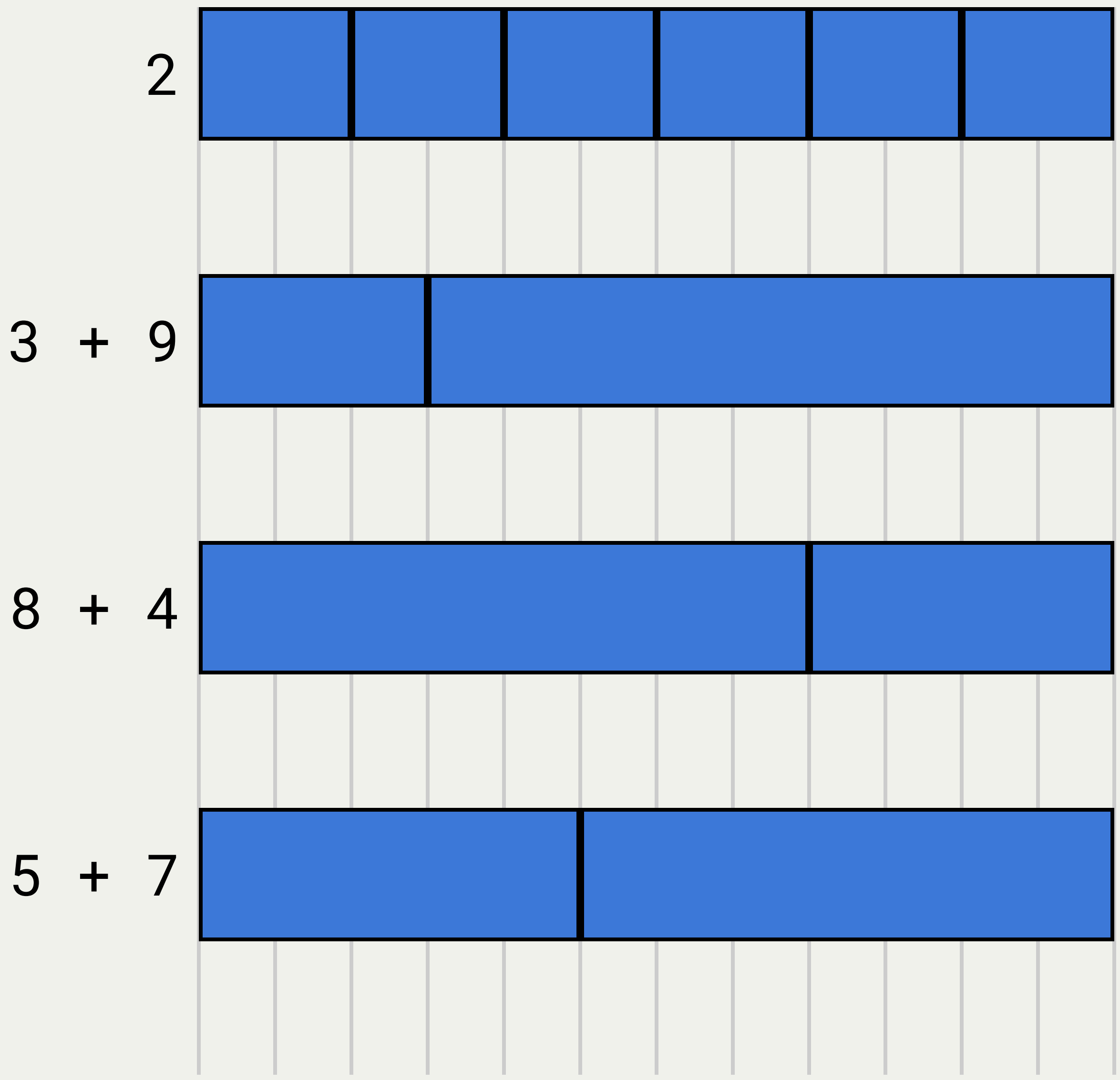
Grid System

- Libraries and frameworks both make it easier to build sites
- One of the main ways they do this, is by providing a grid system
- Lots of these grid systems are built around a 12 column grid:
  - Why 12? Lots of combinations!
- We add classes that define how wide elements should be



3.6


12
6
4
3


2
3 + 9
8 + 4
5 + 7

Materialize

Using it

- Download or reference the CSS and JS files
    - Make sure to include jQuery before their JS, and your JS comes after Materialize too

Let's muck around with it

- Grid
- Buttons
- Forms
- Icons
- Do something with their JS:
  - e.g. Modals

Advanced CSS

GENERAL ASSEMBLY

Vendor Prefixing

- For non-standard, experimental features
- Based on new versions of CSS
- For browser compatibility:
  - Use the vendor prefixes ***first***
  - Then use the un-prefixed version
  - *Can I use can help*

Vendor Prefixing

```
div {
  -webkit-transition: all 4s;
     -moz-transition: all 4s;
      -ms-transition: all 4s;
       -o-transition: all 4s;
          transition: all 4s;
}
```

Vendor Prefixing

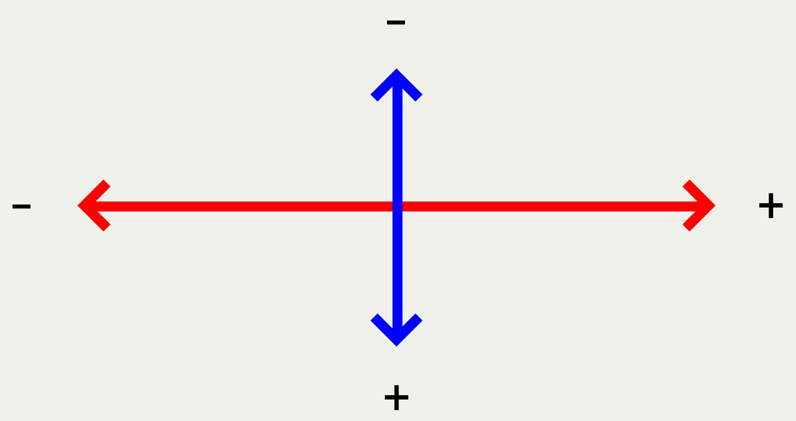
- How to deal with Vendor Prefixes
- Emmet and Prefixing
- Check what needs to be prefixed
- Autoprefixer

CSS3 added a lot...

Box Shadow

```
box-shadow: OFFSET_X OFFSET_Y BLUR_RADIUS SPREAD_RADIUS COLOUR
```

```
div {
    box-shadow: 5px 5px 10px 20px black;
}
```

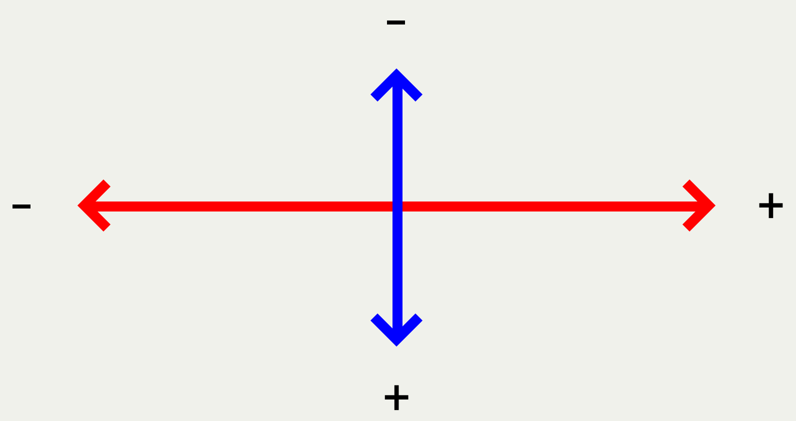


The X axis
The Y axis

Text Shadow

```
text-shadow: OFFSET_X OFFSET_Y BLUR_RADIUS COLOUR
```

```
h1 {
    text-shadow: 5px 5px 10px black;
}
```



The X axis
The Y axis

Transitions

- The CSS property transition is a shorthand property for a bunch of other things
  - transition-property
  - transition-duration
  - transition-timing-function
    - Examples of these
  - transition-delay

Transitions

```css
div {
  transition: all 0.5s;

  transition: width 0.2s, background 0.3s;

  transition: margin-left 4s linear 1s;
}
```

Animations

Process

- Define your animation
- Apply it to elements

Animations: Defining

```
@keyframes my-animation {
    0% {
        opacity: 0;
    }

    100% {
        opacity: 1;
    }
}
```

Animations: Defining

```
@keyframes my-animation {
    0% {
        opacity: 0;
    }

    50% {
        opacity: 0.8;
    }

    100% {
        opacity: 1;
    }
}
```

Animations: Defining

```css
@keyframes my-animation {
    0% {
        opacity: 0;
    }
    30% {
        opacity: 0.8;
    }
    70% {
        opacity: 0.2;
    }
    100% {
        opacity: 1;
    }
}
```

Animations

- Animation is a shorthand property for:
  - animation-name
  - animation-duration
  - animation-timing-function
  - animation-delay
  - animation-iteration-count
  - animation-direction
  - animation-fill-mode
  - animation-play-state

Animations: Applying

```
div {
    /* DURATION | NAME */

    animation: 1s fade-out;

    /* DURATION | TIMING_FUNCTION | DELAY | NAME */

    animation: 1s linear 0.5s fade-out;

    /* DURATION | TIMING_FUNCTION | DELAY |
       ITERATION_COUNT | DIRECTION | FILL_MODE |
       PLAY_STATE | NAME */

    animation: 3s ease-in 1s 2 reverse both paused fade-out;
}
```

Animations: Applying

```
@keyframes fade-out {
    0% {
        opacity: 0;
    }
    100% {
        opacity: 1;
    }
}

div {
    animation: 3s linear 1s just-keep-spinning;
}
```

Transform Origin

- The transform-origin property is used to change the position of the origin of transformation of an element.
    - e.g. Rotating an element about the transform origin will result in different rotation results depending on the position of the origin.

Transform Origin

- Sarah Drasner's explanation

Animations

There are some tools that can help you create these animations:

- Bounce.js
- Animate.css

Homework

- Your project!

Q & A

GENERAL ASSEMBLY

Feedback Time

Lesson 18: **Framework and Advanced CSS**

https://ga.co/fewd32syd

Thanks!

GENERAL
ASSEMBLY