Cubstart Web Lecture 2

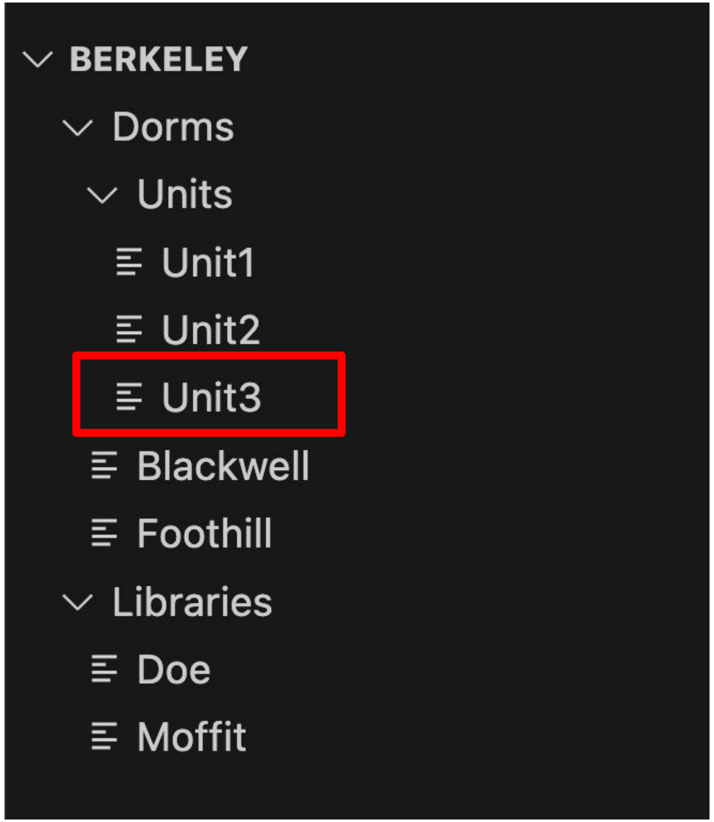
[start recording]

Administrivia

- HW1 is due Sept 22 (this Friday) at midnight!
  - Lab @ Physics 3 from 4-6PM
    - Attendance is mandatory
    - At the end of lab, we will be doing a HW 1 walkthrough
- HW2 will be posted sometime today or tomorrow, due the following Friday

CSS
(code with us!)

Defining Relative Paths

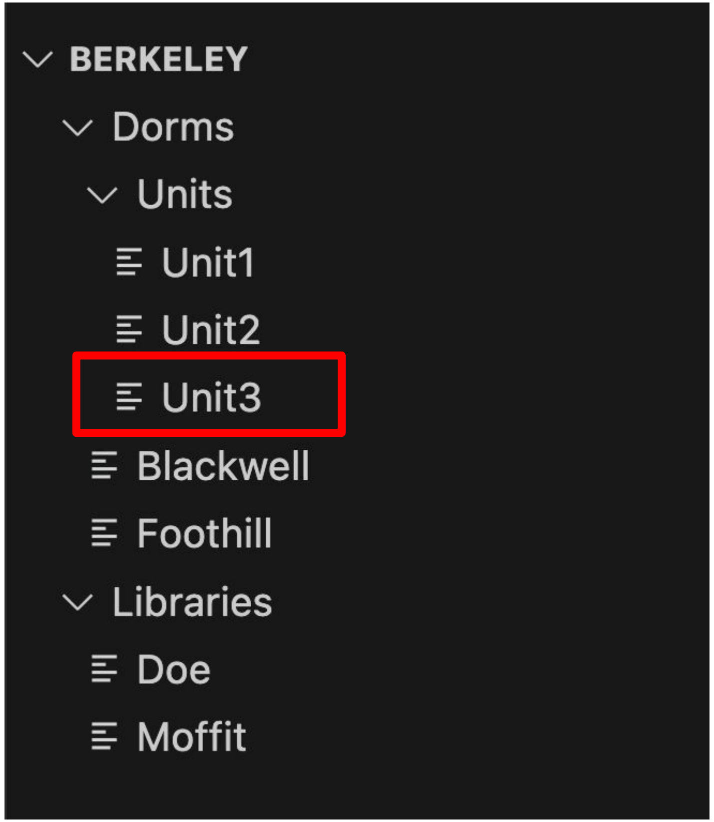


./ -> finds **current** directory (**/Berkeley/Dorms/Units**)

../ -> finds **parent** directory (**/Berkeley/Dorms**)

To link to:	href:
Unit 1	"./Unit1"
Foothill	"../Foothill"
Doe	"../../Libraries/Doe"

Defining Relative Paths


BERKELEY
Dorms
Units
Unit1
Unit2
Unit3
Blackwell
Foothill
Libraries
Doe
Moffit

Current directory: **/Berkeley/Dorms/Units**

To get to:	href:
Unit 1	“Unit1”

Looks for a directory/file named **Unit1** in the **current** directory

Linking to a Stylesheet:

```
<!DOCTYPE html>
<html>
     <head>
          <title>NAME OF WEBSITE</title>
          <link rel="shortcut icon" type="img/png" href="favicon.png">
          <link rel="stylesheet" href="./style.css">
          <meta name="viewport" content="width=device-width, initial-scale=1.0">
     </head>
     <body>
          <h1> This is a header. </h1>
     </body>
</html>
```

Selecting by Tag

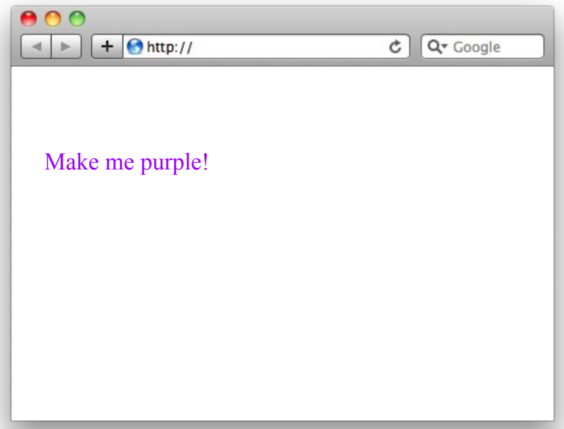
To select or “find” the HTML elements you want to style you can call directly using HTML tags (i.e. p, h1, h2, etc.)

HTML file:

```
<p>
    Make me purple!
</p>
```

CSS file:

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



Make me purple!

Selecting by Class

What if we only want one of an element to have a style?

- Add a "**`class`**" attribute
- `class="name"` → In CSS, use "**`.name`**" to style

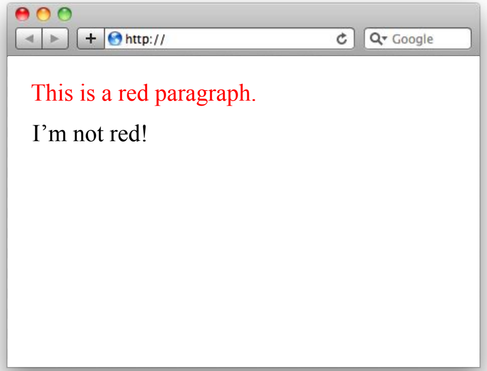
HTML file:

```
<p class="scary"> This is a red paragraph. </p>

<p> I'm not red! </p>
```

CSS file:

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



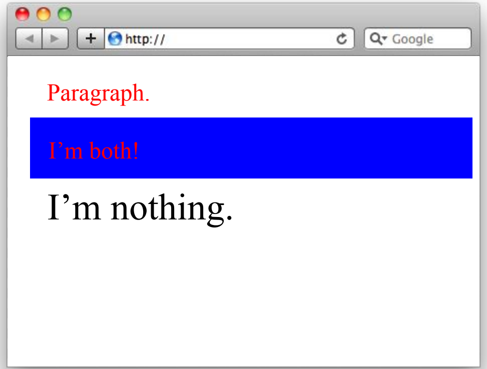
This is a red paragraph.

I'm not red!

Selecting by ID

Add an “**id**” attribute

- `id=“name”` → In CSS, use “**#name**” to select
- They are unique and are only used once


Paragraph.

I’m both!

I’m nothing.

CSS file:

```
#ocean {
    background-color: blue;
}
.scary {
    color: red;
}
```

HTML file:

```
<p class=“scary”> Paragraph. </p>
<p id=“ocean” class=“scary”> I’m both! </p>
<h1> I’m nothing. </h1>
```

Check-In Question 1

Using words, describe what this declaration “says”:

```
h2 {
    height: 50px;
    font-weight: bold;
    color: blue;
}
```

Answer: All <h2> elements will have a height of 50px, be bolded, and blue

Check-In Question 2

What happens here:

1. This is a paragraph!
2. This is a paragraph!
3. This is a paragraph!
4. This is a paragraph!

HTML file:

```
<p class="paragraph">
    This is a paragraph!
</p>
```

CSS file:

```
#paragraph {
    background-color: green;
}
```

Answer: 4 (you select classes with a period, not a hashtag!)

Selecting by Pseudo Class

Let’s get fancy…

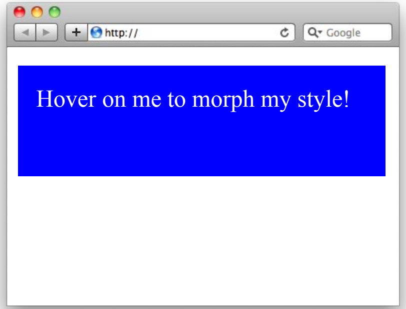
- Change style on hover?
- Attach the “:hover” pseudo-class

HTML file:

```
<p id=“back”>
    Hover on me to morph my style!
</p>
```

CSS file:

```
#back {
    background-color: blue;
    color: white;
}
#back:hover {
    background-color: green;
}
```


Hover on me to morph my style!

Selecting by Pseudo Class

Let’s get fancy...

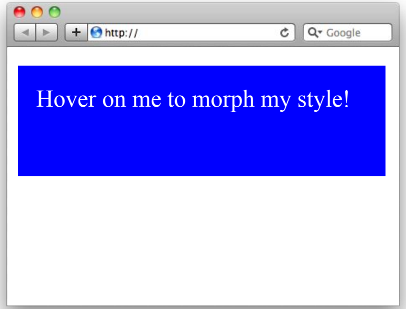
- Change style on hover?
- Attach the “:hover” pseudo-class

HTML file:

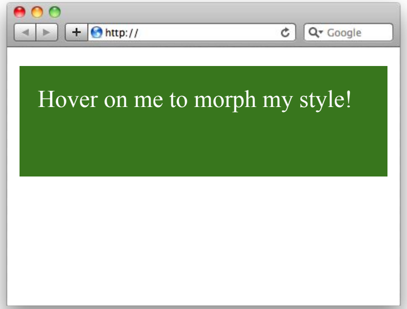
```
<p id="back">
    Hover on me to morph my style!
</p>
```

CSS file:

```
#back {
    background-color: blue;
    color: white;
}
#back:hover {
    background-color: green;
}
```



Hover on me to morph my style!



Hover on me to morph my style!

Conflicting Rules

HTML file:

```
<p id="special">
    This is a paragraph!
</p>
```

CSS file:

```
#special {
    color: blue;
}


p {
    color: gray;
}
```

An element may have conflicting style rules! How does the browser determine what rule to apply?

1. **Specificity**
2. **Source order**

* The specifics are a lot more complicated, but this the gist

Specificity

HTML file:

```
<p id="special">
    This is a paragraph!
</p>
```

CSS file:

```
#special {
    color: blue;
}


p {
    color: gray;
}
```

The more specific selector receives higher priority for its rules.

Specificity from highest to lowest:

1. Inline
2. ID selectors
3. Class selectors
4. Tag/element selectors

* The specifics are a lot more complicated, but this the gist

Source Order

```
<p class=”big small”>
    This is a paragraph!
</p>
```

```
.big {
    font-size: 24px;
}

.small {
    font-size: 12px;
}
```

If specificities are equal, the last rule in the CSS source code is applied.

* The specifics are a lot more complicated, but this the gist

Inheritance

```
<body>
    <p>
        This is a paragraph!
    </p>
</body>
```

```
body {
    color: blue;
}
```

Some styling rules are **inherited** from the parent element.

Inheritance

```
<body>
    <p>
        This is a paragraph!
    </p>
</body>
```

```
body {
    color: blue;
}
```

Some styling rules are **inherited** from the parent element.

Makes sense to inherit “text color” from the parent.

Inheritance

```
<body>
    <p>
        This is a paragraph!
    </p>
</body>
```

```
body {
    border: 1px solid black;
}
```

Other styling rules are **not inherited** from the parent element.

Doesn't make sense to inherit "border" from the parent, otherwise all children of body will have a border!

Check-In Question 3

What happens here:

A) This is a paragraph!

B) This is a paragraph!

C) This is a paragraph!

D) This is a paragraph!

Answer: B

```
<p class="paragraph" id="intro">
    This is a paragraph!
</p>
```

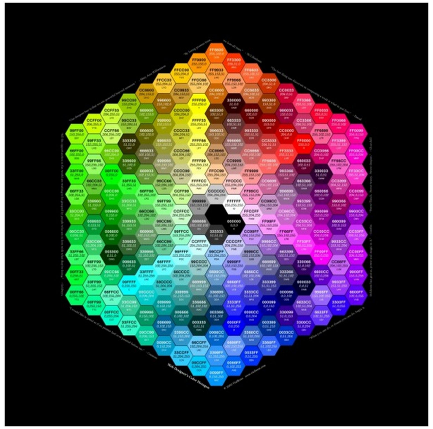
```
#intro {
    background-color: blue;
}
.paragraph {
    text-decoration: underline;
    background-color: green;
}
```

Basic Properties

Color Values

```
p { color: value; }
```

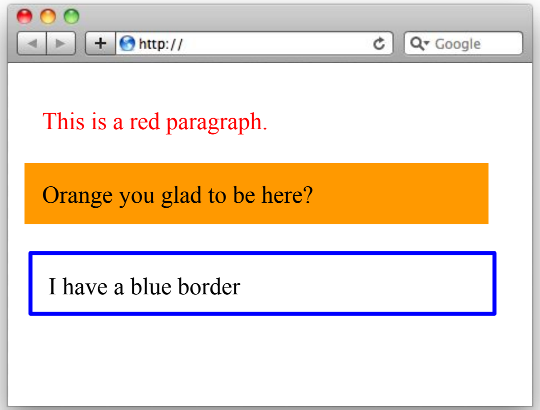
- CSS has default colors by name (i.e. `red`, `blue`, `green`, etc.)
  - Only certain colors work; Doesn’t give you access to all possibilities.
- RGB = Three numbers between 0 and 255
  - All colors are combinations of red, green, blue
  - Specifies amount of each base color
  - `rgb(255,0,0)`, `rgb(210,105,30,.5)`
- Hex = 6 digit combo of letters and numbers with “#”
  - Essentially a shorter representation of RGB
  - `#ff0000`, `#0effe3`, `#000000`



Color Properties

- Changing text color using:
  - `color`
- Changing background color of element using:
  - `background-color`

```
p {
    color: red;
    background-color: #FFA500;
    border-style: solid;
    border-color: blue;
}
```



This is a red paragraph.

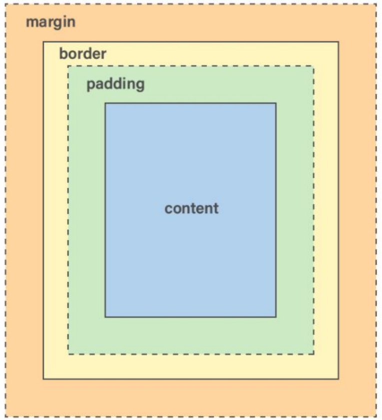
Orange you glad to be here?

I have a blue border

Size Properties

- Dimensions
  - height
  - width
  - max-height, etc.
- Spacing
  - margin
  - padding
  - border
- Positioning
  - top, left, etc.

- Units!
  - px: 1 pixel, dependent on resolution of user's screen
  - vw/vh: 1% of window's width/height
  - em: 1x font-size of element
  - rem: 1x font-size of root element



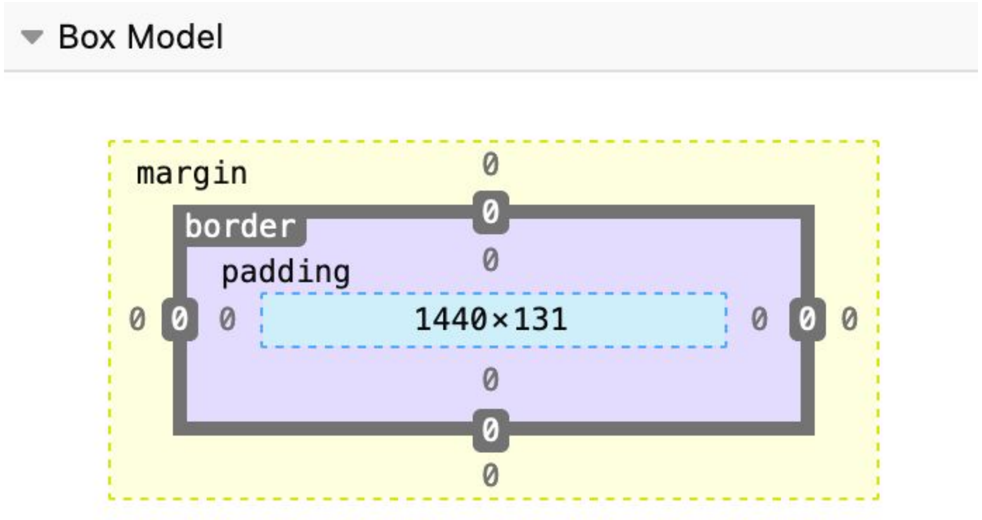
Box Model

This is how CSS thinks about spacing

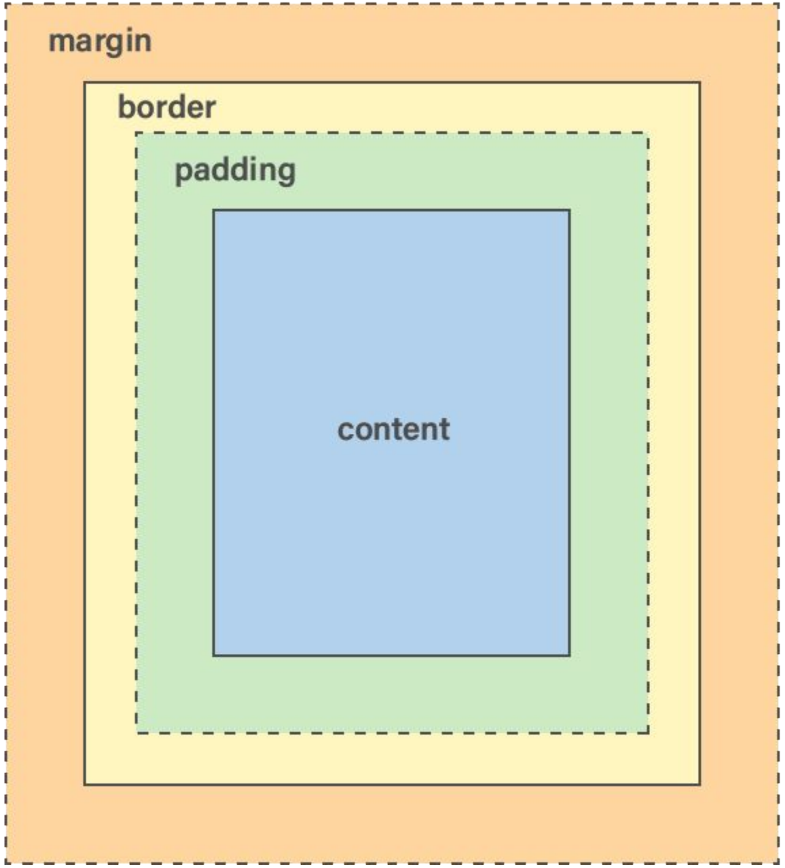
- `margin`
- `border`
- `padding`

You can view this yourself:

- Go to calhacks.io
- Right-click and select “Inspect Element”
- It should appear on the bottom right of your screen



box model:



content
this area contains the "content" of the element, such as text and images

padding
space around the content area and within the border box

border
surrounds the content and any padding

margin
outermost layer that controls the spacing between other elements

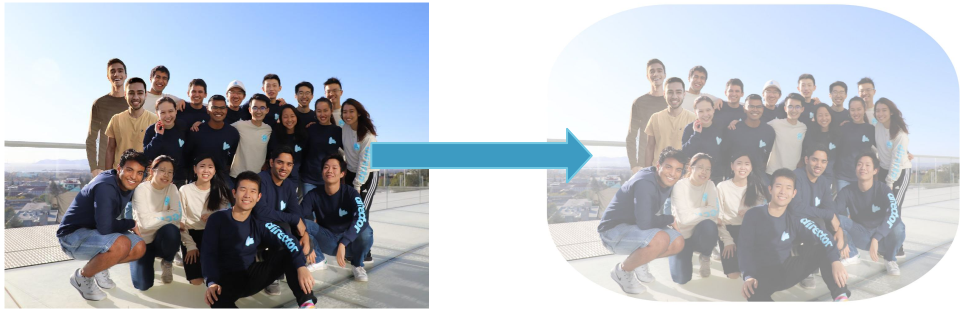
Font Properties

- `font-size`
  - Size of text
- `font-weight:`
  - regular, medium, bold, extra bold, etc.
- `font-family`
  - Specifies a font to use, and a default backup if unavailable

```
p {
    font-size: 20px;
    font-family: “Roboto”, sans-serif;
}
```

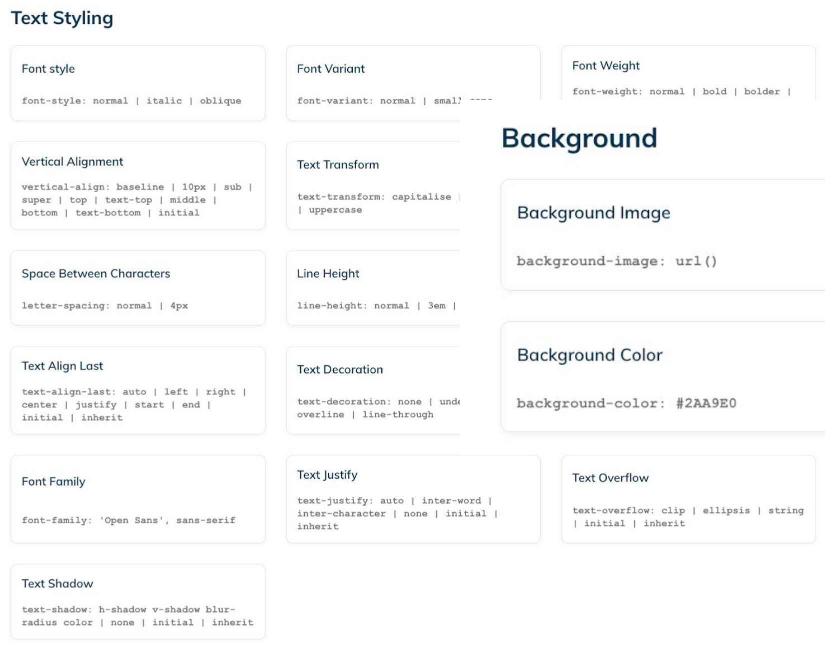
Image Properties

- **`border-radius`**
  - Rounds the edges of the picture
- **`border`**
  - Gives a “border” around the image
- **`opacity`**
  - Changes the visibility of the image
  - Scale from 0 - 1



```css
img {
     border-radius: 30%;
     opacity: 0.5;
}
```

So Much More...



Text Styling

Font style

`font-style: normal | italic | oblique`

Font Variant

`font-variant: normal | small`

Font Weight

`font-weight: normal | bold | bolder |`

Vertical Alignment

```
vertical-align: baseline | 10px | sub |
super | top | text-top | middle |
bottom | text-bottom | initial
```

Text Transform

```
text-transform: capitalise |
| uppercase
```

Space Between Characters

`letter-spacing: normal | 4px`

Line Height

`line-height: normal | 3em |`

Text Align Last

```
text-align-last: auto | left | right |
center | justify | start | end |
initial | inherit
```

Text Decoration

```
text-decoration: none | unde
overline | line-through
```

Font Family

`font-family: 'Open Sans', sans-serif`

Text Justify

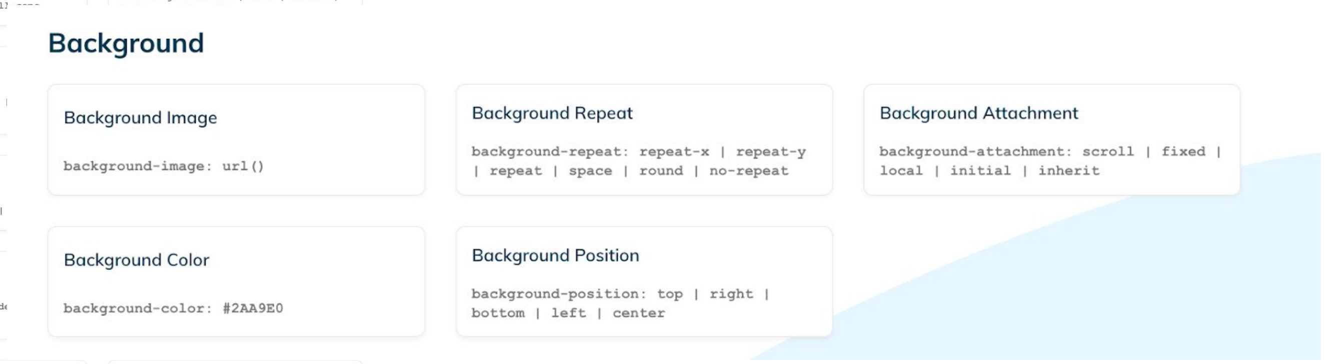
```
text-justify: auto | inter-word |
inter-character | none | initial |
inherit
```

Text Overflow

```
text-overflow: clip | ellipsis | string
| initial | inherit
```

Text Shadow

```
text-shadow: h-shadow v-shadow blur-
radius color | none | initial | inherit
```



Background

Background Image

`background-image: url()`

Background Repeat

```
background-repeat: repeat-x | repeat-y
| repeat | space | round | no-repeat
```

Background Attachment

```
background-attachment: scroll | fixed |
local | initial | inherit
```

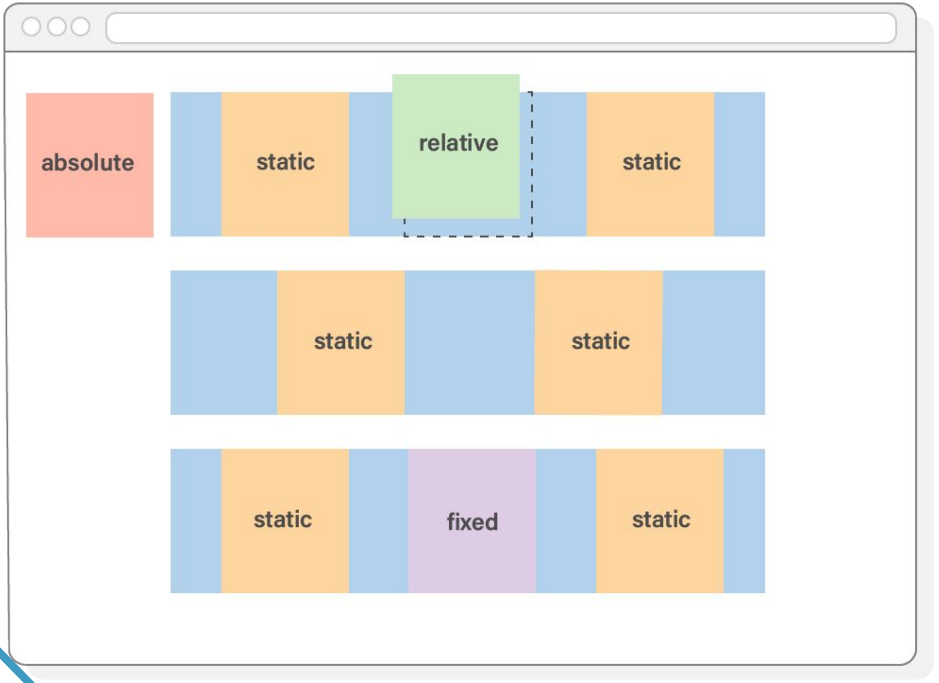
Background Color

`background-color: #2AA9E0`

Background Position

```
background-position: top | right |
bottom | left | center
```

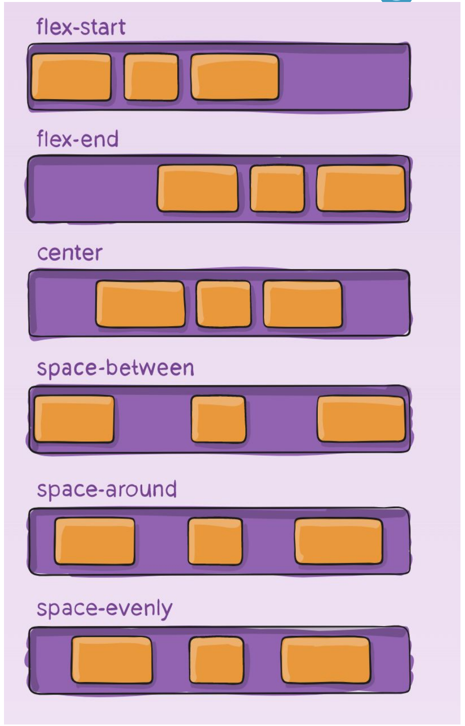
Positioning



- **`static`**: default original position
- **`relative`**: relative to initial position
- **`absolute`:** relative to nearest positioned ancestor
- **`fixed`:** relative to “viewport” (window)

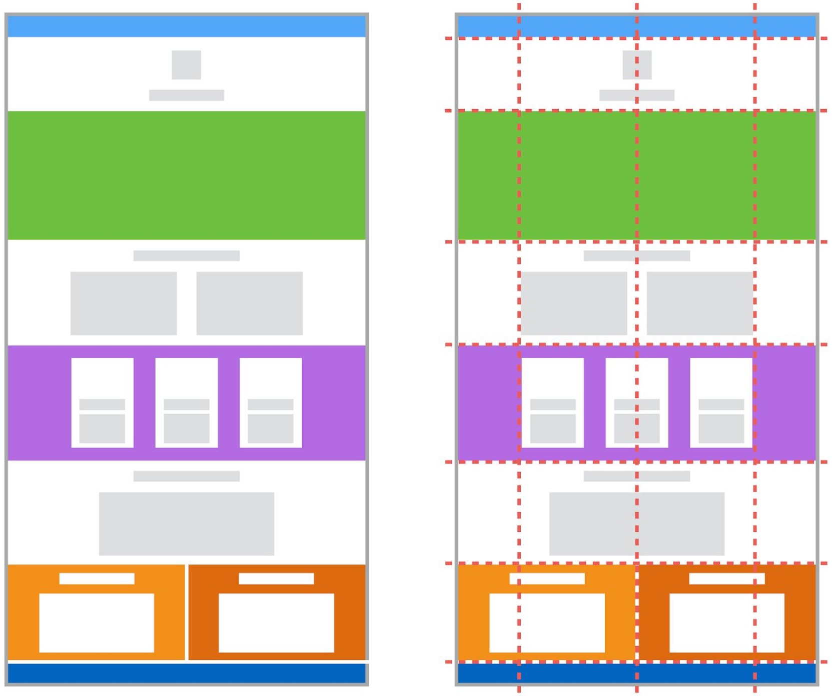
Flexbox

- CSS Property `display: flex;`
  - Defines a flex container
  - Enables a flex context for all its direct children
  - By default, flex items are laid out in the source order
- `flex-direction: column;`
  - Default direction is a row (left to right)
- `justify-content: *;`
  - See the picture!
- For more: https://flexboxfroggy.com/



Grid

- The grid system splits the layout into rows and columns (2D)
  - `display: grid;`
- Adjust the sizes of rows and columns with
  - **`grid-template-columns`**
  - **`grid-template-rows`**
- For more: https://css-tricks.com/snippets/css/complete-guide-grid/



Morten Rand-Hendriksen

Check-In Question

If I wanted offset an image from its default location, what position would I give it? *

A) Relative
B) Absolute
C) Fixed
D) Static

* this can be useful in circumstances where you want an image to cross over multiple sections

Media Queries

@media

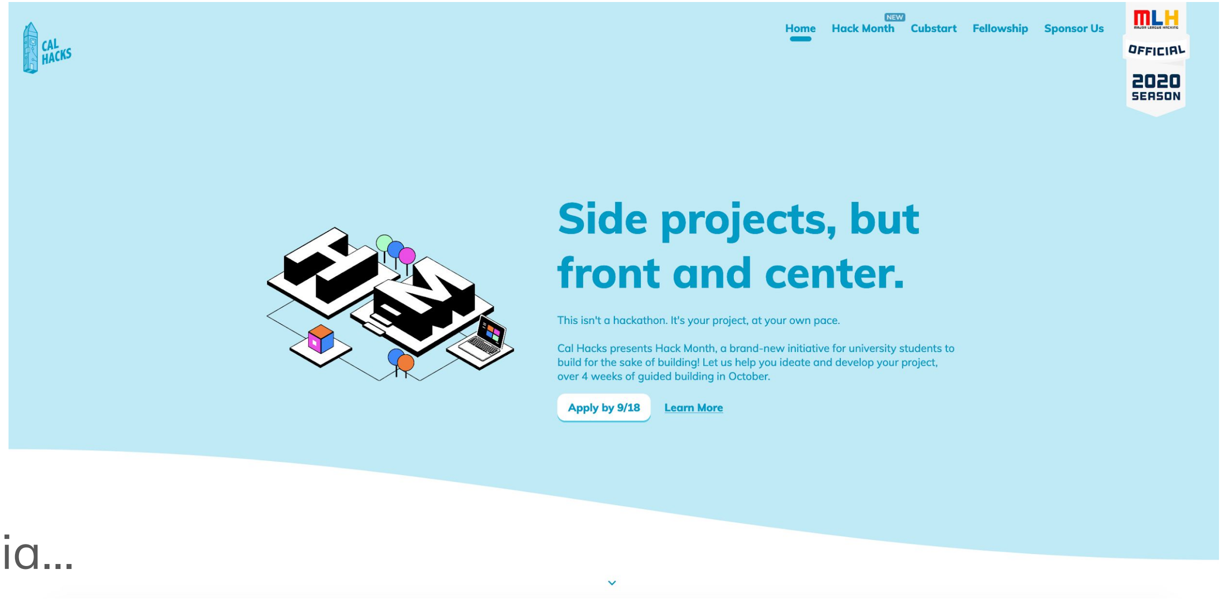
Media queries are extremely important to keep in mind!

What looks good on your screen might not be good on another’s.

```
@media only screen and (max-width: 600px) {
    body {
        margin: 1rem;
    }
}
```

@media Example

This is what our webpage on a laptop screen size looks like, but for a phone this might not be ideal.



Side projects, but front and center.

This isn't a hackathon. It's your project, at your own pace.

Cal Hacks presents Hack Month, a brand-new initiative for university students to build for the sake of building! Let us help you ideate and develop your project, over 4 weeks of guided building in October.

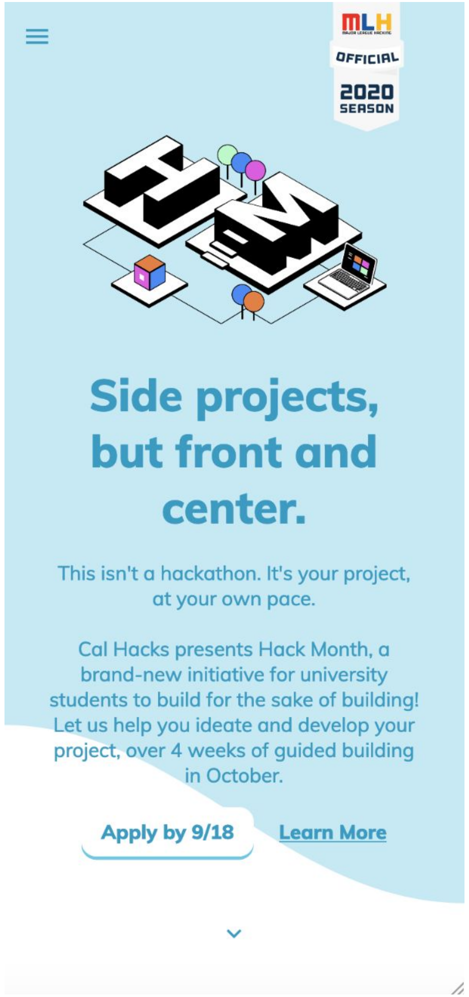
With @media...

@media Example

```
@media only screen and (max-width: 350px) {
  h5 {
    font-size: 12px;
  }
}

p {
  font-weight: normal;
}

a {
  color: #3d9bc2;
  text-decoration: none;
}
@media only screen and (max-width: 350px) {
  a {
    font-size: 12px;
  }
}
```



Side projects, but front and center.

This isn't a hackathon. It's your project, at your own pace.

Cal Hacks presents Hack Month, a brand-new initiative for university students to build for the sake of building! Let us help you ideate and develop your project, over 4 weeks of guided building in October.

You can adjust the page for **another** device, such as a phone (:

More on @media

Some devices you might want to consider:

- Laptops
  - Retina / Non-Retina Screen
  - Use general size ranges rather than targeting a specific device
- For Phones and Tablets, you should try to target designs for many standard and popular devices
  - Tablets
    - iPad / Galaxy / Nexus / Kindle Fire
  - Phones and Handhelds
    - iPhones / Galaxy / Google Pixel / HTC / Windows
- Test with different devices! Chrome has a tool for this :)

Attendance

https://forms.gle/LAAZ28LAEzEcpfP59

Secret word: