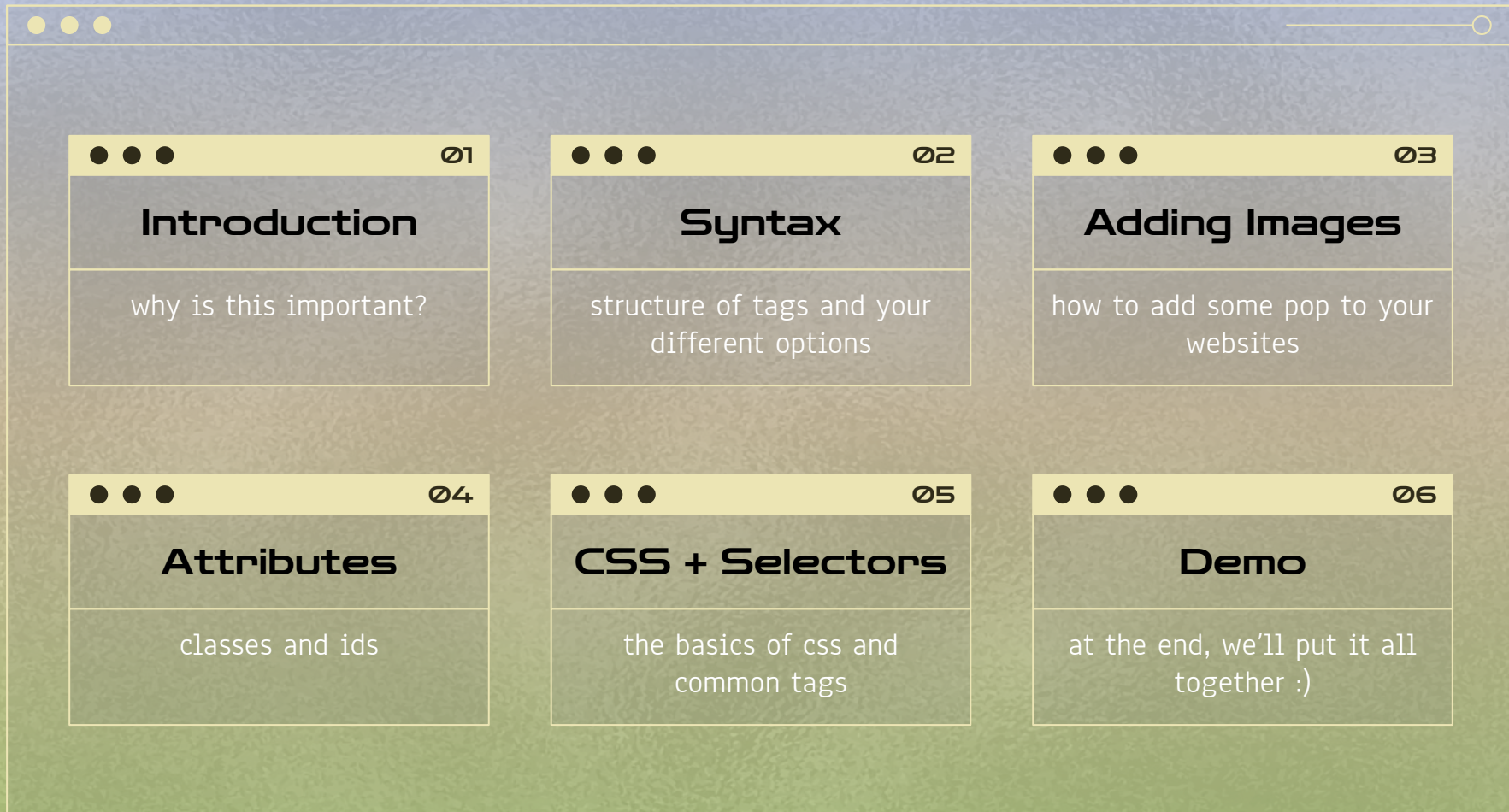


Basics of HTML & CSS



with ***gracehacks***



What's the hype about HTML/CSS?

- **HTML** is a *markup language* that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way.
- **CSS (Cascading Style Sheets)** is used to style and lay out web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.
- We use HTML and CSS to make static websites. These two along with JavaScript (which is a programming language we won't have time to cover) are the foundation of most websites you've ever been on.
- And those three serve as the foundation to all popular front-end frameworks out there like React and Angular! *Once you know this, you're on your way to your first front-end job!*

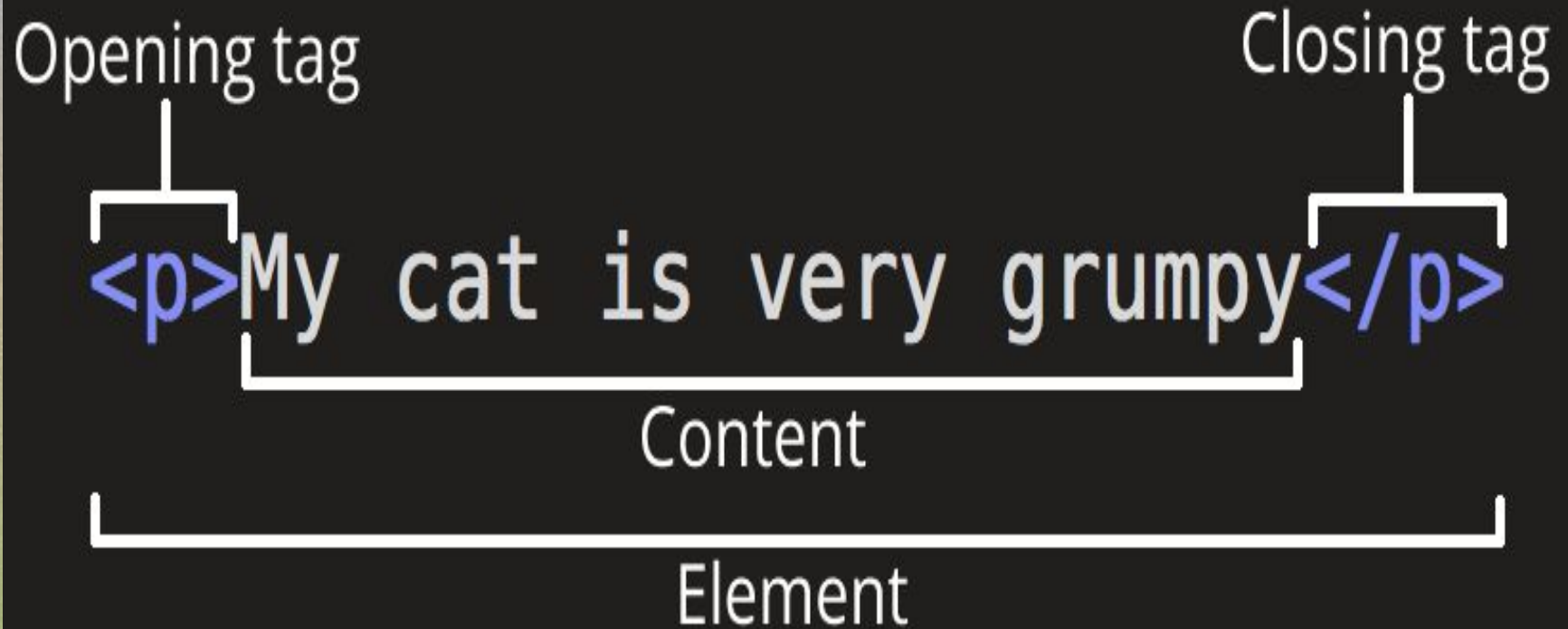


The syntax of HTML

- HTML consists of a series of elements, which you use to enclose, wrap, or mark up different parts of content to make it appear or act in a certain way.
- *It can be as complicated as you want to be.*

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4      </head>
5      <body>
6          <h1>My First Page</h1>
7          <p>This is my first page.</p>
8          <h2>A secondary header.</h2>
9          <p>Some more text.</p>
10     </body>
11 </html>
```

The anatomy of a elements



Common HTML tags

- `<h1>`, `<h2>`, ..., `<h6>` declare headings, sized from largest to smallest.
- `<p>` declares a paragraph.
- `` and `` declares unordered lists and ordered lists, respectively.
- `` declares a single list item and should be declared *inside* a list (either ordered or unordered).
- The `<div>` tag is the most generic element tag used to declare the "parts" of a webpage.
- The ``, `<i>`, `<u>`, and `<s>` tags **bold**, *italicize*, underline, or ~~strikeout~~ text within a tag's content area, respectively.
- The `<a>` tag lets you add hyperlinks. When rendered, all surrounded content will appear light blue and underlined. We can also use the attribute `href` to define a link's URL.

Adding images

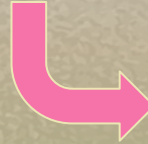
The tag has two required attributes:

- src - Specifies the path to the image
- alt - Specifies an alternate text for the image, if the image for some reason cannot be displayed

Examples (embedding a URL from online)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Presentation Image Demo</title>
  </head>
  <body>
    
  </body>
</html>
```

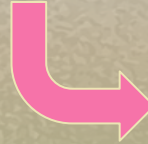
Oftentimes you can get the URL by going to Google Images and right clicking on the one you want, and then clicking “Copy Image Address”



Examples (embedding a URL from online)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Presentation Image Demo</title>
  </head>
  <body>
    
  </body>
</html>
```

Oftentimes you can get the URL by going to Google Images and right clicking on the one you want, and then clicking "Copy Image Address"



Examples (using a local file)

```

```

Supported image types
for :

APNG, AVIF, GIF, JPEG,
PNG, SVG, WebP



HTML attributes

- Some tags already require attributes like and <a> tags like we discussed before.
- However, there's some optional ones that you can add to any tags (i.e. divs, h1, p, etc.)
 - **class** (className in React)
 - **id**
- There's other ones out there, but *these two are the most useful*.

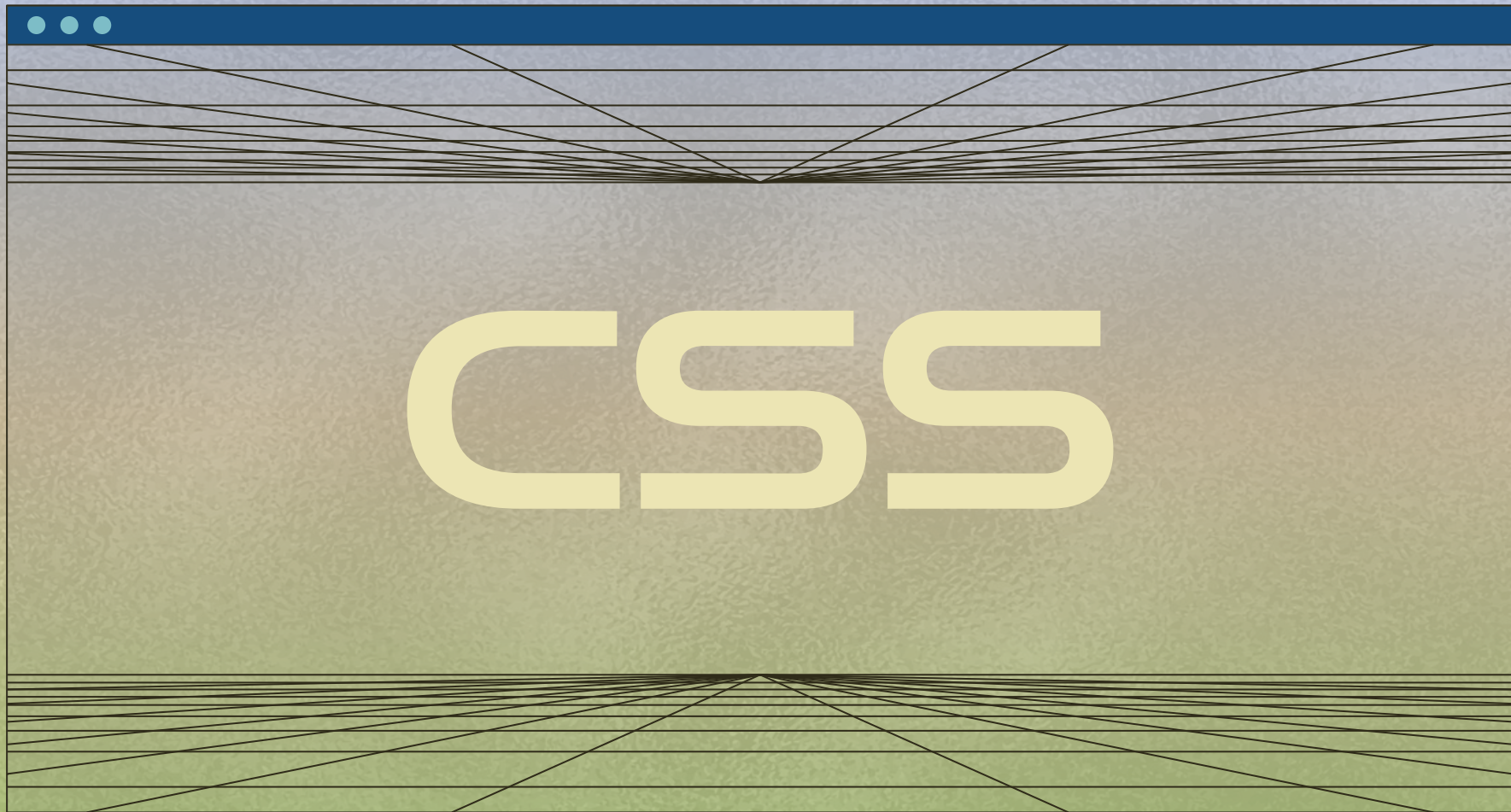
Difference between ids and classes

- Classes allow CSS and JavaScript to select and access specific elements via the class selectors or methods, but they're reusable!
- The id global attribute defines an identifier (ID) which must be unique in the whole document so we can identify the element when linking, scripting, or styling.

```
<!DOCTYPE html>
<head>
  <title>Difference between classes and ids</title>
</head>
<body>
  <div class="reusable">I can use the reusable tag as much as I want</div>
  <div class="reusable">... without it being an issue!</div>
  <div class="reusable">This will be helpful when we start using styling sheets.</div>

  <div id="unique">However ids must be unique.</div>

  <div id="unique2" class="reusable">You could possibly have an id and class for one element though! *gasp*</div>
</body>
</html>
```



Cascading Styling Sheets

- We just learning how to use HTML to create the parts of our webpage, but now we have to learn how to style these parts.
- This will require us to use classes and ids like we just discussed.
- Popular alternatives to plain CSS:
 - SCSS (Sassy Cascading Styling Sheets)
 - LESS (Leaner Styling Sheets)
 - SASS (Syntactically Awesome Style Sheets)
- Same ideas apply, but some of them include nice features like nesting

Targeting elements with selectors

- **Styling with CSS Declarations:** When writing style sheets, we use declarations, which consists of a CSS *property* and a *property value*, separated by a colon like so – “property: value”
 - Each CSS property contains a set of property values that it can take.
- **Targeting Elements with CSS Selectors:** Of course, CSS declarations are only useful if we can specify what elements they should be applied to. To specify the scope of a set of declarations, we can use CSS selectors. A selector binds a set of declarations to a selector expression which defines the elements that the declared styles should apply to.

Example of declarations + selectors

In the example to the right, **div** is a selector expression and its selector encapsulates the CSS declaration **color: red;** within curly braces. This targets all **div** elements and changes their text color to red.

Result:

The Declaration of Independence

We hold these truths to be self-evident, that all men are created equal, that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness.

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS example</title>
    <!-- have your stylesheet in the same folder as your same HTML file! -->
    <link href="basic.css" rel="stylesheet"/>
  </head>
  <body>
    <h1>The Declaration of Independence</h1>
    <div>
      We hold these truths to be self-evident, that all men are created equal,
      that they are endowed by their Creator with certain unalienable Rights,
      that among these are Life, Liberty and the pursuit of Happiness.
    </div>
  </body>
</html>
```

Users > audreyostrom > exampleHTML > # basic.css > div

```
1  div {
2    color: red;
3  }
```

How to target classes and ids

In the example to the right, **div** is a selector expression and its selector encapsulates the CSS declaration **color: red;** within curly braces. This targets all **div** elements and changes their text color to red.

Result:

The Declaration of Independence

We hold these truths to be self-evident, that all men are created equal that they are endowed by their Creator with certain unalienable Rights, that among these are Life, Liberty and the pursuit of Happiness.

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS example</title>
    <!-- have your stylesheet in the same folder as your same HTML file! -->
    <link href="basic.css" rel="stylesheet"/>

    <!-- you can also import fonts! -->
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Nunito">
  </head>
  <body>
    <div id="header-text">The Declaration of Independence</div>
    <div class="body-text">
      We hold these truths to be self-evident, that all men are created equal<br>
      that they are endowed by their Creator with certain unalienable Rights,<br>
      that among these are Life, Liberty and the pursuit of Happiness.
    </div>
  </body>
</html>
```

```
1  html {
2    background-color: black;
3  }
4
5  body {
6    font-family: 'Nunito';
7    color: white;
8  }
9  #header-text {
10   color: pink;
11   font-size: 30px;
12 }
13
14 .body-text {
15   font-style: italic;
16 }
```


Popular Properties

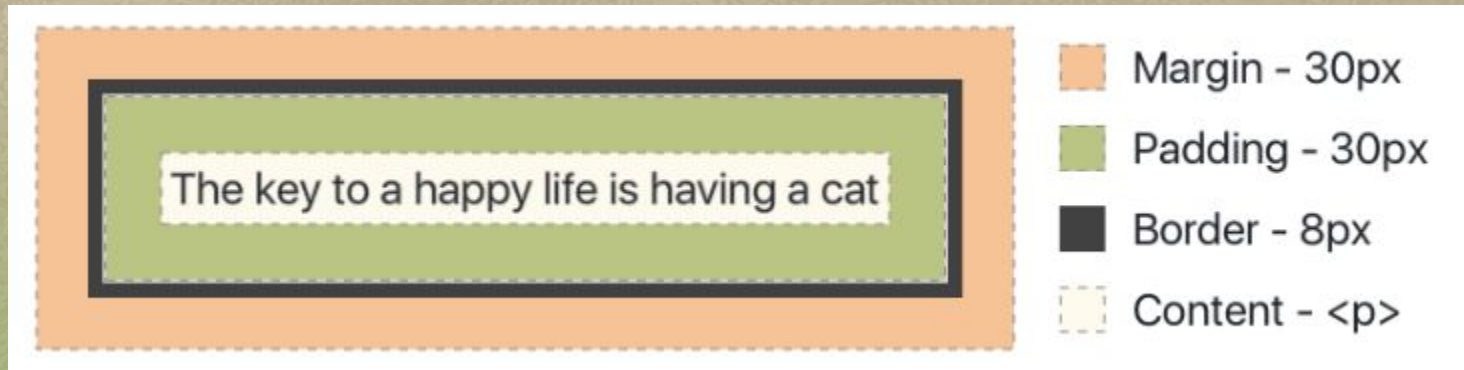
- **color**: allows us to specify the text color of an HTML element.
- **font-size**: allows us to change the size of the font for any text based content. This includes <h1>-<h6>, <p>, <a> and more.
- **background-color**: allows you to set the background color of an HTML element. Just like text color, the value of this property can be a hex value, rgba or named color.
- **height and width**: specify the height and width of a HTML element respectively.
- **border**: allows you set a border for your HTML element.
- **border-radius**: allows you to add rounded corners of a HTML elements.

Modifying the layout

- Let's say we wanted to particular positioning for our elements.
- *Follow the box model:* When a browser displays your HTML elements, each element is displayed as a rectangular box. CSS is then applied to the box to style the HTML element.
- Each box is comprised of:
 - **Content:** The actual content (text, images) of the HTML element.
 - **Padding:** Defined by the padding property. In-between the border and content of an element. Is transparent.
 - **Border:** Surrounds the padding and content of an element.
 - **Margin:** Adds space between the border and other HTML elements. Is transparent.

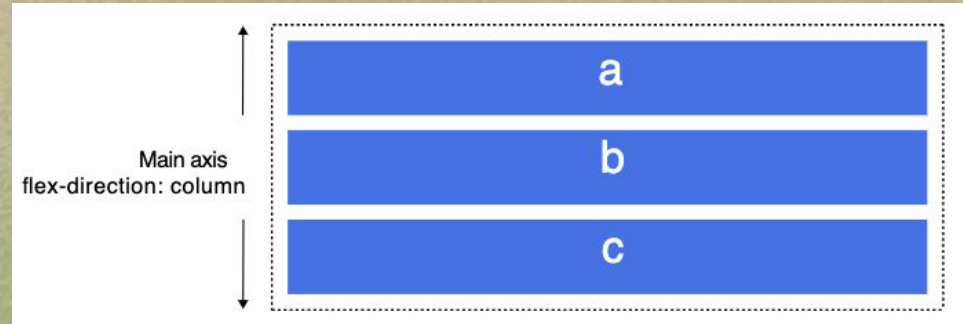
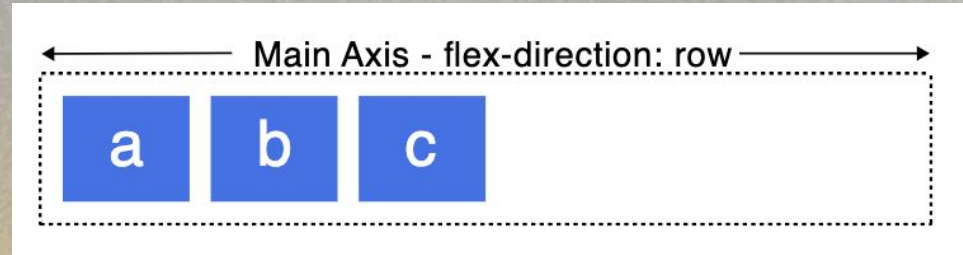
Modifying the layout (cont.)

- We can modify our element's box with the following properties:
 - padding
 - margin
 - border



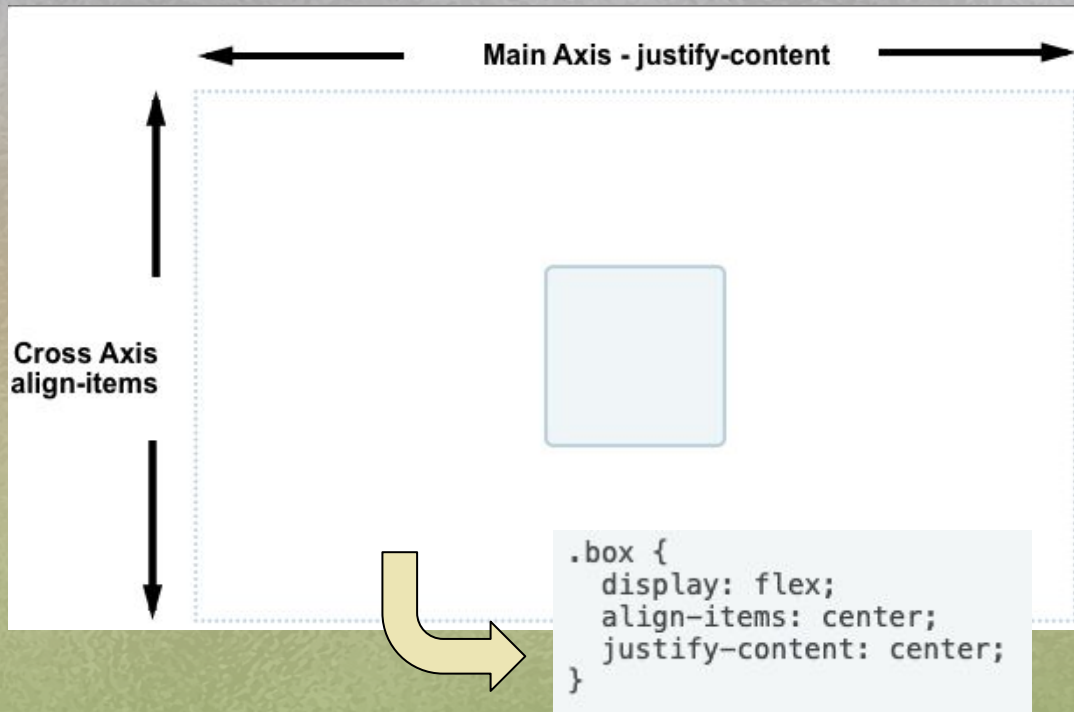
Flexbox styling

- One of the best things to use to simplify your stylesheets is to use flexbox styling, which we specify with “display: flex”.
- Mostly used for rows and columns. Very helpful for centering items
- Properties you can use in conjunction with “display: flex”
 - flex-direction
 - justify-content, justify-self, justify-items
 - align-content, align-self, align-items



Styling with flex rows

- If I have a flex row:
 - I center things horizontally with justify
 - I center things vertically with align-items
- Flip-flop when you have a flex column.





Demo Time!

Requirements to follow along

- For the basic HTML + CSS demo:
 - A computer, text editor, and a browser installed
 - That's literally it :)
- If we have time for the React + JS demo:
 - Have Node.js installed

Other Helpful Resources

CSS Reference

Documentation on popular CSS properties by Mozilla

HTML Reference

Documentation on basic HTML features by Mozilla

Learn JavaScript

The next step in your front-end career!

Learn TypeScript

Popular alternative to plain JS

Learn React

Documentation on the most popular framework right now

Learn Angular

Documentation on another popular framework

Interested in GraceHacks?

Social Media

Instagram: gracehacks

Hackathon

Happening next fall
quarter!

Be an Organizer

Applications should
open soon-ish!

Other Questions?

Pluto is considered a
dwarf planet

