Front-End Web Development

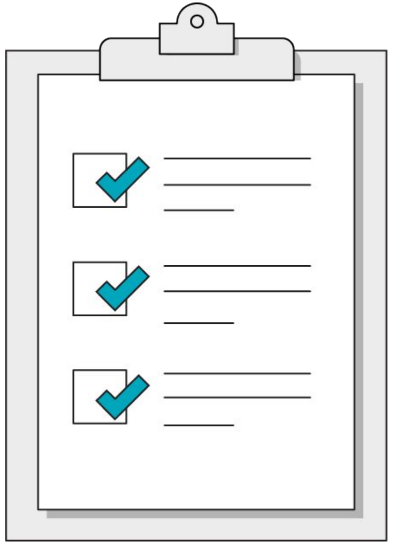
# CSS Layouts

GENERAL ASSEMBLY

## Today’s Learning Objectives

In this lesson, you will:

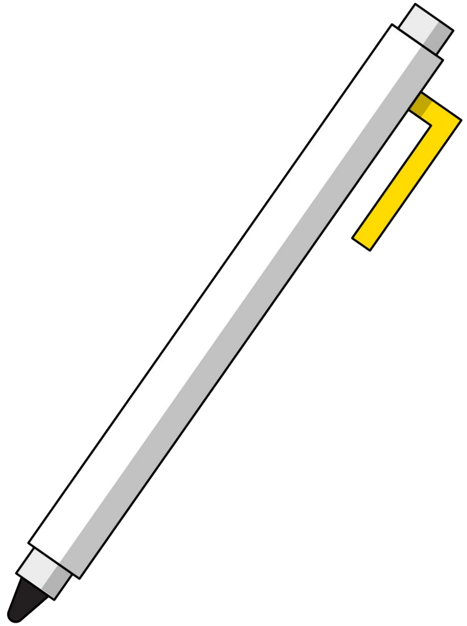
- Link to files from HTML using relative paths.
- Apply normalizing CSS to avoid browser default styling interference.
- Use margins and padding to create spacing between elements.
- Set the display property of elements to create page layouts.



**Discussion:**

## Quick Review

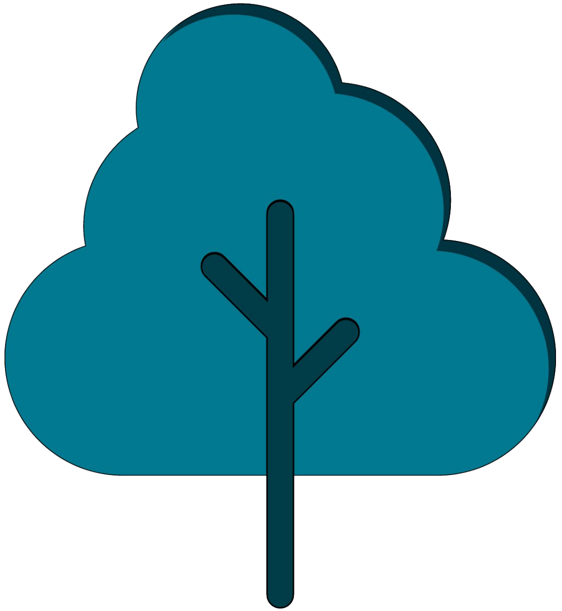
- What are the two major types of HTML tags?
- What are the three parts of a CSS declaration?
- What are some CSS selectors we've learned?
- Who has explored some CSS properties since last class?



Front-End Web Development

## Structuring HTML

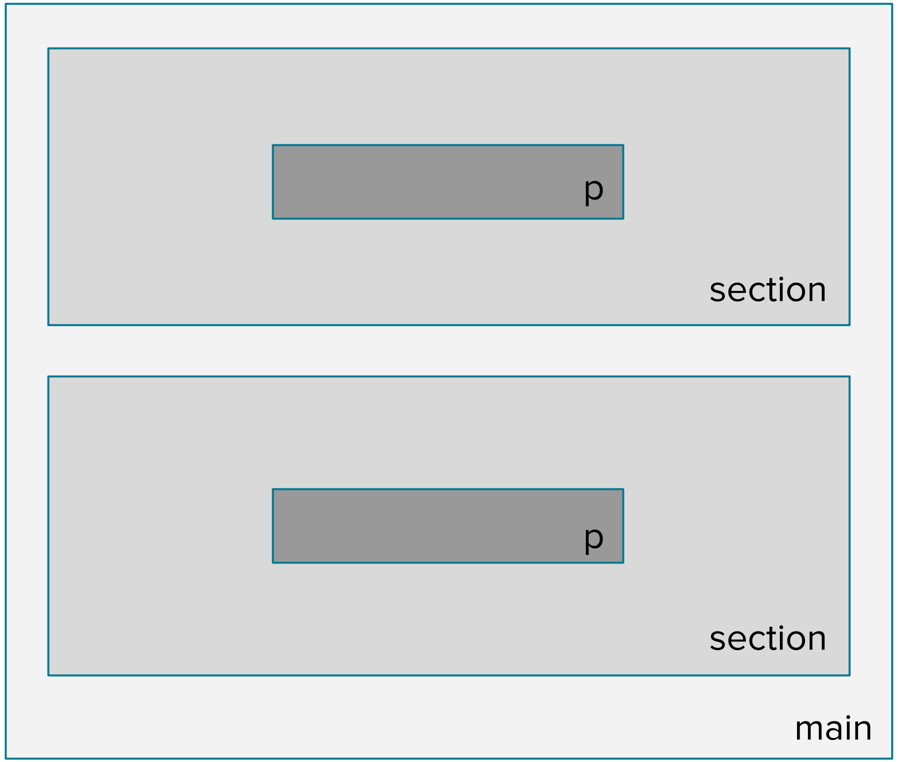
## The DOM Tree



A visual diagram of a webpage’s HTML structure.

## Tree-Like Structures Can Be Visualized

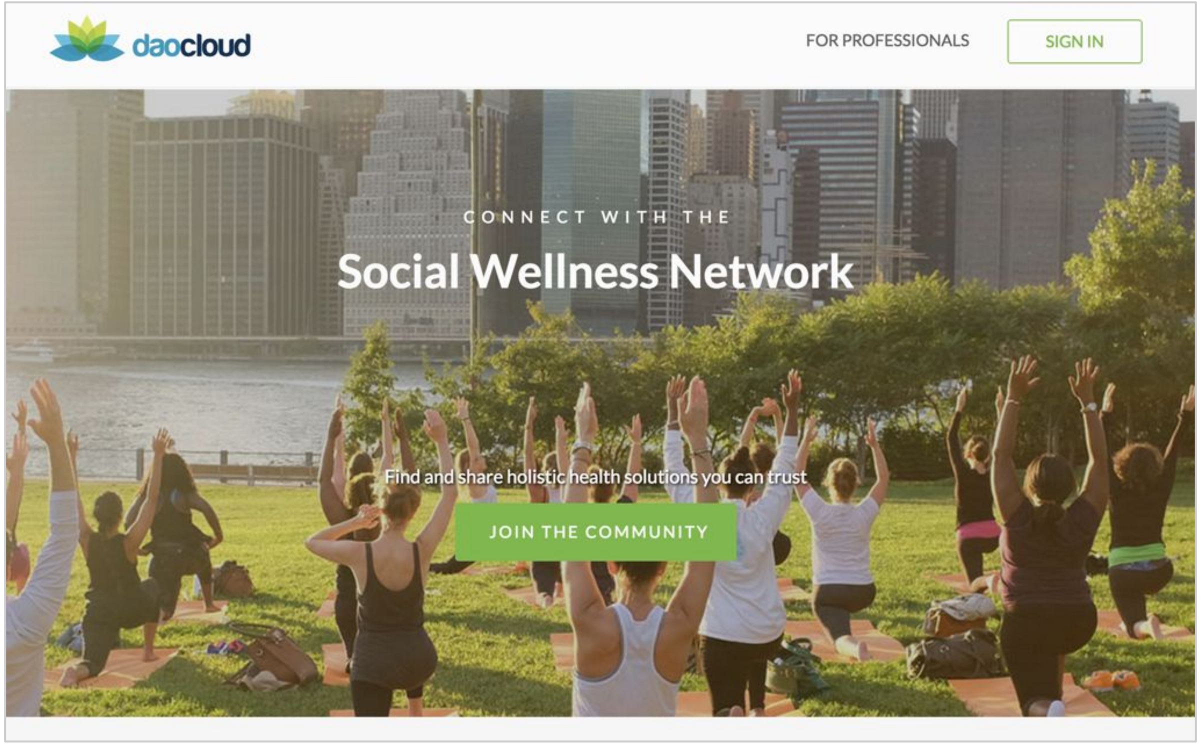
```
<main>
    <section class=”1”>
        <p>
            Content
        </p>
    </section>
    <section class=”2”>
        <p>
            More content
        </p>
    </section>
</main>
```



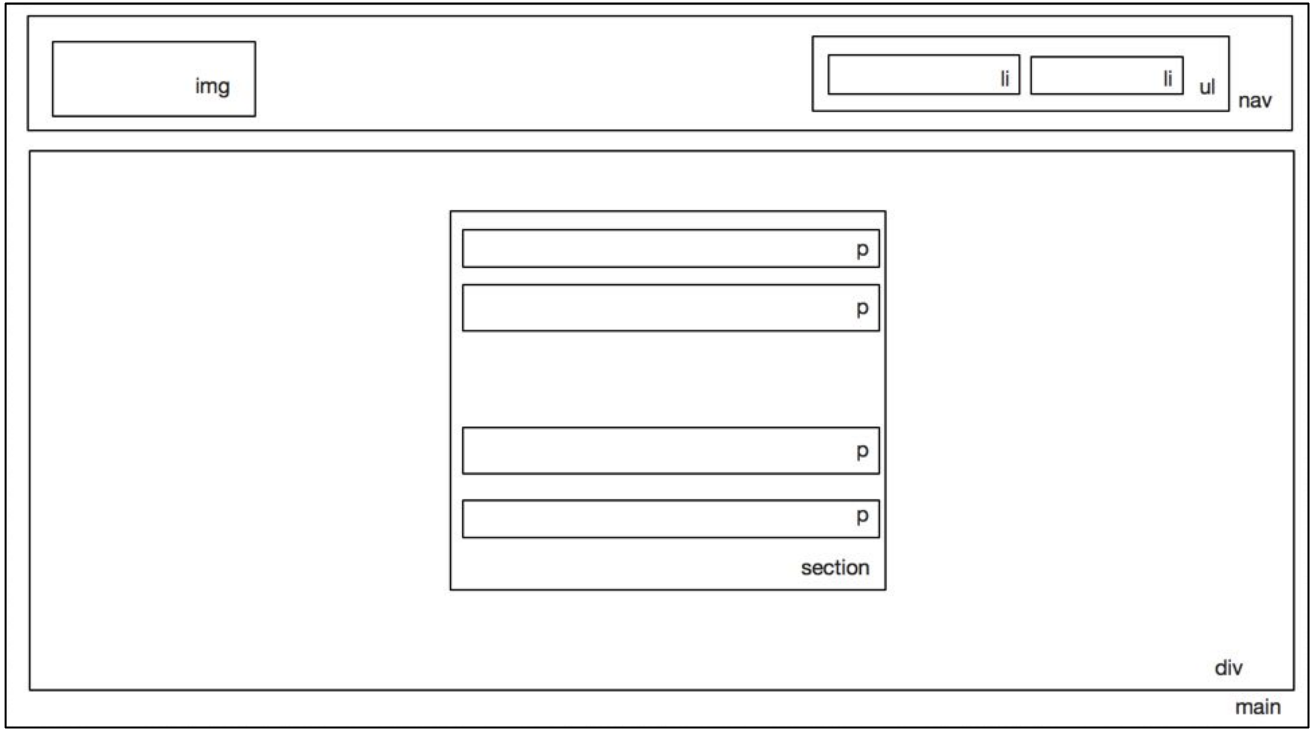
Group Exercise:

# How Is This Page Structured?

15 minutes



# Remember: HTML Is Subjective (to Some Degree)



Front-End Web Development

## The Box Model

Guided Walk-Through:

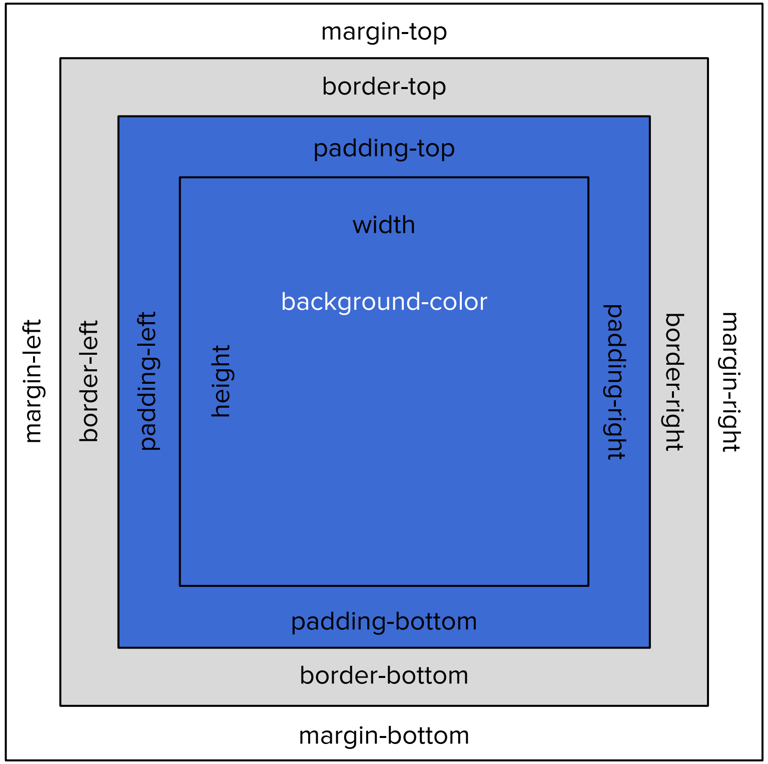
## CSS Box Model

As we discuss the details of the box model, you can reference the following CodePen for specific examples and illustrations of each concept:

**Reference code:**

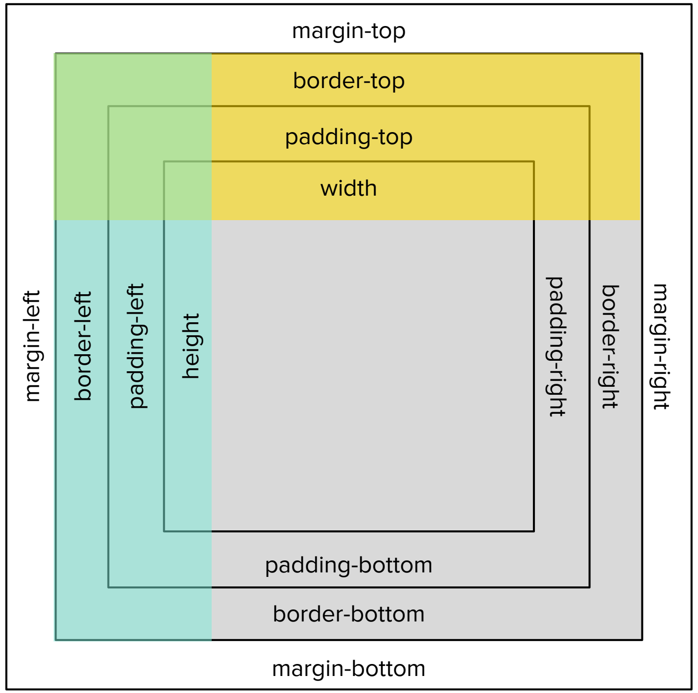
https://codepen.io/GAmarketing/pen/eYYjQXQ

## The Box Model



## Box Model Basics

- Every block-level element on a webpage is a box.
- Block elements have a **`padding`**, **`border`**, and **`margin`**.
- **`width + padding + border`** = actual width of an element's box.
- **`height + padding + border`** = actual height of an element's box.
- **`margin`** is outside the box and does NOT count toward height and width.



Front-End Web Development

## CSS Display Property

## `display`

**`display`** controls the behavior of the box in which content sits. We'll cover several of the most commonly used values of the **`display`** property:

- `Block`
- `Inline`
- `Inline-block`
- `None`

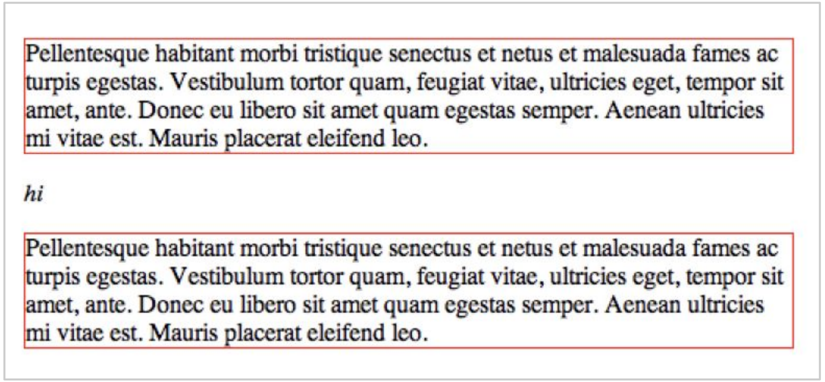
## Block vs. Inline Elements

- Block-level elements inherit the box model.
    - **`div, section, ul, nav, header, footer`**
- Inline-level elements do NOT inherit the box model.
    - **`span, img, sub, sup, textblock`**
- You can apply **`display: block;`** to an inline element via CSS and it will become a block-level element (and vice versa).
- Refer here for list of inline elements: https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements

## Positioning Elements: block

**`display: block;`**

- This element takes up as much width as possible and the following element drops to a new line.



Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

*hi*

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

**Image source**; css-tricks.com

## Positioning Elements: `inline`

`display: inline;`

- This element takes up only as much width as it needs. Padding and margins only work for **`left`** and **`right`**, not **`top`** and **`bottom`**.
- **`top`** and **`bottom`** spacing is controlled by the **`line-height`** property because the content is inline.

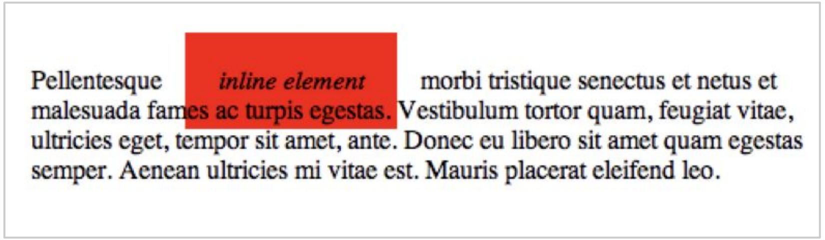
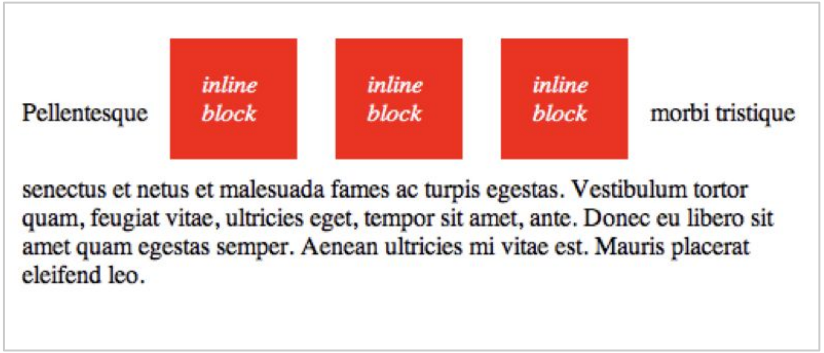

Pellentesque *inline element* morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

**Image source**; css-tricks.com

# Positioning Elements: `inline-block`

`display: inline-block;`

- This combines the two concepts.
- Inline blocks display inline with other items but allow you to use all **`margin`**, **`padding`**, **`height`**, and **`width`** properties.



Pellentesque morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

**Image source**; css-tricks.com

## Positioning Elements: none

**`display: none;`**

- Nothing shows up at all. It's in the DOM the browser sees, but the user won't see it. You can use Chrome DevTools to see the browser's view.

- This may seem useless now, but just wait until we hit JavaScript. You're going to love **`display: none`**.

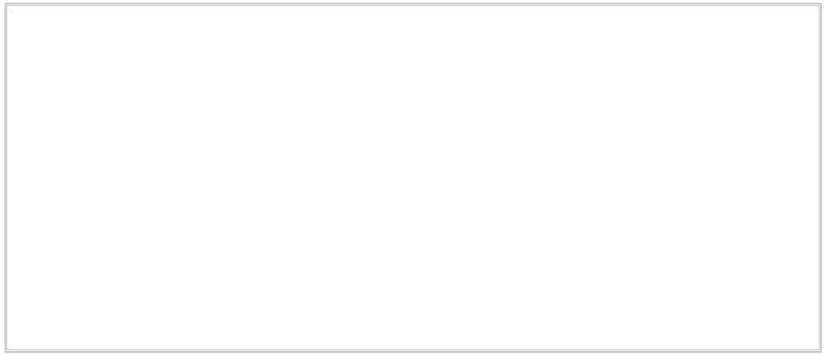
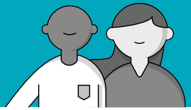


**Image source**; css-tricks.com



Partner Exercise:

## Practicing the Box Model

60 minutes

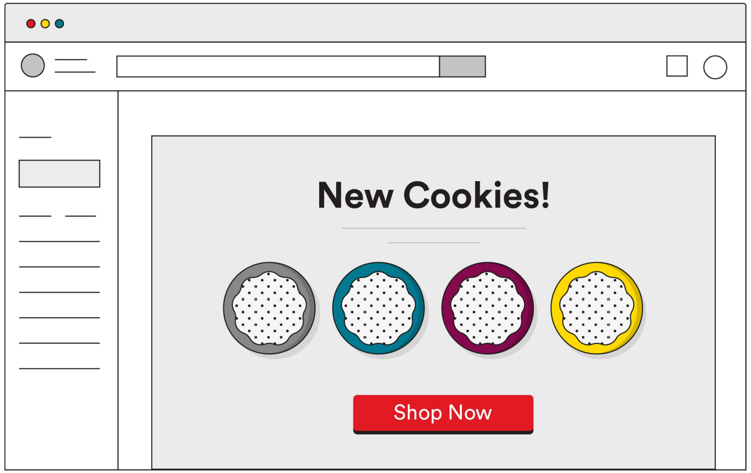
1. Draw a DOM tree to help lay this page out: https://s3-us-west-2.amazonaws.com/s.cdpn.io/2522641/wireframe-basic.pdf
2. Apply the box model CSS elements to the HTML in the sample. Adjust the box model to test what happens.
3. If you're feeling really ambitious, try to align the three elements horizontally.

Front-End Web Development

# Normalizing CSS

## Normalizing Output

- Browsers all have their own **unique** ways of rendering things.
- Very smart people have compared and contrasted these very minor differences and **fixed them for you** — how nice!
- There are many of these fixes, but we'll make your life simple and point you to **the most popular one**, Normalize CSS: http://necolas.github.io/normalize.css/.

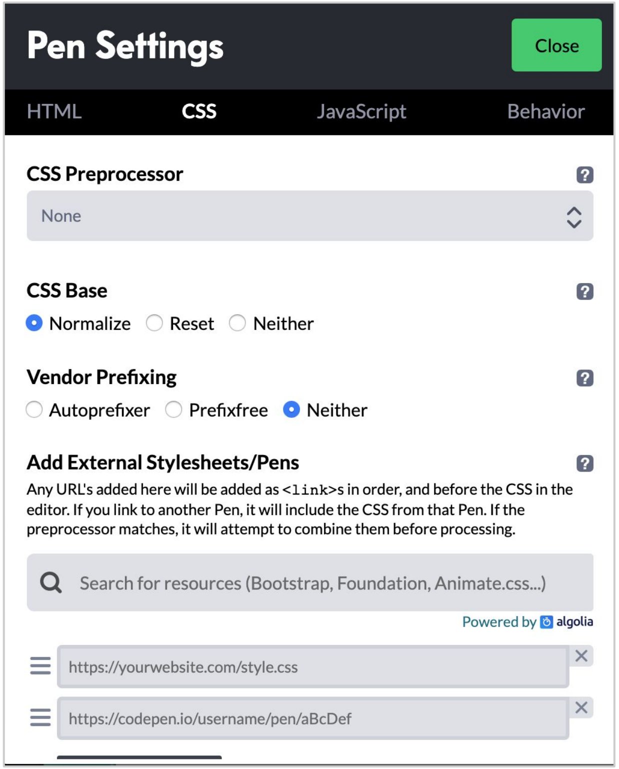


# How to Use Normalize

```
<head>
  <title>Something Unique</title>
  <link rel="stylesheet" href="css/normalize.css">
  <link rel="stylesheet" href="css/main.css">
</head>
```

1. Download Normalize.
2. Place Normalize CSS **before** your external CSS.
3. Code away like normal.

**Note**: In CodePen (see image), you use Normalize as a "CSS Base" by clicking the gear in the CSS panel.



CSS Preprocessor
None

CSS Base
[✓ Normalize] Reset Neither

Vendor Prefixing
Autoprefixer Prefixfree [✓ Neither]

Add External Stylesheets/Pens
Any URL's added here will be added as `<link>`s in order, and before the CSS in the editor. If you link to another Pen, it will include the CSS from that Pen. If the preprocessor matches, it will attempt to combine them before processing.

Search for resources (Bootstrap, Foundation, Animate.css...)
Powered by algolia

https://yourwebsite.com/style.css
https://codepen.io/username/pen/aBcDef

## Use Normalize on this week's homework!

## Key Takeaways

### Display and the Box Model

- Block elements have three key properties affecting their size and spacing:
  - `padding`
  - `margin`
  - `border`
- The **`display`** property is used to set elements next to or on top of each other.

## For Next Time

### Layouts With Flexbox

- Flexbox creates dynamic, responsive layouts.