



Lecture 3: CSS

Submit Homework

steve_fuller@berkeley.edu

Make sure you have submitted your designs

- Homepage
- About Page
- Discography

TrueGif.com

**The moment when you realize
you have homework on a
Sunday night**



TrueGif.com/191



CSS: Last Week Review

- More in depth look into HTML
 - Why HTML is important
 - Browsers are designed to render HTML
 - HTML has 3 main components
 - Text
 - References
 - Markup
 - Parts of HTML Doc
- Set up working 'class-directory'
- HTML Semantics
- Learned what HTML is meant to do and not do
- Elements, Attributes, and Values
- Parent > Child relationships
- URLs, linking items
- File naming

CSS: Questions over reading

- CH 7
 - Colors
 - Transparencies
 - URLs
 - Rules
 - Properties
 - etc.
- CH 9
 - Selectors
 - Relationships
 - Pseudo Elements
 - Pseudo Classes
 - etc.





CSS: This week overview

- What is CSS
- Why is CSS important
- Developer Tools
- Class Directory
 - Adding external style sheets
 - Setting up
- Naming files
- URLs, Linking, Images

There's going to be a lot to cover



CSS: Helpful Resources

- [StackOverflow - CSS](#)
- [W3Schools - CSS](#)
- [Can I Use?](#)
- [CSS Cheat Sheet](#)





CSS: What is CSS?

- CSS stands for Cascading Stylesheets
- CSS is **NOT** a programming language
- Where HTML defines your contents meaning CSS defines the appearance
- CSS can describe how HTML elements are displayed on
 - Screens
 - Paper
 - Other Media
- External stylesheets are stored in .css files
- Where HTML is more rigid, CSS there can be multiple way to do the same thing
- Fun Fact: CSS1 was released proposed in '94 and released in '96

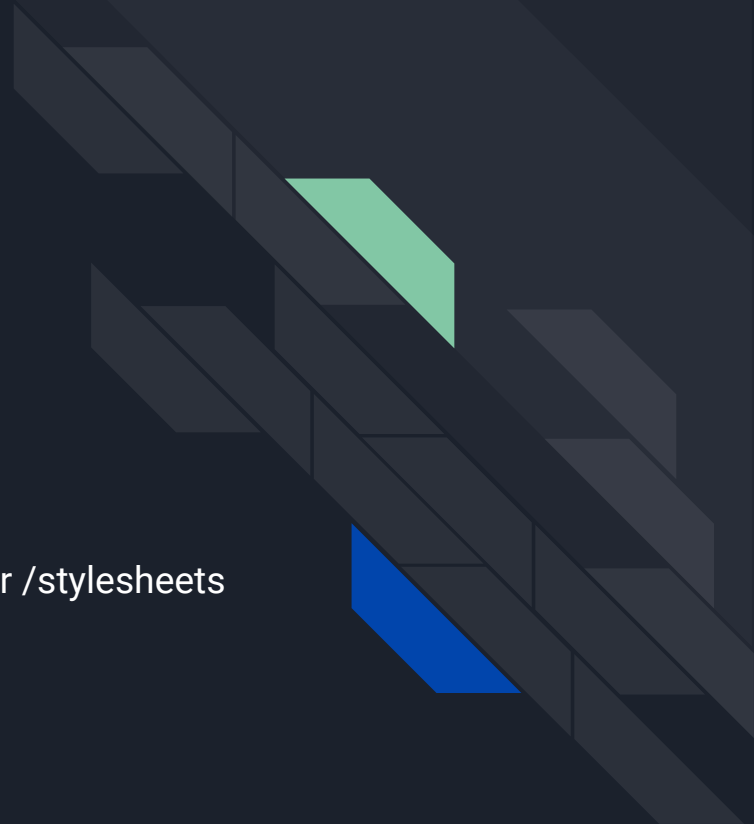


CSS: Solving a Big Problem

- HTML was never intended for presentation
- HTML was created to **describe the content** of a web page
- HTML 3.2, released in '97, allowed for attributes to affect a web pages presentation
- With CSS1 being released it was able to create a “separation of concerns” between presentation and content meaning

Open your class directory

1. Create a new HTML file called “css-basics.html”
2. Create a new CSS file called “css-basics-styles.css” inside your /stylesheets folder





CSS: Ways to add

CSS can be added in 3 way

1. Inline CSS
 - a. which is directly in HTML
 - b. `<tag style="width: 100px; ..." />`
 - c. Not recommended
2. Internal/Embedded CSS
 - a. Using `<style>` tags
 - b. Use sparingly
3. External CSS
 - a. Using a .css file
 - b. Recommended

CSS



Demo CSS methods (inline, internal, and external)



CSS: Rule

The diagram illustrates the components of a CSS rule: `a { background-color: yellow; }`. Brackets above the code identify the **selector** (`a`), **property** (`background-color`), **value** (`yellow`), and **declaration end** (`}`). Arrows below the code point to the **declaration start** (`{`), **property/value separator** (`:`), and **declaration separator** (`;`).

```
selector      property      value      declaration end
┌───┬──────────┬────────┬────────┴───┐
a { background-color: yellow; }
```

↑ declaration start ↑ property/value separator ↑ declaration separator



CSS: Constructing a Style Rule

- *Selector* - identifies the element(s) you want to format
- *Property* - a property is always associated with a *Value*
- *Value* - is the allowable option(s) for a *Property*

```
p { color: red; }
```

- p - is the *selector*
- color - is the *property*
- red - is the *value*

When using one or more properties attached to a selector
you create a rule



Yeah this is intense I know

CSS Basics



CSS: Colors

Colors can be defined in multiple ways

- Name: red, yellow, purple aqua
- Hexadecimal: #ff0000, #00ff00, #0000ff <<<< Preferred
- Hexadecimal Shorthand: #fff, #f0f, #00f
- RGB: rgb(255, 255, 255)
- Hue, Saturation, and Lightness: hsl(56, 100, 50)
- RGBA & HSLA are the same as above but add a transparency property 1 > 0.0

[Defining Colors in CSS](#)





CSS: Fonts

Working with fonts is another big part of your sites presentation

Many browser have predetermined “Web Safe Fonts” which give you a set of fonts ready to use

[Web Safe Fonts](#)

Fonts in CSS have a lot of different properties such as:

- font-family
- font-size
- font-weight
- color

[CSS Font Properties](#)



CSS: Typography Properties

Aside from choosing a font family, size, and weight. You also have the ability to choose *line-height* and *letter-spacing*

- [CSS line-height](#) - similar to Leading
- [CSS letter-spacing](#) - similar to Kering
- [CSS word-spacing](#) - similar to Tracking
- [CSS text](#)



CSS: Selectors

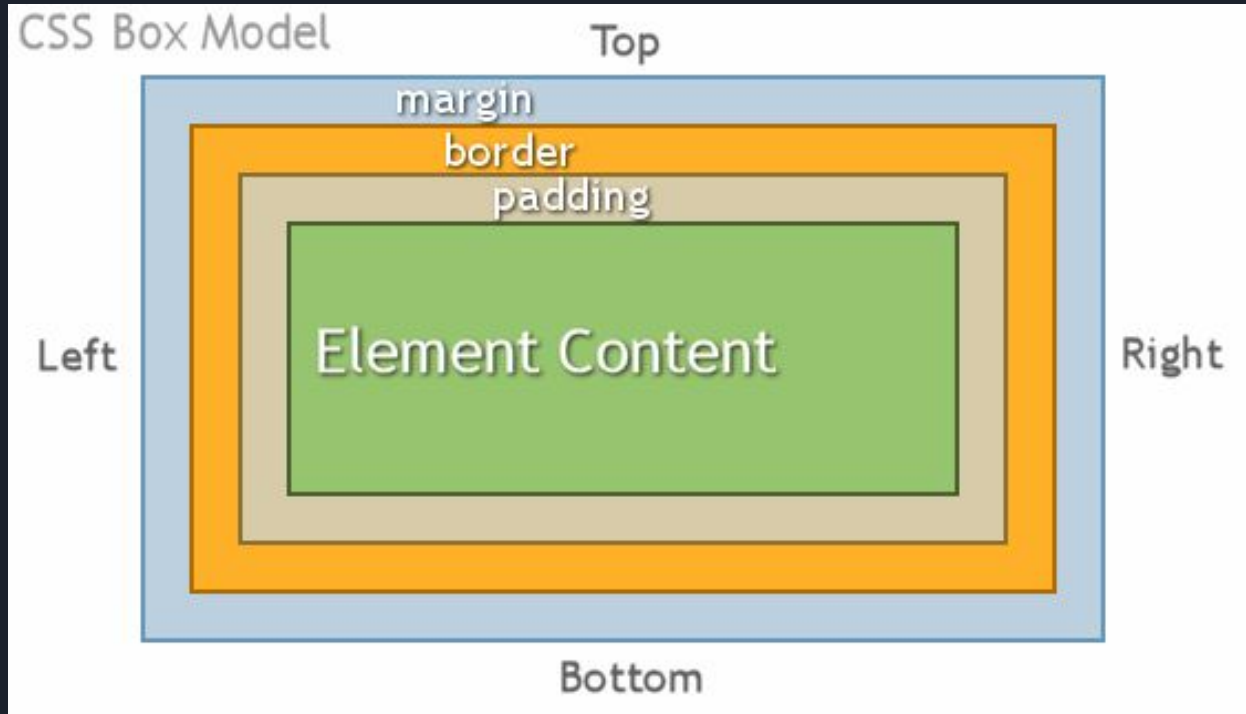
There are many selectors to hook your CSS into your HTML

- Class, most common
 - HTML `<tag class="hi">...</tag>`
 - CSS `.hi{...}`
- Element
 - HTML `<tag>...</tag>`
 - CSS `tag{...}`
- ID, for unique element per HTML page
 - HTML `<tag id="hi">...</tag>`
 - CSS `#tag{...}`
- *, which is a “Universal Selector” and matches elements of any type
 - `{...}`



4GIFs
.com

CSS: Box Model



[Link](#)



CSS: Box Model contd

All HTML elements can be considered boxes

In CSS, the term “box model” is used when talking about a layout

The Box Model is a box that wraps around every HTML Element

- Content: your text, images, etc.
- Padding: area around the content
- Border: wraps around the padding
- Margin: clears area around border

[Another important topic is box-sizing](#) which determines how your boxes are calculated



CSS: Shorthand

With many CSS properties you'll see things written in 1 of 2 ways

```
.class{  
  
margin-top: 10px;  
  
margin-right: 5px;  
  
margin-bottom: 10px;  
  
margin-left: 5px;  
  
}
```

Or in SHORTHAND

```
.class{  
  
margin: 10px 5px;  
  
}
```

[CSS Shorthand Reference](#)



CSS: Comments

Just like how you can make comments in HTML with `<!-- this is an HTML comment -->`

You can do the same in CSS `/* this is a css comment */`





CSS: Layouts

Currently all our elements are taking up full horizontal width. This is because we are using block elements, But now we want to make a layout of sorts

- [CSS Float](#)

This is a common way to create a layout for websites



CSS: Positioning

The position property specifies the type of positing method used for an element

- Static - Default, not affected by top, bottom, left, and right properties
- Relative - Positioned relative to normal position, affected by top, bottom, left, right
- Fixed - Positioned relative to viewport, leaves no gaps affected by top, bottom, left, and right
- Absolute - Positioned relative to nearest ancestor (usually inside Relative) affected by top, bottom, left, right
- Sticky - positioned based on users scroll.

[Link](#)



CSS: Backgrounds

Backgrounds are yet another way you can really enhance the visual appeal of your site

1. Find a background image by google search
2. Save that file to your /assets/images/your-image.png
3. Create a div with a class background image
4. In your CSS reference your image as a “background-image”
 - a. `background-image: url("../images/your-image.png");`

Try playing around with colors as well

[CSS Background property](#)



Hang in there I know...



CSS: Image Optimization

- When talking about performance, images are a big way to increase performance
 - a. Download a **LARGE** Google Image
 - b. Open <https://tinypng.com/>
 - c. Drag & Drop or Select from filesystem
 - d. Compare Images
- Do note I have seen issues with this and shadows on .png files with terrible results



CSS: Loading in Styles

- CSS can be brought in with various methods

- External Files
- Embedded `<style>` blocks
- Inline Styles

`<!doctype>`

`<html>`

`<head>`

`<title></title>`

`<link rel="stylesheet" type="text/css" href="/assets/stylesheets/styles.css" />`

`<style></style>`

`</head>`

`<body><div class="myBox">...</div></body>`

`</html>`

1. Open `/class-directory`
2. Within `/assets/stylesheets`
 - a. New File > `styles.css`
 - b. Add a Comment: `/* CSS Comment */`
3. Create `css-styles.html`
4. Add `<style>` tag in the head
5. Load in `styles.css`

Week 3: Homework



- Start Designing your Website
 - Designs for:
 - Media
 - Merch
 - Tour Dates
- Read Chapters
 - 12 & 14