

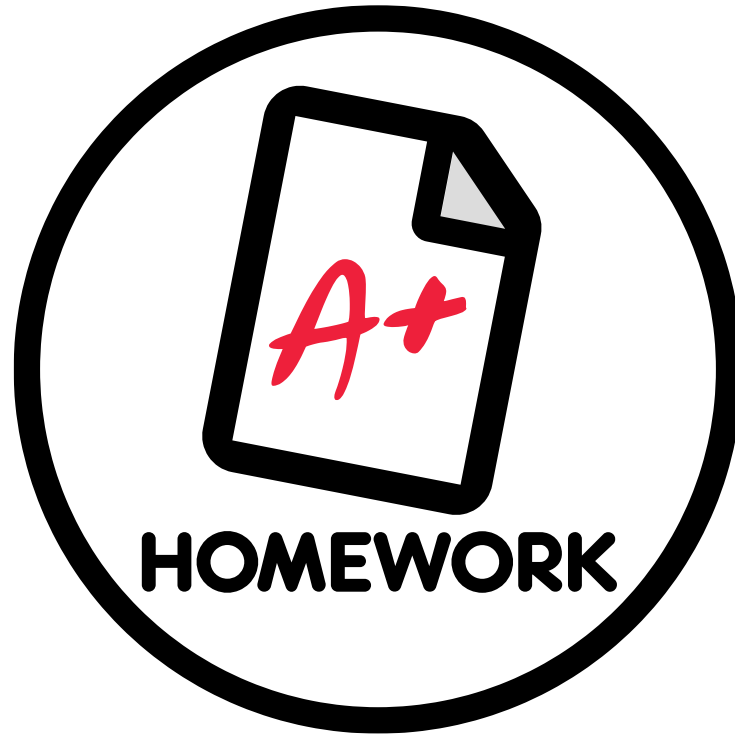


FEWD Week 1 • Class 2: Getting Deeper into HTML & CSS



Quick Review

- Describe the different parts of an HTML page
- What is the doctype used for?
- What tag is used to link to an external stylesheet?



Portfolio

Objectives

- Add images to the page and background
- Add padding and margins to elements
- Use fixed and relative units of measurement
- Identify block and inline elements
- Understand the CSS box model
- Add colors with transparency

Working with Images

Adding Images to Your Page

- Images can be added with the `` tag in HTML or in the background via CSS
- Background images are design elements only and are ignored by screen-readers

Content Images

- The `` tag has no closing tag
- The `src` attribute links the file and is required
- The `alt` attribute is used by screenreaders and for SEO

```

```

Background Images

Background images are added through CSS

```
/* The background-image property places  
the image in the background */  
  
selector {  
  background-image: url('path/to/file');  
}
```

► **FYI:** Elements can have multiple stacked backgrounds

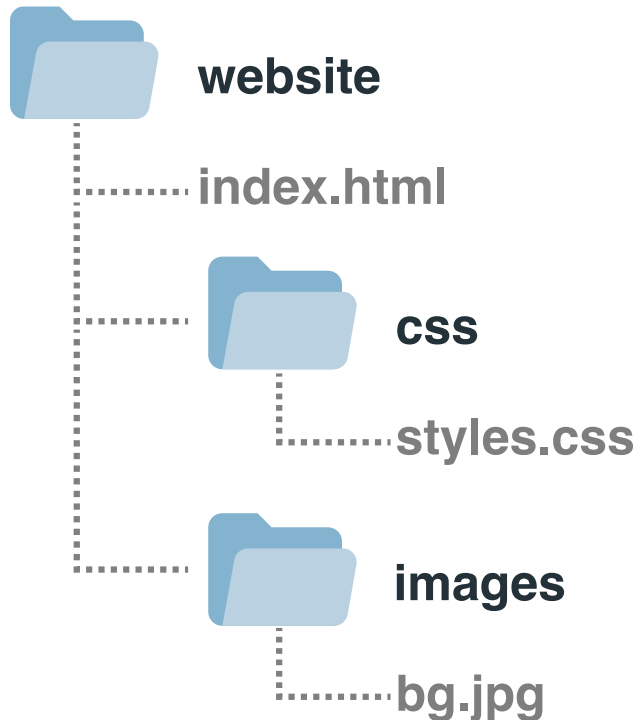
Background Properties

- `background-repeat`: tile the image or place once
- `background-position`: position the image in its containing element
- `background-attachment`: scroll with the page or remain fixed in one place
- `background-size`: the size of the image in the background
- `background-clip`: crop the image at the content-box, border-box or padding-box
- `background-origin`: place the image origin at the content-box, border-box or padding-box

Background Image Syntax

```
header {
  background-image: url('../images/bg.jpg');
  background-size: cover;
}
section { /* Multiple Backgrounds */
  background-image: url('../images/logo.png'), url('../images/bg.jpg');
  background-repeat: no-repeat, repeat;
  background-position: bottom right, top left;
}
main { /* Shorthand Format */
  background: url('../images/logo.png')
             no-repeat
             bottom right / 30%
             fixed;
}
```

Linking Files Refresher



- index.html → bg.jpg:

```

```

- styles.css → bg.jpg:

```
url("../images/bg.jpg")
```



Working with Images

CSS Floats

What is Float?

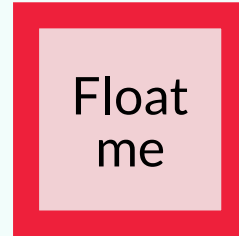
Float places an element on the left or right of its container and allows other elements to wrap around it.

Float Examples

```
div { float: none; }
```

```
div { float: left; }
```

```
div { float: right; }
```



We were the people who were not in the papers. We lived in the blank white spaces at the edges of print. It gave us more freedom. We lived in the gaps between the stories.

Float Property

The `float` property accepts the values:

- `right`
- `left`
- `none` (default)
- `initial` (resets to the default)
- `inherit` (gets its value from its ancestor)

Make it Stop!

Everything after the floated element in your markup will float. To cause elements to stop wrapping, use the *clear* property to the first element you want to go on its own line.

Clear Property

The `clear` property accepts the values:

- `right`
- `left`
- `both`
- `none` (default)
- `inherit` (gets its value from its ancestor)



Floating Elements



Images & Floats

Units of Measure

Units of Measure

Value	Description
px:	A fixed (aka absolute) value in pixels
em:	Relative to the font-size of the element (2em = 2 x the size of the current font)
rem:	Relative to the root element font-size
vh:	% of the viewport height (50vh = 50% of the viewport height)
vw	% of the viewport width
vmax	% of viewport's larger dimension
vmin	% of viewport's smaller dimension
%	It depends 🤖

* Not comprehensive.



Setting the Size of Things

Block and Inline Elements

Block vs. Inline Elements

Block

Block

Block

Inline Inline Inline

Block Example

```
<div style="height: 100px; width: 50%;">  
  <p>contents</p>  
</div>  
<span>I've been blocked!</span>
```



contents

blocked area

I've been blocked!

Inline Example

```
<span style="height: 100px; width: 25%; background: red;">Empty</span>  
<span style="height: 100px; width: 25%; background: red;">spans</span>  
<span style="height: 100px; width: 25%; background: red;">disappear</span>  
<span style="height: 100px; width: 25%; background: red;"></span>
```

Empty spans disappear

Introducing CSS display

The `display` property lets us change an element's default behaviors. It accepts many properties including `block`, `inline` and **`inline-block`**.

```
span {  
  display: inline-block;  
  height: 100px; /* This works now! 🎉 */  
}
```

Block vs Inline Elements

Block Elements

- ***Block*** elements from sitting next to them
- Size can be **specified** with height and width
- Often enclose other block or inline elements

Inline Elements

- Wrap inside containing elements
- Size is the height and width of contents only
- ***Should not*** enclose block elements



Block and Inline Elements

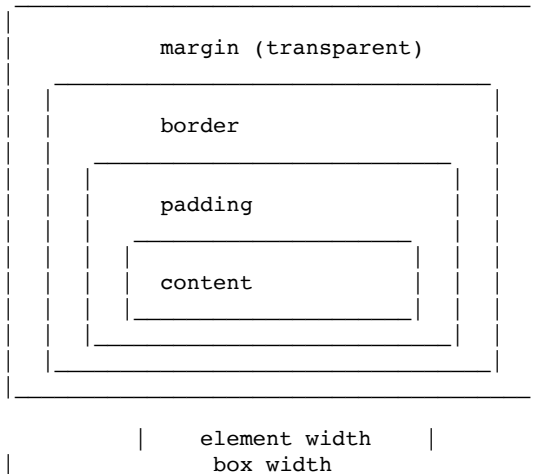
CSS Box Model

The CSS Box Model



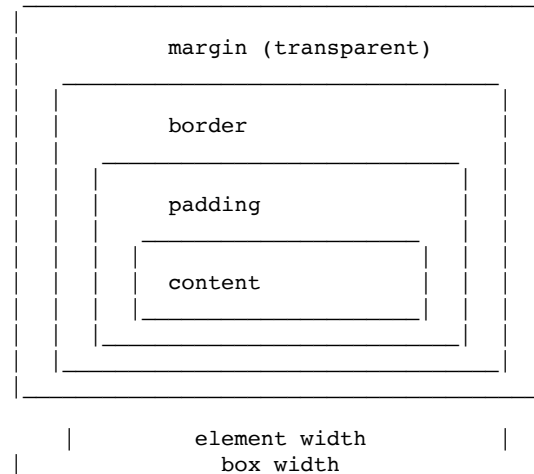
Box Model Drama

Box Model Standard



Cascading Style Sheets, level 1, W3C Recommendation
17 Dec 1996, revised 11 Apr 2008

Microsoft Box Model 🍌

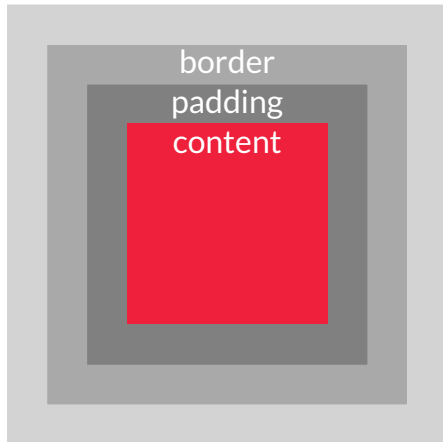


Revenge of the IE Box Model? Jeff Kaufman, February
18, 2012

...and the Mayhem It Created

Content Box

220px



Border Box

140px



Each box height and width is set to **100px** with padding and margin set to **20px**.

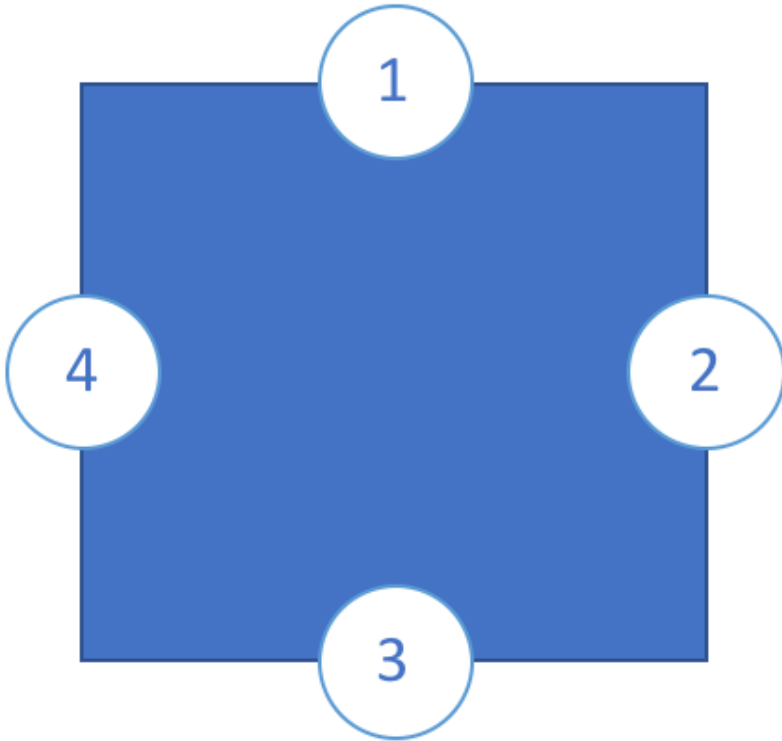
The difference is **box-sizing**!

The “Box Model Fix”

```
html {
  box-sizing: border-box; /* Sets border-box value for the root */
}

*, *::before, *::after {
  box-sizing: inherit; /* Since box-sizing doesn't inherit,
                        we force it to inherit! 💪 */
}
```

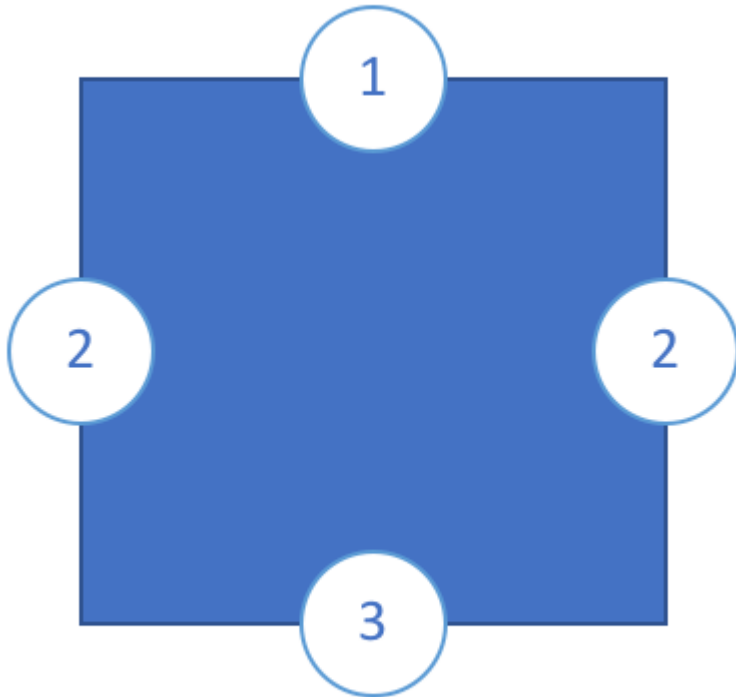
Trouble with Margins?



```
div {  
  margin: 12px 20px 8px 10px;  
}
```

Use this mnemonic:
Top - Right - Bottom - Left

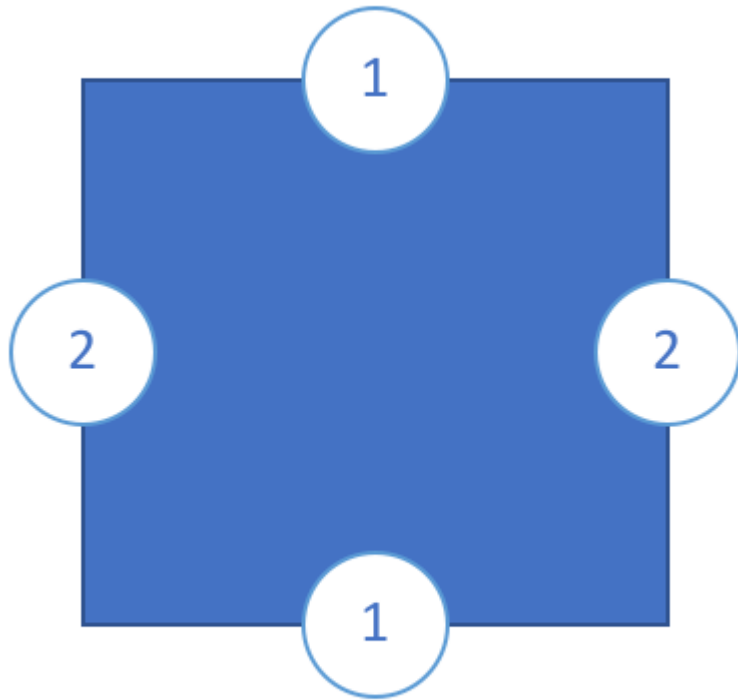
Margins/Padding 3 Values



```
div {  
  margin: 20px 50px 10px;  
}
```

Top - Right/Left - Bottom

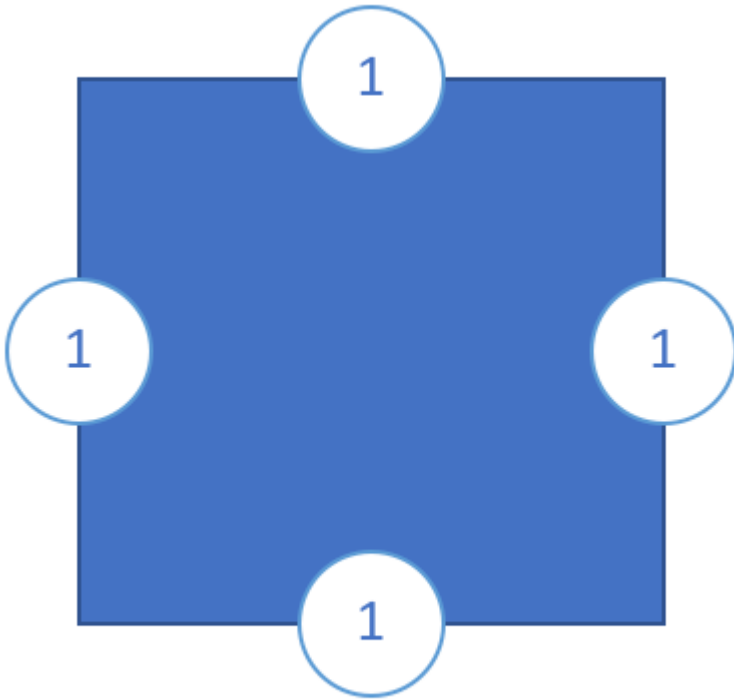
Margins/Padding 2 Values



```
div {  
  margin: 20px 50px;  
}
```

Top/Bottom - Right/Left

Margins/Padding 1 Value



```
div {  
  margin: 20px;  
}
```

All sides the same!



CSS Box Model

Color Values

- Keyword
- RGB & RGBA
- HSL & HSLA
- Hexadecimal

► **Alpha channels** are in values from 0 to 1, where zero is transparent and 1 is 100% opaque.

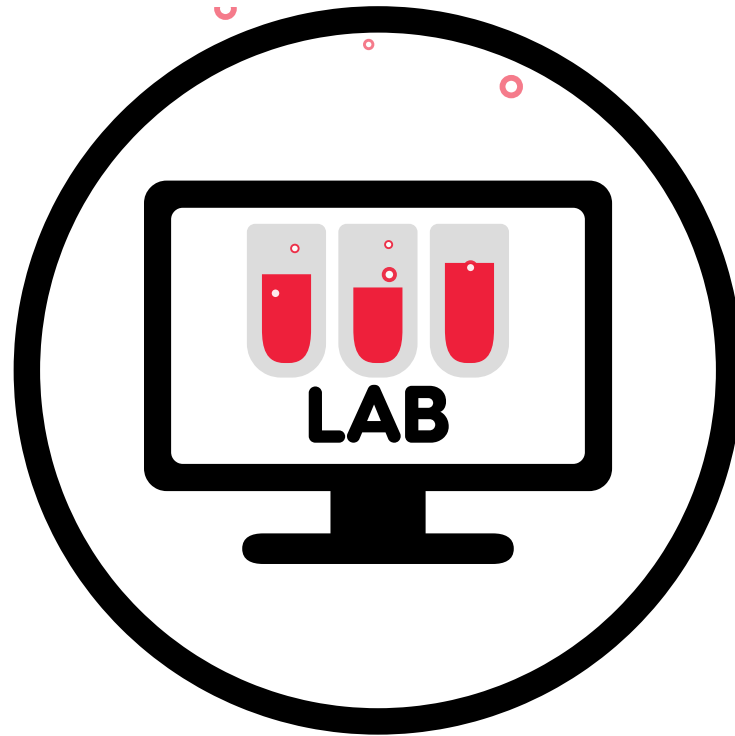
```
/* Keyword Syntax */
h1 {
    background-color: gray;
}

/* RGB & HSL Syntax */
p {
    color: rgba(0,0,0,1);
    border: 2px solid hsl(0,0%,0%);
}

/* Hexadecimal Syntax */
div {
    background: #ff0000;
}
```



Colors



Cookie Recipe More CSS

Go Do *Awesome* Things!