

# Code in One Day:

## HTML & CSS Crash Course



# Introductions

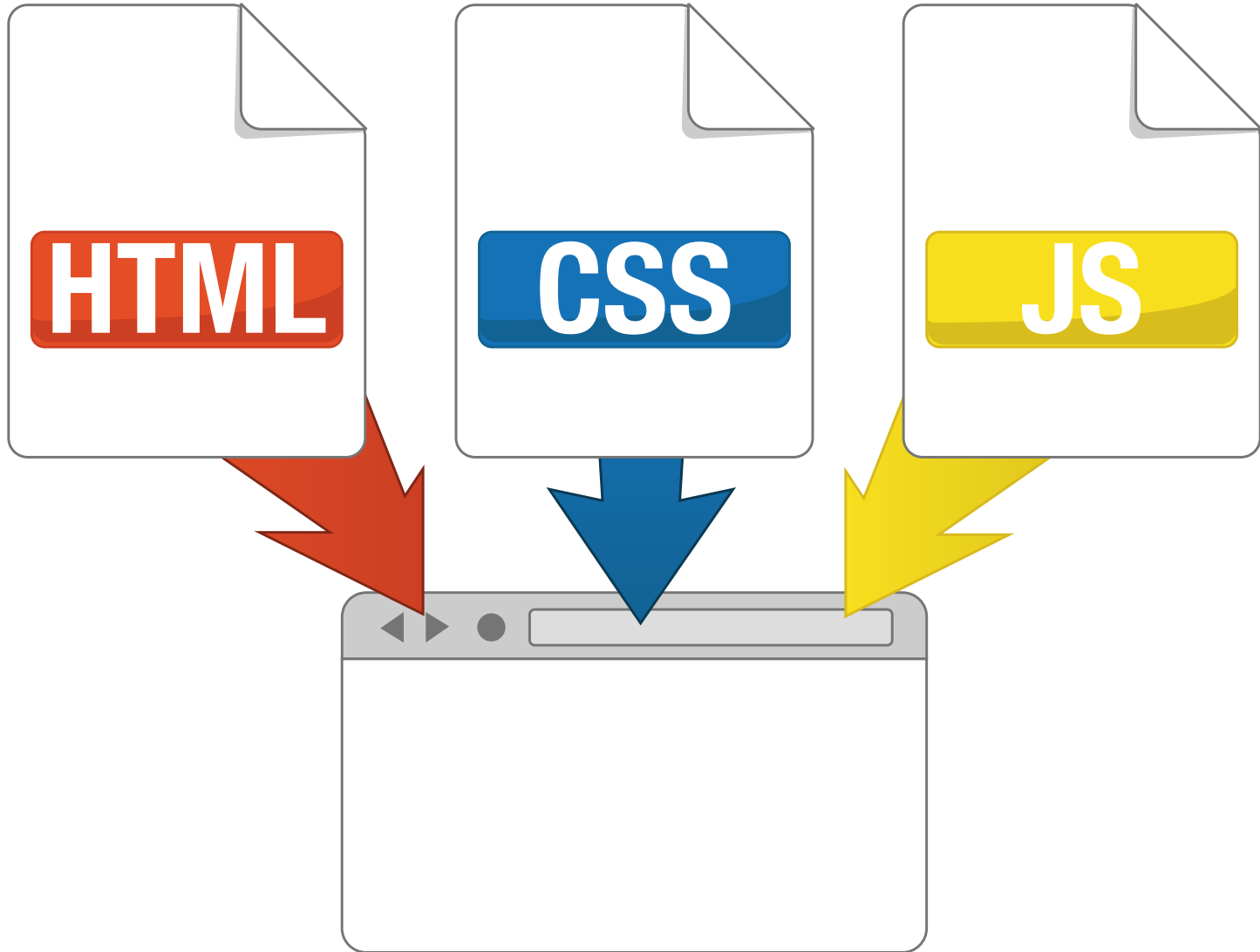


# What We'll Cover

- Building Web Pages with HTML & CSS
- How Stuff Behaves in the Weird World of the Web
- Where to Go from Here

# Primer

# How Web Pages Work

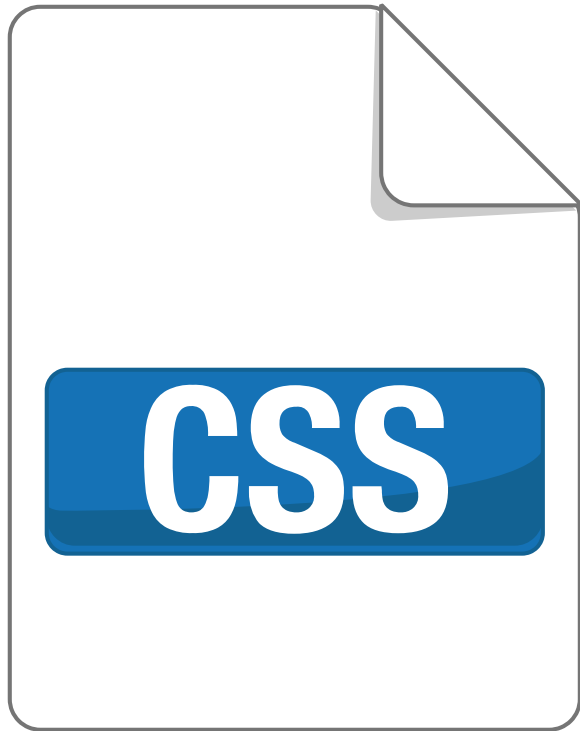


# Hypertext Markup Language



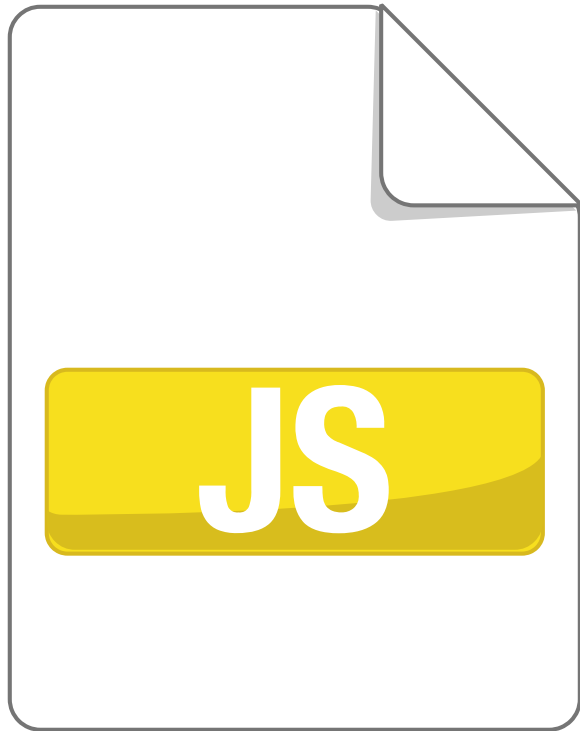
HTML provides the  
**structure** for your web  
pages

# Cascading Style Sheets



CSS defines how the elements of the web page should look

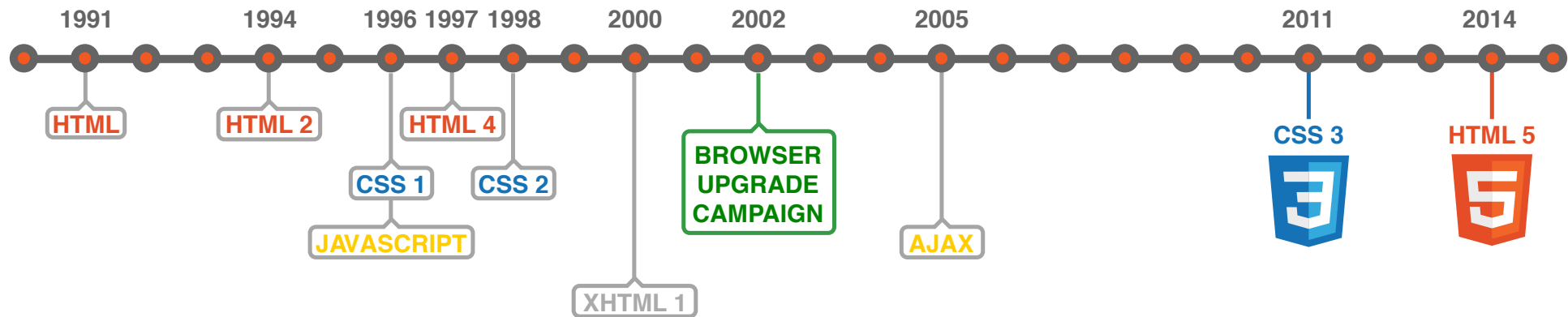
# Javascript



Javascript is used to add  
interactivity



# History of Web Standards



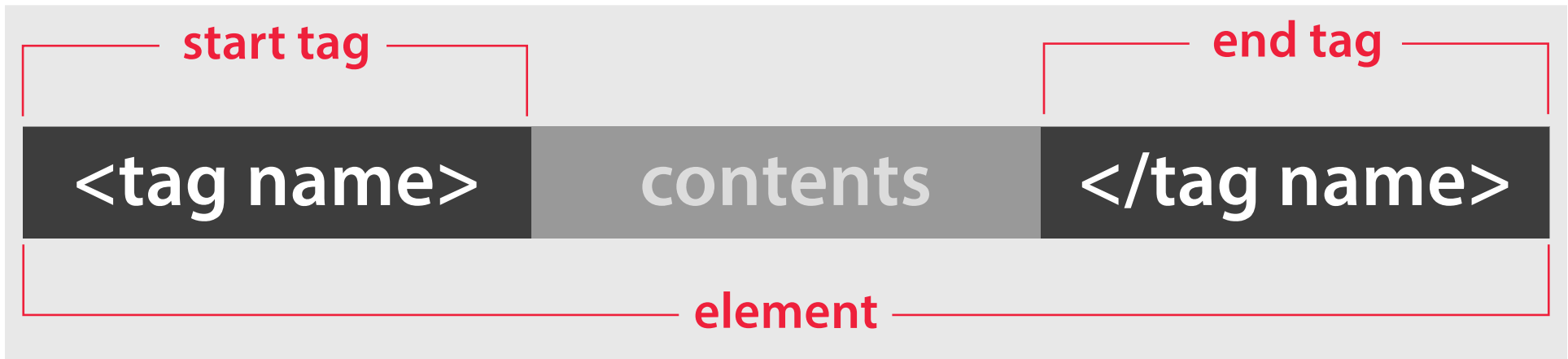
# HTML Tags & CSS Selectors

# Objectives: Part 1

- Describe the anatomy of a web page and create the basic structure of a page in HTML
- Apply HTML tags to web page content
- Style web pages with CSS
- Link external CSS files

# HTML Basics

# HTML Syntax



► **HEADS UP:** Some tags only have a start tag.

# HTML Syntax

**<tag name**

**attribute name = "value"**

**>**

may be optional or required

start tag

# What's up DOCTYPE?

- The HTML5 doctype tells the browser to behave **strictly** according to the standards
- Case-insensitive but conventionally written as:

```
<!DOCTYPE html>
```

# Anatomy of a Web Page

Every HTML page has the same foundational structure

```
<!DOCTYPE html>
<html>
  <head>
    <title>Your Title Here</title>
  </head>
  <body>
    Your Content Here
  </body>
</html>
```



## Content Tags

# Heading Tags

```
<h1>Largest Heading</h1>
```

```
<h2> . . . </h2>
```

```
<h3> . . . </h3>
```

```
<h4> . . . </h4>
```

```
<h5> . . . </h5>
```

```
<h6>Smallest Heading</h6>
```

## Content Tags

# Text Elements

```
<p>paragraph text</p>  
<code>computer code</code>
```

## Content Tags

# Unordered list

```
<ul>...</ul>
```

## Content Tags

# Unordered list item

```
<ul>  
  <li>Item</li>  
  <li>Another Item</li>  
</ul>
```

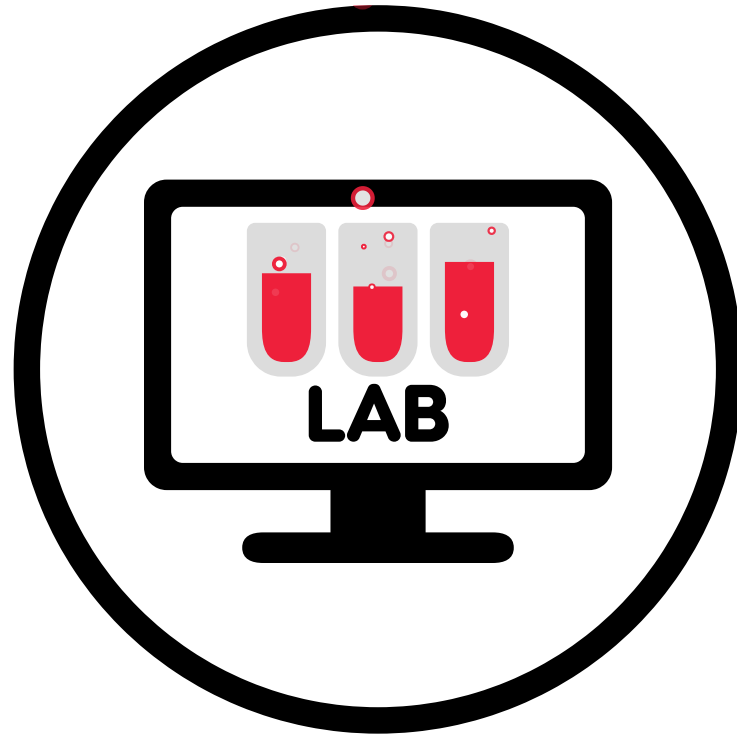
## Content Tags

# Links

```
<a href="url">Link Text</a>
```



# General Assembly Press Release



**Cookie Recipe HTML**

# Lab Setup

<https://github.com/jmeade11/crashcourse>

1. Go to the URL above
2. Click the green **Clone or download** button
3. Click the link to Download ZIP
4. Uncompress the crashcourse-master.zip file and move it to your desktop
5. Open the crashcourse-master and navigate to the Labs folder



# HTML Lab Tasks

<https://github.com/jmeade11/crashcourse/Labs>

1. In Sublime Text, choose **File>Open** and open the the Lab HTML folder
2. Click on the index.html file to open it
3. Add your HTML boilerplater
4. Copy the recipe from the online instructions and paste it in the body
5. Use what you've learned to add tags to the recipe

# CSS Basics

# Selectors, Properties and Values (Oh My!)

```
selector {  
  property: value;  
}
```

# CSS Selectors

Selectors target the elements you want to style

- Element Tags
- Classes & IDs
- Attributes
- Pseudo Classes
- Combinators

# Properties

Properties are the styles you want to apply

- border
- color
- background-color
- margin

# Values

Properties each have a set of specific values that they can accept

- border has values for the width, color and style
- color and background-color accept a color value



# CSS Basics

# External Style Sheets

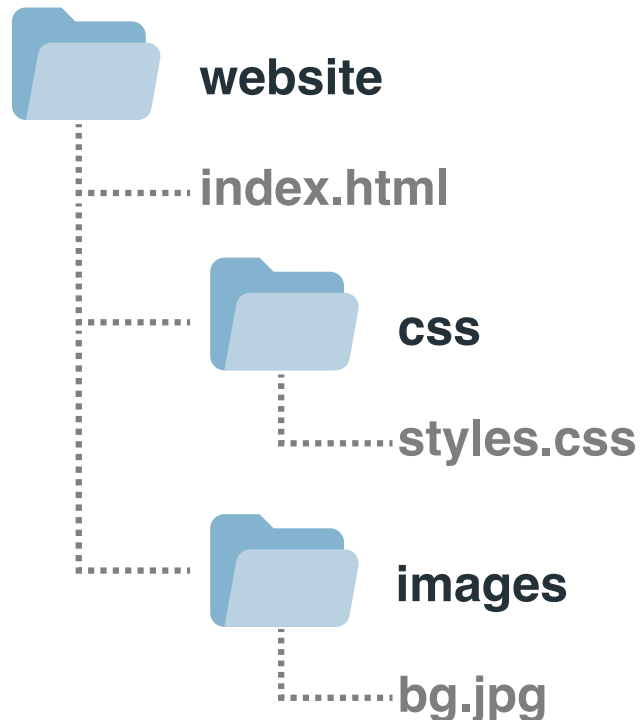


# Linking Files

```
<head>
...
<link rel="stylesheet" href="css/styles.css">
...
</head>
```

► **HEADS UP:** No `<style>` tags are used!

# Linking Files Relatively

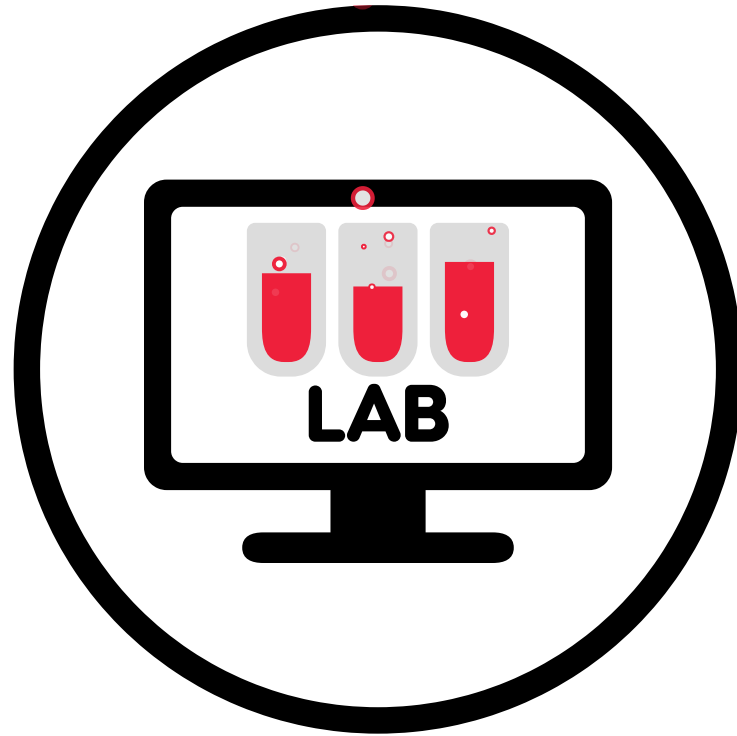


- index.html → styles.css:

```
href="css/styles.css"
```

- styles.css → bg.jpg:

```
"../images/bg.jpg"
```



# Cookie Recipe CSS

# Lab Task

<https://github.com/jmeade11/crashcourse/>

1. Use the recipe you created in the last lab
2. Set the **background color** for the page to **#FFE**
3. Change the **color** for the text reading "Nutrition Information" to **#887**
4. Style the **anchor tag** (link) color as **#900**
5. **Bonus:** Add a border on the page

**Give Yourself a Hand!**

**Ready, Set, Go!**

# 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?

# Objectives: Part 2

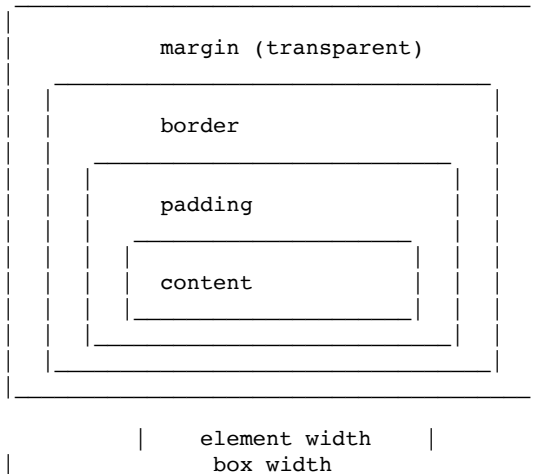
- Understand the CSS box model
- Identify block and inline elements
- Use fixed and relative units of measurement
- Add colors with transparency
- Add images to the page and background



# CSS Box Model

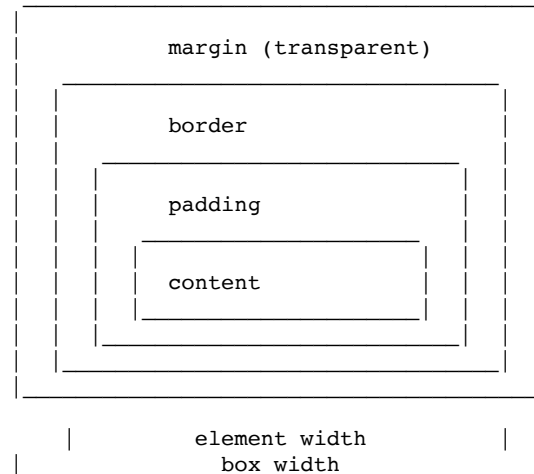
# The Box Model

## Box Model Standard



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

## Microsoft Box Model 🍌

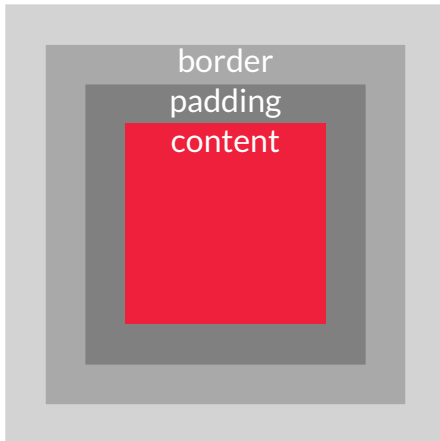


*Revenge of the IE Box Model?* Jeff Kaufman, February  
18, 2012

# Box-Model Mayhem

Content Box

220px



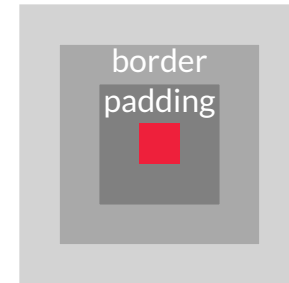
height: 100px;

width: 100px;

box-sizing: content-box

Border Box

140px



height: 100px;

width: 100px;

box-sizing: border-box



# CSS Box Model

# Block and Inline Elements

# Block vs Inline Elements

## Block Elements

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

## Inline Elements

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

# 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





# Block and Inline Elements

# Units of Measure & CSS Colors

# Units of Measure

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

*\* Not comprehensive.*



**Sizing**

# 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.

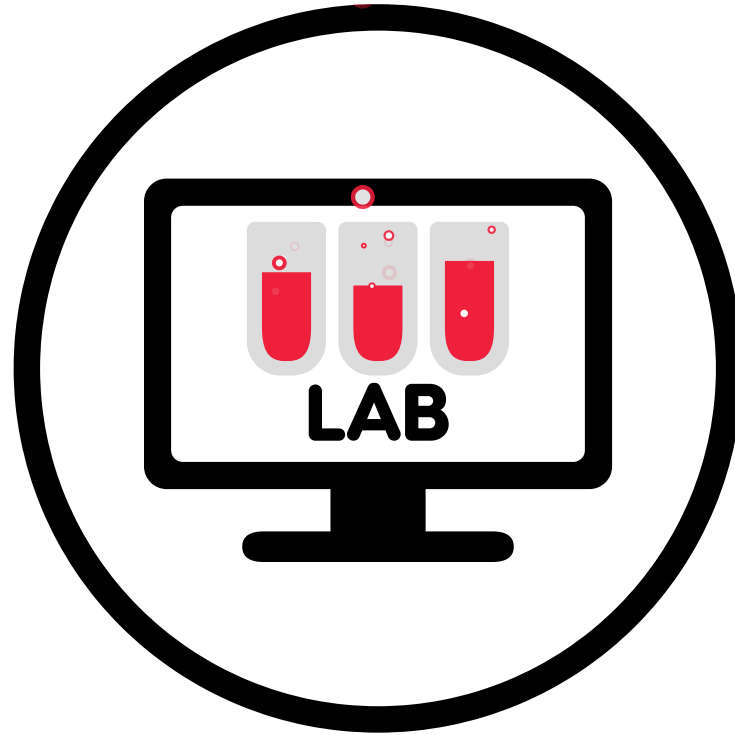
```
/* Keyword Syntax */
h1 {
    background-color: gray;
}

/* 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 CSS Part 2

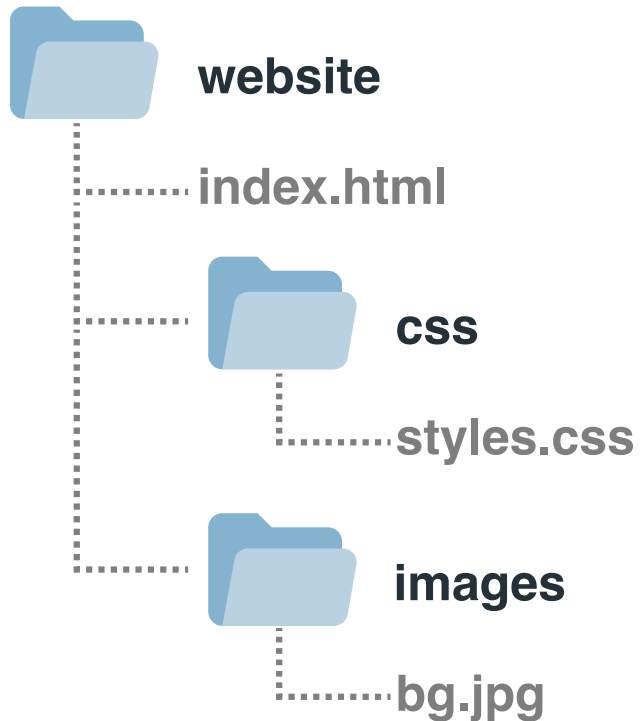
# Working with Images



# Adding Images to Your Page

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

# Linking Files Refresher



- index.html → bg.jpg:

```
href="images/bg.jpg"
```

- styles.css → bg.jpg:

```
"../images/bg.jpg"
```

# Content Images

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

```
<img href="images/dog.jpg" alt="A picture of my dogs">
```

# Background Images

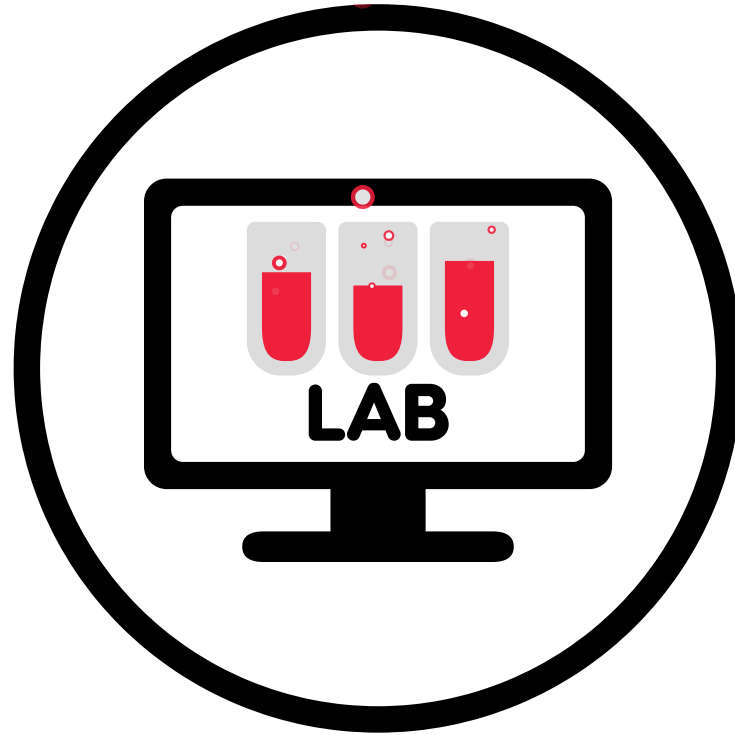
- The `background-image` property takes a `url('path/to/file')` property
- Control the background image with: `background-repeat`, `background-position`, `background-attachment`, `background-size`, `background-clip`, `background-origin`
- Elements can have multiple stacked backgrounds with different properties

# 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;
}
```



# Working with Images



**Cookie Recipe Images**

# Phew! You made it...

You've earned a break



**Ready, Set, Go!**

# Quick Review

- What is one type of color value?
- What's the difference between the content-box and border-box properties?
- What tag do you use to add an image to your HTML?

# Objectives: Part 3

- Use classes, IDs, and combinators to apply styles
- Understand how styles inherit/cascade
- Describe how conflicting styles are applied

# CSS Classes & IDs

# Selector Types

- Element Tags
- **Classes & IDs**
- Combinators
- Attributes
- Pseudo Classes & Pseudo Elements

# Classes & IDs

- Classes and IDs allow us to target elements without having to use the tag as a selector
- You can combine them with other selectors

# IDs

- An ID may only be used once on a page
- An element may only have one ID

```
<div id="extra-special">
```

```
#extra-special {  
  ...  
}
```

# Classes

- Classes are reusable as many times as you want
- An element can have as many classes as you want

```
<div class="big primary">
```

```
.primary {  
  ...  
}  
.big {  
  ...  
}
```





# Using Classes & IDs

# Combinators

# Using Combinators

- Descendant: space
- Child: >
- General Sibling: ~
- Adjacent Sibling: +



# Using CSS Combinators



**Game Time! CSS Diner**

# Cascading & Inheritance

# CSS Cascades

- CSS properties inherit their values from their ancestors
- Properties can be overridden when you provide a rule that has more specificity
- Any rules that are not specifically overridden continue to be inherited

# Selector Weighting

1. Inline Styles (highest)
2. IDs
3. Classes & Attributes
4. Element Tags



# When Specificity Matters

- Last declaration for a selector wins!
- Styles added directly to an element are the last that will be rendered.
- One exception is the special `!important` attribute.
- The more specific the rule the more importance it is given
- When there are multiple rules that contradict one another, the specificity and weighting rules are applied



# Demystifying Cascade

# Another one done!

Get coffee to power through the last part!

**Ready, Set, Go!**

# Quick Review

- What does cascade mean?
- What selector is weighted highest?
- Is there any way to override a style attribute on a tag?

# Objectives: Part 4

- Use floats to have more control over your layout
- Be aware of more modern method for layout control
- Review some helpful shorthand properties

# CSS Floats

# Float Property

The `float` property accepts the values:

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



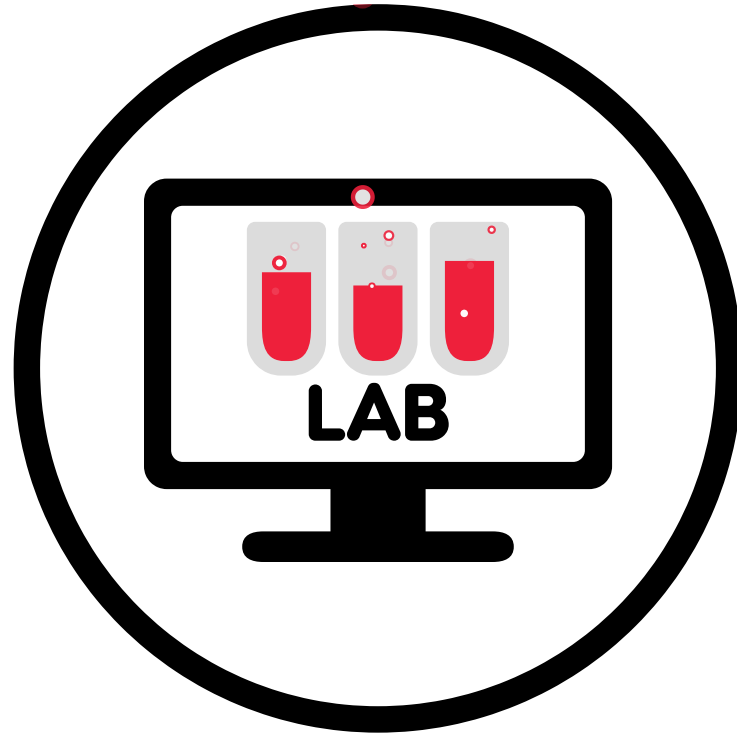
# Clear Property

The `clear` property accepts the values:

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



# Floating Elements



**Create A Portfolio**

# Go Do *Awesome* Things!

[www.linkedin.com/in/jenniferannmeade](http://www.linkedin.com/in/jenniferannmeade)