



# FEWD Week 2 • Class 3: Classes & IDs





**Review Time!**

# Objectives

- Work with colors and transparency
- Understand how to style elements with classes & IDs
- Use combinators for more specificity

# Color Values

# 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

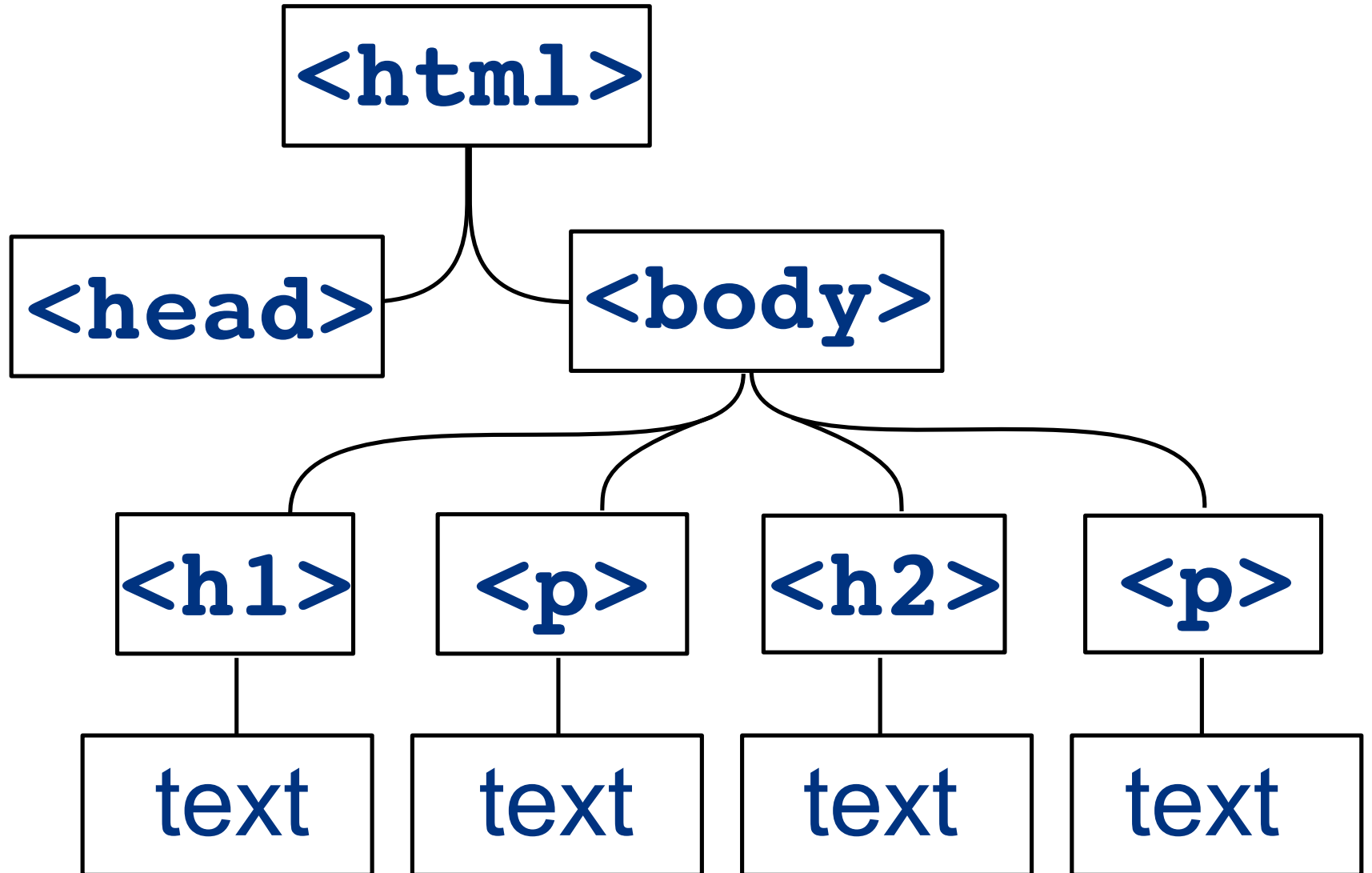
# Selectors & the DOM

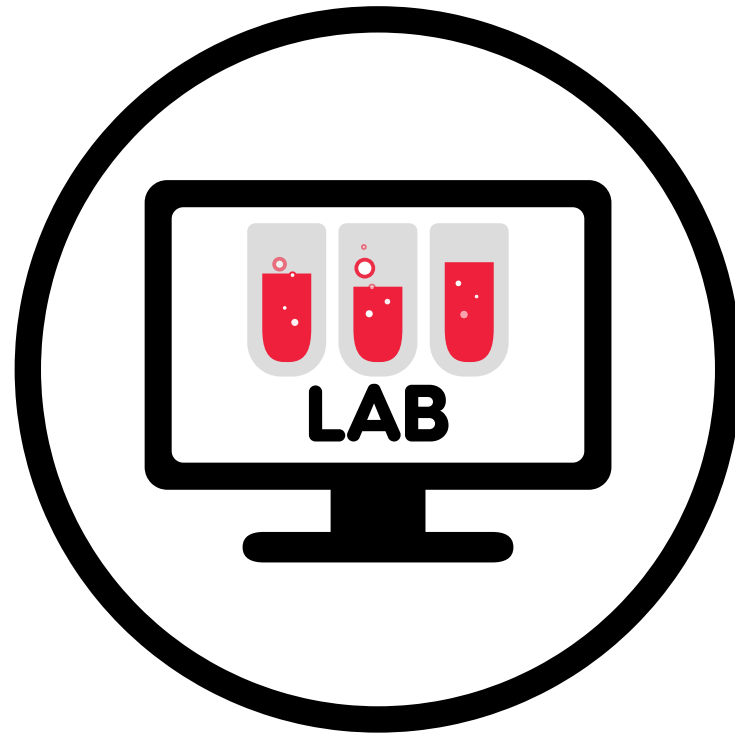
# Selectors

Selectors are how we **target** things in our the **DOM**.  
DOM stands for Document Object Model.



# DOM Tree





**Drawing the DOM Tree**



# Navigating the DOM

# Working with Combinators

# Selector Types

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

# Using Combinators

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



# Using CSS Combinators

# Classes & IDs



# Selector Types

- Element Tags
- Combinators
- **Classes & IDs**
- 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



**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

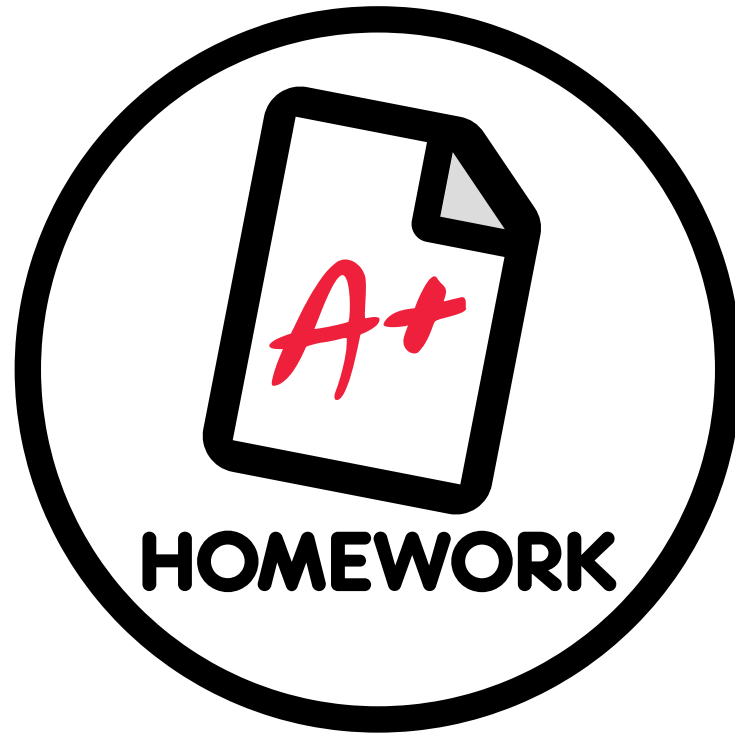
1. Inline Styles (highest)
2. IDs
3. Classes & Attributes
4. Element Tags

# When Specificity Matters

- For identical rules, the last declaration for 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



**Relaxr Blog**

**Go Do Awesome Things!**