

Fundamentals

Day 2

Introduction to CSS

Browsers have Style

Ever wonder why pages look a certain way in the browser?

Every element has a set of rules already defined for it.

CSS lets you change those rules.

What are these rules?

1. Size
2. Positioning
3. Color
4. *Font*
5. Behaviour

What does CSS look like?

```
header {  
    color: black;  
    font-weight: bold;  
    text-align: center;  
}
```

What does that mean?

Some elements are called *inline* elements.

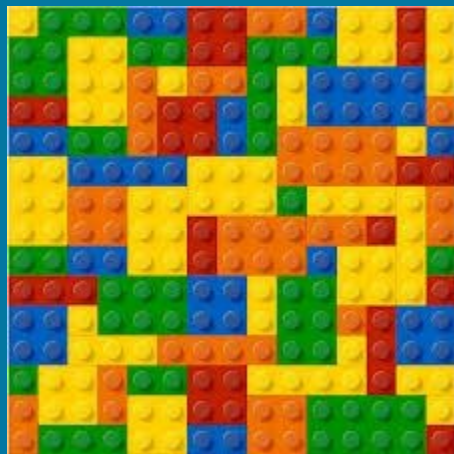
They can change the way they are displayed, but not the way they are aligned.

Other elements are called **block** or **block-level** elements.

Block elements will always try and fill up all of the horizontal space of the container in which you place them.

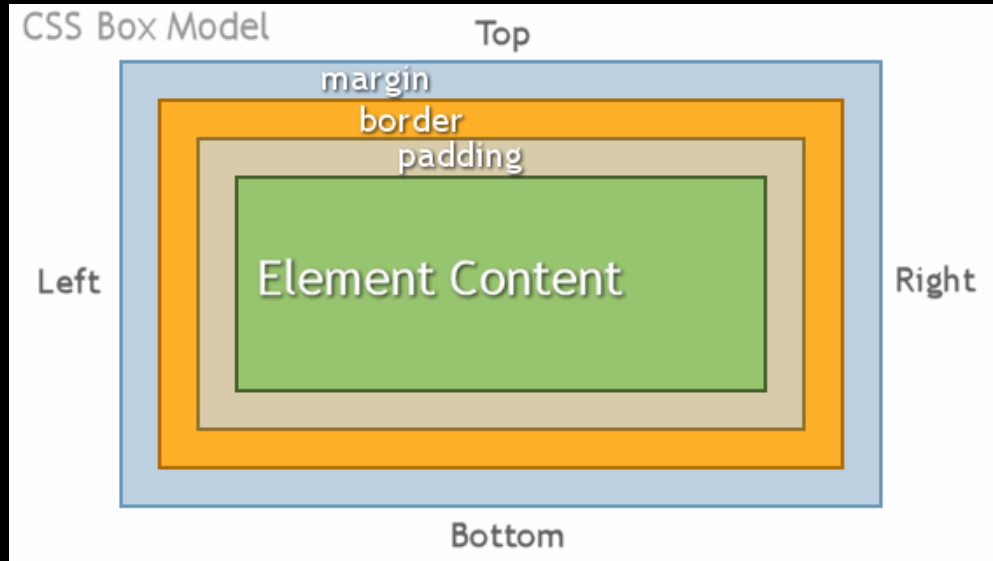
Blocks and blocks and more blocks

Browsers build pages the same way you would assemble a wall out of toy bricks.



Some bricks are longer and larger than others. But, they all have their place.

The Box Model



Okay, so how do we use this?

We rewrite the browser's rules when we need to change the layout of our page.

Then the browser figures everything out from there.

By managing the four box model properties, we can make our page have whatever layout we want.

CSS Properties

All CSS properties are in the format:

key: value;

We use our tag names to define them.

For example:

```
section {  
    margin-bottom: 2px;  
    border: 1px;  
    padding: 3px;  
}
```

Selectors

Whenever we define a rule in CSS, we identify the elements we want to modify using **selectors**.

Selectors tell the browser how specific to be in applying the rules.

```
p {  
  font-family: Arial;  
  font-size: 17px;  
  margin: 2px 5px;  
  padding: 5px;  
  border: 1px 2px 3px;  
  color: white;  
}
```

```
section p {  
  font-size: 13px;  
  color: blue;  
}
```

```
section header p {  
  font-size: 19px;  
  font-weight: bold;  
}
```

How do we customize further?

Classes and IDs:

Classes are used for sets of items on a page. You can have many instances of a class on the page.

Classes are set like this:

```
<p class="active">Text</p>
```

IDs are used for a single unique item on a page. There can only be one of each ID present on a webpage.

IDs are set like this:

```
<p id="main">Text</p>
```

Writing CSS for IDs and Classes

If we want to target this:

```
<section id="main">Our  
content.</section>
```

We need to write this:

```
#main {  
    padding: 10px;  
}
```

If we want to target this:

```
<p class="active">This  
paragraph is classy.</p>
```

We need to write this:

```
.active {  
    font-size: 16px;  
}
```

Specificity

**AKA: Only change what
you need to**

Order of Specificity

1. Inline Styles - These have the highest specificity, and are set up as **style** attributes on an element. Ex:
`<h1 style="color: blue;">Heading</h1>`
2. IDs - These are defined using the **id** attribute on an element. Ex:
`<p id="main">This is the main content.</p>`
3. Classes, attributes, pseudo-classes - This group includes **.classes** **[attributes]** selectors, and pseudo-selectors like **:hover**. Ex:
`<p class="special">This is some special text.</p>`
4. Elements and pseudo-elements - Regular elements like **p** **section** and **li**, as well as pseudo-elements like **:before** and **:after**