CSS 101

The Pizza to the HTML Party

Kara Carrell | Code Platoon

What is CSS?!

Cascading Style Sheet

CSS

IS:

- **A stylesheet language used to describe the presentation of an HTML document.**

  Rules that style both:
  - **Content** (inside an HTML element) and
  - **Element** (the HTML Element “BOX” itself)

- **Important to know**

  - CSS selectors will be integral to dealing with JavaScript and HTML

ISN’T:

- **Just for making HTML “pretty”**
  - Helps make the web accessible
  - Enhances usability
  - Has animations & mathematical components
- **Easy**
  - Takes Discipline
  - Takes an understanding of Scope
- **Hard**
  - Adhering to some rules, a few lines of CSS can improve an HTML document drastically!

Languages

- **Programming Language -** Ruby
- **Scripting Language -** JavaScript, Bash
- **Query Language -** SQL
- **Markup Language -** HTML
- **Style Language - CSS**

Basic rule syntax

Style rule syntax

```
selectorlist {
  property: value;
  [more property:value; pairs]
}
```

What CSS Looks Like

Words To Know:

CSS terms you'll hear most often

- Selector
- Property
- Value
- Specificity
- Pseudo
  - Pseudo-Element
  - Pseudo-Classes
- Layout
- Position
- Float
- Grid

Cascading means:

CSS will apply style properties based on the following four factors:

Importance

Author - person who wrote the CSS

User - person looking at page's CSS Settings

Default - browser/user agent's CSS

Source Order

Style declarations are read in order, and where links are placed will change order of style implementation

Specificity

```
#nav .selected > a:hover
```

```
.wrapper .box ul li
```

Inheritance

Some properties may be passed down to an element's children.

Applying CSS to HTML

Inline

```
<p style="color:red; font-style:bold;">
```

- Highest Specificity. Use VERY rarely.

Embedded

```
<style>
  p {
    color: red;
    font-style: bold;
  }
</style>
```

- Used for unique document styles. Not very sustainable for larger projects

External

```
<link rel="stylesheet" src="style.css">
```

- Best use, separates HTML from CSS coding. Allows for multi-document use. Link Order important
- Lives in <head> section

Selectors

Types of Selectors

Element Type

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

Pseudo-Element

```
p::first-line { color: red; }
```

Class

```
.class-name { color: red; }
```

Pseudo-class

```
a:visited { color: red; }
```

ID

```
#id-name { color: red; }
```

Attribute

```
a[target=”_blank”] { color: red; }
```

Combinators

- Adjacent sibling combinator `A + B`
- General sibling combinator `A ~ B`
- Child combinator `A > B`
- Descendant combinator `A B`

Examples:

```
.alert .intro { color: red; }
```

```
.intro > h1 { color: red; }
```

```
p.class-name { color: red; }
```

Properties

Properties

Popular Properties:

- Color
- Background-
- Font-
- List-
- Text-
- Padding
- Border-
- Margin

Hmmm… some of these look familiar...

Interesting Properties:

- Transition
- Animation
- Opacity
- Transform
- Z-index

Property Shorthands

For some properties, multiple values can be set in one line

Background (color, image, repeat, position)

```
.call-to-action {  background: #000 url(assets/smiley.png) no-repeat left top;  }
```

Border (width, style, color)

```
p {  border: 6px solid red;  }
```

Font (style, weight, size, height, family)

```
#my-page-title {  font: italic bold .8em/1.2 Arial, sans-serif;  }
```

Margin & Padding (top right bottom left)

```
.hot-dogs {  padding: 10px 20px 30px 40px;  }
```
(top, right, bottom, left)

```
.hot-dogs {  padding: 10px 40px;  }
```
(top-bottom, right-left)

```
.hot-dogs {  padding: 10px;  }
```
(top-right-bottom-left)

Display

What kind of box will be used?

Display Types

Block - TAKE ALL THE SPACE!

```
.container {  display: block;  }
```

Inline - Take the space I need...

```
.container {  display: inline;  }
```

None - 👻

```
.container {  display: none;  }
```

Flex - Flexbox Model

```
.container {  display: flex;  }
```

Grid - Grid Model

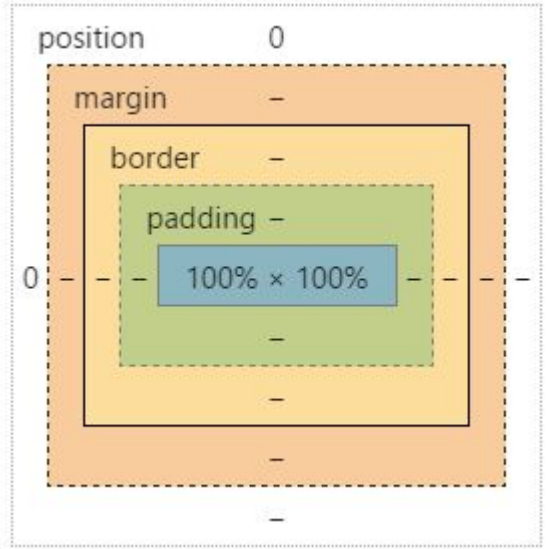
```
.container {  display: grid;  }
```

Back To The Box!

Setting any HTML element in the place you want it requires you to modify the element with the following properties:

- **Height & Width**
  - This pertains to the content itself
- **Padding**
  - The space between the content and the border
- **Border**
  - The line surrounding the element itself
- **Margin**
  - The space between one element and another

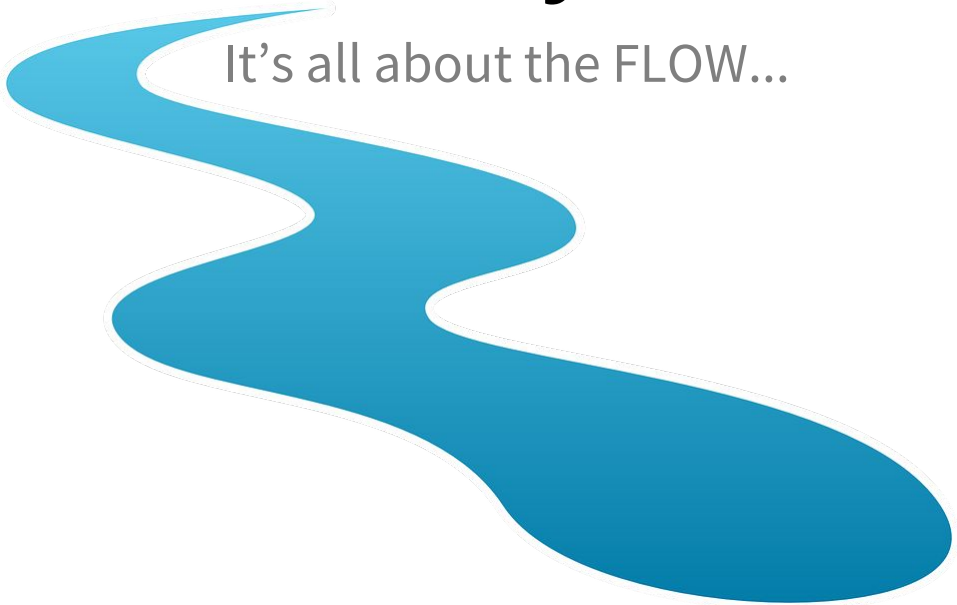
Box Model:



Let’s Make A Box!

CSS Layout

It’s all about the FLOW...



POSITION - The place a given element should be on a document:

Static to the flow of the document (default position)

To use ***Top Right Bottom Left*** *properties, you set a position:*

- **Relative** to it’s default position
- **Absolute**-ly placed , taken out of the document’s flow
- **Fixed** to the viewport, not the document. Stays put.
- **Sticky** will let your element stay on scroll

Stack & Float
The Boxes!