CSS FOUNDATIONS IN-DEPTH

The nitty-gritty of css . . .

What is CSS?

What is CSS?

Cascading Style Sheets

What is CSS?

Cascading Style Sheets

Style sheets define formatting rules that are applied to text, images, forms, and embedded and layout elements.

What is CSS?

How do we write styles?

A style sheet is comprised of "styles," which are governed by styling "rules."

"Rules" are applied to HTML to control visual presentation of pages.

What is CSS?

How do we write styles?

Each styling "rule" is comprised of three things:

selectors

properties

values

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

(This styles all instances of a paragraph tag in the HTML.)

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

p = selector

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

{ … } = declaration block

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

`color: #333333;` = declaration

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

color = property

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

#333333 = value

CSS “Rules”

Here is a simple example of a **CSS rule**:

```
p { color: #333333; }
```

; = declaration terminator

CSS “Selectors”

There are three basic types of selectors:

- **HTML** (or "tag") selectors
- **id** selectors
- **class** selectors

CSS *HTML* Selectors

HTML selectors use standard, predefined HTML tags and format them accordingly in a webpage's code.

```
p { color: #333333; }
```

…all instances of in the webpage will be the dark grey #333333 color.

CSS *ID* Selectors

ID selectors are styles that only occur once in a page, typically as a unique layout element that is not to be repeated.

You can assign any name you want to an ID selector when writing the rule, and the browser understands it as an ID because the selector will have a # symbol in front of it, as follows:

```
#globalNav { padding: 5%; }
```

This ID would be invoked in the HTML page as an attribute of the primary page navigation tag, <nav>, like this:

```
<nav id=”globalNav">…</nav>
```

CSS *Class* Selectors

CLASS selectors are styles that can be repeated as many times as necessary in a page, as in formatted text boxes, image styles, text formatting, etc.

You can assign any name you want to a class selector when writing the rule, and the browser understands it as a class because the selector will have a "." symbol in front of it, as follows:

```
.products { margin-left: 10px; color: #ffffff; }
```

This class could be invoked in the HTML page as an attribute of <section> like this:

```
<section class=”products">…</section>
```

CSS Multiple Classes

You can also define a single tag with multiple classes, as is the case with the following example, defining the text "Premium Water Bottle":

```
<p>Buy the <span class="product premium">Premium
Water Bottle</span>.</p>
```

Where the following two rules existed:

```
.product { color: #666666; }

.premium { background-color: yellow; }
```

CSS *Psuedos*

A psuedo-class/element is defined by a selector that gives specificity to rules based on a **condition**. For instance, maybe you only want the rule to apply when someone hovers over a link with a mouse. Here's the CSS syntax of the rules if you wanted normal links to be yellow, hover states to be orange, and previously visited links to appear as green:

```
a { color: yellow; }

a:hover { color: orange; }

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

Grouped Selectors

You can also **group selectors** that use the same declaration block to write more efficient css rules. Here's an example of when it would be appropriate:

Instead of...

```
h1 {color:#555555; }
h2 { color:#555555; }
h3 { color:#555555; }
```

You could write ...

```
h1, h2, h3 { color:#555555; }
```

Nested Selectors

You can also give your styles a greater level of control by nesting selectors in an ancestor-descendant relationship.

Nested Selectors

For instance, if you want link colors on a page to be green, except for inside a sidebar tag called <aside></aside> …where links need to be pink, it might look like this…

Nested Selectors

	NORMAL ‘a’	‘a’ INSIDE OF <aside>’ TAG
HTML VIEW	 LINK 	<aside> LINK </aside>

CSS VIEW	a {
	  color: lightgreen;
	  text-decoration: none;
	}
		aside a {
		  color: fuchsia;
		  text-decoration: underline;
		}

DISPLAY	LINK	LINK

CSS Properties

Long Format vs. Shorthand

CSS rules can be created and read in either a long or shorthand format. For many beginners, long format is easier to understand because it is more explicit, but it is definitely LONGER. Shorthand is more efficient but takes some getting used to.

Shorthand

```
p { margin: 1px 2px 3px 4px; }
```

Long

```
p {
margin-top: 1px;
margin-right: 2px;
margin-bottom: 3px;
margin-left: 4px;
}
```

Values: Units of Measure

The most basic distinction you need to know about units of measure is the difference between **absolute** unit values and **relative** unit values.

- **Absolute units** are ones that are fixed regardless of the environment the styles are in. An example of unchanging units are pixels (px).

- **Relative units** are ones that are relative to the environment or content around them. Examples of these are percentage (%), and ems (em).

Values: Units In Text and Positioning

Text Formatting and Units

It most common to assign text values in ems.

Layout Positioning and Units

When designing layouts of pages, it is useful to try to allow scalability of layout. Instead of using a pixel fixed width and/or height for layout areas, it can be preferable sometimes to use percentages so that the layout will take up a percentage of the window instead of a fixed size. This can also enhance design at times, too, when viewing the site on a high resolution monitor that would otherwise have a lot of dead space in the design.

Applying CSS to HTML

You have learned the basics of building a bare-bones html page and the basic structure of writing css rules.

Let's look at how the rules are applied to the html pages in the next slides.

Applying CSS to HTML

There are three main ways to apply css rules:

External CSS
Best choice for multi-page sites.

Embedded CSS
Less favorable option for websites with more than one page. Acceptable for a single HTML page.

Inline CSS
Rarely acceptable.

EXTERNAL CSS

External CSS

Separate CSS file

External css rules are written in a completely separate file from the HTML pages.

Can control multiple HTML files

External style sheets can control any number of HTML files.

Preferred Choice

It is the best choice to allow scalability in websites so that a designer can change styles in one place as opposed to several (potentially thousands of) files.

External CSS Example

The HTML files controlled by the external style sheet know to use the specified style sheet because of the <link /> in the <head>. For example:

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

EMBEDDED CSS

Embedded CSS

It is embedded in head of HTML

Embedded css uses the <style></style> **tags in the HEAD of an HTML document** to declare the css rules, and the rules are applied in that same page in the BODY.

Embedded CSS Example

Example of Embedded CSS:

```
3
4   <head>
5   <title>Untitled Document</title>
6   <style type="text/css">
7   p {
8       font-family: Arial, Helvetica, sans-serif;
9       font-size: 10em;
10      color: #333333;
11  }
12  </style>
13  </head>
```

Notice how the paragraph style (rule) is written in the document's HEAD. This style would be applied to all instances of in the same page only.

Embedded CSS

Embedded css is not typically preferable to external css because it lacks global site possibilities.

It is, however, good for single pages that need to be unencumbered by a folder structure.

Embedded CSS

Disadvantages:

1. It will override external css if you mistakenly create or leave embedded rules where it is unintended.
2. It is overridden by inline rules

Advantages:

1. Where intended, it can be useful in overriding global styles in external sheets.
2. It can make a single, styled HTML page portable.

INLINE CSS

Inline CSS

Inline CSS uses rules literally written within the line of HTML code. It employs the style attribute with declaration blocks as values. Example:

```
<p style="padding: 2px;">This is inline css.</p>
```

Inline CSS

Problems with inline CSS

- It is undesirable for most uses due to its inability to have global controls.
- It also overrides embedded and external styles, which makes global changes impossible on elements with inline css.