CSS FOUNDATIONS IN-DEPTH

The nitty-gritty of css . . .

What is CSS?

Cascading Style Sheets

Style sheets define formatting rules that are applied to text, images, forms, and embedded and layout elements. A style sheet is comprised of styling “rules,” which are applied to HTML to control visual presentation of pages. Each style sheet "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. If you have the following rule:

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

…then 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:

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

This ID would be invoked in the HTML page as an attribute of "section" like this:

```
<section id="content">Something Here</section>
```

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:

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

This class could be invoked in the HTML page as an attribute of "p" like this:

```
<p class="listing">Something Here</p>
```

CSS Multiple Classes

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

```
<p>Ron is in the <span class="spanish english">Bilingual Education</span> program.</p>
```

Where the following two rules existed:

```
.english { text-decoration: underline; }

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

CSS *Psuedos*

A psuedo-class/element is defined by a selector that gives specificity to rule 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 red, and previously visited links to appear as green:

```
a { color: yellow; }

a:hover { color: red; }

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...

```
p { color:#555555; }
```

```
.trailer
{ color:#555555; }
```

```
#header
{ color:#555555; }
```

You could write ...

```
p, .trailer, #header { color:#555555; }
```

Nested Selectors

You can also give your styles a greater level of control by nesting selectors in a parent-child relationship. What this means is that if you want only SOME instances of a class or html element to appear one way, but all other instances of it to appear another way, you can.

For instance, if you want link colors on a page to be green, except for in a class called "special" where links need to be pink, it would look like this:

normal link code

```
body a {
   color: green;
   text-decoration: none;
   }
```

special link code

```
.special a {
   color: #CC3366;
   text-decoration: none;
   }
```

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. If you use Dreamweaver, you can edit your Preferences to allow custom shorthand settings.

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

While there are seemingly limitless values you can assign to properties, we will only focus now on "units." Later we will discuss the range of other property values.

The two most basic distinctions you need to know as you are getting started in CSS 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 inches (in), centimeters (cm), and --for our purposes-- 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 areas so that it can accommodate expanded text if users enlarge it. Instead of using a pixel fixed width and/or height for layout areas, it is often preferable 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 actually applied to the html pages. There are three main ways to apply css rules

Inline CSS	• Rarely acceptable.
Embedded CSS	• Less favorable option for websites with more than one page. Acceptable for a single HTML page.
External CSS	• Best choice for multi-page sites.

Inline CSS

INLINE CSS

A rule literally written within the line of HTML code.

- This is inline css.

Deprecated

- It is deprecated due to its inability to have global controls, and it is considered a kind of lazy, quick solution.

Global changes are overridden by Inline CSS

- It also trumps embedded and external styles, which makes global changes impossible on elements with inline css.

Embedded CSS

EMBEDDED CSS

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.
Overridden by Inline, but overrides External	• Embedded css is overridden by inline css, but it will override external css.
Not deprecated, BUT….	• While it is not deprecated, embedded css is still not preferable to external css because it lacks global site possibilities.

Embedded CSS Example

Example of Embedded CSS:

```
1   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2   <html xmlns="http://www.w3.org/1999/xhtml">
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.

External CSS

EXTERNAL CSS	
Separate CSS file	• External css rules are written in a completely separate file from the XHTML pages.
Can control multiple HTML files	• External style sheets can control any number of X/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 hundreds 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></link> in the <head>, as follows on line 6.

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5  <title>Chapter One Lecture Notes</title>
6  <link href="css/main.css" rel="stylesheet" type="text/css" />
7  </head>
```