

Week Two: CSS Anatomy

Types of Styles

Styles: are what control the *presentation* of your *structured* markup

- 1) **Inline Styles:** are added to a tag using the XHTML `style` attribute.

```
<p style="font-size: 25pt; font-weight:bold; font-style:italic; color:red;">By
adding inline CSS styling to this paragraph, we can override the default
styles.</p>
```

- an *inline style* only affects the tag to which it is attached
- *inline styles* are another way of adding presentational markup directly into your pages
- *inline styles* should be used only in special circumstances
- *inline styles* will override other styles

- 2) **Embedded Styles:** can be placed within the `<head>` tag of your document and need to be called out with the `<style>` tag.

```
<head>
<title>Inline Styles example</title>
<meta http-equiv="Content-type" content="text/html; charset=iso-8859-1" />
<meta http-equiv="Content-Language" content="en-us" />

<style type="text/css">
  p {font-variant: small-caps}
</style>

</head>
```

- *embedded styles* are limited to the page in which they are contained
- sometimes it is easier to write the styles as *embedded styles* before you move them to be a linked style
- *embedded styles* will override linked styles

- 3) **Linked Styles:** are placed in a separate document that links to multiple pages as to control your styles globally

```
<head>
<title>Premium Design Works - WEB120 &ndash; Web Authoring II</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

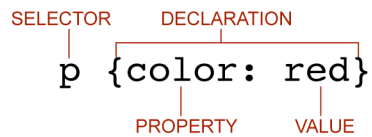
<link href="WEB120.css" rel="stylesheet" type="text/css" />

</head>
```

- *linked styles* are linked to the pages via a `<link>` tag within the `<head>` tag
- *linked styles* are applied to each page's markup as the page loads
- *linked styles* are contained in a text file, called a style sheet, and appended with a `.css`

Style Rules

Style Rules: are *how* your styles *present* your markup



- 1) **Selector:** states which tag the rule selects
- 2) **Declaration:** the declaration states what happens when the rule is applied **and** is made up of *properties* and *values*
- 3) **Property:** states what is to be affected, in this case
- 4) **Value:** states what the property will be set to

A) Words: a word such as bold is a type of *value*

```
font-weight: bold;
```

B) Numbers: numerical *values* are usually followed by a unit type

```
font-size: 12px;
```

C) Colors: color *values* are most commonly written as hexadecimal values

```
color: #336699;
```

- 5) **Multiple Declarations:** rules can have more than one declaration

```
p {
  font-family: Arial, Helvetica, sans-serif;
  color: #85898A;
  font-size: 12px;
  line-height: 18px;
  letter-spacing: 0px;
  text-align: left;
  margin-top: 0px;
  margin-bottom: 8px;
}
```

- 6) **Multiple Selectors:** can be grouped for a single rule

```
h1, h2, h3 {
  font-family: Arial, Helvetica, sans-serif;
  color: #85898A;
  font-size: 12px;
  font-weight: bold;
  line-height: 18px;
  letter-spacing: 0px;
  text-align: left;
}
```

- in this case `h1`, `h2`, `h3` have all be defined with the same font & color by using *multiple selectors*

7) **Multiple Rules:** can be applied to the same selector

```
h1 {
  margin-top: 10px;
  margin-bottom: 8px;
}

h2 {
  margin-top: 5px;
  margin-bottom: 8px;
}

h3 {
  margin-top: 3px;
  margin-bottom: 8px;
}
```

- however, let's say that you want to give `h1`, `h2`, `h3` different margin values
- you would create *multiple rules* to do so

8) **Contextual Selectors:** are selectors that use more than one tag name in the selector

```
p em {
  color: #F20017;
}
```

```
<p>Our mission is to develop and present your brand to a wide range of clientele
via <em>logo design</em>, <em>marketing collateral</em>, <em>advertising</em> and
<em>dynamic publishing</em> to the world wide web.</p>
```

- with this *contextual selector*, I have made the `` tags within my `<p>` tags a different color and style

Inheritance

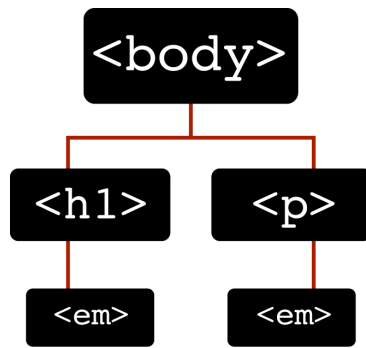
Inheritance: involves passing something down from ancestors to descendants

```
body {
  font-family: Arial, Helvetica, sans-serif;
  color: #85898A;
  background-color: #FFF;
  margin: 0px 0px 0px 0px;
  padding: 0px 0px 0px 0px;
}
```

- in this example we can see that the `<body>` tag, the top most *ancestor*, will *inherit* the properties listed in the rule

The Cascade

Cascading Rules: cascade down from one level of the hierarchy to the next



- understanding the *cascade* will help you write your *style sheets* in the most economically organized way
 - organizing your *style sheets* with the proper *cascade* will also enable them to be edited much more easily
- 1) **Matching Declarations:** as the page loads, the browser looks at every tag in the page to see if the rule *matches*
 - 2) **Sorting Order:** If a matched property is defined again, the browser will update the value in the *order* it is declared
 - 3) **Specificity:** the rule that is more specific wins

Assignment: CSS Document