

CSS: Cascading Style Sheets

- HTML (the original version) was used to describe the structure of a document.
 - paragraphs, lists, tables, etc.
- New tags and attributes were added (by browser vendors) to control appearance.
- People realized that this was not the right approach:
 - a better idea is to separate the document content from it's appearance.

Old vs. New

```
<h2> <font color="red" size="+5">  
    Here is a heading  
</font></h2>
```

```
<h2 style="color:red; font-size:+5pt;">  
    Here is a heading  
</h2>
```

Inline Style Setting

- Style properties are specified as the value of the `style` attribute.

```
style="color:blue"
```

```
style="font-family:fixed; font-size:14px;"
```

```
style="padding:.1in; border-color: black;"
```

- Requires us to learn yet another syntax...

Recap

- To be a Web-Wiz you need to know:
 - html **tag** names
 - html **attributes** that can be used with each tag
 - what kind of values each attribute can have.
 - style **properties** that can be used with each tag.
 - style property values for each type of property.
 - the entire DNA sequence for all known life forms.

Style syntax

- Each style property is defined like this:

property-name: *val1*, *val2*, ..., *valn*;

- *property-name* is something you need to look up.
 - not all properties can be applied to all tags.
- each property can have multiple values!?!
 - generally a list in order of preference.
- The value (or list of value) ends with ';'.

Some types of properties

- font: controls font-family and style
- text: how paragraphs are rendered
- background: colors, images, etc.
- box: spacing, borders, margins, etc.
- positioning: alignment, placement, etc.

property types

- There are a number of different kinds of values that are specified using style settings.
- You have to know the syntax for each data type, or the possible values.
- Many properties can take on a few different values only (you have to look up what they are).
 - for example: `font-style` can be “normal”, “italic” or “oblique”

Specifying Colors

- There are a number of properties that take on color values:

`color` `background-color` `border-color`

- You can use common color names:

`"white"` `"black"` `"red"` `"blue"` ...

- You can use a 6 digit hexadecimal number prefixed by #: `"#rrggbb"`.

`"#808080"` (gray)

`"#ff00ff"` (purple)

`"#ff0000"` (red)

`"#ffffff"` (white)

Specifying length/size

- relevant properties include `font-size`, `border-width`, `padding`, `margin`.
- Generally a number followed by the units.
 - `px` pixels
 - `in` inches
 - `pt` point size

```
<h2 style="font-size: 12pt;">
```

```
<p style="border-width: .1in;">
```

URL properties

- Some properties have a value that is a URL.
 - for example: `background-image`
- You must use the `url` keyword followed by the actual URL in parentheses:

```
<h2 style="background-image:url(http://www.images.com/bricks.gif);">
```

Some font properties and examples

- color - used to specify text color.

```
color:red;
```

```
color:#a132a9
```

- font-family

```
font-family:Courier, fixed;
```

- font-size

```
font-size:12pt;
```

```
font-size:14px;
```

- font-weight

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

```
font-weight:lighter;
```

font properties example

```
<p style="font-family:arial,sans-serif;  
        font-size:12px;  
        font-weight: bold;  
        color:blue;">
```

This is my paragraph. I like arial, it's a nice clean font.

```
<span style="font-size:10px;">Here is a small  
sentence</span>
```

```
I use bold because this paragraph is  
important. Blue is my favorite color.</p>
```

Some text properties

- `text-align`: where to align text
 - “left”, “right”, “center”
- `letter-spacing`: space between characters
 - value is a length, like “2px” or “.01in”
- `line-height`: space between lines.
 - value is a length.

Text properties example

```
<p style="text-align:center;">
```

```
<span style="letter-spacing:10pt;">
```

```
WIDER
```

```
</span>
```

```
is better.</p>
```

W I D E R

is better

Box properties

- Each HTML element is contained within an imaginary box that surrounds the elements contents. You can modify how the browser renders this box.
- `border`, `border-color`, `border-width`
- `margin`, `padding`

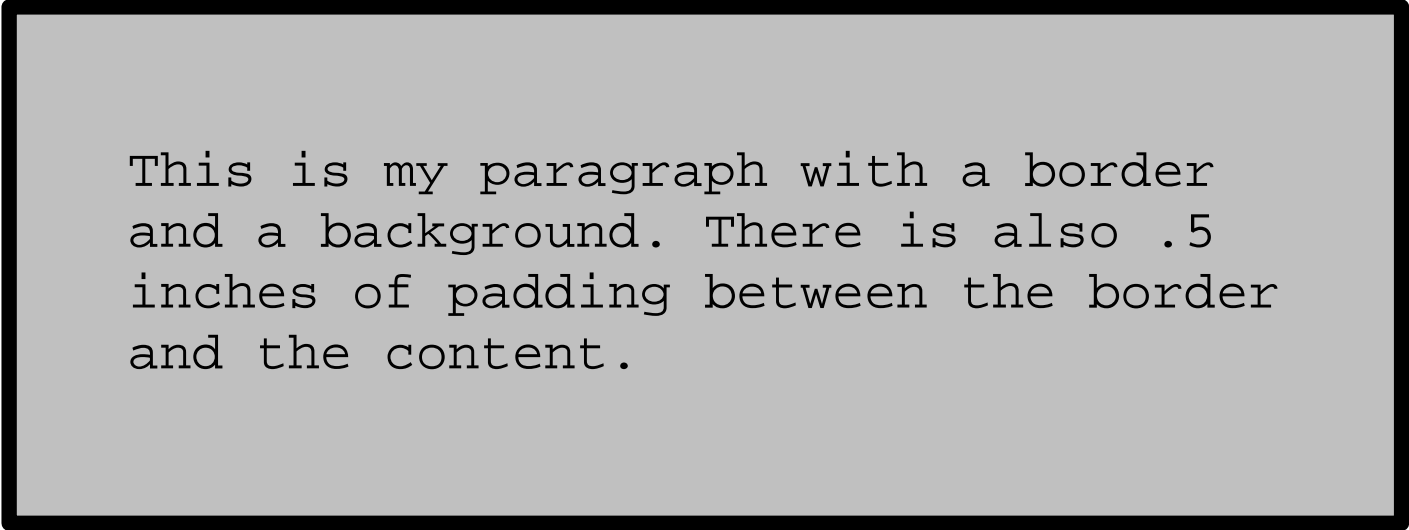
Box properties example

```
<p style="border: solid blue 5px;
```

```
background-color: gray;">
```

This is my paragraph with a border and a background. There is also .5 inches of padding between the border and the content.

```
</p>
```



This is my paragraph with a border and a background. There is also .5 inches of padding between the border and the content.

Positioning

- CSS can be used to position HTML elements.
 - `position` property can be:
 - `static`: normal positioning
 - `relative`: can adjust the position relative to where it would normally be placed.
 - `absolute`: can specify the exact location
- If `position` is not `static`, you can set:
 - `left`: where the left edge of the element is placed.
 - `top`: where the top edge of the element is placed.

Positioning Examples

```
<h2 style="position:absolute;
        top: 100px; left: 100px;">
```

```
I am always at 100,100</h2>
```

```
<p style="position:relative;
        top: -20px">I am 20 pixels higher than
you would expect</p>
```

Other neat-o stuff

- `display: none or block (and others)`
- `float: left or right`
- `visibility: visible or hidden or collapse.`

Document-wide Style Settings

- Inline style settings are fine, but the power of CSS comes from the ability to define style settings that apply to an entire document.
 - as we will see later, you can also apply the same settings to many documents
- You can override document-wide style setting where ever you want by using inline style.
- Document-wide settings can be defined in a style section of a document, or in an external file.

The `style` HTML tag

- You can include a style section in the head of an HTML document:

```
<html>
<head>
<title>blah</title>
<style>
... style rules ...
</style>
<body>
```

...

Style Rules

- Within a style section, you define rules that apply to specific tags within a document.
- You could create a rule that says “all h2 tags should use arial, 14pt in red”:

```
h2 { font-family:arial;  
      font-size: 14pt;  
      color: red  
    }
```

More Rules

- This rule changes the font for h1,h2 and h3 tags:

```
h1,h2,h3 { font-family:fixed; }
```

- This rule changes all li tags to have red text:

```
li { color:red; }
```

- This one changes only li tags found inside an ordered list (OL) to have green text:

```
ol li { color:green; }
```

Class Rules

- You associate a *class* name with any tag:

```
<tagname class="something">
```

- You can create CSS rules that apply on to those tags with a particular class name:

```
tagname.class { CSS rules... }
```

- For example, to set the font size for all paragraphs with class “small”:

```
p.small { font-size: 8px; }
```

More class rules

- You can also define class rules that can be used with any tag:

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

- Note that you can only assign one class attribute to a specific tag, so you can't do this:

```
<p class="small" class="red">blah</p>
```

Pseudo-class Rules

- You can also define rules for some predefined pseudo-classes. For example, the `a` (anchor) tag has pseudo classes `visited`, `hover`, `link` and `active`.

<code>a:link</code>	<code>{color: blue}</code>	<i>never used</i>
<code>a:visited</code>	<code>{color: purple ;}</code>	<i>already used</i>
<code>a:hover</code>	<code>{font-weight:bold;}</code>	<i>mouse is over</i>
<code>a:active</code>	<code>{color: red}</code>	<i>clicked</i>

ID rules

- You can create rules that match any tag that has a specific id (as in ID=fred)

```
#red {color:red;}
```

```
<p id="red">blah blah</p>
```

Rule order

- The order of rules has significance.
 - The topmost rule is applied first, then lower rules.
- The last rule has the highest precedence!
- Strategy:
 - general rules defined first
 - more and more specific rules defined later

Example Document

```
<html>
<head>
<title>Sample</title>
<style>
body { background-color:gray; }
p { font-family:TimesRoman;
    font-size:12px;
    }
ul { list-style-type: square; }
h1,h2,h3 { color: blue}
</style>
<body>
<h1>hello</h1>
<p>This is a page that includes style...
```

Other Samples

- Also available via the course home page:
 - page with no style settings: [foo.html](#)
 - The same document, now with some rules to change the normal font used, background color and text color: [foo1.html](#)
 - Another version of the same document: [foo2.html](#)
 - and yet another version of the same document: [foo3.html](#)
- Note: to see source of a document in your browser, click on “view source” (in menubar).

External Style Sheets

- Rather than putting a style section in each HTML document, you can create a “style sheet” in a file.
- The file just contains CSS rules (no tags!)
- Typically the file name ends in “.css”.

You can associate a style file with a document by using the `link` tag within the HTML document head.

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

Example

```
<html>
<head>
<title>Sample</title>
<link rel=stylesheet type="text/css" href="foo.css">

<body>
<h1>hello</h1>
<p>This is a page that includes style.</p>
<p>blah blah blah</p>
</body>
</html>
```

foo.css

```
body { background-color:gray; }
p { font-family:TimesRoman;
    font-size:12px;
  }
ul { list-style-type: square; }
h1,h2,h3 { color: blue}
```

csszengarden.com

- awesome examples of using CSS.
 - one HTML document.
 - tons of different style sheets, resulting in surprisingly very different appearing pages.
 - many of the style sheets make use of CSS positioning properties.
 - background images.

Positioning Basics

- `position: absolute`
 - also need to set the `top` and `left` properties.
 - element has a fixed position in the document, and is not part of the *normal* document flow.
 - it can cover up other things, or be covered by other things!
 - `z-index` property allows you to control this.

Example: [absolute.html](#)

position: fixed

- Need to set the position using `top`, `left` (and/or `bottom`, `right`).
- Position of the element is fixed within the window (not the document).
- Example: [fixed.html](#)
 - link to course home page is always in top right of the browser window (try scrolling down to see what happens).
 - IE may not support fixed positioning correctly (didn't the last time I tried).

position: relative

- Any changes made to the position using `top`, `left` (and/or `bottom`, `right`) are *relative* to the element's normal position in the document flow.
- Example: [relative.html](#)

Using float

- You can take an element out of the normal document flow and force it to the left or right (of the containing element, or browser window if there is no containing element).
- Other elements flow around the floated element.
- float is used heavily on csszengarden...
- Example: [float.html](#)

3 Column Layout

- Popular format for web pages
 - left and right sections hold links and ads.
 - both sections set to float.
 - both need fixed width
 - middle section holds content.
 - middle section shrinks/grows with window size.

Example: [3col.html](#)

Lots of different approaches and examples