

Lecture 3: CSS Intro

First, some review:

- Space, tab, and return are called whitespace characters.
- Any number or combination of whitespace characters is treated as a single space.
- If we want to add whitespace, we need to tell the browser to add it.

First, some review:

- The `
` tag starts a new line.
- You do not need a closing `
` tag.
- You do not need to start a new line in your html document for the `
` tag to work.
- You can use multiple `
` tags to skip multiple lines.

First, some review:

- The `<hr>` skips to lines, and puts a horizontal line in the gap.
- You do not need a closing `<hr>` tag.

First, some review:

- The `<p>` tag is used for formatting paragraphs.
- You need a closing `</p>` tag.
- We can format the content of the `<p>` tag with the `style` attribute.
- For example, to indent the first line of a paragraph:

```
<p style="text-indent: 10px"> ... </p>
```

First, some review:

- The `<div>` tag creates a page division.
- You need a closing `</div>` tag.
- You can also use the `<div>` tag to create paragraphs.

First, some review:

- The `<div>` and `` tag are block elements.
- The `` tag is an inline element.
- A block fills a rectangular region of the page, and pushes other content out of the way.
- An inline element fits into the flow of text, and does not effect positioning.

One last concept in html

- There many different versions of html.
- We are using version 4.01 strict.
- We need to tell the browser what version we are using.
- This is done with a *document type definition* (DTD).

One last concept in html

The document type definition for html 4.01 strict looks like this:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

You can copy and paste this off the course webpage!

twiki-edlab.cs.umass.edu/bin/view/FosterCS391F/WebHome

One last concept in html

From now on, you should add the document type definition to every page you make.

- The DTD goes **before** the opening html tag:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML  
4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">  
<html>  
  <head> ...
```

- Please don't try to memorize this. You can look it up when you need it.

Now on to CSS

- CSS stands for “Cascading Style Sheets”
- CSS allows you to separate style information from content.
- With CSS allows you to create many effects that you can't achieve with plain html. Let's have a look at some...

Stylesheets

From now on you will be working with two files.

- Your html file contains your Web page's content.
- Your CSS file contains the style information.

Stylesheets

The first step is to create a second file in notepad.

- Save it as “stylesheet.css” (or some other name, as long as it ends in .css)
- Make sure you save it in the same folder as your html file.

Stylesheets

Now we need to tell your html file where to find the css file. Otherwise, your html file won't know that the css file exists.

How do you suppose we do that?

Stylesheets

Now we need to tell your html file where to find the css file. Otherwise, your html file won't know that the css file exists.

How do you suppose we do that?

With a tag!

The <link> tag

- The link tag is used to link information in another file to your page.
- The link tag goes in the head.
- The link tag does not need a closing tag.
- Do not get this confused with the <a> tag, which creates a hyperlink!

The <link> tag

We will use the link tag to link the css file to our html file.

First, add the link tag to the head section of your page:

```
<html>  
  <head>  
    <title>CSS Intro</title>  
    <link>  
  </head>  
  ...
```

Attributes

The link tag requires 3 attributes.

- One tells the browser *what* we are linking.
- One tells the browser the *type* of information that we are linking.
- One tells the browser the *file name* of the linked file.

The rel attribute

The rel attribute tells the browser what we are linking to the html file.

- We are linking a stylesheet.
- “stylesheet” is the attribute's value:

```
<link rel="stylesheet">
```

The type attribute

The type attribute tells the browser what type of information we are linking to the html file.

- The type of information is css.
- “text/css” is the attribute's value:

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

The href attribute

The href attribute tells the browser the file name of the file we are linking.

- We chose stylesheet.css as a file name.
- “text/css” is the attribute's value:

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

The complete <link> tag:

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

That's it! Your stylesheet is linked to your html file.

Editing the css file

Right now the css file is empty, so it has no effect on your web page. Let's change that.

Each element of your css file has 2 parts.

- A selector.
- A list of attributes and values.
- css attributes and html attributes are not the same thing!

Selectors

- The contents of the css file change the way html tags are formatted.
- Selectors choose what html tags we are modifying.

Style Attributes

- Style attributes determine how html tags are modified.
- Don't get these confused with html attributes. They're not the same thing!

CSS Syntax

Here is the basic syntax:

```
selector  
{  
  attribute 1: value 1;  
  attribute 2: value 2;  
  ...  
  attribute n: value n;  
}
```

CSS Syntax

Let's look at an example:

```
body
{
  background: beige;
  color: blue;
}
```

- The selector here is “body”.
- The attributes are background and color.

CSS Syntax

Let's look at an example:

```
body
{
  background: beige;
  color: blue;
}
```

- The background attribute sets the background color.
- The color attribute sets the text color.

Selecting Tags by Name

- The simplest way to select a tag is by its name.
- That is what we did on the previous slide.
- All tags of that type will be selected.

Selecting Tags by Name

Example:

```
body
{
    background: beige;
    color: blue;
}
```

- This makes the body's background color beige, and the text color blue.
- There is only one body tag, so only one tag is selected.

Selecting Tags by Name

Example:

```
span
{
    background: beige;
    color: blue;
}
```

Gives *every* span tag a beige background, and blue text. Let's try this. Add the above text to your css file.

Selecting Tags by Name

Now add the following to the body of your html file:

```
<span>Welcome</span>
```

to

```
<span>my</span>
```

web

```
<span>page!</span>
```

Save both files, and see what happens!

Selecting Tags by Name

Example:

```
p  
{  
  text-indent: 10px;  
}
```

- The selector `p` selects all `<p>` tags.
- The `text-indent` attribute causes the first line to be indented.

Selecting Tags by Name

Example:

```
p
{
    text-indent: 10px;
}
```

- This is a convenient way to indent the first line of every paragraph.
- Before, we had to add the style attribute to every opening `<p>` tag to do this!

Selecting Tags by Name

Example:

```
div
{
  display: inline;
}
```

- Here, all div tags are selected.
- Recall, div tags are block elements by default.

Selecting Tags by Name

Example:

```
div
{
  display: inline;
}
```

- We can change the way elements display with display css attribute.
- Here we've made all div tags behave like span tags!

Selecting Tags by Name

Example:

```
a
{
  text-decoration: none;
}
```

- Here, all anchor tags are selected.
- Remember, anchor tags are underlined by default.

Selecting Tags by Name

Example:

```
a
{
  text-decoration: none;
}
```

- The text-decoration can add or remove underlines, etc.
- We've removed the underline from all hyperlinks!

Practice!

It's time for your first in-class assignment.