

LECTURE 03: CASCADING STYLE SHEETS (CSS)

Web Programming



INSTRUCTOR: DR. HOSSAM ZAWBAA

CREATING TABLE: <TABLE>

- In this section, you'll create the basic layout for your website with a table.
- Table with one row and one column

HTML

```
<table border="1">  
  <tr>  
    <td><p>First Column</p></td>  
  </tr>  
</table>
```

Output

First Column

CREATING TABLE: <TABLE>

- Table with one row and two columns

HTML

```
<table border="1">  
  <tr>  
    <td><p>First Column</p></td>  
    <td><p>Second Column</p></td>  
  </tr>  
</table>
```

Output

First Column	Second Column
--------------	---------------

CREATING TABLE: <TABLE>

- Table with two rows , each row has two columns

HTML

```
<table border="1">  
  <tr>  
    <td><p>First Column</p></td>  
    <td><p>Second Column</p></td>  
  </tr>  
  
  <tr>  
    <td><p>First Column</p></td>  
    <td><p>Second Column</p></td>  
  </tr>  
</table>
```

Output

First Column	Second Column
First Column	Second Column

STYLING

- The bad way to produce styles
- tags such as `b`, `i`, `u`, and `font` are discouraged in strict XHTML.

HTML

```
<p>  
  <font face="Arial">Welcome to Greasy Joe's.</font>  
  You will <b>never</b>, <i>ever</i>, <u>EVER</u> beat  
  <font size="+4" color="red">OUR</font> prices!  
</p>
```

Output

Welcome to Greasy Joe's. You will **never**, *ever*, EVER beat **OUR** prices!

CASCADING STYLE SHEETS (CSS)

- **describe the appearance, layout, and presentation of information on a web page**
 - (as opposed to HTML, which describes the content of the page)
- describe *how information is to be displayed, not what is being displayed*
- can be embedded in HTML document or placed into separate **.css** file
 - with a **.css** file, **one style sheet can be shared across many pages**

CSS: BRIEF HISTORY

- 1997-2001
- Content: **HTML 4.01**
- Presentation: **CSS1**

- 2001-2007
- Content: **XHTML 1**
- Presentation: **CSS2**

- 2008-present
- Content: **HTML5**
- Presentation: **CSS3**

BASIC CSS RULE SYNTAX

- a CSS file consists of one or more rules
- **each rule starts with a selector that specifies an HTML element(s) and then applies style properties to them**
 - a selector of * selects all elements

```
selector {  
    property: value;  
    property: value;  
    ...  
    property: value;  
}
```

Example

```
p {  
    font-family: sans-serif;  
    color: red;  
}
```


EMBEDDING STYLESHEETS : <STYLE>

- CSS code can be embedded within the head of an HTML page
- this is *bad style and should be avoided when possible*

CSS

```
<head>  
  <style type="text/css">  
    p { font-family: sans-serif; color: red; }  
    h2 { background-color: yellow; }  
  </style>  
</head>
```

ATTACHING A CSS FILE: <LINK>

- a page can link to multiple style sheet files
 - in case of a conflict (two sheets define a style for the same HTML element), the latter sheet's properties will be used

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

Example

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

<link href="http://www.google.com/uds/css/gsearch.css"
rel="stylesheet" type="text/css" />
```

INLINE STYLES: THE STYLE ATTRIBUTE

- higher precedence than embedded or linked styles
- used for one-time overrides and styling a particular element
- this is *bad style and should be avoided when possible*

CSS

```
<p style="font-family: sans-serif; color: red;">  
This is a paragraph</p>
```

Output

This is a paragraph

BASIC CSS RULE SYNTAX

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>CSS:The element selector</title>
  <style>
    body {
      background-color: blue;
    }

    h1 {
      color: red;
    }

    p {
      color:yellow;
    }
  </style>
</head>
<body>
  <h1>Learn HTML </h1>
  <p>You can learn about headers and links and images and SO MUCH MORE!
</p>

  <p>Plus, you can learn CSS to make your page stylish!</p>
</body>
</html>
```

Learn HTML

You can learn about headers and links and images and SO MUCH MORE!

Plus, you can learn CSS to make your page stylish!

FAVORITES ICON ("FAVICON")

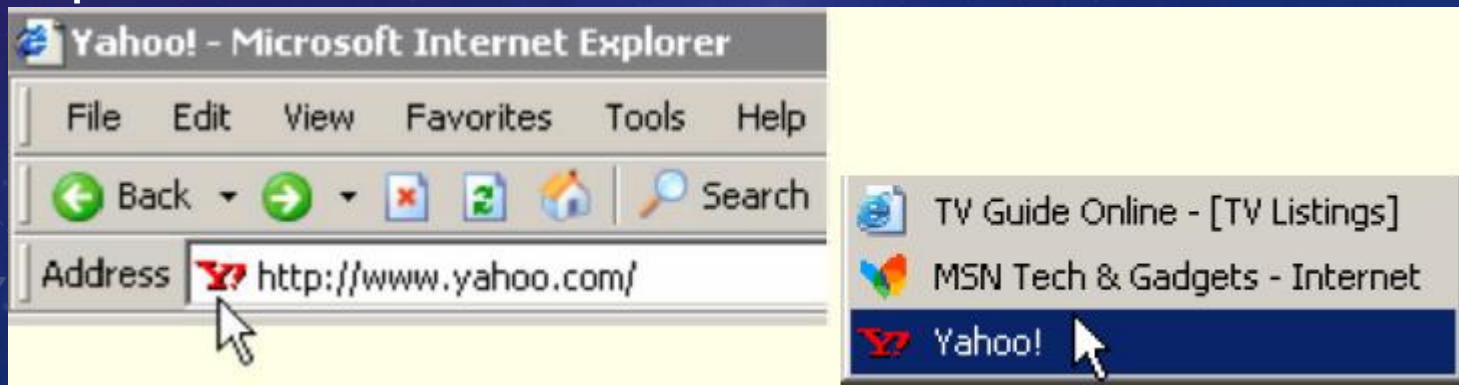
- the link tag, placed in the HTML page's head section, can specify an icon for a web page
 - this icon **will be placed in the browser title bar and bookmark/favorite**
- IE6: this doesn't work; must place a file in **.ico** format named favicon.ico in the root directory of the web server

```
<link href="filename" type="MIME type" rel="shortcut icon" />
```

Example

```
<link href="yahoo.gif" type="image/gif" rel="shortcut icon" />
```

Output



CSS PROPERTIES FOR COLORS

property	description
<code>color</code>	color of the element's text
<code>background-color</code>	color that will appear behind the element

CSS

```
p {  
  color: red;  
  background-color: yellow;  
}
```

Output

This paragraph uses the style above.

SPECIFYING COLORS

- color names: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, (white), yellow
- RGB codes: red, green, and blue values from 0 (none) to 255 (full)
- hex codes: RGB values in base-16 from 00 (0, none) to FF (255, full)

CSS

```
p { color: red; }  
h2 { color: rgb(128, 0, 196); }  
h4 { color: #FF8800; }
```

Output

This paragraph uses the first style above.

This h2 uses the second style above.

This h4 uses the third style above.

GROUPING STYLES

- a style can select multiple elements separated by commas
- the individual elements can also have their own styles (like h2)

CSS

```
p, h1, h2 {  
  color: green;  
}  
h2 {  
  background-color: yellow;  
}
```

Output

This paragraph uses the above style.

This h2 uses the above styles.

CSS COMMENTS: /* ... */

- CSS (like HTML) is usually not commented as rigorously as programming languages such as
- Java
- the // single-line comment style is NOT supported in CSS
- the <!-- ... --> HTML comment style is also NOT supported in CSS

CSS

```
/* This is a comment.  
   It can span many lines in the CSS file. */  
p {  
  color: red; background-color: aqua;  
}
```

CSS PROPERTIES FOR FONTS

property	description
<code>font-family</code>	which font will be used
<code>font-size</code>	how large the letters will be drawn
<code>font-style</code>	used to enable/disable italic style
<code>font-weight</code>	used to enable/disable bold style

FONT-FAMILY

- enclose multi-word font names in quotes

CSS

```
p {  
  font-family: Georgia;  
}  
h2 {  
  font-family: "Courier New";  
}
```

Output

This paragraph uses the first style above.

This h2 uses the second style above.

MORE ABOUT FONT-FAMILY

- can specify multiple fonts from highest to lowest priority
- **generic font names:**

`serif, sans-serif, cursive, fantasy, monospace`

if the first font is not found on the user's computer, the next is tried

- generally should specify similar fonts
- placing a generic font name at the end of your font-family value ensures that every computer will use a valid font

CSS

```
p {  
  font-family: Garamond, "Times New Roman", serif;  
}
```

Output

This paragraph uses the above style.

FONT-SIZE

- units: pixels (px) vs. point (pt) vs. m-size (em) - 16px, 16pt, 1.16em
- vague font sizes: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
- percentage font sizes, e.g.: 90%, 120%
- pt specifies number of *point*, where a point is $1/72$ of an inch onscreen
- px specifies a number of pixels on the screen
- em specifies number of *m-widths*, where 1 em is equal to the font's *current size*

CSS

```
p {  
  font-size: 14pt;  
}
```

Output

This paragraph uses the style above.

FONT-SIZE

1. **“Ems” (em):** The “em” is a scalable unit that is used in web document media. An em is equal to the current font-size, for instance, if the font-size of the document is 12pt, 1em is equal to 12pt. Ems are scalable in nature, so 2em would equal 24pt, .5em would equal 6pt, etc. Ems are becoming increasingly popular in web documents due to scalability and their mobile-device-friendly nature.
2. **Pixels (px):** Pixels are fixed-size units that are used in screen media (i.e. to be read on the computer screen). One pixel is equal to one dot on the computer screen (the smallest division of your screen’s resolution). Many web designers use pixel units in web documents in order to produce a pixel-perfect representation of their site as it is rendered in the browser. One problem with the pixel unit is that it does not scale upward for visually-impaired readers or downward to fit mobile devices.
3. **Points (pt):** Points are traditionally used in print media (anything that is to be printed on paper, etc.). One point is equal to 1/72 of an inch. Points are much like pixels, in that they are fixed-size units and cannot scale in size.
4. **Percent (%):** The percent unit is much like the “em” unit, save for a few fundamental differences. First and foremost, the current font-size is equal to 100% (i.e. 12pt = 100%). While using the percent unit, your text remains fully scalable for mobile devices and for accessibility.

FONT-WEIGHT, FONT-STYLE

CSS

```
p {  
  font-weight: bold;  
  font-style: italic;  
}
```

Output

This paragraph uses the style above.

CSS PROPERTIES FOR TEXT

property	description
<code>text-align</code>	alignment of text within its element
<code>text-decoration</code>	decorations such as underlining
<code>line-height</code> , <code>word-spacing</code> , <code>letter-spacing</code>	gaps between the various portions of the text
<code>text-indent</code>	indents the first letter of each paragraph

TEXT-ALIGN

- text-align can be left, right, center, or **justify** (which widens all full lines of the element so that they occupy its entire width)

CSS

```
blockquote { text-align: justify; }  
h2 { text-align: center; }
```

Output

The Emperor's Quote

[TO LUKE SKYWALKER] The alliance... will die. As will your friends. Good, I can feel your anger. I am unarmed. Take your weapon. Strike me down with all of your hatred and your journey towards the dark side will be complete.

TEXT-DECORATION

CSS

```
p {  
  text-decoration: underline;  
}
```

Output

This paragraph uses the style above.

can also be overline, ~~line-through~~, blink, or none

- effects can be combined:

```
text-decoration: overline underline;
```

Output

Test font and line styles

THE LIST-STYLE-TYPE PROPERTY

CSS

```
ol { list-style-type: lower-roman; }
```

■ Possible values:

- i. none : No marker
- ii. disc (default), circle, square
- iii. decimal : 1, 2, 3, etc.
- iv. decimal-leading-zero : 01, 02, 03, etc.
- v. lower-roman : i, ii, iii, iv, v, etc.
- vi. upper-roman : I, II, III, IV, V, etc.
- vii. lower-alpha : a, b, c, d, e, etc.
- viii. upper-alpha : A, B, C, D, E, etc.
- ix. lower-greek : alpha, beta, gamma, etc.
- x. others: hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha

BODY STYLES

- to apply a style to the entire body of your page, **write a selector for the body element**
- saves you from manually applying a style to each element

CSS

```
body {  
  font-size: 16px;  
}
```

CASCADING STYLE SHEETS ORDER

- browser's default styles
- external style sheet files (in a `<link>` tag)
- internal style sheets (inside a `<style>` tag in the page's header)
- inline style (the `style` attribute of the HTML element)

INHERITING STYLES

- when multiple styles apply to an element, they are **inherited**
- The **more tightly matching rule can override the more general inherited rule**

CSS

```
body { font-family: sans-serif; background-color: yellow; }  
p { color: red; background-color: aqua; }  
a { text-decoration: overline underline; }  
h2 { font-weight: bold; text-align: center; }
```

Output

This is a heading.

A styled paragraph. Previous slides are available on the web site.

- a bulleted list

STYLES THAT CONFLICT

- when two styles set conflicting values for the same property, **the latter style takes precedence**

CSS

```
p, h1, h2 { color: blue; font-style: italic; }  
h2 { color: red; background-color: yellow; }
```

Output

This paragraph uses the first style above.

This heading uses both styles above.

THE HTML ID ATTRIBUTE

- allows you to give a unique ID to any element on a page
- each ID must be unique; can only be used once in the page

HTML

```
<p>Spatula City!  Spatula City!</p>  
<p id="mission">Our mission is to provide the most  
spectacular spatulas and splurge on our specials until our  
customers <q>esplode</q> with splendor!</p>
```

Output

Spatula City! Spatula City!

Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers “esplode” with splendor!

LINKING TO SECTIONS OF A WEB PAGE

- a link target can include an ID at the end, preceded by a #
- browser will load that page and scroll to element with given ID

HTML

```
<p>Visit <a href="http://www.textpad.com/download/index.html#downloads">
textpad.com</a> to get the TextPad editor.</p>

<p><a href="#mission">View our Mission Statement</a></p>
```

Output

Visit [textpad.com](http://www.textpad.com/download/index.html#downloads) to get the TextPad editor.

[View our Mission Statement](#)

CSS ID SELECTORS

CSS

```
#mission {  
  font-style: italic;  
  font-family: "Garamond", "Century Gothic", serif;  
}
```

- applies style only to the paragraph that has the ID of `mission`
- element can be specified explicitly: `p#mission` {

CSS .CLASS SELECTOR

CSS Syntax

```
.class {  
    css declarations;  
}
```

CSS

```
p.hometown {  
    background-color: yellow;  
}
```

Style all <p> elements with class="hometown":

CSS .CLASS SELECTOR

HTML

```
<!DOCTYPE html>
<html>
<head>
<style>
p.hometown {
    background: yellow;
}
</style>
</head>
<body>

<p>My name is Samir.</p>
<p class="hometown">I live in Cairo.</p>

<p>My name is Waleed.</p>
<p class="hometown">I also live in Beni Suef.</p>
|
</body>
</html>
```

Output

My name is Samir.

I live in Cairo.

My name is Waleed.

I also live in Beni Suef.