

Chapter 7: Working With Cascading Style Sheets

- 7:1
- Learn about the history and theory of cascading style sheets.
 - Learn how to create inline styles, embedded styles, and external style sheets.
 - Study how styles are applied to a wide range of tags in your Web site.
 - Learn how the styles are cascaded through the structure of nested tags on your Web pages.

I. INTRODUCTION TO CASCADING STYLE SHEETS

-Do **Assignment 7:1**

- HTML and Page Layout
 - Tables have proven to be the most popular way of defining page layouts.
- History and Support of CSS
 - One principle of design theory is to separate, as much as possible, the content of a document from the design.
 - Because tags are intertwined with content if a change needs to be made every tag must be located. This can be a long daunting task.
 - One way around this problem is to create a **style** defining the appearance of a document element.
 - The collection of styles for a Web page or Web site is known as a **style sheet**.
 - Style sheets use a common language and syntax. The main style sheet standards are **Cascading Style Sheets (CSS)**.
 - The World Wide Web Consortium, the same organization that develops specifications for HTML, has developed CSS. The first CSS standard, **CSS1**, was released in 1996, and a second standard, **CSS2**, was released in 1998.
- Style Types
 - There are three ways of employing CSS in your Web pages:
 - Inline styles** in which styles are added to each tag within the HTML file. The style affects that particular tag but does not affect other tags in the document.
 - Embedded or global styles** applied to an entire HTML file, allowing the Web designer to modify the appearance of any tag in the document.
 - Linked or external style sheets** placed in an external file and linked with pages in the Web site, allowing the Web designer to modify the appearance of tags in several documents.

II. USING INLINE

- To create an inline style, you add the style attribute to the HTML tag using the following syntax: `<tag style="style declarations">`
where tag is the name of the tag and style declarations are the styles you'll define for that particular tag. Note: the style declaration must be enclosed within double quotation marks.
- A style declaration consists of attribute names that specify such features as the font size, color, and font type, followed by a colon and then the value of the attribute. Multiple attributes can be used as long as you separate each one by a semicolon.
- The general syntax for the style declaration is therefore:
`attribute1:value1;attribute2:value2;...`
where attribute1 is the name of a particular style attribute, and value1 is the value the style applies to that attribute
- Do **Assignment 7:2**

III. CREATING AN EMBEDDED STYLE

- Within the head section of your HTML file, insert the following HTML code:

```
<style type="style sheet language">
  style declarations
</style>
```

where style sheet language is the language of your cascading style sheets. If no language is specified, the default value "text/css" is used.

- To enter a style declaration, use the following syntax:

```
selector {attribute1:value1;attribute2:value2;...}
```

where selector identifies an element in your document, attribute1 is the first style attribute and value1 is the value assigned to the first attribute, and so forth.

- Selectors and Declarations

- Style declarations within the `<style>` tags obey the following syntax:

```
selector {attribute1:value1;attribute2:value2;...}
```

where selector identifies an element in your document, such as a heading or paragraph, and the attributes and values within the curly braces indicate the styles applied to all the occurrences of the element. This collection of attributes and values is also referred to as the declaration of the selector.

- ii. Example:


```
<style>
  h1 {color:gold; font-family:sans-serif}
</style>
```

 In this example, “h1” is the selector and the text enclosed in the braces is the declaration.
 - iii. Note: the type attribute was not included within the <style> tag. This is because “text/css” is the default style language, and unless you specify a different style language, you don’t need to enter the type attribute.
 - iv. Since you are using the <style> tags, you don’t need to include double quotes around the attributes and attribute values as you did for inline styles.
 - v. Do [Assignment 7:3](#)
- b. Grouping Selectors
- i. You can apply the same declarations to a group of selectors by including all of the selector names separated by commas.
 - ii. Do [Assignment 7:4](#)

IV. USING AN EXTERNAL STYLE SHEET

- Do [Assignment 7:5](#)
- a. Linking to Style Sheets with the <link> Tag
 - Create a text file containing the style definitions that you want to apply to the pages on the Web site.
 - i. For each Web page that you want to apply the styles to, insert the following tag in the head section of the HTML file:


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

 where URL is the URL or filename of the style sheet.
 - ii. Do [Assignment 7:6](#)
 - iii. Do [Assignment 7:7](#)
 - iv. Once the final version for the external style sheet is determined, you can modify the design of any of the page by simply adding a <link> tag pointing to the style sheet.
- b. Linking to Style sheets with @import.
 - i. Another way to link to a style sheet is the use the @import command, which accesses the style sheet definitions from another file.
 - ii. To use @import with your styles, you enclose the @import command within the embedded <style> tags as follows:


```
<style>
  @import url(stylesheet.css);
  style declarations
</style>
```

 where stylesheet.css is the URL of the style sheet file.
 - iii. If you want to access a style sheet from within another style sheet, simply add the @import command to your style sheet file:


```
@import url<stylesheet.css>;
styles
```

 The advantage of this approach is that it allows you to easily combine different style sheets.
 - iv. The @import command provides greater flexibility than the <link> tag when working with multiple style sheets, but it has limited browser support. You are probably better off using the <link> tag.

V. RESOLVING STYLE PRECEDENCE

- a. In cases where the styles conflict, precedence is determined in the following order:
 - i. An inline style overrides any embedded style or external style sheet.
 - ii. An embedded style overrides an external style sheet.
 - iii. An external style sheet overrides the internal style rules set by the Web browser.
 - iv. Any style attributes left undefined by an inline style, an embedded style, or an external style sheet are left to the Web browser.
- b. Do [Assignment 7:8](#)

VI. WORKING WITH STYLE INHERITANCE

- a. Parent and Descendant Elements
 - i. An element that lies within another element is called a **descendant** or **descendant element**.
 - ii. An element that contains another element is called the **parent** or **parent element**.
 - iii. The <body> tag is perhaps the prime example of a parent, because it contains all of the other tags used to format the content of your page.
 - iv. Using the principle of **inheritance**, styles defined for each parent tag are transferred to its descendants.


```
body {color: blue}
```

- v. You can override this inheritance by specifying a different style for the element's descendants.
 - body {color: blue}
 - h1, h2 {color: green}
- vi. Do **Assignment 7:9**
- b. Contextual Selectors
 - i. b {color: blue}
 - all boldface text is displayed in blue
 - ii. li b {color: blue}
 - indicates that any boldface text that lies within the tag is displayed as blue
 - iii. li>b {color: blue}
 - If you want to apply a style only to the direct descendant of a parent element.
 - iv. h1 + h2 {color: blue}
 - applies a blue color to h1 and h2 headings that follow each other in the document.

- 7:2
- Learn more about the CSS language.
 - Learn about specific attributes that you can use with styles to modify the appearance of fonts, including font size and alignment.
 - Learn how to use styles to work with colors and background images.
 - Use style attributes to create and format lists.

I. SETTING FONT AND TEXT ATTRIBUTES

- a. Using Font Families
 - i. CSS works with two types of font faces:
 - 1. **specific font** is a font such as Arial, Garamond, or Times New Roman
 - 2. **generic font** is a general description of a font, allowing the operating system to determine which installed font best matches it.
 - a. serif
 - b. sans-serif
 - c. monospace
 - d. cursive
 - e. fantasy
 - ii. To choose a font family for a Web page element, use the style:
 - font-family:font1, font2, ...
 - where font1, font2, and so forth are either specific or generic font names. Generic font names must be serif, sans-serif, monospace, cursive, or fantasy
 - iii. Whenever possible, use specific fonts. Provide the Web browser with several fonts to choose from. List specific font names first, followed by a generic font name for the browser to use if none are available.
 - Body {font-family: Times Roman, Century Schoolbook, serif}
 - iv. Do **Assignment 7:10**
- b. Managing Font Size
 - <address>: is displayed in normal-sized type, italicized, and aligned with the left edge of the Web page.
 - With HTML the size attribute limits you to only seven font sizes, 1 through 7, and the Web browser determines what the precise sizes are.
 - i. In CSS, you use the font-size attribute to manage font sizes. Font sizes can be expressed:
 - 1. as a unit of length
 - a. A unit of length can be absolute units or relative units
 - i. **Absolute units** define the font size based on one of the following standard units of measurement:
 - 1. **mm (millimeter)**
 - 2. **cm (centimeter)**
 - 3. **in (inch)**
 - 4. **pt (point)**
 - a. 72 points equal an inch
 - b. 12 points equal a pica
 - 5. **pc (pica)**
 - a. 6 picas equal an inch
 - ii. Size values can be whole numbers (0, 1, 2, ...) or decimal numbers.
 - iii. These measurement units are useful if you intend to have users print, they don't work as well in the browser window.
 - iv. To overcome this problem, you can use a **relative unit**, one that expresses the font size relative to a size of a standard character. There are two standard typesetting characters, referred to as "em" and "ex."
 - 1. The **em unit** is equal to the width of the capital letter "M"

4. Values of the Vertical Alignment Attribute

baseline	Aligns the element with the baseline
bottom	Aligns the bottom of the element with the bottom of the lowest element (text or image) in the line
middle	Aligns the element in the middle of the text
sub	Aligns the element as a subscript
super	Aligns the element as a superscript
text-bottom	Aligns the element with the font's bottom
text-top	Aligns the element with the top of the tallest letter
top	Aligns the element with the top of the tallest element (text or image) in the line

f. Indenting Text

- i. CSS allows you to indent the first line of a paragraph. The syntax for creating an indentation is:
text-indent: indentation
where indentation is either the length, in either absolute or relative units, of the indentation or a percentage of the width of the paragraph.
- ii. The length and percentage values also can be negative, which extends the first line to the left by the specified value or percentage, and then indents the rest of the lines in the paragraph. The particular effect, called a **hanging indent**, works sporadically on many browsers.

g. Special Text Attributes

- i. Formatting Your Text with Special Attributes
 1. To decorate your text, use the style:
text-decoration: decoration
where decoration equals blink, line-through, overline, underline, or none.
 2. To change the case of the font, use the style:
text-transform: transform
where transform equals capitalize, lowercase, uppercase, or none.
 3. To display a variant of the font's appearance, use the style:
font-variant: variant
where variant equals small-caps or none
- ii. Do **Assignment 7:14**

h. The Font Attribute

- i. You can pool many of the individual text and font attributes you've learned about so far into a single attribute, called the font attribute. The syntax for the font attribute is:
font: font-style font-variant font-weight font-size/line-height font-family
where font-style, font-variant, and so forth are the values for font and text style attributes.
- ii. The font attribute requires that you specify the font size, font variant, and font weight while the other font attributes are optional.

II. WORKING WITH COLOR AND BACKGROUND

a. The Color Attribute

- i. You can use any of the following styles:
 1. body {color: teal}
 2. body {color: #008080}
 3. body {color: rgb(0%, 50%, 50%)}
 4. body {color: rgb(1, 128, 128)}
- ii. Remember that RGB color values range from 0 to 255
- iii. Do **Assignment 7:15**

b. Working with Background Color

- i. To change the background color of almost any element in your Web page, you can use the background-color style.
- ii. Each comment has been formatted with the <blockquote> tag.
Blockquote {background-color: silver}
- iii. Do **Assignment 7:16**

c. Working with Background Images

- i. Almost any element on the page can also be displayed with its own background image. The background image has four attributes:
 1. The source of the image file
 - a. Use the syntax:
background-image: url(URL)
where URL is the location of the image file.
 2. How the image is repeated in the background

- a. By default, background images are tiled both horizontally and vertically behind the element until the entire element is filled.
- b. You can control the way the tiling occurs using the background-repeat style attribute. The background-repeat attribute has four possible values:

repeat	The image is tiled both horizontally and vertically until the entire background of the element is covered.
repeat-x	The image is tiled only horizontally across the width of the element.
repeat-y	The image is tiled only vertically across the height of the element.
No-repeat	The image is not repeated at all.

- 3. Where the image is placed on the background
 - a. Background images are placed in the upper-left corner of their element, and then repeated from there. You can move the background image to a different location using the background-position style attribute. The background-position attribute has two values: the first indicates the distances from the left edge of the element, and the second indicates the distance from the element's top edge. These values can be expressed as a percentage of the display area, in units of length, or with keywords.
 - b. If you enter one attribute value, the browser applies that value to the horizontal position and vertically centers the image.
 - c. For a more general description of image position, you can use left, center, right, and top, center, bottom.
- 4. Whether the image scrolls with the display window
 - a. By default, background images move along with the background of the page as the user scrolls through the Web page. You can change this with the background-attachment attribute. The syntax of this style is:


```
background-attachment: attach
```

 where attach is either "scroll," to scroll the image along with the element, or "fixed," which places the image in a fixed place in the browser's display window, preventing it from moving even if the user scroll down through the Web page.
 - b. Fixed background images are often used to create the impression of a **watermark**, a term that refers to a translucent graphic impressed into the very fabric of the paper and used in specialized stationery.

d. The Background Attribute

- i. You can combine all the various attributes for backgrounds into one attribute, called the background attribute. The syntax for the background attribute is:


```
background: background-color background-image background-repeat
background-attachment background-position
```

 Where background-color, background-image, etc., are the values for the various background attributes.
- ii. You do not have to enter all of the attribute values for the background attribute, but the ones you do specify should follow the order indicated by the syntax.
- iii. Do [Assignment 7:17](#)

III. WORKING WITH LIST STYLES

a. Choosing a List Style Type

- i. The list-style-type attribute allows you to choose the type of label to display alongside text formatted with the , , or tags.

disc (the default)	•
circle	o
square	□
decimal	1, 2, 3, ...
decimal-leading-zero	01, 02, 03, ...
lower-roman	i, ii, iii, ...
upper-roman	I, II, III, ...
lower-alpha	a, b, c, ...
upper-alpha	A, B, C, ...

- ii. You can use contextual selectors to create an outline style for several levels of nested lists.

- iii. To specify the appearance of the list label, use the style:

list-style-type: type

where type equals disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, or upper-alpha.

b. Using a List Style Image

- i. If you want to use a label not included in the list-style-type values, you can create your own with an image file and the list-style-image attribute.
 - ii. To use an image in place of the list label, enter the style:
list-style-image: url(URL)
where URL is the location and filename of the image file.
 - iii. The attribute is not supported by Netscape version 4.7 or earlier. Because of this, it's a good idea to include the list-style-type attribute along with the list-style-image attribute.
- c. Defining the List Style Position
 - i. List items are treated by CSS as if they have an invisible box around them.
 - ii. The labels for the list items can be placed either outside or inside this box.
 - iii. To control the placement of the list label, use the style:
list-style-position: location
where location is "inside" to place the label inside the box, or "outside" to place the label outside the box.
- d. The List-Style Attribute
 - i. You can combine all of these attributes into the list-style attribute. The syntax for this style is:
list-style: list-style-type list-style-image list-style-position
where list-style-type, list-style-image, and list-style-position are the attribute values for each of the individual list style attributes.
 - ii. Do **Assignment 7:18**
 - iii. Do Assignment 7:19

7:3

- Format hypertext links on your Web page.
- Learn how to create a "rollover" effect for them.
- Work with CSS classes and pseudo-classes.
- See how to use the <div> and tags to create containers for blocks of text.
- Learn how to format block-level elements in order to format the layout of your Web page.

I. FORMATTING HYPERTEXT LINKS

- a. Do **Assignment 7:20**
- b. To do away with the default style of underlining hypertext. The following style is one way to accomplish this:
a {text-decoration: none}
- c. A hypertext link can be in one of four states:
 - i. a:visited {styles for previously visited targets}
 - ii. a:link {styles for targets that have never been visited}
 - iii. a:active {styles for links that are currently being clicked}
 - iv. a:hover {styles when the mouse cursor is hovering over the link}
- d. The hypertext links to change in appearance when the pointer passes over them is called a **rollover effect**.
- e. Do **Assignment 7:21**

II. WORKING WITH IDS AND CLASSES

- a **pseudo-class** is a classification of an element based on its status or its use.
- The element itself, a hypertext link with the pointer located over it, is called a **pseudo-element**.
- If you wish to display the first line of your paragraphs in uppercase, you can create the following style definition:
p:first-line {text-transform: uppercase}
- You can increase the size of the first letter of your paragraphs by using this pseudo-class style:
p:first-letter {font-size: 200%}

a. The Class Attribute

- i. You can create your own customized classes by adding the class attribute to your HTML tags. The syntax for creating a class is:
`<tag class="class_name">`
where tag is the HTML tag and class_name is the name of the class.
- ii. Once a class has been created, you can apply a style specific to the class with the declaration:
tag.class_name {style attributes and values}
where tag is the HTML tag and class_name is the name of the class you created.
- iii. If the same class name is used for several different types of tags, you would omit the tag name from the style declaration, following the syntax:
.class_name {style attributes and values}
- iv. Class names cannot contain blank spaces, and they are case-sensitive.

b. The id Attribute

- i. Applies an id to a specific element in the document.
- ii. The id attribute must be unique; there cannot be more than one tag with the same id value.

- iii. The syntax for creating an id is:
`<tag id="id_name">`
 where tag is the HTML tag and id_name is an id name assigned to the tag.
- iv. To apply a style to an id, use the style declaration:
`#id_name {style attributes and values}`
- v. Do **Assignment 7:22**
- vi. Do **Assignment 7:23**

III. WORKING WITH CONTAINER ELEMENTS

-To enclose contents of an article in a **container** that identifies that the start and ending points.

-HTML supports two types of container tags:

- a. The `<div>` Tag
 - i. The `<div>` tag is used to group blocks of text such as paragraphs, block quotes, headings, or lists. Collectively, these text blocks are known as **block-level elements**.
 - ii. To enclose a block-level element within a div container, you use the syntax:
`<div> block-level elements</div>`
 - iii. Note that the `<div>` tag does not actually format the block-level elements; it merely groups them as a unit. For this reason, the `<div>` tag always includes either a class or id attribute that identifies that group.
- b. The `` Tag
 - i. The `` tag is used to contain **inline elements** such as individual letters, words, phrases or inline images. Typically, inline elements appear within block-level elements.
 - ii. The syntax for the `` container is:
`inline elements`
 - iii. You almost always include an id or class attribute with the `` tag.
 - iv. Do **Assignment 7:24**

IV. FORMATTING BLOCK-LEVEL ELEMENT BOXES

-With CSS you can control the layout of a Web page by manipulating the size and location of block-level elements.

-This is a powerful feature of CSS because you can move elements around on the page, apply borders, set the internal margins for a box, etc.

-Here are a few of the HTML tags that can be treated as block-level elements:

- `<h1>` - `<h6>` tags
 - `<p>` tag
 - `<blockquote>` and `<address>` tags
 - ``, ``, and `<dl>` list tags
 - ``, `<dt>`, or `<dd>` tags (individual list items)
 - `<div>` tag
 - `<body>` tag
 - `<hr>` tag
 - `` tag
- b. Parts of the Block-Level Element Box
 - i. There are three elements:
 - **margin**, between the box and the parent element
 - **border**, of the box
 - **padding**, which is the space between the box around the block-level element and the border
 - ii. CSS provides attributes you can use to control the appearance and behavior of each of these elements.
 - c. Controlling Margins
 - i. The margin is the space between the block-level element and the parent element. There are four attributes that control the margin size:
 - **margin-top**—the space between the top and the box and the top margin
 - **margin-right**—the space between the right side of the box and the right margin
 - **margin-bottom**—the space between the bottom of the box and the bottom margin
 - **margin-left**—the space between the left side of the box and the left margin
 - ii. Margin sizes can be expressed in units of length (points, pixels, em units, etc.) or as a percentage of the width of the parent element box.
`li {margin-left: 2em; margin-right: 2em; margin-top: 1em; margin-bottom: 1em}`
 or
`body {margin-left: 5%; margin-right: 5%; margin-top: 5%; margin-bottom: 5%;}`
 - iii. Web page designers can use negative margins to place one block-level element on top of another, creating an “overlay” effect.
 - iv. The four margin attributes can be combined into a single attribute with the syntax:
`margin: margin-top margin-right margin-bottom margin-left`

where margin-top, margin-right, margin-bottom, margin-left are the top, right, bottom, and left margin values.

- v. If you only include three values in the combined attribute, they are applied in the following order: top, right, bottom, and the browser sets the left margin to match the right margin
 - vi. If two values are specified, they are applied to the top and right margins, and the browser sets the bottom and left margins to match the top and right margins, respectively
 - vii. If only one value is entered, the browser applies the value to all four margins
- d. Setting Padding Size
- i. Padding refers to the amount of space between the element and its border.
 - ii. Four attributes are used to control the size of the element's padding:
 - padding-top
 - padding-right
 - padding-bottom
 - padding-left
 - iii. One different to keep in mind is that when padding is expressed as a percentage, it is the percentage of the width of the block-level element rather than the parent element.
- e. Formatting the Border
- i. CSS provides a variety of attributes for managing the box's border width, border color, and border style. These attributes can be applied to all four borders at once, or you can work with individual borders.
 - ii. Different Border Attributes
 - border-top-width Specifies the width of the top border
 - border-right-width Specifies the width of the right border
 - border-bottom-width Specifies the width of the bottom border
 - border-left-width Specifies the width of the left border
 - border-width Specifies the width of any or all of the borders

 - border-top-color Specifies the color of the top border
 - border-right-color Specifies the color of the right border
 - border-bottom-color Specifies the color of the bottom border
 - border-left-color Specifies the color of the left border
 - border-color Specifies the color of any or all of the borders

 - border-top-style Specifies the line style of the top border
 - border-right-style Specifies the line style of the right border
 - border-bottom-style Specifies the line style of the bottom border
 - border-left-style Specifies the line style of the left border
 - border-style Specifies the line style of any or all of the borders
 - iii. Border widths can be expressed using units of length or with the keywords thin, medium, or thick.
 - iv. The border color can be defined using color names or color values.
 - v. There are nine different styles that can be applied to the border.
 - solid
 - dashed
 - dotted
 - double
 - outset
 - inset
 - groove
 - ridge
 - none
 - vi. You can combine all of the border attributes into a single style declaration. The syntax is:
border: border-width border-style border-color;
 - vii. The syntax for formatting individual borders is:
border-top: border-width border-style border-color;
border-right: border-width border-style border-color;
border-bottom: border-width border-style border-color;
border-left: border-width border-style border-color;
 - viii. Do [Assignment 7:25](#)

V. RESIZING AND MOVING BLOCK-LEVEL BOXES

a. Formatting the Width and Height of Block-Level Boxes

- i. To change the width of a box, you use the width attribute. Box width can be expressed in terms of absolute or relative units of length, or as a percentage of the width of the parent element.

Body {width: 75%}

reduces the width of the Web page body to 75% of the width of the browser's display area.

- ii. The height attribute sets the height of the element. Heights can be expressed in absolute or relative lengths, but not percentages. Typically, you won't set the height of a block-level element because problems can arise when the amount of text in the element exceeds the height allowed.
- b. Using the Float Attribute
- i. Floating a Block-Level Element
 - To float a block-level element, use the style declaration:
float: margin
where margin indicates the margin of the parent element to be aligned with the floating element. Possible values of margin are "right" and "left".
 - To prevent a floating element from appearing alongside a block-level element, insert the following style declaration into the nonfloating element:
clear: margin
where margin can be "right", "left", or "both", to leave the right, left, or both both margins clear, respectively.
 - ii. The float attribute works like the align="left" or align="right" attributes used with the tags. It places the block-level element on the left or right margin of the parent element.
 - iii. You can prevent other elements from wrapping around the floating elements from wrapping around the floating element by adding the clear attribute to the element below the floating element. When the value of the clear attribute is set to "right," the browser displays the element on the page at the point where the right margin is clear. Other possible values for the clear attribute are "left" and "both" (for both margins).
 - iv. **Do Assignment 7:26**