Chapter -3

Cascading Style Sheets (CSS)

Introduction:

- A **style sheet** is a syntactic mechanism for specifying style information. In other words, a style sheet language is a computer language that expresses the presentation of structured documents.
- The best example of style sheet is Cascading Style Sheet (CSS). Word processors and desktop publishing systems have long used style sheets to impose a particular style on documents.
- Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in markup language.
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media.**
- XHTML/HTML style sheets are called **Cascading Style Sheets (CSS)** because they can be defined at three different levels (inline level, internal or document level & external level) to specify the style of a document.
- Lower level style sheets can override higher level style sheets, so the style of the content of a tag is determined, in effect, through a cascade of style-sheet applications.

Importance of CSS in Web Design:

Consistency:

- CSS impose **consistency** on the style of Web documents. For example, they allow the author to specify that all occurrences of a particular tag use the same presentation style. Moreover, it can control the layout of multiple web pages all at once.

Viewing Options:

- CSS allows the same markup page to be presented in different viewing styles for different devices with different screen sizes, like- mobile devices, printer, etc. along with desktop presentation. Moreover, browser compatibility ensure that visitors will be able to view websites in their intended way using different browsers.

Bandwidth Reduction:

- CSS separates website's content from design language that drastically reduce file transfer size. The CSS document will be stored externally and will be accessed only once when a visitor requests a website. In contrast, if we create websites using tables (without CSS), every page of websites will be accessed with each visit. Thus, website with CSS require faster load time and thus lower hosting costs.

Search Engines:

- CSS separates website content from designing language, and thus search engines can easily read contents which is critical to search engine success.

Evolution:

Version	Year (A.D.)
CSS1 (by W3C)	1996
CSS2 (with many property values than CSS1)	Mid-1998
CSS 2.1	Late 1990s-2009
CSS3 (still under development)	Late 1990s – till date

Levels of Style Sheets:

- There are three levels of style sheets :
    1. Inline level (Lowest level),
    2. Internal (or Document) level, and
    3. External level (Highest level)

1. Inline CSS:

- Inline style sheets apply to the content of a single XHTML/HTML element
- Inline style specifications appear within the opening tag and apply only to the content of that tag
- `<style>` attribute is used to style a particular HTML tag.
- Inline CSS is mostly used if we don't have access to external CSS files or need to apply style for a single element only

An example of HTML page with inline CSS would be as follows:

```
<! DOCTYPE html>
<html>
<body style="background-color:black;">

<h1 style="color:white;padding:30px;">CSS Tutorials</h1>
<p style="color:white;"> This is our paragraph.</p>

</body>
</html>
```

Advantages of Inline CSS:

- Useful if you want to test, debug and preview changes faster in web pages.
- Lower HTTP requests.
- Inline CSS have precedence over Internal and external style sheets.

Disadvantages of Inline CSS:

- Each HTML tag needs to be styled individually. So, managing website may become too hard if we use only inline CSS.
- Overriding of the CSS properties may occur.
- It's impossible to style pseudo-elements and classes with inline styles. For example, with external and internal style sheets, we can style the visited, hover, active, and link color of an anchor tag. But with an inline style all you can style is the link itself.

2. Internal CSS:

- Internal or document-level CSS apply to the whole body of a single document.
- Internal CSS code is put in between <style></style> tags inside the <head> section of a particular page.
- The classes and IDs can be used to refer to the CSS code, but they are only active on that particular page.
- Internal CSS is useful for sending someone a page template – as everything is in one page, it is easier to see a preview.

An example of internal stylesheet:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
body {
    background-color: black;
}
h1 {
    color: white;
    padding: 30px;
}
.content{
        Color:white;
}

</style>
</head>
<body>

<h1>CSS Tutorials</h1>
<p class="content">This is our paragraph.</p>

</body>
</html>
```

Advantages of Internal CSS:

- Only one page is affected by stylesheet.
- Classes and IDs can be used by internal stylesheet.
- There is no need to upload multiple files. HTML and CSS can be in the same file.
- Internal CSS have precedence over external style sheets.

Disadvantages of Internal CSS:

- Increased page loading time.
- Increased file size.
- It affects only one page – not useful if you want to use the same CSS on multiple documents.

3. External CSS:

- The most convenient way to add CSS to website, is to link it to an external **.css** file.
- Any changes made to an external CSS file will be reflected on your website globally.
- A reference to an external CSS file is put in the `<head>` section of the page
- External style sheets are written as text files with the MIME type text/css
- External style sheets can be stored on any computer on the Web. The browser fetches external style sheets just as it fetches documents.
- The <link> tag is used to specify external style sheets. Within <link>, the rel attribute is used to specify the relationship of the linked-to document to the document in which the link appears.
- The href attribute of <link> is used to specify the URL of the style sheet document, as in the following example:

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

Now, the **style.css** contains all the style rules. For example:

```
body {
    background-color: black;
}
h1 {
    color: white;
    padding: 30px;
}
.content{
        Color:white;
}
```

- The @import directive is an alternative way to use style specifications from other files. The form of this directive is
  @import url(*style.css*);
  Notice that the file name is not quoted.
- There are two differences between link and @import:
  - @import can appear only at the beginning of the content of a style element, and
  - The imported file can contain markup, as well as style, rules. In fact, sometimes the imported file contains other @import directives, along with some style rules.

Advantages of External CSS:

- Smaller size of HTML pages and cleaner structure.
- Faster loading speed and higher page ranking by search engines.
- Same **.css** file can be used on multiple pages.

Disadvantages of External CSS:

- Until external CSS is loaded, the page may not be rendered correctly.

Style Specification Formats:

- The format of a style specification depends on the level of style sheet.
- Inline style specifications appear as values of the style attribute of a tag, the general form of which is as follows:

```
style = "property_1:value_1; property_2:value_2; ...;
         property_n:value_n;"
```

- Internal (document) style specifications appear as the content of a style element within the header of a document. The general form of the content of a style element is as follows:

```
<style type = "text/css">
  rule_list
</style>
```

- The type attribute of the <style> tag tells the browser the type of style specification, which is always text/css.
- The type of style specification is necessary because there are other kinds of style sheets. For example, JavaScript, which can be embedded in an XHTML document, also provides style sheets that can appear in style elements.
- Each style rule in a rule list has two parts: a selector, which indicates the tag or tags affected by the rule, and a list of property–value pairs.
- The list has the same form as the quoted list for inline style sheets, except that it is delimited by braces rather than double quotes. So, the form of a style rule is as follows:

```
selector {property_1:value_1; property_2:value_2; ...;
          property_n:value_n;}
```

- If a property is given more than one value, those values usually are separated with spaces. For some properties, however, multiple values are separated with commas.
- External style sheets have a form similar to that of document style sheets. The external file consists of a list of style rules

CSS Selectors:

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

The CSS Selectors can have variety of forms which are as follows:

1. **The element Selector**
   - It is the simplest selector form.
   - The property values in the rule apply to all occurrences of the named element.
   - The selector could be a list of element names separated by commas, in which case the property values apply to all occurrences of all of the named elements.
   - For example:

```
p {
   text-align: center;
   color: red;
}
```

     In this example, all p elements will be aligned-center with red text color

   - Selectors can also specify that the style should apply only to elements in certain positions in the document. This is done by listing the element hierarchy in the selector, with only white space separating the element names. For example, the rule
     div p {font-size: 14px; }

2. **The class selector**

   - Class selectors are used to allow different occurrences of the same tag to use different style specifications.
   - To select elements with a specific class, write a period (.) character, followed by the name of the class.
   - For example, if you want two paragraph styles in a document—say, **normal** and **warning**—we could define these two classes in the content of a <style> tag as follows:
     p.normal {*property-value list*}
     p.warning {*property-value list*}

     This is normal paragraph.

     This is warning paragraph.

- HTML elements can also refer to more than one class.

In the example below, the element will be styled according to class="center" and to class="large":

this paragraph refers to two classes.

Note: A class name cannot start with a number!

3. The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- For example, the style rule below will be applied to the HTML element with id="para1":

```
#para1 {
   text-align: center;
   color: red;
}
```

Note: The id name cannot start with a number!

4. The Universal Selector

- The universal selector, denoted by an asterisk (*), applies its style to all elements in a document. For example,
  * {color: red;}

  makes all elements in the document red.

5. The Generic Selector

- Sometimes it is convenient to have a class of style specifications that applies to the content of more than one kind of tag.
- This is done by using a generic class, which is defined without a tag name in its name.
- In place of the tag name, you use the name of the generic class, which must begin with a period, as in the following generic
  style class:
  .sale {*property-value list*}
  Now, in the body of a document, you could have the following markup:

```
<h3 class = "sale"> Weekend Sale </h3>
...
<p class = "sale">
...
</p>
```

6. **Pseudo Classes**
   - Pseudo classes are styles that apply when something happens, rather than because the target element simply exists.
   - In this section, we describe and illustrate two pseudo classes: hover and focus.
   - Whereas the names of style classes and generic classes begin with a period, the names of pseudo classes begin with a colon.
   - The style of the hover pseudo class applies when its associated element has the mouse cursor over it. The style of the focus pseudo class applies when its associated element has focus.

```
<head> <title> Pseudo Classes </title>
  <style type = "text/css">
    input:hover {background: pink; color: red;}
    input:focus {background: lightblue; color: blue;}
  </style>
</head>
<body>
  <form action = "">
    <p>
      <label>
        Your name:
        <input type = "text"  />
      </label>
    </p>
  </form>
</body>
```

Property Value Forms:

- CSS includes different properties in different categories like: fonts, lists, alignment of text, margins, colors, backgrounds, and borders.
- The possible unit names for length values are **px**, for pixels; **in**, for inches; cm, for centimeters; mm, for millimeters; pt, for points (a point is 1/72 inch); and pc for picas (= 12 points). Their actual values depend on screen resolution. There are also two relative length values: em, which is the value of the current font size in pixels

Font Properties:

i. Font-family:
   - The font-family property is used to specify a list of font names.

- The browser uses the first font in the list that it supports. For example, the property:
  font-family: Arial, Helvetica, Futura
  tells the browser to use Arial if it supports that font. If not, it will use Helvetica if it supports it. If the browser supports neither Arial nor Helvetica, it will use Futura if it can. If the browser does not support any of the specified fonts, it will use an alternative of its choosing.
- If a font name has more than one word, the whole name should be delimited by single quotes, as in the following example:
  font-family: 'Times New Roman'
- A generic font can be specified as a font-family value. The possible generic fonts and examples of each are shown below:

Table 3.1 Generic fonts

Generic Name	Examples
serif	Times New Roman, Garamond
sans-serif	MS Arial, Helvetica
cursive	Caflisch Script, Zapf-Chancery
fantasy	Critter, Cottonwood
monospace	Courier, Prestige

ii. Font size:

- The font-size property specifies the font size of the specified document. For example, the following property specification sets the font size for text to 10 points:
  font-size: 10pt
- Many relative font-size values are defined, including xx-small, x-small, small, medium, large, x-large, and xx-large. In addition,
  smaller or larger can be specified. Furthermore, the value can be a percentage relative to the current font size.
- Using relative font size has many disadvantages during browser variation.

iii. Font-variant:

- The font-variant property specifies whether or not a text should be displayed in a small-caps font (upper case letters but with smaller font-size).

CSS Syntax:

font-variant: normal|small-caps|initial|inherit;

Property Values

Value	Description
normal	The browser displays a normal font. This is default
small-caps	The browser displays a small-caps font
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element

iv. Font-style:
- The font-style property specifies the font style for a text.

CSS Syntax:

font-style: normal|italic|oblique|initial|inherit;

Property Values

Value	Description
normal	The browser displays a normal font style. This is default
italic	The browser displays an italic font style
oblique	The browser displays an oblique font style (slightly slanted right)
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

v. Font-weight:
- The font-weight property sets how thick or thin characters in text should be displayed.

CSS Syntax

font-weight: normal|bold|bolder|lighter|*number*|initial

Property Values

Value	Description
normal	Defines normal characters. This is default
bold	Defines thick characters
bolder	Defines thicker characters
lighter	Defines lighter characters
100 200 300 400 500	Defines from thin to thick characters. 400 is the same as normal, and 700 is the same as bold

600 700 800 900	
initial	Sets this property to its default value.
inherit	Inherits this property from its parent element.

vi. Font shorthand:

- If more than one font property must be specified, the values can be stated in a list as the value of the font property. The browser then has the responsibility for determining which properties to assign from the forms of the values. For example, the property:
  font: bold 14pt 'Times New Roman' Palatino

  specifies that the font weight should be bold, the font size should be 14 points, and either Times New Roman or Palatino font should be used, with precedence given to Times New Roman.
- The order in which the property values are given in a font value list is important.

vii. Text-decoration:

- The text-decoration property is used to specify some special features of text.

CSS Syntax

text-decoration: none|underline|overline|line-through|;

Property values:

Value	**Description**
none	Defines a normal text. This is default
underline	Defines a line below the text
overline	Defines a line above the text
line-through	Defines a line through the text

- Many browsers implicitly underline links. The none value can be used to avoid this underlining.
- Note that text-decoration is not inherited.

List Properties:

- The list-style-type property is used to specify styles for both ordered and unordered lists:
- The list-style-type property of an unordered list can be set to disc, circle, square, or none. A disc is a small filled circle, a circle is an unfilled circle, and a square is a filled square. The default property value for bullets is disc

- Any image can be used in a list item bullet. Such a bullet is specified with the liststyle-image property, whose value is specified with the url form.

```
<style type = "text/css">
  li.image {list-style-image: url(small_airplane.gif)}
</style>
  ...
  <li class = "image"> Beechcraft Bonanza </li>
```

- When ordered lists are nested, it is best to use different kinds of sequence values for the different levels of nesting. The list-style-type property can be used to specify the types of sequence values

Property Values	Sequence Type
decimal	Arabic numerals starting with 1
decimal-leading-zero	Arabic numerals starting with 0
lower-alpha	Lowercase letters
upper-alpha	Uppercase letters
lower-roman	Lowercase Roman numerals
upper-roman	Uppercase Roman numerals
lower-greek	Lowercase Greek letters
lower-latin	Same as lower-alpha
upper-latin	Same as upper-alpha
armenian	Traditional Armenian numbering
georgian	Traditional Georgian numbering

```
<! DOCTYPE html>
<html>
 <head> <title> Sequence types </title>
   <style type = "text/css">
     ol {list-style-type: upper-roman;}
     ol ol {list-style-type: upper-alpha;}
     ol ol ol {list-style-type: decimal;}
   </style>
 </head>
 <body>
   <h3> Aircraft Types </h3>
   <ol>
     <li> General Aviation (piston-driven engines)
       <ol>
         <li> Single-Engine Aircraft
           <ol>
             <li> Tail wheel </li>
             <li> Tricycle </li>
           </ol>
         </li>
         <li> Dual-Engine Aircraft
           <ol>


             <li> Wing-mounted engines </li>
             <li> Push-pull fuselage-mounted engines </li>
           </ol>
         </li>
       </ol>
     </li>
     <li> Commercial Aviation (jet engines)
       <ol>
         <li> Dual-Engine
           <ol>
             <li> Wing-mounted engines </li>
             <li> Fuselage-mounted engines </li>
           </ol>
         </li>
         <li> Tri-Engine
           <ol>
             <li> Third engine in vertical stabilizer </li>
             <li> Third engine in fuselage </li>
           </ol>
          </li>
         </ol>
       </li>
     </ol>
   </body>
</html>
```

Output:

Aircraft Types

I. General Aviation (piston-driven engines)
   A. Single-Engine Aircraft
      1. Tail wheel
      2. Tricycle
   B. Dual-Engine Aircraft
      1. Wing-mounted engines
      2. Push-pull fuselage-mounted engines
II. Commercial Aviation (jet engines)
   A. Dual-Engine
      1. Wing-mounted engines
      2. Fuselage-mounted engines
   B. Tri-Engine
      1. Third engine in vertical stabilizer
      2. Third engine in fuselage

Color:

- The color property is used to specify the foreground color of XHTML elements
- The background-color property is used to set the background color of an element, where the element could be the whole body of the document.
- Color property values can be specified as color names, as six-digit hexadecimal numbers, or in RGB form. RGB form is just the word **rgb** followed by a parenthesized list of three numbers that specify the levels of red, green, and blue, respectively. The RGB values can be given either as decimal numbers between 0 and
  255 or as percentages. Hexadecimal numbers must be preceded with pound signs (#), as in #43AF00. For example, powder blue could be specified with
  fuchsia
  or
  rgb(255, 0, 255)
  or
  #FF00FF

Table 3.3 Named colors

Name	Hexadecimal Code	Name	Hexadecimal Code
aqua	00FFFF	olive	808000
black	000000	orange	FFA500
blue	0000FF	purple	800080
fuchsia	FF00FF	red	FF0000
gray	808080	silver	C0C0C0
green	008000	teal	008080
lime	00FF00	white	FFFFFF
maroon	800000	yellow	FFFF00
navy	000080		

- A larger set of colors, called the **Web palette**, consists of 216 colors. These colors, which are often called Web-safe colors, are displayable by Windows- and Macintosh-based browsers.

Text-alignment property

- The text-align property specifies the horizontal alignment of text in an element.

CSS Syntax

text-align: left|right|center|justify|initial|inherit;

Property Values:

Value	Description
left	Aligns the text to the left
right	Aligns the text to the right
center	Centers the text
justify	Stretches the lines so that each line has equal width (like in newspapers and magazines)

- The text-indent property can be used to indent the first line of a paragraph. This property takes either a length or a percentage value.

  For example:

```
<style type = "text/css">
  p.indent {text-indent: 0.5in}
</style>
...
<p class = "indent">
  Now is the time for all good Web programmers to begin
  using cascading style sheets for all presentation
  details in their documents. No more deprecated tags
  and attributes, just nice, precise style sheets.
</p>
```

This paragraph would be displayed as follows:

```
      Now is the time for all good Web programmers to begin
using cascading style sheets for all presentation details
in their documents. No more deprecated tags and attributes,
just nice, precise style sheets.
```

The Box Model:

- Virtually all document elements can have borders with various styles, such as color and width.
- The amount of space between the content of an element and its border, is known as **padding**, and the space between the border and an adjacent element, is known as the **margin.**
- When there is no border, the **margin** plus the **padding** is the space between the content of an element and its neighbors.
- Although margin and padding seem similar when no border, but the background extends into the **padding**, but not into the margin.
- Padding creates extra space within an element, while margin creates extra space around an element.
- This model is shown below:

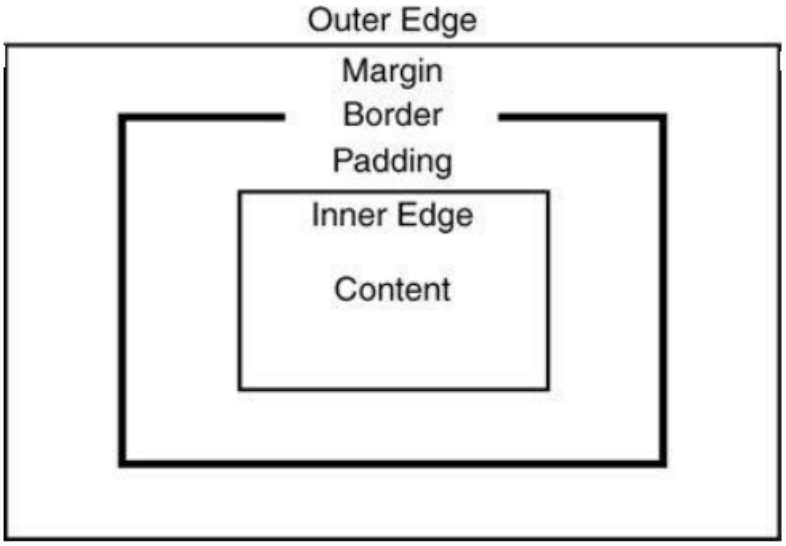

Figure The box model

1) Borders:

a. Border-style:

- The **border-style** property sets the style of an element's four borders.

CSS Syntax

border-style: none|dotted|dashed|solid|double|;

Property Values:

Value	Description
None	Default value. Specifies no border
dotted	Specifies a dotted border
dashed	Specifies a dashed border
solid	Specifies a solid border
double	Specifies a double border

- This property can have from one to four values.

For example:

- **border-style: dotted; (** all four borders are dotted)
- **border-style: dotted solid double dashed;** (top border is dotted, right border is solid, bottom border is double, left border is dashed)
- **border-style: dotted solid double;** (top border is dotted, right and left borders are solid, bottom border is double)
- **border-style: dotted solid;** (top and bottom borders are dotted, right and left borders are solid)

- The styles of one particular side of an element can also be set with **border-top-style, border-bottom-style, border-left-style, and border-right-style.**

b. border-width:

- The border-width property sets the width of an element's four borders

CSS Syntax

border-width: medium|thin|thick|*length*|;

Property Values:

Value	Description
Medium	Specifies a medium border. This is default
thin	Specifies a thin border
thick	Specifies a thick border

Length (e.g. 1px, 5px,etc)	Allows you to define the thickness of the border.

c. Border-color:

- The border-color property sets the color of an element's four borders.

CSS Syntax:

border-color: *color*|transparent|;

Property Values:

Value	Description
color	Specifies the background color. There is a defined list of possible color values. Default color is black.
transparent	Specifies that the border color should be transparent

- This property can have from one to four values.

If the border-color property has four values:

- border-color: red green blue pink;
  - top border is red
  - right border is green
  - bottom border is blue
  - left border is pink

If the border-color property has three values:

- border-color: red green blue;
  - top border is red
  - right and left borders are green
  - bottom border is blue

If the border-color property has two values:

- border-color: red green;
  - top and bottom borders are red
  - right and left borders are green

If the border-color property has two values:

- border-color: red;

- all four borders are red

- The individual borders of an element can be colored differently through the properties: **border-top-color, border-bottom-color, border-left-color, and border-right-color**.
- Always **border-style** property must be declared before the **border-color** property. An element must have borders before you can change the color.

d. Border :

- The border property is a shorthand property for border-width, border-style and border-color.

CSS Syntax

border: ***border-width border-style border-color*****|;**

Property Values:

Value	Description
Border-width	Specifies the width of the border. Default value is "medium"
Border-style	Specifies the style of the border. Default value is "none"
Border-color	Specifies the color of the border. Default value is the color of the text

Example:

```
h1 {
   border: 5px solid red;
 }
```

2) Margins and Padding:

- **Margin:**
  - The margin property is a shorthand property for the **margin-top**, **margin-right, margin-bottom, and margin-left** properties.
  - This property can have from one to four values.

If the margin property has four values:

- margin: 10px 5px 15px 20px;
  - top margin is 10px
  - right margin is 5px
  - bottom margin is 15px
  - left margin is 20px

If the margin property has three values:

- margin: 10px 5px 15px;
  - top margin is 10px
  - right and left margins are 5px
  - bottom margin is 15px

If the margin property has two values:

- margin: 10px 5px;
  - top and bottom margins are 10px
  - right and left margins are 5px

If the margin property has one value:

- margin: 10px;
  - all four margins are 10px

➢ **Example:**
To set the margin for a <p> element to 35 pixels for top, 70 pixels for right, 50 pixels for bottom, and to 90 pixels for left:

```
p {
  margin: 35px 70px 50px 90px;
}
```

- **Padding:**

➢ The padding property is a shorthand property for the padding-top, padding-right, padding-bottom, and padding-left properties.

If the padding property has four values:

- padding:10px 5px 15px 20px;
  - top padding is 10px
  - right padding is 5px
  - bottom padding is 15px
  - left padding is 20px

If the padding property has three values:

- padding:10px 5px 15px;
  - top padding is 10px
  - right and left padding are 5px
  - bottom padding is 15px

If the padding property has two values:

- padding:10px 5px;
  - top and bottom padding are 10px
  - right and left padding are 5px

If the padding property has one value:

- padding:10px;
  - all four paddings are 10px

Note: Negative values are not allowed.

Background Images

- The **background-image** property is used to place an image in the background of an element.
- Note that the background image is **replicated** as necessary to fill the area of the element. This replication is called *tiling*.
- Tiling can be controlled with the **background-repeat** property, which can take the value repeat (the default), **no-repeat, repeat-x, or repeat-y.**
- The **no-repeat** value specifies that just one copy of the image is to be displayed.
- The **repeat-x** value means that the image is to be repeated horizontally; **repeat-y** means that the image is to be repeated vertically.
- In addition, the position of a non-repeated background image can be specified with the background-position property, which can take a large number of different values.
- The keyword values are **top, center, bottom, left, and right**, all of which can be used in many different combinations.

For example:

Set two background images for the <body> element:

```
body {
   background-image: url("img_tree.gif"), url("paper.gif");
   background-color: #cccccc;
}
```

Conflict Resolution:

- When there are two different values for the same property on the same element in a document, there is an obvious conflict that the browser (or other XHTML processor) must resolve.
- This particular kind of conflict is resolved by the precedence of the three different levels of style sheets.

- Inline style sheets have precedence over document and external style sheets, and document (internal) style sheets have precedence over external style sheets.
- But a conflict may also occur within a single style sheet. Consider the following style specifications, which are next to each other in the same document-level style sheet. For example:
  h3 {color: blue;}
  body h3 {color: red;}
  Both of these specifications apply to all h3 elements in the body of the document.
- Inheritance is another source of property value conflicts. Such conflicts can occur if a property on a particular element has a value assigned by some style sheet and also inherits a value for that same property.
- Finally, property value specifications can be marked as being important by including !important in the specification. For example, in the specification
  p.special {font-style: italic !important; font-size: 14}
  font-style: italic is important, but font-size: 14 is normal.
- Whether a specification has been marked as being important is called the weight of the specification. The weight can be either normal or important.
- If there are conflicts after the sorting just described takes place, the next step in their resolution is a sort by specificity. This sort is based on the following rules, in
  which the first has the highest precedence:
  1. id selectors
  2. Class and pseudo class selectors
  3. Contextual selectors (more element type names means that they are more specific)
  4. Universal selectors
- If there are still conflicts, they are resolved by giving precedence to the most recently seen specification.
- For example, if a style sheet specifies the following, and there are no further conflicting specifications before the element is displayed, the value used will be the last (in this case,10pt):
  p {font-size: 12pt}
  p {font-size: 10pt}