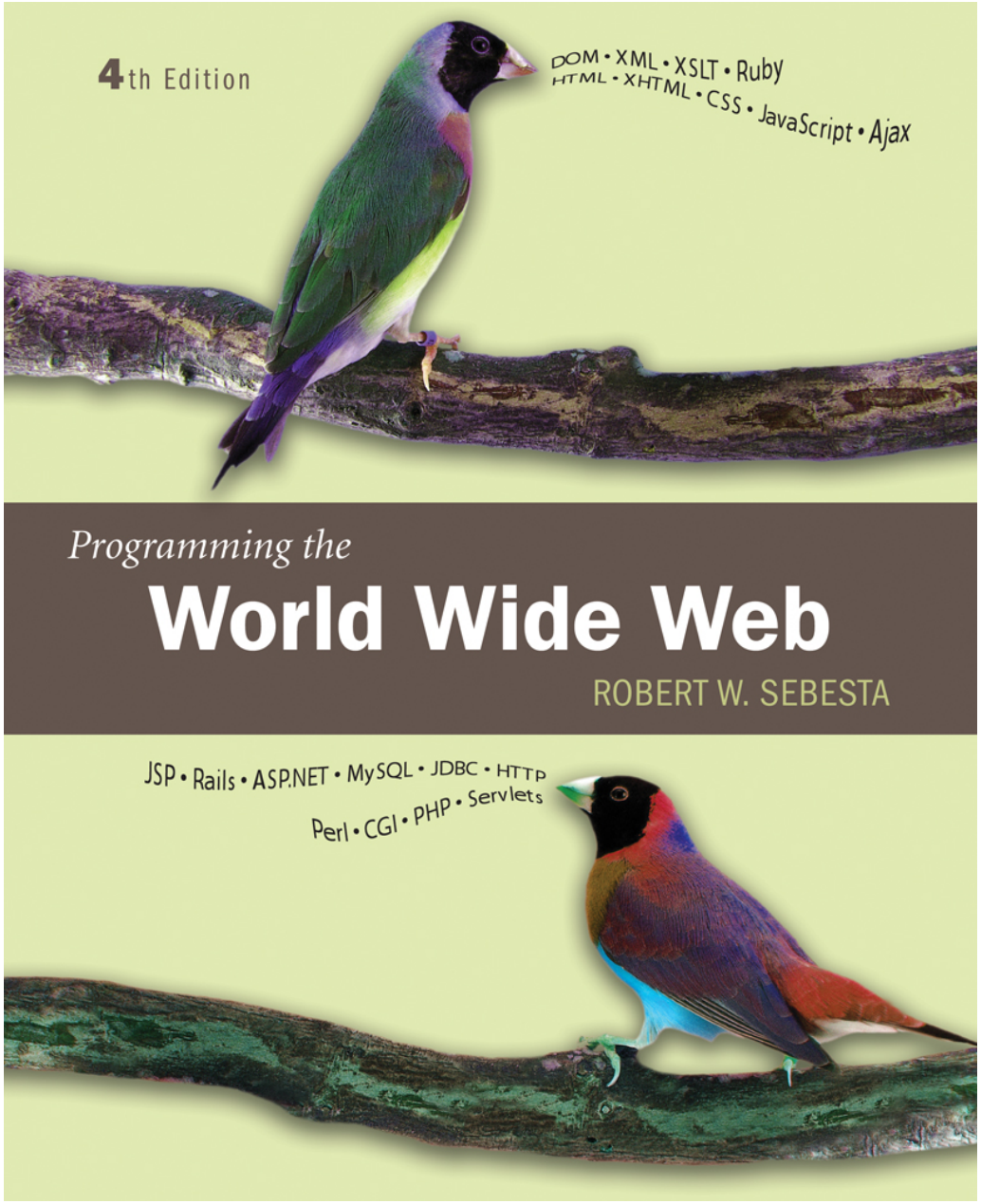


Chapter 3
Cascading Style Sheets

PEARSON Addison Wesley

TOPICS

- Introduction
- Levels in a Style Sheet
- Style specification formats
- Selector forms
- Property value forms
- Font properties
- List properties
- Color
- Alignment of Text
- The Box Model
- Background Images
- The and the tags

Introduction

- Generally a design of a website or a bigger web documents include several pages of design
- Hence its necessary to maintain uniformity in terms of the design of the document, in terms of font , colors used etc..
- Therefore , in such cases developing single pages and integrating them may often lead to haphazard web pages as not all the pages would be developed by the same developer.
- Also to change the design its necessary to change every page in the document
- Therefore , in order to overcome all the above issues style sheets were introduced . A style sheet is a common set of rules which are applicable to a wide range of web documents using them .
- So since all the web documents use the same set of style sheets there is consistency in design
- Design changes are simpler since they involve only changing the style sheets and not all the web documents .

Introduction

- XHTML is primarily concerned with content rather than the details of how that content is presented by browsers.

- For presentation browser makes use of default properties if it is not been specified in the document.
- Eg: <h2> tag uses default value for font as 18 points.

  Style sheet could specify that the font-size property for <h2> be set to 20 points, and can be used for all occurrences of <h2> in the document.

- The first style-sheet specification for use in XHTML documents, CSS1 specification was developed in 1996 by W3C.

- CSS2 was released in 1998.

- CSS3 has been under development since late 1990's.

- CSSs provide the means to control and change presentation of HTML documents.
- CSS is not technically HTML, but can be embedded in HTML documents.
- Style sheets allow you to impose a standard style on a whole document, or even a whole collection of documents.
- Style is specified for a tag by the values of its properties.
- XHTML style sheets are called cascading style sheets because they can be defined at three different levels to specify the style to the 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.

Levels of Style Sheets

- The three levels of style sheets, in order from lowest level to highest level, are inline, document level and external.
- Inline - specified for a specific occurrence of a tag and apply only to that tag. (apply to the content of a single element).
- Document-level style sheets - apply to the whole document in which they appear.
- External style sheets - can be applied to the bodies of any number of documents.
- When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence.
  - In a sense, the browser searches for a style property spec, starting with inline, until it finds one (or there isn't one).
  - Inline style sheets have precedence over document style sheets, which have precedence over external style sheets.
  - If no style sheet information is specified, the browser default property values are used.

- A particular browser may not be capable of using the property values specified in a style sheet, in that case it ignores the values and uses the default value.

  E.g.: If font-size property of a paragraph is set to 18 points, but browser can display font-size only up to 16 points, then it ignores the given font-size value.

Inline style sheets:

- Inline style specifications appear within the opening tag and apply only to the content of that tag.
- Inline style sheets appear in the tag itself.

- This fine-grain application of style defeats one of the primary advantage of style sheets – imposing uniform style on the tags of atleast one whole document.
- This also resulted in style information being embedded in various places in documents.
- Hence inline style specifications should be used sparingly.

- *e.g. use of inline style sheets*

htmlhtml text

Document-level style sheets:

- Document-level style specifications appear in the document head section and apply to the entire body of the document.
- This imposes a uniform style on the presentation of all the content of a document.

External-level style sheets:

- To have a style-sheet apply to more than one document.
- External style sheets are stored in separate files, and are referenced by documents that use them.
  - Written as text files with the MIME type text/css.
  - They can be stored on any computer on the web.

Linking an External Style sheet:

- A <link> tag is used to specify that the browser is to fetch and use an external style sheet file

```
<link rel = "stylesheet"  type = "text/css"
 href = "http://www.wherever.org/termpaper.css">
</link>
```

<link rel = "stylesheet" type = “text/css” href = "http://www.wherever.org/termpaper.css">
</link>

- Within link, the rel attribute is used to specify the relationship of the linked-to document to the document in which link appears.
- The link to an external style sheet must appear in the head of the document.

- The @import directive can be used to use style specifications from different files.
  @import url(filename);

- Two differences between link and @import are:
  1. @import can appear only at the beginning of the content of a style element.
  2. The imported file can contain markup, as well as style rules.

- External style sheets can be validated
  http://jigsaw.w3.org/css-validator/validator-upload.html

Style Specification Formats

- The format of a style specification depends on the level of the style sheet.

- Inline style sheet:
  - Style specifications appears as the value of the style attribute of a tag.
  - General form:

    style = "property_1: value_1; property_2: value_2; …;
    property_n: value_n;"

  - Document style sheet:
  - Style specification appear as the content of a style element within the header of a document.
  - General form:

    <style type = "text/css">
    rule list
    </style>

  - The type attribute of style tag tells the browser the type of style specification, text/css.

- Style sheet appears as a list of rules that are the content of a <style> tag.

- 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 form of style rule is as follows:

  selector {property_1: value_1; property_2: value_2; …;
  property_n: value_n;}

  - Each property/value pair has the form:
    property: value
  - Pairs are separated by semicolons, just as in the value of a <style> tag

- Comments in the rule list must have a different form - use C comments (/*…*/).

- External style sheets have a form similar to that of document style sheets.
- The external file consists of list of style rules.

Selector Forms: Simple

- The selector is a single element name (tag) or a list of element (tag) names, separated by commas.
- The property values in the rule apply to all occurrences of the named elements.

- Consider the following examples, in which the property font-size and property value is a number of points.
- h1 {font-size: 24pt;}
- h2, h3 {font-size : 20pt;}

- Selectors can also specify that the style should only apply to elements in certain positions in the document.
- This is done by listing the element hierarchy in the selector, with only whitespace separating the element names.

- body b em {font-size: 14pt;} only applies its style of emphasis elements that are descendants of body elements in the body of the document.
- *Contextual selector or descendant selector*
- ol ol li

Class Selectors

- Used to allow different occurrences of the same tag to use different style specifications.

- A style class is defined in a style element, giving it a name, which is attached to a tag's name with a period.
  - p.narrow {property-value list}
  - p.wide {property-value list}

- The class you want on a particular occurrence of a tag is specified with the class attribute of the tag, written within the document body.
- For example,

```
<p class = "narrow">
...
</p>
...
<p class = "wide">
...
</p>
```

Generic Selectors

- A generic class can be defined, if you want to have a class of style specifications to apply to the content of more than one kind of tag.

- A generic class is defined without a tag name in its name, and the tag name must begin with a period.

- Example,

  .really-big { property-value list }

- Use it as if it were a normal style class

```
<h1 class = "really-big">
…
</h1>
...
<p class = "really-big">
…
</p>
```

id *Selectors*

- An id selector allow the application of a style to one specific element.
- The style specified in the id selector applies to the element with a specific id.

- General form:

  #specific-id {property-value list}

- Example:

  #section14 {font-size: 20}

  Specifies a font size of 20 points to the element

  <h2 id= “section14”> 1.4 Calico cats</h2>

Universal selectors

- Denoted by an asterisk (*), applies its style to all elements in the document.
- Example:

  * {color: red;}

  makes all elements in the document red.

Pseudo Classes

- Pseudo classes are styles that apply when something happens, rather than because the target element simply exists.
- Names of pseudo classes begin with colons.
- *hover* classes apply when the mouse cursor is over the element.
- *focus* classes apply when an element has focus.

```
<!-- pseudo.html -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Checkboxes </title>
    <style type = "text/css">
      input:hover {color: red;}
      input:focus {color: green;}
    </style>
  </head>
  <body>
    <form action = "">
      <p>
        Your name:
        <input type = "text" />
      </p>
    </form>
  </body>
 </html>
```

Output of the above mentioned code is as shown in HTML here.

Property value forms

- There are 60 different properties in 7 categories:
  - Fonts
  - Lists
  - Alignment of text
  - Margins
  - Colors
  - Backgrounds
  - Borders
- Property values can appear in a variety of forms.
- Keywords property values are used when there are only a few possible values and they are predefined- left, small, medium …
  - Not case sensitive
  - Length - numbers, maybe with decimal points.

- Colors
  - Color name
  - rgb(n1, n2, n3) rgb values from 0 to 255.
    - Numbers can be decimal or percentages
  - Hex form: #XXXXXX

- Length- values are specified as number values followed immediately by a unit name:
  - px - pixels
  - in - inches
  - cm - centimeters
  - mm - millimeters
  - pt - points
  - pc - picas (12 points)

  - The actual value of these depends on the screen resolution.

  - No space is allowed between the number and the unit specification e.g., 1.5 in is illegal!

  - Percentage - just a number followed immediately by a percent sign.

- URL values
  - url(protocol://server/pathname)

Font Properties

- Most commonly used of the style-sheet properties.

- The different forms of the font properties are:

Font-families:

- The font-family property is used to specify a list of font names.
- Value is a list of font names - browser uses the first in the list it supports.

- Example:

  font-family: Arial, Helvetica, Courier
  font-family: 'Times New Roman'

If a font name has more than one word, it should be single-quoted.

- A generic font can be specified as a font-family value.
- Each browser has a font defined for each of these generic names.

- Use generic font as the last font in the value of a font-family property.
- Generic fonts: serif, sans-serif, cursive, fantasy, and monospace (defined in CSS).
  - Browser has a specific font for each.

Font sizes:

- font-size: Property does what its name implies.

  Eg: font-size: 10pt;

  - Possible values: a length number or a name (relative font-size values), such as smaller, larger, small, large, medium, x-large, x-small, xx-small, xx-large.

Font variants:

- Default value of font-variant property is normal, specifies usual character font.
- Property can be set to small-caps for small capital characters.

Font Styles:

- font-style: Most commonly used to specify italic.
  - italic, oblique (identical to italic), normal

Font weights:

- font-weight - degrees of boldness;

  font-weight: bold

  - bolder, lighter, bold, normal(default).
    - Could specify as a multiple of 100 (100 – 900).

Font Shorthands:

- font
  - For specifying a list of font properties, font property can be stated in a list as value of font property.

    font: bolder 14pt Arial Helvetica
  - The order in which property values are given are as follows:
  - style, weight, size, name(s).
  - Font names must be last, font size second last, font style, variant and weight can be in any order but must precede font size and font names.

  - Source code Source code and HTML for demonstrating font properties is as shown.

- Source code for the fonts.html when styles are written in an external stylesheet

fonts2.txt fonts2.txt and styles.css is as follows.

Text Decoration:

- The text-decoration property is used to specify some special features of text.
  - line-through, overline, underline, none(default).
  - letter-spacing – value is any length property value, controls the amount of space between characters in text.

  - Source code Source code and HTMLSource code and HTML for this property is as follows.

List Properties

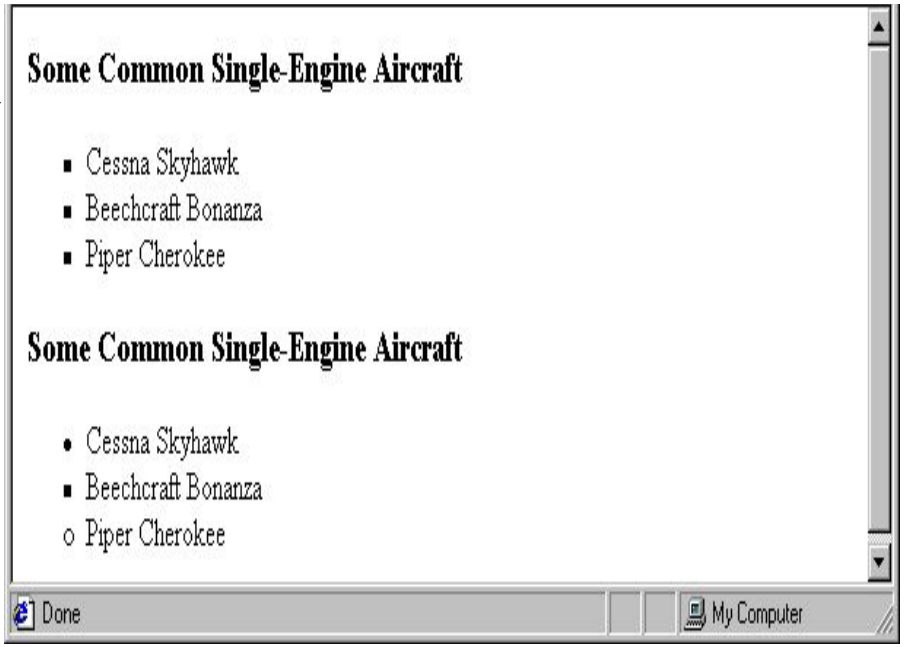
- List-style-type: property is used to specify the shape of the bullets that precede the items in an unordered list and the sequence of values that precede items in an ordered list.

- Unordered list: property values can be disc, circle, square or none.
- Default property value for bullets is disc.

- Ordered list: Different kind of sequencing values can be used for different levels of nesting.

Property value	*Sequence type*	*First four*
`Decimal`	Arabic numerals	1, 2, 3, 4
`upper-alpha`	Uc letters	A, B, C, D
`lower-alpha`	Lc letters	a, b, c, d
`upper-roman`	Uc Roman	I, II, III, IV
`lower-roman`	Lc Roman	i, ii, iii, iv

```
<style type="text/css">
ul {list-style-type: square}
</style>
……
 <h3> Some Common Single-Engine Aircraft </h3>
  <ul>
     <li> Cessna Skyhawk </li>
     <li> Beechcraft Bonanza </li>
     <li> Piper Cherokee </li>
  </ul>
```



- On <li>, list-style-type applies to just that item

- In case of using an inline style specification:

```
<h3> Some Common Single-Engine Aircraft </h3>
 <ul>
    <li style = "list-style-type: disc">
        Cessna Skyhawk </li>
    <li style = "list-style-type: square">
        Beechcraft Bonanza </li>
```

```
<li style = "list-style-type: circle">
        Piper Cherokee </li>
 </ul>
```

OR

In case of document style specification:

```
<style type="text/css">
  li.disc {list-style-type: disc}
  li.square {list-style-type: square}
  li.circle {list-style-type: circle}
<style>
…….
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
     <li class="disc" >  Cessna Skyhawk </li>
     <li class="disc" > Beechcraft Bonanza </li>
     <li class="disc" >   Piper Cherokee </li>
</ul>
```

- An image can be used in a list item bullet.
- Such a bullet is specified with the list-style-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>
```

- Example source code Example source code for an ordered list along with its output is as follows.

Color

- Color is a problem for the Web for two reasons:
  1. Monitors vary widely
  2. Browsers vary widely.

  3. There is a set of 16 colors that are guaranteed to be displayable by all graphical browsers on all color monitors
  2. There is a much larger set, the Web Palette
  - 216 colors
  - Use hex color values of 00, 33, 66, 99, CC, and FF.
  3. Any one of 16 million different colors

- Color names and their hexadecimal codes:

black	000000	green	008000
silver	C0C0C0	lime	00FF00
gray	808080	olive	808000
white	FFFFFF	yellow	FFFF00

maroon	800000	navy	000080
red	FF0000	blue	0000FF
purple	800080	teal	008080
fuchia	FF00FF	aqua	00FFFF

- Colors are referred to in css as in xhtml in 3 ways
- In terms of Hexadecimal codes.
- In terms of the color names(named colors).
- RGB form with the word rgb followed by parenthesized list of three numbers(0 to 255) or as percentages

Color properties:

- The `color` property specifies the foreground color of elements.

- The background-color property specifies the background color of elements.

```
<style type = “text/css”>
   th.red {color: red}
   th.orange {color: orange}
 </style>
 …
 <table border = "5">
   <tr>
     <th class = "red"> Apple </th>
     <th class = "orange"> Orange </th>
     <th class = "orange"> Screwdriver </th>
   </tr>
 </table>
```

- An example source code An example source code and HTML for using color properties is as follows.

.

Alignment of Text

- The text-indent property allows indentation , used to indent first line of a paragraph.
  - Takes either a length or a % value.

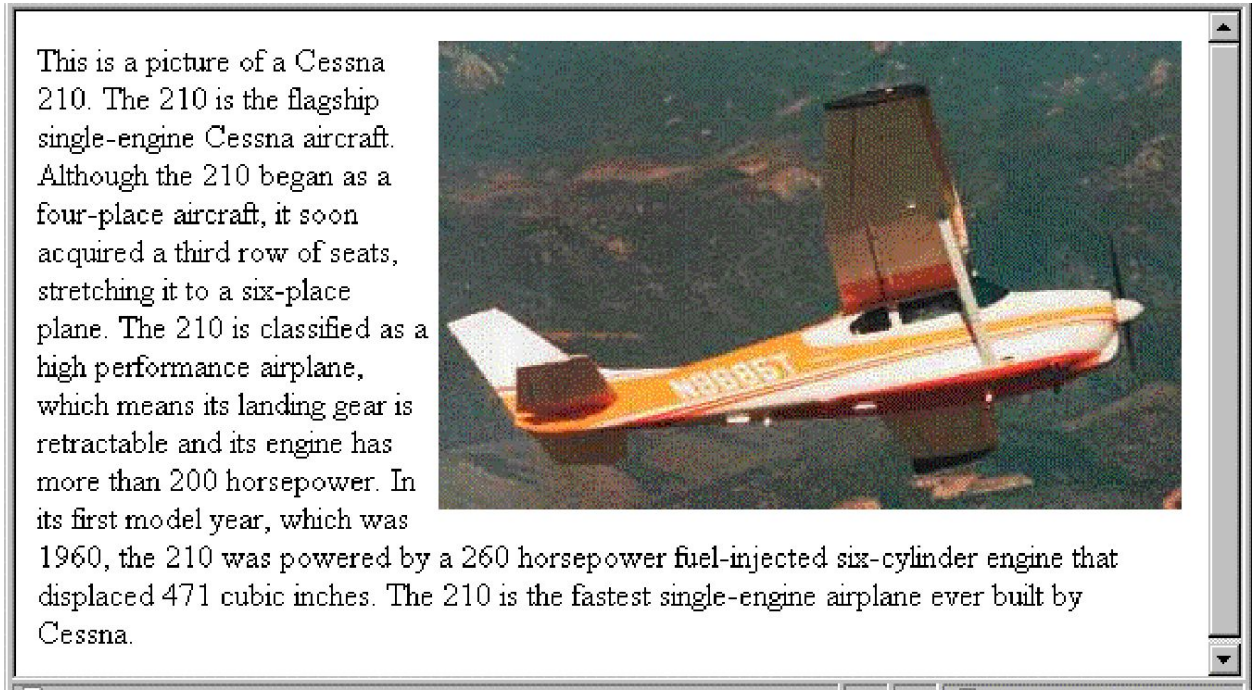
```
<style type="text/css">
  p.indent {text-indent: 0.5in}
</style>
```

- The text-align property has the possible values, left (the default), center, right, or justify, is used to arrange text horizontally.

- Sometimes we want text to flow around another element - the float property, is often set for images and tables, used to specify that text should flow around some element.
  - The float property has the possible values, left, right, and none (the default)
  - If we have an element we want on the right, with text flowing on its left, we use the default text-align value (left) for the text and the right value for float on the element we want on the right.

```
<style type="text/css">
   img {float:right}
</style>
……
<body>
  <img src="   " alt= "   " />
<p> …… </p>
```

- Some text with the default alignment - left



The Box Model

- **All document elements have borders. These borders have various styles such as color and width.**

- **The amount of space between the content of an element and its border – padding.**
- **The space between the border and adjacent element – margin.**

Borders – every element has a border-style property.

- **Controls whether the element has a border and if so, the style of the border, default style is solid, when border attribute is set to border or a pixel width.**
- **The style of one of the four sides of an element can be set with border-top-style, border-bottom-style, border-left-style, border-right-style.**
- **border-style values: none, dotted, dashed, and double.**

- **border-width property is used to specify the thickness of the border– thin, medium (default), thick, or a length value in pixels.**
- **Border width can be specified for any of the four borders (e.g., border-top-width, border-bottom-width, border-left-width, border-right-width)**

- **border-color – color of a border is controlled by this property.**
- **Border color can be specified for any of the four borders (e.g., border-top-color, border-bottom-color, border-left-color, border-right-color).**

Source code Source code and HTML for the use of the borders is as shown.

Margins and Padding:

- **Margin – the space between the border of an element and its neighbor element.**
- **Padding – the distance between the content of an element and its border.**

- **The margin properties are named margin, and can be set around an element with margin-left, margin-right, margin-top, margin-bottom. - just assign them a length value.**

- **The padding properties named padding, which applies to all sides, padding-left, padding-right, padding-top and padding-bottom.**

Source code Source code and HTML for use of margin and padding are as follows.

Background Images

- **The background-image property: Is used to place an image in the background of an element.**

  Source code **Source code and HTML to demonstrate the use of background is as follows.**

- **Background image is replicated as necessary to fill the area of the element. This replication is called tiling.**

- **Repetition can be controlled**
  - **background-repeat property**
  - **Possible values: repeat (default), no-repeat, repeat-x, or repeat-y.**

  no-repeat: specifies that just one copy of the image is to be displayed.

  repeat-x: image is to be repeated horizontally.

  repeat-y: image is to be repeated vertically.
  - **background-position property: position of non repeated image can be specified.**
    - **Possible values: top, center, bottom, left, or right.**
    - **Can use combinations as top left, bottom right, top center.**

The and tags

- One problem with the font properties is that they apply to whole elements, which are often too large.
  - Solution: a new tag to define an element in the content of a larger element -
  - The default meaning of is to leave the content as it is

```
<p>
Now is the <span> best time </span> ever!
</p>
```

  - Use to apply a document style sheet to its content

```
<style type = "text/css">?
.bigred {font-size: 24pt;
         font-family: Ariel; color: red;
 </style>
<p>
Now is the
<span class = "bigr
 best time </span> ever!
 </p>
```

- **Another tag that is useful for style specifications: **
  - **Used to create document sections (or divisions) for which style can be specified.**
    - **e.g., A section of five paragraphs for which you want some particular style.**

```
<div class = “primary”>
  <p>
    ……..
  </p>
  <p>
    ……..
  </p>
  <p>
  ……
  </p>
</div>
```

Conflict Resolution

- **When two or more rules apply to the same tag there are rules for deciding which rule applies.**

- **Document level:**
  - **In-line style sheets have precedence over document style sheets,**
  - **Document style sheets have precedence over external style sheets.**

- **Within the same level there can be conflicts:**
  - **A tag may be used twice as a selector.**
  - **A tag may inherit a property and also be used as a selector.**

- **Style sheets can have different sources**
  - **The author of a document may specify styles.**
  - **The user, through browser settings, may specify styles.**

- **Individual properties can be specified as important.**

Precedence Rules

- **From highest to lowest.**

1. **Important declarations with user origin.**
2. **Important declarations with author origin.**
3. **Normal declarations with author origin.**
4. **Normal declarations with user origin.**
5. **Any declarations with browser (or other user agent) origin.**

Tie-Breakers

- **Specificity**
  1. **id selectors**
  2. **Class and pseudo-class selectors**
  3. **Contextual selectors**
  4. **General selectors**
- **Position**
  - **Essentially, later has precedence over earlier**