

Web Programming

Lecture 2 – Cascading Style Sheets

Introduction

- HTML (and XHTML) concern themselves primarily with content, not presentation concerns.
- Many tags have built-in default values for certain formatting concerns, e.g., `<h2>` has a default setting for font size.
- Style concerns is handled much better by Cascading Style Sheets.

Levels of Style Sheets

There are three different levels of style sheets:

- Inline
 - these have style information contained within the block tag.

```
<p style="font-size: 1em; color: fuchsia">Aren't style sheets wonderful?</p>
```
- Document - style information is embedded within the block tag **<style>**
- External – style information is contained in a separate file.

Inline Style Specifications

- Inline style specifications are written as values of the style attribute for the tag in which they appear:

```
style = "property1: value1;  
property2: value2, ..."
```

Document-Level Styles

- The style information is embedded within a block tag:

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

- Comments are enclosed between /* and */:

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

```
  /* Style for initial
```

```
     paragraph */
```

```
  /* Style for other paragraphs */
```

```
</style>
```

Style Specification Formats

- Each style rule has two parts:
 - A selector, which indicates the tag or tags affected by the rule
 - A list of property/value pairs

- General syntax:

```
selector { property1: value1;  
           property2 : value2; ... }
```

- If the property has more than one value, it is usually separates by blank space (or sometimes commas).

Selector Forms

- Selectors can take a variety of forms
 - Selectors can be simple, such as a single element name or can apply only when certain combinations of tags are used.
 - Selectors can be classes defined by the user.

Simple Selector Forms

- The simplest selector is for a single element, such as `h1`:
`h1 { font-size: 1.5em; }`
- It can also apply to more than one element if necessary:
`h2, h3 { font-size: 1.25em; }`
- Selector can also specify styles that apply only to elements in certain positions.
`body b i { font-family: Courier; }`
- If italics were outside boldface, this would not be applied

Class Selectors

- Users can define classes that enable different occurrences of the same tags to be formatted differently:

```
p.normal { font-size: 100%;}  
p.narrow {font-size: 80%;}
```

- It could be used to distinguish between these two paragraphs:

```
<p class="normal">This is a test of the  
  emergency styling system.</p>
```

```
<p class = "narrow">If this were a real  
  emergency, you would be in big  
  trouble!</p>
```

Generic Selectors

- Sometimes it is convenient to have a class of style specifications that applied to more than one kind of tag. These are called *generic classes*.
- You declare it by writing:

```
.navbar {  
  background-color: #881010;  
  font-family:  
    verdana,arial,sans-serif;  
  font-size: small; }  
You can use it by writing:
```

```
<p class= "navbar">Some stuff</p>
```

id Selectors

- An id selector applies a style to one specific element. The general form is:
`#specific-id {property-value list}`
- Example
`#specific14 {font-size: 110%}`
- Usage
`< h2 id = "specific14"> 1.4 Calico
Cats</h2>`

Pseudo Classes

- Pseudo classes are styles that apply when certain events take place.
- There is not always support for pseudo classes in all browsers.

Pseudo-Classes: An Example

```
<!DOCTYPE html>

<!-- pseudo.html
  Illustrates the :hover and :focus pseudo classes
-->

<html lang = "en">
  <head> <title> Checkboxes </title> </head>
  <meta charset="utf-8">
  <style type = "text/css" >
    input:hover {color: red;}
    input:focus {color: green;}
  </style>
```

```
<! - hover is when the mouse hovers
  over the element ->
<! - focus is when the mouse is clicked
  while hovering over the element ->
<body>
  <form action ="">
    <p>
      Your name:
      <input type = "text" />
    </p>
  </form>
</body>
</html>
```

Property Value Forms

- The original standard for Cascading Style Sheets (CSS1) includes 60 different properties in seven categories:
 1. Fonts
 2. Lists
 3. Text Alignment
 4. Margins
 5. Colors
 6. Backgrounds
 7. Border

Keyword Property Values

- Keyword property values are used when there is a limited number of possible values and they are defined.
- Examples of this include small, medium and large.
- Keyword property values are case-insensitive.

Numerical Property Values

- Number values are used when there aren't meaningful units that can be attached to a number property value.
- A number property value can be:
 - integer
 - string of digits with a decimal point
 - can be preceded by a + or – sign.

Length Property Values

- Length property values are numbers followed by a 2-letter abbreviation for a unit of length (without a space in-between).
- The units of length are:

- px	pixels	- in	inches
- pt	points	- cm	centimeters
- pc	picas	- mm	millimeters
- There are also two relative lengths
 - em (height of an "m")
 - ex (height of an "x")

Percentage Property Values

- Percentage property values consist of a number immediately followed by a "%".
- The size of a percentage of the previously used size (whatever that may be).

URL Property Values

- URL property values are different from references to URLs.
- URL values consist of the word url immediately followed by a URL within parentheses.
- Example
`url(tetons.jpg)`

Color Property Values

- Colors can be specified in three different ways:
 - Color Name (e.g., **white**)
 - rgb form (e.g., **rgb(255, 255, 255)**)
 - Six-digit hexadecimal number (e.g., **#FFFFFF**)
- These all represent the same color, i.e., white.

Inherited Property Values

- The CSS2 standard specifies that some property values can be inherited by elements within elements with specified values.
- **Background-color** cannot be inherited but **font-size** can be.
- This is not well-supported at the present.

Font Properties

- Fonts have several properties that may be specified:
 - Font Families
 - Font Sizes
 - Font Styles
 - Font Weights
 - Font Shorthands
 - Text Decoration

Font Families

- **font-family** is used to specify a list of fonts.
- The first font on the list supported by the browser is the one used to display the text.
font-family: Arial, Helvetica, Futura, sans-serif
- If the font's name contains one (or more) blank spaces, place it in single quotes (double quotes are sometimes used to specify the entire style).
font-family: "Times New Roman"

Generic Fonts

<u>Generic Name</u>	<u>Examples</u>
<code>serif</code>	Times New Roman, Garamond
<code>sans-serif</code>	MS Ariel, Helvetica
<code>cursive</code>	Caflisch Script, Zapf-Chancery
<code>fantasy</code>	Critter, Cottonwood
<code>monospace</code>	Courier, Prestige

Font Sizes

- Font size can be specified in points
`font-size: 10pt`
- Font size can also be specified using predefined names
`xx-small` `large`
`x-small` `x-large`
`medium` `xx-large`
- Absolute sizing gives complete size control to the web designer but is problematic when accessibility is considered.

Font Styles

- The most commonly used font style is italics
`font-style: italics`
- Oblique is also used on occasion, but it is not as widely supported as italics; consequently, some browsers display oblique text in italics.

Font Weights

- `font-weight` is most commonly used to specify boldface type.
- There are four named values:
`normal` (default) `bold`
`lighter` (relative) `bolder` (relative)
- Number values may be used.
 - Multiples of 100 from 100 to 900.
 - `normal` is 400.
 - `bold` is 700.

Font Shorthands

- Sometimes it's handy to be able to specify several font properties at once.
- This can be done by specifying font.
- Example
`font: bold 1.1em 'Times New Roman' Palatino`
- The order in which properties are specified is required: **font-style**, **font-weight**, **font-size** and **font-family**.
- Only font size and font family are required

`fonts.html`

```
<!DOCTYPE html>

<!-- fonts.html
  An example to illustrate font properties
-->
<html lang = "en">
  <head> <title> Font properties </title>
  <meta charset="utf-8">
  <style type = "text/css">
    p.major {font-size: 1.1em;
             font-style: italic;
             font-family: 'Times New Roman'; }
    p.minor {font: 0.9em bold 'Courier New';}
    h2 {font-family: 'Times New Roman';
        font-size: 2em; font-weight: bold}
```

```
    h3 {font-family: 'Courier New';
        font-size: 1.5em;}
</style>
</head>

<body>
  <p class = "major">
    If a job is worth doing, it's worth doing
    right.
  </p>
  <p class = "minor">
    Two wrongs don't make a right, but they
    certainly can get you in a lot of trouble.
  </p>

  <h2>Chapter 1 Introduction</h2>
  <h3> 1.1 The Basics of Computer Networks</h3>
</body>
</html>
```

fonts.html – As Displayed

If a job is worth doing, it's worth doing right.

Two wrongs don't make a right, but they
certainly can get you in a lot of trouble.

Chapter 1 Introduction

1.1 The Basics of Computer Networks

fonts2.html

```
<!DOCTYPE html>

<!-- fonts2.html
  An example to illustrate font properties
-->

<html lang = "en">
  <head> <title> Font properties </title>
    <meta charset="utf-8">
    <link rel="stylesheet" type = "text/css"
      href = "styles.css">
  </head>
```

```
<body>
  <p class = "big">
    If a job is worth doing, it's worth doing
    right.
  </p>

  <p class = "small">
    Two wrongs don't make a right, but they
    certainly can get you in a lot of trouble.
  </p>

  <h2>Chapter 1 Introduction</h2>
  <h3> 1.1 The Basics of Computer Networks</h3>
</body>
</html>
```

styles.css

```
/* styles.css - an external style sheet
   for use with fonts2.html
*/

p.major {font-size: 1.1em;
font-style: italic;
font-family: 'Times New Roman';
}
p.minor {font: 0.9em bold 'Courier New';}
h2 {font-family: 'Times New Roman';
font-size: 2em; font-weight: bold}

h3 {font-family: 'Courier New';
font-size: 1.5;}
```

Text Decoration

- Text decoration refers to other features not otherwise specified.
- The values include **line-through**, **underline** and **overline**, and **none** (the default).
- **underline** is problematic because it can be confused with links.

decoration.html

```
<!DOCTYPE html>

<!-- decoration.html
  An example that illustrates several of the
  possible text decoration values
-->

<html lang = "en">
  <head> <title> Text decoration </title>
    <meta charset = "utf-8">
    <style type = "text/css">
      p.through {text-decoration: line-through}
      p.over    {text-decoration: overline}
      p.under   {text-decoration: underline}
    </style>
  </head>
```

```
<body>
  <p class = "through">
    This illustates line-through
  </p>

  <p class = "over">
    This illustates overline
  </p>

  <p class = "under">
    This illustates underline
  </p>

</body>
</html>
```

`decoration.html` – As Displayed

This illustrates ~~line-through~~

This illustrates overline

This illustrates underline

Text Spacing

- There are three properties that control the spacing of text within an HTML document:
- Letter spacing
- Word spacing
- Leading

Letter spacing

- Letter spacing refers to the amount of space between letters within a word.
 - This spacing is referred to as *tracking*.
- Positive values increase spacing; negative values decrease spacing.
- Examples
 - `letter_spacing: 1px;` spreads the letters
 - `letter_spacing: -1px;` squeezes the letters
 - `letter_spacing: normal;`
 resets to the spacing of the parent element

Word spacing

- Word spacing controls the amount of spaces between words.
- As with letter spacing, a positive value increases spacing, a negative value decreases spacing and a normal value resets it to the word spacing property of the parent element.

Leading

- Leading is the space between lines of text.
- Leading can be controlled by the **line-height**
- The value of **line-height** can be a number (referring to the number of lines of spacing) or normal (resetting it the **line-height** property of the parent tag).

`text_space.html`

```
<!DOCTYPE html>

<!-- test_space.html
     An example that illustrates text spacing
     properties
-->

<html lang = "en">
  <head> <title> Text spacing properties </title>
  <meta charset = "utf-8">
```

```
<style type = "text/css">
  p.big_tracking    {letter-spacing: 0.4em;}
  p.small_tracking {letter-spacing: -0.08em;}
  p.big_between_words {word-spacing: 0.4em;}
  p.small_between_words {word-spacing: -0.1em;}
  p.big_leading {line-height: 2.5;}
  p.small_leading {line-height: 1.0;}
</style>
</head>
```

```
<body>
  <p class = "big_tracking">
    On the planes of hesitation
    [letter-spacing: 04em;]
  </p> <p />

  <p class = "small_tracking">
    Bleach the bones of countless millions
    [letter-spacing: -0.08em]
  </p> <br />

  <p class = "big_between_words">
    Who at the dawn of victory
    [word-spacing:0.4em]
  </p> <p />
```

```
<p class = "small_between_words">
  Sat down to wait and waiting died
  [word_pacing: -0.1em]
</p> <br />
```

```
<p class = "big_leading">
  If you think that CSS is simple,
  [line-height: 2.5] <br />
  You are quite mistaken
</p> <br />
```

```
<p class = "small_leading">
  If you think that HTML5 is all old stuff,
  [line-height: 1.0] <br />
  You are quite mistaken
</p>

</body>
</html>
```


List Properties

- The two list properties that are most frequently specified are:
 - the shape of bullets (on unordered lists)
 - the sequencing values (on ordered lists).

list-style-type

- **list-style-type** determines the shape that is used as a bullet for unordered (or bulleted) lists.
- There are four values:
 - **disc** (filled-in circle)
 - **circle** (empty circle)
 - **square**
 - **none**
- The default value is **disc**.

bullets1.html

```
<? xml version = "1.0" encoding "utf-8"?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/XHTML11.dtd"?

<!-- bullets1 -->
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Text decoration </title>
    <style type = "text/css">
      ul {list-style-type: square}
    </style>
  </head>
  <body>
    <h3> Some Common Single-Engine Aircraft </h3>
    <ul>
      <li> Cessna Skyhawk </li>
      <li> Beechcraft Bonanza </li>
      <li> Piper Cherokee </li>
    </ul>
  </body>
</html>
```

bullets1.html – as Displayed

Some Common Single-Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- Piper Cherokee

bullet2.html

```
<!DOCTYPE html>

<!-- bullets2 -->

<html lang = "en">
  <head> <title> Text decoration </title>
    <meta charset = "utf-8">
    <style type = "text/css">
      li.disc { list-style-type: disc}
      li.square { list-style-type: square}
      li.circle { list-style-type: circle}
    </style>
  </head>
```

```
<body>
  <h3> Some Common Single-Engine Aircraft </h3>
  <ul>
    <li class = "disc"> Cessna Skyhawk </li>
    <li class = "square"> Beechcraft Bonanza </li>
    <li class = "circle"> Piper Cherokee </li>
  </ul>
</body>
</html>
```

bullet2.html – as Displayed

Some Common Single-Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- o Piper Cherokee

Using Images as Bullets

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

Sequencing Values in Ordered Lists

<u>Property Values</u>	<u>Sequence Type</u>	<u>First Four Values</u>
decimal	Arabic numerals	1, 2, 3, 4
decimal-leading-zero	Arabic numerals beginning with 0	0, 1, 2, 3
upper-alpha	Uppercase letters	A, B, C, D
lower-alpha	Lowercase letter	a, b, c, d
upper-roman	Uppercase Roman numerals	I, II, III,IV
lower-roman	Lowercase Roman numerals	I, ii, iii, iv

<u>Property Values</u>	<u>Sequence Type</u>	<u>First Four Values</u>
lower-greek	Lowercase Greek letters	α , β , γ , δ
upper-latin	Same as upper-alpha	A, B, C, D
lower-latin	Same as lower-alpha	a, b, c, d
upper-roman	Uppercase Roman numerals	I, II, III,IV
lower-roman	Lowercase Roman numerals	I, ii, iii, iv
armenian	Traditional Armenian numbering	Ա, Բ, Գ, Դ
georgian	Traditional Georgian numbering	ⴁ, ⴂ, ⴃ, ⴄ
none	No bullet	

sequence_types.html

```
<!DOCTYPE html>

<!-- sequence_types.html
  An example to illustrate sequence type styles
-->

<html lang = "en">
  <head> <title> Sequence Types </title>
    <meta charset = "utf-8">
    <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>
  </ol>
```

```
<li> Commercial Aviation (jet engine)
<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>
```

Indenting Text

- The **text-indent** property can be used to indent the first line of a paragraph.

indents.html

```
<style type = "text/css">
  p.indent {text-indent: 2em}
</style>
</head>

<body>
  <p class = "indent">
    Now is the time for all good Web programmers to begin
    using cascading style sheets for all presentation
    detail in their documents. No more deprecated
    tags and attributes, just nice, precise style
    sheets.
  </p>
```

indents.html – As Displayed

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

Alignment of Text

- The `text-align` property allows text to be left- or right-justified as well as centered or fully justified.
- Left-justified is the default:
- Examples

```
p { text-align: right}
p { text-align: left}
p { text-align: center}
p { text-align: justify}
```

float Property

- The **float** property is used to specify that text will flow around the image (or table or some other element).
- The default value is **none**.
- Other values are **left** and **right**.

float.html

```
<!DOCTYPE html>

<!-- float.html
      An example to illustrate the float property
-->
<html lang = "en">
  <head> <title> Font properties </title>
    <style type = "text/css">
      img {float: right}
    </style>
  </head>
  <body>
    <p>
      
    </p>
```

```
<p>
  This is a picture of a Cessna 210. The 210 is
  the flagship single-engine Cessna aircraft.
  Although the 120 began as a four-place
  aircraft, it soon acquired a third row of
  seats, stretching it to a six-place plane.
  The 210 is classified as a high-performance
  airplane, which means its landing gear is
  retractable and its engine has more than 200
  horsepower. In its first model year, which
  was 1960, the 210 was powered by a 260-
  horsepower fuel-injected six-cylinder engine
  that displaced 471 cubic inches.
  The 210 is the fastest single-engine airplace
  ever built by Cessna.
</p>
</body>
</html>
```

float.html – As Displayed

This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft.

Although the 120 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane.

The 210 is classified as a high-performance airplane, which means its landing gear is retractable and its engine has more than 200

horsepower. In its first model year, which was 1960, the 210 was powered by a 260-horsepower fuel-injected six-cylinder engine that displaced 471 cubic inches. The 210 is the fastest single-engine airplane ever built by Cessna.



Color

- Color is not a simple issue, largely because of older browsers and older systems running those browsers.
- Colors may be displayed by browsers in ways that the designer did not intend.
- Some colors have names.
- All colors can be defined by specifying the amount of red, green and blue as 2-digit hexadecimal value for each color.

Color Groups

- There are three separate collections of colors that go from more restrictive to least restrictive.
- They are:
 - Named colors
 - Web palette (Web-safe colors)
 - 24-bit colors

Named Colors

<u>Name</u>	<u>Hexadecimal code</u>	<u>Name</u>	<u>Hexadecimal code</u>
Aqua	#00FFFF	Navy	#000080
Black	#000000	Olive	#808000
Blue	#0000FF	Purple	#800080
Fuchsia	#FF00FF	Red	#FF0000
Gray (Grey)	#808080	Silver	#C0C0C0
Green	#008000	Teal	#008080
Lime	#00FF00	White	#FFFFFF
Maroon	#800000	Yellow	#FFFF00

Other Named Colors

- In addition to these 17, there are 130 other colors recognized by most modern browsers.

- They include:

AliceBlue	Bisque	BurlyWood
AntiqueWhite	Black	CadetBlue
Aqua	BlanchedAlmond	Chartreuse
Aquamarine	Blue	Chocolate
Azure	BlueViolet	Coral
Beige	Brown	CornflowerBlue

Web Palette

- The Web palette includes 216 colors, which were considered all displayable by Windows- and Macintosh-based browsers. For this reason, they are called **Web-safe colors**.
- Given that modern are no longer as restricted in the colors that they display, web-safe colors never became part of the standard for HTML or CSS.
- These colors restrict the hexadecimal values for red, green and blue to 00, 33, 66, 99, CC, and FF. This limits the increments for all three colors to 20%.

The Full Range of 16 millions Colors

- With six hexadecimal digits (two for red, 2 for yellow, 2 for blue), this leaves over 16 million colors that can be defined.
- This used to be problematic because not all of these colors could be accurately and consistently displayed in a browser window or on all monitors.
- This is no considered to be longer the case.

Color Properties

- The color property specifies the foreground color for an HTML element.
- The background-color property sets the background color for an element.

colors.html

```
<style type = "text/css">
  th.red {color: red}
  th.orange {color: orange}
</style>
... ..
<table border = "1 px">
  <tr>
    <th class = "red"> Apple </th>
    <th class = "orange"> Orange </th>
    <th class = "orange"> Screwdriver </th>

  </tr>
</table>
```

colors.html – As Displayed

Apple	Orange	Screwdriver
-------	--------	-------------

colors2.html

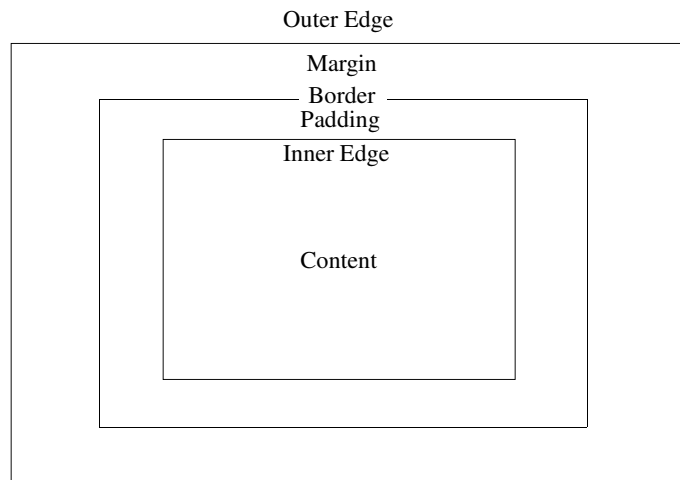
```
<style type = "text/css">
  p.redback {font-size: 1.5em; color: blue;
             background-color: red;}
</style>
</head>

<body>
  <p class = "redback">
    To really make it stand out, use a red
    background!
  </p>
</body>
```

colors2.html – As Displayed

To really make it stand out, use a red background!

The Box Model



Borders

- Each element has a **border-style** property, which control whether the elements content has a border and what style that border has.
- CSS1 only requires a **border-style** of solid.
- CSS2 provides several styles, including **dotted**, **dashed** and **double**.
- The default value for border-style is **none**.

border-width

- **border-width** specifies how thick the line must be.
- Values include **thin**, **medium**, **thick** of a length in pixels.
- **medium** is the default value.
- The four sides can have different values by specifying **border-top-width**, **border-left-width**, etc.

borders.html

```
<!DOCTYPE html >
<!-- borders.html
      An example of a simple table with various
      borders
-->
<html lang = "en">
  <head> <title> Font properties </title>
  <meta charset = "utf-8" />
  <style type = "text/css">
    table {border-top-width: medium;
           border-bottom-width: thick;
           border-right-width: thin;
           border-top-color: red;
           border-bottom-color: blue;
           border-left-color: green;
           border-top-style: dotted;
           border-top-style: dashed;
          }
  
```

```
    p {border-style: dashed; border-width: thin;
      border-color: green;
    }
  </style>
</head>

<body>
  <table border = "5">
    <tr>
      <th> </th>
      <th> Apple </th>
      <th> Orange </th>
      <th> Screwdriver </th>
    </tr>
```

```
    <tr>
      <th> Breakfast </th>
      <td> 0 </td>
      <td> 1 </td>
      <td> 0 </td>
    </tr>

    <tr>
      <th> Lunch </th>
      <td> 1 </td>
      <td> 0 </td>
      <td> 0 </td>
    </tr>
```

```

    <tr>
      <th> Dinner </th>
      <td> 0 </td>
      <td> 0 </td>
      <td> 1 </td>
    </tr>
  </table>

  <p>
    Now is the time for all good Web programmer to
    learn to use style sheets.
  </p>
</body>
</html>

```

borders.html – As Displayed

	Apple	Orange	Screwdriver
Breakfast	0	1	0
Lunch	1	0	0
Dinner	0	0	1

Now is the time for all good Web programmer to learn to use style sheets

Margin and Padding

- Padding is the space between the content and its border.
- The margin is the space between the border and the adjacent element.
- Without a border, the margin and padding is the space between elements, although the elements may have different backgrounds.

Margin Properties

- The property **margin** is the margin spacing around the element.
- Individual sides of the margin can be specified using **margin-top**, **margin-bottom**, **margin-left** and **margin-right**.
- They are assigned length values.

Padding Properties

- The property **padding** is the margin spacing around the element.
- Individual sides of the padding can be specified using **padding-top**, **padding-bottom**, **padding-left** and **padding-right**.
- They are assigned length values.

marpads.html

```
<!DOCTYPE html>

<!-- marpads.html
      An example to illustrate margins and padding
-->
<html lang = "en">
  <head> <title> Borders and Margins </title>
  <style type = "text/css">
    p.one {margin: 15px;
           padding: 15px;
           background-color:#C0C0C0;
           border-style: solid;
           }
  </style>
</html>
```

```
p.two {margin: 5px;
padding: 25px;
background-color:#C0C0C0;
border-style: solid;
}

p.three {margin: 25px;
padding: 5px;
background-color:#C0C0C0;
border-style: solid;
}

p.four {margin: 25px;
background-color:#C0C0C0;
}
```

```
p.five {padding: 25px;
background-color:#C0C0C0;
}

</style>
</head>

<body>
<p>
Here is the first line.
</p>

<p class = "one">
Now is the time for all good Web programmer to
learn to use style sheets.<br /> [margin =
15 px, padding = 15 px]
</p>
```

```
<p class = "two">
  Now is the time for all good Web programmer to
  learn to use style sheets.<br /> [margin =
  5 px, padding = 25 px]
</p>

<p class = "three">
  Now is the time for all good Web programmer to
  learn to use style sheets.<br /> [margin =
  25 px, padding = 5 px]
</p>

<p class = "four">
  Now is the time for all good Web programmer to
  learn to use style sheets.<br /> [margin =
  25 px, no padding, no border]
</p>
```

```
<p class = "five">
  Now is the time for all good Web programmer to
  learn to use style sheets.<br /> [padding =
  25 px, no margin, no border]
</body>
</html>
```


Background Images

- The **background-image** property can be used to place an image in the background of an element.
- The other relevant properties are:
 - **background-repeat**, whose values may include **repeat** (the default), **repeat-x**, **repeat-y** and **no-repeat**
 - **background-position**, whose values can be **top**, **bottom**, **left**, **right** or **center**. If only one value is used, the other is assumed to be **center** (e.g., **top center** or **center left**)

back-image.html

```
<!DOCTYPE html>

<!-- back-image.html
      An example to illustrate the background images
-->
<html lang = "en">
  <head> <title> Background images </title>
    <meta charset = "utf-8">
    <style type = "text/css">
      body {background-image: url(c210.jpg);
            background-size: 375px 300 px}
      p {margin-left: 30px; margin-right: 30px;
        margin-top: 50px; font-size: 1.1em;}
    </style>
  </head>
  <body>
    <p>
```

The Cessna 172 is the most common general aviation airplane in the world. It is an all-metal, single-engine piston, high-wing four-place monoplane. It has fixed-gear and is categorized as a non-high-performance aircraft. The current model is the 172R.

The wingspan of the 172R is 36'1". Its fuel capacity is 56 gallons in two tanks, one in each wing. The takeoff weight is 2,450 pounds. Its maximum useful load is 837 pounds. The maximum speed of the 172R at sea level is 142 mph. The plane is powered by a 360 cubic inch gasoline engine that develops 160 horsepower. The climb rate of the 172R at sea level is 720 feet per minute.

```
</p>  
</body>  
</html>
```

**** and **<div>** Tags

- These tags both allow text to have a different format than the text around it.
- **** is meant to distinguish a word or a phrase.
- **<div>** is intended to distinguish one or more paragraphs.
- **<div>** can have formatting attributes of its own, unlike ****, but this is discouraged in favor of style sheets.

span.html

```
<!DOCTYPE html>

<!-- span.html
     An example to illustrate the span tag
     -->
<html lang = "en">
  <head> <title> Background images </title>
    <meta charset = "utf-8">
    <style type = "text/css">
      .spanred {font-size: 2em;
                font-family: Ariel; color: red}
    </style>
  </head>
```

```
<body>
  <p>
    It sure is fun to be in <span class =
    "spanred"> total </span> control of text.
  </p>
</body>
</html>
```

divs.html

```
<!DOCTYPE html>

<!-- colors.html
      An example to illustrate color properties
-->
<html lang = "en">
  <head> <title> Font properties </title>
  <meta charset = "utf-8">
  <style type = "text/css">
    .indent {text-indent: 25px;
             font-family: Helvetica}
  </style>
</head>
```

```
<body>
  <div class = "indent">
    <p>
      The Cessna 172 is the most common general
      aviation airplane in the world. It is an all-
      metal, single-engine piston, high-wing four-
      place monoplane. It has fixed-gear and is
      categorized as a non-high-performance
      aircraft. The current model is the 172R.
    </p>

    <p>
      The wingspan of the 172R is 36'1". Its fuel
      capacity is 56 gallons in two tanks, one in
      each wing. The takeoff weight is 2,450
      pounds. Its maximum useful load is 837
      pounds. The maximum speed of the 172R at sea
```

```
level is 142 mph. The plane is powered by a
360 cubic inch gasoline engine that develops
160 horsepower. The climb rate of the 172R
at sea level is 720 feet per minute.
</p>
</div>
<p>
  Now is the time for all good Web programmers
  to begin using cascading style sheets for all
  presentation detail in their documetns. No
  more deprecated tags and attributes, just
  nice, precise style sheets.
</p>
</body>
</html>
```

Conflict Resolution

- Conflict can happen when there re two different values for the same property on the same element in a document.
- Order of precedence:
 - Inline style sheets take precedence over document and external style sheets
 - Document style sheets take precedence over external style sheets.

cascade.html

```
<!DOCTYPE html>
<html lang = "en">

  <head>
    <title> An example to illustrate the three
      levels </title>
    <meta charset = "utf-8">
    <link rel = "stylesheet" type = "text/css"
      href = "cstyle.css" />
    <style type = "text/css">
      p.docstyle {font-size: 1.2em;}
    </style>
  </head>
```

```
<body>
  <p>
    Now is the time
  </p>

  <p class = "docstyle">
    for all good men
  </p>

  <p class = "docstyle"
    style = "font-size: 1.6em">
    to come to the aid of their country
  </p>
</body>
```

`cstyle.css`

```
/* cstyle.css - an external style sheet
   for use with cascade.html
*/

p { font-size: 0.8em;}
```

How `cascade.html` Appears

Now is the time
for all good men
to come to the aid of their country