



# DHTML and Cascading Style Sheets (CSS)



# Introduction to DHTML

- What is DHTML?
  - – Acronym for the term "Dynamic HTML".
  - – Combination of HTML, Cascading Style Sheet and some scripting language.
  - – Makes the pages more dynamic.
- Features of DHTML
  - – Provides more control over the appearance, layout and behavior of the web page.
  - – New event model.
- Provides a wide set of event handlers

# Cascading Style sheets

## ■ Features

- – Separates the Presentation and Contents of the HTML Document.
- – Provide numerous attributes to create dynamic effects.
- – Simple.
- – Reusable.

## ■ Style Sheet

- – A set of statements that specify presentation of a document.
- – A powerful mechanism for adding styles.
- – Styles can be assigned by the **<STYLE> </STYLE>** tag.



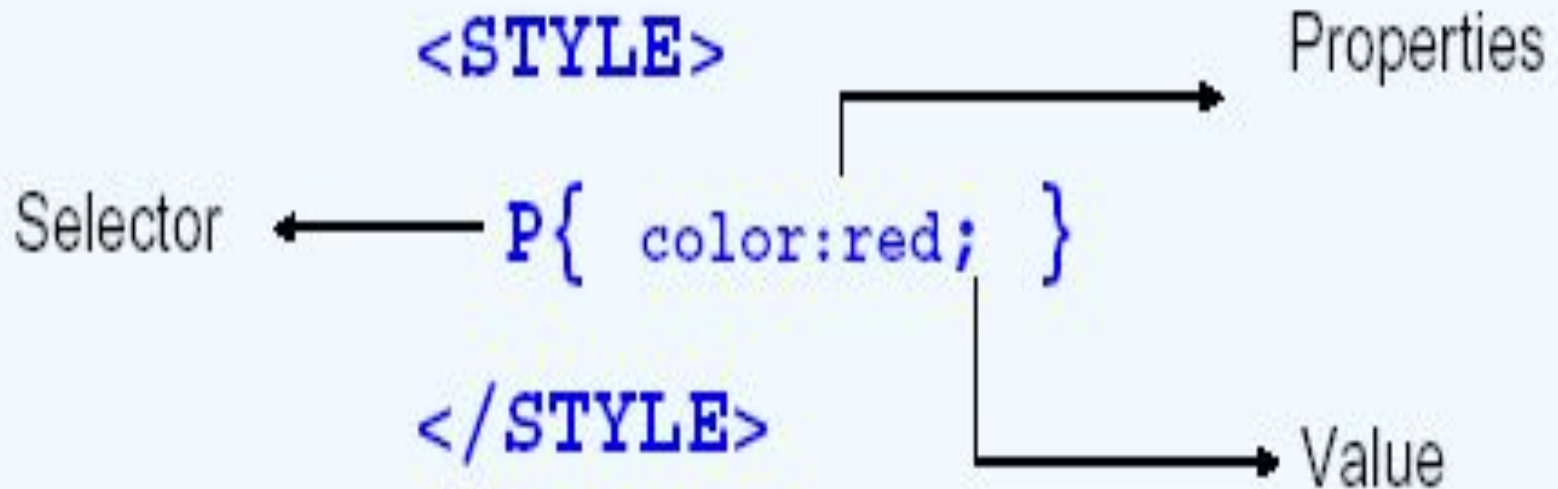
# Advantages

- Good control over the presentation.
- Consistency : A Standard flow, look & feel can be maintained for all pages of a Web Site
- Ability to make global changes to all the documents from a single location.
- Reduces the time spent on maintaining HTML Document

# How do Style Sheets Work?

- Separate Section is defined to place the Style Properties of the Document
    - – Style Rule is composed of two parts
      - **Selectors:** To which the rule is applied
      - **Declarations**
        - Property
        - Value
- TH{color: red; }.**
- Defined sections in the document are attached with their respective properties
  - **<STYLE >** tag is used to define styles.

# EXAMPLE



CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets.



# Selector Types

- Class Selector
- Id Selector

In addition to setting style for a HTML element, CSS allows you to specify your own selectors called “id” and “class”

# Class selector

- The class selector is used to specify a style for a group of element. The class selector is most often used on several elements.
- This allow you to set a particular style for any HTML element with the same class.
- This class selector uses the HTML class attribute and is defined with a “.”
- In the example below, all HTML elements with class =“center” will be center-aligned.

## Example

```
.center  
{  
text-align:center;  
}
```



# Class selector

- Different styles for same HTML tag.
- **P.right {text-align: right}**
- **P.center {text-align: center}**
- Class attribute in your HTML document:
- **<P CLASS="right">This paragraph will be right-aligned. </P>**
- If you omit the tag name in the selector this style can be applied to all the tags
- **.center {text-align: center}**

# The id Selector

- The id selector is used to specify a style for a single, unique element.
- The id selector is defined as a #.
- The style rule below will match the element that has an id attribute with a value of "green":
  - `#green {color: green}`
- The style rule below will match the p element that has an id with a value of "para1":
  - `p#para1 { text-align: center; color: red }`
- Do **NOT** start an ID name with a number! It will not work in Mozilla/Firefox.



# Style types

- Inline Styles
- Internal Stylesheets
- External Stylesheets

# Inline styles

- **`<P style="color=blue; margin-right: 10px">`**
- **Styled paragraph**
- **`</P>`**
- Can be applied to a single occurrence of an element
- Losses a lot of advantages of style sheets
- Mixes content with presentation

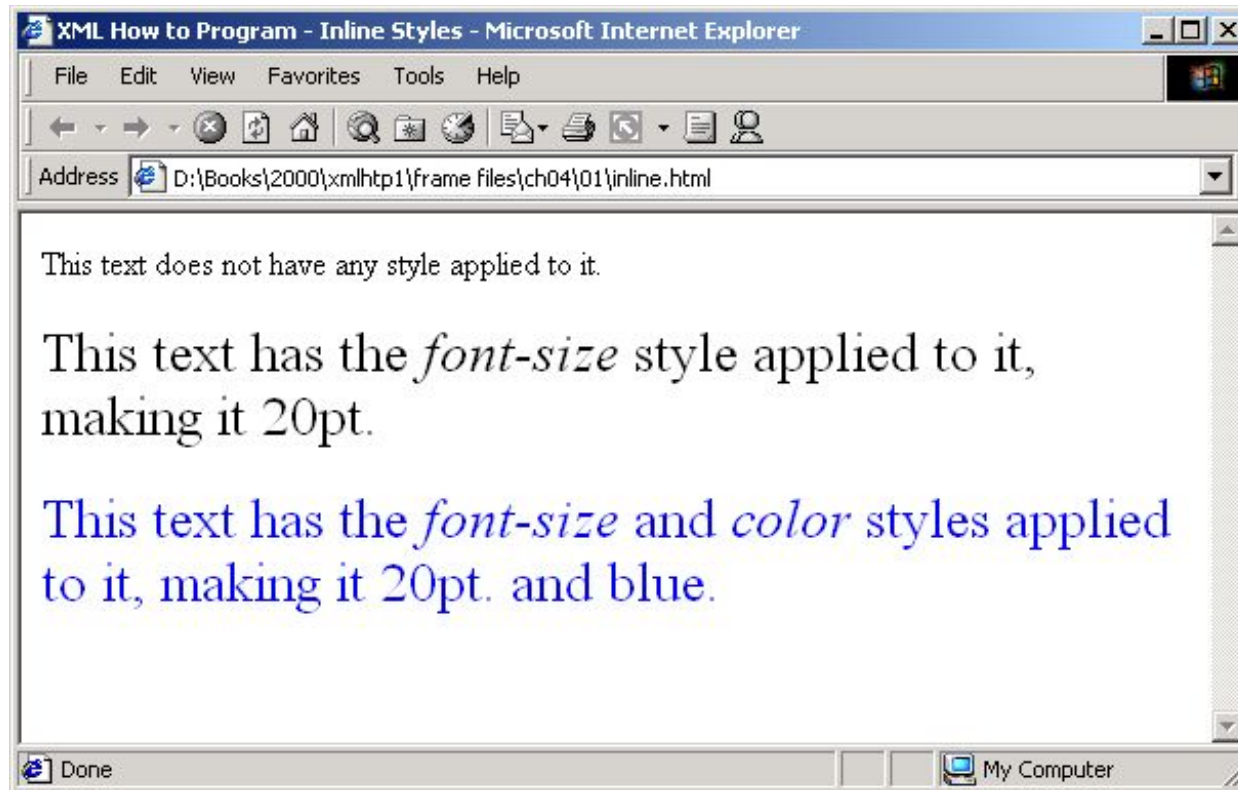
```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig. 4.1: inline.html -->
6  <!-- Using inline styles -->
7
8  <head>
9      <title>XML How to Program - Inline Styles</title>
10 </head>
11
12 <body>
13
14     <p>This text does not have any style applied to it.</p>
15
16     <!-- The style attribute allows you to declare inline -->
17     <!-- styles. Separate multiple styles with a semicolon. -->
18     <p style = "font-size: 20pt">This text has the
19 <em>font-size</em>
20     style applied to it, making it 20pt.</p>
21
22     <p style = "font-size: 20pt; color: #0000ff">This text has
23     <em>font-size</em> and <em>color</em> styles applied to it,
24     making it 20pt. and blue.</p>
25
26 </body>
27 </html>
```

Define style for following text

the

# Output

Example



Back

# Internal styles sheet

- Internal styles sheet also known as embedded style
- Can be used by single document.
- Enclosed within the HEAD tag.

**<HEAD>**

**<!--<STYLE TYPE="text/css">**

**HR{color:blue}**

**P{margin-right:10px}**

**</STYLE> -->**

**</HEAD>**

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig 4.3: advanced.html      -->
6  <!-- More advanced style sheets -->
7
8  <head>
9      <title>XML How to Program - More Styles</title>
10
11     <style type = "text/css">
12
13         a.nodect { text-decoration: none }
14
15         a:hover  { text-decoration: underline;
16                     color: red;
17                     background-color: #ccffcc }
18
19         li em    { color: red;
20                     font-weight: bold }
21
22         ul       { margin-left: 75px }
23
24         ul ul    { text-decoration: underline;
25                     margin-left: 15px }
26
27     </style>
28 </head>

```

Assign attribute **nodect** to all **a** elements (override default underline attribute of element **a**)

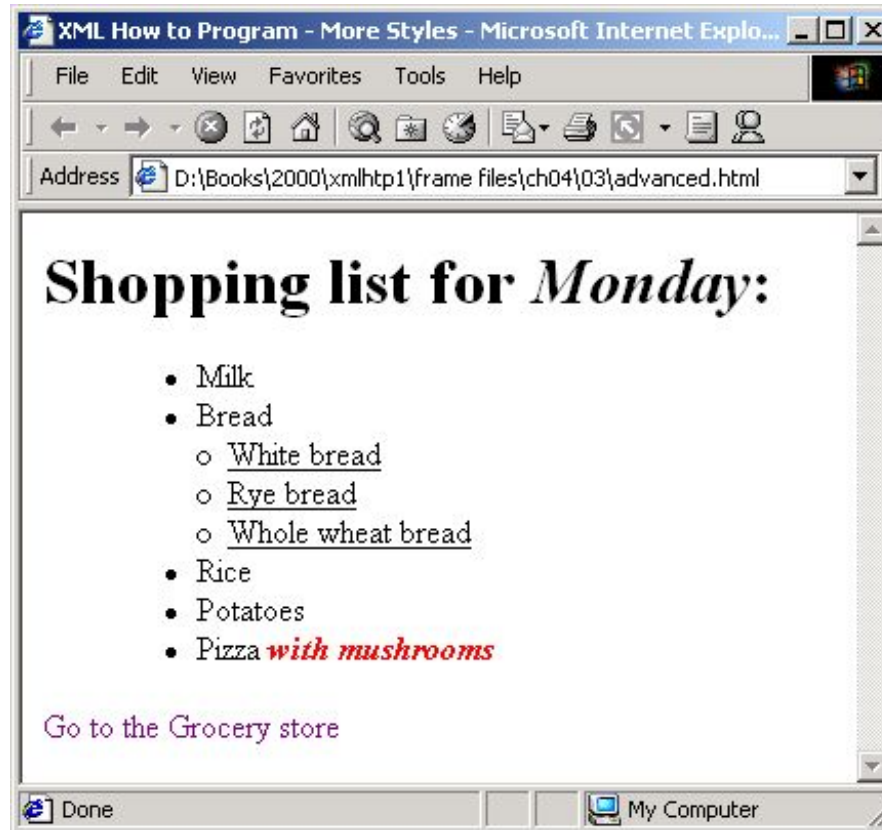
Define style for any **em** element contained in **li** tag



```
30  <body>
31
32    <h1>Shopping list for <em>Monday</em>:</h1>
33
34    <ul>
35      <li>Milk</li>
36      <li>Bread
37        <ul>
38          <li>White bread</li>
39          <li>Rye bread</li>
40          <li>Whole wheat bread</li>
41        </ul>
42      </li>
43      <li>Rice</li>
44      <li>Potatoes</li>
45      <li>Pizza <em>with mushrooms</em></li>
46    </ul>
47
48    <p><a class = "nodec" href = "http://food.com">Go to the
Grocery
49    store</a></p>
50
51  </body>
52  </html>
```

# Output

Example



Back

# External Style sheets

- An external style sheet is ideal when style is applied to many pages. With an external style sheet, you can change the look of entire web site by changing one file. Each page must link to the style sheet using <link> tag. The <link> tag goes inside the head section.
- Style Properties are defined and placed in external files.
- File saved with extension .css.
- These files are then Cascaded with the HTML Documents and properties are suitably applied.

**<HEAD>**

**<link rel =“stylesheet” type=“text/css” href=“mystyle.css”>**

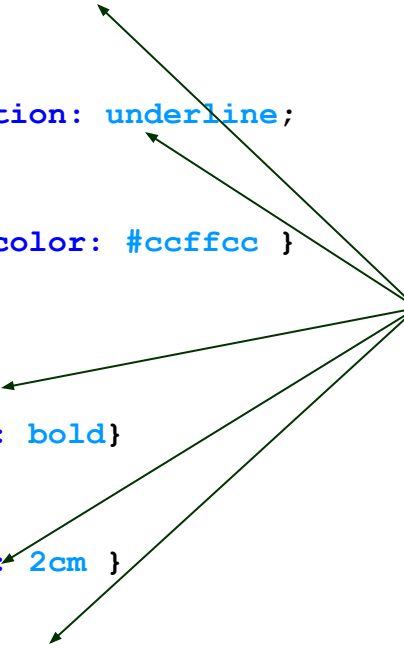
**</HEAD>**

# Ways to cascade style sheets

- Using the <LINK> Tag
  - REL =""
  - TYPE=""
  - HREF=""
- By importing the Style sheet
  - @import url("<filename>.css ")
  - works only in IE

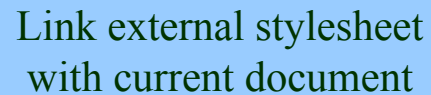
```
1  /* Fig. 4.4: styles.css */
2  /* An external stylesheet */
3
4  a      { text-decoration: none }
5
6  a:hover { text-decoration: underline;
7            color: red;
8            background-color: #ccffcc }
9
10 li em   { color: red;
11           font-weight: bold}
12
13 ul      { margin-left: 2cm }
14
15 ul ul   { text-decoration: underline;
16           margin-left: .5cm }
```

Define attributes used for  
linking documents

A diagram consisting of a central light blue box with a black border containing the text "Define attributes used for linking documents". Five black arrows originate from the left side of this box and point to specific lines of CSS code in the adjacent block: one to line 4 (the 'a' selector), one to line 6 (the 'a:hover' selector), one to line 10 (the 'li em' selector), one to line 13 (the 'ul' selector), and one to line 15 (the 'ul ul' selector).

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5
6  <!-- Fig. 4.5: imported.html      -->
7  <!-- Linking external style sheets
8
9  <head>
10     <title>XML How to Program - Importing Style Sheets</title>
11     <link rel = "stylesheet" type = "text/css" href =
"styles.css">
12 </head>
13
```

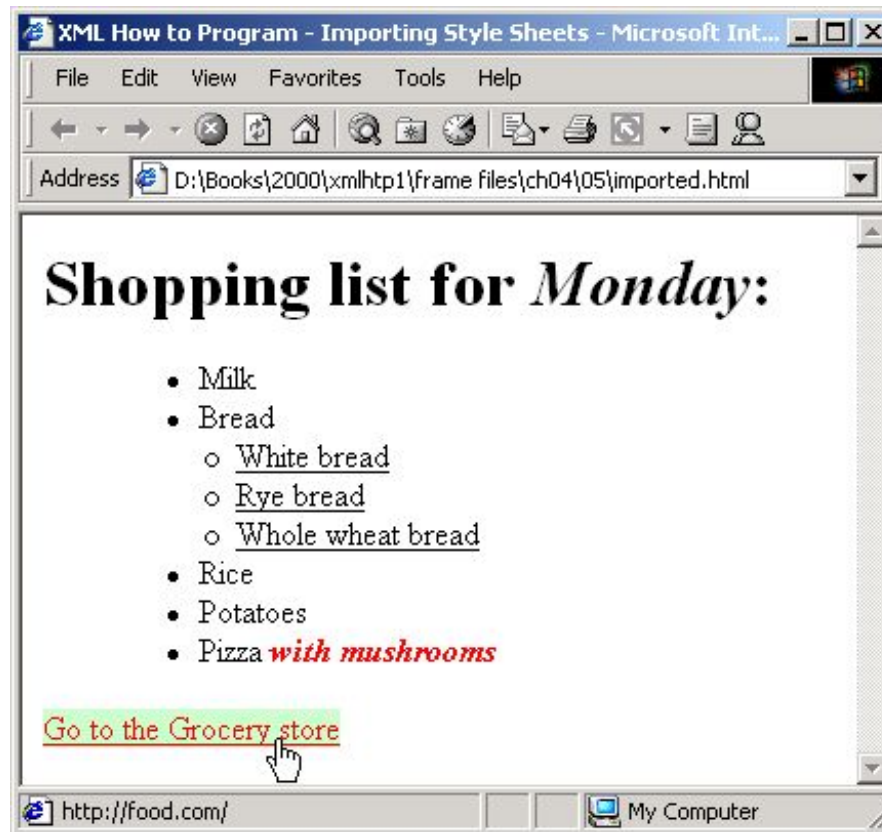
Link external stylesheet  
with current document



```
14 <body>
15
16 <h1>Shopping list for <em>Monday</em>:</h1>
17 <ul>
18 <li>Milk</li>
19 <li>Bread
20 <ul>
21 <li>White bread</li>
22 <li>Rye bread</li>
23 <li>Whole wheat bread</li>
24 </ul>
25 </li>
26 <li>Rice</li>
27 <li>Potatoes</li>
28 <li>Pizza <em>with mushrooms</em></li>
29 </ul>
30
31 <p>
32 <a href = "http://food.com">Go to the Grocery
store</a>
33 </p>
34
35 </body>
36 </html>
```

# Output

Example





# Cascading Order

- **What style will be used when there is more than one style specified for an HTML element?**

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number four has the highest priority:

- Browser default
- External style sheet
- Internal style sheet (inside the <head> tag)
- Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style declared inside the <head> tag, in an external style sheet, or in a browser (a default value).

# Related Tags

- Tags used to apply styles to parts of the HTML document
- `<SPAN>`
- `<DIV>`
- `<DIV>` Holds paragraphs and other block-level elements (starts with a new block)
  - `ALIGN` = LEFT (Default) | CENTER | RIGHT | JUSTIFY
  - `NOWRAP` = FALSE (default) | TRUE
  - `CLASS` = “userdefinedstring”
  - `ID` = “uniqueidentification”
- The primary difference between the SPAN and DIV elements is that SPAN doesn't do any formatting of its own. The DIV element includes a paragraph break. The SPAN element simply tells the browser to apply the style rules to whatever is within the SPAN.

# EXAMPLE

DIV element example	Output
<p>This is some text &lt;DIV style="color:#FF0000;"&gt; &lt;H4&gt;This is a header in a div section&lt;/H4&gt; &lt;P&gt;This is a paragraph in a div section&lt;/P&gt; &lt;/DIV&gt;</p>	<p>This is some text</p> <p><b>This is a header in a div section</b> <b>This is a paragraph in a div section</b></p>

# EXAMPLE

SPAN element example	Output
<pre>&lt;P&gt;This is a paragraph &lt;SPAN style="color:#0000FF;"&gt; This is a paragraph&lt;/SPAN&gt; This is a paragraph&lt;/P&gt; &lt;P&gt; &lt;SPAN style="color:#00DD45;"&gt; This is another paragraph &lt;/SPAN&gt; &lt;/P&gt;</pre>	<p>This is a paragraph <b>This is a paragraph</b> This is a paragraph <b>This is another paragraph</b></p>



# Style properties

- Color Properties
- Background Properties
- Font Properties
- Text Properties
- Margin Properties
- Border Properties
- Classification Properties

# Background Properties

Property	Description	Values
background	A shorthand property for setting all background properties in one declaration	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i>
background-attachment	Sets whether a background image is fixed or scrolls with the rest of the page	scroll fixed
background-color	Sets the background color of an element	<i>color-rgb</i> , <i>color-hex</i> , <i>color-name</i> transparent
background-image	Sets an image as the background	<i>url</i> none
background-position	Sets the starting position of a background image	top left, top center, top right, center left, center center, center right, bottom left, bottom center, bottom right, <i>x-% y-%</i> <i>x-pos y-pos</i>
background-repeat	Sets if/how a background image will be repeated	repeat, repeat-x, repeat-y, no-repeat

Images

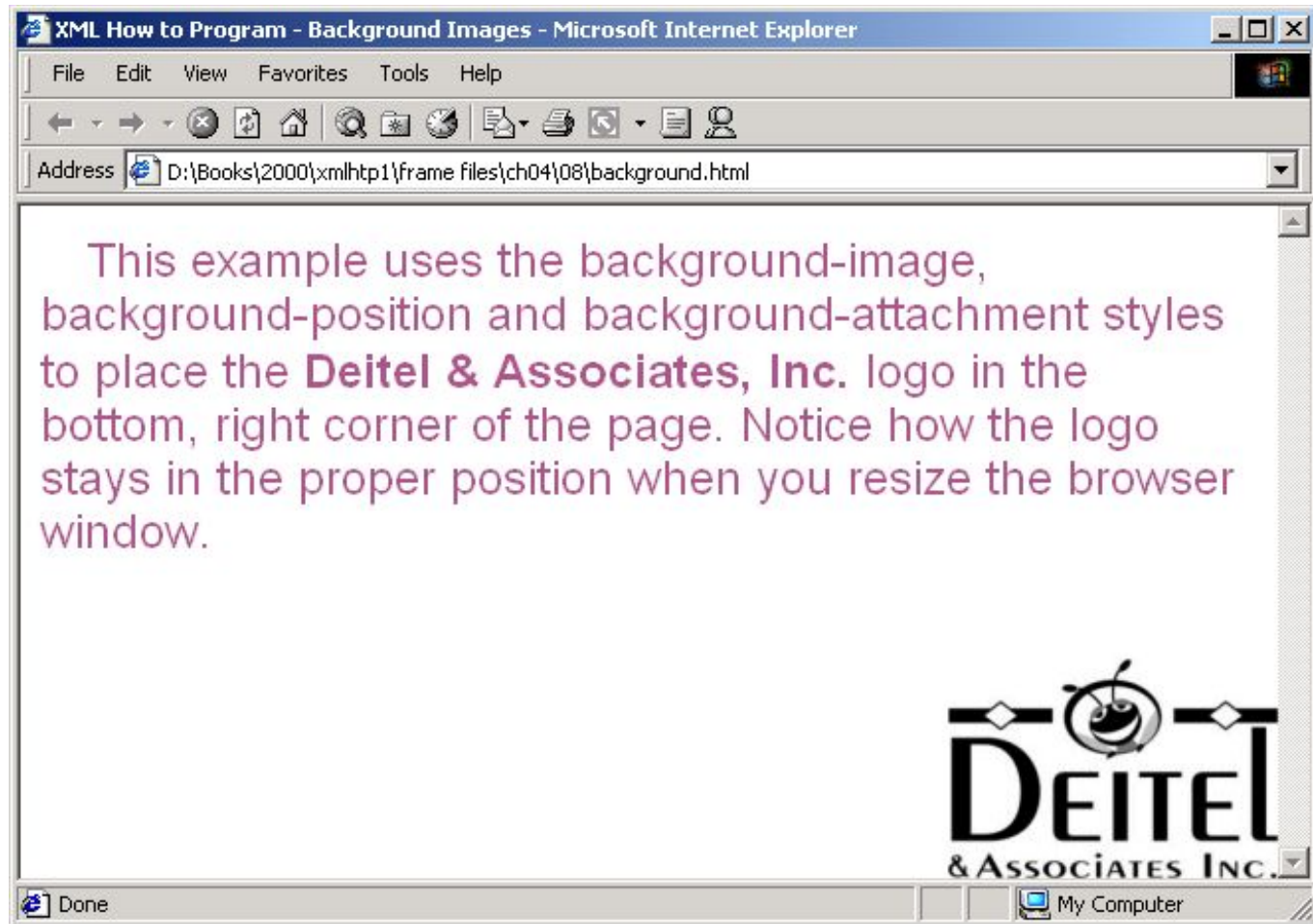
```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig. 4.8: background.html          -->
6  <!-- Adding background images and indentation -->
7
8  <head>
9      <title>XML How to Program - Background
10 </title>
11      <style type = "text/css">
12
13          body { background-image: url(logo.gif);
14                  background-position: bottom right;
15                  background-repeat: no-repeat;
16                  background-attachment: fixed; }
17
18          p      { font-size: 18pt;
19                  color: #aa5588;
20                  text-indent: 1em;
21                  font-family: arial, sans-serif; }
22
23          .dark { font-weight: bold; }
24
25      </style>
26 </head>
27
```

Place image at bottom-right of screen

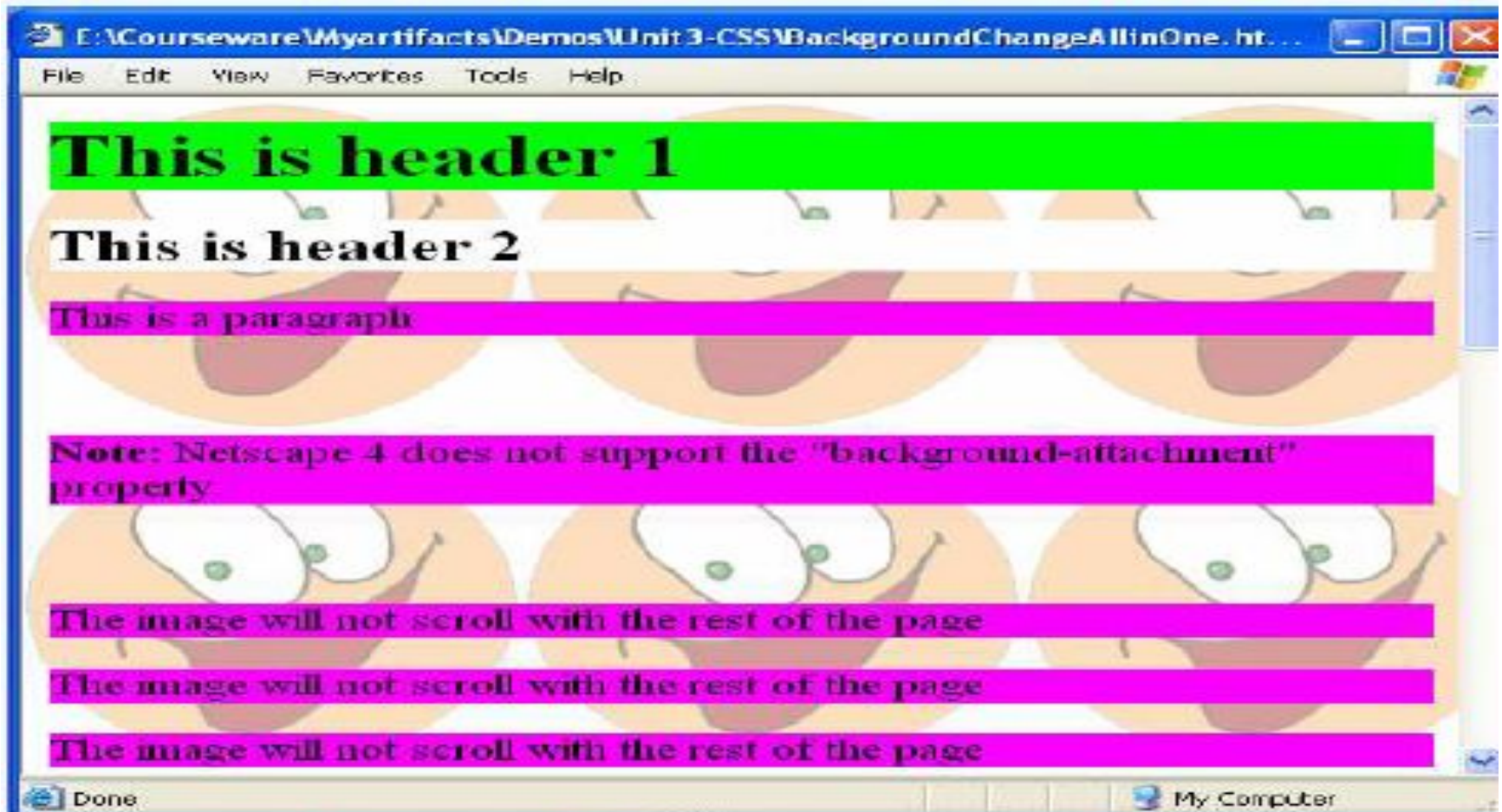
```
28  <body>
29
30  <p>
31      This example uses the background-image,
32      background-position and background-attachment
33      styles to place the <span class = "dark">Deitel
34      & Associates, Inc.</span> logo in the bottom,
35      right corner of the page. Notice how the logo
36      stays in the proper position when you resize the
37      browser window.
38  </p>
39
40  </body>
41  </html>
```



# Output



# Background Properties -Example



# EXAMPLE-CODE

```
<HTML><HEAD>
  <STYLE>
    body { background: white
    url ("smiley.bmp")
    repeat fixed }
    h1 {background-color: #00ff00}
    h2 {background-color: white}
    p {background-color: rgb(250,0,255)}
  }
</STYLE></HEAD>
<BODY>
  <H1>This is header 1</H1>
  <H2>This is header 2</H2>
  <P>This is a paragraph</P> <BR><BR> <BR>
  <p><b>Note:</b> Netscape 4 does not support the "background-attachment" property.</p>
  <BR><BR>
  <P>The image will not scroll with the rest of the page</P><BR><BR>
  <P>The image will not scroll with the rest of the page</P><BR><BR>
  <P>The image will not scroll with the rest of the page</P><BR><BR>
  <P>The image will not scroll with the rest of the page</P><BR><BR>
</BODY>
</HTML>
```

# Font Properties

Properties	Values
<i>font-family</i>	Arial, Monospace,
<i>font-style</i>	Normal, italic, oblique
<i>font-size</i>	normal, small-caps x-small, small, medium, large
<i>font-weight</i>	normal, bold, bolder, light, x-large

# TRY IT

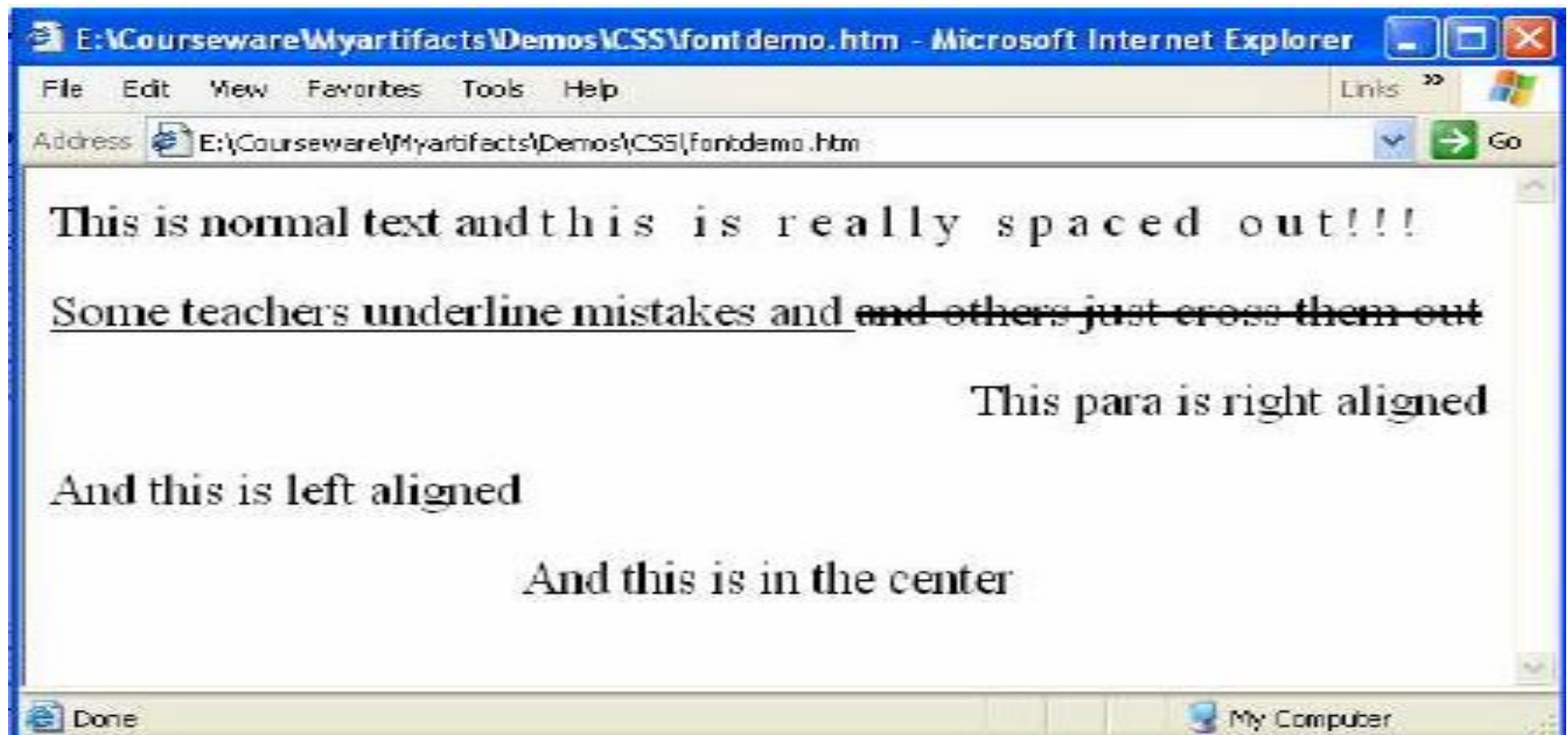
```
<HTML>
<HEAD>
  <STYLE TYPE="text/css">
    P.FontStyle {font-style:italic;}
    P.FontFamily {font-family:"Comic Sans MS", Arial;}
    P.FontWeight {font-weight:bold;}
    P.FontSize {font-size:xx-small;}
  </STYLE>
</HEAD>
<BODY>
  <P CLASS=FontFamily>What type of font is this - Comic Sans or Arial???</P>
  <P CLASS=FontStyle>This is now italicized</P>
  <P CLASS=FontWeight>The bold and the beautiful</P>
  <P CLASS=FontSize>Small is beautiful</P>
</BODY>
</HTML>
```

# Text Properties

Property	Description	Possible Values
color	Sets the color of a text	<i>color</i>
direction	Sets the text direction	ltr, rtl
letter-spacing	Increase or decrease the space between characters	normal <i>length</i>
text-align	Aligns the text in an element	left , right, center justify
text-decoration	Adds decoration to text	None, underline overline, line-through blink
text-indent	Indents the first line of text in an element	<i>length</i> , %
text-shadow		none, <i>color</i> , <i>length</i>
text-transform	Controls the letters in an element	none, capitalize uppercase, lowercase
word-spacing	Increase or decrease the space between words	normal <i>length</i>



# Example



# EXAMPLE-CODE

```
<HTML>
<HEAD>
  <STYLE TYPE="text/css">
    SPAN.LettersFar {letter-spacing:5px;}
    SPAN.Underlined {text-decoration:underline;}
    SPAN.LineThrough {text-decoration:line-through;}
    P.RightAlign {text-align:right;}
    P.LeftAlign {text-align:left;}
    P.CenterAlign {text-align:center;}
  </STYLE>
</HEAD>
<BODY>This is normal text and
  <SPAN CLASS=LettersFar>this is really spaced out!!!</SPAN>
  <P><SPAN CLASS=Underlined>Some teachers underline mistakes and
</SPAN>
  <SPAN CLASS=LineThrough>and others just cross them out</SPAN>
  <P><P CLASS=RightAlign>This para is right aligned</SPAN>
  <P CLASS=LeftAlign>And this is left aligned</SPAN>
  <P CLASS=CenterAlign>And this is in the center</SPAN>
</BODY>
</HTML>
```



# Margin Properties

Property	Description	Values
margin	A shorthand property for setting the margin properties in one declaration	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>
margin-bottom	Sets the bottom margin of an element	auto <i>length</i> %
margin-left	Sets the left margin of an element	auto <i>length</i> %
margin-right	Sets the right margin of an element	auto <i>length</i> %
margin-top	Sets the top margin of an element	auto <i>length</i> %

# Border Properties

Property	Description	Values
border-color	Sets the color of the four borders, can have from one to four colors	<i>Color</i>
border-style	Sets the style of the four borders, can have from one to four styles	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset
border-width	A shorthand property for setting the width of the four borders in one declaration, can have from one to four values	thin medium thick <i>length</i>

P {border-top-color:red} will sets the color of the top border of the paragraph to red.

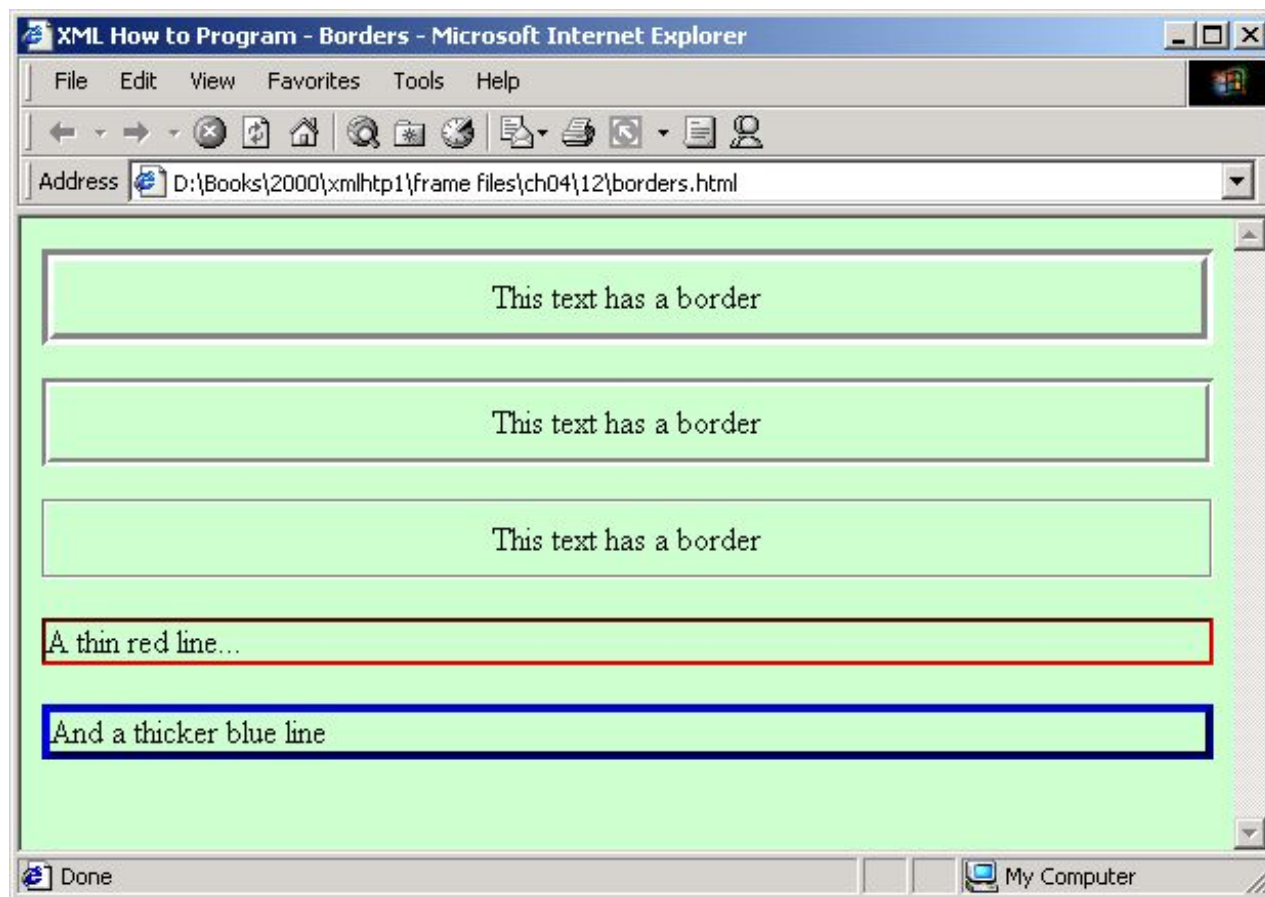
```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig. 4.12: borders.html      -->
6  <!-- Setting borders of an element -->
7
8  <head>
9      <title>XML How to Program - Borders</title>
10
11      <style type = "text/css">
12
13          body      { background-color: #ccffcc }
14
15          div        { text-align: center;
16                      margin-bottom: 1em;
17                      padding: .5em }
18
19          .thick      { border-width: thick }
20
21          .medium     { border-width: medium }
22
23          .thin       { border-width: thin }
24
25          .groove     { border-style: groove }
26
27          .inset      { border-style: inset }
28
```



Define various borders

```
29      .outset { border-style: outset }
30
31      .red    { border-color: red }
32
33      .blue   { border-color: blue }
34
35  </style>
36 </head>
37
38 <body>
39
40     <div class = "thick groove">This text has a
border</div>
41     <div class = "medium groove">This text has a
border</div>
42     <div class = "thin groove">This text has a border</div>
43
44     <p class = "thin red inset">A thin red line...</p>
45     <p class = "medium blue outset">And a thicker blue
line</p>
46
47 </body>
48 </html>
```

# Output





# Text Flow and the Box Model

- Floating
  - Allows repositioning of elements
    - Nearby text will wrap
    - `clear` property overrides wrapping
- Margin
  - Defines size of element's margins

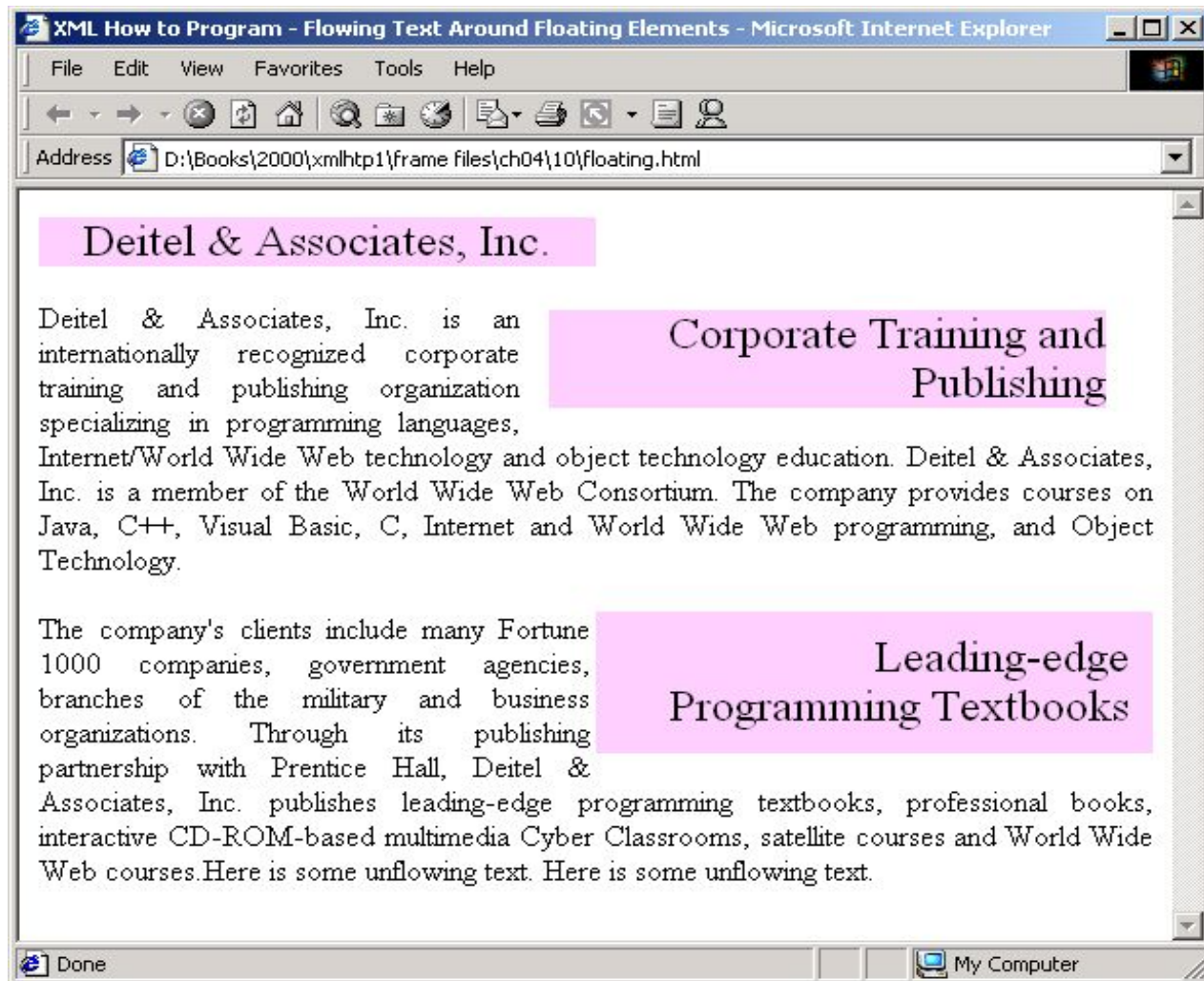
```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig. 4.10: floating.html          -->
6  <!-- Floating elements and element boxes -->
7
8  <head>
9      <title>XML How to Program - Flowing Text Around
10         Floating Elements</title>
11
12     <style type = "text/css">
13
14         div { background-color: #ffccff;
15             margin-bottom: .5em;
16             font-size: 1.5em;
17             width: 50% }
18
19         p   { text-align: justify; }
20
21     </style>
22
23 </head>
24
```

Float text in box on right-side  
with .5-pixel margin

```
25 <body>
26
27 <div style = "text-align: center">Deitel & Associates,
Inc.</div>
28
29 <div style = "float: right; margin: .5em; text-align:
right">
30 Corporate Training and Publishing</div>
31
32 <p>Deitel & Associates, Inc. is an internationally
recognized corporate training and publishing organization
specializing in programming languages, Internet/World Wide Web
technology and object technology education. Deitel & Associates,
36 Inc. is a member of the World Wide Web Consortium. The
company provides courses on Java, C++, Visual Basic, C, Internet
and 38 World Wide Web programming, and Object Technology.</p>
39
40 <div style = "float: right; padding: .5em; text-align:
right">
41 Leading-edge Programming Textbooks</div>
42
43 <p>The company's clients include many Fortune 1000
companies, government agencies, branches of the military and
business organizations. Through its publishing partnership with
Prentice Hall, Deitel & Associates, Inc. publishes leading-edge
47 programming textbooks, professional books, interactive
48 CD-ROM-based multimedia Cyber Classrooms, satellite
courses and World Wide Web courses.<span style = "clear:
right">Here is some unflowing text. Here is some unflowing
text.</span></p>
51
52 </body>
53 </html>
```



# Output





Yours to Do Stuff

# Classification Properties

Property	Description	Values
clear	Sets the sides of an element where other floating elements are not allowed	left, right, both, none
cursor	Specifies the type of cursor to be displayed	url, auto, crosshair default, pointer, move
display	Sets how/if an element is displayed	none, inline, block list-item, run-in compact, marker
float	Sets where an image or a text will appear in another element	left, right, none
list-style-type	Sets the type of the list-item marker	none, disc, circle, square decimal, lower-roman upper-roman, lower-alpha, upper-alpha
list-style-image	Sets an image as the list-item marker	none, url
visibility	Sets if an element should be visible or invisible	visible, hidden, collapse

# Positioning with Style Sheets

Property	Description	Values
bottom	Sets how far the bottom edge of an element is above/below the bottom edge of the parent element Similarly we have left , right and top.	auto, %, <i>length</i>
Position	Places an element in a static, relative, absolute or fixed position	static, relative, absolute fixed
overflow	Sets what happens if the content of an element overflow its area	visible, hidden, scroll auto
vertical-align	Sets the vertical alignment of an element	baseline, sub, super, top, text-top, middle, bottom text-bottom, <i>length</i> , %
z-index	Sets the stack order of an element. Default z-index is 0. Element with smaller Z-index will be placed "behind" another element having larger z-index,	Auto, <i>number</i>

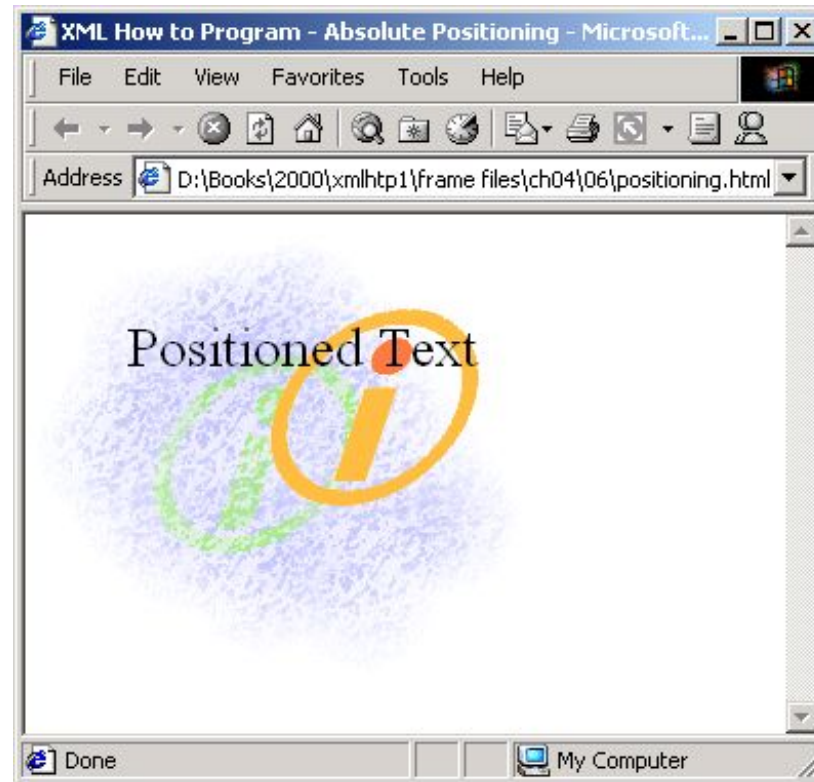
```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig 4.6: positioning.html      -->
6  <!-- Absolute positioning of elements -->
7
8  <head>
9      <title>XML How to Program - Absolute Positioning</title>
10 </head>
11
12 <body>
13
14     <p><img src = "i.gif" style = "position: absolute; top:
0px; 15     left: 0px; z-index: 1" alt = "First positioned
image"></p> 16 <p style = "position: absolute; top: 50px; left: 50px;
17     z-index: 3; font-size: 20pt;">Positioned Text</p>
18 <p><img src = "circle.gif" style = "position: absolute; top:
25px; 19     left: 100px; z-index: 2" alt = "Second positioned
image"></p> 20
21 </body>
22 </html>

```

Use inline style to position images: place second image over the first (because second image has higher z index)

# Output



```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
2      "http://www.w3.org/TR/html4/strict.dtd">
3  <html>
4
5  <!-- Fig. 4.7: positioning2.html      -->
6  <!-- Relative positioning of elements -->
7
8  <head>
9      <title>XML How to Program - Relative Positioning</title>
10
11     <style type = "text/css">
12
13         p          { font-size: 1.3em;
14                     font-family: verdana, arial, sans-serif
15
16         span       { color: red;
17                     font-size: .6em;
18                     height: 1em }
19
20         .super     { position: relative;
21                     top: -1ex }
22
23         .sub       { position: relative;
24                     bottom: -1ex }
25

```

Define attributes that position elements relative to browser-defined location

```
26     .shiftleft { position: relative;
27                 left: -1ex }
28
29     .shiftright { position: relative;
30                  right: -1ex }
31
32     </style>
33 </head>
34
35 <body>
36
37     <p>The text at the end of this sentence
38     <span class = "super">is in superscript</span>.</p>
39
40     <p>The text at the end of this sentence
41     <span class = "sub">is in subscript</span>.</p>
42
43     <p>The text at the end of this sentence
44     <span class = "shiftleft">is shifted left</span>.</p>
45
46     <p>The text at the end of this sentence
47     <span class = "shiftright">is shifted
right</span>.</p>
48
49 </body>
50 </html>
```



# Output

