



# Chapter 4

# Introduction to Cascading Style Sheets (CSS): Part 1

Internet & World Wide Web  
How to Program, 5/e



## OBJECTIVES

In this chapter you'll:

- Control a website's appearance with style sheets.
- Use a style sheet to give all the pages of a website the same look and feel.
- Use the `class` attribute to apply styles.
- Specify the precise font, size, color and other properties of displayed text.
- Specify element backgrounds and colors.
- Understand the box model and how to control margins, borders and padding.
- Use style sheets to separate presentation from content.



- 4.1** Introduction
- 4.2** Inline Styles
- 4.3** Embedded Style Sheets
- 4.4** Conflicting Styles
- 4.5** Linking External Style Sheets
- 4.6** Positioning Elements: Absolute Positioning, `z-index`
- 4.7** Positioning Elements: Relative Positioning, `span`
- 4.8** Backgrounds
- 4.9** Element Dimensions
- 4.10** Box Model and Text Flow
- 4.11** Media Types and Media Queries
- 4.12** Drop-Down Menus
- 4.13** (Optional) User Style Sheets
- 4.14** Web Resources



# 4.1 Introduction

- ▶ Cascading Style Sheets 3 (CSS3)
  - Used to specify the presentation of elements separately from the structure of the document.
- ▶ CSS validator
  - `jigsaw.w3.org/css-validator/`
  - This tool can help you make sure that your code is correct and will work on CSS3-compliant browsers.





## 4.2 Inline Styles

- ▶ Inline style
  - declare an individual element's format using the HTML5 attribute `style`.
- ▶ Figure 4.1 applies inline styles to `p` elements to alter their font size and color.
- ▶ Each CSS property is followed by a colon and the value of the attribute
  - Multiple property declarations are separated by a semicolon



## **Software Engineering Observation 4.1**

---

Inline styles do not truly separate presentation from content. To apply similar styles to multiple elements, use embedded style sheets or external style sheets, introduced later in this chapter.



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.1: inline.html -->
4  <!-- Using inline styles -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Inline Styles</title>
9      </head>
10     <body>
11         <p>This text does not have any style applied to it.</p>
12
13         <!-- The style attribute allows you to declare -->
14         <!-- inline styles. Separate multiple -->
15         <!-- style properties with a semicolon. -->
16         <p style = "font-size: 20pt;">This text has the
17             <em>font-size</em> style applied to it, making it 20pt.
18         </p>
19
20         <p style = "font-size: 20pt; color: deepskyblue;">
21             This text has the <em>font-size</em> and
22             <em>color</em> styles applied to it, making it
23             20pt and deep sky blue.</p>
24     </body>
25 </html>
```

Fig. 4.1 | Using inline styles. (Part I of 2.)



**Fig. 4.1** | Using inline styles. (Part 2 of 2.)



## 4.2 Inline Styles (Cont.)

- ▶ color property sets text color
  - Color names and hexadecimal codes may be used as the color property value.
  - Figure 4.2 contains the HTML standard color set.
  - A list of extended hexadecimal color codes and color names is provided in Appendix B.
  - You can also find a complete list of HTML standard and extended colors at
    - [www.w3.org/TR/css3-color/](http://www.w3.org/TR/css3-color/)



Color name	Value	Color name	Value
aqua	#00FFFF	navy	#000080
black	#000000	olive	#808000
blue	#0000FF	purple	#800080
fuchsia	#FF00FF	red	#FF0000
gray	#808080	silver	#C0C0C0
green	#008000	teal	#008080
lime	#00FF00	yellow	#FFFF00
maroon	#800000	white	#FFFFFF

**Fig. 4.2** | HTML standard colors and hexadecimal RGB values.



## 4.3 Embedded Style Sheets

- ▶ A second technique for using style sheets is **embedded style sheets**, which enable you to embed a CSS3 document in an HTML5 document's head section.
- ▶ Figure 4.3 creates an embedded style sheet containing four styles.



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.3: embedded.html -->
4  <!-- Embedded style sheet. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Embedded Style Sheet</title>
9
10         <!-- this begins the style sheet section -->
11         <style type = "text/css">
12             em        { font-weight: bold;
13                         color: black; }
14             h1        { font-family: tahoma, helvetica, sans-serif; }
15             p          { font-size: 12pt;
16                         font-family: arial, sans-serif; }
17             .special { color: purple; }
18         </style>
19     </head>
```

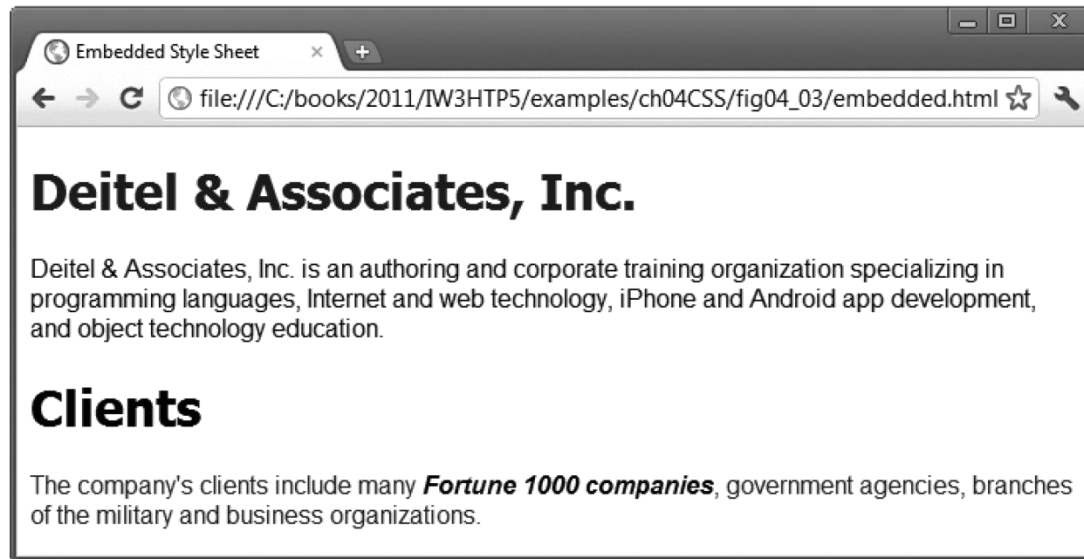
**Fig. 4.3** | Embedded style sheet. (Part 1 of 3.)





```
20 <body>
21   <!-- this attribute applies the .special style class -->
22   <h1 class = "special">Deitel & Associates, Inc.</h1>
23
24   <p>Deitel & Associates, Inc. is an authoring and
25     corporate training organization specializing in
26     programming languages, Internet and web technology,
27     iPhone and Android app development, and object
28     technology education.</p>
29
30   <h1>Clients</h1>
31   <p class = "special"> The company's clients include many
32     <em>Fortune 1000 companies</em>, government agencies,
33     branches of the military and business organizations.</p>
34 </body>
35 </html>
```

**Fig. 4.3** | Embedded style sheet. (Part 2 of 3.)



**Fig. 4.3** | Embedded style sheet. (Part 3 of 3.)



## 4.3 Embedded Style Sheets (cont.)

### *The style Element and MIME Types*

- ▶ Styles that are placed in a style element use selectors to apply style elements throughout the entire document
- ▶ style element type attribute specifies the MIME type (the specific encoding format) of the style sheet. Style sheets use text/css.
- ▶ Figure 4.4 lists common MIME types used in this book. For a complete list, visit:
  - [www.w3schools.com/media/media\\_mimeref.asp](http://www.w3schools.com/media/media_mimeref.asp)



MIME type	Description
text/css	CSS documents
image/png	PNG images
text/javascript	JavaScript markup
text/plain	Plain text
image/jpeg	JPEG image
text/html	HTML markup

**Fig. 4.4** | A few common MIME types.



## 4.3 Embedded Style Sheets (cont.)

- ▶ The style sheet's body declares the **CSS rules** for the style sheet.
- ▶ To achieve the separation between the CSS3 code and the HTML5 that it styles, we'll use a **CSS selector** to specify the elements that will be styled according to a rule.
- ▶ An **em element** indicates that its contents should be *emphasized*.
- ▶ Each rule body in a style sheet is enclosed in curly braces ({ and }).



## 4.3 Embedded Style Sheets (Cont.)

- ▶ font-weight property specifies the “boldness” of text. Possible values are:
  - bold
  - normal (the default)
  - bolder (bolder than bold text)
  - lighter (lighter than normal text)
  - Boldness also can be specified with multiples of 100, from 100 to 900 (e.g., 100, 200, ..., 900). Text specified as normal is equivalent to 400, and bold text is equivalent to 700



## 4.3 Embedded Style Sheets (Cont.)

### *Style Classes*

- ▶ Style-class declarations are preceded by a period (.).
- ▶ They define styles that can be applied to *any* element.
- ▶ In this example, class special sets color to purple.
- ▶ You can also declare id selectors.
- ▶ If an element in your page has an id, you can declare a selector of the form *#elementId* to specify that element's style.



## 5.3 Embedded Style Sheets (Cont.)

### *font-family Property*

- ▶ font-family property specifies the name of the font to use.
  - Generic font families allow authors to specify a type of font instead of a specific font, in case a browser does not support a specific font.



Generic font families	Examples
serif	times new roman, georgia
sans-serif	arial, verdana, futura
cursive	script
fantasy	critter
monospace	courier, fixedsys

**Fig. 4.5** | Generic font families.



## 4.3 Embedded Style Sheets (Cont.)

### *font-size Property*

- ▶ font-size property specifies the size used to render the font.
- ▶ You can specify a point size or a relative value such as xx-small, x-small, small, smaller, medium, large, larger, x-large and xx-large.
- ▶ Relative font-size values are preferred over points, because an author does not know the specific measurements of each client's display.
- ▶ Relative values permit more flexible viewing of web pages.
  - For example, users can change font sizes the browser displays for readability.



## 4.3 Embedded Style Sheets (Cont.)

### *Applying a Style Class*

- ▶ In many cases, the styles applied to an element (the **parent** or **ancestor element**) also apply to the element's *nested elements* (**child** or **descendant elements**).
- ▶ Multiple values of one property can be set or inherited on the same element, so the browser must reduce them to one value for that property per element before they're rendered.
- ▶ We discuss the rules for resolving these conflicts in the next section.



## 5.4 Conflicting Styles

- ▶ Styles may be defined by a **user**, an **author** or a **user agent**.
  - Styles **cascade** (and hence the term “Cascading Style Sheets”), or flow together, such that the ultimate appearance of elements on a page results from combining styles defined in several ways.
  - Styles defined by the user take precedence over styles defined by the user agent.
  - Styles defined by authors take precedence over styles defined by the user.
- ▶ Most styles defined for parent elements are also **inherited** by child (nested) elements.



## 5.4 Conflicting Styles (Cont.)

- ▶ text-decoration property applies decorations to text in an element
  - underline
  - overline
  - line-through
  - blink



## 5.4 Conflicting Styles (Cont.)

- ▶ Figure 4.3 contains an example of inheritance in which a child `em` element inherits the `font-size` property from its parent `p` element.
- ▶ However, in Fig. 4.3, the child `em` element has a `color` property that conflicts with (i.e., has a different value than) the `color` property of its parent `p` element.
- ▶ Properties defined for child and descendant elements have a higher specificity than properties defined for parent and ancestor elements.
- ▶ Conflicts are resolved in favor of properties with a higher specificity, so the child's styles take precedence.
- ▶ Figure 4.6 illustrates examples of inheritance and specificity.



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.6: advanced.html -->
4  <!-- Inheritance in style sheets. -->
5  <html>
6    <head>
7      <meta charset = "utf-8">
8      <title>More Styles</title>
9      <style type = "text/css">
10         body      { font-family: arial, helvetica, sans-serif; }
11         a.nodect   { text-decoration: none; }
12         a:hover    { text-decoration: underline; }
13         li em      { font-weight: bold; }
14         h1, em     { text-decoration: underline; }
15         ul         { margin-left: 20px; }
16         ul ul      { font-size: .8em; }
17      </style>
18    </head>
```

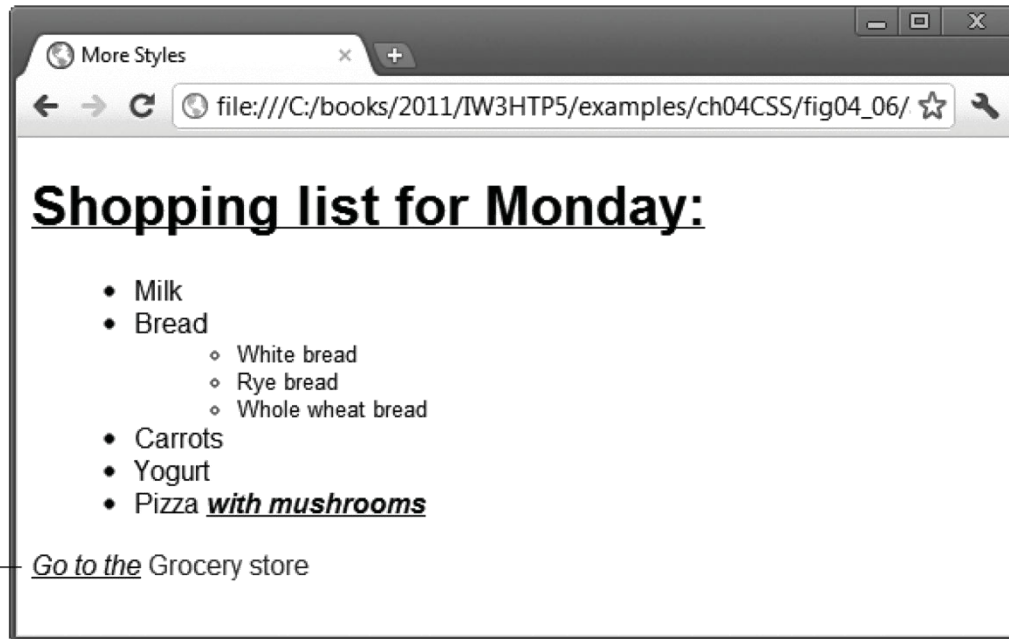
**Fig. 4.6** | Inheritance in style sheets. (Part I of 4.)



```
19 <body>
20   <h1>Shopping list for Monday:</h1>
21
22   <ul>
23     <li>Milk</li>
24     <li>Bread
25       <ul>
26         <li>white bread</li>
27         <li>Rye bread</li>
28         <li>Whole wheat bread</li>
29       </ul>
30     </li>
31     <li>Carrots</li>
32     <li>Yogurt</li>
33     <li>Pizza <em>with mushrooms</em></li>
34   </ul>
35
36   <p><em>Go to the</em>
37     <a class = "nodec" href = "http://www.deitel.com">
38       Grocery store</a>
39   </p>
40 </body>
41 </html>
```

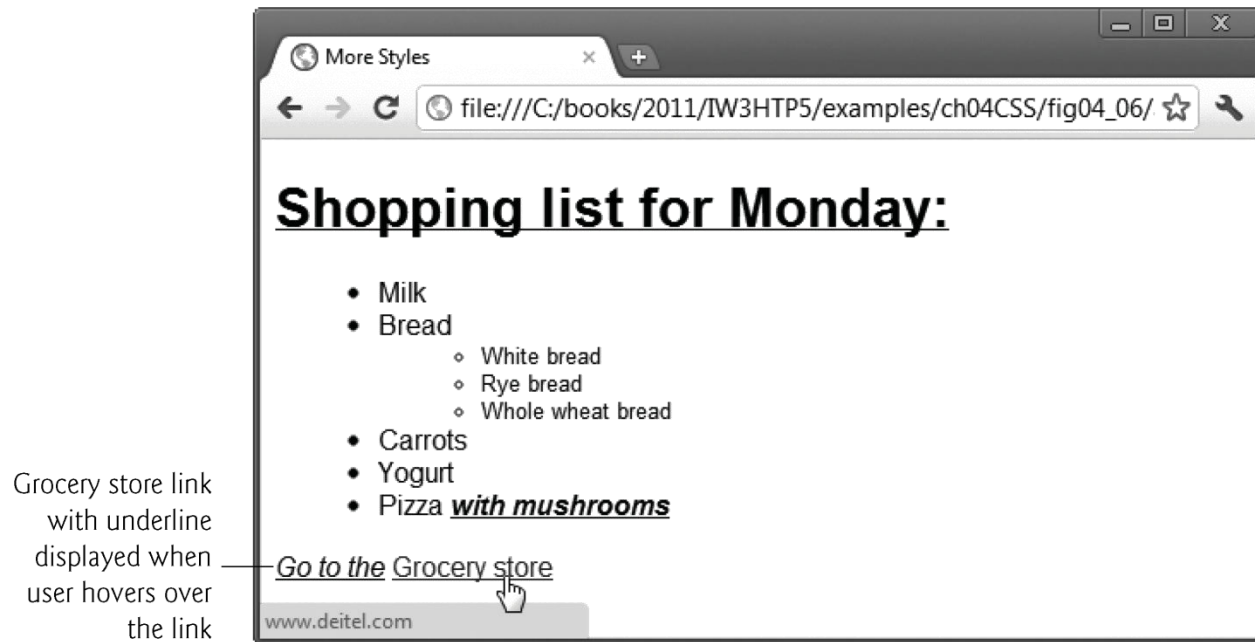
**Fig. 4.6** | Inheritance in style sheets. (Part 2 of 4.)





Grocery store link  
with no decoration

**Fig. 4.6** | Inheritance in style sheets. (Part 3 of 4.)



**Fig. 4.6** | Inheritance in style sheets. (Part 4 of 4.)



### **Portability Tip 4.1**

---

To ensure that your style sheets work in various web browsers, test them on many client web browsers, and use the W3C CSS Validator.



## 4.4 Conflicting Styles (Cont.)

- ▶ Pseudoclasses give you access to content that's not declared in the document.
- ▶ hover pseudoclass is activated when the user moves the mouse cursor over an element.



## **Common Programming Error 4.1**

---

Including a space before or after the colon separating a pseudo-class from the name of the element to which it's applied prevents the pseudo-class from being applied properly.



## 4.4 Conflicting Styles (Cont.)

- ▶ Relative length measurements:
  - px (pixels – size varies depending on screen resolution)
  - em (usually the height of a font's uppercase M)
  - ex (usually the height of a font's lowercase x)
  - Percentages (of the font's default size)
- ▶ Absolute-length measurements (units that do not vary in size):
  - in (inches)
  - cm (centimeters)
  - mm (millimeters)
  - pt (points;  $1 \text{ pt} = 1/72 \text{ in}$ )
  - pc (picas;  $1 \text{ pc} = 12 \text{ pt}$ )



### **Good Programming Practice 4.1**

---

Whenever possible, use relative-length measurements. If you use absolute-length measurements, your document may not scale well on some client browsers (e.g., smartphones).



## 4.5 Linking External Style Sheets

- ▶ External style sheets are separate documents that contain only CSS rules.
- ▶ Help create a uniform look for a website
  - Separate pages can all use the same styles.
  - Modifying a single style-sheet file makes changes to styles across an entire website (or to a portion of one).
- ▶ When changes to the styles are required, you need to modify only a single CSS file to make style changes across *all* the pages that use those styles. This concept is sometimes known as **skinning**.





## 4.5 Linking External Style Sheets (Cont.)

- ▶ Figure 4.7 presents an external style sheet.
- ▶ **CSS comments** may be placed in any type of CSS code (i.e., inline styles, embedded style sheets and external style sheets) and always start with `/*` and end with `*/`.



```
1  /* Fig. 4.7: styles.css */
2  /* External style sheet */
3  body      { font-family: arial, helvetica, sans-serif; }
4  a.nodect  { text-decoration: none; }
5  a:hover   { text-decoration: underline; }
6  li em     { font-weight: bold; }
7  h1, em    { text-decoration: underline; }
8  ul        { margin-left: 20px; }
9  ul ul     { font-size: .8em; }
```

**Fig. 4.7** | External style sheet.



## 4.5 Linking External Style Sheets (Cont.)

- ▶ Figure 4.8 contains an HTML5 document that references the external style sheet.
- ▶ Link element
  - Uses rel attribute to specify a relationship between two documents
  - rel attribute declares the linked document to be a stylesheet for the document
- ▶ type attribute specifies the MIME type of the related document
- ▶ href attribute provides the URL for the document containing the style sheet



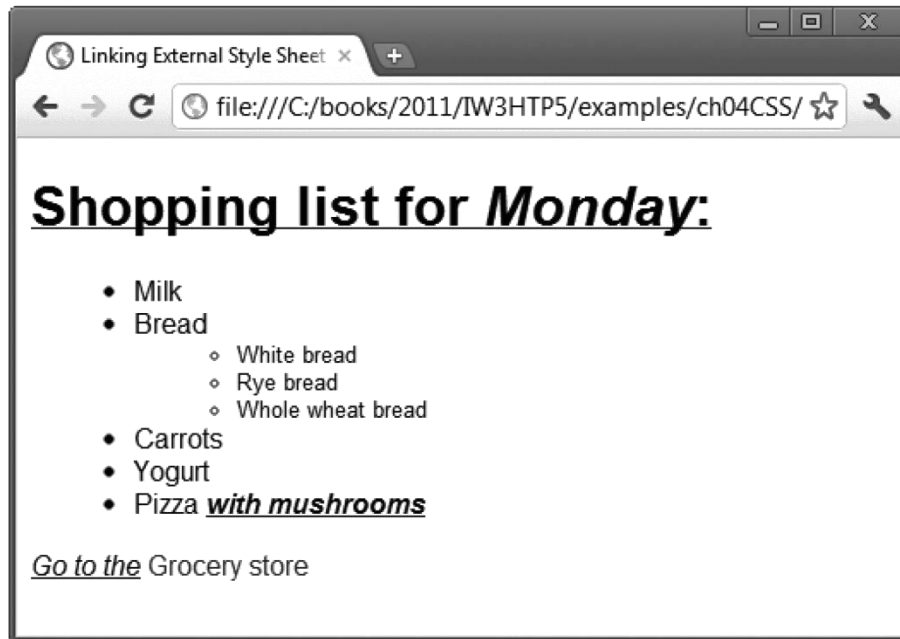
```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.8: external.html -->
4  <!-- Linking an external style sheet. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Linking External Style Sheets</title>
9          <link rel = "stylesheet" type = "text/css"
10             href = "styles.css">
11      </head>
12      <body>
13          <h1>Shopping list for <em>Monday</em>:</h1>
14
15          <ul>
16              <li>Milk</li>
17              <li>Bread
18                  <ul>
19                      <li>white bread</li>
20                      <li>Rye bread</li>
21                      <li>Whole wheat bread</li>
22                  </ul>
23              </li>
```

**Fig. 4.8** | Linking an external style sheet. (Part I of 4.)

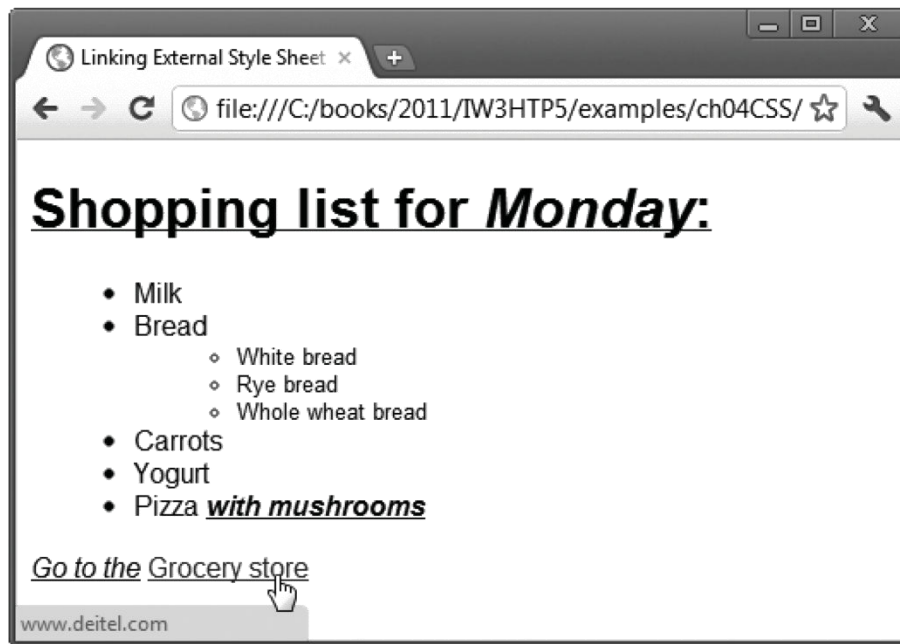


```
24         <li>Carrots</li>
25         <li>Yogurt</li>
26         <li>Pizza <em>with mushrooms</em></li>
27     </ul>
28
29     <p><em>Go to the</em>
30         <a class = "nodec" href = "http://www.deitel.com">
31         Grocery store</a>
32     </p>
33 </body>
34 </html>
```

**Fig. 4.8** | Linking an external style sheet. (Part 2 of 4.)



**Fig. 4.8** | Linking an external style sheet. (Part 3 of 4.)



**Fig. 4.8** | Linking an external style sheet. (Part 4 of 4.)

# 4.6 Positioning Elements: Absolute Positioning, z-index

- ▶ CSS position property
  - Allows absolute positioning, which provides greater control over where on a page elements reside
  - Normally, elements are positioned on the page in the order in which they appear in the HTML5 document
  - Specifying an element's position as absolute removes it from the normal flow of elements on the page and positions it according to distance from the top, left, right or bottom margin of its parent element

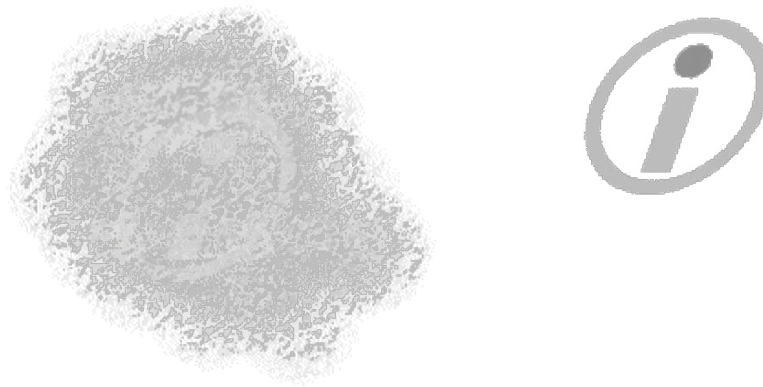




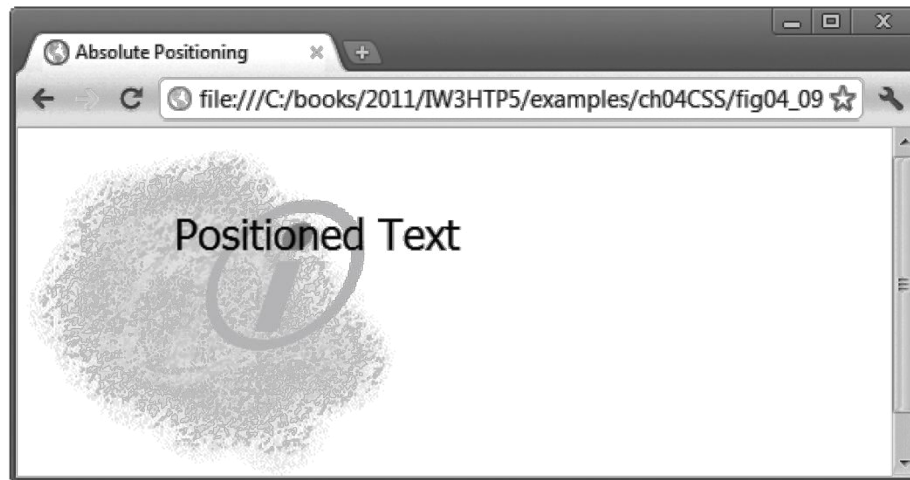
```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.9: positioning.html -->
4  <!-- Absolute positioning of elements. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Absolute Positioning</title>
9          <style type = "text/css">
10             .background_image { position: absolute;
11                                 top: 0px;
12                                 left: 0px;
13                                 z-index: 1; }
14             .foreground_image { position: absolute;
15                                 top: 25px;
16                                 left: 100px;
17                                 z-index: 2; }
18             .text               { position: absolute;
19                                 top: 25px;
20                                 left: 100px;
21                                 z-index: 3;
22                                 font-size: 20pt;
23                                 font-family: tahoma, geneva, sans-serif; }
24          </style>
25      </head>
```

**Fig. 4.9** | Absolute positioning of elements. (Part I of 3.)

```
26 <body>
27   <p><img src = "background_image.png" class = "background_image"
28     alt = "First positioned image" /></p>
29
30   <p><img src = "foreground_image.png" class = "foreground_image"
31     alt = "Second positioned image" /></p>
32
33   <p class = "text">Positioned Text</p>
34 </body>
35 </html>
```



**Fig. 4.9** | Absolute positioning of elements. (Part 2 of 3.)



**Fig. 4.9** | Absolute positioning of elements. (Part 3 of 3.)

## 4.6 Positioning Elements: Absolute Positioning, z-index (Cont.)

- ▶ The z-index property allows a developer to layer overlapping elements
- ▶ Elements that have higher z-index values are displayed in front of elements with lower z-index values

## 4.7 Positioning Elements: Relative

### Positioning, span

- ▶ Figure 4.10 demonstrates relative positioning, in which elements are positioned *relative to other elements*.



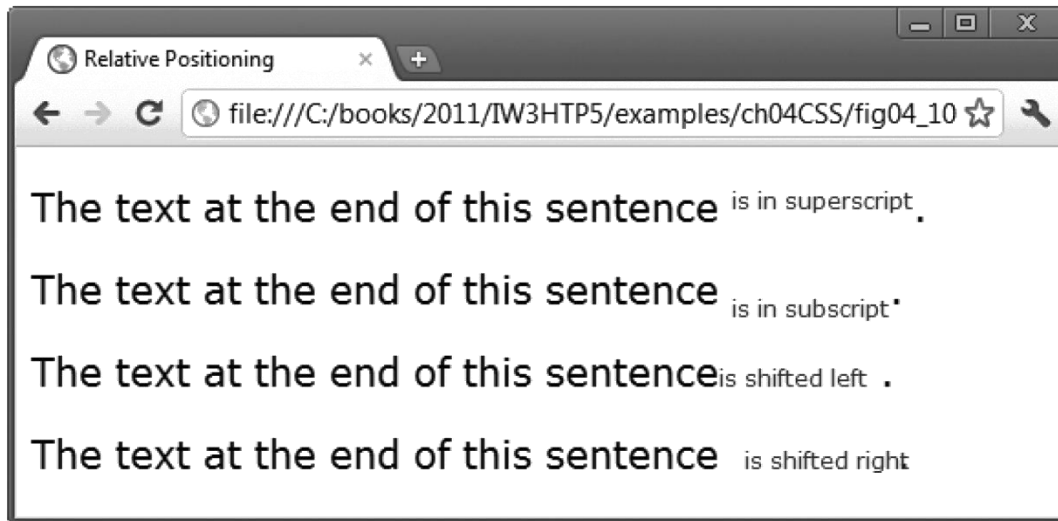
```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.10: positioning2.html -->
4  <!-- Relative positioning of elements. -->
5  <html>
6    <head>
7      <meta charset = "utf-8">
8      <title>Relative Positioning</title>
9      <style type = "text/css">
10         p          { font-size: 1.3em;
11                     font-family: verdana, arial, sans-serif; }
12         span       { color: red;
13                     font-size: .6em;
14                     height: 1em; }
15         .super     { position: relative;
16                     top: -1ex; }
17         .sub       { position: relative;
18                     bottom: -1ex; }
19         .shiftleft { position: relative;
20                     left: -1ex; }
21         .shiftright { position: relative;
22                      right: -1ex; }
23      </style>
24    </head>
```

**Fig. 4.10** | Relative positioning of elements. (Part I of 3.)



```
25 <body>
26   <p>The text at the end of this sentence
27     <span class = "super">is in superscript</span>.</p>
28
29   <p>The text at the end of this sentence
30     <span class = "sub">is in subscript</span>.</p>
31
32   <p>The text at the end of this sentence
33     <span class = "shiftleft">is shifted left</span>.</p>
34
35   <p>The text at the end of this sentence
36     <span class = "shiftright">is shifted right</span>.</p>
37 </body>
38 </html>
```

**Fig. 4.10** | Relative positioning of elements. (Part 2 of 3.)



**Fig. 4.10** | Relative positioning of elements. (Part 3 of 3.)



# 4.7 Positioning Elements: Relative Positioning, span (Cont.)



## *Inline and Block-Level Elements*

### ▶ Inline-level elements

- Do not change the flow of the document
- Examples:

- `img`
- `a`
- `em`
- `strong`
- `span`

Grouping element

Does not apply any formatting to its contents

Creates a container for CSS rules or id attributes to be applied to a section

# 4.7 Positioning Elements: Relative Positioning, span (Cont.)

- ▶ Block-level elements
  - Displayed on their own line
  - Have virtual boxes around them
  - Examples:
    - p
    - all headings (h1 through h6)
    - div
      - A grouping element like span



## 4.8 Backgrounds

- ▶ CSS can control the backgrounds of block-level elements by adding:
  - Colors
  - Images
- ▶ Figure 4.11 adds a corporate logo to the bottom-right corner of the document. This logo stays fixed in the corner even when the user scrolls up or down the screen.



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.11: background.html -->
4  <!-- Adding background images and indentation -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Background Images</title>
9          <style type = "text/css">
10             body { background-image: url(logo.png);
11                  background-position: bottom right;
12                  background-repeat: no-repeat;
13                  background-attachment: fixed;
14                  background-color: lightgrey; }
15             p     { font-size: 18pt;
16                  color: Darkblue;
17                  text-indent: 1em;
18                  font-family: arial, sans-serif; }
19             .dark { font-weight: bold; }
20          </style>
21      </head>
```

**Fig. 4.11** | Adding background images and indentation. (Part I of 3.)



```
22     <body>
23         <p>
24             This example uses the background-image,
25             background-position and background-attachment
26             styles to place the <span class = "dark">Deitel
27             & Associates, Inc.</span> logo in the
28             bottom-right corner of the page. Notice how the logo
29             stays in the proper position when you resize the
30             browser window. The background-color fills in where
31             there is no image.
32         </p>
33     </body>
34 </html>
```

**Fig. 4.11** | Adding background images and indentation. (Part 2 of 3.)



**Fig. 4.11** | Adding background images and indentation. (Part 3 of 3.)



## 4.8 Backgrounds (Cont.)

### *background-image Property*

- ▶ Specifies the URL of the image, in the format `url(fileLocation)`

### *background-position Property*

- ▶ Places the image on the page using the values `top`, `bottom`, `center`, `left` and `right` individually or in combination for vertical and horizontal positioning. You can also position by using lengths



## 4.8 Backgrounds (Cont.)

### *background-repeat Property*

- ▶ background-repeat property controls the **tiling** of the background image
  - Setting the tiling to no-repeat displays one copy of the background image on screen
  - Setting to repeat (the default) tiles the image vertically and horizontally
  - Setting to repeat-x tiles the image only horizontally
  - Setting to repeat-y tile the image only vertically





## 4.8 Backgrounds (Cont.)

### *background-attachment: fixed Property*

- ▶ Fixes the image in the position specified by background-position.
- ▶ Scrolling the browser window will not move the image from its set position.
- ▶ The default value, scroll, moves the image as the user scrolls the window



## 4.8 Backgrounds (Cont.)

### *text-indent Property*

- ▶ Indents the first line of text in the element by the specified amount

### *font-style Property*

- ▶ Allows you to set text to none, italic or oblique



# 4.9 Element Dimensions

- ▶ Figure 4.12 demonstrates how to set the dimensions of elements.



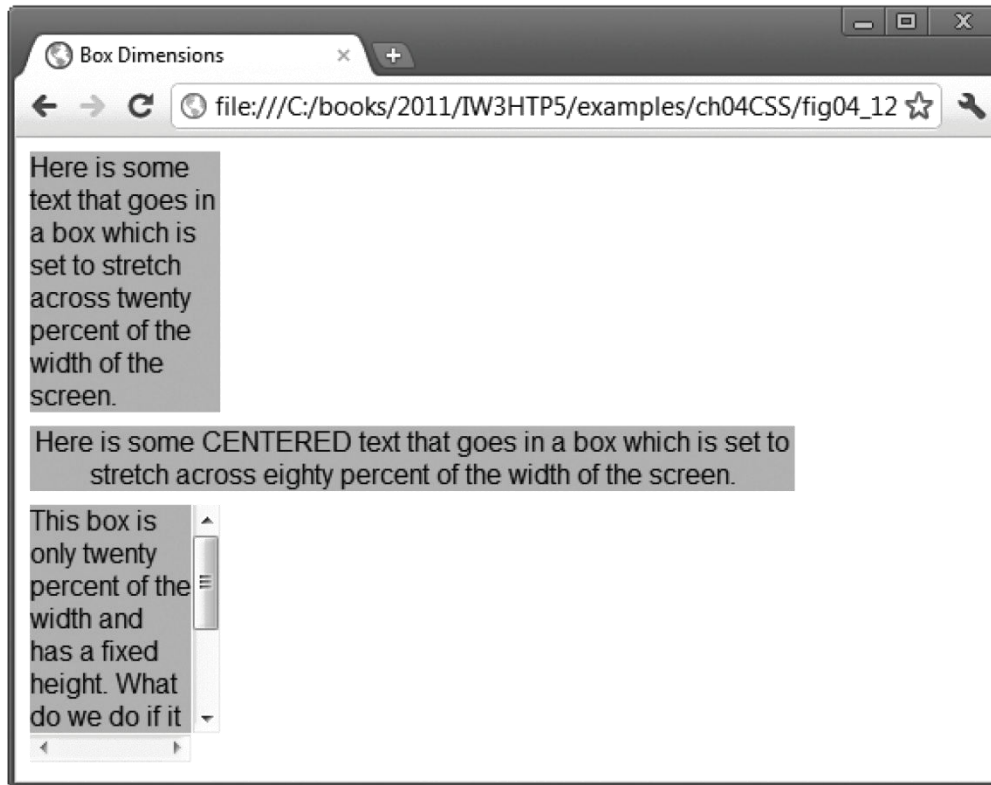
```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.12: width.html -->
4  <!-- Element dimensions and text alignment. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Box Dimensions</title>
9          <style type = "text/css">
10             p { background-color: lightskyblue;
11                 margin-bottom: .5em;
12                 font-family: arial, helvetica, sans-serif; }
13          </style>
14      </head>
15      <body>
16          <p style = "width: 20%">Here is some
17              text that goes in a box which is
18              set to stretch across twenty percent
19              of the width of the screen.</p>
20
21          <p style = "width: 80%; text-align: center">
22              Here is some CENTERED text that goes in a box
23              which is set to stretch across eighty percent of
24              the width of the screen.</section>
```

**Fig. 4.12** | Element dimensions and text alignment. (Part I of 3.)



```
25
26     <p style = "width: 20%; height: 150px; overflow: scroll">
27         This box is only twenty percent of
28         the width and has a fixed height.
29         What do we do if it overflows? Set the
30         overflow property to scroll!</p>
31     </body>
32 </html>
```

**Fig. 4.12** | Element dimensions and text alignment. (Part 2 of 3.)



**Fig. 4.12** | Element dimensions and text alignment. (Part 3 of 3.)



## 4.9 Element Dimensions

### *Specifying the width and height of an Element*

- ▶ Dimensions of elements on a page can be set with CSS by using properties height and width
  - Their values can be relative or absolute

### *text-align Property*

- ▶ Text in an element can be centered using `text-align: center;` other values for the `text-align` property are `left` and `right`



## 4.9 Element Dimensions (Cont.)

### *overflow Property and Scroll Bars*

- ▶ Problem with setting both vertical and horizontal dimensions of an element
  - Content might sometimes exceed the set boundaries, in which case the element must be made large enough for all the content to fit
  - Can set the `overflow` property to `scroll`, which adds scroll bars if the text overflows the boundaries set for it





# 4.10 Box Model and Text Flow

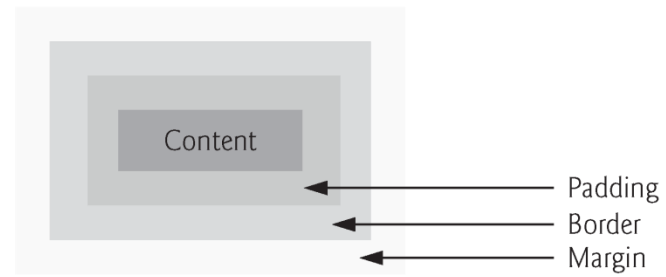
- ▶ Block-level HTML5 elements have a virtual box drawn around them based on the box model
- ▶ When the browser renders an element using the box model, the content is surrounded by padding, a margin and a border.
- ▶ **Padding**
  - The padding property determines the distance between the content inside an element and the edge of the element
  - Padding be set for each side of the box by using padding-top, padding-right, padding-left and padding-bottom
- ▶ **Margin**
  - Determines the distance between the element's edge and any outside text
  - Margins for individual sides of an element can be specified by using margin-top, margin-right, margin-left and margin-bottom

# 4.10 Box Model and Text Flow (Cont.)



## ▶ Border

- The border is controlled using the properties:
- `border-width`
  - May be set to any of the CSS lengths or to the predefined value of thin, medium or thick
- `border-color`
  - Sets the color used for the border
- `border-style`
  - Options are: none, hidden, dotted, dashed, solid, double, groove, ridge, inset and outset



**Fig. 4.13** | Box model for block-level elements.

## 4.10 Box Model and Text Flow (Cont.)



- ▶ CSS controls the border using three properties: border-width, border-color and border-style.
- ▶ We illustrate these properties in Fig. 4.14.



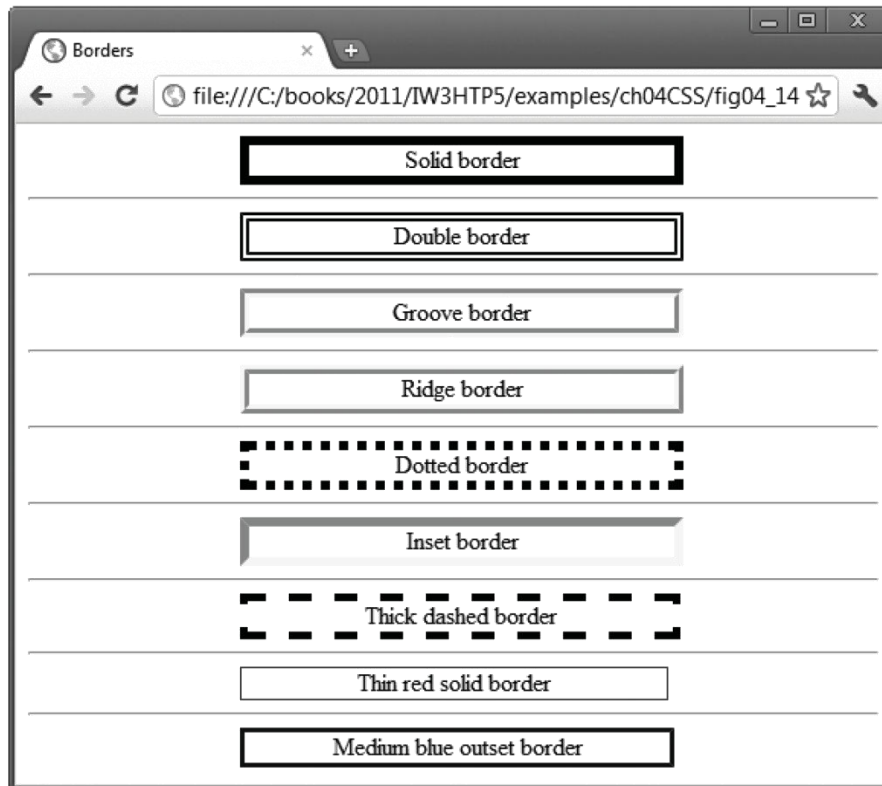
```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.14: borders.html -->
4  <!-- Borders of block-level elements. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Borders</title>
9          <style type = "text/css">
10             div      { text-align: center;
11                        width: 50%;
12                        position: relative;
13                        left: 25%;
14                        border-width: 6px; }
15             .thick   { border-width: thick; }
16             .medium  { border-width: medium; }
17             .thin    { border-width: thin; }
18             .solid   { border-style: solid; }
19             .double  { border-style: double; }
20             .groove  { border-style: groove; }
21             .ridge   { border-style: ridge; }
22             .dotted  { border-style: dotted; }
23             .inset   { border-style: inset; }
24             .outset  { border-style: outset; }
```

**Fig. 4.14** | Borders of block-level elements. (Part I of 3.)



```
25     .dashed { border-style: dashed; }
26     .red    { border-color: red; }
27     .blue   { border-color: blue; }
28 </style>
29 </head>
30 <body>
31     <div class = "solid">Solid border</div><hr>
32     <div class = "double">Double border</div><hr>
33     <div class = "groove">Groove border</div><hr>
34     <div class = "ridge">Ridge border</div><hr>
35     <div class = "dotted">Dotted border</div><hr>
36     <div class = "inset">Inset border</div><hr>
37     <div class = "thick dashed">Thick dashed border</div><hr>
38     <div class = "thin red solid">Thin red solid border</div><hr>
39     <div class = "medium blue outset">Medium blue outset border</div>
40 </body>
41 </html>
```

**Fig. 4.14** | Borders of block-level elements. (Part 2 of 3.)



**Fig. 4.14** | Borders of block-level elements. (Part 3 of 3.)

# 4.10 Box Model and Text Flow (Cont.)



## *Floating Elements*

- ▶ Floating allows you to move an element to one side of the screen; other content in the document then *flows around* the floated element.
- ▶ Figure 4.15 demonstrates how floating elements and the box model can be used to control the layout of an entire page.





```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.15: floating.html -->
4  <!-- Floating elements. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Flowing Text Around Floating Elements</title>
9          <style type = "text/css">
10             header    { background-color: skyblue;
11                         text-align: center;
12                         font-family: arial, helvetica, sans-serif;
13                         padding: .2em; }
14             p          { text-align: justify;
15                         font-family: verdana, geneva, sans-serif;
16                         margin: .5em; }
17             h1         { margin-top: 0px; }
```

**Fig. 4.15** | Floating elements. (Part I of 4.)



```
18     .floated { background-color: lightgrey;
19                font-size: 1.5em;
20                font-family: arial, helvetica, sans-serif;
21                padding: .2em;
22                margin-left: .5em;
23                margin-bottom: .5em;
24                float: right;
25                text-align: right;
26                width: 50%; }
27     section { border: 1px solid skyblue; }
28 </style>
29 </head>
30 <body>
31     <header><img src = "deitel.png" alt = "Deitel" /></header>
32     <section>
33         <h1 class = "floated">Corporate Training and Authoring</h1>
34         <p>Deitel & Associates, Inc. is an internationally
35             recognized corporate training and authoring organization
36             specializing in programming languages, Internet/web
37             technology, iPhone and Android app development and
38             object technology education. The company provides courses
39             on Java, C++, C#, Visual Basic, C, Internet and web
40             programming, Object Technology and iPhone and Android
41             app development.</p>
42     </section>
```

**Fig. 4.15** | Floating elements. (Part 2 of 4.)



```
43     <section>
44         <h1 class = "floated">Programming Books and Videos</h1>
45         <p>Through its publishing
46             partnership with Pearson, Deitel & Associates,
47             Inc. publishes leading-edge programming textbooks,
48             professional books and interactive web-based and DVD
49             LiveLessons video courses.</p>
50     </section>
51 </body>
52 </html>
```

**Fig. 4.15** | Floating elements. (Part 3 of 4.)



**Fig. 4.15** | Floating elements. (Part 4 of 4.)

# 4.10 Box Model and Text Flow (Cont.)



## *margin and padding Properties*

- ▶ The **margin** property sets the space between the outside of an element's border and all other content on the page.
- ▶ The **padding** property determines the distance between the content inside an element and the inside of the element's border.
- ▶ Margins for individual sides of an element can be specified by using the properties **margin-top**, **margin-right**, **margin-left** and **margin-bottom**.
- ▶ Padding can be specified in the same way, using **padding-top**, **padding-right**, **padding-left** and **padding-bottom**.

# 4.11 Media Types and Media Queries



- ▶ CSS media types
  - allow you to decide what a page should look like depending on the kind of media being used to display the page
  - Most common media type for a web page is the screen media type, which is a standard computer screen



## 4.11 Media Types and Media Queries (Cont.)

- ▶ A block of styles that applies to all media types is declared by `@media all` and enclosed in curly braces
- ▶ To create a block of styles that apply to a single media type such as `print`, use `@media print` and enclose the style rules in curly braces



# 4.11 Media Types and Media Queries (Cont.)

- ▶ Other media types in CSS include:
  - handheld
    - Designed for mobile Internet devices
  - braille
    - For machines that can read or print web pages in braille
  - speech
    - Allow the programmer to give a speech-synthesizing web browser more information about the content of the web page
  - print
    - Affects a web page's appearance when it is printed





## 4.11 Media Types and Media Queries (Cont.)

- ▶ Figure 4.16 gives a simple classic example that applies one set of styles when the document is viewed on all media (including screens) other than a printer, and another when the document is printed.
- ▶ To see the difference, look at the screen captures below the paragraph or use the Print Preview feature in your browser if it has one.



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.16: mediatypes.html -->
4  <!-- CSS media types. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>Media Types</title>
9          <style type = "text/css">
10             @media all
11             {
12                 body { background-color: steelblue; }
13                 h1   { font-family: verdana, helvetica, sans-serif;
14                       color: palegreen; }
15                 p    { font-size: 12pt;
16                       color: white;
17                       font-family: arial, sans-serif; }
18             } /* End @media all declaration. */
```

**Fig. 4.16** | CSS media types. (Part I of 4.)

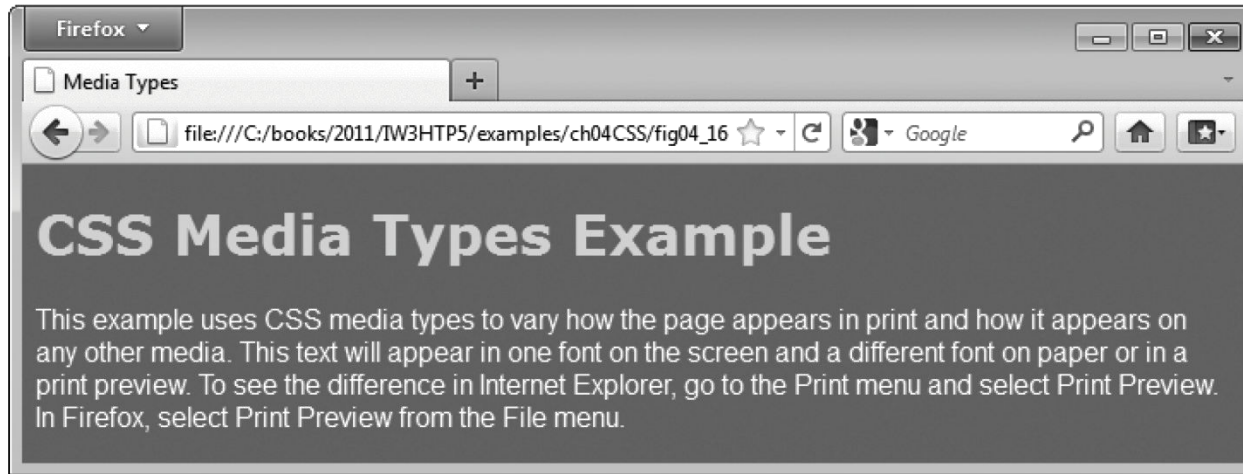


```
19      @media print
20      {
21          body { background-color: white; }
22          h1   { color: seagreen; }
23          p    { font-size: 14pt;
24                color: steelblue;
25                font-family: "times new roman", times, serif; }
26      } /* End @media print declaration. */
27  </style>
28  </head>
29  <body>
30      <h1>CSS Media Types Example</h1>
31
32      <p>
33          This example uses CSS media types to vary how the page
34          appears in print and how it appears on any other media.
35          This text will appear in one font on the screen and a
36          different font on paper or in a print preview. To see
37          the difference in Internet Explorer, go to the Print
38          menu and select Print Preview. In Firefox, select Print
39          Preview from the File menu.
40      </p>
41  </body>
42  </html>
```

**Fig. 4.16** | CSS media types. (Part 2 of 4.)



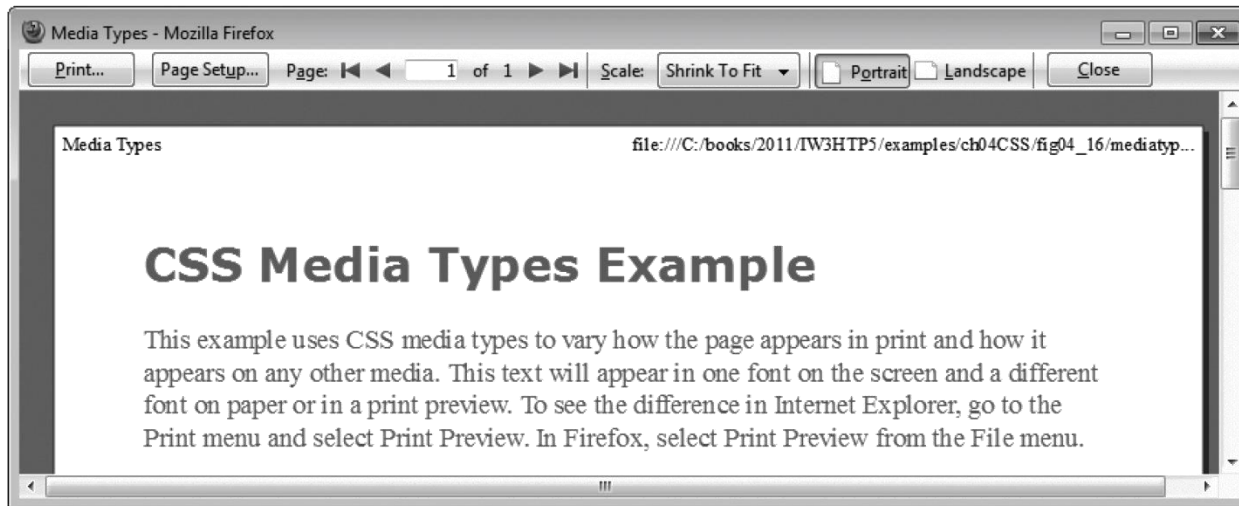
a) Background color appears on the screen.



**Fig. 4.16** | CSS media types. (Part 3 of 4.)



b) Background color is set to white for the `print` media type.



**Fig. 4.16** | CSS media types. (Part 4 of 4.)



## Look-and-Feel Observation 4.1

---

Pages with dark background colors and light text use a lot of ink and may be difficult to read when printed, especially on a black-and white-printer. Use the print media type to avoid this.



## Look-and-Feel Observation 4.2

---

In general, sans-serif fonts look better on a screen, while serif fonts look better on paper. The `print` media type allows your web page to display a sans-serif font on a screen and change to a serif font when it's printed.



# 4.11 Media Types and Media Queries (Cont.)

## *Media Queries*

- ▶ Allow you to format your content to specific output devices.
- ▶ Include a media type and expressions that check the media features of the output device.
- ▶ Common media features include:
  - width—the width of the part of the screen on which the document is rendered, including any scrollbars
  - height—the height of the part of the screen on which the document is rendered, including any scrollbars
  - device-width—the width of the screen of the output device
  - device-height—the height of the screen of the output device
  - orientation—if the height is greater than the width, orientation is portrait, and if the width is greater than the height, orientation is landscape
  - aspect-ratio—the ratio of width to height
  - device-aspect-ratio—the ratio of device-width to device-height





## 4.12 Drop-Down Menus

- ▶ `:hover` pseudoclass
  - used to apply styles to an element when the mouse cursor is over it
- ▶ `display` property
  - allows a programmer to decide if an element is displayed as a block element, inline element, or is not rendered at all (none)



```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.17: dropdown.html -->
4  <!-- CSS drop-down menu. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>
9              Drop-Down Menu
10         </title>
11         <style type = "text/css">
12             body                { font-family: arial, sans-serif }
13             nav                  { font-weight: bold;
14                                   color: white;
15                                   border: 2px solid royalblue;
16                                   text-align: center;
17                                   width: 10em;
18                                   background-color: royalblue; }
19             nav ul               { display: none;
20                                   list-style: none;
21                                   margin: 0;
22                                   padding: 0; }
23             nav:hover ul         { display: block }
```

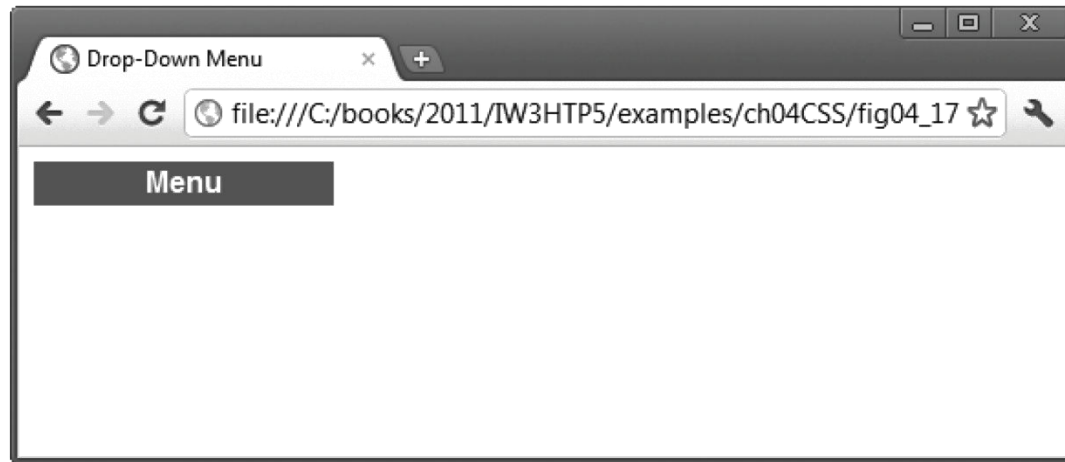
**Fig. 4.17** | CSS drop-down menu. (Part 1 of 5.)



```
24         nav ul li      { border-top: 2px solid royalblue;
25                           background-color: white;
26                           width: 10em;
27                           color: black; }
28         nav ul li:hover { background-color: powderblue; }
29         a                { text-decoration: none; }
30     </style>
31 </head>
32 <body>
33     <nav>Menu
34         <ul>
35             <li><a href = "#">Home</a></li>
36             <li><a href = "#">News</a></li>
37             <li><a href = "#">Articles</a></li>
38             <li><a href = "#">Blog</a></li>
39             <li><a href = "#">Contact</a></li>
40         </ul>
41     </nav>
42 </body>
43 </html>
```

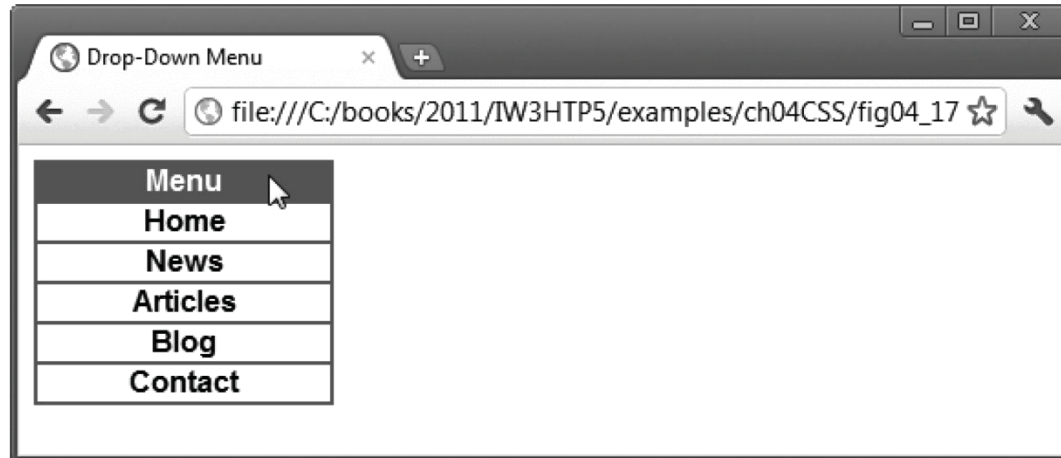
**Fig. 4.17** | CSS drop-down menu. (Part 2 of 5.)

a) A collapsed menu



**Fig. 4.17** | CSS drop-down menu. (Part 3 of 5.)

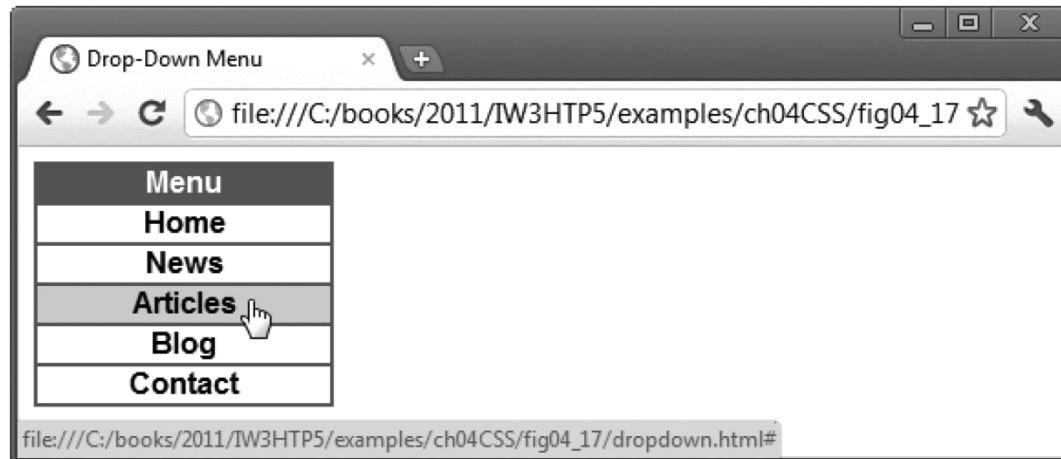
b) A drop-down menu is displayed when the mouse cursor is hovered over **Menu**



**Fig. 4.17** | CSS drop-down menu. (Part 4 of 5.)



c) Hovering the mouse cursor over a menu link highlights the link



**Fig. 4.17** | CSS drop-down menu. (Part 5 of 5.)



## 4.13 (Optional) User Style Sheets

- ▶ Users can define their own user style sheets to format pages based on their preferences
- ▶ Absolute font size measurements override user style sheets, while relative font sizes will yield to a user-defined style
- ▶ User style sheets are not linked to a document; rather, they are set in the browser's options



## 4.13 (Optional) User Style Sheets

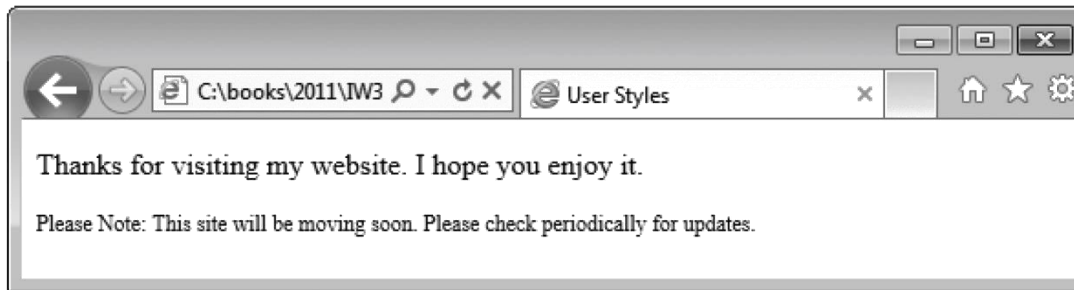
- ▶ Figure 4.18 contains an author style.
- ▶ User style sheets are external style sheets. Figure 4.19 shows a user style sheet that sets the body's font-size to 20pt, color to yellow and background-color to navy. The font-size value specified in the user style sheet conflicts with the one in line 10 of Fig. 4.18.





```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.18: user_absolute.html -->
4  <!-- pt measurement for text size. -->
5  <html>
6      <head>
7          <meta charset = "utf-8">
8          <title>User Styles</title>
9          <style type = "text/css">
10             .note { font-size: 9pt; }
11          </style>
12      </head>
13      <body>
14          <p>Thanks for visiting my website. I hope you enjoy it.
15          </p><p class = "note">Please Note: This site will be
16          moving soon. Please check periodically for updates.</p>
17      </body>
18  </html>
```

**Fig. 4.18** | pt measurement for text size. (Part I of 2.)



**Fig. 4.18** | pt measurement for text size. (Part 2 of 2.)



```
1  /* Fig. 4.19: userstyles.css */
2  /* A user style sheet */
3  body      { font-size: 20pt;
4              color: yellow;
5              background-color: navy; }
```

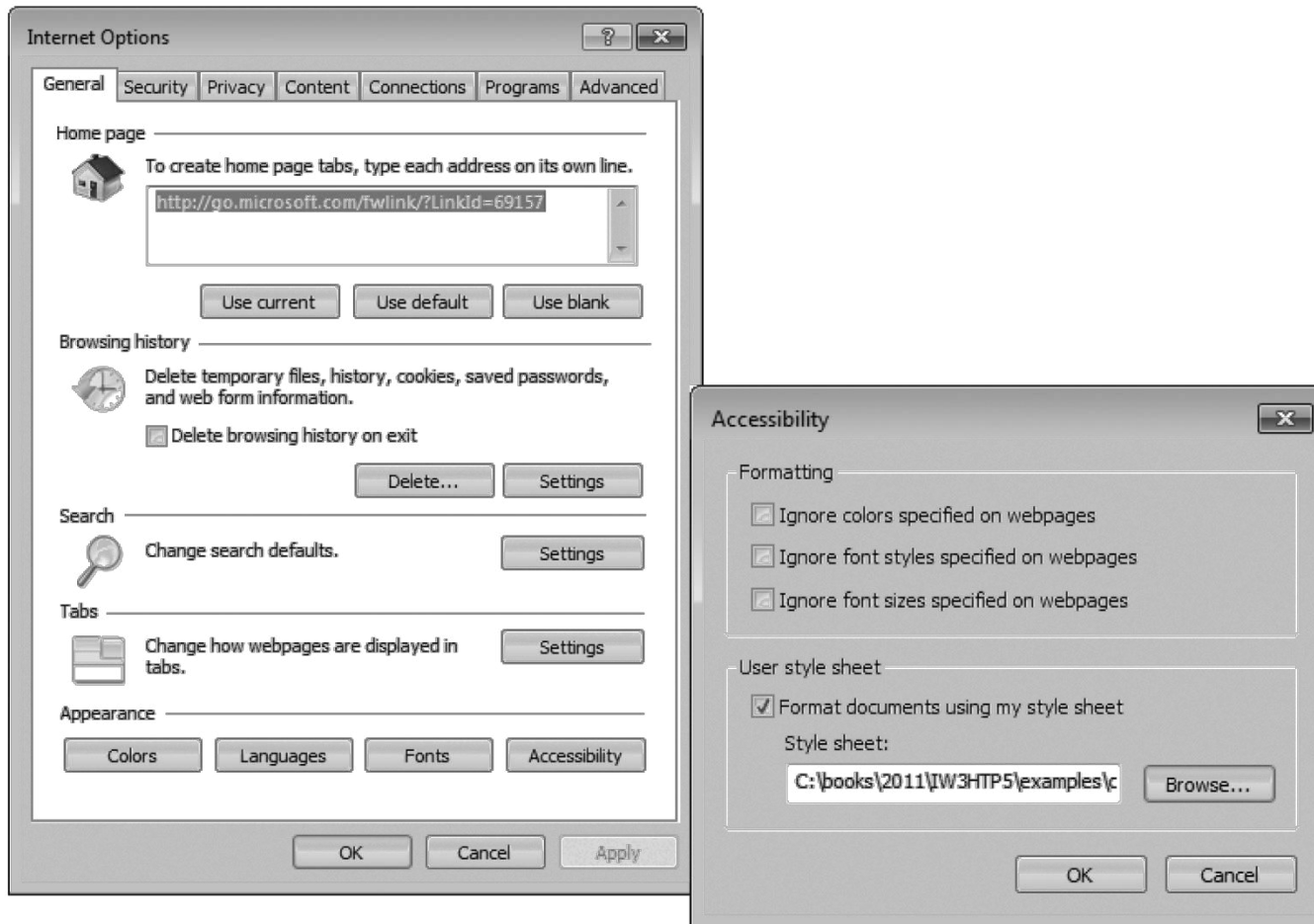
**Fig. 4.19** | A user style sheet.



## 4.13 (Optional) User Style Sheets

### *Adding a User Style Sheet*

- ▶ User style sheets are not linked to a document; rather, they're set in the browser's options.
- ▶ To add a user style sheet in IE9, select **Internet Options...**, located in the **Tools** menu. In the **Internet Options** dialog (Fig. 4.20) that appears, click **Accessibility...**, check the **Format documents using my style sheet** checkbox, and type the location of the user style sheet.
- ▶ To add a user style sheet in Firefox, find your Firefox profile using the instructions at
  - [www.mozilla.org/support/firefox/profile#locate](http://www.mozilla.org/support/firefox/profile#locate)
- ▶ Place a style sheet called `userContent.css` in the chrome subdirectory.
- ▶ For information on adding a user style sheet in Chrome, see
  - [www.google.com/support/forum/p/Chrome/thread?tid=1fa0dd079dbdc2ff&hl=en](http://www.google.com/support/forum/p/Chrome/thread?tid=1fa0dd079dbdc2ff&hl=en).

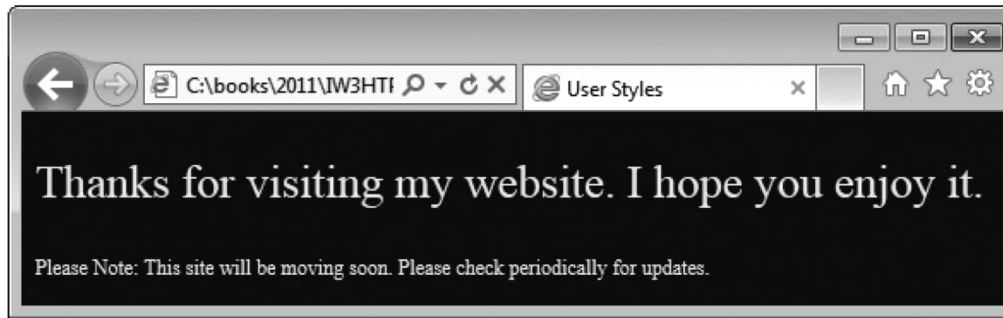


**Fig. 4.20** | User style sheet in Internet Explorer 9.



## 4.13 (Optional) User Style Sheets

- ▶ The web page from Fig. 4.18 is displayed in Fig. 4.21, with the user style sheet from Fig. 4.19 applied.



**Fig. 4.21** | User style sheet applied with pt measurement.



## 4.13 (Optional) User Style Sheets

### *Defining font-size in a User Style Sheet*

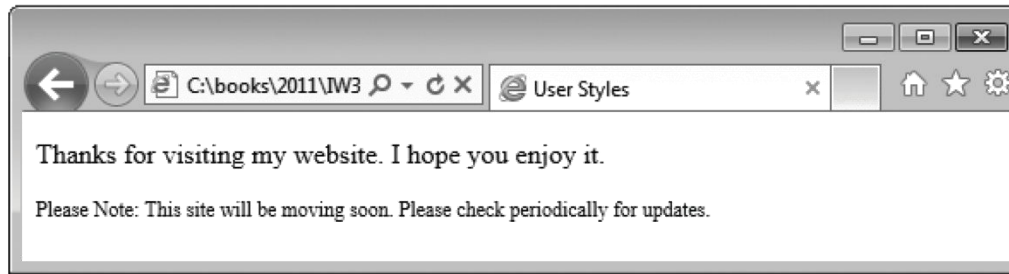
- ▶ Figure 4.22 changes the font-size property to use a relative measurement that does not override the user style set in Fig. 4.19.
- ▶ Instead, the font size displayed is relative to the one specified in the user style sheet.





```
1  <!DOCTYPE html>
2
3  <!-- Fig. 4.22: user_relative.html -->
4  <!-- em measurement for text size. -->
5  <html>
6    <head>
7      <meta charset = "utf-8">
8      <title>User Styles</title>
9      <style type = "text/css">
10        .note { font-size: .75em; }
11      </style>
12    </head>
13    <body>
14      <p>Thanks for visiting my website. I hope you enjoy it.
15      </p><p class = "note">Please Note: This site will be
16      moving soon. Please check periodically for updates.</p>
17    </body>
18  </html>
```

**Fig. 4.22** | em measurement for text size. (Part I of 2.)

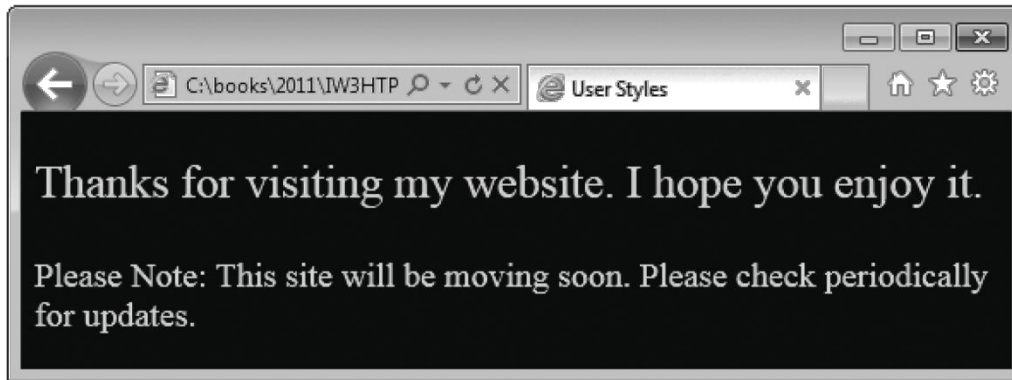


**Fig. 4.22** | em measurement for text size. (Part 2 of 2.)



## 4.13 (Optional) User Style Sheets

- ▶ Figure 4.23 displays the web page from Fig. 4.22 in Internet Explorer with the user style sheet from Fig. 4.19 applied.
- ▶ Note that the second line of text displayed is larger than the same line of text in Fig. 4.21.



---

**Fig. 4.23** | User style sheet applied with em measurement.