19

Dynamic HTML: Cascading Style Sheets™ (CSS)

- Objectives
- To take control of the appearance of a Web site by creating your own style sheets.
- To use a style sheet to give all the pages of a Web site the same look and feel.
- To use the **`CLASS`** attribute to apply styles.
- To specify the precise font, size, color and other properties of displayed text.
- To specify element backgrounds and colors.
- To understand the box model and be able to control the margins, borders, padding.
- To truly separate content and presentation.

Fashions fade, style is eternal.
Yves Saint Laurent

A style does not go out of style as long as it adapts itself to its period. When there is an incompatibility between the style and a certain state of mind, it is never the style that triumphs.
Coco Chanel

How liberating to work in the margins, outside a central perception.
Don DeLillo

Our words have wings, but fly not where we would.
George Eliot

There are aphorisms that, like airplanes, stay up only while they are in motion.
Vladimir Nabokov

Outline

19.1 Introduction

Cascading Style Sheets (*CSS*) allow you to specify the style of your page elements (spacing, margins, etc.) separately from the structure of your document (section headers, body text, links, etc.). This *separation of structure from content* allows greater manageability and makes changing the style of your document easier.

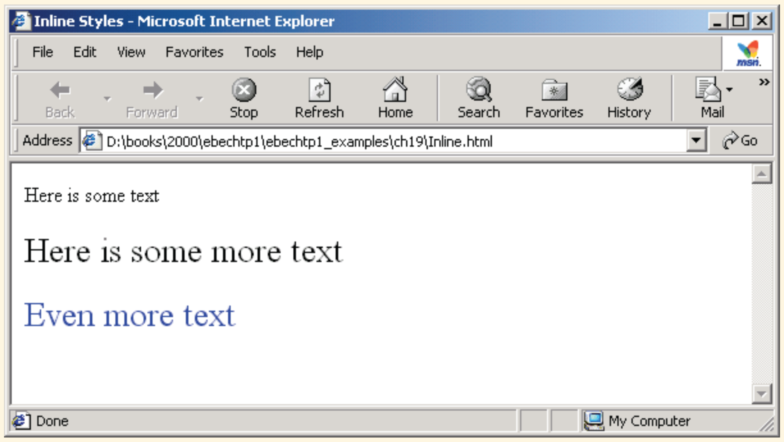
19.2 Inline Styles

There are many ways to declare styles for a document. Figure 19.1 presents *inline styles* in which an individual element's style is declared using the ***STYLE*** attribute.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig. 19.1: inline.html -->
<!-- Using inline styles    -->

<HEAD><TITLE>Inline Styles</TITLE></HEAD>

<BODY>

<P>Here is some text</P>

<!-- The STYLE attribute allows you to declare inline   -->
<!-- styles. Separate multiple styles with a semicolon. -->
<P STYLE = "font-size: 20pt">Here is some more text</P>
<P STYLE = "font-size: 20pt; color: #0000FF">Even more text</P>
```

Fig. 19.1 Inline styles (part 1 of 2).

```
17
18  </BODY>
19  </HTML>
```


Inline Styles - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\Inline.html
Go
Done
My Computer

Here is some text

Here is some more text

Even more text

Fig. 19.1 Inline styles (part 2 of 2).

Our first inline style declaration appears on line 15:

```
<P STYLE = "font-size: 20pt">Here is some more text</P>
```

The ***`STYLE` attribute*** allows you to specify a style for an element. Each *CSS property* (in this case, **`font-size`**) is followed by a colon then the value of that attribute. In the preceding HTML line we declare the **`P`** element (of only that line) to have 20-point text size.

Line 16

```
<P STYLE = "font-size: 20pt; color: #0000FF">Even more text</P>
```

specifies two properties separated by a semicolon. In this line we also set the **`color`** of the text to blue using the hex code **`#0000FF`**. Color names (see Appendix B) may be used in place of hex codes as we will see in the next example. Note that inline styles override any other styles applied by the methods we cover later in this chapter.

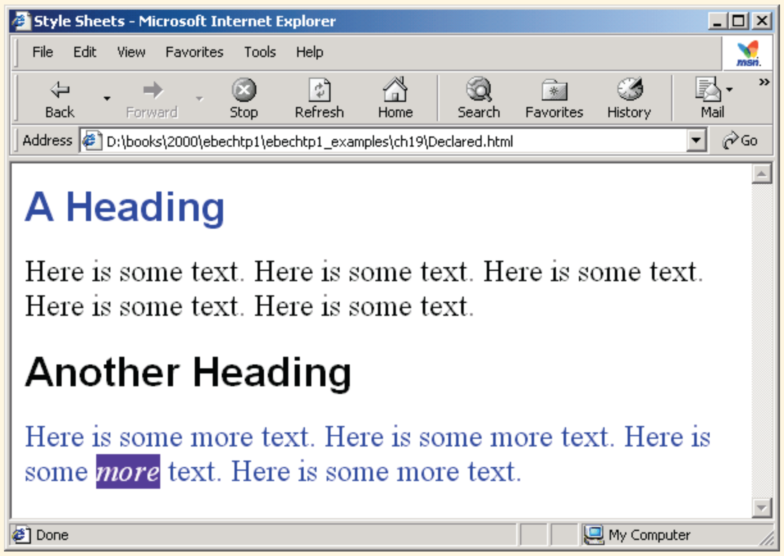
19.3 Creating Style Sheets with the `STYLE` Element

In Fig. 19.2 we declare, in the header section of the document, styles that may be applied to the entire document.

```
1   <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2   <HTML>
3
4   <!-- Fig. 19.2: declared.html                    -->
5   <!-- Declaring a style sheet in the header section. -->
6
```

Fig. 19.2 Declaring styles in the header section of a document (part 1 of 2).

```
<HEAD>
<TITLE>Style Sheets</TITLE>

<!-- This begins the style sheet section. -->
<STYLE TYPE = "text/css">

   EM     { background-color: #8000FF;
            color: white }

   H1     { font-family: Arial, sans-serif }

   P      { font-size: 18pt }

   .blue { color: blue }

</STYLE>
</HEAD>

<BODY>

<!-- This CLASS attribute applies the .blue style -->
<H1 CLASS = "blue">A Heading</H1>
<P>Here is some text. Here is some text. Here is some text.
Here is some text. Here is some text.</P>

<H1>Another Heading</H1>
<P CLASS = "blue">Here is some more text. Here is some more text.
Here is some <EM>more</EM> text. Here is some more text.</P>

</BODY>
</HTML>
```


Style Sheets - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\Declared.html
Go
Done
My Computer

A Heading

Here is some text. Here is some text. Here is some text. Here is some text. Here is some text.

Another Heading

Here is some more text. Here is some more text. Here is some *more* text. Here is some more text.

Fig. 19.2 Declaring styles in the header section of a document (part 2 of 2).

The element in the header section on line 11

```
<STYLE TYPE = "text/css">
```

begins the *style sheet*. Styles that are placed here apply to the whole document, not just a single element. The ***TYPE attribute*** specifies the *MIME type* of the following style sheet. MIME is a standard for specifying the format of content—some other MIME types are **text/html**, **image/gif**, and **text/javascript**. Regular text style sheets always use the MIME type **text/css**.

Look-and-Feel Observation 19.1

Without style sheets, the browser completely controls the look and feel of Web pages. With style sheets, the designer can specify the look and feel of all elements on a Web page.

The body of the **STYLE** sheet on lines 13 through 20

```
EM    { background-color: #8000FF;
        color: white }

H1    { font-family: Arial, sans-serif }

P     { font-size: 18pt }

.blue { color: blue }
```

declares the *CSS rules* for this style sheet. We declare rules for the **EM**, **H1** and **P** elements. All **EM**, **H1** and **P** elements in this document will be modified in the specified manner. CSS is a powerful tool for applying universal formatting. Notice that each rule body begins and ends with a curly brace (**{** and **}**). We also declare a *style class* named **blue** on line 20. All class declarations are preceded with a period and are applied to elements only of that specific class (as we will see below).

The CSS rules in a style sheet use the same format as inline styles—the property is followed by a colon (**:**) and the value of that property. Multiple properties are separated with a semicolon (**;**) as in the preceding **EM** style rule.

The ***color*** property specifies the color of text in an element. The ***background-color*** property specifies the background color of the element (like the **BGCOLOR** attribute in HTML does).

The ***font-family*** property (line 16) specifies the name of the font that should be displayed. In this case, we use the **Arial** font. The second value, ***sans-serif***, is a *generic font family*. Generic font families allow you to specify a type of font instead of a specific font. This allows much greater flexibility in your site display. In this example, if the **Arial** font is not found on the system, the browser will instead display another **sans-serif** font (such as **Helvetica** or **Verdana**). Other generic font families are ***serif*** (e.g., **Times New Roman** or **Georgia**), ***cursive*** (e.g., **Script**), ***fantasy*** (e.g., **Critter**) and ***monospace*** (e.g., **Courier** or **Fixedsys**).

The ***font-size*** property specifies the size to use to render the font—in this case we use 18 points. Other possible measurements besides **pt** are covered later in the chapter. You can also use the relative values ***xx-small***, ***x-small***, ***small***, ***smaller***, ***medium***, ***large***, ***larger***, ***x-large*** and ***xx-large***.

On line 28

```
<H1 CLASS = "blue">A Heading</H1>
```

the ***`CLASS` attribute*** applies a style class, in this case **`blue`** (this was declared as **`.blue`** in the **`STYLE`** sheet). Note that the text appears on screen with *both* the properties of an **`H1`** element and the properties of the **`.blue`** style class applied.

On lines 33 and 34

```
<P CLASS = "blue">Here is some more text. Here is some more text.
Here is some <EM>more</EM> text. Here is some more text.</P>
```

The **`P`** element and the **`.blue`** class style are both applied to the whole text. All styles applied to an element (the *parent element*) also apply to elements inside that element (*child elements*). The word inside the **`EM`** element *inherits* the **`P`** style (namely, the 18-point font size of line 18), but it conflicts with the **`color`** attribute of the **`blue`** class. Because styles declared in child element are more specific (have greater *specificity*) than parent element styles, the **`EM`** style overrides the styles set in the **`blue`** class.

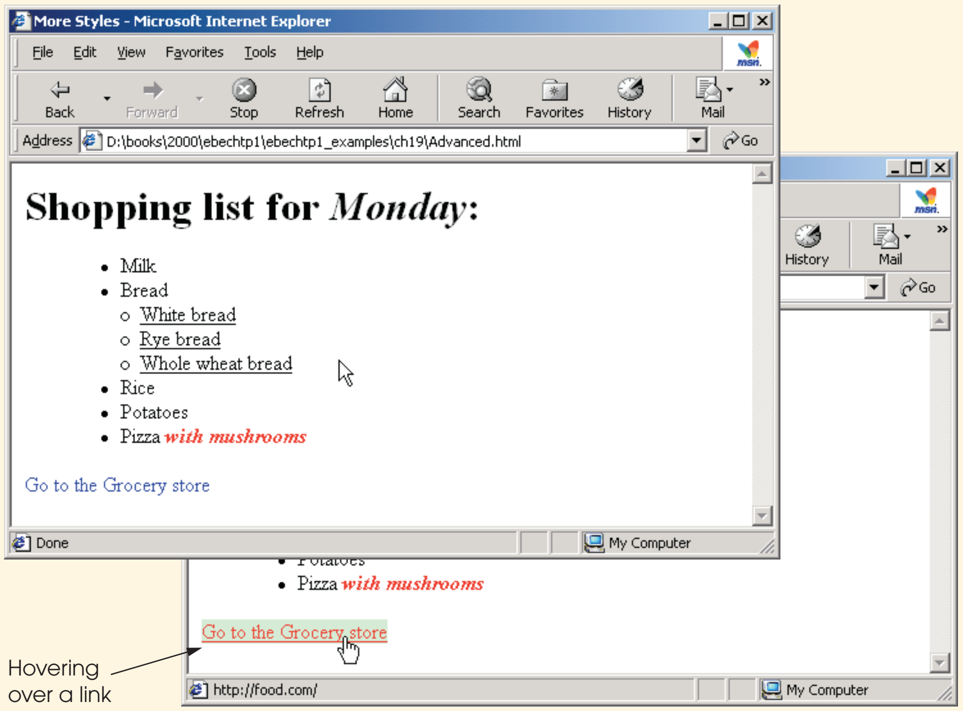
19.4 Conflicting Styles

Figure 19.3 has more examples of *inheritance* and *specificity*.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig 19.3: advanced.html    -->
<!-- More advanced style sheets -->

<HEAD>
<TITLE>More Styles</TITLE>
<STYLE TYPE = "text/css">

   A.nodec  { text-decoration: none }

   A:hover  { text-decoration: underline;
              color: red;
              background-color: #CCFFCC }

   LI EM    { color: red;
              font-weight: bold }

   UL       { margin-left: 75px }

   UL UL    { text-decoration: underline;
              margin-left: 15px }

```

Fig. 19.3 Inheritance in style sheets (part 1 of 2).

```
</STYLE>
</HEAD>

<BODY>

<H1>Shopping list for <EM>Monday</EM>:</H1>
<UL>
<LI>Milk</LI>
<LI>Bread
   <UL>
   <LI>White bread</LI>
   <LI>Rye bread</LI>
   <LI>Whole wheat bread</LI>
   </UL></LI>
<LI>Rice</LI>
<LI>Potatoes</LI>
<LI>Pizza <EM>with mushrooms</EM></LI>
</UL>

<P><A CLASS = "nodec" HREF = "http://food.com">Go to the Grocery
   store</A></P>

</BODY>
</HTML>
```


More Styles - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\Advanced.html
Go
Done
My Computer
http://food.com/
Hovering over a link

Shopping list for *Monday*:

- Milk
- Bread
  - White bread
  - Rye bread
  - Whole wheat bread
- Rice
- Potatoes
- Pizza *with mushrooms*

Go to the Grocery store

Fig. 19.3 Inheritance in style sheets (part 2 of 2).

Line 11

```
A.nodec  { text-decoration: none }
```

applies the ***`text-decoration property`*** to all **`A`** elements whose **`CLASS`** attribute is set to **`nodec`**. The default browser rendering of an **`A`** element is to underline, but here we set it to **`none`**. The **`text-decoration`** property applies *decorations* to text within an element. Other possible values are ***`overline`***, ***`line-through`*** and ***`blink`***.

The **`.nodec`** appended to **`A`** is an extension of class styles—this style will apply only to **`A`** elements that specify **`nodec`** as their class.

Lines 13 through 15

```
A:hover  { text-decoration: underline;
           color: red;
           background-color: #CCFFCC }
```

specify a style for **`hover`**, which is a *pseudo-class*. Pseudo-classes give the author access to content not specifically declared in the document. The **`hover`** pseudo-class is dynamically activated when the user moves the mouse cursor over an **`A`** element.

Portability Tip 19.1

*Browsers are not required to support the **`blink`** value of the **`text-decoration`** property, so do not use it as a mechanism for important highlighting.*

Portability Tip 19.2

Always test DHTML programs on all intended client platforms to ensure that the display is reasonable, especially for those client platforms with older browsers.

Lines 17 and 18

```
LI EM   { color: red;
          font-weight: bold }
```

declare a style for all **`EM`** elements that are children of **`LI`** elements. In the screen output of Fig. 19.3 notice that **`Monday`** is not made red and bold, because it is not encapsulated by an **`LI`** element as **`with mushrooms`** is.

The declaration syntax for applying rules to multiple elements is similar. If you instead wanted to apply the rule on lines 17 and 18 to both **`LI`** and **`EM`** elements, you would separate the elements with commas, as follows:

```
LI, EM  { color: red;
          font-weight: bold }
```

Lines 22 and 23

```
 UL UL   { text-decoration: underline;
           margin-left: 15px }
```

specify that all nested lists (**`UL`** elements that are children of **`UL`** elements) will be underlined and have a left-hand margin of 15 pixels (margins and the box model will be covered in Section 19.9).

A pixel is a *relative-length* measurement—it varies in size based on screen resolution. Other relative lengths are **em** (the size of the font), **ex** (the so-called "x-height" of the font, which is usually set to the height of a lowercase x) and percentages (e.g., **margin-left: 10%**). To set an element to display text at 150% of its normal size, you could use the syntax

```
font-size: 1.5em
```

The other units of measurement available in CSS are *absolute-length* measurements, i.e., units that do not vary in size based on the system. These are **in** (inches), **cm** (centimeters), **mm** (millimeters), **pt** (points—1 **pt**=1/72 **in**) and **pc** (picas—1 **pc** = 12 **pt**).

Good Programming Practice 19.1

Whenever possible, use relative length measurements. If you use absolute length measurements, you might override styles preset by the user.

Software Engineering Observation 19.1

*There are three possible sources for styles sheets—browser defaults, preset user styles, and author styles (e.g., in the **STYLE** section of a document). Author styles have a greater precedence than preset user styles, so any conflicts will be resolved in favor of the author styles.*

In Fig. 19.3, the whole list is indented because of the 75-pixel left-hand margin for top-level **UL** elements, but the nested list is indented only 15 pixels (not another 75 pixels) because the child **UL** element's **margin-left** property overrides the parent **UL** element's **margin-left** property.

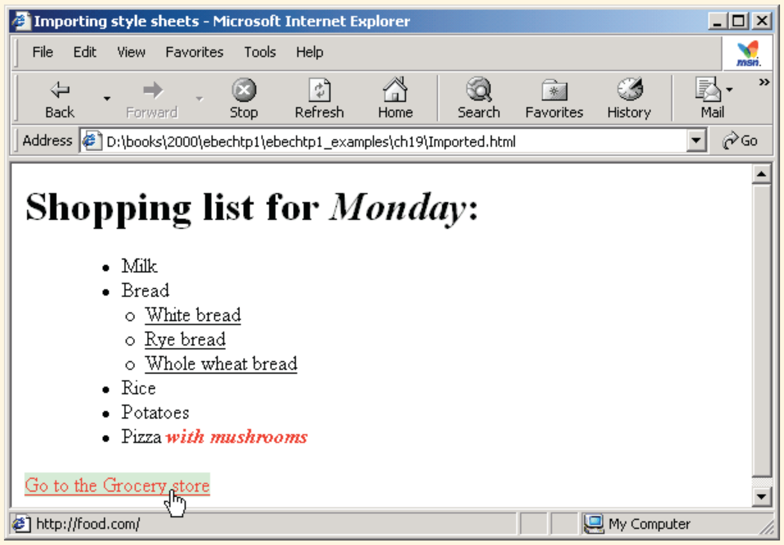
19.5 Linking External Style Sheets

As we have seen, style sheets are an efficient way to give a document a uniform theme. With *external linking*, you can give your whole Web site the same uniform look—separate pages on your site could all use the same style sheet, and you would have to modify only a single file to make changes to styles across your whole Web site. Figure 19.4 shows the external style sheet, and Fig. 19.5 shows the syntax for including the external style sheet.

```
A       { text-decoration: none }

A:hover { text-decoration: underline;
          color: red;
          background-color: #CCFFCC }

LI EM   { color: red;
          font-weight: bold}

UL      { margin-left: 2cm }

UL UL   { text-decoration: underline;
          margin-left: .5cm }
```

Fig. 19.4 An external style sheet (**styles.css**).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig. 19.5: imported.html        -->
<!-- Linking external style sheets  -->

<HEAD>
<TITLE>Importing style sheets</TITLE>
<LINK REL = "stylesheet" TYPE = "text/css" HREF = "styles.css">
</HEAD>

<BODY>

<H1>Shopping list for <EM>Monday</EM>:</H1>
<UL>
<LI>Milk</LI>
<LI>Bread
   <UL>
   <LI>White bread</LI>
   <LI>Rye bread</LI>
   <LI>Whole wheat bread</LI>
   </UL></LI>
<LI>Rice</LI>
<LI>Potatoes</LI>
<LI>Pizza <EM>with mushrooms</EM></LI>
</UL>

<A HREF = "http://food.com">Go to the Grocery store</A>

</BODY>
</HTML>
```


Importing style sheets - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\Imported.html
Go

Shopping list for *Monday*:

- Milk
- Bread
  - White bread
  - Rye bread
  - Whole wheat bread
- Rice
- Potatoes
- Pizza *with mushrooms*

Go to the Grocery store

http://food.com/
My Computer

Fig. 19.5 Linking an external style sheet.

Line 9

```
<LINK REL = "stylesheet" TYPE = "text/css" HREF =
"styles.css">
```

shows a ***`LINK element`***, which specifies a *relationship* between the current document and another document using the ***`REL attribute`***. In this case, we declare the linked document to be a ***`stylesheet`*** for this document. We use the **`TYPE`** attribute to specify the MIME type as **`text/css`** and provide the URL for the stylesheet with the **`HREF`** attribute.

Software Engineering Observation 19.2

Style sheets are reusable. Creating style sheets once and reusing them reduces programming effort.

Software Engineering Observation 19.3

*The **`LINK`** element can be placed only in the header section. Other relationships you can specify between documents are **`next`** and **`previous`**, which would allow you to link a whole series of documents. This could let browsers print a large collection of related documents at once (in Internet Explorer, select **Print all linked documents** in the **Print...** submenu of the **File** menu).*

19.6 Positioning Elements

In the past, controlling the positioning of elements in an HTML document was difficult; positioning was basically up to the browser. CSS introduces the **`position`** property and a capability called *absolute positioning*, which gives us greater control over how our documents are displayed (Fig. 19.6).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig 19.6: positioning.html       -->
<!-- Absolute positioning of elements -->

<HEAD>
<TITLE>Absolute Positioning</TITLE>
</HEAD>

<BODY>

<IMG SRC = "i.gif" STYLE = "position: absolute; top: 0px;
   left: 0px; z-index: 1">
<H1 STYLE = "position: absolute; top: 50px; left: 50px;
   z-index: 3">Positioned Text</H1>
<IMG SRC = "circle.gif" STYLE = "position: absolute; top: 25px;
   left: 100px; z-index: 2">

</BODY>
</HTML>
```

Fig. 19.6 Positioning elements with CSS (part 1 of 2).

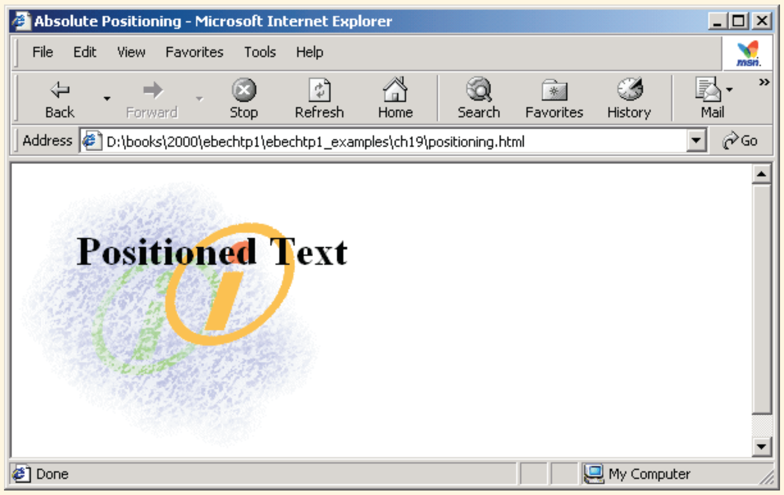

Absolute Positioning - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\positioning.html
Go
Positioned Text
Done
My Computer

Fig. 19.6 Positioning elements with CSS (part 2 of 2).

Lines 13 and 14

```
<IMG SRC = "i.gif" STYLE = "position: absolute; top: 0px;
   left: 0px; z-index: 1">
```

position the first **`IMG`** element (**`i.gif`**) on the page. Specifying an element's **`position`** as ***`absolute`*** removes it from the normal flow of elements on the page and instead, positions the element according to distance from the ***`top`***, ***`left`***, ***`right`*** or ***`bottom`*** margins of its parent element. Here we position the element to be **`0`** pixels away from both the **`top`** and **`left`** margins of the **`BODY`** element (the parent element).

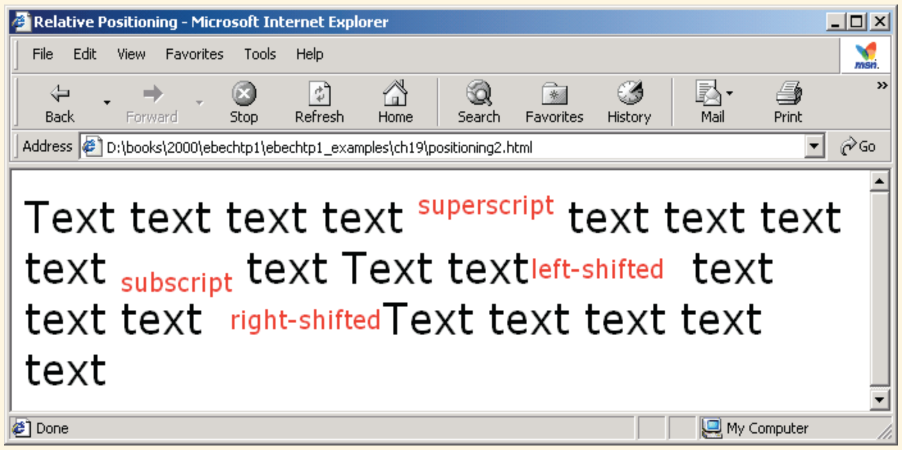
The ***`z-index attribute`*** allows you to properly layer overlapping elements. Elements that have higher **`z-index`** values are displayed in front of elements with lower **`z-index`** values. In this example, **`i.gif`**, with a **`z-index`** of 1, is displayed at the back; **`circle.gif`**, with a **`z-index`** of 2, is displayed in front of that; the **`H1`** element ("Positioned Text"), with a **`z-index`** of 3, is displayed in front of both of the others. If you do not specify **`z-index`**, the elements that occur later in the document are displayed in front of those that occur earlier.

Absolute positioning is not the only way to specify page layout—*relative positioning* is shown in Fig. 19.7.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig 19.7: positioning2.html      -->
<!-- Relative positioning of elements -->

<HEAD>
<TITLE>Relative Positioning</TITLE>
```

Fig. 19.7 Relative positioning of elements (part 1 of 2).

```

<STYLE TYPE = "text/css">

   P       { font-size: 2em;
             font-family: Verdana, Arial, sans-serif }

   SPAN    { color: red;
             font-size: .6em;
             height: 1em }

   .super  { position: relative;
             top: -1ex }

   .sub    { position: relative;
             bottom: -1ex }

   .shiftl { position: relative;
             left: -1ex }

   .shiftr { position: relative;
             right: -1ex }
</STYLE>
</HEAD>

<BODY>

<P>
Text text text text <SPAN CLASS = "super">superscript</SPAN>
text text text text <SPAN CLASS = "sub">subscript</SPAN>
text Text text <SPAN CLASS = "shiftl">left-shifted</SPAN>
text text text <SPAN CLASS = "shiftr">right-shifted</SPAN>
Text text text text text
</P>

</BODY>
</HTML>
```


Relative Positioning - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\positioning2.html
Go
Done
My Computer

Text text text text superscript text text text text subscript text Text textleft-shifted text text text right-shiftedText text text text text

Fig. 19.7 Relative positioning of elements (part 2 of 2).

Setting the **`position`** property to ***`relative`***, as in lines 19 and 20,

```
.super  { position: relative;
          top: -1ex }
```

will first lay out the element on the page, then offset the element by the specified **`top`**, **`bottom`**, **`left`** or **`right`** values. Unlike absolute positioning, relative positioning keeps elements in the general flow of elements on the page.

Common Programming Error 19.1

Because relative positioning keeps elements in the flow of text in your documents, be careful to avoid overlapping text unintentionally.

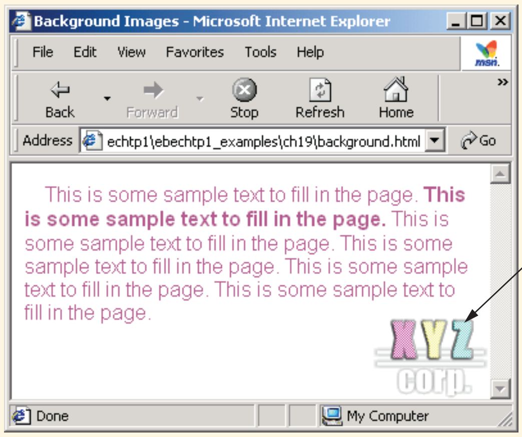
19.7 Backgrounds

CSS also gives you more control over backgrounds than simple HTML attributes. We have used the **`background-color`** property in previous examples. You can also add background images to your documents using CSS. In Fig. 19.8, we add a corporate watermark to the bottom-right corner of the document—this watermark stays fixed in the corner, even when the user scrolls up or down the screen.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig. 19.8: background.html               -->
<!-- Adding background images and indentation -->

<HEAD>
<TITLE>Background Images</TITLE>

<STYLE TYPE = "text/css">

   BODY  { background-image: url(watermark.gif);
           background-position: bottom right;
           background-repeat: no-repeat;
           background-attachment: fixed }

   P     { font-size: 2em;
           color: #AA5588;
           text-indent: 1em;
           font-family: Arial, sans-serif }

   .dark { font-weight: bold }

</STYLE>
</HEAD>

<BODY>

<P>
This is some sample text to fill in the page.
```

Fig. 19.8 Adding a background image with CSS (part 1 of 2).

```
<SPAN CLASS = "dark">This is some sample
text to fill in the page.</SPAN>
This is some sample text to fill in the page.
This is some sample text to fill in the page.
This is some sample text to fill in the page.
This is some sample text to fill in the page.
</P>

</BODY>
</HTML>
```


Background Images - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home
Address echtp1\ebechtp1_examples\ch19\background.html Go
Done My Computer
Background image

This is some sample text to fill in the page. **This is some sample text to fill in the page.** This is some sample text to fill in the page. This is some sample text to fill in the page. This is some sample text to fill in the page. This is some sample text to fill in the page.

Fig. 19.8 Adding a background image with CSS (part 2 of 2).

The code that adds the background image in the bottom-right corner of the window is on lines 12 through 15:

```
BODY  { background-image: url(watermark.gif);
        background-position: bottom right;
        background-repeat: no-repeat;
        background-attachment: fixed }
```

The ***background-image property*** specifies the URL of the image to use, in the format **url(fileLocation)**. You can also specify ***background-color*** to use in case the image is not found.

The ***background-position property*** positions the image on the page. You can use the keywords **top**, **bottom**, **center**, **left** and **right** individually or in combination for vertical and horizontal positioning. You can also position using lengths, specifying the horizontal length followed by the vertical length. For example, to position the image centered vertically (positioned at 50% of the distance across the screen) and 30 pixels from the top, you would use

```
background-position: 50% 30px;
```

The **background-repeat** property controls the *tiling* of the background image (tiling was discussed in Section 9.8). Here we set the tiling to **no-repeat** so that only one copy of the background image is placed on screen. The **background-repeat** property can be set to ***repeat*** (the default) to tile the image vertically and horizontally, to ***repeat-x*** to tile the image only horizontally or ***repeat-y*** to tile the image only vertically.

The final property setting, ***background-attachment: fixed***, 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 browser window down.

On line 19, we introduce a new text-formatting property:

```
text-indent: 1em;
```

This indents the first line of text in the element by the specified amount. You might use this to make your Web page read more like a novel, in which the first line of every paragraph is indented.

Another new property is introduced on line 22

```
.dark { font-weight: bold }
```

The ***font-weight property*** specifies the "boldness" of affected text. Values besides ***bold*** and ***normal*** (the default) are ***bolder*** (bolder than **bold** text) and ***lighter*** (lighter than **normal** text). You can also specify the value using multiples of 100 from 100 to 900 (i.e., **100**, **200**, ..., **900**). Text specified as **normal** is equivalent to **400** and **bold** text is equivalent to **700**. Most systems do not have fonts that can be scaled this finely so using the **100**...**900** values might not display the desired effect.

Another CSS property you can use to format text is the ***font-style property***, which allows you to set text to ***none***, ***italic*** or ***oblique*** (**oblique** will default to **italic** if the system does not have a separate font file for oblique text, which is normally the case).

We introduce the ***SPAN element*** in lines 31 and 32:

```
<SPAN CLASS = "dark">This is some sample
text to fill in the page.</SPAN>
```

SPAN is a generic grouping element—it does not apply any inherent formatting to its contents. Its main use is to apply styles or ***ID attributes*** to a block of text. It is displayed inline (a so-called *inline-level element*) with other text, with no line breaks. A similar element is the ***DIV element***, which also applies no inherent styles, but is displayed on its own line, with margins above and below (a so-called *block-level element*).

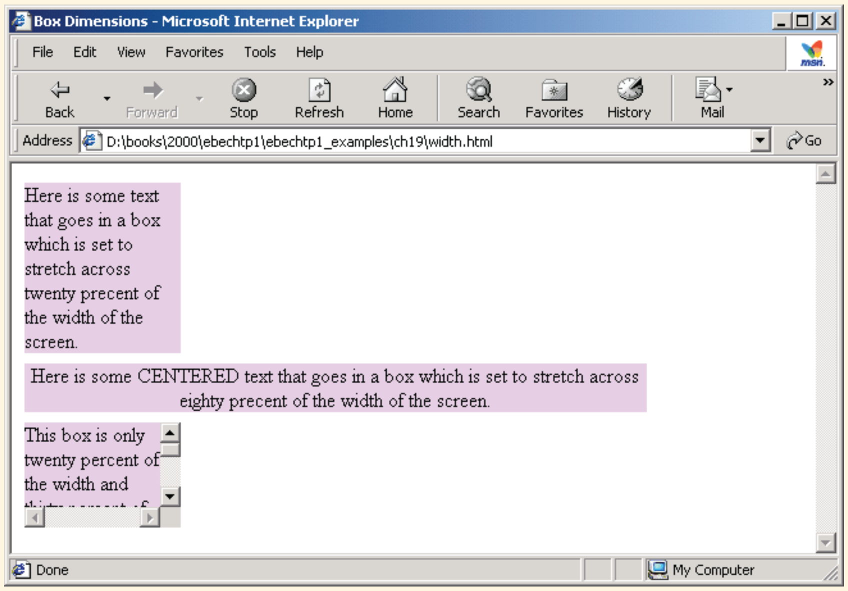
19.8 Element Dimensions

The dimensions of each element on the page can be set using CSS (Fig. 19.9).

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2  <HTML>
3
4  <!-- Fig. 19.9: width.html                      -->
5  <!-- Setting box dimensions and aligning text -->
6
```

Fig. 19.9 Setting box dimensions and aligning text (part 1 of 2).

```
<HEAD>
<TITLE>Box Dimensions</TITLE>
<STYLE TYPE = "text/css">

   DIV { background-color: #FFCCFF;
         margin-bottom: .5em }

</STYLE>
</HEAD>

<BODY>

<DIV STYLE = "width: 20%">Here is some
text that goes in a box which is
set to stretch across twenty precent
of the width of the screen.</DIV>

<DIV STYLE = "width: 80%; text-align: center">
Here is some CENTERED text that goes in a box
which is set to stretch across eighty precent of
the width of the screen.</DIV>

<DIV STYLE = "width: 20%; height: 30%; overflow: scroll">
This box is only twenty percent of
the width and thirty percent of the height.
What do we do if it overflows? Set the
overflow property to scroll!</DIV>

</BODY>
</HTML>
```


Box Dimensions - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\width.html
Go
Done
My Computer

Here is some text that goes in a box which is set to stretch across twenty precent of the width of the screen.

Here is some CENTERED text that goes in a box which is set to stretch across eighty precent of the width of the screen.

This box is only twenty percent of the width and

Fig. 19.9 Setting box dimensions and aligning text (part 2 of 2).

The inline style of line 19

```
<DIV STYLE = "width: 20%">Here is some
```

shows how to set the ***width*** of an element on screen; here we indicate that this **DIV** element should occupy 20% of the screen width (which 20% of the screen depends on how the element is aligned, most elements are left-aligned by default). The height of an element can be set similarly, using the **height** property. Relative lengths and absolute lengths may also be used to specify **height** and **width**. For example, you could set the width of an element using

```
width: 10em
```

to have the element's width be equal to 10 times the size of the font.

Line 24

```
<DIV STYLE = "width: 80%; text-align: center">
```

shows that text within an element can be ***center***ed—other values for the ***text-align*** property are ***left*** and ***right***.

One problem with setting both element dimensions is that content inside might sometimes exceed the set boundaries, in which case the element is simply made large enough for all the content to fit. However, as we see on line 29

```
<DIV STYLE = "width: 20%; height: 30%; overflow: scroll">
```

we can set the ***overflow property*** to ***scroll***; this adds scrollbars if the text overflows the boundaries.

19.9 Text Flow and the Box Model

A browser normally places text and elements on screen in the order they are in the HTML file. However, as we saw with absolute positioning, it is possible to remove elements from the normal flow of text. *Floating* allows you to move an element to one side of the screen—other content in the document will then flow around the floated element. In addition, each block-level element has a box drawn around it, known as the *box model*—the properties of this box are easily adjusted (Fig. 19.10).

Line 21

```
<DIV STYLE = "text-align: center">Centered Text</DIV>
```

shows that text inside an element can be aligned by setting the **text-align** property, whose possible values are **left**, **center**, **right** and **justify**.

In addition to text, whole elements can be *floated* to the left or right of a document. This means that any nearby text will wrap around the floated element. For example, in lines 24 and 25

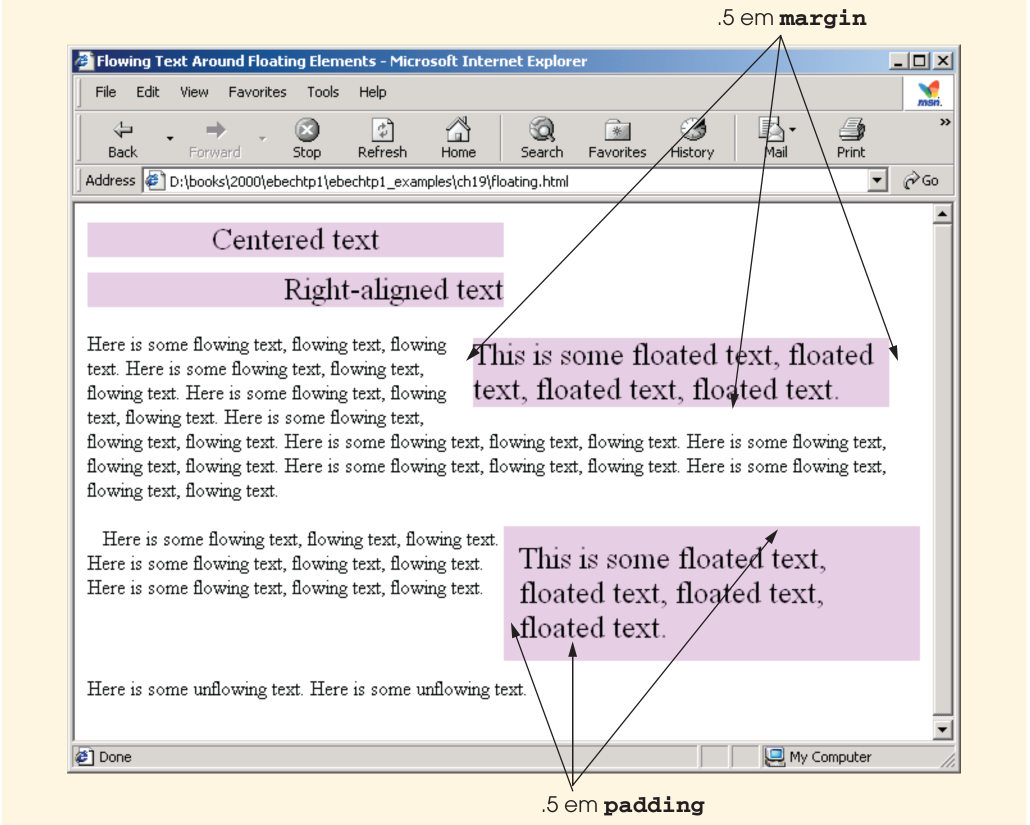
```
<DIV STYLE = "float: right; margin: .5em">This is some floated
text, floated text, floated text, floated text.</DIV>
```

we float a **DIV** element to the **right** side of the screen. As you can see, the text from lines 27 through 34 flows cleanly to the left and underneath this **DIV** element.

The second property we set in line 24, ***`margin`***, determines the distance between the edge of the element and any text outside the element. When elements are rendered on the screen using the box model, the content of each element is surrounded by *padding*, a *border* and *margins* (Fig. 19.11).

```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2  <HTML>
3
4  <!-- Fig. 19.10: floating.html           -->
5  <!-- Floating elements and element boxes -->
6
7  <HEAD>
8  <TITLE>Flowing Text Around Floating Elements</TITLE>
9  <STYLE TYPE = "text/css">
10
11    DIV { background-color: #FFCCFF;
12          margin-bottom: .5em;
13          font-size: 1.5em;
14          width: 50% }
15
16 </STYLE>
17 </HEAD>
18
19 <BODY>
20
21 <DIV STYLE = "text-align: center">Centered text</DIV>
22 <DIV STYLE = "text-align: right">Right-aligned text</DIV>
23
24 <DIV STYLE = "float: right; margin: .5em">This is some floated
25 text, floated text, floated text, floated text.</DIV>
26 <P>
27 Here is some flowing text, flowing text, flowing text.
28 Here is some flowing text, flowing text, flowing text.
29 Here is some flowing text, flowing text, flowing text.
30 Here is some flowing text, flowing text, flowing text.
31 Here is some flowing text, flowing text, flowing text.
32 Here is some flowing text, flowing text, flowing text.
33 Here is some flowing text, flowing text, flowing text.
34 Here is some flowing text, flowing text, flowing text.
35 </P>
36
37 <P><DIV STYLE ="float: right; padding: .5em">This is some floated
38 text, floated text, floated text, floated text.</DIV>
39 Here is some flowing text, flowing text, flowing text.
40 Here is some flowing text, flowing text, flowing text.
41 Here is some flowing text, flowing text, flowing text.
42 <SPAN STYLE = "clear: right">Here is some unflowing text.
43 Here is some unflowing text.</SPAN>
44 </P>
45
46 </BODY>
47 </HTML>
```

Fig. 19.10 Floating elements, aligning text and setting box dimensions (part 1 of 2).


.5 em **margin**
Flowing Text Around Floating Elements - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\floating.html
Go
Done
My Computer
.5 em **padding**

Centered text

Right-aligned text

Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text.

This is some floated text, floated text, floated text, floated text.

Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text. Here is some flowing text, flowing text, flowing text.

This is some floated text, floated text, floated text, floated text.

Here is some unflowing text. Here is some unflowing text.

Fig. 19.10 Floating elements, aligning text and setting box dimensions (part 2 of 2).

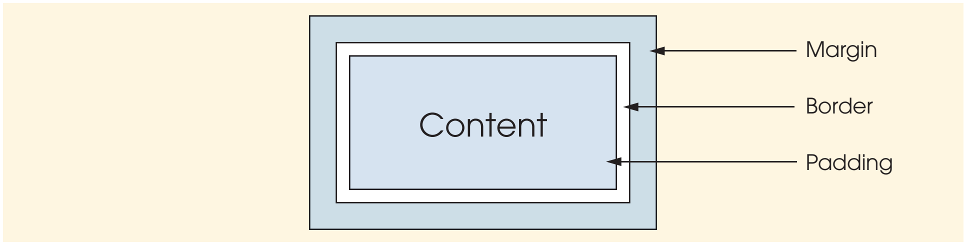


Fig. 19.11 Box model for block-level elements.

Margins for individual sides of an element can be specified by using ***margin-top***, ***margin-right***, ***margin-left***, and ***margin-bottom***.

A related property, ***padding***, is set for the **DIV** element in line 37:

```
<DIV STYLE = "float: right; padding: .5em">This is some floated
```

The *padding* is the distance between the content inside an element and the edge of the element. Like the margin, the padding can be set for each side of the box with **padding-top**, **padding-right**, **padding-left**, and **padding-bottom**.

Line 42

```
<SPAN STYLE = "clear: right">Here is some unflowing text.
```

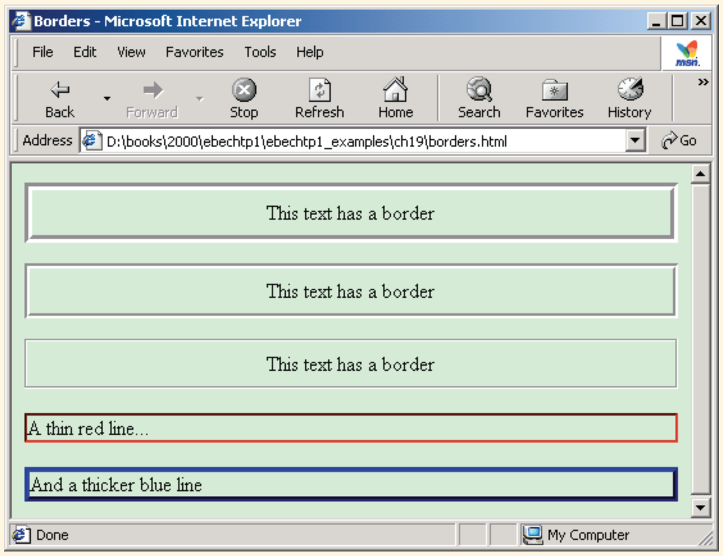
shows that you can interrupt the flow of text around a **`float`**ed element by setting the ***`clear`*** *property* to the same direction the element is **`float`**ed—**`right`** or **`left`**. Setting the **`clear`** property to ***`all`*** interrupts the flow on both sides of the document. Note that the box model only applies to block-level elements such as **`DIV`**, **`P`** and **`H1`**—the box model does not apply to inline-level elements such as **`EM`**, **`STRONG`** and **`SPAN`**.

Another property included around every block-level element on screen is the border. The border lies between the padding space and the margin space, and has numerous properties to adjust its appearance (Fig. 19.12).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig. 19.12: borders.html      -->
<!-- Setting borders of an element -->

<HEAD>
<TITLE>Borders</TITLE>
<STYLE TYPE = "text/css">

   BODY    { background-color: #CCFFCC }

   DIV     { text-align: center;
             margin-bottom: 1em;
             padding: .5em }

   .thick  { border-width: thick }

   .medium { border-width: medium }

   .thin   { border-width: thin }

   .groove { border-style: groove }

   .inset  { border-style: inset }

   .outset { border-style: outset }

   .red    { border-color: red }

   .blue   { border-color: blue }

</STYLE>
</HEAD>

<BODY>

<DIV CLASS = "thick groove">This text has a border</DIV>
```

Fig. 19.12 Applying borders to elements (part 1 of 2).

```
39  <DIV CLASS = "medium groove">This text has a border</DIV>
40  <DIV CLASS = "thin groove">This text has a border</DIV>
41
42  <P CLASS = "thin red inset">A thin red line...</P>
43  <P CLASS = "medium blue outset">And a thicker blue line</P>
44
45  </BODY>
46  </HTML>
```


Borders - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\borders.html Go
Done
My Computer

This text has a border

This text has a border

This text has a border

A thin red line...

And a thicker blue line

Fig. 19.12 Applying borders to elements (part 2 of 2).

In this example, we set three properties: the ***border-width***, ***border-style*** and ***border-color***. The **border-width** property may be set to any of the CSS lengths, or the predefined values of ***thin***, ***medium*** or ***thick***. The **border-color** sets the color used for the border (this has different meanings for different borders).

As with padding and margins, each of the border properties may be set for individual sides of the box (e.g., ***border-top-style*** or ***border-left-color***).

Also, as shown on line 38,

```
<DIV CLASS = "thick groove">This text has a border</DIV>
```

it is possible to assign more than one class to an HTML element using the **CLASS** attribute.

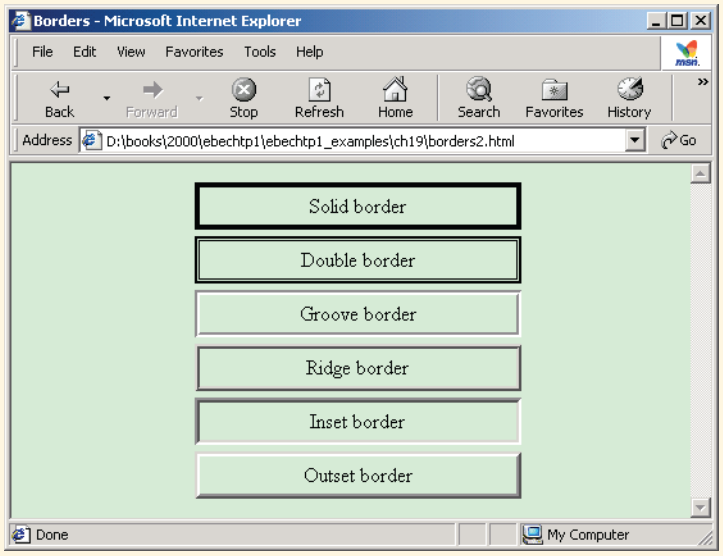
The **border-style**s are ***none***, ***hidden***, ***dotted***, ***dashed***, ***solid***, ***double***, ***groove***, ***ridge***, ***inset*** and ***outset***. Figure 19.13 illustrates these border styles.

Portability Tip 19.3

Keep in mind that the ***dotted*** *and* ***dashed*** *styles are available only for Macintosh systems.*

As you can see, the **groove** and **ridge** **border-style**s have opposite effects, as do **inset** and **outset**.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig. 19.13: borders2.html  -->
<!-- Various border-styles      -->

<HEAD>
<TITLE>Borders</TITLE>

<STYLE TYPE = "text/css">

   BODY    { background-color: #CCFFCC }

   DIV     { text-align: center;
             margin-bottom: .3em;
             width: 50%;
             position: relative;
             left: 25%;
             padding: .3em }
</STYLE>
</HEAD>

<BODY>

<DIV STYLE = "border-style: solid">Solid border</DIV>
<DIV STYLE = "border-style: double">Double border</DIV>
<DIV STYLE = "border-style: groove">Groove border</DIV>
<DIV STYLE = "border-style: ridge">Ridge border</DIV>
<DIV STYLE = "border-style: inset">Inset border</DIV>
<DIV STYLE = "border-style: outset">Outset border</DIV>
</BODY>
</HTML>
```


Borders - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\borders2.html
Go
Done
My Computer

Solid border

Double border

Groove border

Ridge border

Inset border

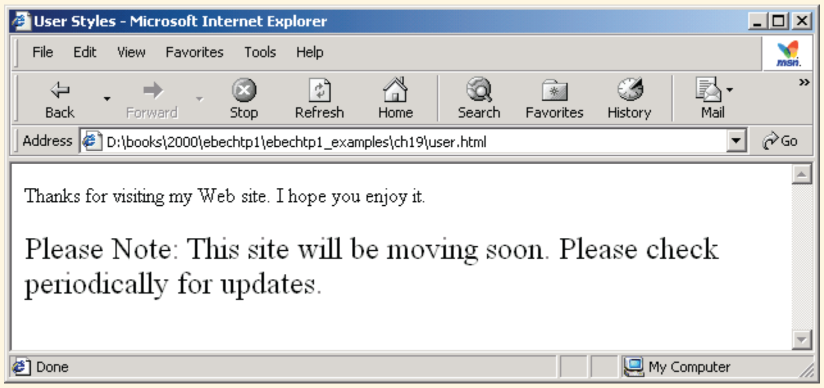
Outset border

Fig. 19.13 Various **`border-style`**s.

19.10 User Style Sheets

An important issue to keep in mind when adding style sheets to your site is what kind of users will be viewing your site. Users have the option to define their own *user style sheets* to format pages based on their own preferences—for example, visually impaired people might want to increase the text size on all pages they view. As a Web-page author, if you are not careful, you might inadvertantly override user preferences with the styles defined on your Web pages. This section explores possible conflicts between *user styles* and *author styles*. Figure 19.14 is a simple example of a Web page using the **em** measurement for the **font-size** property to increase text size on the page.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>

<!-- Fig. 19.14: user.html  -->
<!-- User styles            -->

<HEAD>
<TITLE>User Styles</TITLE>

<STYLE TYPE = "text/css">

   .note { font-size: 1.5em }

</STYLE>
</HEAD>

<BODY>

<P>Thanks for visiting my Web site. I hope you enjoy it.</P>
<P CLASS = "note">Please Note: This site will be moving soon.
Please check periodically for updates.</P>

</BODY>
</HTML>
```


User Styles - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\user.html
Go
Done
My Computer

Thanks for visiting my Web site. I hope you enjoy it.

Please Note: This site will be moving soon. Please check periodically for updates.

Fig. 19.14 Modifying text size with the **em** measurement.

In line 12

```
        .note { font-size: 1.5em }
```

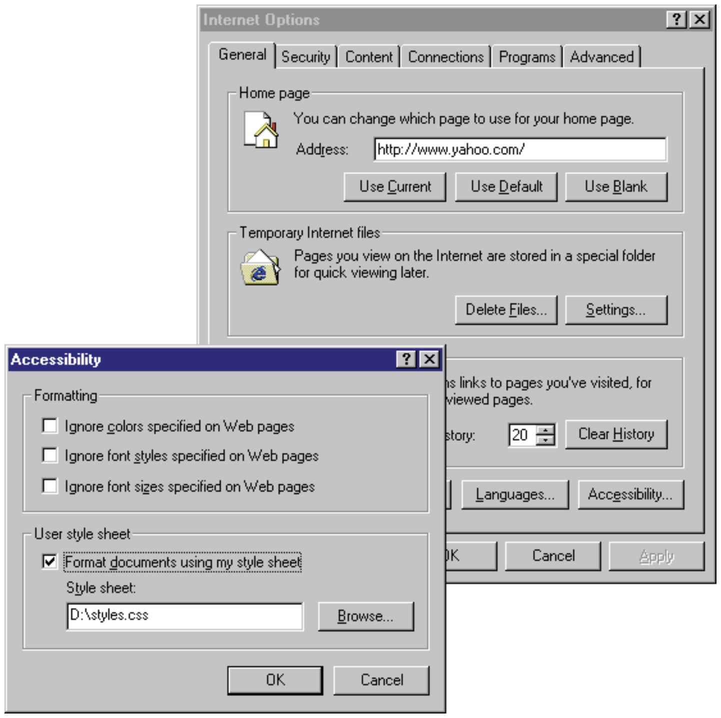
we multiply by 1.5 the font size of all elements with **`CLASS = "note"`** (see lines 20 and 21). Assuming the default browser font size of 12 points, this same text size increase could also have been accomplished by specifying

```
    .note { font-size:  18pt }
```

However, what if the user had defined their own **`font-size`** in a user style sheet? Because the CSS specification gives precedence to author styles over user styles, this conflict would be resolved with the author style overriding the user style. This can be avoided by using relative measurements (such as **`em`** or **`ex`**) instead of absolute measurements (such as **`pt`**).

Adding a user style sheet (Fig. 19.15) in Internet Explorer 5 is done by selecting **Internet Options...** located in the **Tools** menu. In the dialog box that appeares, click on **Accessibility...**, check the **Format documents using my style sheet** check box and type in the location of your user style sheet. Note that you also have the option of overriding colors, font styles, and font sizes specified on Web pages with your own user styles.

User style sheets are created in the same format as the linked external style sheet shown in Fig. 19.4. A sample user style sheet is shown in Fig. 19.16.



You can change which page to use for your home page.
Address: http://www.yahoo.com/

Pages you view on the Internet are stored in a special folder for quick viewing later.

...s links to pages you've visited, for ...viewed pages.
...story: 20

☐ Ignore colors specified on Web pages
☐ Ignore font styles specified on Web pages
☐ Ignore font sizes specified on Web pages
☑ Format documents using my style sheet
Style sheet: D:\styles.css

Fig. 19.15 Adding a user style sheet in Internet Explorer 5.

The Web page shown in Fig. 19.14 is re-rendered in Figure 19.17, this time with the user style sheet from 19.16 applied.

Because the code for this page uses a relative **`font-size`** measurement of **`1.5em`**, it multiplies the original size of the affected text (**`20pt`**) by **`1.5`** times, giving it an effective size of **`30pt`**.

19.11 Internet and World Wide Web Resources

`www.w3.org/TR/REC-CSS2/`
The W3C *Cascading Style Sheets, Level 2* specification contains a list of all the CSS properties. The specification is also filled with helpful examples detailing the use of many of the properties.

`style.webreview.com`
This site has several charts of CSS properties, including a listing of which browsers support which attributtes, and to what extent.

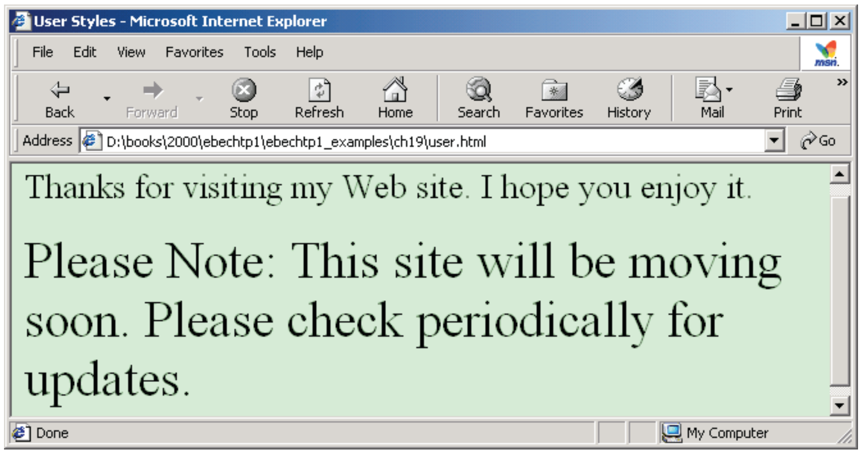
`www.w3.org/TR/REC-CSS1-961217.html`
This site contains the W3C *Cascading Style Sheets, Level 1* specification.

SUMMARY

- The inline style allows you to declare a style for an individual element using the **`STYLE`** attribute in that element's opening HTML tag.
- Each CSS property is followed by a colon, then the value of that attribute.
- The **`color`** property sets the color of text. Color names and hex codes may be used as the value.
- Styles that are placed in the **`<STYLE>`** section apply to the whole document.

```
BODY      { font-size: 20pt;
            background-color: #CCFFCC }
A         { color: red }
```

Fig. 19.16 A sample user style sheet.


User Styles - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address D:\books\2000\ebechtp1\ebechtp1_examples\ch19\user.html
Go
Done
My Computer

Thanks for visiting my Web site. I hope you enjoy it.

Please Note: This site will be moving soon. Please check periodically for updates.

Fig. 19.17 A Web page with user styles enabled.

- The **TYPE** attribute of the **STYLE** element specifies the MIME type (the specific format of binary encoding) of the following style sheet. Regular text style sheets always use **text/css**.
- Each rule body begins and ends with a curly brace (**{** and **}**).
- Style class declarations are preceded with a period and are applied to elements of that specific class.
- The CSS rules in a style sheet use the same format as inline styles—the property is followed by a colon (**:**) and the value of that property. Multiple properties are separated with a semicolon (**;**).
- The **background-color** attribute specifies the background color of the element.
- The **font-family** attribute specifies the name of the font that should be displayed. Generic font families allow you to specify a type of font instead of a specific font for greater display flexibility. The **font-size** property specifies the size to use to render the font.
- The **CLASS** attribute applies a style class to an element.
- All styles applied to a parent element also apply to child elements inside that element.
- Pseudo-classes give the author access to content not specifically declared in the document. The **hover** pseudo-class is activated when the user moves the mouse cursor over an **A** element.
- The **text-decoration** property applies decorations to text within an element, such as **underline**, **overline**, **line-through** and **blink**
- To apply rules to multiple elements separate the elements with commas in the stylesheet.
- A pixel is a relative-length measurement—it varies in size based on screen resolution. Other relative lengths are **em** (font size), **ex** ("x-height" of the font—the height of a lowercase x) and percentages.
- The other units of measurement available in CSS are absolute-length measurements, i.e., units that do not vary in size based on the system. These are **in** (inches), **cm** (centimeters), **mm** (millimeters), **pt** (points—1 **pt**=1/72 **in**) and **pc** (picas—1 **pc** = 12 **pt**).
- External linking can help give a Web site a uniform look—separate pages on a site can all use the same styles. Modifying a single file can then make changes to styles across an entire Web site.
- The **LINK** element's **REL** attribute specifies a relationship between the current document and another document.
- The CSS **position** property allows absolute positioning, which gives us greater control over how documents are displayed. 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** margins of its parent element.
- The **z-index** property allows you to properly layer overlapping elements. Elements that have higher **z-index** values are displayed in front of elements with lower **z-index** values.
- Unlike absolute positioning, relative positioning keeps elements in the general flow of elements on the page, and offsets them by the specified **top**, **left**, **right** or **bottom** values.
- Property **background-image** specifies the URL of the image to use, in the format **url(***fileLocation***)**. Specify the **background-color** to use if the image is not found. The property **background-position** positions 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 using lengths.
- The **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. The **background-repeat** property can be set to **repeat** (the default) to tile the image vertically and horizontally, to **repeat-x** to tile the image only horizontally or **repeat-y** to tile the image only vertically.
- The property setting **background-attachment: fixed** 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.

- The **`text-indent`** property indents the first line of text in the element by the specified amount.
- The **`font-weight`** property specifies the "boldness" of text. Values besides **`bold`** and **`normal`** (the default) are **`bolder`** (bolder than **`bold`** text) and **`lighter`** (lighter than **`normal`** text). You can also specify the value using multiples of 100 from 100 to 900 (i.e., **`100`**, **`200`**, ..., **`900`**). Text specified as **`normal`** is equivalent to **`400`** and **`bold`** text is equivalent to **`700`**.
- The **`font-style`** property allows you to set text to **`none`**, **`italic`** or **`oblique`** (**`oblique`** will default to **`italic`** if the system does not have a separate font file for oblique text, which is normally the case).
- **`SPAN`** is a generic grouping element—it does not apply any inherent formatting to its contents. Its main use is to apply styles or **`ID`** attributes to a block of text. It is displayed inline (a so-called inline element) with other text, with no line breaks. A similar element is the **`DIV`** element, which also applies no inherent styles, but is displayed on a separate line, with margins above and below (a so-called block-level element).
- The dimensions of page elements can be set using CSS using the **`height`** and **`width`** properties.
- Text within an element can be **`center`**ed using **`text-align`**—other values for the **`text-align`** property are **`left`** and **`right`**.
- One problem with setting both element dimensions is that content inside might sometimes exceed the set boundaries, in which case the element is simply made large enough for all the content to fit. However, you can set the **`overflow`** property to **`scroll`**; this adds scroll bars if the text overflows the boundaries we have set for it.
- Browsers normally place text and elements on screen in the order they appear in the HTML file. Elements can be removed from the normal flow of text. Floating allows you to move an element to one side of the screen—other content in the document will then flow around the floated element.
- Each block-level element has a box drawn around it, known as the box model—the properties of this box are easily adjusted.
- The **`margin`** property determines the distance between the element's edge and any outside text.
- CSS uses a box model to render elements on screen—the content of each element is surrounded by padding, a border and margins
- Margins for individual sides of an element can be specified by using **`margin-top`**, **`margin-right`**, **`margin-left`** and **`margin-bottom`**.
- The padding, as opposed to the margin, is the distance between the content inside an element and the edge of the element. Padding can be set for each side of the box with **`padding-top`**, **`padding-right`**, **`padding-left`** and **`padding-bottom`**.
- You can interrupt the flow of text around a **`float`**ed element by setting the **`clear`** property to the same direction the element is **`float`**ed—**`right`** or **`left`**. Setting the **`clear`** property to **`all`** interrupts the flow on both sides of the document.
- A property of every block-level element on screen is its border. The border lies between the padding space and the margin space and has numerous properties to adjust its appearance.
- The **`border-width`** property may be set to any of the CSS lengths, or the predefined values of **`thin`**, **`medium`** or **`thick`**.
- The **`border-style`**s available are **`none`**, **`hidden`**, **`dotted`**, **`dashed`**, **`solid`**, **`double`**, **`groove`**, **`ridge`**, **`inset`** and **`outset`**. Keep in mind that the **`dotted`** and **`dashed`** styles are available only for Macintosh systems.
- The **`border-color`** property sets the color used for the border.
- It is possible to assign more than one class to an HTML element using the **`CLASS`** attribute.

TERMINOLOGY

<LINK> element
absolute positioning
absolute-length measurement
Arial font
background
background-attachment
background-color
background-image
background-position
background-repeat
blink
block-level element
border
border-color
border-style
border-width
box model
Cascading Style Sheet (CSS) specification
child element
CLASS attribute of an element
clear: all
clear: left
clear: right
cm (centimeters)
colon (**:**) in a CSS rule
color
CSS rule
cursive generic font family
dashed border style
dotted border style
double border style
em (size of font)
embedded style sheet
ex (x-height of font)
float property
font-style property
generic font family
groove border style
hidden border style
hover pseudo-class
HREF attribute of **<LINK>** element
importing a style sheet
in (inches)
inline styles
inline-level element
inset border style
large font size
larger font size
left
line-through text decoration
linking to an external style sheet
margin
margin-bottom property
margin-left property
margin-right property
margin-top property
medium border width
medium font size
mm (millimeters)
monospace generic font family
none border style
outset border style
overflow property
overline text decoration
padding
parent element
pc (picas)
position: absolute
position: relative
pseudo-class
pt (points)
REL attribute of **<LINK>** element
relative positioning
relative-length measurement
repeat-x
repeat-y
ridge border style
right
rule in CSS
sans-serif generic font family
scroll
separation of structure from content
serif generic font family
small font size
smaller font size
solid border style
style
STYLE attribute
style class
style in header of document
style sheet (CSS rules separate text file)
text flow
text/css MIME type
text-align
text-decoration
text-indent
thick border width
thin border width

user style sheet	**xx-large** font size
x-large font size	**xx-small** font size
x-small font size	**z-index**

SELF-REVIEW EXERCISES

19.1 Assume that the size of the base font on a system is 12 points.

a) How big is 36 point font in ems?
b) How big is 8 point font in ems?
c) How big is 24 point font in picas?
d) How big is 12 point font in inches?
e) How big is 1 inch font in picas?

19.2 Fill in the blanks in the following questions:

a) Using the __________ element allows you to use external style sheets in your pages.
b) To apply a CSS rule to more than one element at a time, separate the element names with a __________.
c) Pixels are a __________ length measurement unit.
d) The **hover** __________-__________ is activated when the user moves the mouse cursor over the specified element.
e) Setting the **overflow** property to __________ provides a mechanism for containing inner content without compromising specified box dimensions.
f) While __________ is a generic inline element that applies no inherent formatting, the __________ is a generic block-level element that applies no inherent formatting.
g) Setting the **background-repeat** property to __________ will tile the specified **background-image** only vertically.
h) If you **float** an element, you can stop the flowing text by using the __________ property.
i) The __________ property allows you to indent the first line of text in an element.
j) Three components of the box model are the __________, __________ and __________.

ANSWERS TO SELF-REVIEW EXERCISES

19.1 a) 3 ems. b) .75 ems. c) 2 picas. d) 1/6 inch. e) 6 picas.

19.2 a) **LINK**. b) comma. c) relative. d) pseudo-element. e) **scroll**. f) **SPAN**, **DIV**. g) **y-repeat**. h) **clear**. i) **text-indent**. j) content, padding, border or margin.

EXERCISES

19.3 Write a CSS rule that makes all text 1.5 times larger than the base font of the system and colors it red.

19.4 Write a CSS rule that removes the underline from all links inside list items (**LI**) and shifts them left by 3 **em**s.

19.5 Write a CSS rule that places a background image halfway down the page, tiling horizontally. The image should remain in place when the user scrolls up or down.

19.6 Create a CSS rule that changes the effect of an **EM** element from italic text to underlined text.

19.7 Write a CSS rule that gives all **H1** and **H2** elements a padding of .5 **em**s, a **groove**d border style and a margin of .5 **em**s.

19.8 Write a CSS rule that changes the color of all elements with attribute **CLASS="greenMove"** to green and shifts them down 25 pixels and right 15 pixels.

19.9 Write a CSS rule that centers an element horizontally in 60% of the browser window's width.