CSS

ITCS 210 Web Programming

Class Objectives

- To control the appearance of a website by creating style sheets.
- To know how to apply CSS including inline styles, internal style sheets and external style sheets.
- To specify the precise font, size, color and other properties of displayed text.
- To specify element backgrounds and colors.
- To understand how to control margins, borders and padding.
- To know how to apply CSS3 and Bootstrap

Introduction

- Cascading Style Sheets (CSS) allow the author of a web page to **specify the presentation** of elements separately from the structure of the document

- CSS types
  - Inline Style Sheet
  - Internal Style Sheet
  - External Style Sheet

Inline Styles

- The inline style allows you to declare a style for an individual element by using the **`style`** attribute in the element's start tag
- Each CSS property is followed by a colon and the value of the attribute, while multiple property declarations are separated by a semicolon

FORMAT property:value; property:value

HTML Style

- Every HTML element has a **default style** (background color is white and text color is black).
- The HTML style attribute has the following **syntax**:

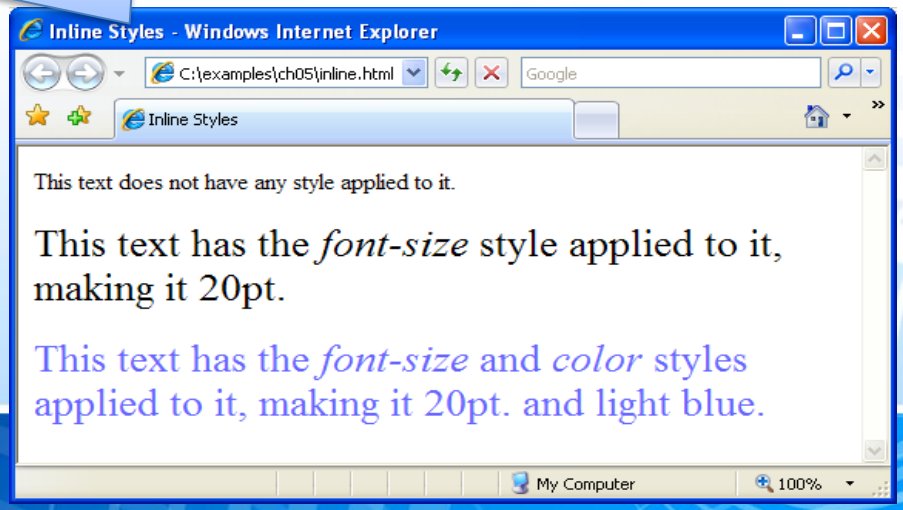
style="property:value;property:value"

- The ***property*** is a CSS property. The ***value*** is a CSS value.

EXAMPLE

```
<p>This text does not have any style applied to it.</p>

<p style = "font-size: 20pt">This text has the
<em>font-size</em> style applied to it, making it 20pt.
</p>

<p style = "font-size: 20pt; color: #6666ff">
  This text has the <em>font-size</em> and
  <em>color</em> styles applied to it, making it
  20pt. and light blue.</p>
```

Inline styles do not truly separate presentation from content.



HTML block

- Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is **block** or **inline**.

- **A block-level element** always *starts on a new line* *and takes up the full width available* *(stretches out to the left and right as far as it can).*

  E,g., , <h1> - <h6>, <p>, <form>

- **An inline element** does not start on a new line and only takes up as much width as necessary.

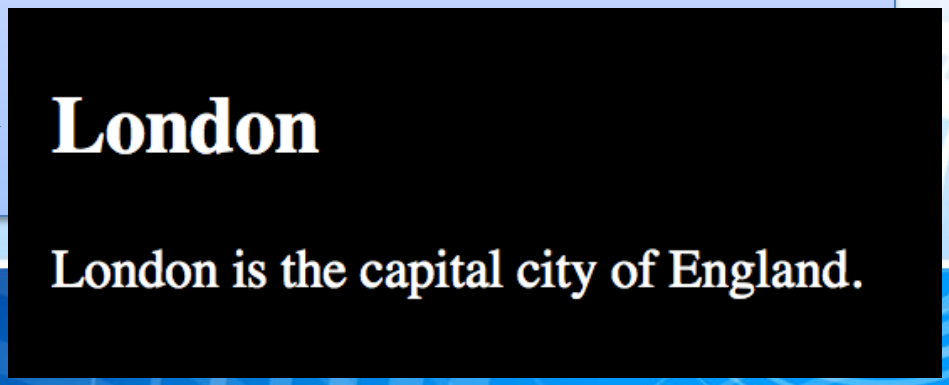
  E.g.,, <a>, <img>

HTML block ( )

The element is a block-level element that is often used as a container for other HTML elements.

```
<!DOCTYPE html>
<html>
<body>

<div style="background-color:black; color:white; padding:20px;”>
<h2>London</h2>
<p>London is the capital city of England.</p>
</div>

</body>
</html>
```

output



HTML block ()

- The element is an inline element that is often used as a container for some text.
- The element has no required attributes, but **style** and **class** are common.
- When used together with CSS, the element can be used to style parts of the text:

```
<!DOCTYPE html>
<html>
<body>
<h1>My <span style="color:red">Important</span> Heading</h1>
</body>
</html>
```

output

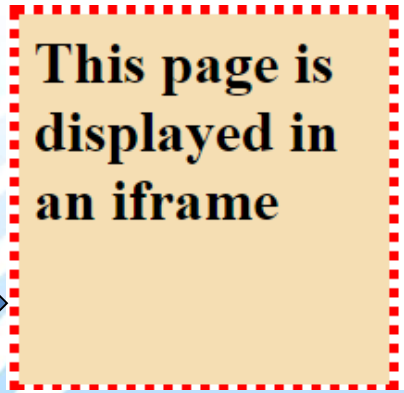
HTML Iframes

- An iframe is used to display a web page within a web page.
- The syntax for adding an iframe is:

<iframe src="*URL*"></iframe>

```
<!DOCTYPE html>
<html>
<body>

<iframe src="demo_iframe.htm" width="200" height="200"
style="border:5px dotted red"></iframe>

</body>
</html>
```

output



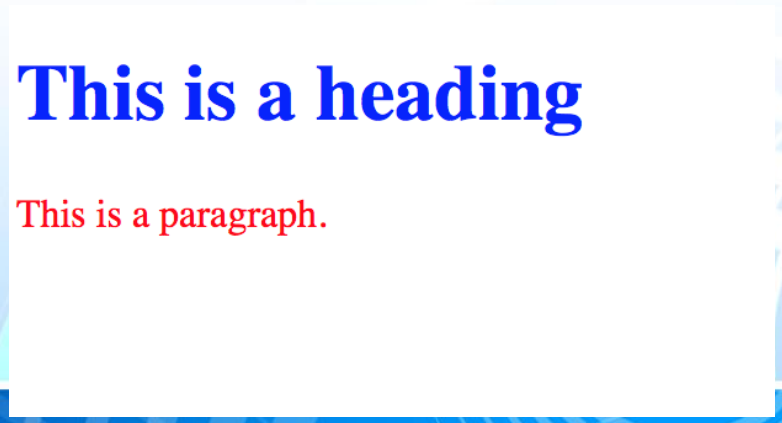
HTML Style (Example)

```
<!DOCTYPE html>
<html>
<body style="background-color:lightgrey">

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

output


```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue">This is a heading</h1>
<p style="color:red">This is a paragraph.</p>

</body>
</html>
```

output



HTML Style (Example)

```
<!DOCTYPE html>
<html>
<body>

<h1 style="font-family:verdana">This is a heading</h1>
<p style="font-family:courier">This is a paragraph.</p>

</body>
</html>
```

output

This is a heading

This is a paragraph.

```
<!DOCTYPE html>
<html>
<body>

<h1 style="font-size:300%">This is a heading</h1>
<p style="font-size:160%">This is a paragraph.</p>

</body>
</html>
```

output

This is a heading

This is a paragraph.

HTML Style (Example)

```
<!DOCTYPE html>
<html>
<body>

<h1 style="text-align:center">Centered heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

output

Is it good?

```
<font color="#9933ff">this font is purple</font>
```

```
<table>
   <tr>
      <td valign="middle">
         I am in the middle
      <td>
   </tr>
</table>
```

```
<table border="1"
   cellpadding="5">
   <tr>
      <td>
         ...
      <td>
   </tr>
</table>
```

```
<strong>I am bold !!</strong>
```

```
<i>This is italic</i>
```

So what is a good practice?

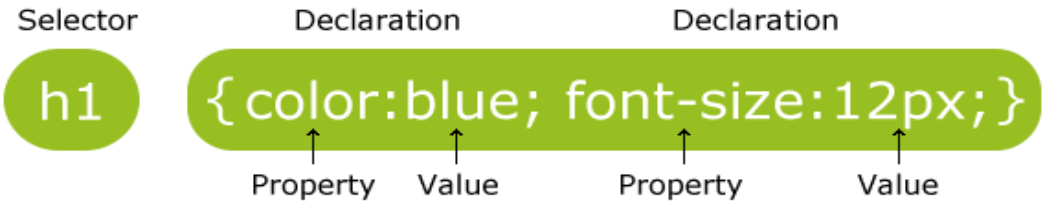
- Web design = Layout + Content + Style

- HTML
  - **Page Layout** and Structure
  - Define the content of a document
    - This is a paragraph.
    - <h1>This is a big header.</h1>
- CSS
  - Styling and Formatting
  - Positioning

Internal Style Sheets

- Or Embedded Style Sheets
- Styles that are placed in a `style` element use **selectors** to apply style elements throughout the entire document
- Each rule body in a style sheet begins and ends with a curly brace ({ and }).
- 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 by semicolons (;)

CSS Syntax




```
p
{
    color:red;
    text-align:center;
}
```

- **Selector** - the HTML element
- Declaration consists of a property and a value.
- Property is the style attribute you want to change. Each property has a value.

```
8     <head>
9        <title>Style Sheets</title>
10
11       <!-- this begins the style sheet section -->
12       <style type = "text/css">
13          em        { font-weight: bold;
14                      color: black }
15          h1        { font-family: tahoma, helvetica, sans-serif }
16          p         { font-size: 12pt;
17                      font-family: arial, sans-serif }
18          .special { color: #6666ff }
19       </style>
20    </head>
21    <body>
22       <!-- this class attribute applies the .special style -->
23       <h1 class = "special">Deitel &amp; Associates, Inc.</h1>
24
25       <p>Deitel &amp; Associates, Inc. is an internationally
26       recognized corporate training and publishing organization
27       specializing in programming languages, Internet/World
28       Wide Web technology and object technology education.
29       The company provides courses on Java, C++, Visual Basic,
30       C#, C, Internet and World Wide Web programming, Object
31       Technology, and more.</p>
```

```
8     <head>
9        <title>Style Sheets</title>
10
11       <!-- this begins the style sheet section -->
12       <style type = "text/css">
13          em        { font-weight: bold;
14                      color: black }
15          h1        { font-family: tahoma, helvetica, sans-serif }
16          p         { font-size: 12pt;
17                      font-family: arial, sans-serif }
18          .special { color: #6666ff }
19       </style>
20    </head>
21    <body>
22       <!-- this class attribute applies the .special style -->
23       <h1 class = "special">Deitel &amp; Associates, Inc.</h1>
24
25       <p>Deitel &amp; Associates, Inc. is an internationally
26       recognized corporate training and publishing organization
27       specializing in programming languages, Internet/World
28       Wide Web technology and object technology education.
29       The company provides courses on Java, C++, Visual Basic,
30       C#, C, Internet and World Wide Web programming, Object
31       Technology, and more.</p>
32
33       <h1>Clients</h1>
34       <p class = "special"> The company's clients include many
35       <em>Fortune 1000 companies</em>, government agencies,
36       branches of the military and business organizations.
37       Through its publishing partnership with Prentice Hall,
38       Deitel &amp; Associates, Inc. publishes leading-edge
39       programming textbooks, professional books, interactive
40       web-based multimedia Cyber Classrooms, satellite
41       courses and World Wide Web courses.</p>
42    </body>
43 </html>
```

External Style Sheets

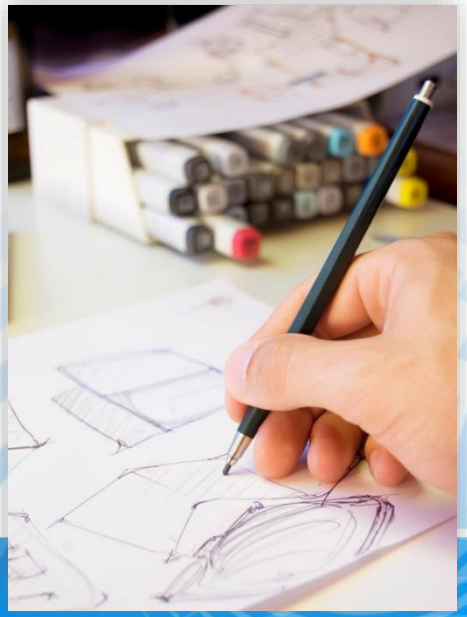
- External style sheets are separate documents that contain only CSS rules
- External linking of style sheets can create a uniform look for a website, since separate pages can all use the same styles
- Modifying a single style-sheet file makes changes to styles across an entire website

Software Engineering Observation

- Always use an external style sheet when developing a website with multiple pages. External style sheets separate content from presentation, allowing for more consistent look-and-feel, more efficient development, and better performance.

- Moreover, it's reusable!

reduces load time and bandwidth usage on a server



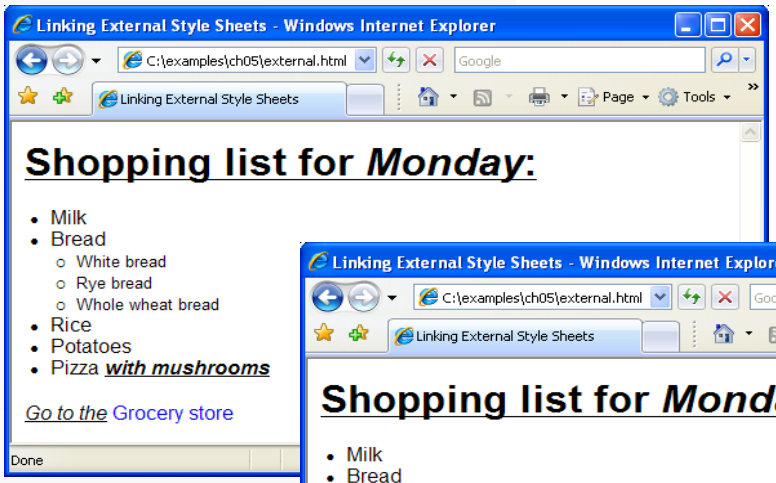
External Style Sheets

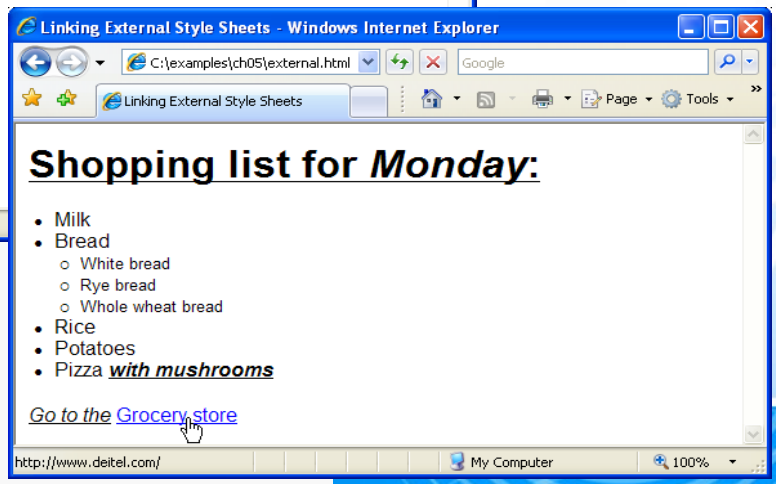
- The `link` element in a HTML document uses the `rel` attribute to specify a relationship between two documents
  - The `rel` attribute declares the linked document to be a stylesheet for the document
- The `href` attribute provides the URL for the document containing the style sheet

```html
<html>
   <head>
      <title>Linking External 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>

       <p><em>Go to the</em>
          <a class = "nodec" href = "http://www.deitel.com">
             Grocery store</a>
       </p>
    </body>
</html>
```

styles.css

```css
body    { font-family: arial, helvetica, sans-serif }

a.nodec  { text-decoration: none }

a:hover  { text-decoration: underline }

li em    { font-weight: bold }

h1, em   { text-decoration: underline }

ul       { margin-left: 20px }

ul ul    { font-size: .8em; }
```





Selector – HTML Tag

- The selector can be simply the name of a **HTML tag**
  - p {…}, h1 {…}, h4 {…}, td {…}, tr {…}, and many other tags

CSS (ex1.css)

```
body{background-color:yellow;}

h1{font-size:36pt;}

h2{color:blue;}

p{margin-left:50px;}
```

HTML

```
<html>
 <head>
  <link rel="stylesheet" type="text/css" href="ex1.css"/>
 </head>

 <body>

  <h1>This header is 36 pt</h1>
  <h2>This header is blue</h2>

  <p>This paragraph has a left margin of 50 pixels</p>

 </body>
</html>
```

Selector – HTML Tag

- This is an output

Conflicting Styles

- Most styles are inherited from parent elements. Styles defined for children have higher specificity and take precedence over the parent's styles
- Conflicts are resolved in favor of properties with a higher specificity

CHILDREN > PARENT

Why it is Cascading

- What'll happen:
  - When the **properties** have been set to the **same selector**

External

```
p: {
color:blue;
text-align:center;
font-size:8px;
}
```

Internal

```
p: {
text-align:left;
font-size:20px;
}
```

Result

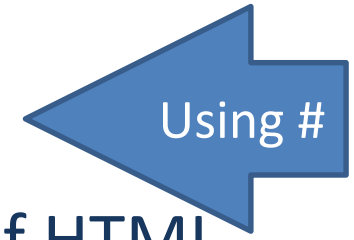
```
p: {
color:blue;
text-align:left;
font-size:20px;
}
```

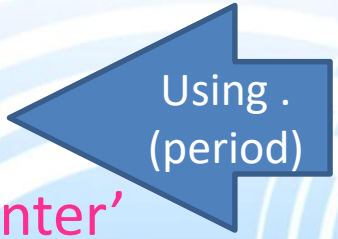
- **Multiple Styles will cascade into one**
  - Inline > Internal > External

Order: External Style Sheet → Internal Style Sheet → Inline Style

Selector – ID and Class

- ID Selector: to specify a style for a unique HTML element
  - The style's applied for tag(s) with a particular id Attribute
  - #itcs201 {color:green;}
  - This will be applied for a tag with id='itcs201'
- Class Selector: to specify a style for a group of HTML elements
  - The style's applied for any tags with a particular class Attribute
  - .center {text-align:center;}
  - h1.center {color:blue;}
  - This will be applied for every h1 tag with class='center'





Selector – ID and Class

CSS (ex2.css)

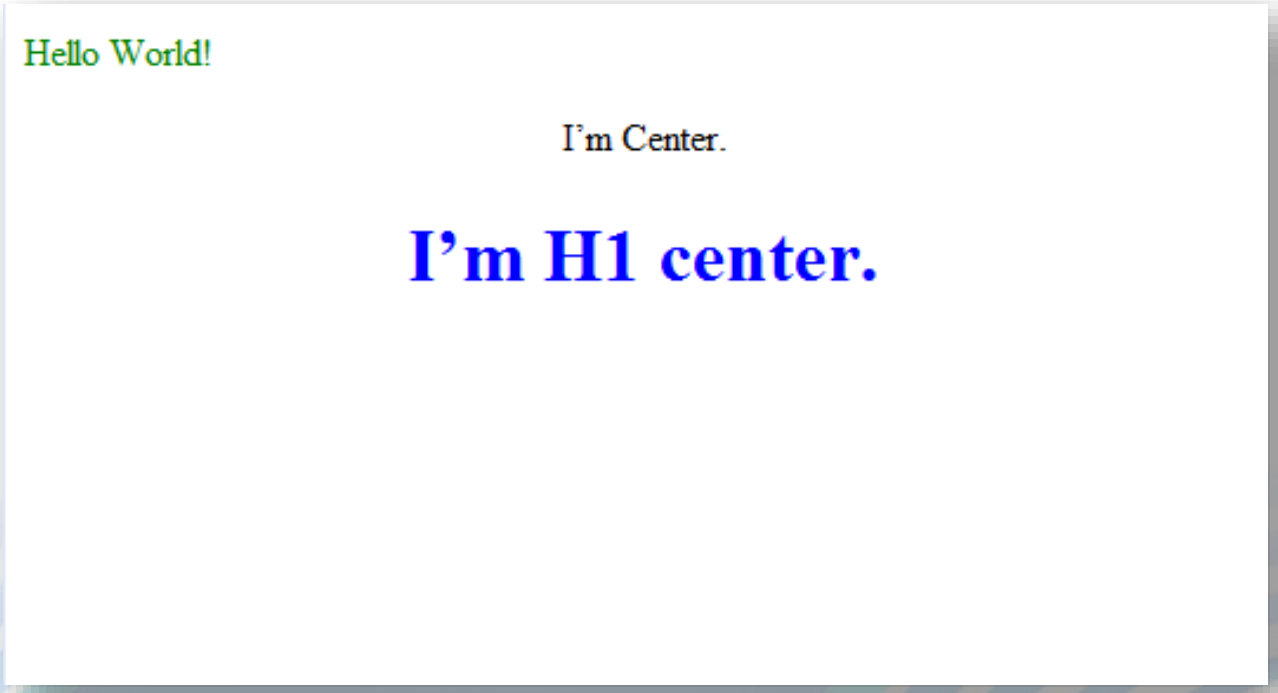
```
#itcs201
{ color:green;}

.center
{ text-align:center;}

h1.center
{ color:blue;}
```

HTML

```
<html>
 <head>
   <link rel="stylesheet"
   type="text/css" href="ex2.css" />
 </head>

 <body>
  <p id="itcs201">Hello World!</p>

  <p class='center'>I'm Center.</p>

  <h1 class='center'> I'm H1 center.
  </h1>
 </body>
</html>
```

http://www.w3schools.com/css/tryit.asp?filename=trycss_syntax_id

Selector – ID and Class



Selector – Grouping

- Group many selectors with the same style definition

```
h1
{
    color:green;
}
h2
{
    color:green;
}
p
{
    color:green;
}
```

```
h1,h2,p
{
    color:green;
}
```

Selector – Nesting

- To apply a style for a selector within a selector
- Apply only one class to the top level tag

```
table.myTableStyle
{
    border-color: #CCCCCC;
    border-width: 1px 1px 1px 1px;
    border-style: solid;
}

table.myTableStyle td
{
    padding: 8px 10px 8px 10px;
    vertical-align: top;
}
```

Selector - Pseudoclasses

- Pseudoclasses give the author access to content not specifically declared in the document
- Pseudoclasses are separated by a colon (with no surrounding spaces) from the name of the element to which they are applied
- The hover pseudoclass is activated when the user moves the mouse cursor over an element

a:hover

Selector - Pseudoclasses

- Used to add special effects to some selectors

```
a:link {
    color: Blue
}
a:visited {
    color: DarkBlue;
    text-decoration: underline;
}
a:hover {
    color: Green;
    text-decoration: none;
}
a:active {
    color: Orange;
    text-decoration: underline;
}
```

Selector

p,h1 {...}

- To apply rules to multiple elements, separate the elements with commas in the style sheet
- To apply rules to only a certain type of element that is a child of another type, separate the element names with spaces

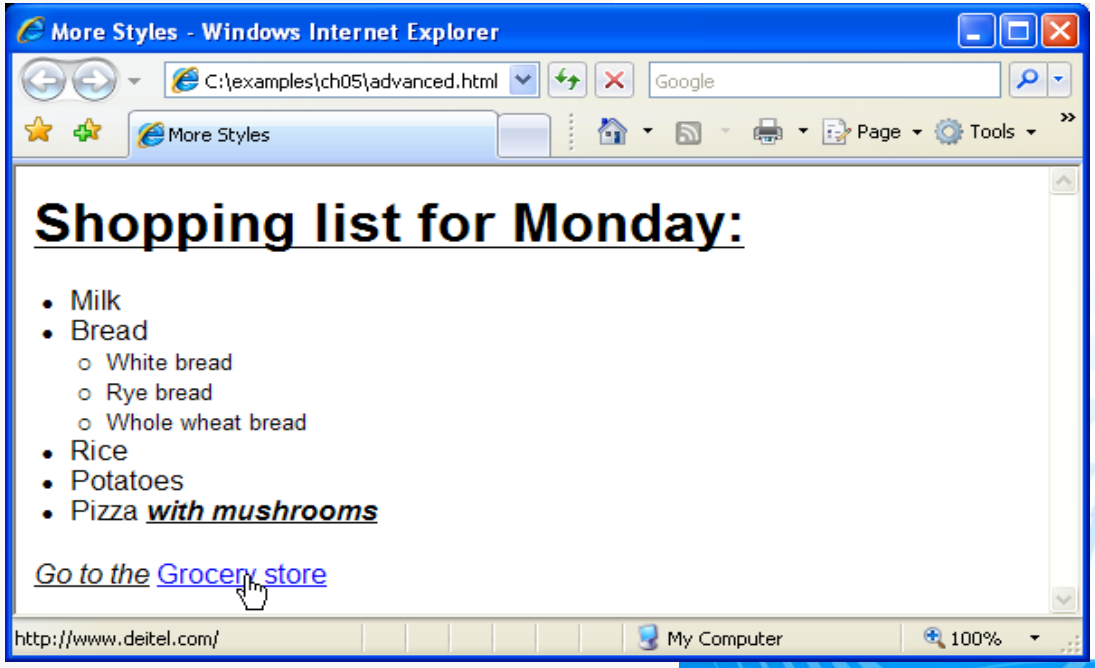
p b {...}

<h1>This is a heading 1</h1>

This is the first paragraph

This is a bold text in a paragraph two

```
7  <html>
8     <head>
9        <title>More Styles</title>
10       <style type = "text/css">
11          body     { font-family: arial, helvetica, sans-serif }
12          a.nodec  { text-decoration: none }
13          a:hover  { text-decoration: underline }
14          li em    { font-weight: bold }
15          h1, em   { text-decoration: underline }
16          ul       { margin-left: 20px }
17          ul ul    { font-size: .8em }
18       </style>
19    </head>
20    <body>
21       <h1>Shopping list for Monday:</h1>
23       <ul>
24          <li>Milk</li>
25          <li>Bread
26             <ul>
27                <li>White bread</li>
28                <li>Rye bread</li>
29                <li>Whole wheat bread</li>
30             </ul>
31          </li>
32          <li>Rice</li>
33          <li>Potatoes</li>
34          <li>Pizza <em>with mushrooms</em></li>
35       </ul>
37       <p><em>Go to the</em>
38          <a class = "nodec" href = "http://www.deitel.com">
39             Grocery store</a></p>
```



CSS Comments

- begins with "/*", and ends with "*/"

```
/*This is a comment*/
p
{
    text-align:center;
    /*This is another comment*/
    color:black;
    font-family:arial;
}
```

Example

- The `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

Example

- The `background-color` attribute specifies the background color of the element
- The `font-family` attribute names a specific font that should be displayed
  - 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
- The `font-size` property specifies the size used to render the font

Example

- The `text-decoration` property applies decorations to text in an element, such as
  - `underline`,
  - `overline`,
  - `line-through`

Relative vs Absolute length

- Relative length measurements:
  - px (pixels – size varies depending on screen resolution)
  - em (1em is equal to the current font size. 2em means 2 times the size of the current font.)
  - ex (usually the height of a font's lowercase x)
  - % (of the default size)
- Absolute-length measurements (units that do not vary in size):
  - in (inches)
  - cm (centimeters)
  - mm (millimeters)
  - pt (points; 1 pt = 1/72 in)
  - pc (picas; 1 pc = 12 pt)

Good Programming Practice

- Whenever possible, use relative-length measurements. If you use absolute-length measurements, your document may not be readable on some client browsers (e.g., wireless phones).

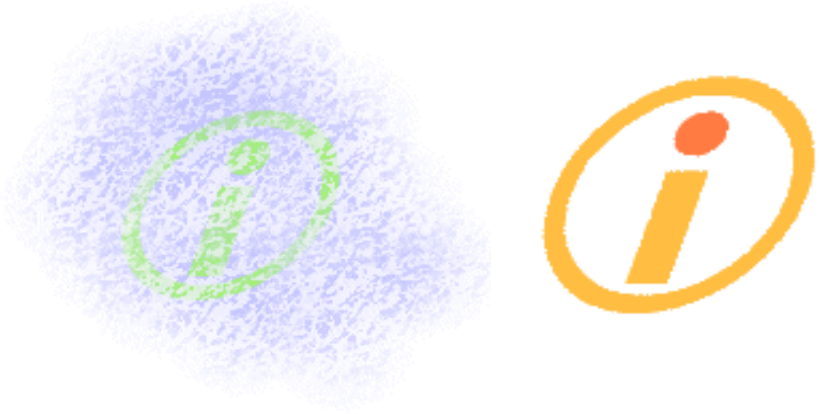
Positioning Elements

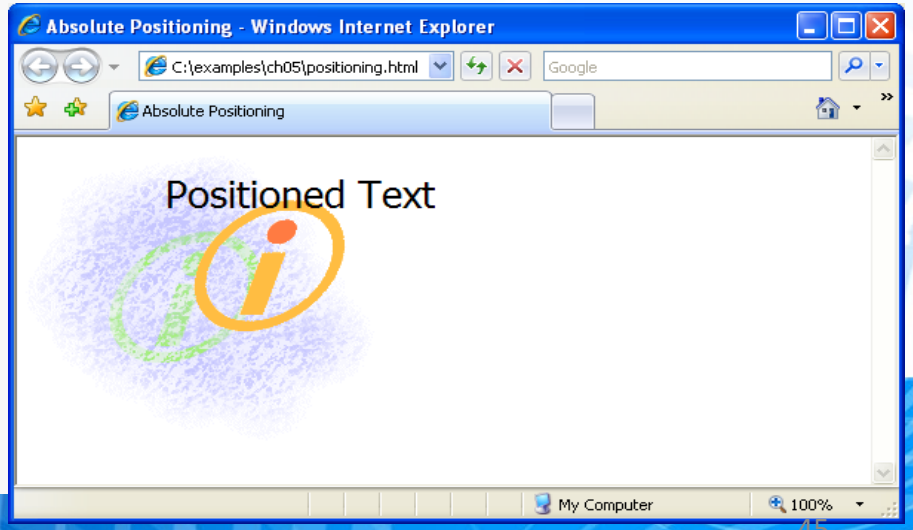
- The CSS position property allows **absolute positioning**, which provides greater control over where on a page elements reside
  - Normally, elements are positioned on the page in the order that they appear in the HTML document
- Specifying an element's position as absolute removes it from the normal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

Positioning Elements

- 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

```
<html>
   <head>
      <title>Absolute Positioning</title>
      <style type = "text/css">
         .bgimg   { position: absolute;
                    top: 0px;
                    left: 0px;
                    z-index: 1 }
         .fgimg   { position: absolute;
                    top: 25px;
                    left: 100px;
                    z-index: 2 }
         .text    { position: absolute;
                    top: 25px;
                    left: 100px;
                    z-index: 3;
                    font-size: 20pt;
                    font-family: tahoma, geneva, sans-serif }
      </style>
   </head>
   <body>
      <p><img src = "bgimg.gif" class = "bgimg"
         alt = "First positioned image" /></p>

      <p><img src = "fgimg.gif" class = "fgimg"
         alt = "Second positioned image" /></p>

      <p class = "text">Positioned Text</p>
   </body>
</html>
```

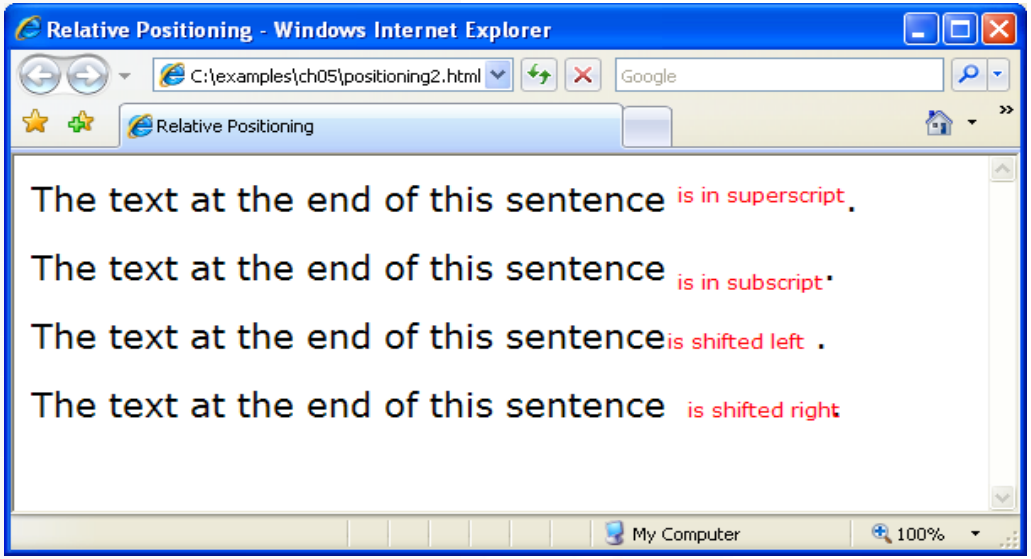




Positioning Elements

- **Relative positioning** keeps elements in the general flow on the page and offsets them by the specified top, left, right or bottom value
  - Inline-level elements
    - Do not change the flow of the document
    - E.g.,<img>,<a>,,,
  - Block-level elements
    - Display on their own line
    - Have virtual boxes around them
    - E.g.,<p>,<h1>-<h6>,

```
<html>
  <head>
    <title>Relative Positioning</title>
    <style type = "text/css">
        p           { font-size: 1.3em;
                      font-family: verdana, arial, sans-serif }
        span        { color: red;
                      font-size: .6em;
                      height: 1em }
        .super      { position: relative;
                      top: -1ex }
        .sub        { position: relative;
                      bottom: -1ex }
        .shiftleft  { position: relative;
                      left: -1ex }
        .shiftright { position: relative;
                      right: -1ex }
    </style>
  </head>
  <body>
    <p>The text at the end of this sentence
    <span class = "super">is in superscript</span>.</p>
    <p>The text at the end of this sentence
    <span class = "sub">is in subscript</span>.</p>

    <p>The text at the end of this sentence
    <span class = "shiftleft">is shifted left</span>.</p>

    <p>The text at the end of this sentence
    <span class = "shiftright">is shifted right</span>.</p>
  </body>
</html>
```



Backgrounds

- CSS can control the backgrounds of block-level elements by adding:
  - Colors
  - Images

Example

- Property `background-image` specifies the URL of the image, in the format url(fileLocation)
- The property `background-position` 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

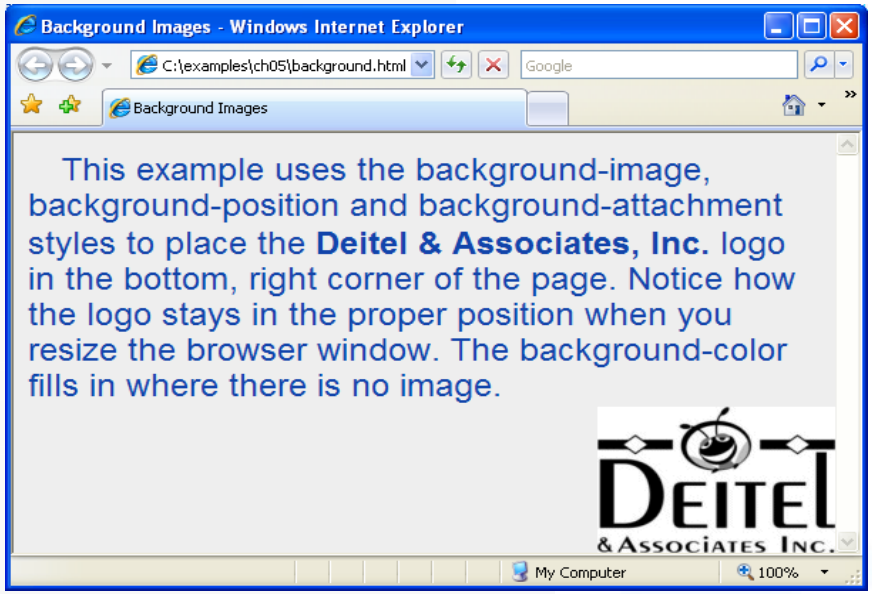
Example

- 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
  - 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

Example

- 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 text in the element by the specified amount

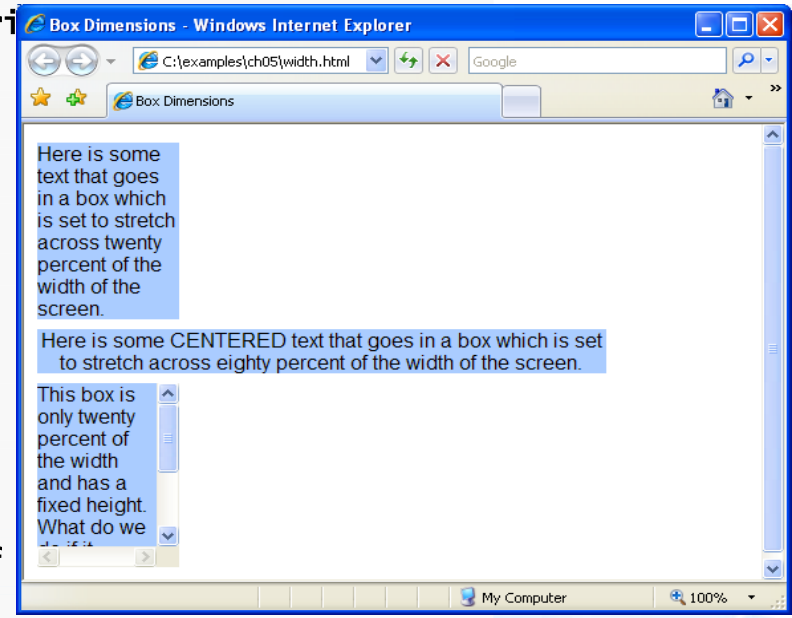
```
<html>
   <head>
      <title>Background Images</title>
      <style type = "text/css">
         body  { background-image: url(logo.gif);
                 background-position: bottom right;
                 background-repeat: no-repeat;
                 background-attachment: fixed;
                 background-color: #eeeeee }
         p     { font-size: 18pt;
                 color: #1144AA;
                 text-indent: 1em;
                 font-family: arial, sans-serif; }
         .dark { font-weight: bold }
      </style>
   </head>
   <body>
      <p>
      This example uses the background-image,
      background-position and background-attachment
      styles to place the <span class = "dark">Deitel
      &amp;amp; Associates, Inc.</span> logo in the bottom,
      right corner of the page. Notice how the logo
      stays in the proper position when you resize the
      browser window. The background-color fills in where
      there is no image.
      </p>
    </body>
</html>
```



Element Dimensions

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

```
7  <html>
8     <head>
9        <title>Box Dimensions</title>
10       <style type = "text/css">
11          div { background-color: #aaccff;
12                margin-bottom: .5em;
13                font-family: arial, helvetica, sans-seri
14       </style>
15    </head>
16    <body>
17       <div style = "width: 20%">Here is some
18       text that goes in a box which is
19       set to stretch across twenty percent
20       of the width of the screen.</div>
21
22       <div style = "width: 80%; text-align: center">
23       Here is some CENTERED text that goes in a box
24       which is set to stretch across eighty percent of
25       the width of the screen.</div>
 27       <div style = "width: 20%; height: 150px; overflow: scroll">
 28       This box is only twenty percent of
 29       the width and has a fixed height.
 30       What do we do if it overflows? Set the
 31       overflow property to scroll!</div>
 32    </body>
 33 </html>
```



Text Flow

- 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`
- A 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`

Border

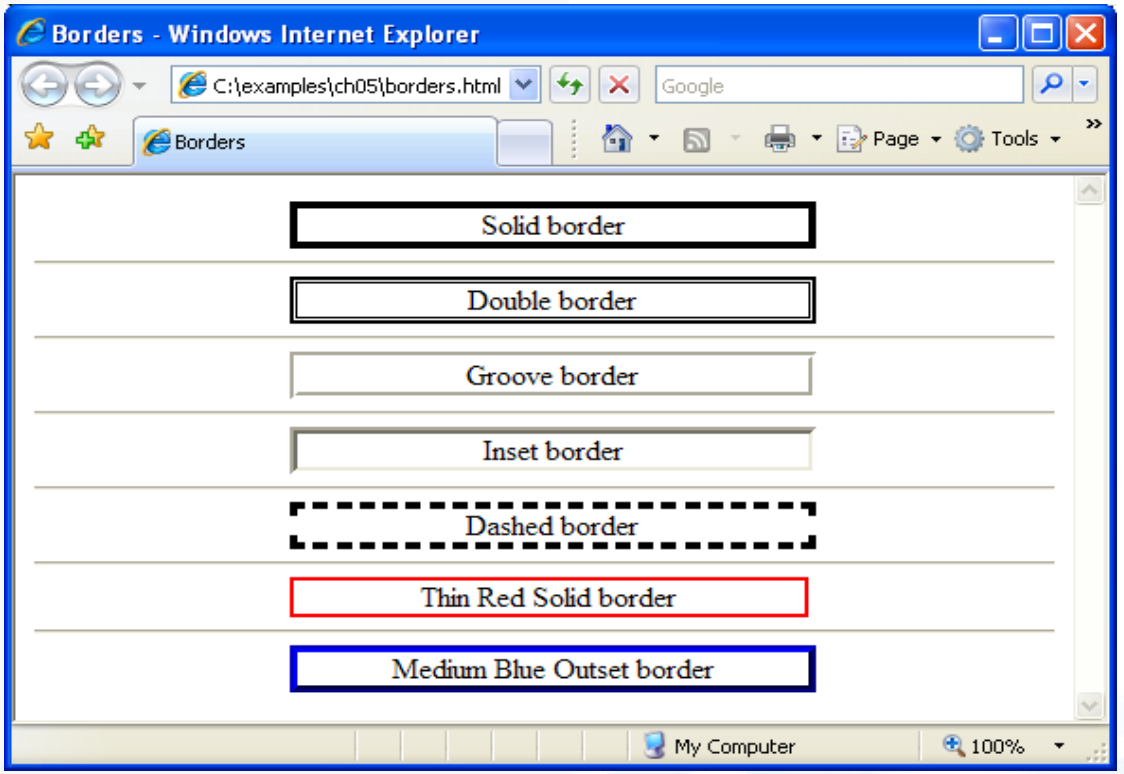
- The border is controlled using the properties:
  - `border-width`
    - May be set to any of the CSS lengths or to the predefined value of `thin`, `medium` or `thick`
  - `border-color`
    - Sets the color used for the border
  - `border-style`
    - Options are: `none`, `hidden`, `dotted`, `dashed`,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

Box model for block-level elements.

Class

- The `class` attribute allows more than one class to be assigned to a HTML element by separating each class name from the next with a space

```
<html>
   <head>
      <title>Borders</title>
      <style type = "text/css">
         div     { text-align: center;
                   width: 50%;
                   position: relative;
                   left: 25%;
                   border-width: 4px }
         .medium { border-width: medium }
         .thin   { border-width: thin }
         .solid  { border-style: solid }
         .double { border-style: double }
         .groove { border-style: groove }
         .inset  { border-style: inset }
         .outset { border-style: outset }
         .dashed { border-style: dashed }
         .red    { border-color: red }
         .blue   { border-color: blue }
      </style>
   </head>
   <body>
      <div class = "solid">Solid border</div><hr />
      <div class = "double">Double border</div><hr />
      <div class = "groove">Groove border</div><hr />
      <div class = "inset">Inset border</div><hr />
     <div class = "dashed">Dashed border</div><hr />
     <div class = "thin red solid">Thin Red Solid border</div><hr />
      <div class = "medium blue outset">Medium Blue Outset border</div>
   </body>
</html>
```

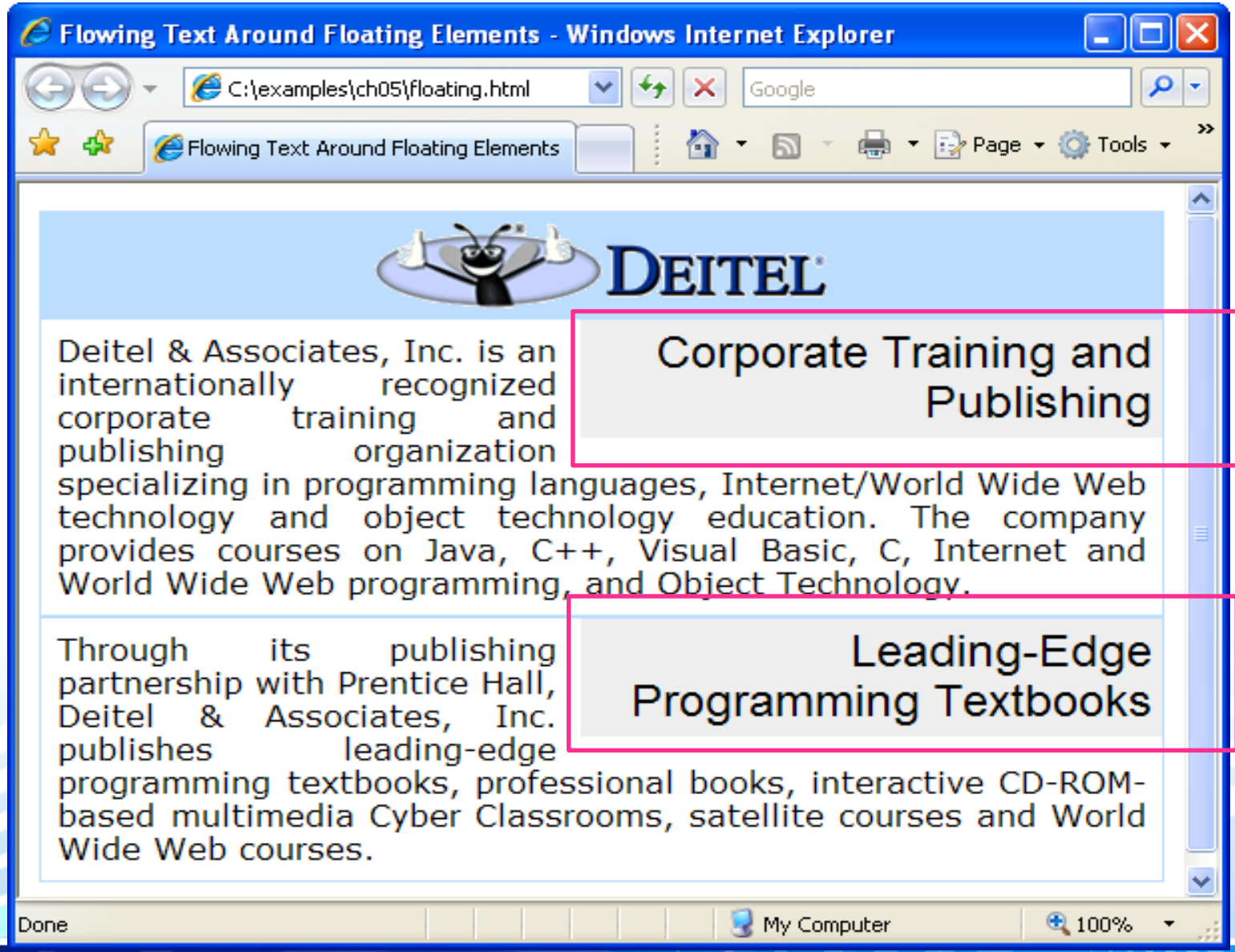


Float

- Browsers normally place text and elements on screen in the order in which 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

```
<html>
   <head>
      <title>Flowing Text Around Floating Elements</title>
      <style type = "text/css">
         div.heading { background-color: #bbddff;
                       text-align: center;
                       font-family: arial, helvetica, sans-serif;
                       padding: .2em }
         p           { text-align: justify;
                       font-family: verdana, geneva, sans-serif;
                       margin: .5em }
         div.floated { background-color: #eeeeee;
                       font-size: 1.5em;
                       font-family: arial, helvetica, sans-serif;
                       padding: .2em;
                       margin-left: .5em;
                       margin-bottom: .5em;
                       float: right;
                       text-align: right;
                       width: 50% }
         div.section { border: 1px solid #bbddff }
      </style>
   </head>
```

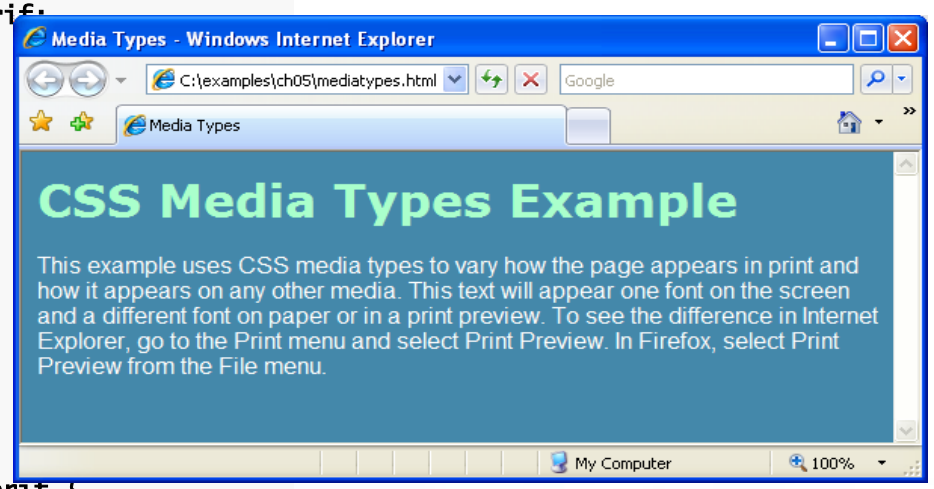
```
30    <body>
31       <div class = "heading"><img src = "deitel.png" alt = "Deitel" />
32          </div>
33       <div class = "section">
34          <div class = "floated">Corporate Training and Publishing</div>
35          <p>Deitel &amp;amp; Associates, Inc. is an internationally
36          recognized corporate training and publishing organization
37          specializing in programming languages, Internet/World
38          Wide Web technology and object technology education.
39          The company provides courses on Java, C++, Visual Basic, C#,
40          C, Internet and web programming, Object
41          Technology, and more.</p>
42       </div>
43       <div class = "section">
44          <div class = "floated">Leading-Edge Programming Textbooks</div>
45          <p>Through its publishing
46          partnership with Prentice Hall, Deitel &amp;amp; Associates,
47          Inc. publishes leading-edge programming textbooks,
48          professional books, interactive CD-ROM-based multimedia
49          Cyber Classrooms, satellite courses and DVD and web-based
50          video courses.</p>
51       </div>
52    </body>
53 </html>
```

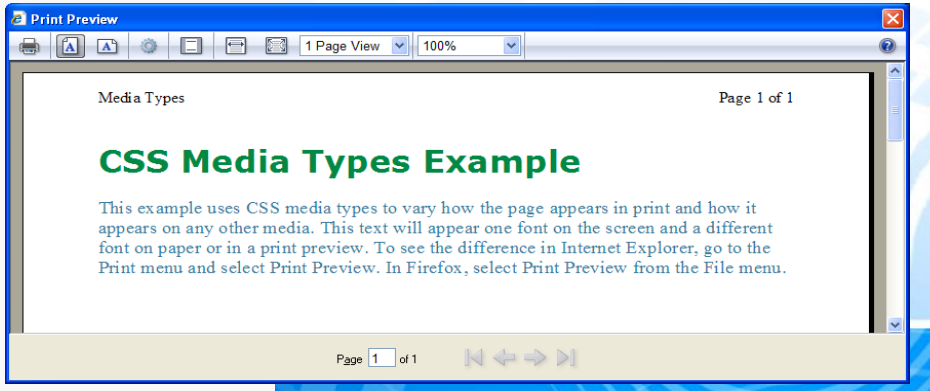

Flowing Text Around Floating Elements - Windows Internet Explorer
C:\examples\ch05\floating.html
Google
Flowing Text Around Floating Elements
Page
Tools
DEITEL
Corporate Training and Publishing
Deitel & Associates, Inc. is an internationally recognized corporate training and publishing organization specializing in programming languages, Internet/World Wide Web technology and object technology education. The company provides courses on Java, C++, Visual Basic, C, Internet and World Wide Web programming, and Object Technology.
Leading-Edge Programming Textbooks
Through its publishing partnership with Prentice Hall, Deitel & Associates, Inc. publishes leading-edge programming textbooks, professional books, interactive CD-ROM-based multimedia Cyber Classrooms, satellite courses and World Wide Web courses.
Done
My Computer
100%

Media Types

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

```
<html>
   <head>
      <title>Media Types</title>
      <style type = "text/css">
         @media all
         {
            body  { background-color: #4488aa }
            h1    { font-family: verdana, helvetica, sans-serif;
                    color: #aaffcc }
            p     { font-size: 12pt;
                    color: white;
                    font-family: arial, sans-serif }
         } /* end @media all declaration. */
         @media print
         {
            body  { background-color: white }
            h1    { color: #008844}
            p     { font-size: 14pt;
                    color: #4488aa;
                    font-family: "times new roman", times, serif }
         } /* end @media print declaration. */
      </style>
   </head>
   <body>
      <h1>CSS Media Types Example</h1>
      <p>
      This example uses CSS media types to vary how the page
      appears in print and how it appears on any other media.
      ......
      Preview from the File menu.
      </p>
   </body>
</html>
```





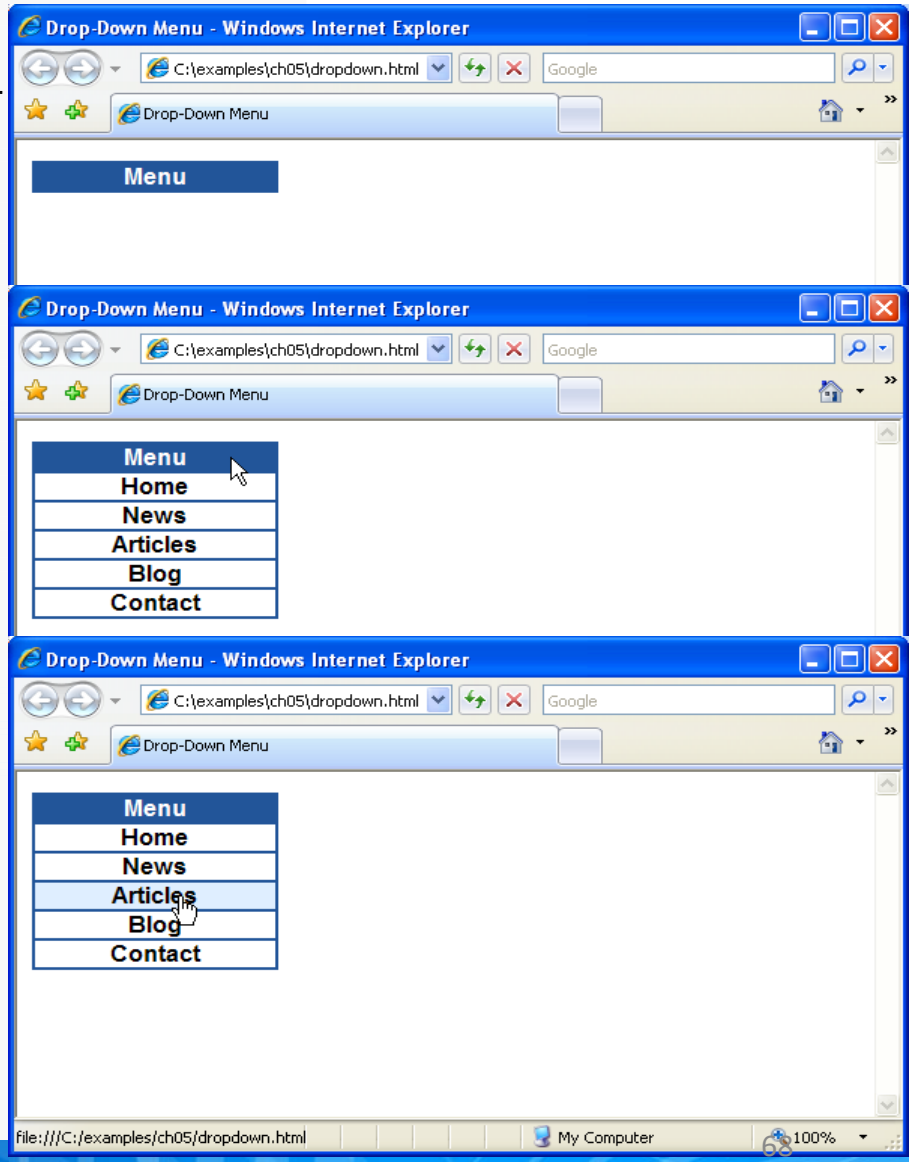
Media Types

- Other media types include:
  - `handheld`
    - Designed for mobile Internet devices
  - `braille`
    - For machines that can read or print web pages in braille
  - `aural`
    - 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

Building a CSS Drop-Down Menu

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

```
7  <html xmlns = "http://www.w3.org/1999/xhtml">
8     <head>
9        <title>
10          Drop-Down Menu
11       </title>
12       <style type = "text/css">
13          body              { font-family: arial, sans-serif }
14          div.menu          { font-weight: bold;
15                              color: white;
16                              border: 2px solid #225599;
17                              text-align: center;
18                              width: 10em;
19                              background-color: #225599 }
20          div.menu:hover a  { display: block }
21          div.menu a        { display: none;
22                              border-top: 2px solid #225599;
23                              background-color: white;
24                              width: 10em;
25                              text-decoration: none;
26                              color: black }
27          div.menu a:hover  { background-color: #dfeeff }
28       </style>
29    </head>
30    <body>
31       <div class = "menu">Menu
32          <a href = "#">Home</a>
33         <a href = "#">News</a>
34         <a href = "#">Articles</a>
35          <a href = "#">Blog</a>
36          <a href = "#">Contact</a>
37       </div>
38    </body>
39 </html>
```



WHAT IS CSS3?

CSS3 has been split into "modules". It contains the old CSS specification and new modules are added.
Some of the most important CSS3 modules are:

- Selectors
- Box Model
- Backgrounds and Borders
- Image Values and Replaced Content
- Text Effects
- 2D/3D Transformations
- Animations
- Multiple Column Layout
- User Interface

Prefix

- Different web browsers require different prefixes to run CSS3
  - Internet Explorer → no prefix
  - Firefox → -moz-
  - Chrome and Safari → -webkit-
  - Opera → -o-

Border

- border-radius

```
border-radius:2px;

is equivalent to:

border-top-left-radius:2px;
border-top-right-radius:2px;
border-bottom-right-radius:2px;
border-bottom-left-radius:2px;
```

This box has rounded corners!

- border-image

```
border-image:url("border.png") 30 30 round;
```

The border-image property allows you to specify an image as a border!

Shadow

- box-shadow: *h-shadow v-shadow blur spread color* inset;
  - h-shadow → Required. The position of the horizontal shadow.
  - v-shadow → Required. The position of the vertical shadow.
  - blur → Optional. The blur distance
  - spread → Optional. The size of shadow
  - color → Optional. The color of the shadow
  - inset → Optional. Changes the shadow from an outer shadow (outset) to an inner shadow

  E.g., box-shadow: 10px 10px 5px #888888;

Animation

- To create animations in CSS3, you will have to learn about the @keyframes rule
- Specify a CSS style inside the @keyframes rule and the animation will gradually change from the current style to the new style

```
@keyframes myfirst
{
from {background: red;}
to {background: yellow;}
}
@-moz-keyframes myfirst /* Firefox */
{…}
@-webkit-keyframes myfirst /* Safari and Chrome */
{…}
@-o-keyframes myfirst /* Opera */
{…}
```

Animation

- Bind @keyframe to a selector

```
div
{
animation: myfirst 5s;
-moz-animation: myfirst 5s; /* Firefox */
-webkit-animation: myfirst 5s; /* Safari and Chrome */
-o-animation: myfirst 5s; /* Opera */
}
```

Animation

- In @keyframes, there may be many points of time
- At any points of time, there may be many styles

```
@keyframes myfirst
{
0%   {background: red; left:0px; top:0px;}
25%  {background: yellow; left:200px; top:0px;}
50%  {background: blue; left:200px; top:200px;}
75%  {background: green; left:0px; top:200px;}
100% {background: red; left:0px; top:0px;}
}
```

Animation

Property	Description
@keyframes	Specifies the animation
animation	A shorthand property for all the the animation properties, except the animation-play-state property
animation-name	Specifies the name of the @keyframes animation
animation-duration	Specifies how many seconds or milliseconds an animation takes to complete one cycle. Default 0
animation-timing-function	Describes how the animation will progress over one cycle of its duration. Default "ease"
animation-delay	Specifies when the animation will start. Default 0
animation-iteration-count	Specifies the number of times an animation is played. Default 1
animation-direction	Specifies whether or not the animation should play in reverse on alternate cycles. Default "normal"
animation-play-state	Specifies whether the animation is running or paused. Default "running"

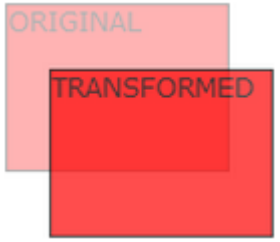
Fonts

- Before CSS3, web designers had to use fonts that were already installed on the user's computer.
- With CSS3, web designers can use whatever font he/she likes.

```
@font-face
{
font-family: myFirstFont;
src: url('Sansation_Light.ttf')
font-weight:bold;
}
div
{
font-family:myFirstFont;
}
```

2D Transform

- translate(*x*,*y*)
  - The element moves from its current position, depending on the parameters given for the left (X-axis) and the top (Y-axis) position
  - E.g., transform: translate(50px,100px);



- rotate(angle)
  - The element rotates clockwise at a given degree. Negative values are allowed and rotates the element counter-clockwise.
  - E.g., transform:rotate(50deg);

2D Transform

- scale(x,y)
  - The element increases or decreases the size, depending on the parameters given for the width (X-axis) and the height (Y-axis)
  - E.g., transform: scale(2,4);

- skew(x,y)
  - The element turns in a given angle, depending on the parameters given for the horizontal (X-axis) and the vertical (Y-axis) lines
  - E.g., transform: skew(30deg,20deg);

- 2D Transform and JavaScript
  - *object*.style.transform="rotate(7deg)“
  - EX. http://www.w3schools.com/cssref/trycss3_transform_inuse.htm
- 3D Transform is also possible

Transition

- CSS3 transitions are effects that let an element gradually change from one style to another.

```
div
{
transition: width 2s;
-moz-transition: width 2s; /* Firefox 4 */
-webkit-transition: width 2s; /* Safari and Chrome */
-o-transition: width 2s; /* Opera */
}
div:hover
{
width:300px;
}
```

Transition: Multiple changes

```
div
{
width:100px;
height:100px;
background:red;
transition:width 2s, height 2s;
-moz-transition:width 2s, height 2s, -moz-transform 2s; /* Firefox 4 */
-webkit-transition:width 2s, height 2s, -webkit-transform 2s; /* Safari and Chrome */
-o-transition:width 2s, height 2s, -o-transform 2s; /* Opera */
}

div:hover
{
width:200px;
height:200px;
transform:rotate(180deg);
-moz-transform:rotate(180deg); /* Firefox 4 */
-webkit-transform:rotate(180deg); /* Safari and Chrome */
-o-transform:rotate(180deg); /* Opera */
}
```

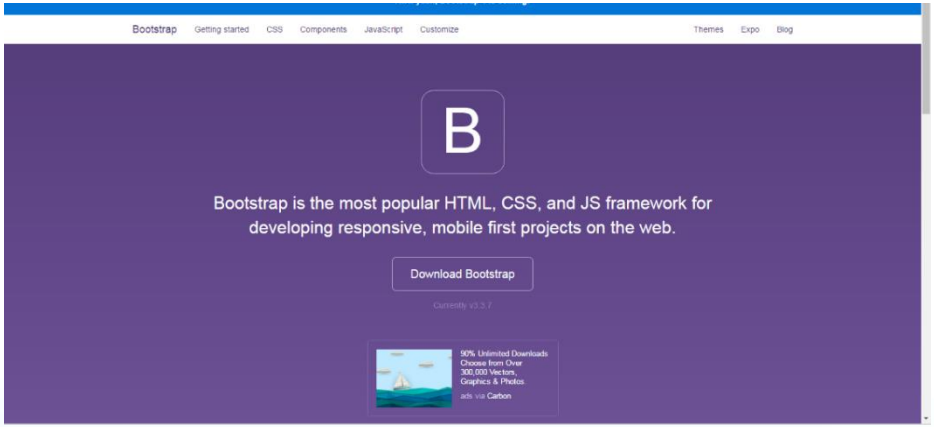
Transition

Property	Description
transition	A shorthand property for setting the four transition properties into a single property
transition-property	Specifies the name of the CSS property to which the transition is applied
transition-duration	Defines the length of time that a transition takes. Default 0
transition-timing-function	Describes how the speed during a transition will be calculated. Default "ease"
transition-delay	Defines when the transition will start. Default 0

Value	Description
linear	Specifies a transition effect with the same speed from start to end (equivalent to cubic-bezier(0,0,1,1))
ease	Specifies a transition effect with a slow start, then fast, then end slowly (equivalent to cubic-bezier(0.25,0.1,0.25,1))
ease-in	Specifies a transition effect with a slow start (equivalent to cubic-bezier(0.42,0,1,1))
ease-out	Specifies a transition effect with a slow end (equivalent to cubic-bezier(0,0,0.58,1))
ease-in-out	Specifies a transition effect with a slow start and end (equivalent to cubic-bezier(0.42,0,0.58,1))
cubic-bezier(*n*,*n*,*n*,*n*)	Define your own values in the cubic-bezier function. Possible values are numeric values from 0 to 1

- Cubic-Bezier Generator: http://cubic-bezier.com/#.17,.67,.83,.67
- Additional Example: http://www.impressivewebs.com/css3-transitions-without-hover/

CSS BOOSTRAP



http://getbootstrap.com/

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.

Bootstrap is completely free to download and use!

Features

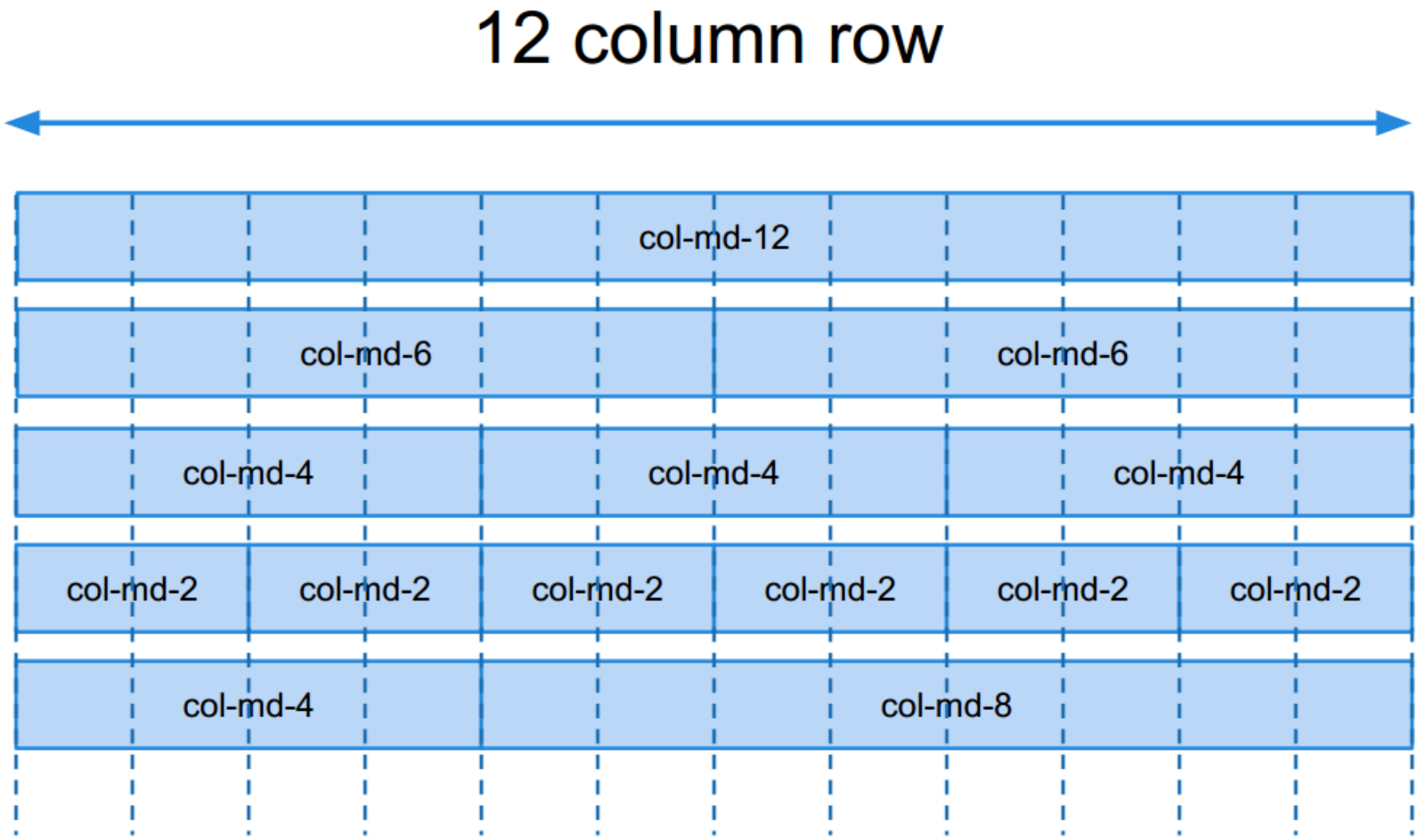
- **CSS resetting for cross browser compatibility**
  - Grid scaffolding for design
  - Multi-screen support (responsive design)
  - “Mobile first”, like jQuery Mobile
  - And a really good looking UI framework

How to add Bootstrap

You should use a CDN (content delivery network) for loading resources.
http://www.bootstrapcdn.com/

```
<!DOCTYPE html>
<html>
<head>
<title>Page title</title>
<link href="//netdna.bootstrapcdn.com/bootstrap/3.0.0/css/bootstrap.min.css"
rel="stylesheet">
<script src="//netdna.bootstrapcdn.com/bootstrap/3.0.0/js/bootstrap.min.js"></script>
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-
scale=1.0, userscalable=no">
</head>
<body>
<!-- page content goes here -->
</body>
</html>
```

The grid layout



How to write pages using a grid:

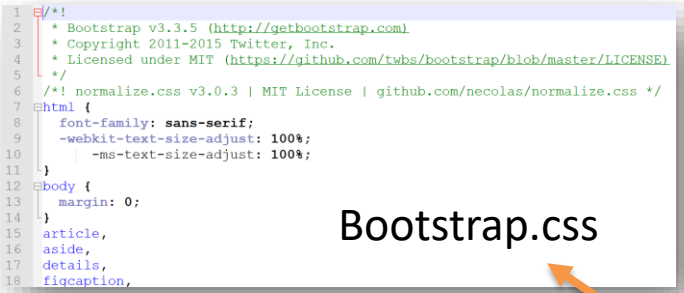
```
<!-- every row must have 12 columns -->
<div class=”row”>
<div class=”col-md-4”>
<!-- content -->
</div>
<!-- need to complete 8 more columns -->
</div>
```

Bootstrap 3 features an always-responsive grid with a maximum size:

1. col-xs-[num] grids have no maximum size (fluid)
2. col-sm-[num] grids resize up to 750px
3. col-md-[num] grids resize up to 970px
4. col-lg-[num] grids resize up to 1170px

You should choose col-md or col-lg for desktop sites.

How to write pages using a grid:



Bootstrap.css

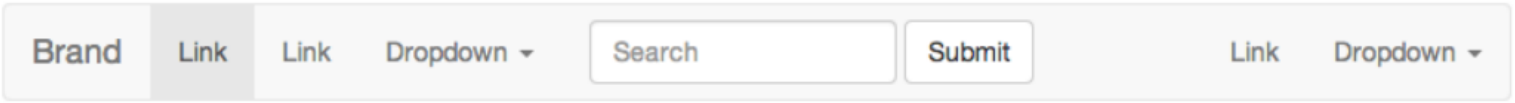
```css
.leftbar{
    background-color: red;
}
.center{
    background-color: green;
}
.rightbar{
    background-color: blue;
}
```

styleAct3.css

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Bootstrap Demo</title>
        <link rel="stylesheet" type="text/css" href="bootstrap.css">
        <link rel="stylesheet" type="text/css" href="styleAct3.css">
    </head>
    <body>
        <div class="container">
            <div class="row">
                <div class="col-md-4 hidden-sm hidden-xs">
                    <div class="leftbar">left sidebar</div>
                </div>
                <div class="col-md-4 col-sm-8 col-xs-12">
                    <div class="center">center</div>
                </div>
                <div class="col-md-4 col-sm-4 hidden-xs">
                    <div class="rightbar">right sidebar</div>
                </div>
            </div>
        </div>
    </body>
</html>
```

How to write pages using a grid:

Bootstrap elements



- Dropdowns
- Tabs
- Pills
- Navbars
- Alerts
- Lists
- Buttons

Check out the Bootstrap reference for more:
http://getbootstrap.com/components/