



```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD
2 "http://www.w3.org/TR/html4/strict
3 <html>
4   <head>
5     <title>Example</title>
6     <link rel="stylesheet" href="s
7   </head>
8   <body>
9     <div id="header">
10       <h1><a href="#" title="Back
11         </div>
12       <div id="toolbar">
13         <span class="left">Today <sp
14         <span class="right">
15           <span id="time">&ampnbsp</sp
16           <select id="timezone">
17             <option value="-12"> (GMT
18             <option value="-11"> (GMT
```

HTML Basics

HTML, Text, Images, Tables

1. Introduction to HTML

- How the Web Works?
- What is a Web Page?
- My First HTML Page
- Basic Tags: Hyperlinks, Images, Formatting
- Headings and Paragraphs

2. HTML in Details

- The `<head>` Section: Title, Meta, Script, Style

2. HTML in Details

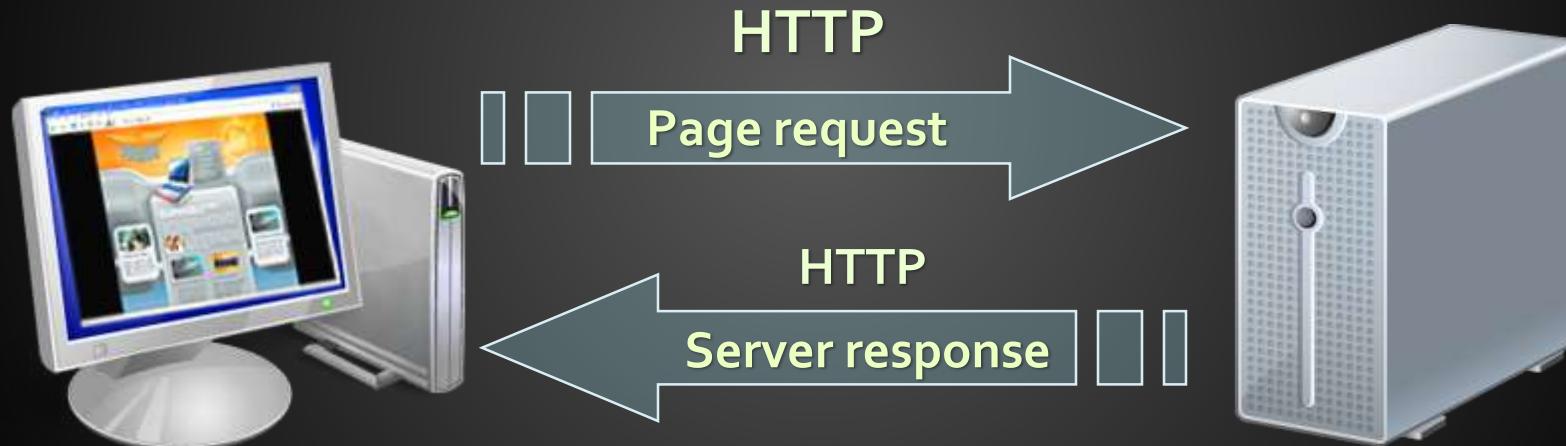
- The `<body>` Section
- Text Styling and Formatting Tags
- Hyperlinks: `<a>`, Hyperlinks and Sections
- Images: ``
- Lists: ``, `` and `<dl>`

3. The `<div>` and `` elements

4. HTML Tables

5. HTML Forms

- ◆ WWW use classical client / server architecture
 - HTTP is request-response protocol

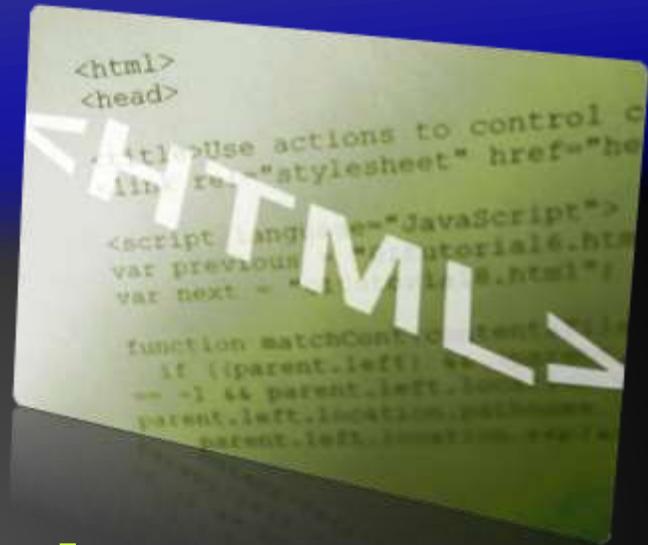


Client running a
Web Browser

Server running Web
Server Software
(IIS, Apache, etc.)

- ◆ Web pages are text files containing HTML
- ◆ HTML – Hyper Text Markup Language
 - ◆ A notation for describing
 - ◆ document structure (semantic markup)
 - ◆ formatting (presentation markup)
 - ◆ Looks (looked?) like:
 - ◆ A Microsoft Word document
 - ◆ The markup tags provide information about the page content structure

- ◆ An HTML file must have an .htm or .html file extension
- ◆ HTML files can be created with text editors:
 - ◆ NotePad, NotePad ++, PSPad
- ◆ Or HTML editors :
 - ◆ Microsoft FrontPage
 - ◆ Macromedia Dreamweaver
 - ◆ Netscape Composer
 - ◆ Microsoft Word
 - ◆ Visual Studio



HTML Basics

Text, Images, Tables, Forms



- ◆ HTML is comprised of “elements” or “tags”
 - ◆ Begins with `<html>` and ends with `</html>`
- ◆ Elements (tags) are nested one inside another:

```
<html> <head></head> <body></body> </html>
```

- ◆ Tags have attributes:

```

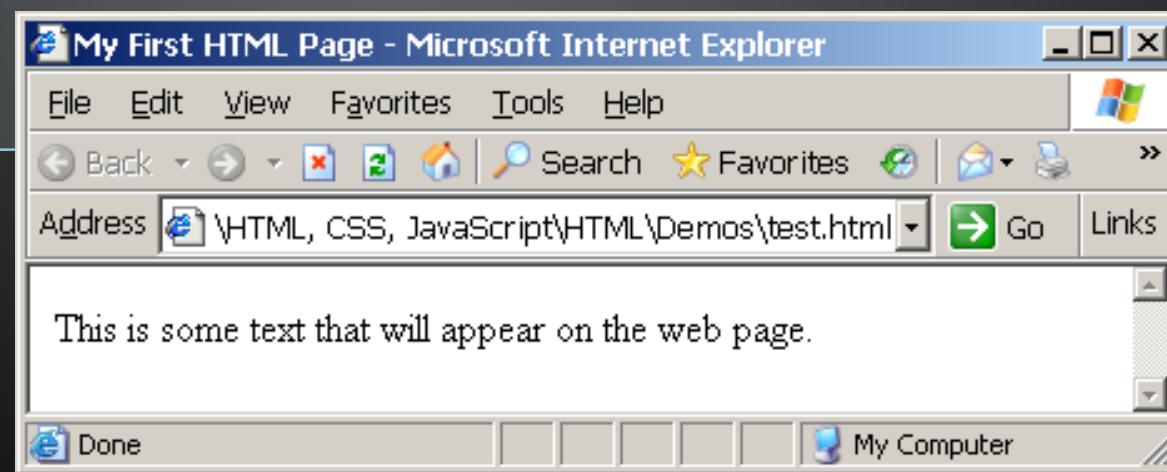
```

- ◆ HTML describes structure using two main sections:
`<head>` and `<body>`

- ◆ The HTML source code should be formatted to increase readability and facilitate debugging.
 - ◆ Every block element should start on a new line.
 - ◆ Every nested (block) element should be indented.
 - ◆ Browsers ignore multiple whitespaces in the page source.

test.html

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```



```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

Opening tag

Closing tag

An HTML element consists of an opening tag, a closing tag and the content inside.

First HTML Page: Header

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```

HTML header

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>This is some text...</p>
  </body>
</html>
```



HTML body

- ◆ Hyperlink Tags

```
<a href="http://www.telerik.com/"  
    title="Telerik">Link to Telerik Web site</a>
```

- ◆ Image Tags

```

```

- ◆ Text formatting tags

```
This text is <em>emphasized.</em>  
<br />new line<br />  
This one is <strong>more emphasized.</strong>
```

some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.com/" title=
  "Telerik site">This is a link.</a>
<br />

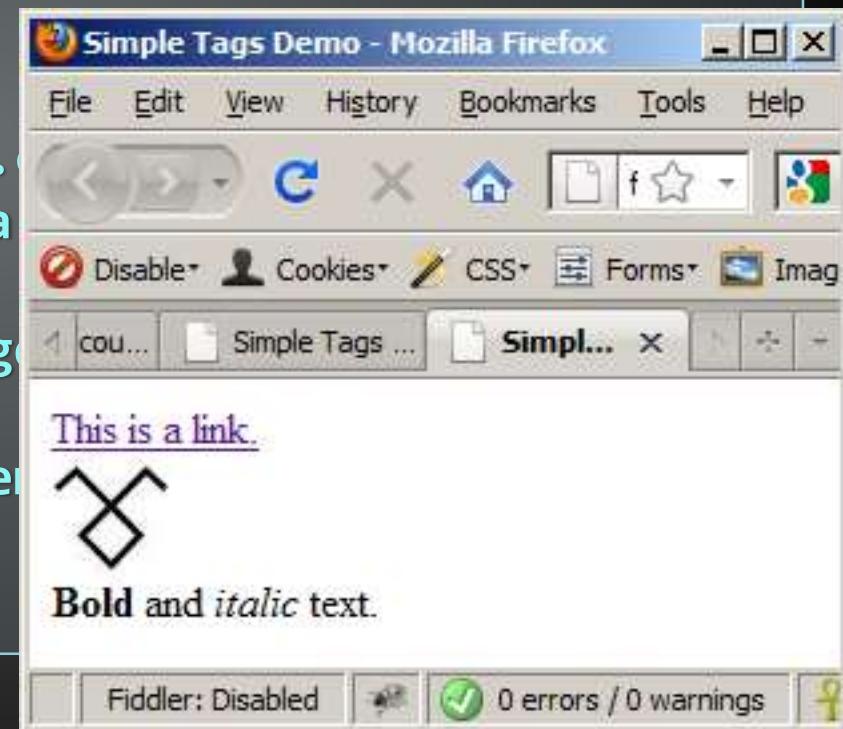
<br />
<strong>Bold</strong> and <em>italic</em> text.
</body>
</html>
```

telerik Some Simple Tags – Example (2)

some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
    <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.com">Telerik site</a>
<br />

<br />
<strong>Bold</strong> and <em>italic</em>
</body>
</html>
```



- ◆ Tags can have attributes
 - ◆ Attributes specify properties and behavior
 - ◆ Example:

Attribute alt with value "logo"

```

```

- ◆ Few attributes can apply to every element:
 - ◆ id, style, class, title
 - ◆ The id is unique in the document
 - ◆ Content of title attribute is displayed as hint when the element is hovered with the mouse
 - ◆ Some elements have obligatory attributes

- ◆ Heading Tags (h1 – h6)

```
<h1>Heading 1</h1>
<h2>Sub heading 2</h2>
<h3>Sub heading 3</h3>
```

- ◆ Paragraph Tags

```
<p>This is my first paragraph</p>
<p>This is my second paragraph</p>
```

- ◆ Sections: div

```
<div style="background: skyblue;">
    This is a div</div>
```

Headings and Paragraphs – Example

headings.html

```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and paragraphs</title></head>
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>

    <div style="background:skyblue">
      This is a div</div>
  </body>
</html>
```

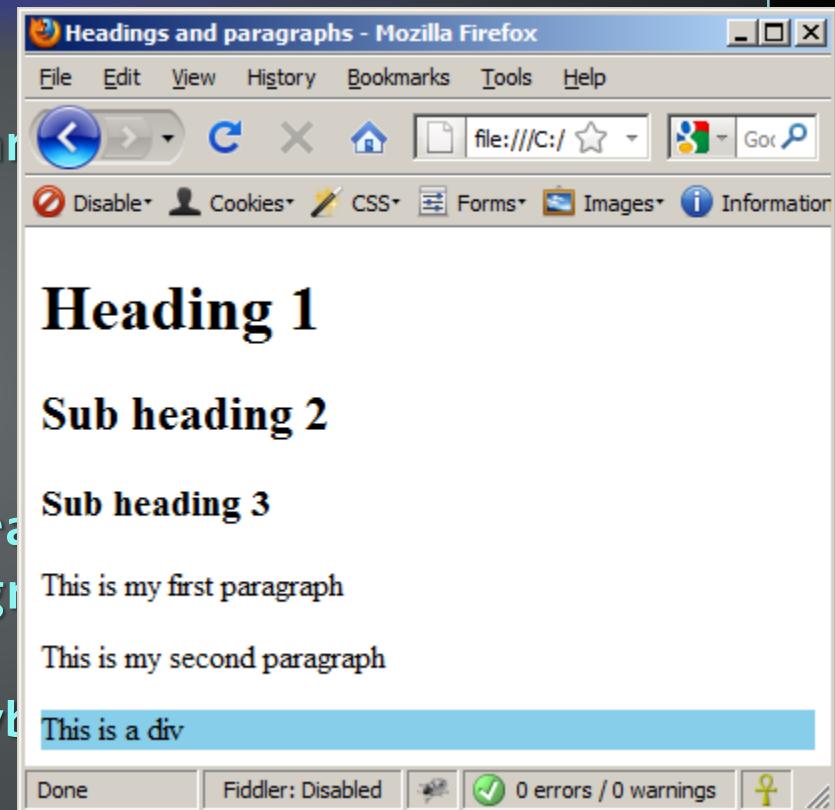
Headings and Paragraphs – Example (2)

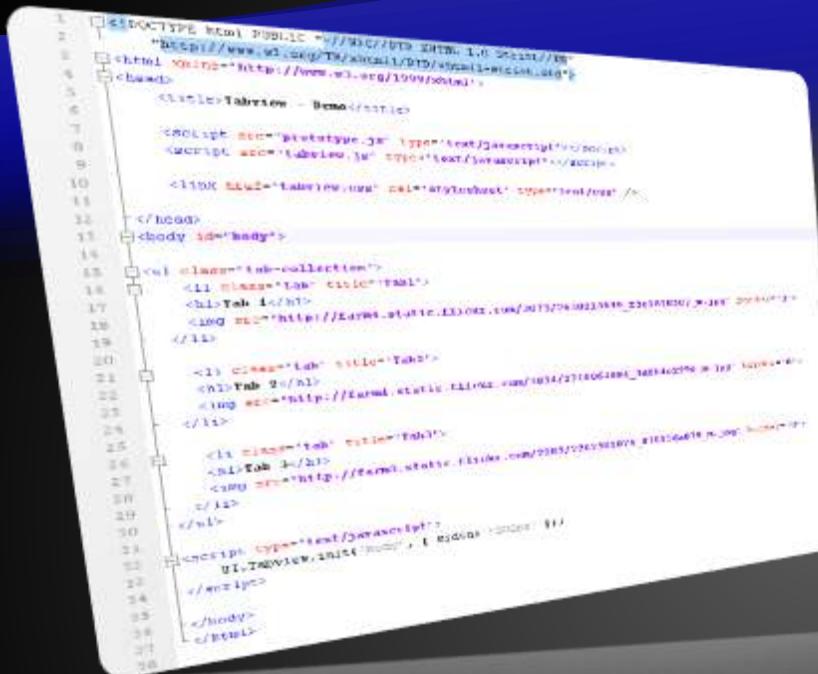
headings.html

```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and paragraphs</title>
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

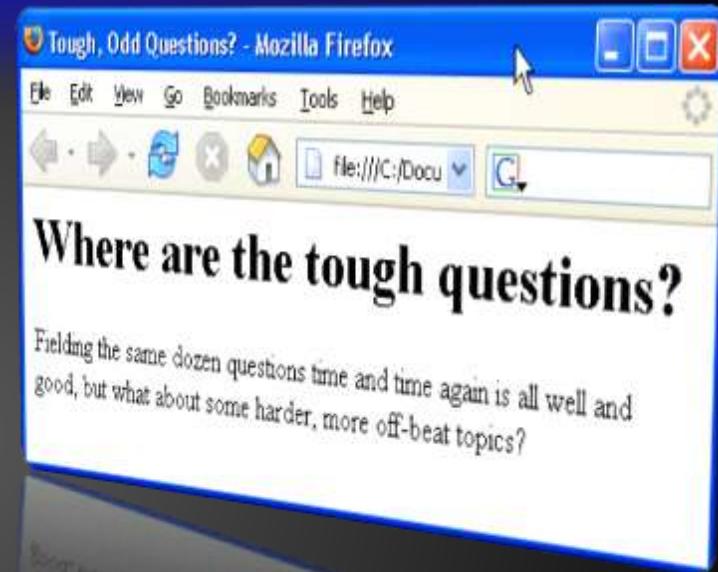
    <p>This is my first paragraph</p>
    <p>This is my second paragraph</p>

    <div style="background:skyblue">
      This is a div</div>
  </body>
</html>
```





```
1 <!DOCTYPE html PUBLIC "-//IETF//DTD HTML 2.0//EN">
2   "http://www.w3.org/1999/xhtml/xhtml-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4   <head>
5     <title>Telerik - Demo</title>
6
7     <script src="prototype.js" type="text/javascript"></script>
8     <script src="telerik.js" type="text/javascript"></script>
9
10    <link rel="stylesheet" href="style.css" type="text/css"/>
11
12  </head>
13  <body id="body">
14
15    <ul class="list-collection">
16      <li class="list-item">
17        <ul>
18          <li><a href="http://telerik-static.telerik.com/od/1526022388_2361800/2-18">2-18</a>
19        </li>
20
21        <li class="list-item"><a href="http://telerik-static.telerik.com/od/1526022388_3694629/3-17">3-17</a>
22        </li>
23      </ul>
24    </li>
25
26    <li class="list-item"><a href="http://telerik-static.telerik.com/od/1526022388_3694629/3-12">3-12</a>
27    </li>
28
29  </ul>
30
31  <script type="text/javascript">
32    UI.TableView.init();
33  </script>
34
35  </body>
36  </html>
```



Introduction to HTML

HTML Document Structure in Depth

- ◆ It is important to have the correct vision and attitude towards HTML
 - ◆ HTML is only about structure, not appearance
 - ◆ Browsers tolerate invalid HTML code and parse errors – you should not.



- ◆ XHTML is more strict than HTML
 - Tags and attribute names must be in lowercase
 - All tags must be closed (`
`, ``) while HTML allows `
` and ``
 - XHTML allows only one root `<html>` element (HTML allows more than one)

- ◆ Many element attributes are deprecated in XHTML, most are moved to CSS
- ◆ Attribute minimization is forbidden, e.g.

```
<input type="checkbox" checked>
```



```
<input type="checkbox" checked="checked" />
```

- ◆ Note: Web browsers load XHTML faster than HTML and valid code faster than invalid!

- ◆ Contains information that doesn't show directly on the viewable page
- ◆ Begins with <head> and ends with </head>
- ◆ Contains mandatory single <title> tag
- ◆ Can contain some other tags, e.g.
 - ◆ <meta>
 - ◆ <script>
 - ◆ <style>
 - ◆ <!-- comments -->

<head> Section: <title> tag

- ◆ Title should be placed between <head> and </head> tags

```
<title>Telerik Academy - Winter Season 2009/2010</title>
```



- ◆ Used to specify a title in the window title bar
- ◆ Search engines and people rely on titles

<head> Section: <meta>

- ◆ Meta tags additionally describe the content contained within the page

```
<meta name="description" content="HTML  
tutorial" />
```

```
<meta name="keywords" content="html, web  
design, styles" />
```

```
<meta name="author" content="Chris Brewer" />
```

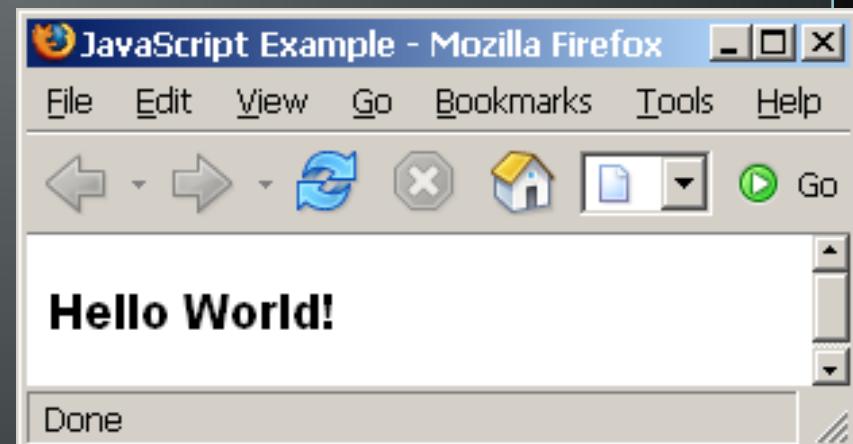
```
<meta http-equiv="refresh" content="5;  
url=http://www.telerik.com" />
```

<head> Section: <script>

- ◆ The <script> element is used to embed scripts into an HTML document
 - ◆ Script are executed in the client's Web browser
 - ◆ Scripts can live in the <head> and in the <body> sections
- ◆ Supported client-side scripting languages:
 - ◆ JavaScript (it is not Java!)
 - ◆ VBScript
 - ◆ JScript

The <script> Tag – Example

```
<!DOCTYPE HTML>                                         scripts-example.html
<html>
  <head>
    <title>JavaScript Example</title>
    <script type="text/javascript">
      function sayHello() {
        document.write("<p>Hello World!</p>");
      }
    </script>
  </head>
  <body>
    <script type=
      "text/javascript">
      sayHello();
    </script>
  </body>
</html>
```

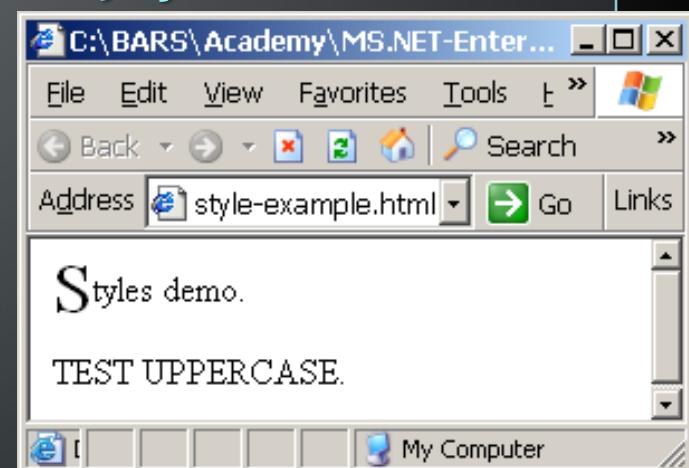


<head> Section: <style>

- ◆ The <style> element embeds formatting information (CSS styles) into an HTML page

```
<html>
  <head>
    <style type="text/css">
      p { font-size: 12pt; line-height: 12pt; }
      p:first-letter { font-size: 200%; }
      span { text-transform: uppercase; }
    </style>
  </head>
  <body>
    <p>Styles demo.<br />
      <span>Test uppercase</span>.
    </p>
  </body>
</html>
```

style-example.html



- ◆ Comments can exist anywhere between the `<html></html>` tags
- ◆ Comments start with `<!--` and end with `-->`

```
<!-- Telerik Logo (a JPG file) -->

<!-- Hyperlink to the web site -->
<a href="http://telerik.com/">Telerik</a>
<!-- Show the news table -->
<table class="newstable">
    ...

```

- ◆ The <body> section describes the viewable portion of the page
- ◆ Starts after the <head> </head> section
- ◆ Begins with <body> and ends with </body>

```
<html>
  <head><title>Test page</title></head>
  <body>
    <!-- This is the Web page body -->
  </body>
</html>
```

- ◆ Text formatting tags modify the text between the opening tag and the closing tag
 - ♦ Ex. **Hello** makes “Hello” bold

	bold
<i></i>	<i>italicized</i>
<u></u>	<u>underlined</u>
	Sample ^{superscript}
	Sample _{subscript}
	strong
	<i>emphasized</i>
<pre></pre>	Preformatted text
<blockquote></blockquote>	Quoted text block
	Deleted text – strike through

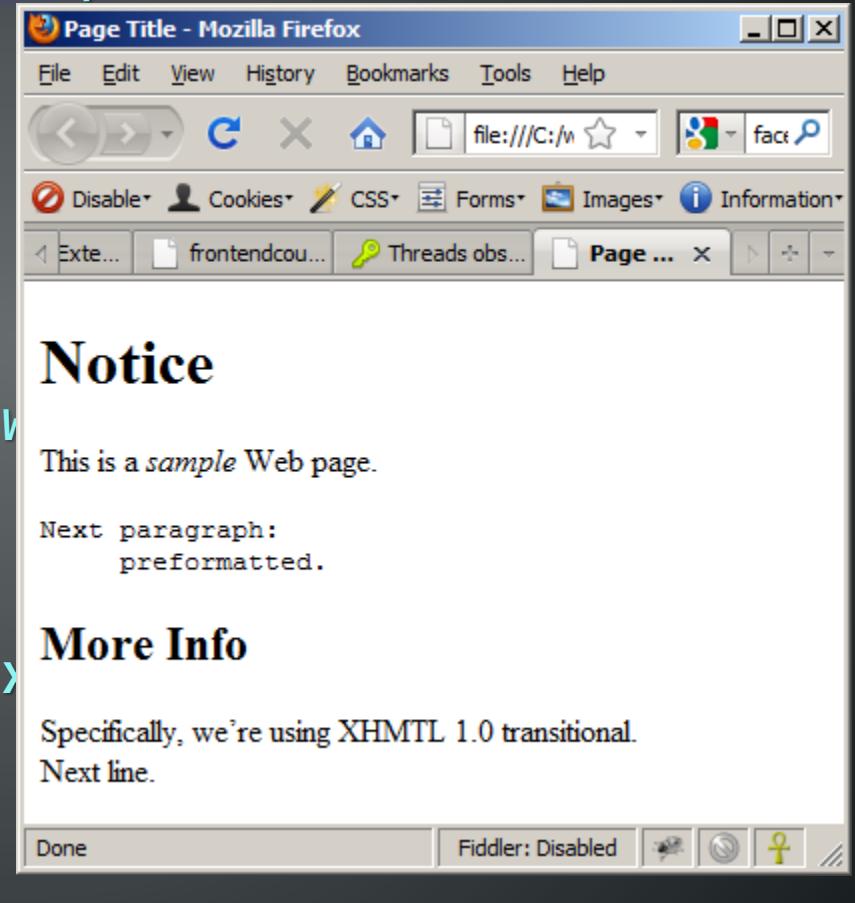
text-formatting.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Notice</h1>
    <p>This is a <em>sample</em> Web page.</p>
    <p><pre>Next paragraph:
      preformatted.</pre></p>
    <h2>More Info</h2>
    <p>Specifically, we're using XHTML 1.0 transitional.<br />
      Next line.</p>
  </body>
</html>
```

telerik Text Formatting – Example (2)

text-formatting.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Notice</h1>
    <p>This is a <em>sample</em> \n
    <p><pre>Next paragraph:
      preformatted.</pre></p>
    <h2>More Info</h2>
    <p>Specifically, we're using >
      Next line.</p>
  </body>
</html>
```



- ◆ Link to a document called `form.html` on the same server in the same directory:

```
<a href="form.html">Fill Our Form</a>
```

- ◆ Link to a document called `parent.html` on the same server in the parent directory:

```
<a href="../parent.html">Parent</a>
```

- ◆ Link to a document called `cat.html` on the same server in the subdirectory `stuff`:

```
<a href="stuff/cat.html">Catalog</a>
```

- ◆ Link to an external Web site:

```
<a href="http://www.devbg.org" target="_blank">BASD</a>
```

- ◆ Always use a full URL, including "http://", not just "www.somesite.com"
- ◆ Using the target="_blank" attribute opens the link in a new window
- ◆ Link to an e-mail address:

```
<a href="mailto:bugs@example.com?subject=Bug+Report">  
Please report bugs here (by e-mail only)</a>
```

Hyperlinks: <a> Tag (3)

- ◆ Link to a document called `apply-now.html`
 - On the same server, in same directory
 - Using an image as a link button:

```
<a href="apply-now.html"></a>
```

- ◆ Link to a document called `index.html`
 - On the same server, in the subdirectory `english` of the parent directory:

```
<a href="../english/index.html">Switch to  
English version</a>
```

- ◆ Link to another location in the same document:

```
<a href="#section1">Go to Introduction</a>
...
<h2 id="section1">Introduction</h2>
```

- ◆ Link to a specific location in another document:

```
<a href="chapter3.html#section3.1.1">Go to Section
3.1.1</a>

<!-- In chapter3.html -->
...
<div id="section3.1.1">
    <h3>3.1.1. Technical Background</h3>
</div>
```

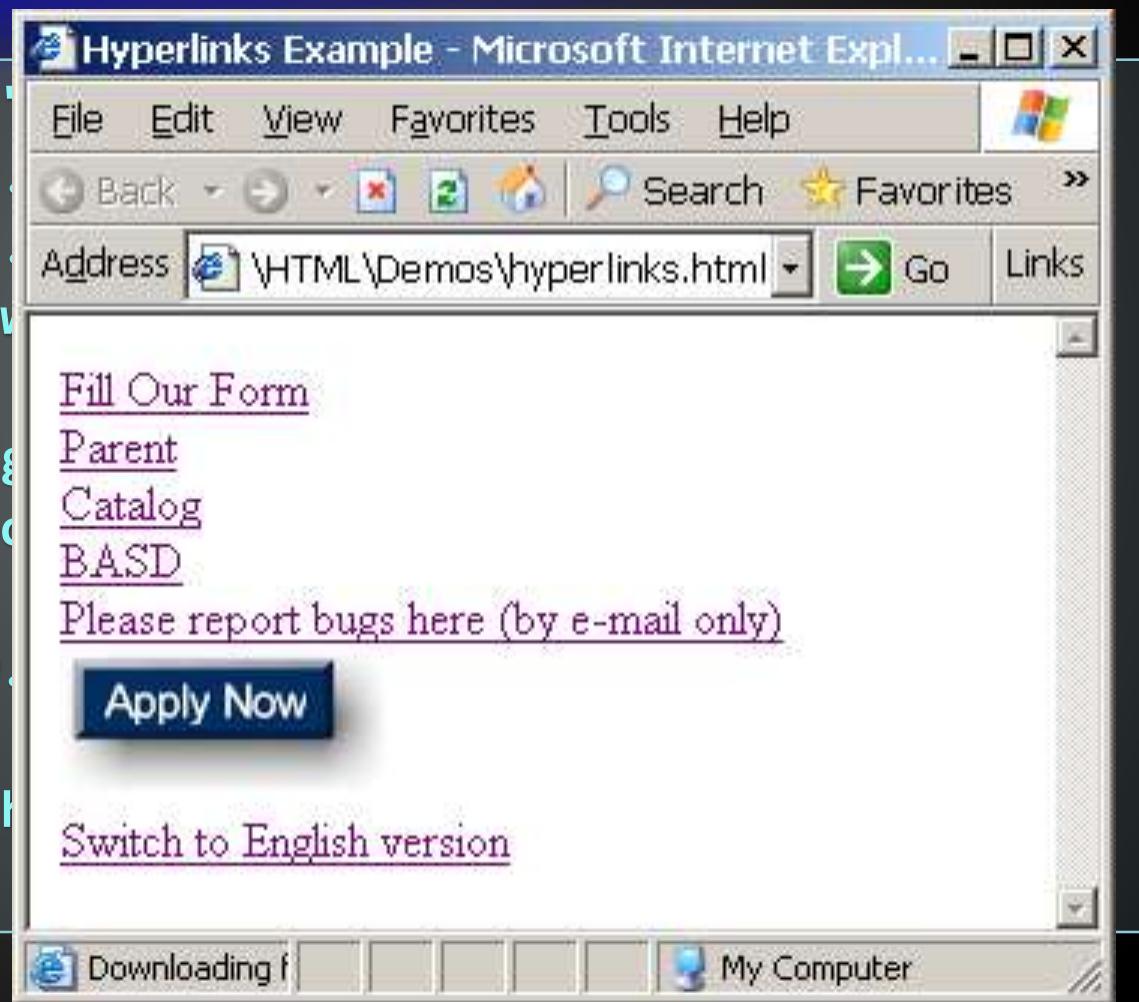
hyperlinks.html

```
<a href="form.html">Fill Our Form</a> <br />
<a href="../parent.html">Parent</a> <br />
<a href="stuff/cat.html">Catalog</a> <br />
<a href="http://www.devbg.org" target="_blank">BASD</a>
<br />
<a href="mailto:bugs@example.com?subject=Bug
Report">Please report bugs here (by e-mail only)</a>
<br />
<a href="apply-now.html"></a> <br />
<a href="../english/index.html">Switch to English
version</a> <br />
```

Hyperlinks – Example (2)

hyperlinks.html

```
<a href="form.html">
<a href="..../parent">
<a href="stuff/catalog">
<a href="http://www.basd.com">
<br />
<a href="mailto:bug-report@basd.com">Please report bugs here (by e-mail only)
<br />
<a href="apply-now.html">Apply Now</a> <br />
<a href="..../english-version">Switch to English version</a> <br />
```



links-to-same-document.html

```
<h1>Table of Contents</h1>

<p><a href="#section1">Introduction</a><br />
<a href="#section2">Some background</A><br />
<a href="#section2.1">Project History</a><br />
...the rest of the table of contents...

<!-- The document text follows here -->

<h2 id="section1">Introduction</h2>
... Section 1 follows here ...
<h2 id="section2">Some background</h2>
... Section 2 follows here ...
<h3 id="section2.1">Project History</h3>
... Section 2.1 follows here ...
```

links-to-same-document.html

```
<h1>Table
```

```
<p><a href="#>
```

```
<a href="#">
```

```
<a href="#">
```

```
...the rest
```

```
<!-- The -->
```

```
<h2 id="sec1">
```

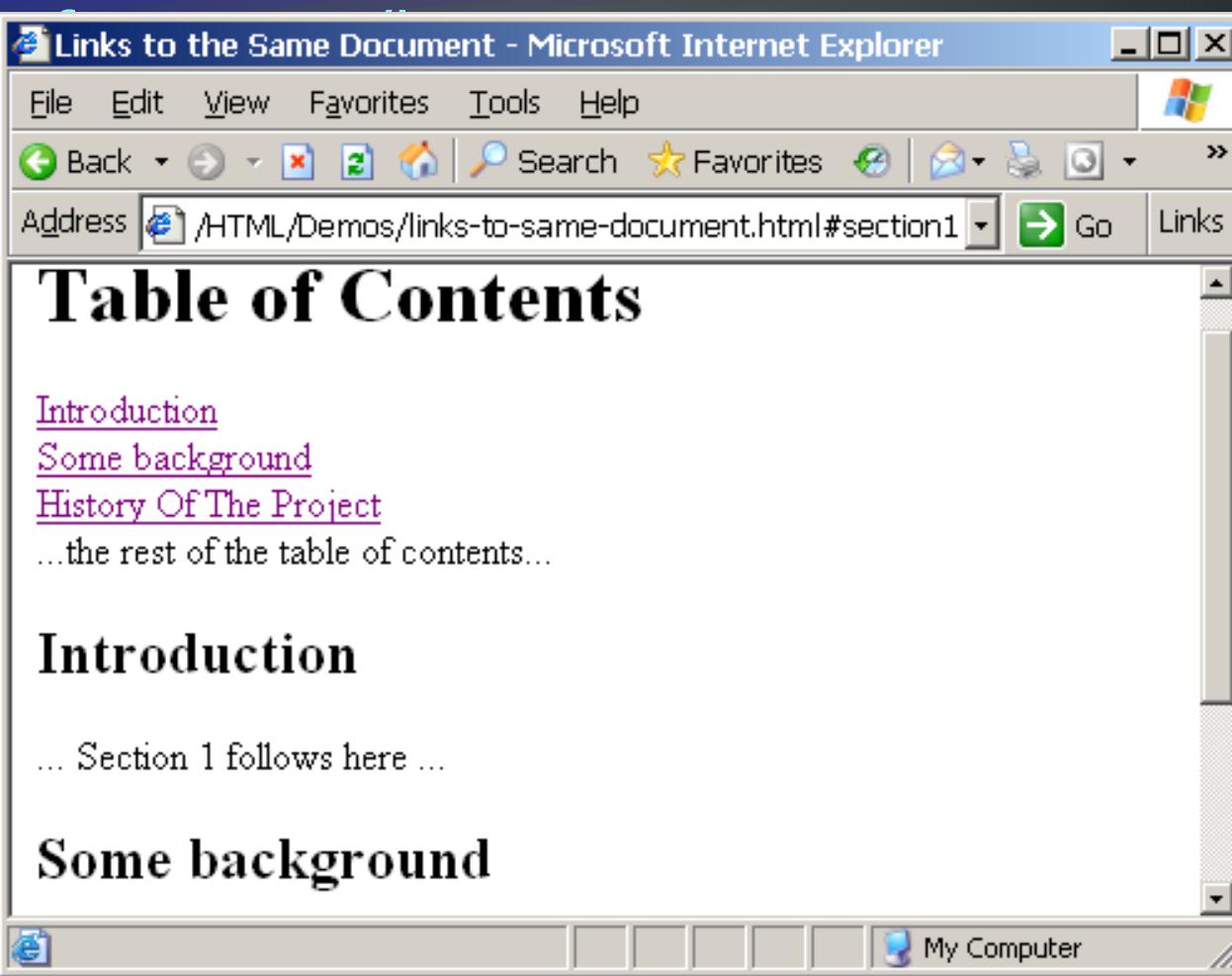
```
... Section 1
```

```
<h2 id="sec2">
```

```
... Section 2
```

```
<h3 id="sec3">
```

```
... Section 3
```



- ◆ Inserting an image with tag:

```

```

- ◆ Image attributes:

src	Location of image file (relative or absolute)
alt	Substitute text for display (e.g. in text mode)
height	Number of pixels of the height
width	Number of pixels of the width
border	Size of border, 0 for no border

- ◆ Example:

```

```

- ◆ <hr />: Draws a horizontal rule (line):

```
<hr size="5" width="70%" />
```

- ◆ <center></center>: Deprecated!

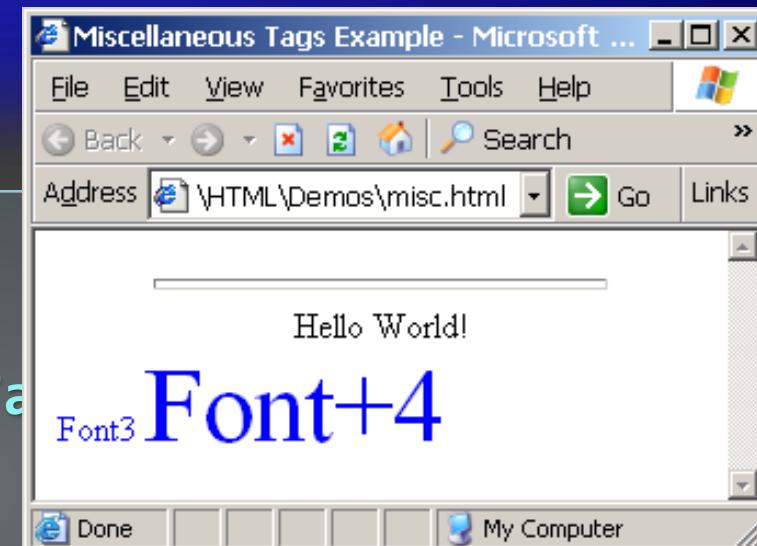
```
<center>Hello World!</center>
```

- ◆ : Deprecated!

```
<font size="3" color="blue">Font3</font>
<font size="+4" color="blue">Font+4</font>
```

misc.html

```
<html>
  <head>
    <title>Miscellaneous Tag Example</title>
  </head>
  <body>
    <hr size="5" width="70%" />
    <center>Hello World!</center>
    <font size="3" color="blue">Font3</font>
    <font size="+4" color="blue">Font+4</font>
  </body>
</html>
```



- ◆ Create an Ordered List using :

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>
```

- ◆ Attribute values for type are 1, A, a, I, or i

1. Apple
2. Orange
3. Grapefruit

A. Apple
B. Orange
C. Grapefruit

a. Apple
b. Orange
c. Grapefruit

I. Apple
II. Orange
III. Grapefruit

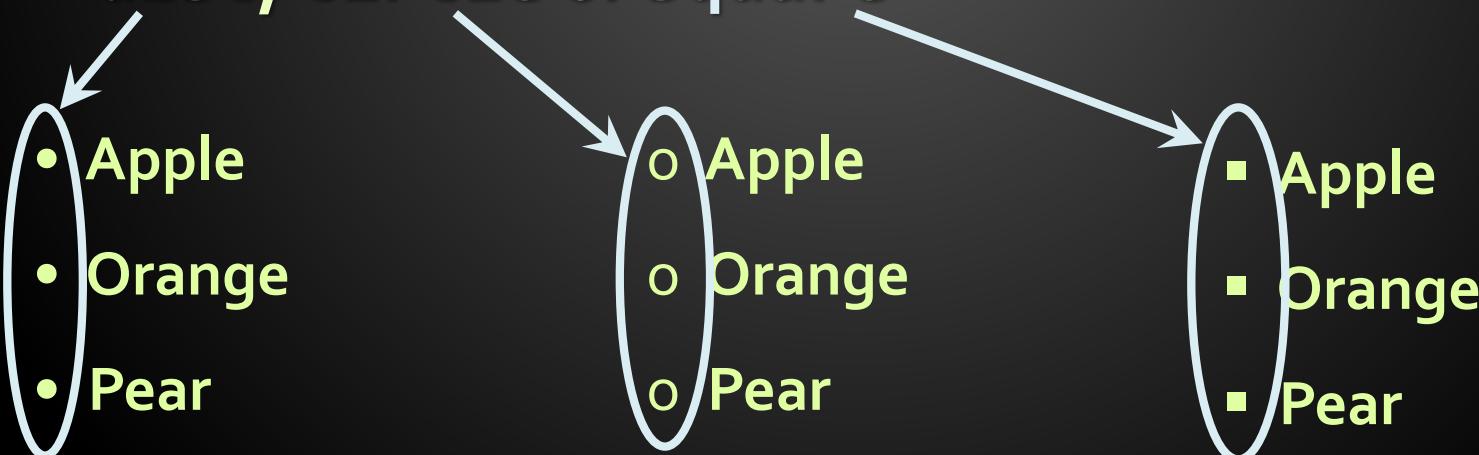
i. Apple
ii. Orange
iii. Grapefruit

- ◆ Create an Unordered List using :

```
<ul type="disk">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>
```

- ◆ Attribute values for type are:

- ◆ disc, circle or square



- ◆ Create definition lists using <dl>
 - ◆ Pairs of text and associated definition; text is in <dt> tag, definition in <dd> tag

```
<dl>
  <dt>HTML</dt>
  <dd>A markup language ...</dd>
  <dt>CSS</dt>
  <dd>Language used to ...</dd>
</dl>
```

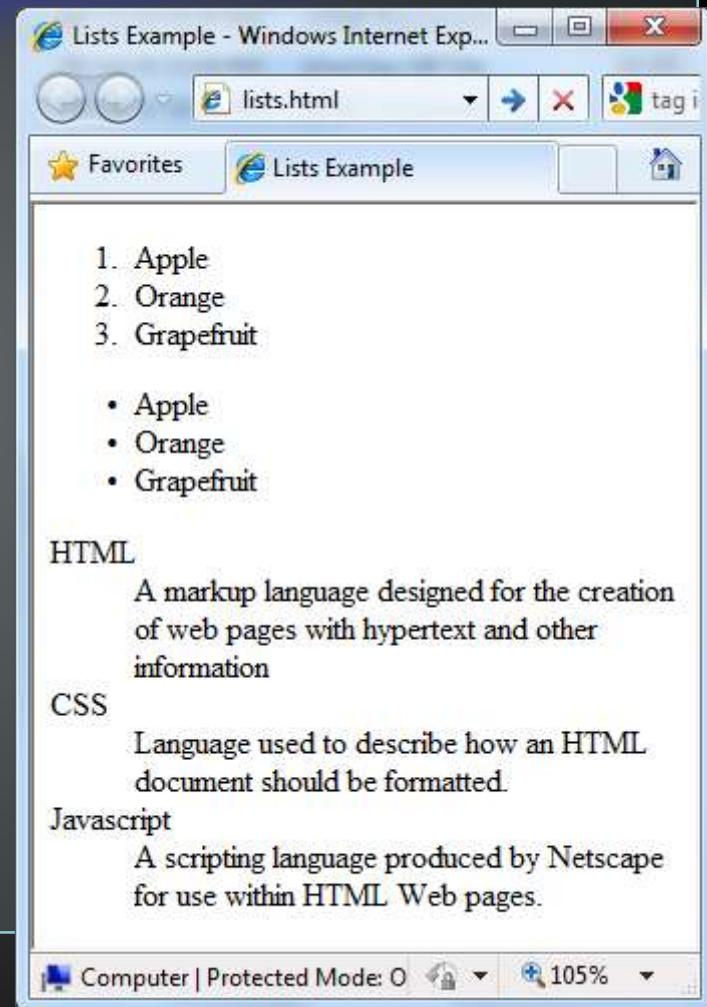
- ◆ Renders without bullets
- ◆ Definition is indented

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>

<ul type="disc">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>

<dl>
  <dt>HTML</dt>
  <dd>A markup lang...</dd>
</dl>
```

lists.html



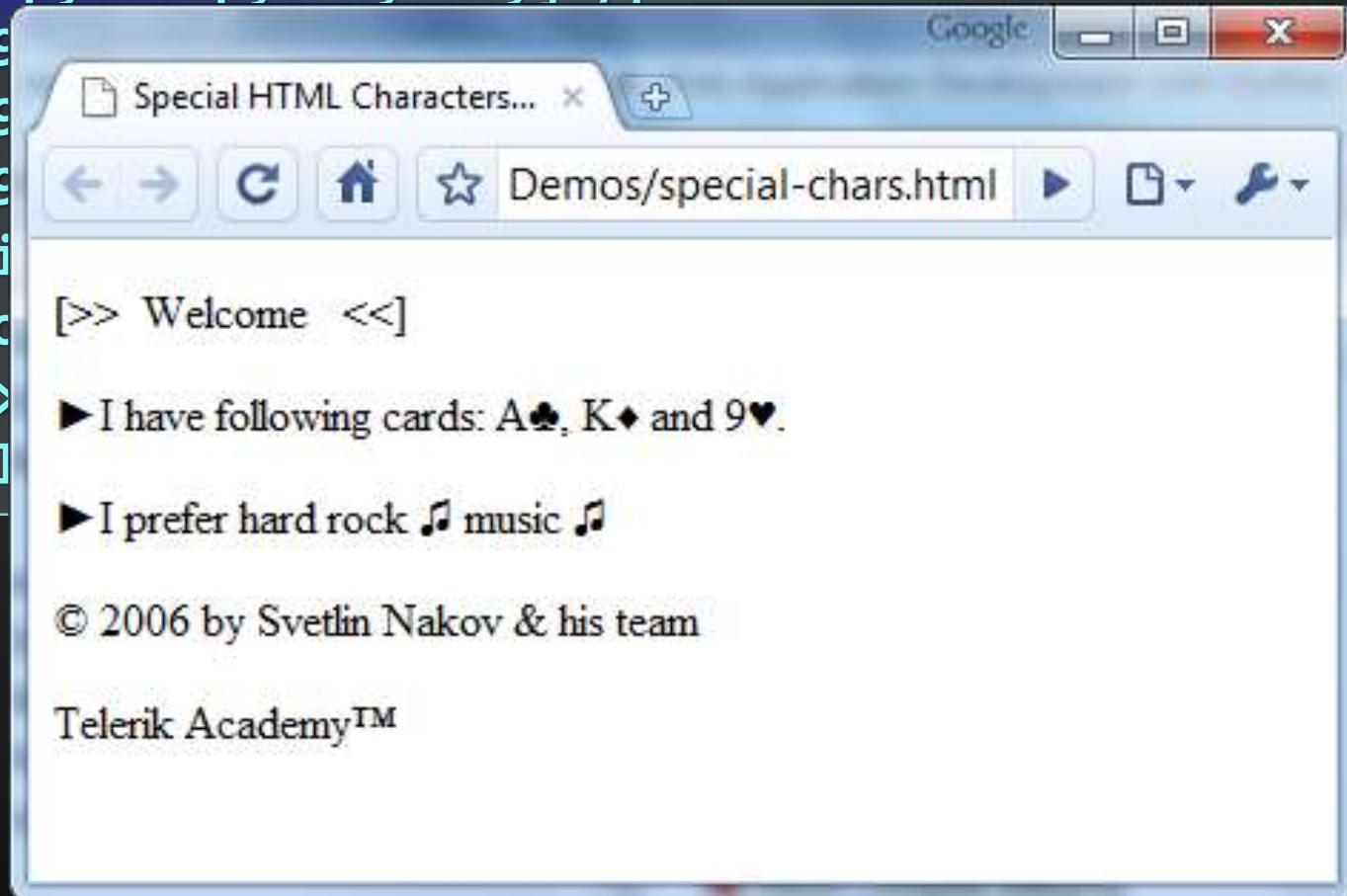
HTML Special Characters

Symbol Name	HTML Entity	Symbol
Copyright Sign	©	©
Registered Trademark Sign	®	®
Trademark Sign	™	™
Less Than	<	<
Greater Than	>	>
Ampersand	&	&
Non-breaking Space	 	
Em Dash	—	—
Quotation Mark	"	"
Euro	€	€
British Pound	£	£
Japanese Yen	¥	¥

```
<p>[&gt;&gt;&ampnbsp&ampnbspWelcome    special-chars.html  
     &ampnbsp&ampnbsp&lt;&lt;]</p>  
<p>I have following cards:  
    A, K and 9.</p>  
<p>I prefer hard rock ;  
    music ;</p>  
<p> 2006 by Svetlin Nakov & his  
team</p>  
<p>Telerik Academy</p>
```

Special Chars – Example (2)

```
<p>[&gt;&gt;&ampnbsp&ampnbspWelcome    special-chars.html  
     &ampnbsp&ampnbsp&lt;&lt;]</p>  
<p>&#9824;  
A&#9824;  
<p>&#9824;  
musi  
<p>&cc  
team</p>  
<p>Te]
```



```
4 <head>
5 <meta http-equiv="Content-Type"
6 <title>Home</title>
7 <link rel="stylesheet" href="sty
8 <style type="text/css">
9 .style1 {
10   color: #FF0000;
11 }
12 </style>
13 </head>
```

```
<span class="style1">You will have to buy a separate license to use the OpenCube 3D library. </span>
```



Using <DIV> and Block and Inline Elements

- ◆ Block elements add a line break before and after them
 - ◆ <div> is a block element
 - ◆ Other block elements are <table>, <hr>, headings, lists, <p> and etc.
- ◆ Inline elements don't break the text before and after them
 - ◆ is an inline element
 - ◆ Most HTML elements are inline, e.g. <a>

- ◆ <div> creates logical divisions within a page
- ◆ Block style element
- ◆ Used with CSS
- ◆ Example:

div-and-span.html

```
<div style="font-size:24px; color:red">DIV  
example</div>  
  
<p>This one is <span style="color:red; font-  
weight:bold">only a test</span>.</p>
```



- ◆ Inline style element
- ◆ Useful for modifying a specific portion of text
 - Don't create a separate area (paragraph) in the document
- ◆ Very useful with CSS

span.html

```
<p>This one is <span style="color:red; font-weight:bold">only a test</span>.</p>  
<p>This one is another <span style="font-size:32px; font-weight:bold">TEST</span>.</p>
```



US date	European date (D-M-Y) & time	S-M-D date & time	Dollar	Chinese money	IP address	Names	Numbers
29/10/1985		03-01-28	\$148.4	98.176.24.30		16.32 E+03	
Fri, Mar 22 21:48:49 UTC+0200 1987		1987-03-22 06:07:16 PM	\$148.413E	162.117.253.34	dave chik		
Fri, 14 Feb 2002 04:24:20 UTC	06-07-09 06:48:01 AM	03-02-28 09:09:44 AM	\$148.408E	122.205.30.6	hichard daniel	-191.45E-09	
Monday, Mar 26, 1991 04:21:29E	06/09/05 05:11:18 AM			\$148.33.16	dsty bala	131.20E-01	
29/10/2000	24/11/1987		\$148.77	1546 112.42	15.392.191.209		
		07-08-13 00:00:35 AM	\$14.5	148.1.14	99.99.147.150	dychai moshai	-187.26E-05
		07-10-18	\$14.66	1546 61.14		chik matic	-128.19E-03
Sat, 9 Jan 1982 05:45:06 UTC	04/06/88	04-10-20	\$20.47		121.169.225.22	dyna force	138.11E+02
04/06/88		2001-03-29	\$68.34	1546 98.19	239.133.227.68	east lewe	198.44 E+03
Monday, Mar 15, 2002 10:51:02 AM	01/02/1981 09:48:18 AM		\$97.9	1546 44.28	223.566.228.114	extra wie	-107
data footer	now	marked	0.0E0	adz	adz	adz	adz

```

<html>
<head>
<title>How To Create HTML Tables</title>
</head>
<body>
<table border=1 cellspacing=0 cellpadding=0>
<tr>
<td width=110 valign=top>
<br><upper left corner>
</td>
<td width=110 valign=top>
<br><upper right corner>
</td>
</tr>
<tr>
<td width=110 valign=top>
<br><left center cell>
</td>
<td width=110 valign=top>
<br><right center cell>
</td>
</tr>
<tr>
<td width=110 valign=top>
<br><lower left corner>
</td>
<td width=110 valign=top>
<br><lower right corner>
</td>
</tr>
</table>
</body>
</html>

```

HTML Tables

Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data

- ◆ Tables represent tabular data
 - ◆ A table consists of one or several rows
 - ◆ Each row has one or more columns
- ◆ Tables comprised of several core tags:
 - <table></table>: begin / end the table
 - <tr></tr>: create a table row
 - <td></td>: create tabular data (cell)
- ◆ Tables should not be used for layout. Use CSS floats and positioning styles instead

- ◆ Start and end of a table

```
<table> ... </table>
```

- ◆ Start and end of a row

```
<tr> ... </tr>
```

- ◆ Start and end of a cell in a row

```
<td> ... </td>
```

Simple HTML Tables – Example

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2-demos.zip">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```

telerik Simple HTML Tables – Example (2)

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td></td>
    <td><a href="lecture2-demos">
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```



- ◆ Table data “cells” (<td>) can contain nested tables (tables within tables):

```
<table>
  <tr>
    <td>Contact:</td>
    <td>
      <table>
        <tr>
          <td>First Name</td>
          <td>Last Name</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```

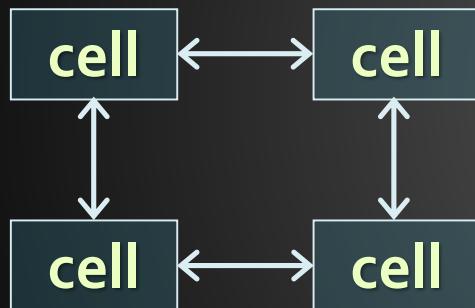
nested-tables.html



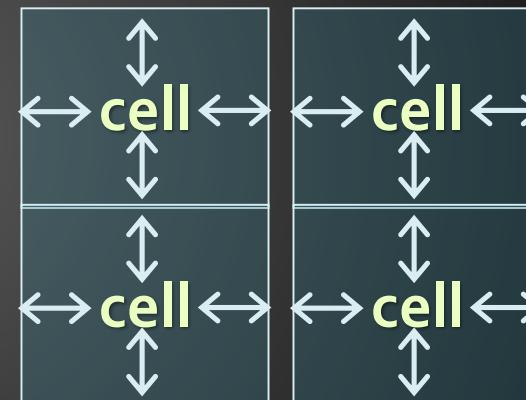
Cell Spacing and Padding

- ◆ Tables have two important attributes:

- ◆ cellspacing



- ◆ cellpadding



- ◆ Defines the empty space between cells

- ◆ Defines the empty space around the cell content

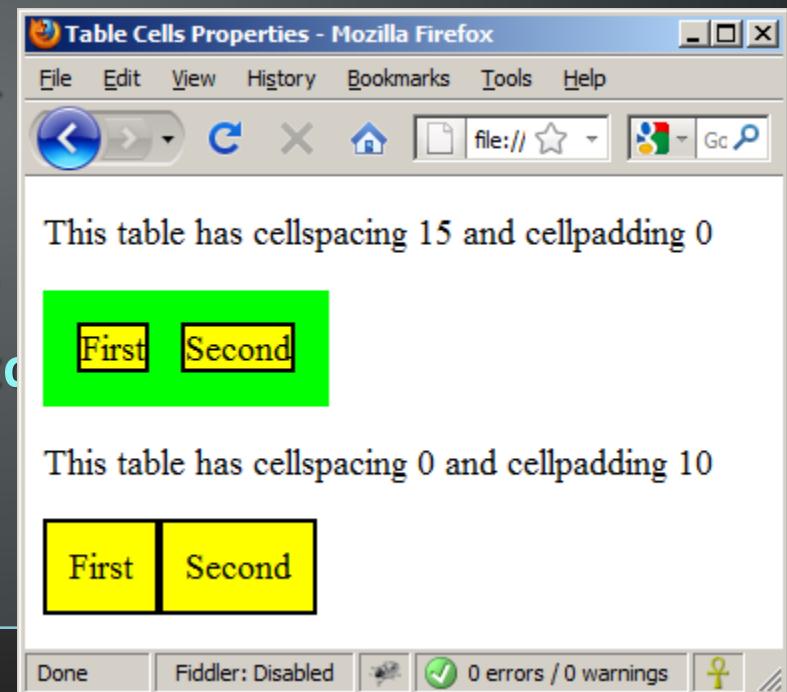
table-cells.html

```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15" cellpadding="0">
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0" cellpadding="10">
      <tr><td>First</td><td>Second</td></tr>
    </table>
  </body>
</html>
```

Cell Spacing and Padding – Example (2)

table-cells.html

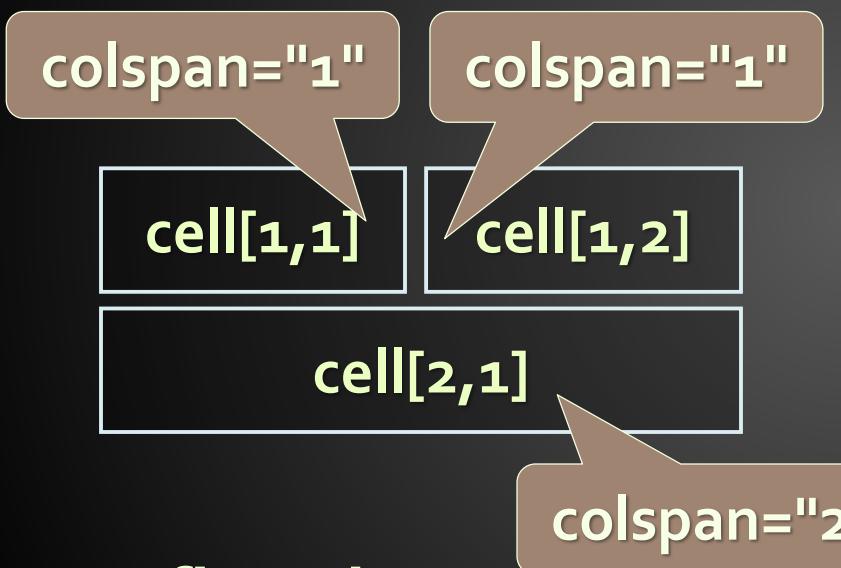
```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15" cellpadding="0">
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0" cellpadding="10">
      <tr><td>First</td><td>Second</td></tr>
    </table>
  </body>
</html>
```



Column and Row Span

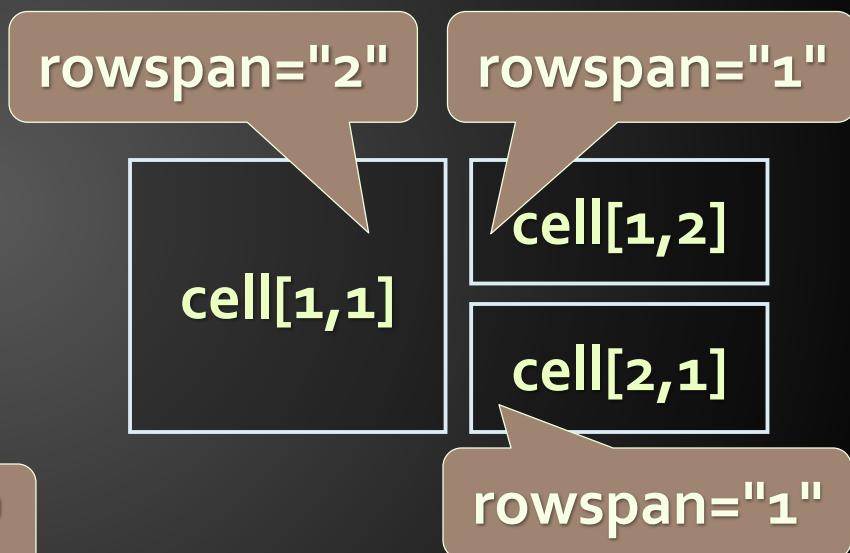
- ◆ Table cells have two important attributes:

- ◆ **colspan**



- ◆ Defines how many columns the cell occupies

- ◆ **rowspan**



- ◆ Defines how many rows the cell occupies

table-colspan-rowspan.html

```
<table cellspacing="0">
  <tr class="1"><td>Cell[1,1]</td>
    <td colspan="2">Cell[2,1]</td></tr>
  <tr class="2"><td>Cell[1,2]</td>
    <td rowspan="2">Cell[2,2]</td>
    <td>Cell[3,2]</td></tr>
  <tr class="3"><td>Cell[1,3]</td>
    <td>Cell[2,3]</td></tr>
</table>
```

Column and Row Span – Example (2)

table-colspan-rowspan.html

```
<table cellspacing="0">
  <tr class="1"><td>Cell[1,1]</td>
    <td colspan="2">Cell[2,1]</td></tr>
  <tr class="2"><td>Cell[1,2]</td>
    <td rowspan="2">Cell[2,2]</td>
    <td>Cell[3,2]</td></tr>
  <tr class="3">
    <td>Cell[1,3]</td>
  </tr>
</table>
```

Cell[1,1]	Cell[2,1]
Cell[1,2]	Cell[2,2]
Cell[1,3]	Cell[2,3]



HTML Forms

Entering User Data from a Web Page



- ◆ Forms are the primary method for gathering data from site visitors
- ◆ Create a form block with

```
<form></form>
```

The "method" attribute tells how the form data should be sent – via GET or POST request

- ◆ Example:

```
<form name="myForm" method="post"  
action="path/to/some-script.php">  
...  
</form>
```

The "action" attribute tells where the form data should be sent

- ◆ Single-line text input fields:

```
<input type="text" name="FirstName" value="This  
is a text field" />
```

- ◆ Multi-line textarea fields:

```
<textarea name="Comments">This is a multi-line  
text field</textarea>
```

- ◆ Hidden fields contain data not shown to the user:

```
<input type="hidden" name="Account" value="This  
is a hidden text field" />
```

- ◆ Often used by JavaScript code

- ◆ Fieldsets are used to enclose a group of related form fields:

```
<form method="post" action="form.aspx">
  <fieldset>
    <legend>Client Details</legend>
    <input type="text" id="Name" />
    <input type="text" id="Phone" />
  </fieldset>
  <fieldset>
    <legend>Order Details</legend>
    <input type="text" id="Quantity" />
    <textarea cols="40" rows="10"
              id="Remarks"></textarea>
  </fieldset>
</form>
```

- ◆ The **<legend>** is the fieldset's title.

- ◆ Checkboxes:

```
<input type="checkbox" name="fruit"  
value="apple" />
```

- ◆ Radio buttons:

```
<input type="radio" name="title" value="Mr." />
```

- ◆ Radio buttons can be grouped, allowing only one to be selected from a group:

```
<input type="radio" name="city" value="Lom" />  
<input type="radio" name="city" value="Ruse" />
```

- ◆ Dropdown menus:

```
<select name="gender">
    <option value="Value 1"
        selected="selected">Male</option>
    <option value="Value 2">Female</option>
    <option value="Value 3">Other</option>
</select>
```

- ◆ Submit button:

```
<input type="submit" name="submitBtn"
value="Apply Now" />
```

- ◆ **Reset button** – brings the form to its initial state

```
<input type="reset" name="resetBtn"  
value="Reset the form" />
```

- ◆ **Ordinary button** – used for Javascript, no default action

```
<input type="button" value="click me" />
```

- ◆ Password input – a text field which masks the entered text with * signs

```
<input type="password" name="pass" />
```

- ◆ Multiple select field – displays the list of items in multiple lines, instead of one

```
<select name="products" multiple="multiple">
    <option value="Value 1"
        selected="selected">keyboard</option>
    <option value="Value 2">mouse</option>
    <option value="Value 3">speakers</option>
</select>
```

- ◆ File input – a field used for uploading files

```
<input type="file" name="photo" />
```

- ◆ When used, it requires the form element to have a specific attribute:

```
<form enctype="multipart/form-data">  
...  
  <input type="file" name="photo" />  
...  

```

- ◆ Form labels are used to associate an explanatory text to a form field using the field's ID.

```
<label for="fn">First Name</label>
<input type="text" id="fn" />
```

- ◆ Clicking on a label focuses its associated field (checkboxes are toggled, radio buttons are checked)

form.html

```
<form method="post" action="apply-now.php">
    <input name="subject" type="hidden" value="Class" />
    <fieldset><legend>Academic information</legend>
        <label for="degree">Degree</label>
        <select name="degree" id="degree">
            <option value="BA">Bachelor of Art</option>
            <option value="BS">Bachelor of Science</option>
            <option value="MBA" selected="selected">Master of
                Business Administration</option>
        </select>
        <br />
        <label for="studentid">Student ID</label>
        <input type="password" name="studentid" />
    </fieldset>
    <fieldset><legend>Personal Details</legend>
        <label for="fname">First Name</label>
        <input type="text" name="fname" id="fname" />
        <br />
        <label for="lname">Last Name</label>
        <input type="text" name="lname" id="lname" />
```

form.html (continued)

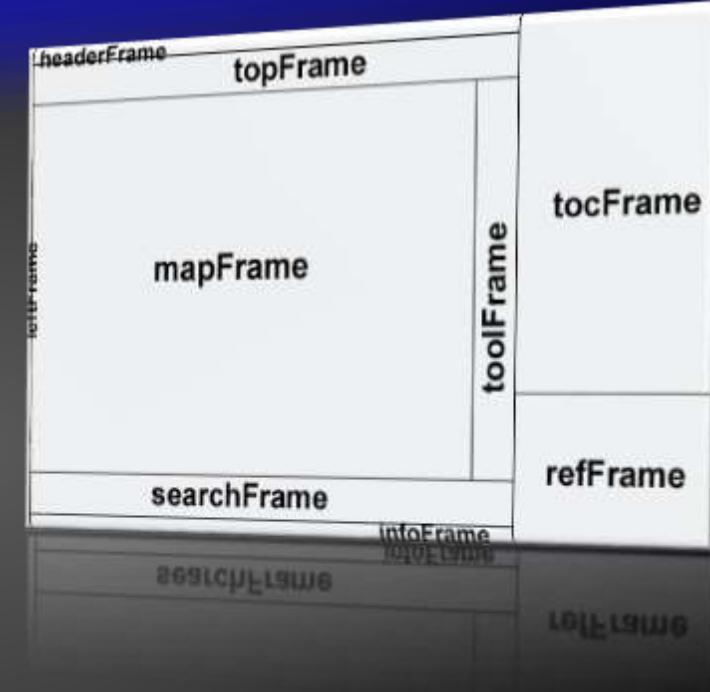
```
<br />
Gender:
<input name="gender" type="radio" id="gm" value="m" />
<label for="gm">Male</label>
<input name="gender" type="radio" id="gf" value="f" />
<label for="gf">Female</label>
<br />
<label for="email">Email</label>
<input type="text" name="email" id="email" />
</fieldset>
<p>
<textarea name="terms" cols="30" rows="4"
readonly="readonly">TERMS AND CONDITIONS...</textarea>
</p>
<p>
<input type="submit" name="submit" value="Send Form" />
<input type="reset" value="Clear Form" />
</p>
</form>
```

HTML Forms – Example (3)

form.html (continued)

The screenshot shows a Mozilla Firefox browser window displaying an HTML form. The title bar reads "HTML Forms Example - Mozilla Firefox". The address bar shows "file:///C:/work/Di".
Academic information
Degree: Master of Business Administration
Student ID:
Classes attended: (with options: Geography, Mathematics, English)
Personal Details
First Name:
Last Name:
Gender: Male Female
Email:
TERMS AND CONDITIONS...

At the bottom, there are two buttons: "Send Form" and "Clear Form".
At the very bottom of the browser window, there are status bars: "Done", "Fiddler: Disabled", "0 errors / 0 warnings", and a yellow icon.



HTML Frames

<frameset> and <frame>

- ◆ Frames provide a way to show multiple HTML documents in a single Web page
- ◆ The page can be split into separate views (frames) horizontally and vertically
- ◆ Frames were popular in the early ages of HTML development, but now their usage is rejected

frames.html

```
<html>

<head><title>Frames Example</title></head>

<frameset cols="180px,* ,150px">
  <frame src="left.html" />
  <frame src="middle.html" />
  <frame src="right.html" />
</frameset>

</html>
```

The screenshot displays a browser window with a frameset. The left frame contains the HTML code for the current page. The middle frame shows a 404 error page with the title "404 - The page cannot be found". The right frame also shows a 404 error page with the same title.

Left Frame Content:

```
<html>
<head><title>Frames Example</title></head>
<frameset cols="180px,* ,150px">
  <frame src="left.html" />
  <frame src="middle.html" />
  <frame src="right.html" />
</frameset>
</html>
```

Middle Frame Content:

404 - The page cannot be found

Sorry, we cannot find the page you are looking for.

It might have been removed, had its name changed, or is temporarily unavailable.

Please check that the Web site address is spelled correctly.

Or go to our [home page](#) and use the menus to navigate to a specific section.

Other things to try:

- Go to www.w3schools.com/tags/
- Search www.w3schools.com:

tags middle

Google Search

Right Frame Content:

404 - The page cannot be found

Sorry, we cannot find the page you are looking for.

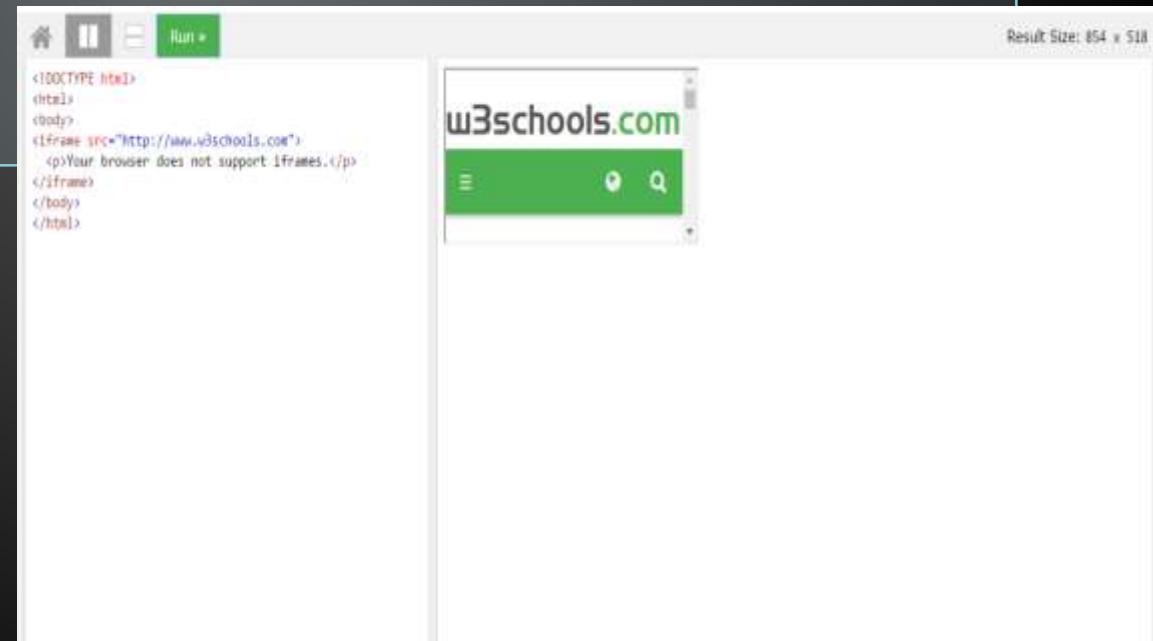
It might have been removed, had its name changed, or is temporarily unavailable.

Please check that the Web site address is spelled correctly.

Or go to our [home page](#) and use the menus

iframes.html

```
<!DOCTYPE html>
<html>
<body>
<iframe src="http://www.w3schools.com">
    <p>Your browser does not support iframes.</p>
</iframe>
</body>
</html>
```





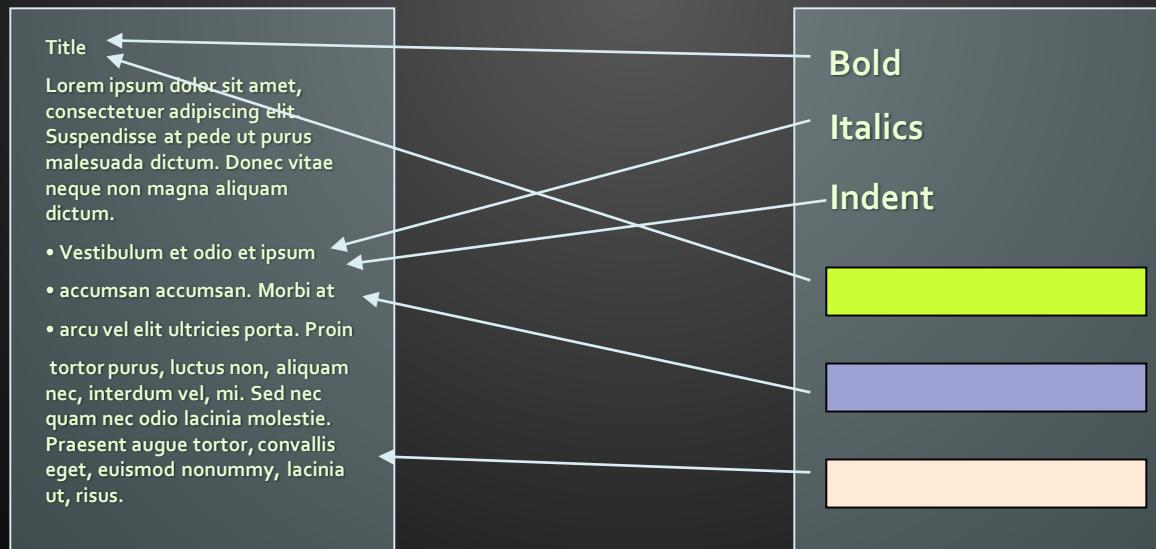
Cascading Style Sheets (CSS)

```
171 #content .article img.left.border {
172     padding: 0 9px 9px 0;
173     border-right: 1px dotted #999;
174     border-bottom: 1px dotted #999; }
175 #content .article blockquote {
176     margin-left: 10px;
177     padding-left: 10px;
178     border-left: 3px solid #252525; }
179 #content .article ul {
180     padding-left: 1em;
181     list-style-type: circle; }
```

◆ Separate content from presentation!

Content
(HTML document)

Presentation
(CSS Document)



Title

Consectetuer adipiscing elit.
Suspendisse at pede ut purus malesuada dictum. Donec vitae neque non magna aliquam dictum.

- *Vestibulum et odio et ipsum*
- *accumsan accumsan. Morbi at*
- *arcu vel elit ultricies porta. Proin*

Tortor purus, luctus non, aliquam nec, interdum vel, mi. Sed nec quam nec odio lacinia molestie. Praesent augue tortor, convallis eget, euismod nonummy, lacinia ut, risus.



CSS Intro

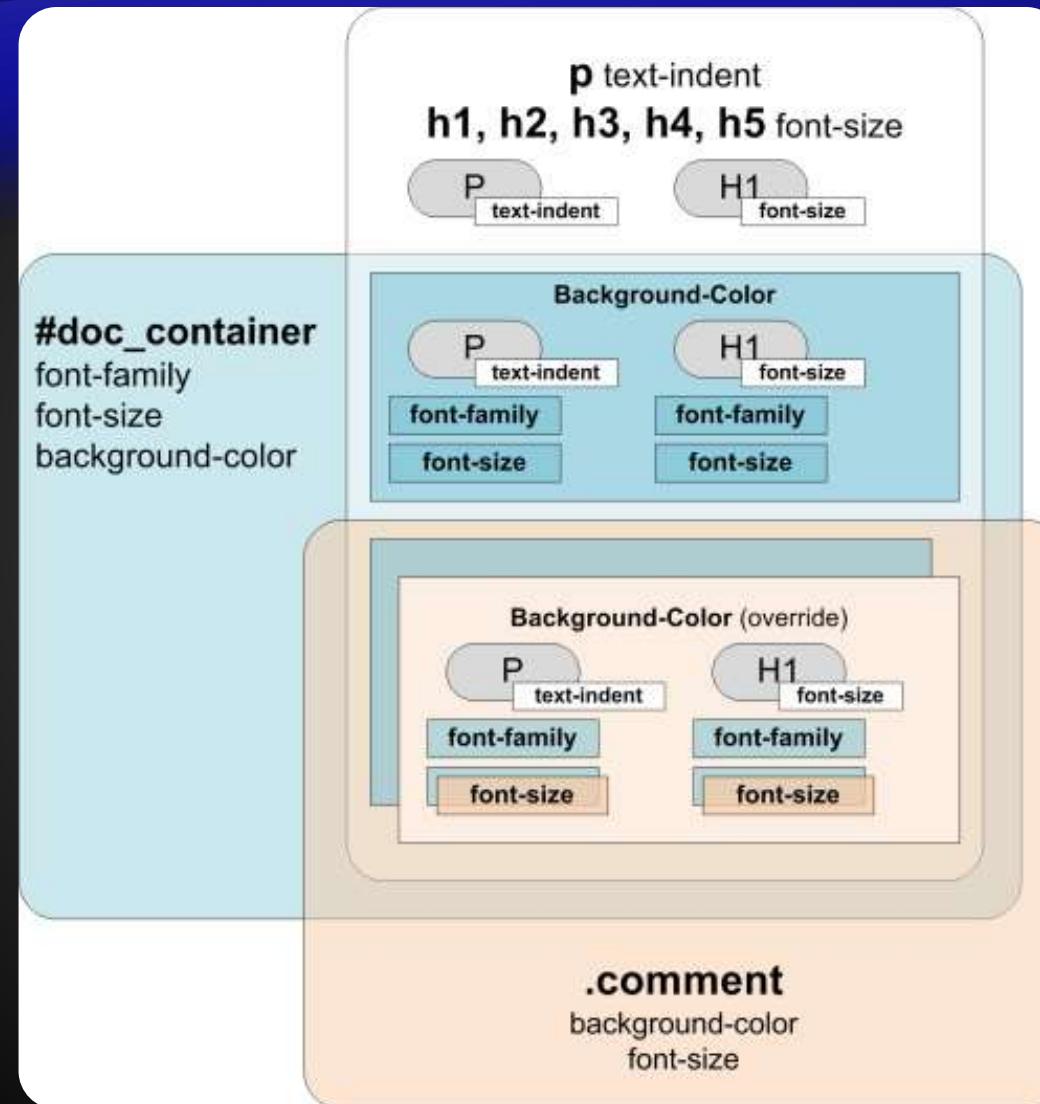
Styling with Cascading Stylesheets

- ◆ Cascading Style Sheets (CSS)
 - Used to describe the presentation of documents
 - Define sizes, spacing, fonts, colors, layout, etc.
 - Improve content accessibility
 - Improve flexibility
- ◆ Designed to separate presentation from content
- ◆ Due to CSS, all HTML presentation tags and attributes are deprecated, e.g. font, center, etc.

Why “Cascading”?

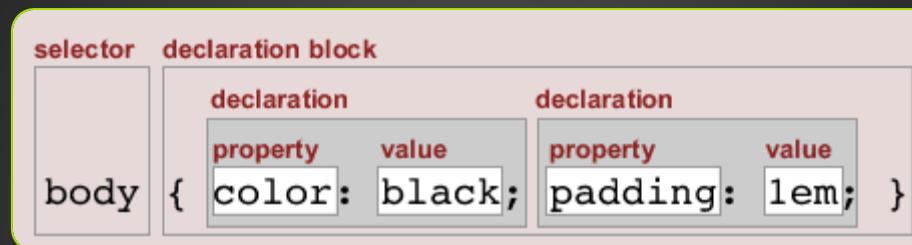
- ◆ Priority scheme determining which style rules apply to element
 - ◆ Cascade priorities or specificity (weight) are calculated and assigned to the rules
 - ◆ Child elements in the HTML DOM tree inherit styles from their parent
 - ◆ Can override them

Why “Cascading”? (2)



- ◆ Some CSS styles are inherited and some not
 - Text-related and list-related properties are inherited - **color, font-size, font-family, line-height, text-align, list-style, etc**
 - Box-related and positioning styles are not inherited - **width, height, border, margin, padding, position, float, etc**
 - **<a> elements do not inherit color and text-decoration**

- ◆ Stylesheets consist of rules, selectors, declarations, properties and values



- ◆ Selectors are separated by commas
- ◆ Declarations are separated by semicolons
- ◆ Properties and values are separated by colons

```
h1,h2,h3 { color: green; font-weight: bold; }
```

- ◆ Selectors determine which element the rule applies to:
 - All elements of specific type (tag)
 - Those that match a specific attribute (id, class)
 - Elements may be matched depending on how they are nested in the document tree (HTML)
- ◆ Examples:

```
.header a { color: green }
```

```
#menu>li { padding-top: 8px }
```

- ◆ Three primary kinds of selectors:

- By tag (type selector):

```
h1 { font-family: verdana, sans-serif; }
```

- By element id:

```
#element_id { color: #ff0000; }
```

- By element class name (only for HTML):

```
.myClass { border: 1px solid red }
```

- ◆ Selectors can be combined with commas:

```
h1, .link, #top-link { font-weight: bold }
```

This will match `<h1>` tags, elements with class link, and element with id top-link

- ◆ Pseudo-classes define state
 - ◆ :hover, :visited, :active

- ◆ HTML (content) and CSS (presentation) can be linked in three ways:
 - ◆ Inline: the CSS rules in the `style` attribute
 - ◆ No selectors are needed
 - ◆ Embedded: in the `<head>` in a `<style>` tag
 - ◆ External: CSS rules in separate file (best)
 - ◆ Usually a file with `.css` extension
 - ◆ Linked via `<link rel="stylesheet" href=...>` tag or `@import` directive in embedded CSS block

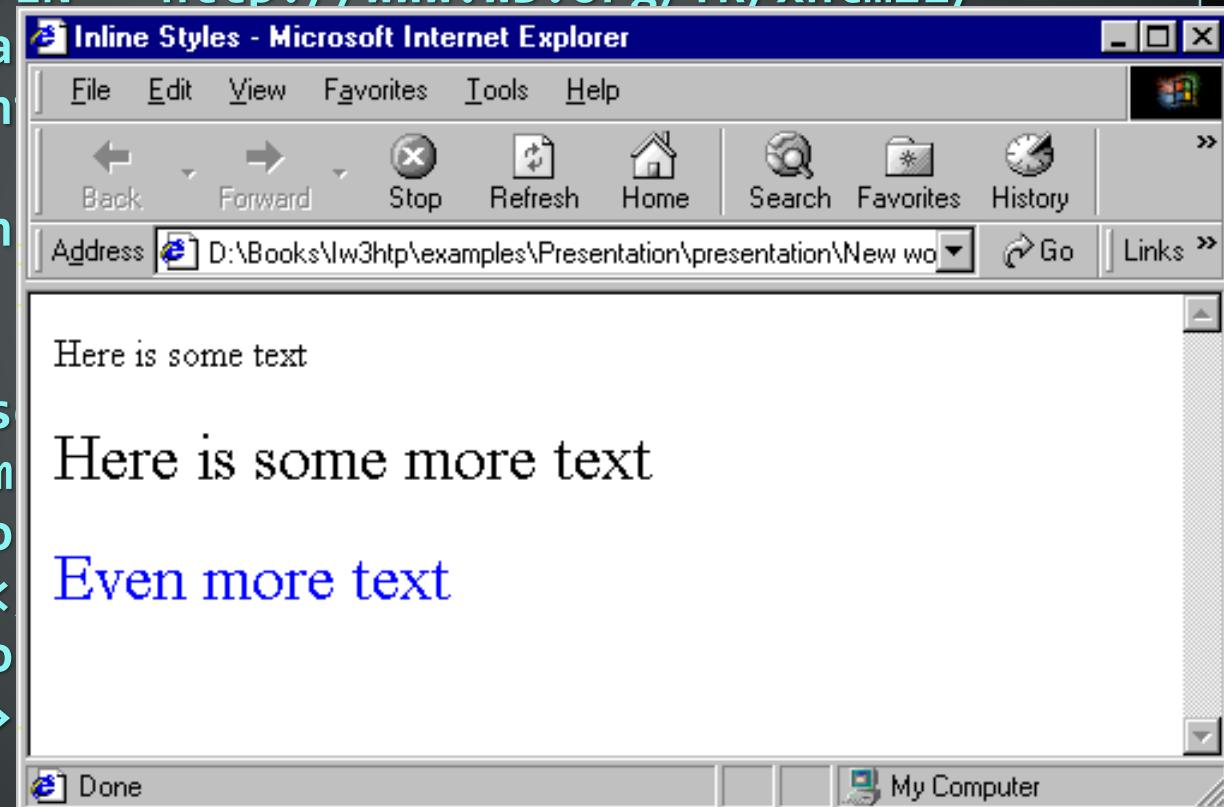
- ◆ Using external files is highly recommended
 - Simplifies the HTML document
 - Improves page load speed as the CSS file is cached

inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Inline Styles</title>
</head>
<body>
    <p>Here is some text</p>
<!--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>
</body>
</html>
```

inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/  
DTD/xhtml1-transitional.dtd"  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Inline Styles</title>  
</head>  
<body>  
    <p>Here is some text</p>  
    <!--Separate margin from content-->  
    <p style="font-size: 1.5em; margin-left: 20px;">  
        more text</p>  
    <p style="font-size: 1.2em; color: #0000FF;">  
        Even more text</p>  
</body>  
</html>
```



- ◆ Embedded in the HTML in the `<style>` tag:

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

- ◆ The `<style>` tag is placed in the `<head>` section of the document
- ◆ type attribute specifies the MIME(Multipurpose Internet Mail Extensions) type
 - ◆ MIME describes the format of the content
 - ◆ Other MIME types include `text/html`, `image/gif`, `text/javascript` ...
- ◆ Used for document-specific styles

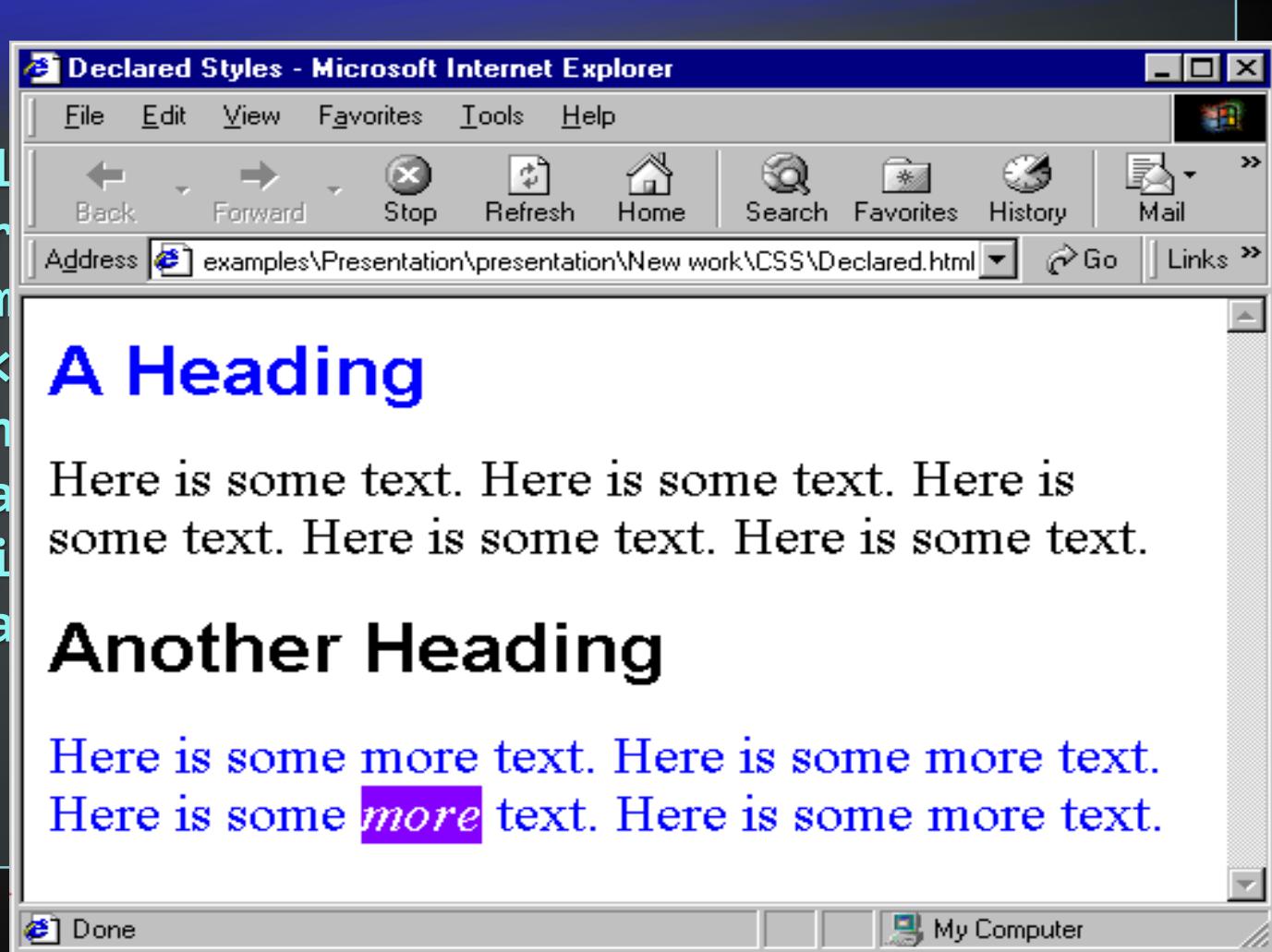
embedded-stylesheets.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Style Sheets</title>
    <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>
  <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.</p>
  <p class="blue">Here is some <em>more</em>
  text. Here is some more text.</p>
</body>
</html>
```

```
...
<body>
  <h1 class="big">A Heading</h1>
  <p>Here is some text.</p>
  <h1>Another Heading</h1>
  <p class="big">Here is some more text.</p>
</body>
</html>
```



- ◆ External linking
 - ◆ Separate pages can all use a shared style sheet
 - ◆ Only modify a single file to change the styles across your entire Web site (see <http://www.csszengarden.com/>)
- ◆ link tag (with a rel attribute)
 - ◆ Specifies a relationship between current document and another document

```
<link rel="stylesheet" type="text/css"
      href="styles.css">
```

- ◆ link elements should be in the <head>

@import

- Another way to link external CSS files
- Example:

```
<style type="text/css">
  @import url("styles.css");
  /* same as */
  @import "styles.css";
</style>
```

- Ancient browsers do not recognize @import

styles.css

```
/* CSS Document */

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 }
```

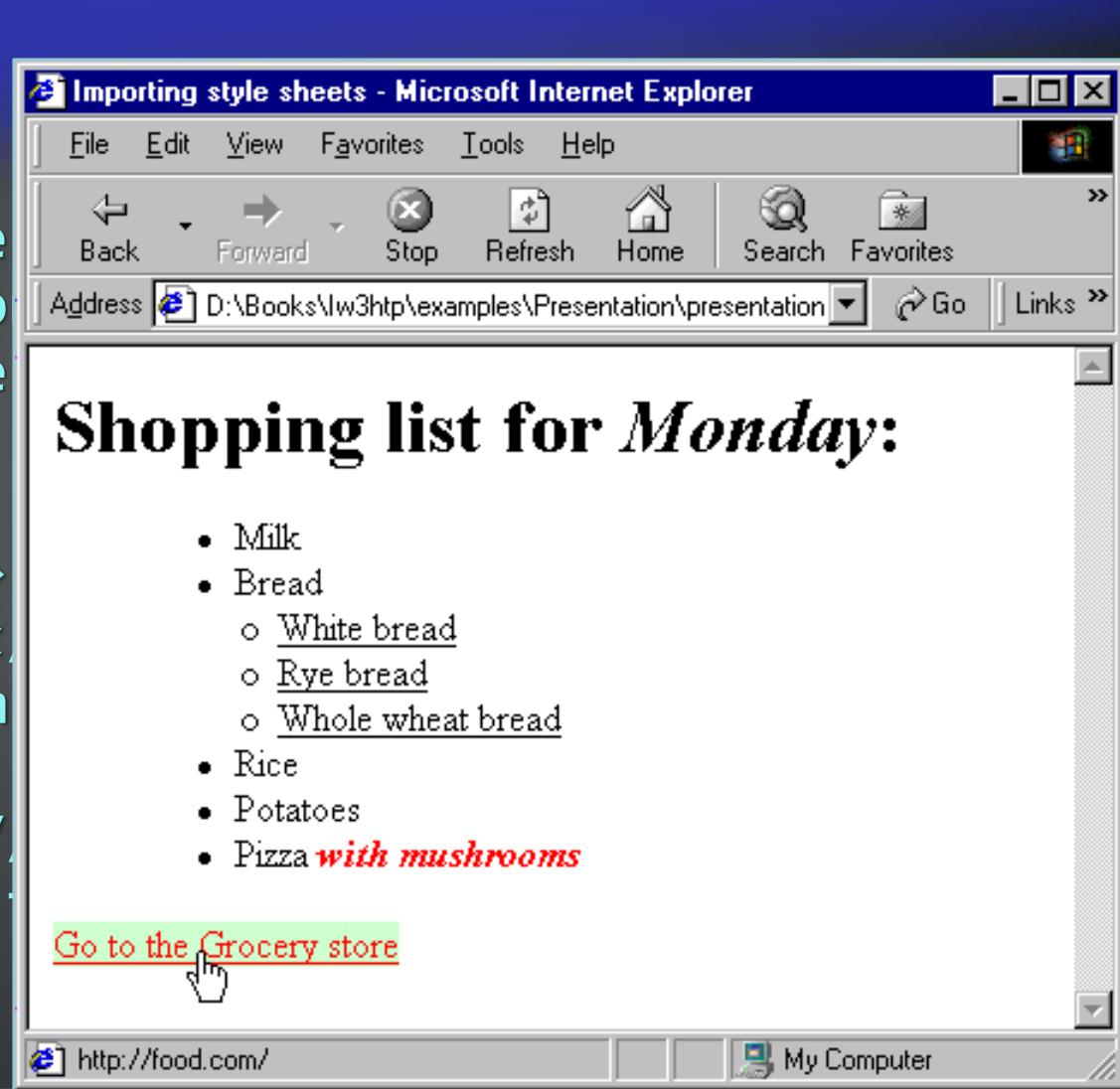
external-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
    <title>Importing style sheets</title>  
    <link type="text/css" rel="stylesheet"  
        href="styles.css" />  
</head>  
<body>  
    <h1>Shopping list for <em>Monday</em></h1>  
    <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" title="grocery
  store">Go to the Grocery store</a>
</body>
</html>
```

External Styles: Example (4)

```
...
<li>Bread
  <ul>
    <li>White
    <li>Rye b
    <li>Whole
  </ul>
</li>
<li>Rice</li>
<li>Potatoes<
<li>Pizza <em>
</ul>
<a href="http://
  store">Go to
</body>
</html>
```

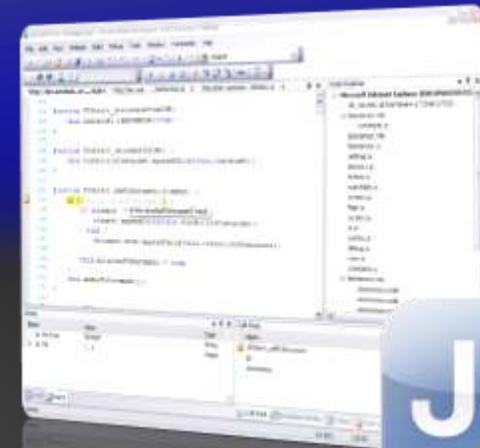
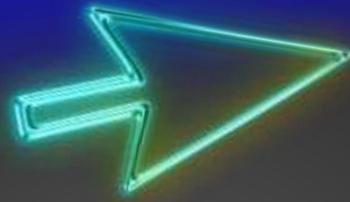


- ◆ More powerful formatting than using presentation tags
- ◆ Your pages load faster, because browsers cache the .css files
- ◆ Pages are easier to maintain and update

javascript
for
beginner

```
String.prototype.trim =  
function ()  
{  
    return this  
        .replace (^Ms+/, "")  
        .replace (/S+$/, "");  
}
```

.js



Introduction to JavaScript



- ◆ What is DHTML?
- ◆ DHTML Technologies
 - ◆ XHTML, CSS, JavaScript, DOM



- ◆ **Introduction to JavaScript**
 - ◆ **What is JavaScript**
 - ◆ **Implementing JavaScript into Web pages**
 - ◆ In <head> part
 - ◆ In <body> part
 - ◆ In external .js file



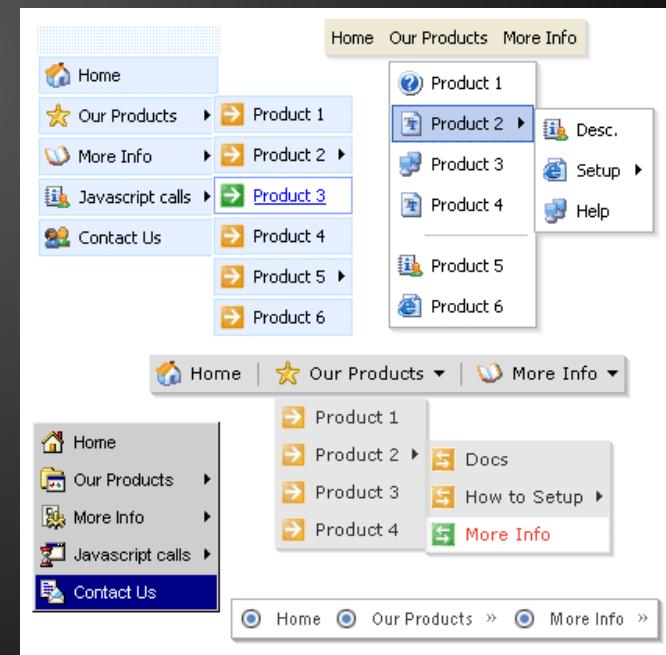
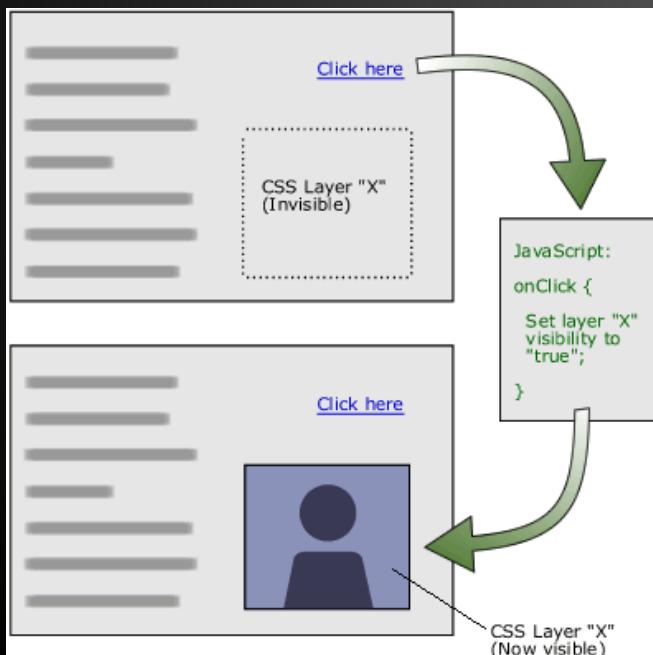
- ◆ **JavaScript Syntax**
 - ◆ **JavaScript operators**
 - ◆ **JavaScript Data Types**
 - ◆ **JavaScript Pop-up boxes**
 - ◆ **alert, confirm and prompt**
 - ◆ **Conditional and switch statements, loops and functions**
- ◆ **Document Object Model**
- ◆ **Debugging in JavaScript**



DHTML



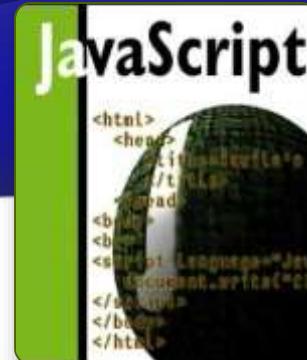
Dynamic Behavior at the Client Side



- ◆ Dynamic HTML (DHTML)
 - ◆ Makes possible a Web page to react and change in response to the user's actions
- ◆ DHTML = HTML + CSS + JavaScript



- ◆ HTML defines Web sites content through semantic tags (headings, paragraphs, lists, ...)
- ◆ CSS defines 'rules' or 'styles' for presenting every aspect of an HTML document
 - Font (family, size, color, weight, etc.)
 - Background (color, image, position, repeat)
 - Position and layout (of any object on the page)
- ◆ JavaScript defines dynamic behavior
 - Programming logic for interaction with the user, to handle events, etc.



JavaScript

JavaScript

Dynamic Behavior in a Web Page

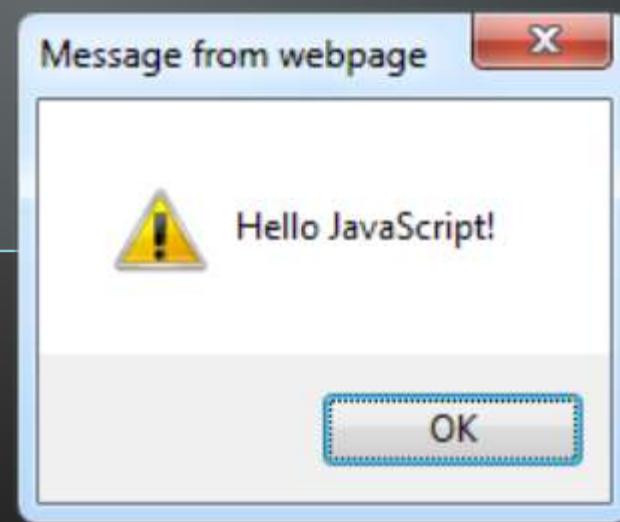
- ◆ JavaScript is a front-end scripting language developed by Netscape for dynamic content
 - ◆ Lightweight, but with limited capabilities
 - ◆ Can be used as object-oriented language
- ◆ Client-side technology
 - ◆ Embedded in your HTML page
 - ◆ Interpreted by the Web browser
- ◆ Simple and flexible

- ◆ JavaScript allows interactivity such as:
 - Implementing form validation
 - React to user actions, e.g. handle keys
 - Changing an image on moving mouse over it
 - Sections of a page appearing and disappearing
 - Content loading and changing dynamically
 - Performing complex calculations
 - Custom HTML controls, e.g. scrollable table

- ◆ Can handle events
- ◆ Can validate form data
- ◆ Can access / modify browser cookies
- ◆ Can detect the user's browser and OS
- ◆ Can be used as object-oriented language
- ◆ Can handle exceptions

first-script.html

```
<html>  
  
<body>  
  <script type="text/javascript">  
    alert('Hello JavaScript!');  
  </script>  
</body>  
  
</html>
```



small-example.html

```
<html>  
  
<body>  
  <script type="text/javascript">  
    document.write('JavaScript rulez!');  
  </script>  
</body>  
  
</html>
```



- ◆ The JavaScript code can be placed in:
 - ◆ <script> tag in the head
 - ◆ <script> tag in the body – not recommended
 - ◆ External files, linked via <script> tag the head
 - ◆ Files usually have .js extension

```
<script src="scripts.js" type="text/javascript">
  <!-- code placed here will not be executed! --&gt;
&lt;/script&gt;</pre>
```

- ◆ Highly recommended
- ◆ The .js files get cached by the browser

JavaScript – When is Executed?

- ◆ JavaScript code is executed during the page loading or when the browser fires an event
 - All statements are executed at page loading
 - Some statements just define functions that can be called later
- ◆ Function calls or code can be attached as "event handlers" via tag attributes
 - Executed when the event is fired by the browser

```

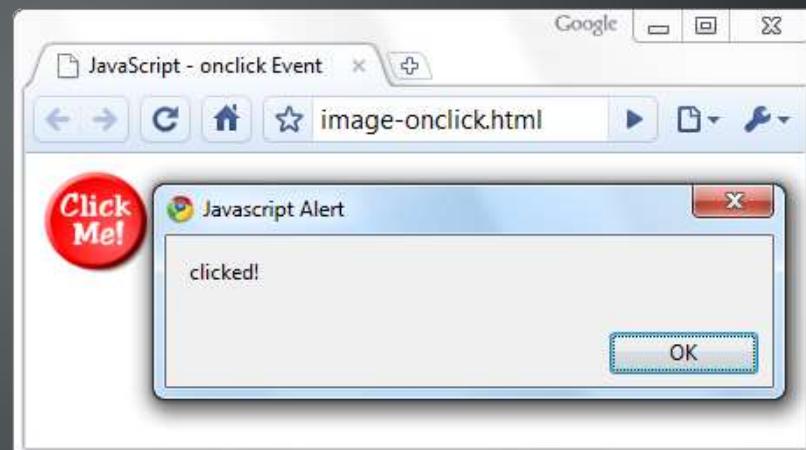
```

Calling a JavaScript Function from Event Handler – Example

```
<html>
<head>
<script type="text/javascript">
    function test (message) {
        alert(message);
    }
</script>
</head>

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

image-onclick.html



- ◆ Using external script files:

```
<html>                                external-JavaScript.html
<head>
  <script src="sample.js" type="text/javascript">
  </script>
</head>                                     The <script> tag is always empty.
<body>
  <button onclick="sample()" value="Call JavaScript
    function from sample.js" />
</body>
</html>
```

- ◆ External JavaScript file:

```
function sample() {
  alert('Hello from sample.js!')
}
```



sample.js

The JavaScript Syntax

```
if (pop < 10)
{
    map.graphics.add(features[i].setSymbol(onePopSymbol));
}
else if (pop >= 10 && pop < 95)
{
    map.graphics.add(features[i].setSymbol(twoPopSymbol));
}
else if (pop >= 95 && pop < 365)
{
    map.graphics.add(features[i].setSymbol(threePopSymbol));
}
else if (pop >= 365 && pop < 1100)
{
    map.graphics.add(features[i].setSymbol(fourPopSymbol));
}
else
{
    map.graphics.add(features[i].setSymbol(fivePopSymbol));
}
```



JAVA
SCRIPT

- ◆ The JavaScript syntax is similar to C# and Java
 - Operators (+, *, =, !=, &&, ++, ...)
 - Variables (typeless)
 - Conditional statements (if, else)
 - Loops (for, while)
 - Arrays (my_array[]) and associative arrays (my_array['abc'])
 - Functions (can return value)
 - Function variables (like the C# delegates)

- ◆ JavaScript data types:
 - ◆ Numbers (integer, floating-point)
 - ◆ Boolean (true / false)
- ◆ String type – string of characters

```
var myName = "You can use both single or double  
quotes for strings";
```

- ◆ Arrays

```
var my_array = [1, 5.3, "aaa"];
```

- ◆ Associative arrays (hash tables)

```
var my_hash = {a:2, b:3, c:"text"};
```

- ◆ Every variable can be considered as object
 - ◆ For example strings and arrays have member functions:

objects.html

```
var test = "some string";
alert(test[7]); // shows letter 'r'
alert(test.charAt(5)); // shows letter 's'
alert("test".charAt(1)); //shows letter 'e'
alert("test".substring(1,3)); //shows 'es'
```

```
var arr = [1,3,4];
alert (arr.length); // shows 3
arr.push(7); // appends 7 to end of array
alert (arr[3]); // shows 7
```

- ◆ The + operator joins strings

```
string1 = "fat ";  
string2 = "cats";  
alert(string1 + string2); // fat cats
```

- ◆ What is "9" + 9?

```
alert("9" + 9); // 99
```

- ◆ Converting string to number:

```
alert(parseInt("9") + 9); // 18
```

telerik Arrays Operations and Properties

- ◆ Declaring new empty array:

```
var arr = new Array();
```

- ◆ Declaring an array holding few elements:

```
var arr = [1, 2, 3, 4, 5];
```

- ◆ Appending an element / getting the last element:

```
arr.push(3);
```

```
var element = arr.pop();
```

- ◆ Reading the number of elements (array length):

```
arr.length;
```

- ◆ Finding element's index in the array:

```
arr.indexOf(1);
```

- ◆ Alert box with text and [OK] button
 - ◆ Just a message shown in a dialog box:

```
alert("Some text here");
```

- ◆ Confirmation box
 - ◆ Contains text, [OK] button and [Cancel] button:

```
confirm("Are you sure?");
```

- ◆ Prompt box
 - ◆ Contains text, input field with default value:

```
prompt ("enter amount", 10);
```

sum-of-numbers.html

```
<html>

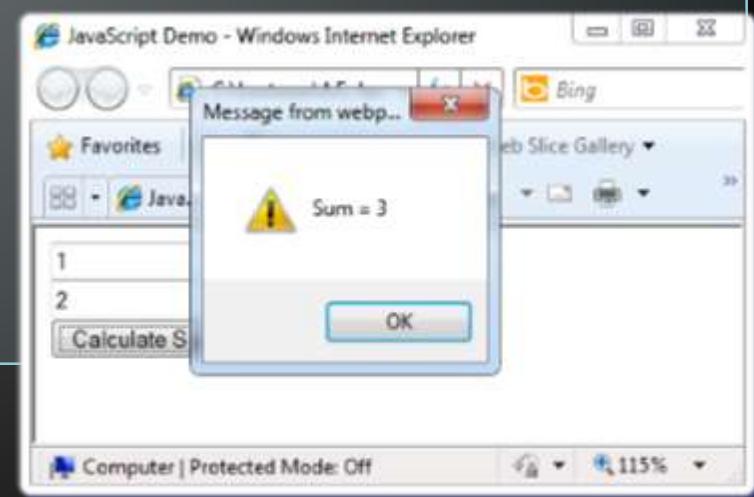
<head>
    <title>JavaScript Demo</title>
    <script type="text/javascript">
        function calcSum() {
            value1 =
                parseInt(document.mainForm.textBox1.value);
            value2 =
                parseInt(document.mainForm.textBox2.value);
            sum = value1 + value2;
            document.mainForm.textBoxSum.value = sum;
        }
    </script>
</head>
```

Sum of Numbers – Example (2)

sum-of-numbers.html (cont.)

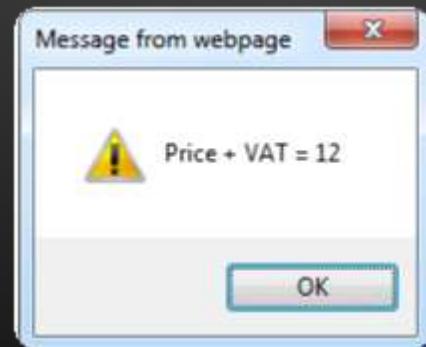
```
<body>
  <form name="mainForm">
    <input type="text" name="textBox1" /> <br/>
    <input type="text" name="textBox2" /> <br/>
    <input type="button" value="Process"
      onclick="javascript: calcSum()" />
    <input type="text" name="textBoxSum"
      readonly="readonly"/>
  </form>
</body>

</html>
```



prompt.html

```
price = prompt("Enter the price", "10.00");
alert('Price + VAT = ' + price * 1.2);
```

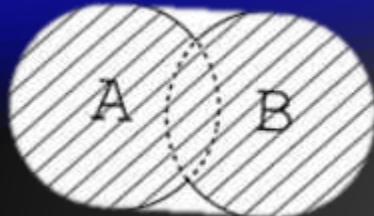


Conditional Statement (if)

```
unitPrice = 1.30;  
if (quantity > 100) {  
    unitPrice = 1.20;  
}
```

Symbol	Meaning
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal
!=	Not equal

- The condition may be of Boolean or integer type:



conditional-statements.html

```
var a = 0;
var b = true;
if (typeof(a)=="undefined" || typeof(b)=="undefined") {
    document.write("Variable a or b is undefined.");
}
else if (!a && b) {
    document.write("a==0; b==true;");
} else {
    document.write("a==" + a + "; b==" + b + ";");
}
```

- ◆ The switch statement works like in C#:

```
switch (variable) {  
    case 1:  
        // do something  
        break;  
    case 'a':  
        // do something else  
        break;  
    default:  
        // something completely different  
}
```

[switch-statements.html](#)

- ◆ Like in C#

- ◆ for loop
- ◆ while loop
- ◆ do ... while loop



```
var counter;  
for (counter=0; counter<4; counter++) {  
    alert(counter);  
}  
while (counter < 5) {  
    alert(++counter);  
}
```



loops.html

- ◆ Code structure – splitting code into parts
- ◆ Data comes in, processed, result returned

```
function average(a, b, c)
{
    var total;
    total = a+b+c;
    return total/3;
}
```

Parameters come
in here.

Declaring variables
is optional. Type is
never declared.

Value returned
here.

Function Arguments and Return Value

- When calling function it is not obligatory to specify all of its arguments
 - The function has access to all the arguments passed via arguments array

```
function sum() {  
    var sum = 0;  
    for (var i = 0; i < arguments.length; i++)  
        sum += parseInt(arguments[i]);  
    return sum;  
}  
alert(sum(1, 2, 4));
```

functions-demo.html

Questions?