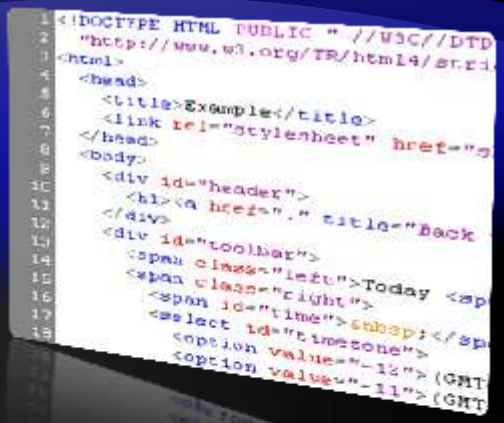
telerik



HTML Basics

HTML, Text, Images, Tables

telerik

Table of Contents

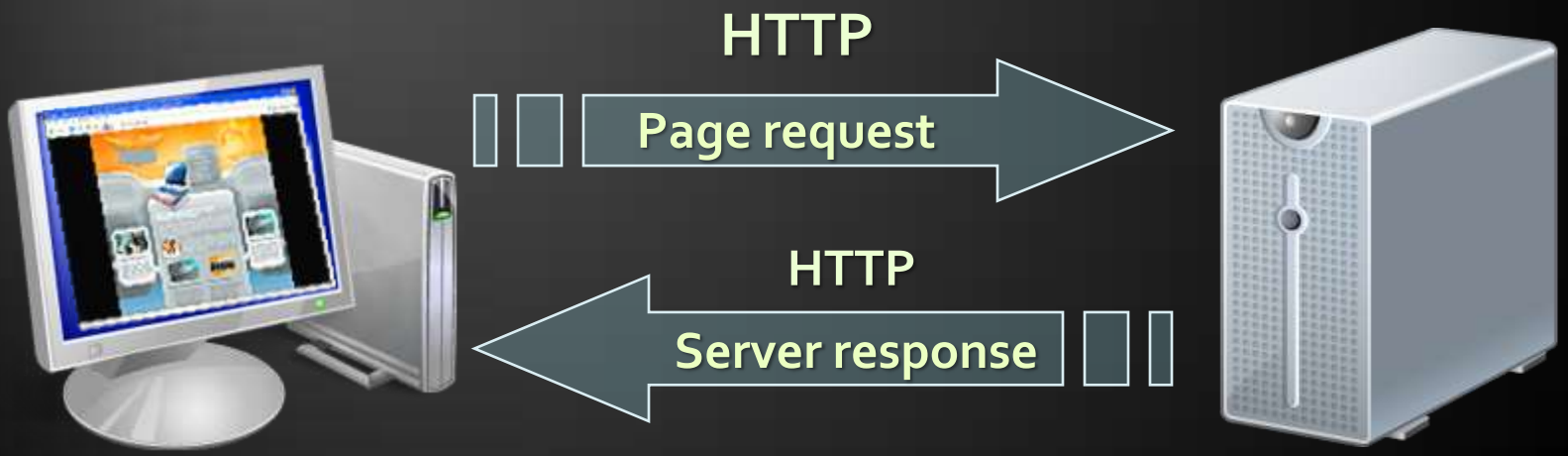
telerik

Table of Contents (2)

telerik

How the Web Works?

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



Client running a Web Browser

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

telerik

What is a Web Page?

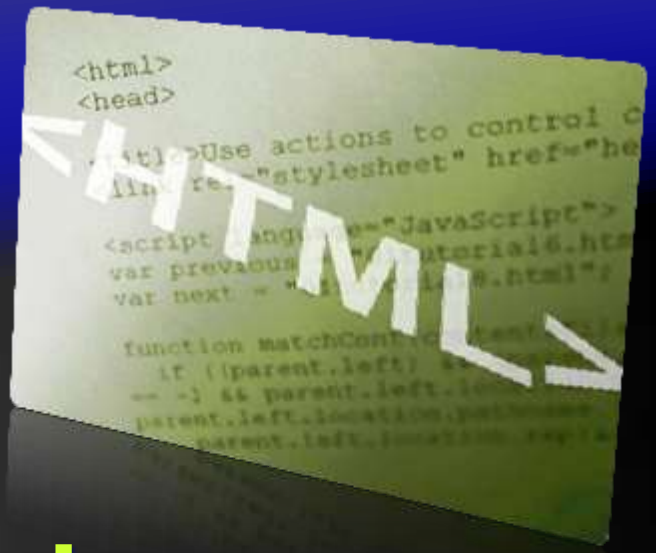
- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
  - Looks (looked?) like:
    - A Microsoft Word document
- The markup tags provide information about the page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

telerik

Creating HTML Pages

- 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 (WYSIWYG Editors):
  - Microsoft FrontPage
  - Macromedia Dreamweaver
  - Netscape Composer
  - Microsoft Word
  - Visual Studio

telerik



HTML Basics

Text, Images, Tables, Forms

telerik

HTML Structure

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

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

- Tags have attributes:

```
<img src="logo.jpg" alt="logo" />
```

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

telerik

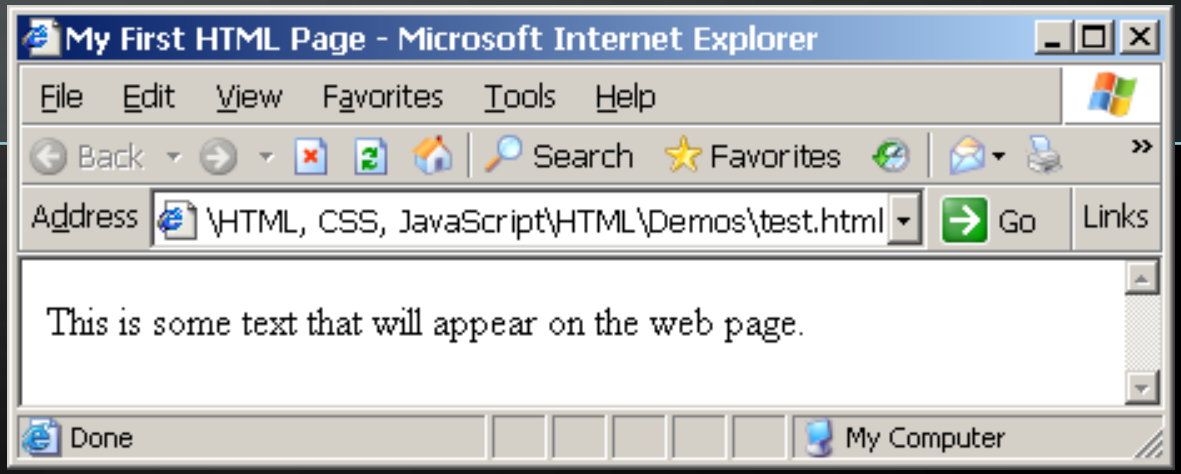
HTML Code Formatting

- 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, so formatting is harmless.
- For performance reasons, formatting can be sacrificed

First HTML Page

test.html

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



First HTML Page: Tags

```
<!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.

telerik

First HTML Page: Header

HTML header

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

First HTML Page: Body

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

HTML body

telerik

Some Simple Tags

- Hyperlink Tags

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

- Image Tags

```
<img src="logo.gif" alt="logo" />
```

- Text formatting tags

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

Some Simple Tags – Example

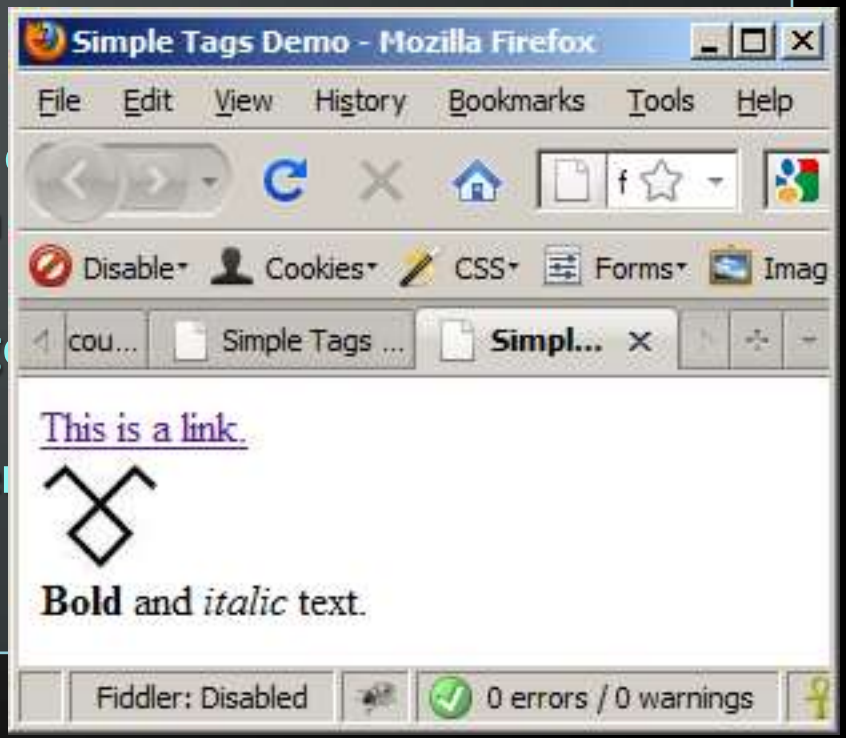
some-tags.html

```
<!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 />
<img src="logo.gif" alt="logo" />
<br />
<strong>Bold</strong> and <em>italic</em> text.
</body>
</html>
```

Some Simple Tags – Example (2)

some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.telerik.
    "Telerik site">This is a
<br />
<img src="logo.gif" alt="log
<br />
<strong>Bold</strong> and <e
</body>
</html>
```



telerik

Tags Attributes

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

Attribute `alt` with value "`logo`"

```
<img src="logo.gif" alt="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

telerik

Headings and Paragraphs

- 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 and span

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

Headings and Paragraphs – Example

headings.html

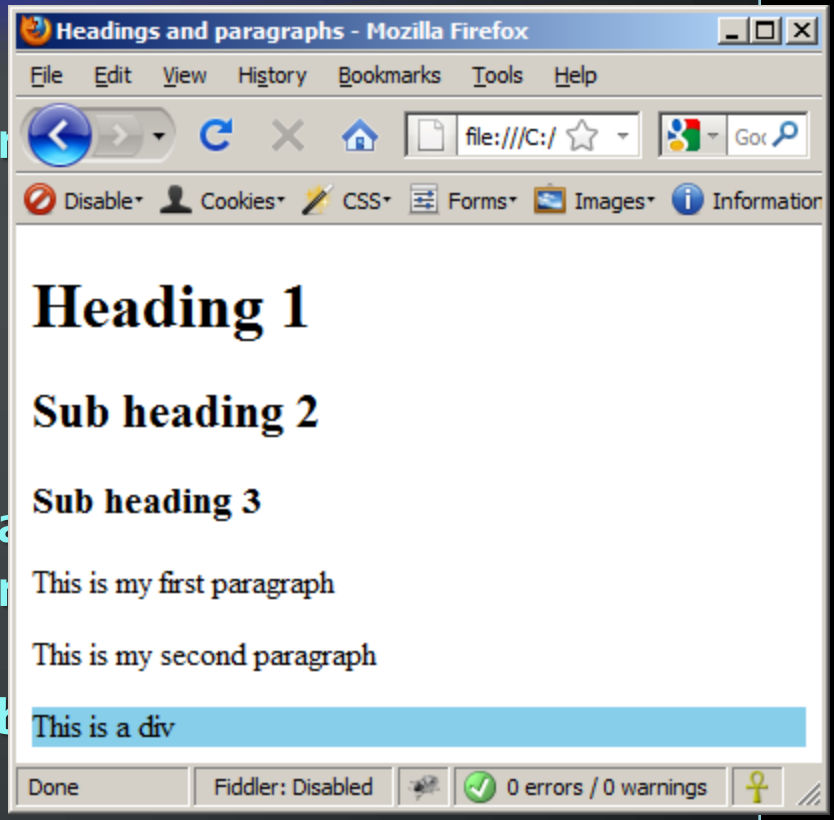
```
<!DOCTYPE HTML>
<html>
  <head><title>Headings and paragraphs</title></head>
  <body>
    <h1>Heading 1</h1>
    <h2>Sub heading 2</h2>
    <h3>Sub heading 3</h3>

    <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>
```

telerik

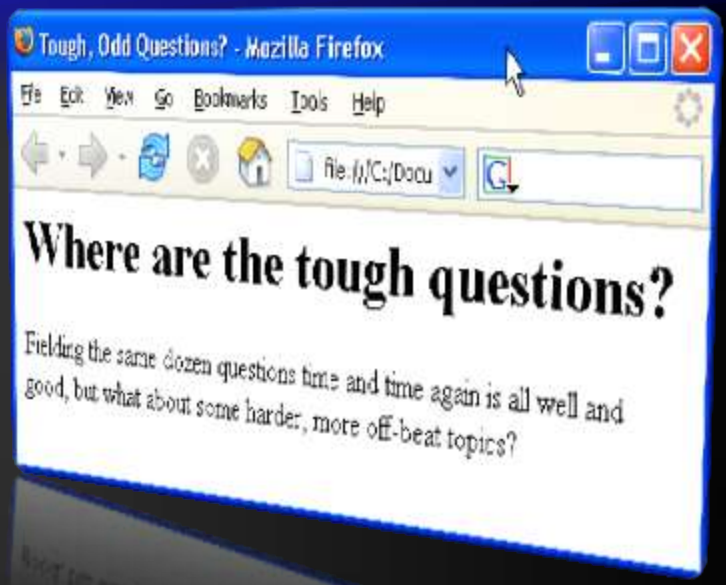
Headings and Paragraphs – Example (2)

headings.html

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

    <p>This is my first paragra
    <p>This is my second paragr

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



telerik



Introduction to HTML

HTML Document Structure in Depth

telerik

Preface

- 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.

The <!DOCTYPE> Declaration

- HTML documents must start with a document type definition (DTD)
  - It tells web browsers what type is the served code
  - Possible versions: HTML 4.01, XHTML 1.0 (Transitional or Strict), XHTML 1.1, HTML 5
- Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

  - See http://w3.org/QA/2002/04/valid-dtd-list.html for a list of possible doctypes

telerik

HTML vs. XHTML

- XHTML is more strict than HTML
  - Tags and attribute names must be in lowercase
  - All tags must be closed (`<br/>`, `<img/>`) while HTML allows `<br>` and `<img>` and implies missing closing tags (`<p>par1 <p>par2`)
  - XHTML allows only one root `<html>` element (HTML allows more than one)

telerik

XHTML vs. HTML (2)

- 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!

telerik

The <head> Section

- Contains information that doesn't show directly on the viewable page
- Starts after the <!doctype> declaration
- 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

telerik

<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" />
```

telerik

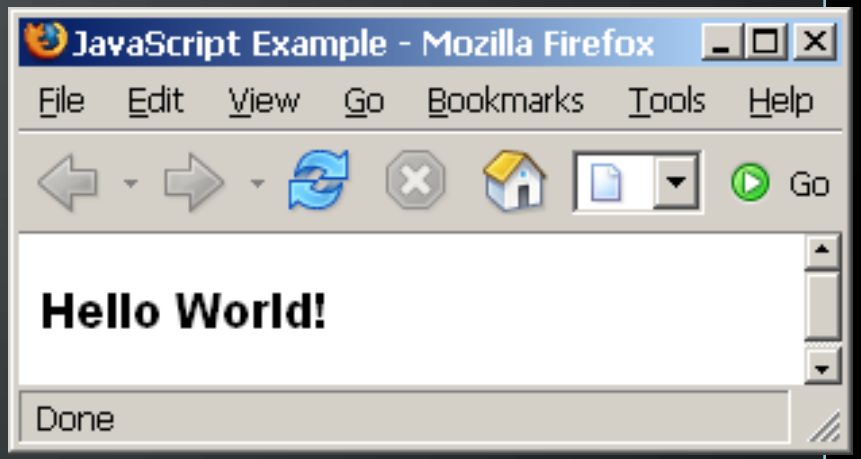
<head> Section: <script>

- 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

scripts-example.html

```
<!DOCTYPE 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>
```

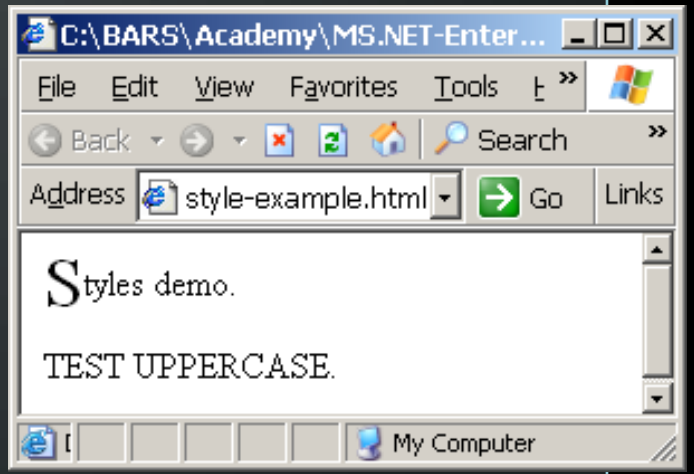


<head> Section: <style>

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

style-example.html

```
<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>
```



Comments: <!-- --> Tag

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

```
<!-- Telerik Logo (a JPG file) -->
<img src="logo.jpg" alt="Telerik Logo">
<!-- Hyperlink to the web site -->
<a href="http://telerik.com/">Telerik</a>
<!-- Show the news table -->
<table class="newstable">
...
```

telerik

<body> Section: Introduction

- 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>
```

telerik

Text Formatting

- Text formatting tags modify the text between the opening tag and the closing tag
  - Ex. `<b>Hello</b>` makes "Hello" bold

`<b></b>`	**bold**
`<i></i>`	*italicized*
`<u></u>`	underlined
`<sup></sup>`	Sample^{superscript}
`<sub></sub>`	Sample$_{subscript}$
`<strong></strong>`	**strong**
`<em></em>`	*emphasized*
`<pre></pre>`	`Preformatted text`
`<blockquote></blockquote>`	`Quoted text block`
`<del></del>`	Deleted text – ~~strike through~~

telerik

Text Formatting – Example

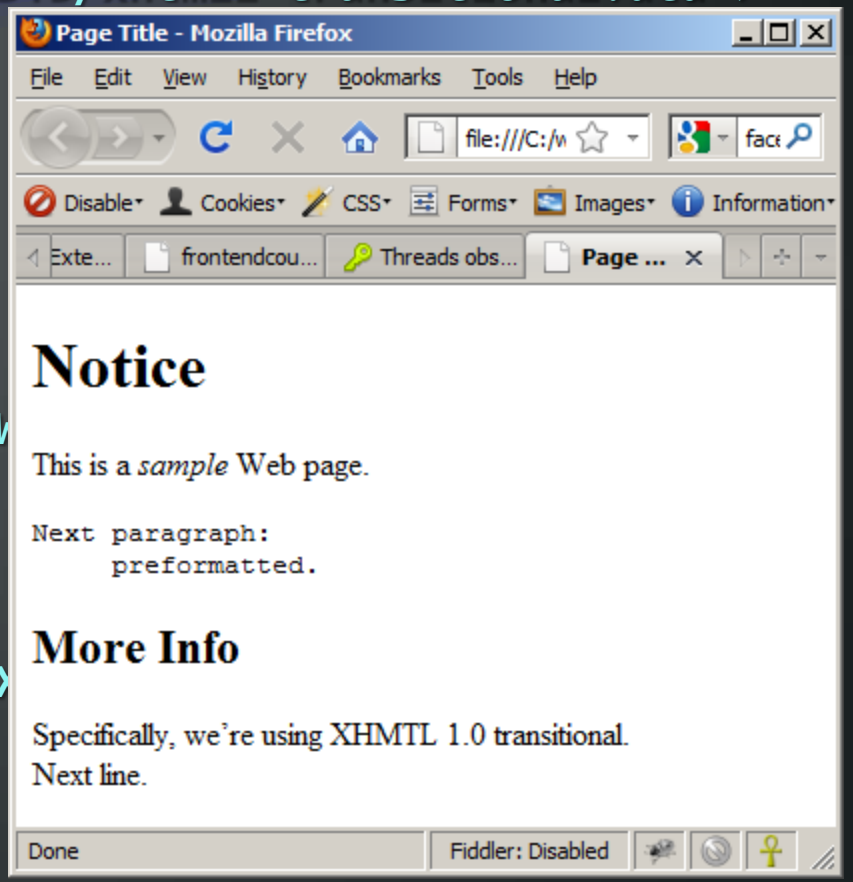
text-formatting.html

```
<!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 XHMTL 1.0 transitional.<br />
    Next line.</p>
  </body>
</html>
```

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> W
  <p><pre>Next paragraph:
     preformatted.</pre></p>
  <h2>More Info</h2>
  <p>Specifically, we're using X
    Next line.</p>
  </body>
</html>
```



telerik

Hyperlinks: <a> Tag

- 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>
```

telerik

Hyperlinks: <a> Tag (2)

- 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>
```

telerik

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"><img
  src="apply-now-button.jpg" /></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>
```

telerik

Hyperlinks and Sections

- 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>
```

telerik

Hyperlinks – Example

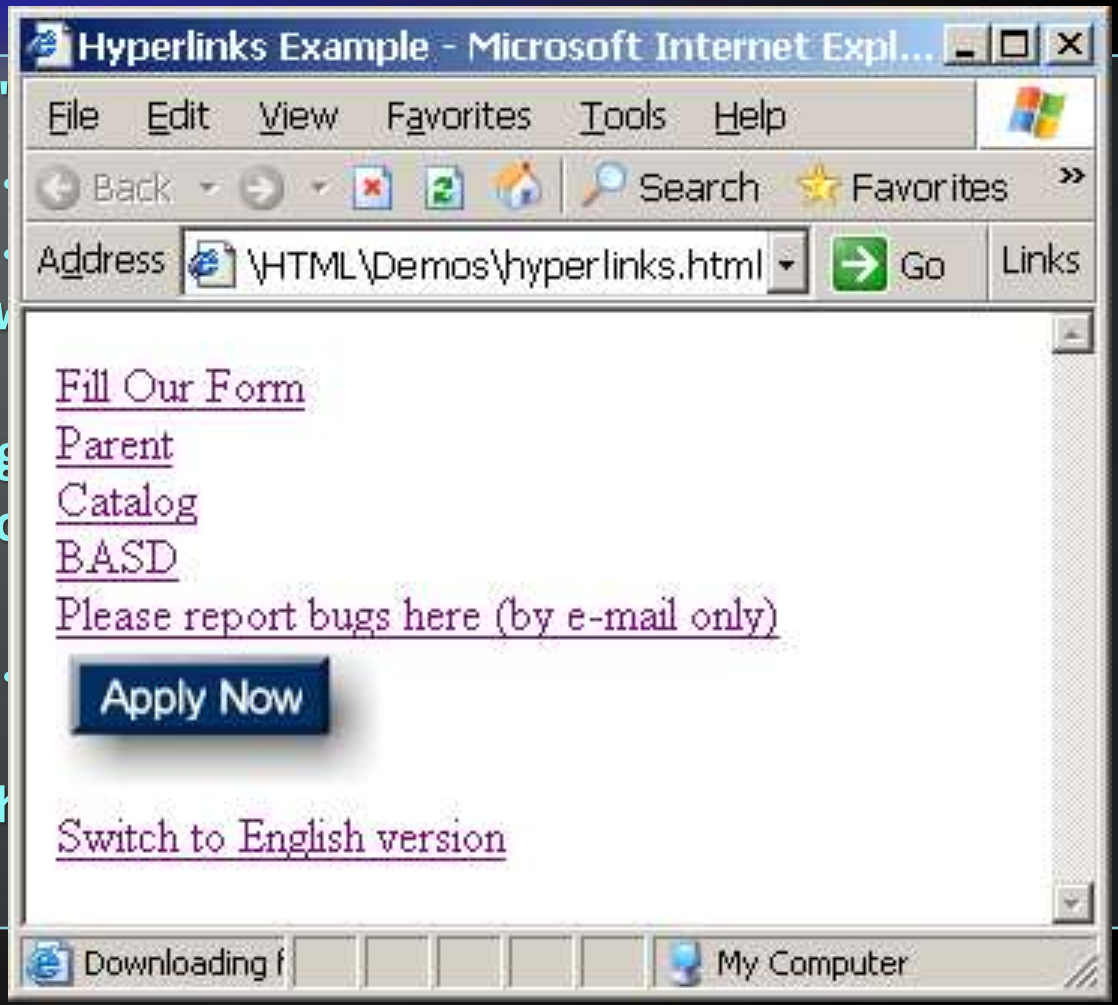
hyperlinks.html

```
<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"><img src="apply-now-button.jpg”
/></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/cat.
<a href="http://www
<br />
<a href="mailto:bug
Report">Please repo
<br />
<a href="apply-now.
/></a> <br />
<a href="../englisH
version</a> <br />
```



Links to the Same Document – Example

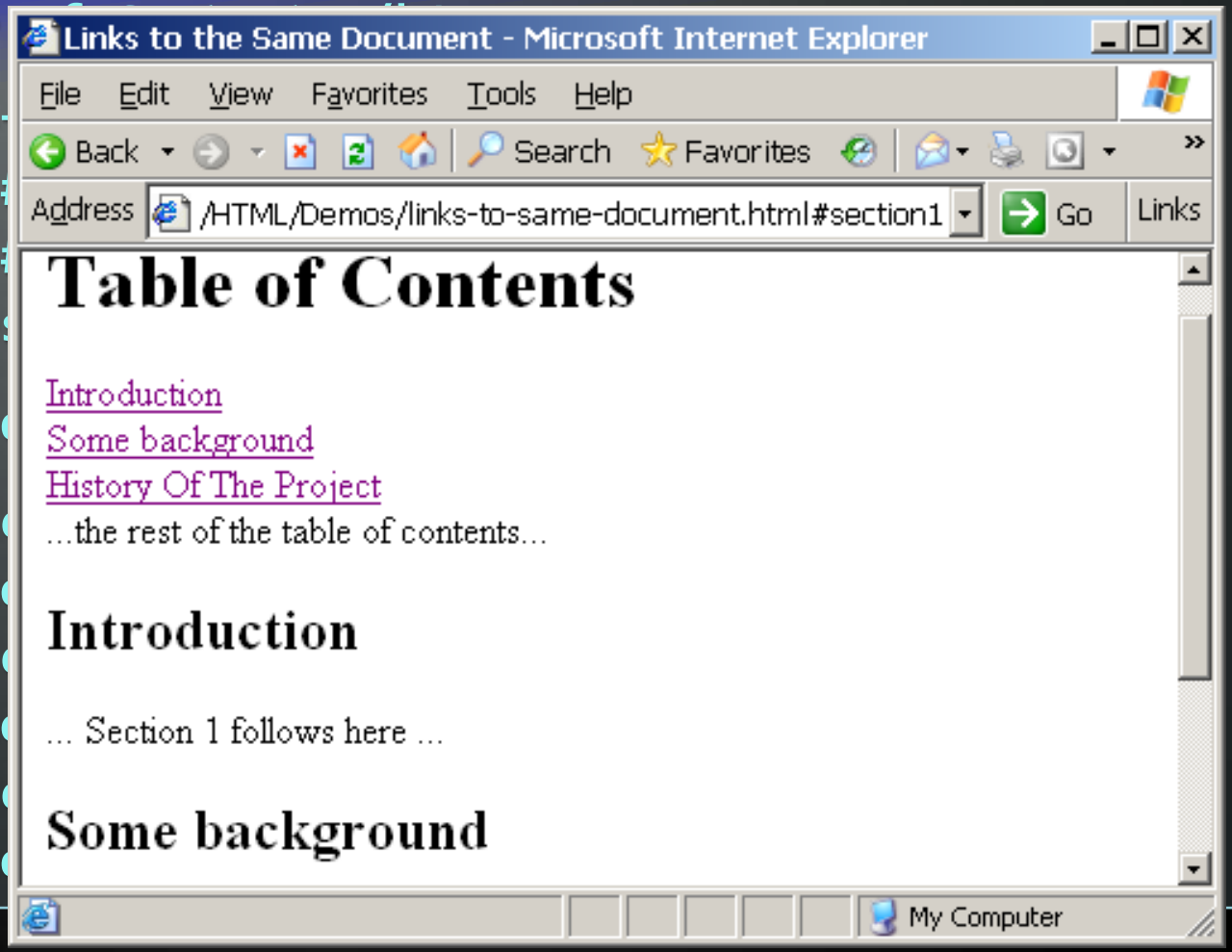
links-to-same-document.html

```
<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 the Same Document – Example (2)

links-to-same-document.html

```
<h1>Table
<p><a hre
<a href="
<a href="
...the re
<!-- The
<h2 id="s
... Secti
<h2 id="s
... Secti
<h3 id="s
... Secti
```



telerik

Images: <img> tag

- Inserting an image with <img> tag:

```
<img src="/img/basd-logo.png">
```

- 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:

```
<img src="./php.png" alt="PHP Logo" />
```

telerik

Miscellaneous Tags

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

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

- : Deprecated!

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

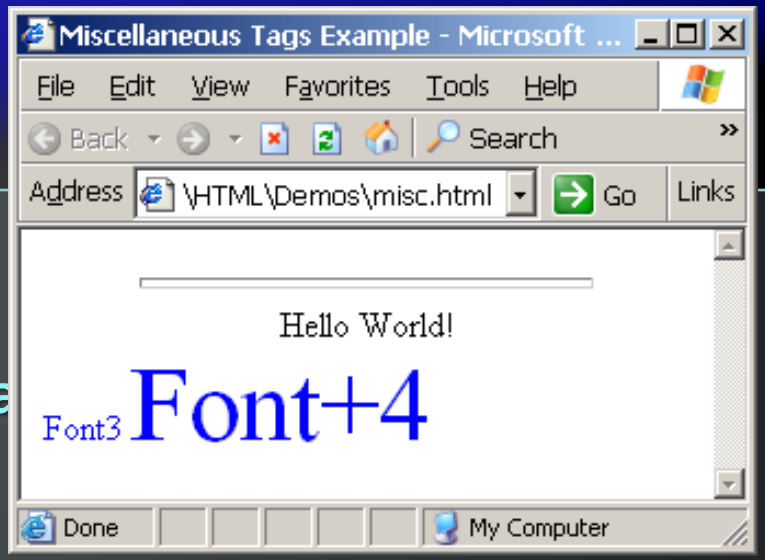
- : Deprecated!

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

Miscellaneous Tags – Example

misc.html

```
<html>
  <head>
    <title>Miscellaneous Ta
  </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>
```

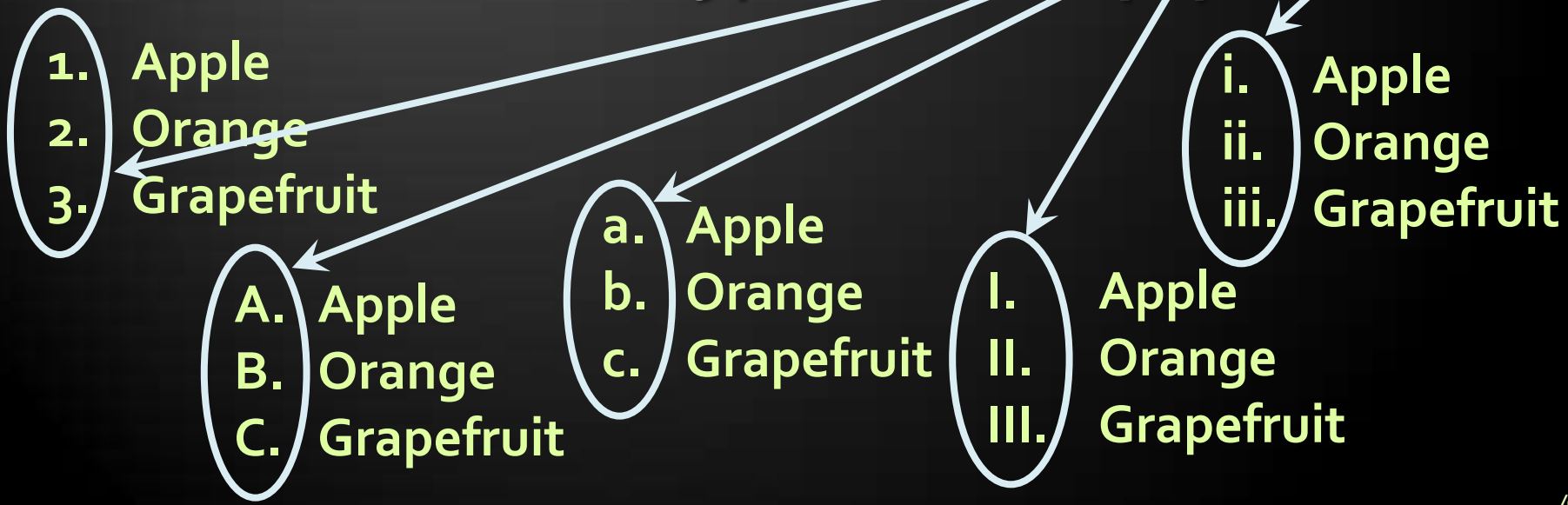


Ordered Lists: <ol> Tag

- 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

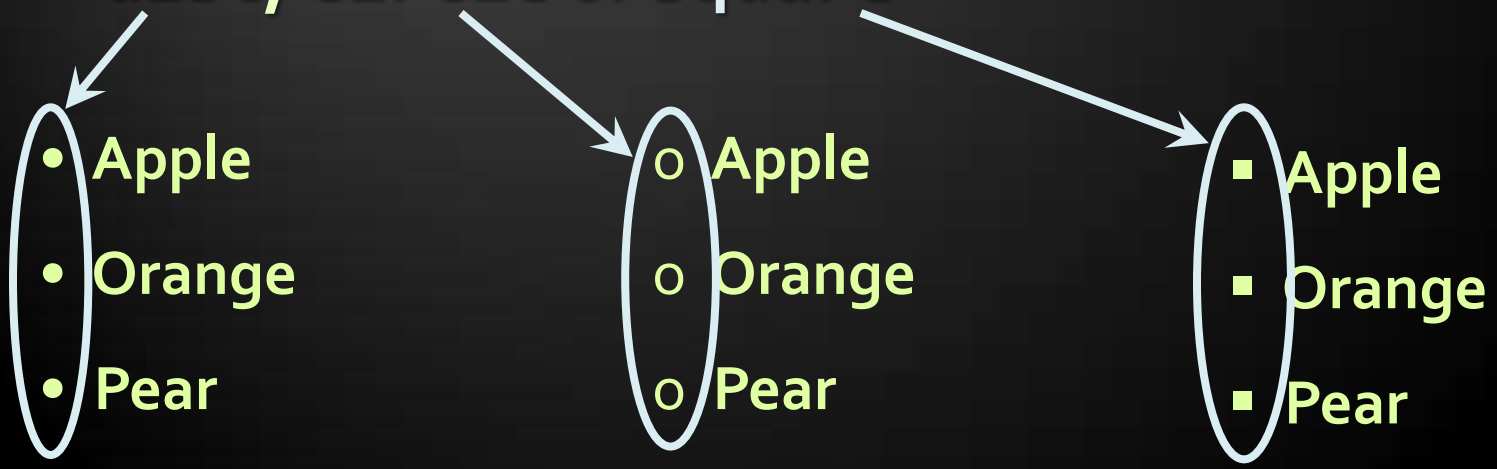


Unordered Lists: <ul> Tag

- 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



telerik

Definition lists: <dl> tag

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

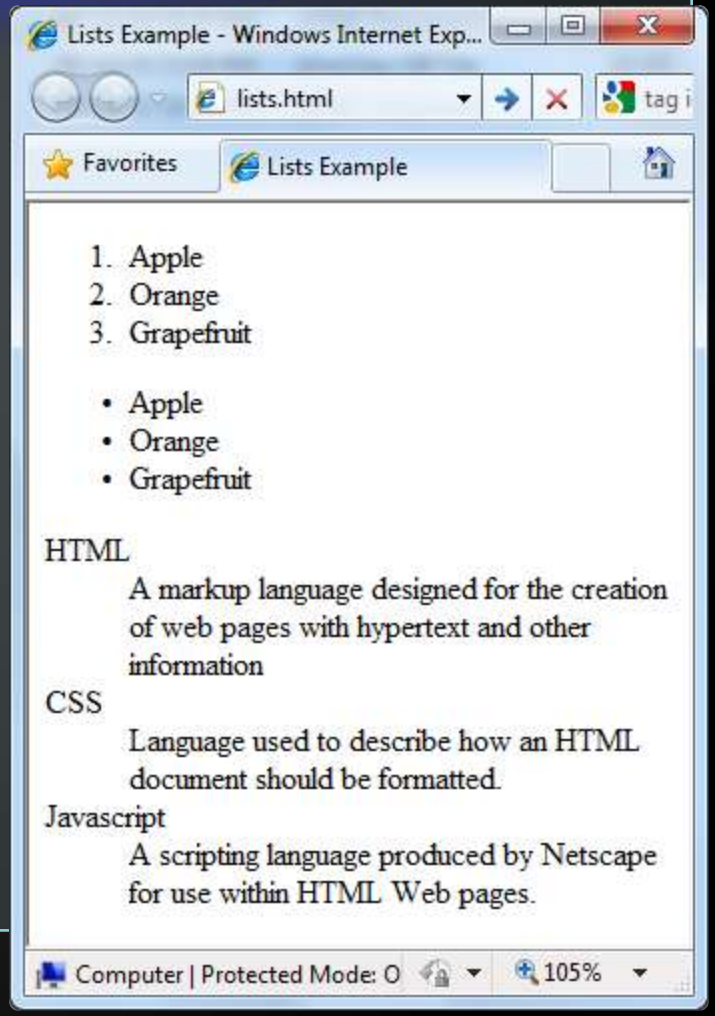
```
<dl>
   <dt>HTML</dt>
   <dd>A markup language …</dd>
   <dt>CSS</dt>
   <dd>Language used to …</dd>
</dl>
```

  - Renders without bullets
  - Definition is indented

Lists – Example

lists.html

```
<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>
```



telerik

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	¥	¥

Special Characters – Example

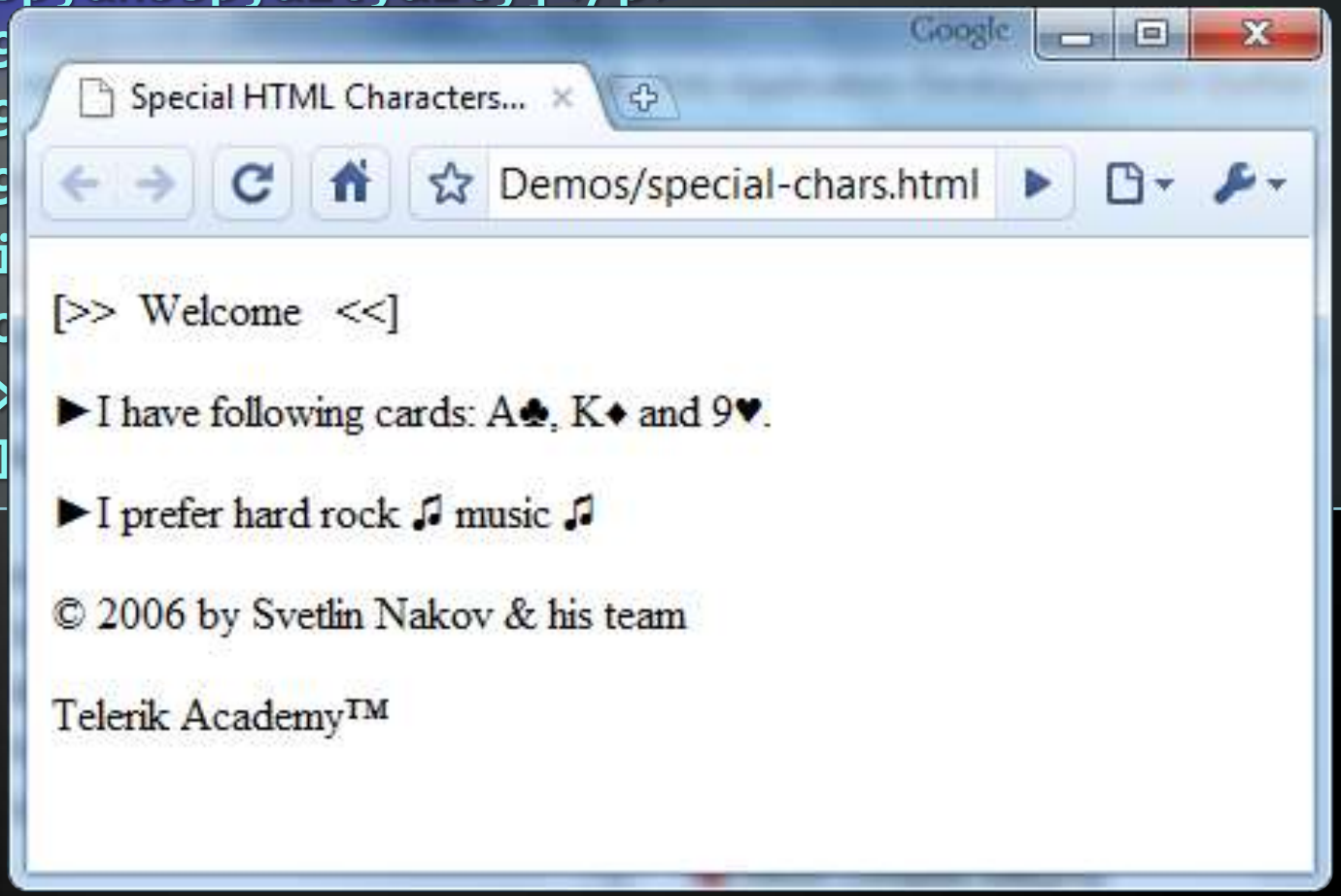
special-chars.html

```
<p>[&gt;&gt;  Welcome
      &lt;&lt;]</p>
  <p>&#9658;I have following cards:
    A&#9827;, K&#9830; and 9&#9829;.</p>
  <p>&#9658;I prefer hard rock &#9835;
    music &#9835;</p>
  <p>&copy; 2006 by Svetlin Nakov &amp; his
team</p>
  <p>Telerik Academy™</p>
```

Special Chars – Example (2)

special-chars.html

```
<p>[&gt;&gt;  Welcome
      &lt;&lt;]</p>
  <p>&#9
    A&#9
  <p>&#9
    musi
  <p>&co
team</p>
  <p>Tel
```



telerik

```
<head>
<meta http-equiv="Content-Type"
<title>Home</title>
<link rel="stylesheet" href="sty
<style type="text/css">
.stylel {
    color: #FF0000;
}
</style>
</head>

<span class="stylel">You will have to
separate license to use the OpenCube
```

Using <DIV> and <SPAN> Block and Inline Elements

telerik

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
  - `<span>` is an inline element
  - Most HTML elements are inline, e.g. `<a>`

The Tag

- 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>
```

The Tag

- 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>
```

telerik

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

telerik

HTML Tables

- 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

telerik

HTML Tables (2)

- 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

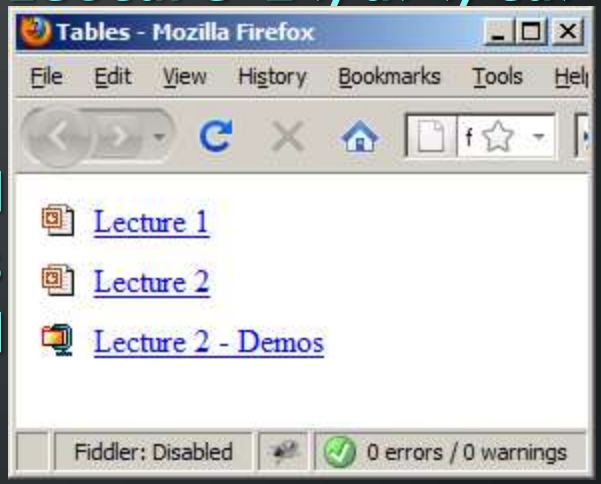
```
<td> ... </td>
```

Simple HTML Tables – Example

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

Simple HTML Tables – Example (2)

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



telerik

Complete HTML Tables

- Table rows split into three semantic sections: header, body and footer
  - `<thead>` denotes table header and contains `<th>` elements, instead of `<td>` elements
  - `<tbody>` denotes collection of table rows that contain the very data
  - `<tfoot>` denotes table footer but comes BEFORE the `<tbody>` tag
  - `<colgroup>` and `<col>` define columns (most often used to set column widths)

Complete HTML Table: Example

```
<table>
<colgroup>
  <col style="width:100px" /><col />
</colgroup>
<thead>
  <tr><th>Column 1</th><th>Column 2</th></tr>
</thead>
<tfoot>
  <tr><td>Footer 1</td><td>Footer 2</td></tr>
</tfoot>
<tbody>
  <tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>
  <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
</tbody>
</table>
```

columns

header

th

footer

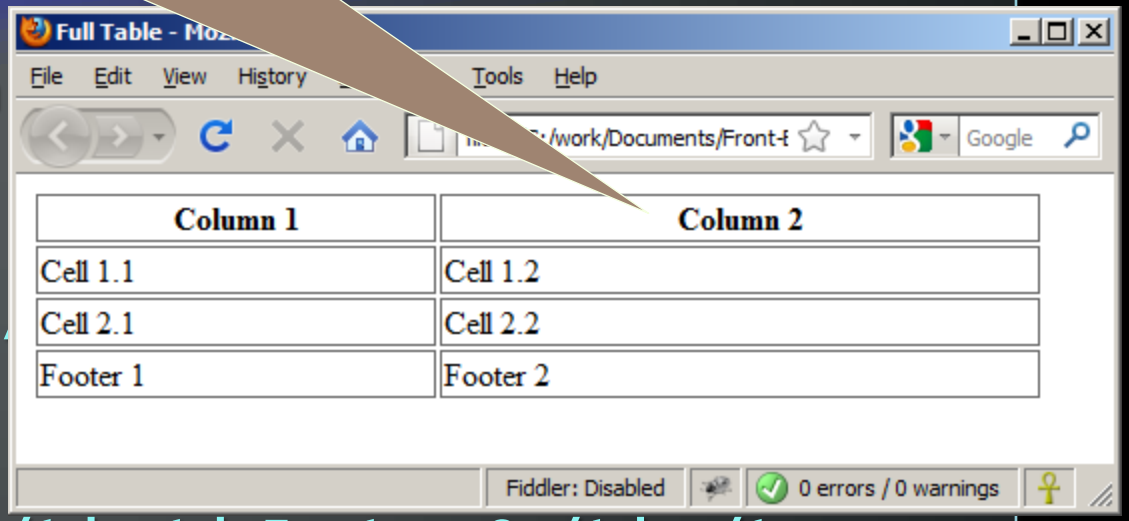
Last comes the body (data)

Complete HTML Table: Example (2)

By default, header text is bold and centered.

table-full.html

```
<table>
<colgroup>
  <col style="width
</colgroup>
<thead>
  <tr><th>Column 1</
</thead>
<tfoot>
  <tr><td>Footer 1</td><td>Footer 2</td></tr>
</tfoot>
<tbody>
  <tr><td>Ce                                    d></tr>
  <tr><td>Ce                                    d></tr>
</tbody>
</table>
```



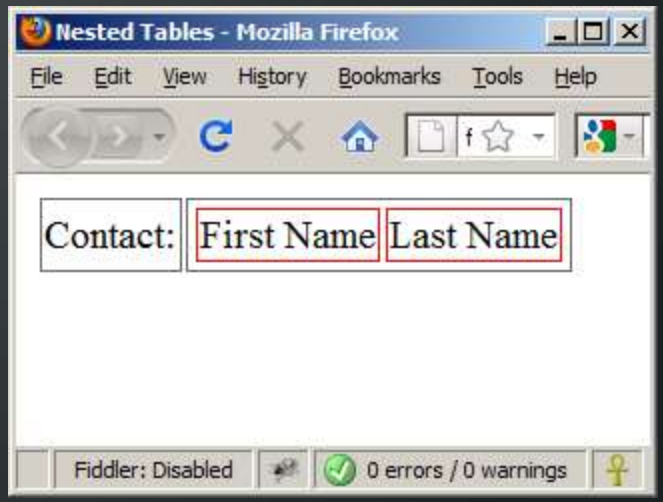
Although the footer is before the data in the code, it is displayed last

Nested Tables

- Table data "cells" (<td>) can contain nested tables (tables within tables):

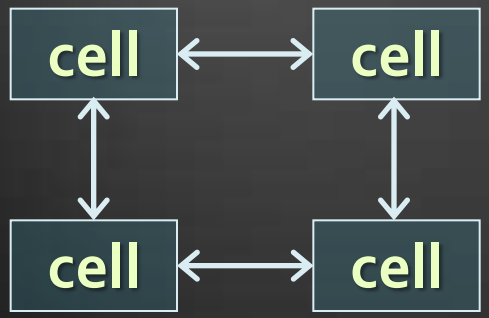
nested-tables.html

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

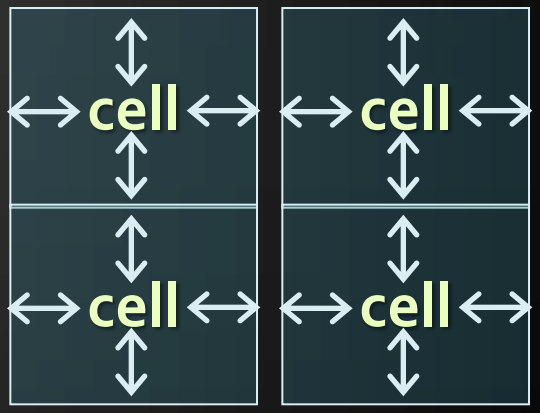


Cell Spacing and Padding

- Tables have two important attributes:
  - `cellspacing`
    - Defines the empty space between cells



  - `cellpadding`
    - Defines the empty space around the cell content



telerik

Cell Spacing and Padding – Example

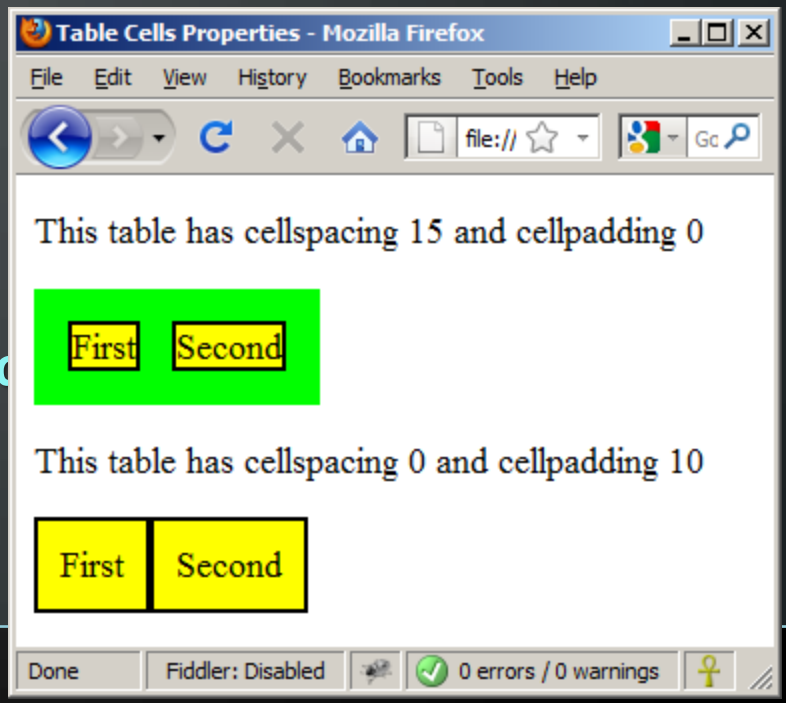
table-cells.html

```
<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)

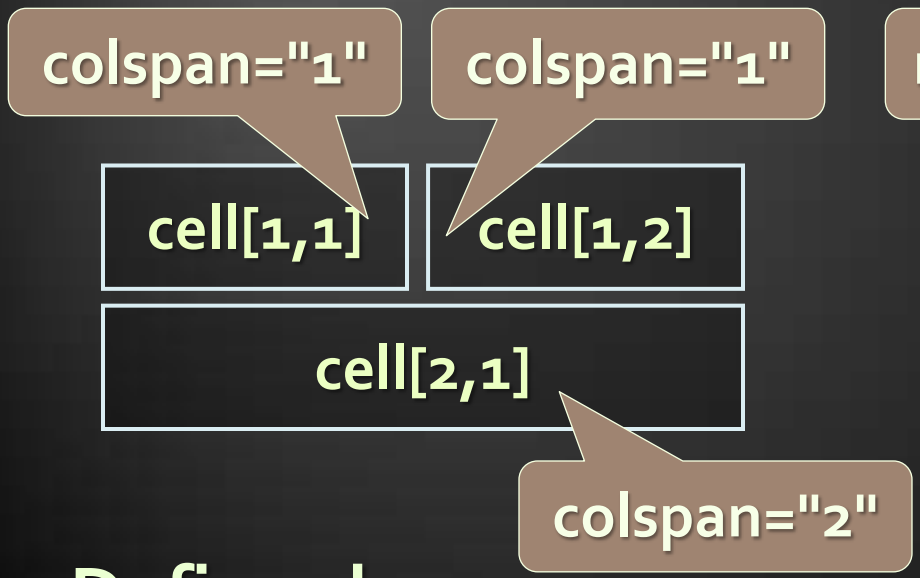
table-cells.html

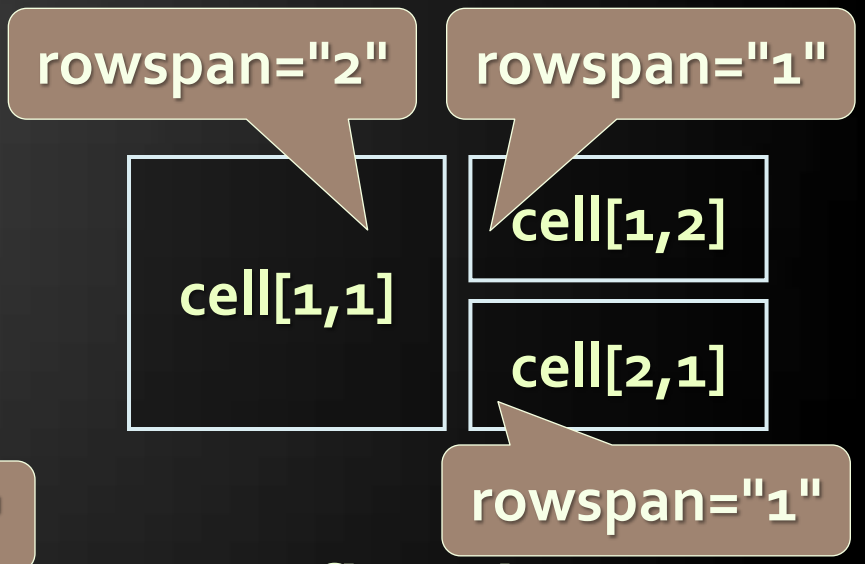
```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15" cellpadding="0">
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0"
      <tr><td>First</td><td
    </table>
  </body>
</html>
```



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





Column and Row Span – Example

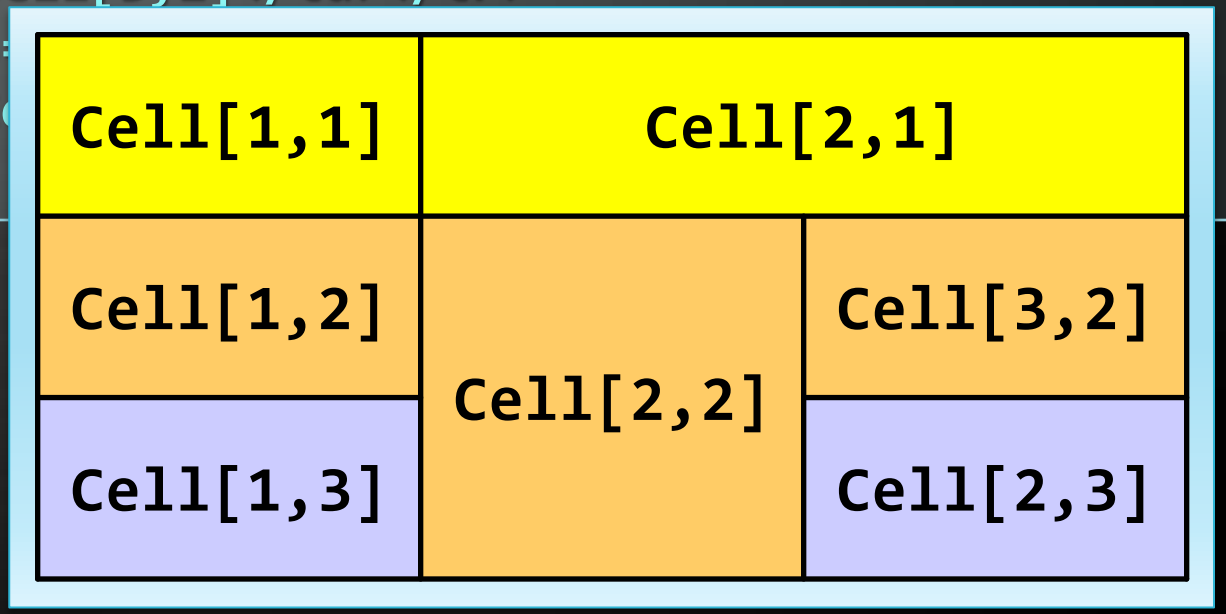
table-colspan-rowspan.html

```
<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=
        <td>C
</table>
```



telerik

HTML Forms

Entering User Data from a Web Page

telerik

HTML Forms

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

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

- Example:

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

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

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

telerik

Form Fields

- 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

telerik

Fieldsets

- 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.

telerik

Form Input Controls

- 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" />
```

telerik

Other Form Controls

- 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" />
```

telerik

Other Form Controls (2)

- Reset button – brings the form to its initial state

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

- Image button – acts like submit but image is displayed and click coordinates are sent

```
<input type="image" src="submit.gif"
name="submitBtn" alt="Submit" />
```

- Ordinary button – used for Javascript, no default action

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

telerik

Other Form Controls (3)

- 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>
```

telerik

Other Form Controls (4)

- 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>
```

telerik

Labels

- 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)
- Labels are both a usability and accessibility feature and are required in order to pass accessibility validation.

telerik

HTML Forms – Example

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" />
```

telerik

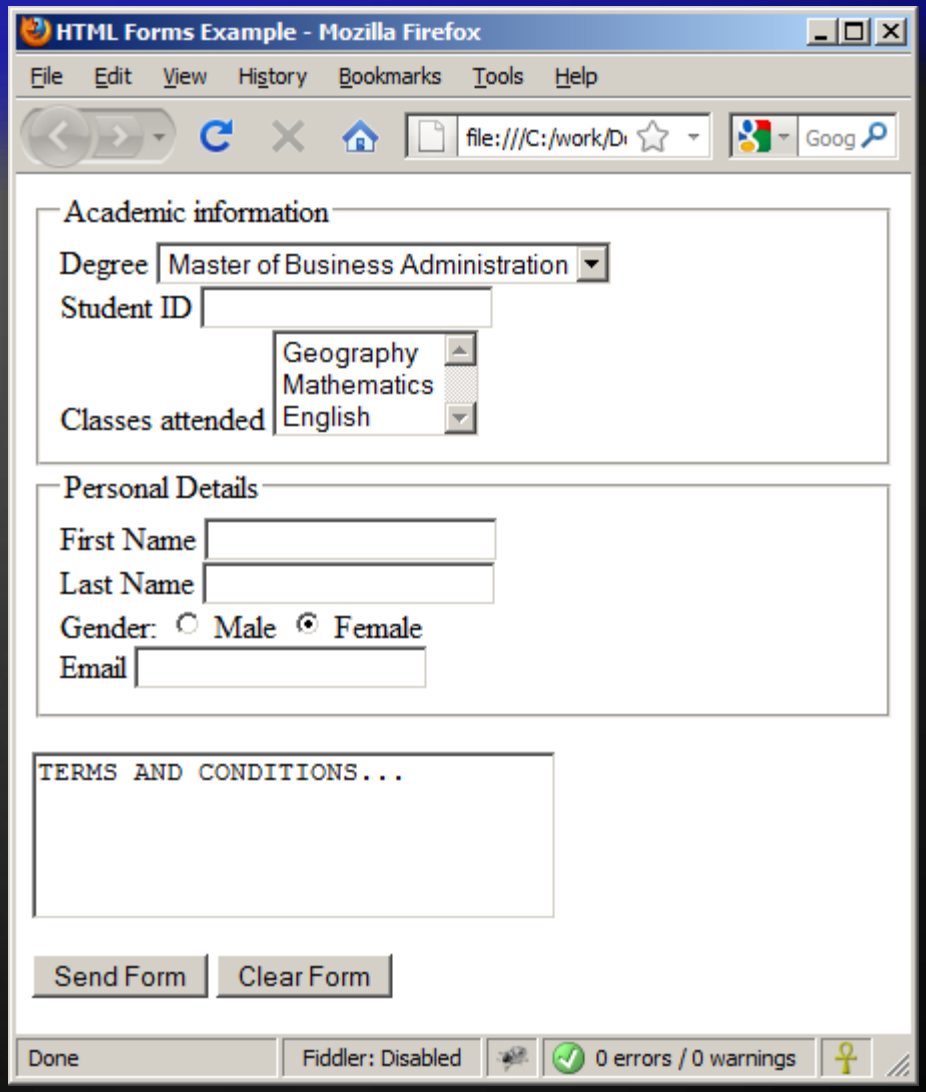
HTML Forms – Example (2)

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)



telerik

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; }
```

telerik

CSS Introduction

- 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.

telerik

CSS Introduction (2)

- CSS can be applied to any XML document
  - Not just to HTML / XHTML
- CSS can specify different styles for different media
  - On-screen
  - In print
  - Handheld, projection, etc.
  - … even by voice or Braille-based reader

Inline Styles: Example

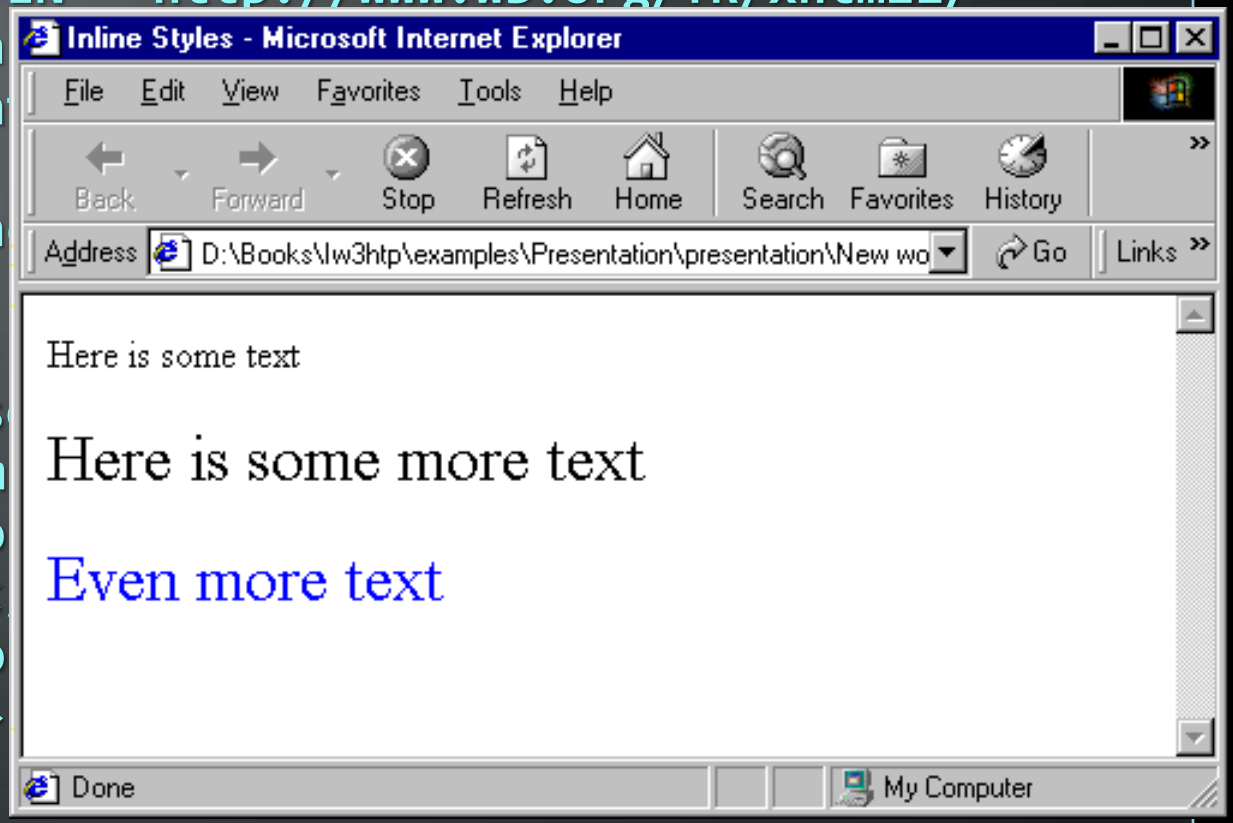
inline-styles.html

```
<!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: Example

inline-styles.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-tra
<html xmlns="h
<head>
  <title>Inlin
</head>
<body>
  <p>Here is s
<!--Separate m
  <p style="fo
    more text<
  <p style="fo
    #0000FF" >
</body>
</html>
```



Embedded Styles: Example

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>
```

External Styles: Example

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: Example (2)

external-styles.html

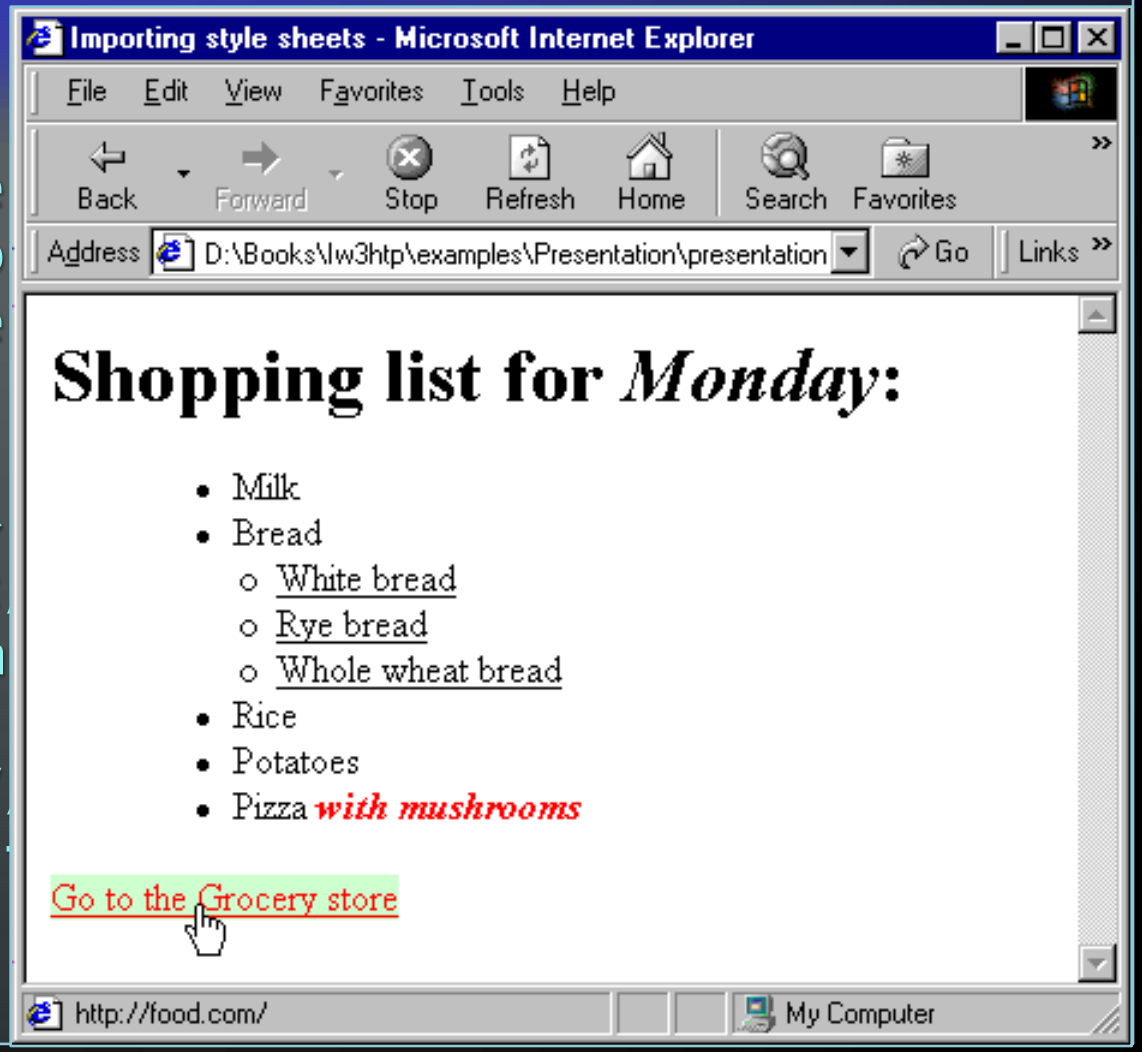
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Importing style sheets</title>
  <link type="text/css" rel="stylesheet"
    href="styles.css"  />
</head>
<body>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

External Styles: Example (3)

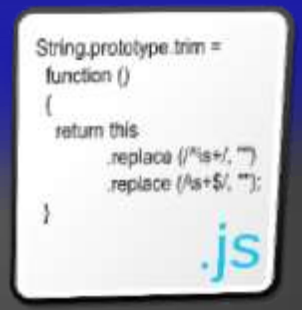
```
  …
  <li>Bread
    <ul>
      <li>White bread</li>
      <li>Rye bread</li>
      <li>Whole wheat bread</li>
    </ul>
  </li>
  <li>Rice</li>
  <li>Potatoes</li>
  <li>Pizza <em>with mushrooms</em></li>
</ul>
<a href="http://food.com" title="grocery
  store">Go to the Grocery store</a>
</body>
</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>
```



telerik



Introduction to JavaScript

telerik

Table of Contents (2)

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

telerik

JavaScript

- 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

telerik

JavaScript Advantages

- 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
  - Implementing AJAX functionality

telerik

JavaScript Limitations

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multi-threading or multiprocessor capabilities.
- Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages

telerik

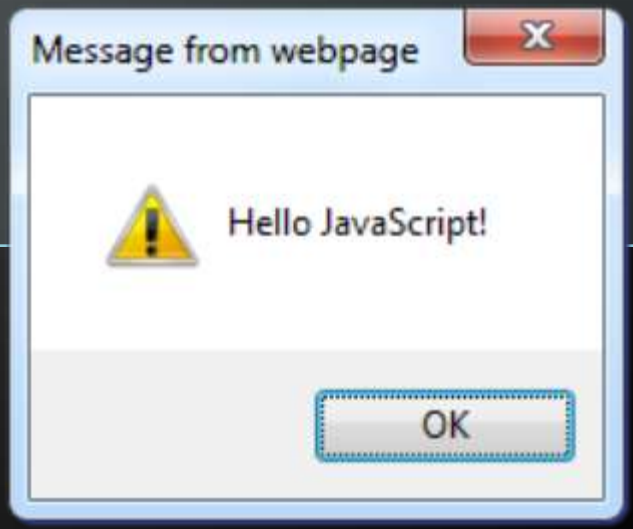
What Can JavaScript Do?

- Can handle events
- Can read and write HTML elements and modify the DOM tree
- 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
- Can perform asynchronous server calls (AJAX)

The First Script

first-script.html

```
<html>

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

</html>
```



Another Small Example

small-example.html

```
<html>

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

</html>
```

telerik

Using JavaScript Code

- 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

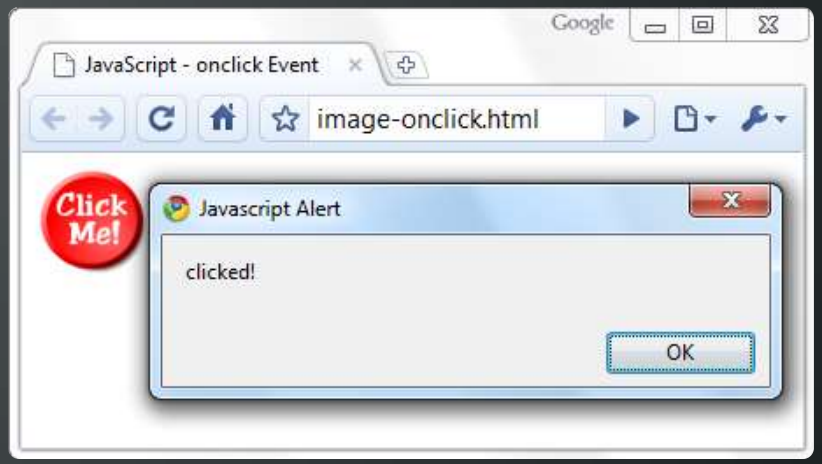
```
<script src="scripts.js" type="text/javscript">
<!- code placed here will not be executed! -->
</script>
```

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

Calling a JavaScript Function from Event Handler – Example

image-onclick.html

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

<body>
  <img src="logo.gif"
    onclick="test('clicked!')" />
</body>
</html>
```



Using External Script Files

- Using external script files:

external-JavaScript.html

```
<html>
<head>
  <script src="sample.js" type="text/javascript">
  </script>
</head>
<body>
  <button onclick="sample()" value="Call JavaScript
    function from sample.js" />
</body>
</html>
```

The <script> tag is always empty.

- External JavaScript file:

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

sample.js

telerik

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));
}
```

telerik

JavaScript Syntax

- 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)

telerik

Control Statements

- 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

telerik

Functions

- 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

- Functions are not required to return a 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

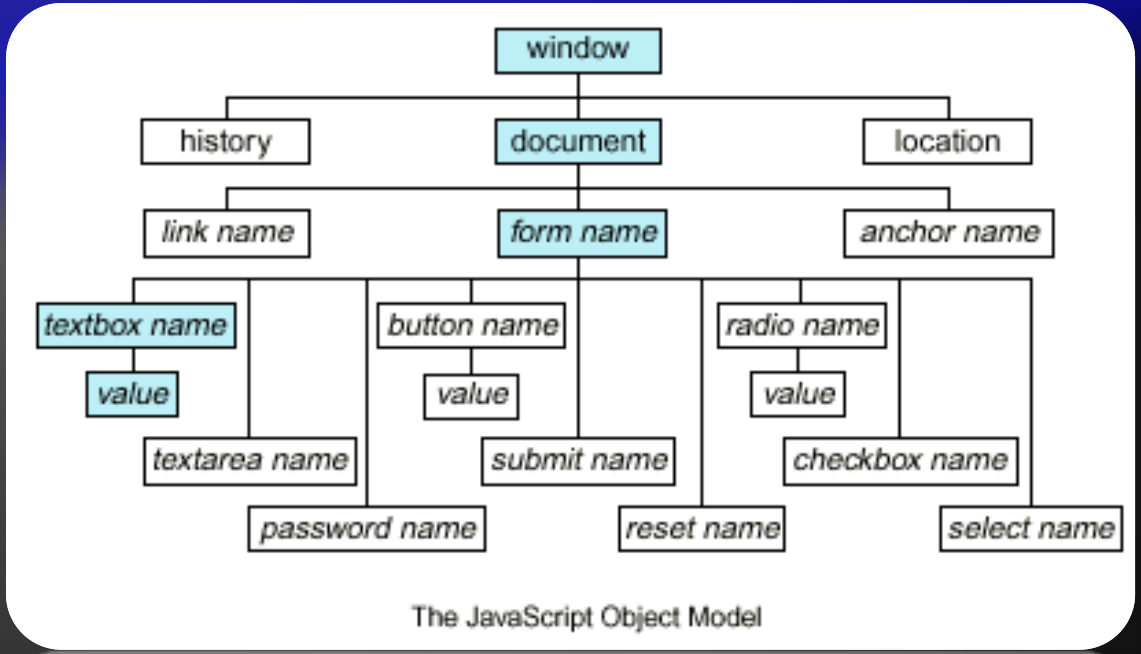
telerik

Events

- When JavaScript is used in HTML pages, JavaScript can **"react"** on these events.

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

telerik



The JavaScript Object Model

Document Object Model (DOM)

Document Object Model (DOM)

- Every HTML element is accessible via the JavaScript DOM API
- Most DOM objects can be manipulated by the programmer
- The event model lets a document to react when the user does something on the page
- Advantages
  - Create interactive pages
  - Updates the objects of a page without reloading it

telerik

Accessing Elements

- Access elements via their ID attribute

```
var elem = document.getElementById("some_id")
```

- Via the name attribute

```
var arr = document.getElementsByName("some_name")
```

- Via tag name

```
var imgTags = el.getElementsByTagName("img")
```

  - Returns array of descendant `<img>` elements of the element "el"

telerik

DOM Manipulation

- Once we access an element, we can read and write its attributes

DOM-manipulation.html

```
function change(state) {
  var lampImg = document.getElementById("lamp");
  lampImg.src = "lamp_" + state + ".png";
  var statusDiv =
    document.getElementById("statusDiv");
  statusDiv.innerHTML = "The lamp is " + state";
}
…
<img src="test_on.gif" onmouseover="change('off')"
  onmouseout="change('on')" />
```

telerik

jQuery Introduction

- jQuery is a lightweight, "write less, do more", JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

telerik

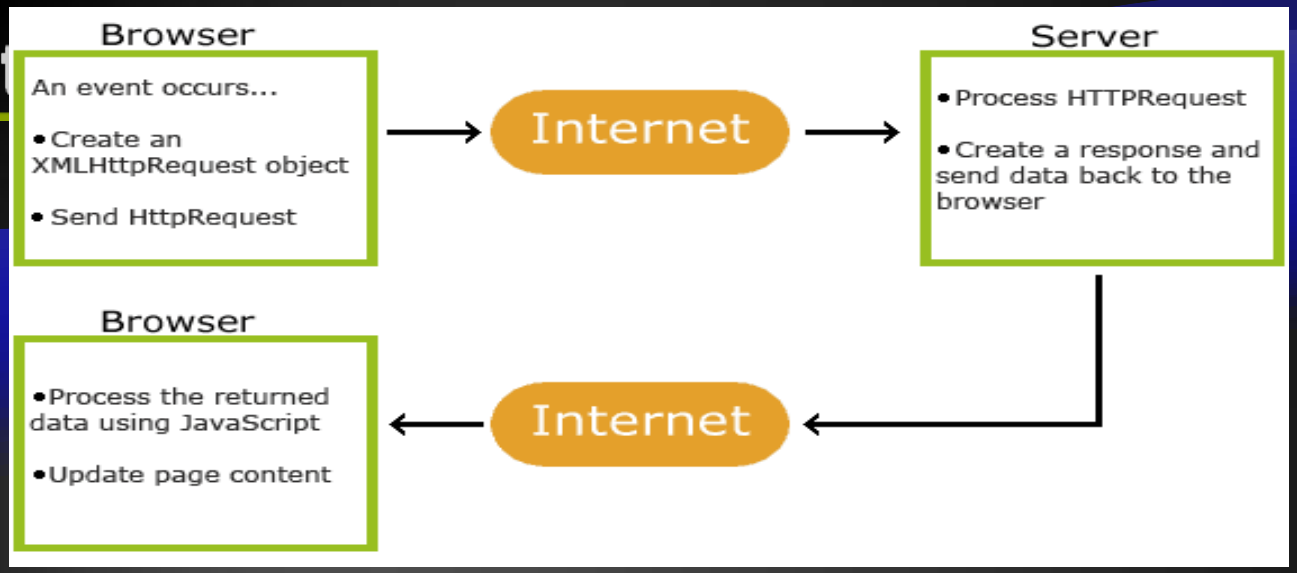
jQuery Introduction

- The jQuery library contains the following features:
- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

telerik

Ajax Introduction

- AJAX = Asynchronous JavaScript and XML.
- In short; AJAX is about loading data in the background and display it on the webpage, without reloading the whole page.
- Examples of applications using AJAX: Gmail, Google Maps, Youtube, and Facebook tabs.



1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

telerik

- AJAX allows web pages to be updated asynchronously by exchanging data with a web server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- AJAX is not a programming language.
- AJAX just uses a combination of:
- A browser built-in XMLHttpRequest object (to request data from a web server) All modern browsers support the XMLHttpRequest object. The XMLHttpRequest object can be used to exchange data with a server behind the scenes.

- JavaScript and HTML DOM (to display or use the data)

telerik

What About jQuery and AJAX?

- jQuery provides several methods for AJAX functionality.
- With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!

- A secure connection is a connection that is encrypted by one or more security protocols to ensure the security of data flowing between two or more nodes. When a connection is not encrypted, it can be easily listened to by anyone with the knowledge on how to do it, or even prone to threats by malicious software and rogue and unexpected events.
- Anyone who wants to get information from a non-secured connection can do so since they can easily go through, in and out of the computer's network taking with them important data such as login, passwords and other private information.

- Secure connections, as they supposed to protect the data being transferred from one computer to another, must be able to do three main things.
- Prevent third parties from getting hold of confidential data
- It must first validate the identification of the person who wishes to access and exchange the data
- It must protect information from being viewed or altered by unknown parties

telerik

- There are many methods to be able to establish a secure connection, but most of them involve data encryption. Data encryption is a method which hides information from other unauthorized parties. This method usually needs an appropriate program installed on both computers involved in the connection that will encrypt and decrypt the information. Among these are our basic security protocols embedded in main communication protocols like TCP/IP, HTTPS, POP3 or IMAP.

telerik

The object model itself closely resembles the structure of the documents it models. For instance, consider this table, taken from an HTML document:

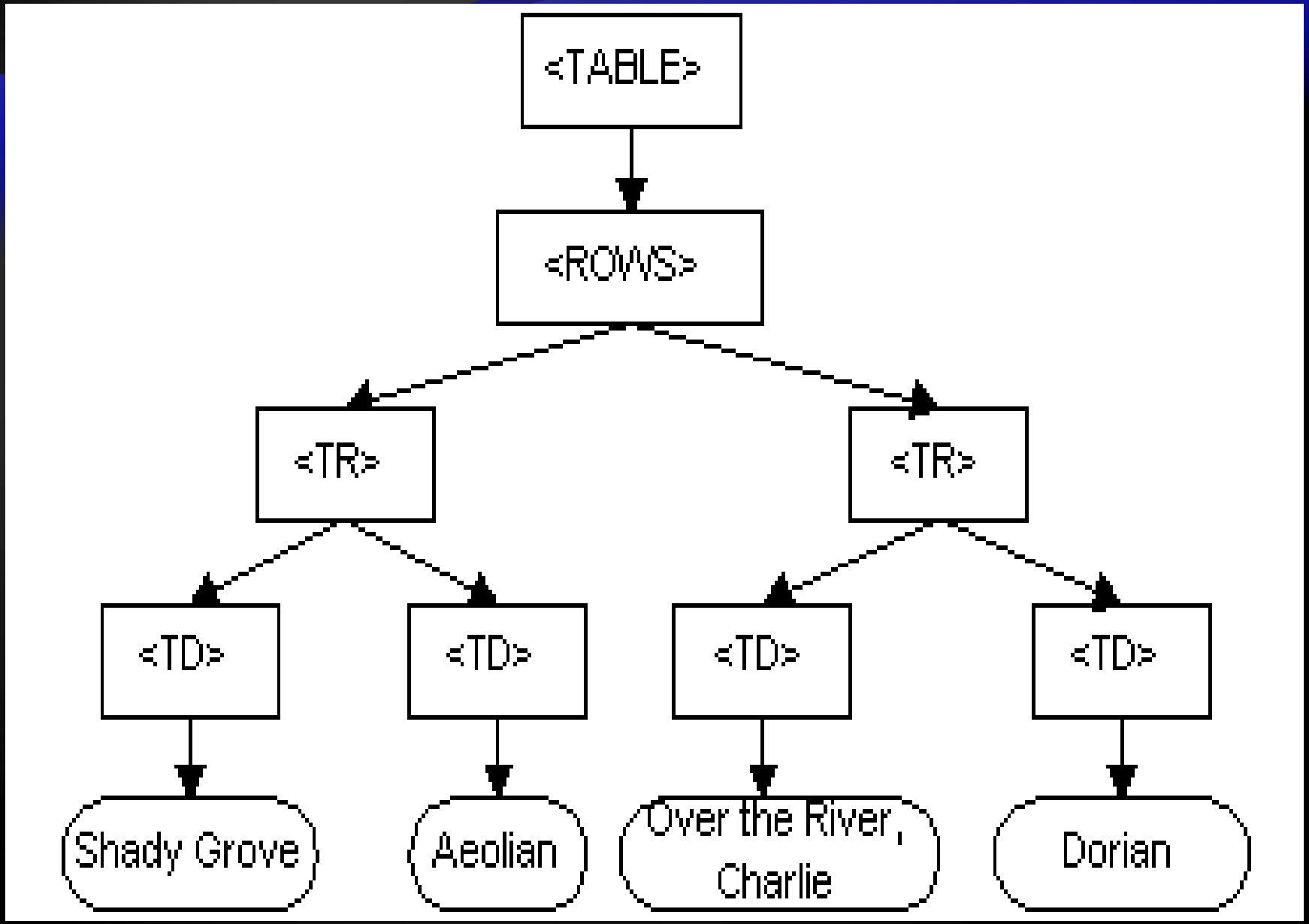
- <TABLE>
- <ROWS>
- <TR>
- <TD>Shady Grove</TD>
- <TD>Aeolian</TD>
- </TR>
- <TR>

<TD>Over the River, Charlie</TD>
<TD>Dorian</TD>

- </TR>

</ROWS>
</TABLE>

telerik


<TABLE>
<ROWS>
<TR>
<TR>
<TD>
<TD>
<TD>
<TD>
Shady Grove
Aeolian
Over the River, Charlie
Dorian

- When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.
- The **HTML DOM** model is constructed as a tree of **Objects**:
- The HTML DOM is a standard **object** model and **programming interface** for HTML. It defines:
- The HTML elements as **objects**
- The **properties** of all HTML elements
- The **methods** to access all HTML elements
- The **events** for all HTML elements
- In other words: **The HTML DOM is a standard for how to get, change, add, or delete HTML elements.**

telerik

The HTML DOM Document Object

- The document object represents your web page.
- If you want to access any element in an HTML page, you always start with accessing the document object.
- Below are some examples of how you can use the document object to access and manipulate HTML.

telerik

Finding HTML Elements

Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name