Web Technology

Unit -1

Learning Objective

- Overview of Internet and web
- HTML Tags, Forms & Frames
- Introduction to Java Script and
- Cascading Style Sheets
- DHTML
- Using various Web Design Tools like Dream Weaver

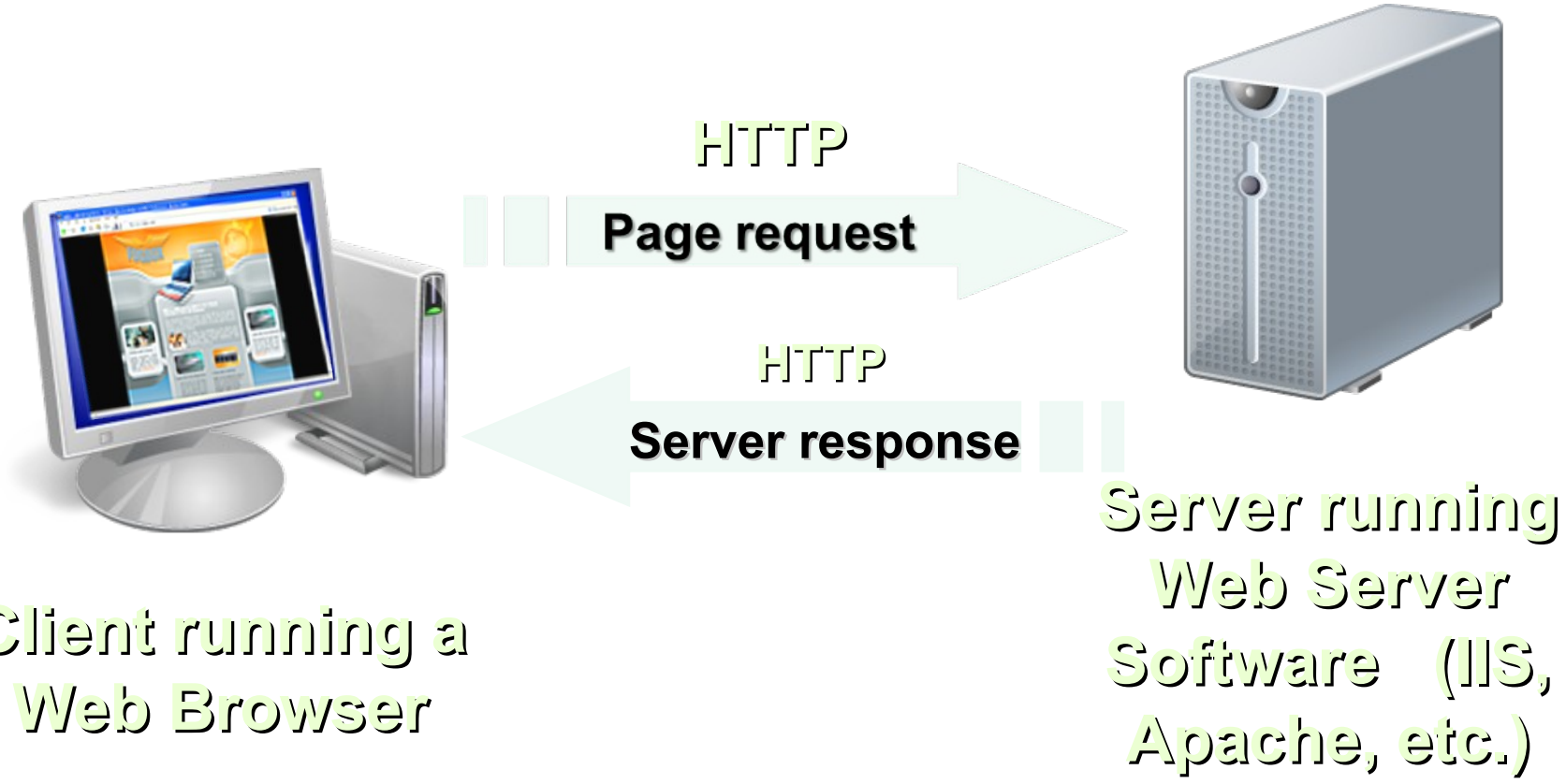
Table of Contents

Table of Contents (2)

2. HTML in Details
   - The <body> Section
   - Text Styling and Formatting Tags
   - Hyperlinks: <a>, Hyperlinks and Sections
   - Images: <img>
   - Lists: <ol>, <ul> and <dl>
- The and elements
- HTML Tables
- HTML Forms

How the Web Works?

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



What is a Web Page?

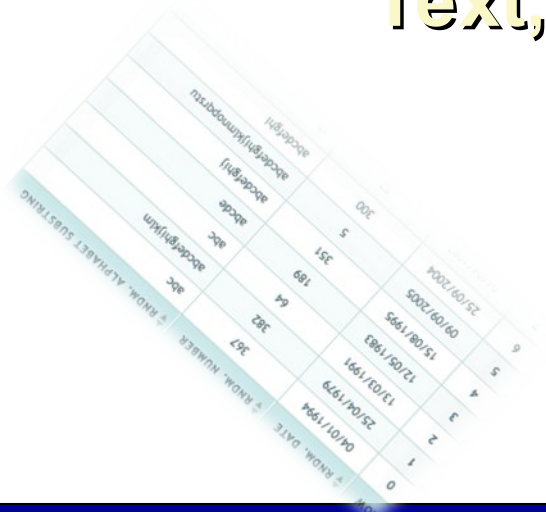
- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
  - Looks (looked?) like:
    - A Microsoft Word document
- The markup tags provide information about the page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

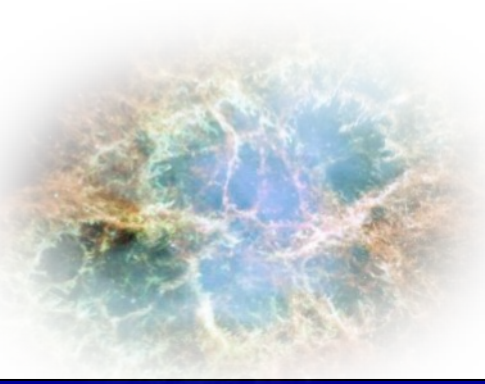
Creating HTML Pages

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

HTML Basics

Text, Images, Tables, Forms





HTML Structure

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

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

- Tags have attributes:

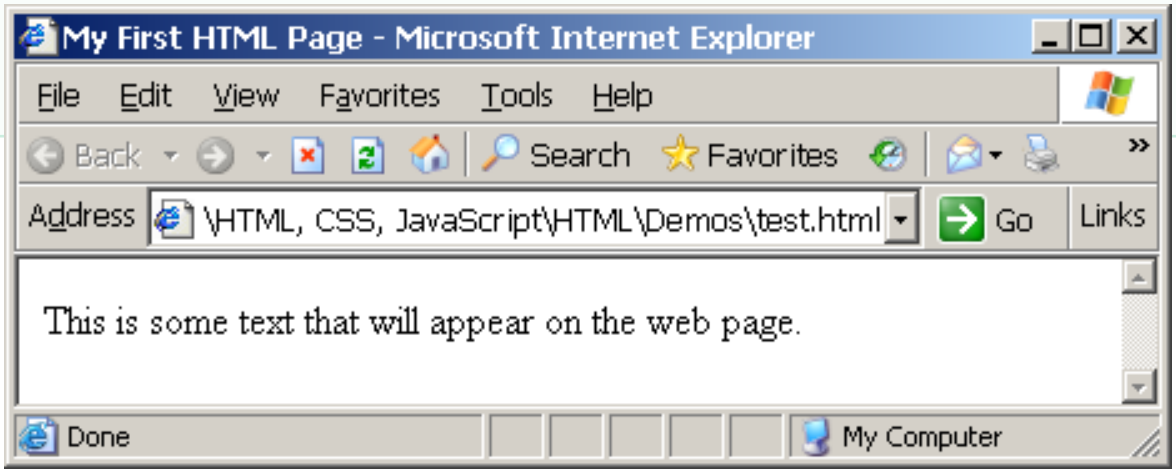
```
<img src="logo.jpg" alt="logo" />
```

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

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.

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

What types of information does <he

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

What type of information does

Some Simple Tags

- Hyperlink Tags

```
<a href="http://www.bvicam.in"
  title="BVICAM">Link to BVICAM 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

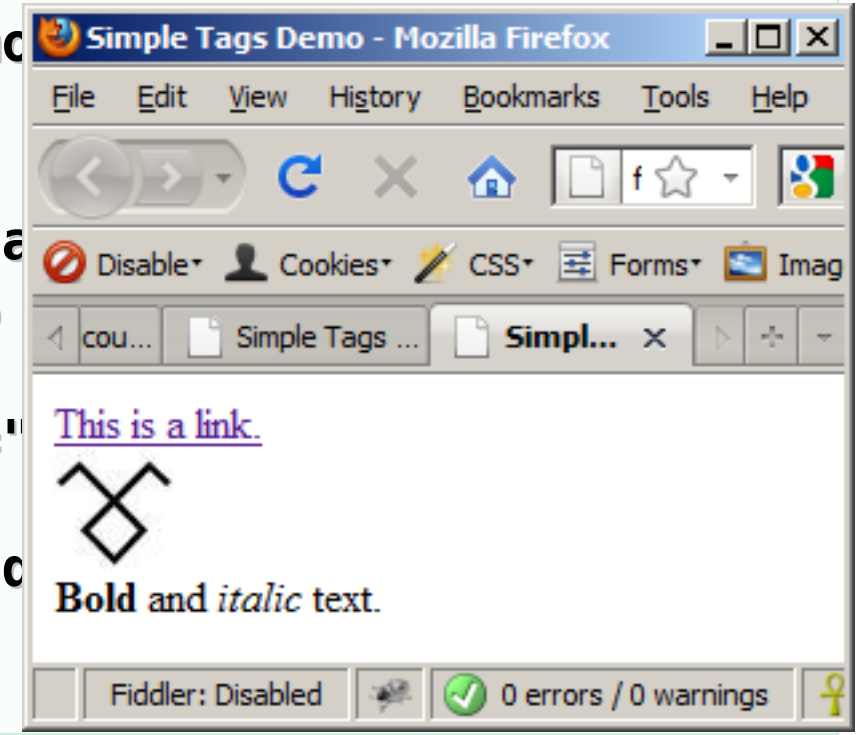
some-tags.html

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Simple Tags Demo</title>
</head>
<body>
<a href="http://www.bvicam.in"
  title="BVICAM">Link to BVICAM Web site</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
</head>
<body>
<a href="http://www.bvica
  title="BVICAM">Link to
<br />
<img src="logo.gif" alt="
<br />
<strong>Bold</strong> and
</body>
</html>
```



Tags Attributes

- Tags can have attribute
  - Attributes specify proper
  - Example:

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

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

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

```
<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>
```

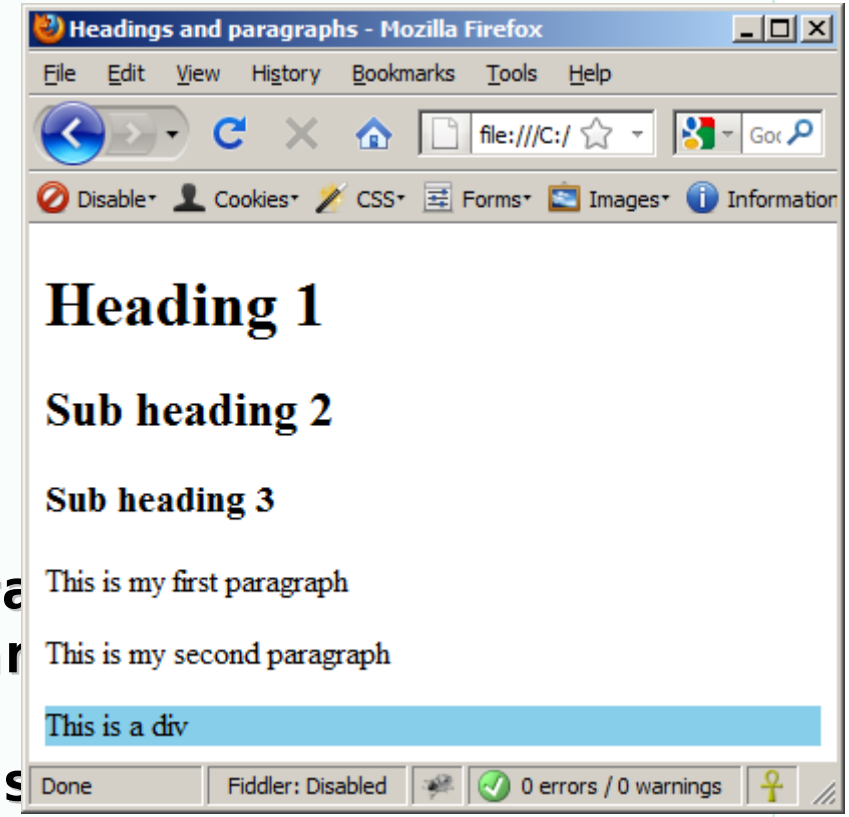
Headings and Paragraphs – Example (2)

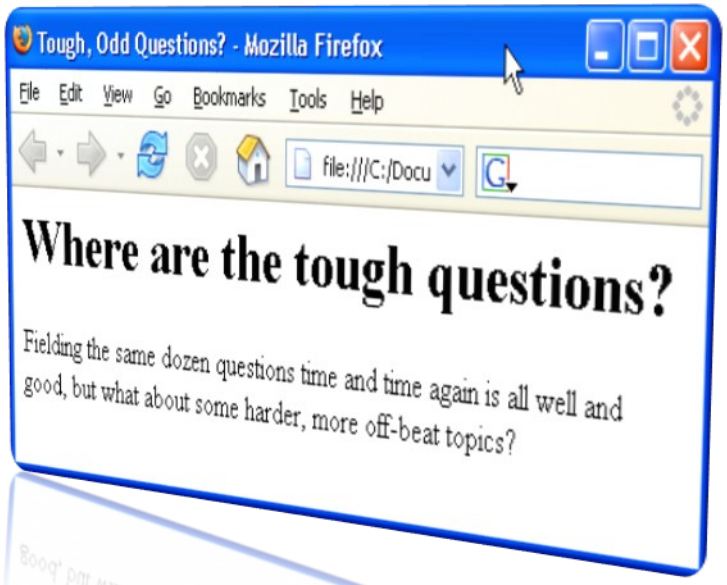
headings.html

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

    <p>This is my first para
    <p>This is my second par

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





Introduction to HTML

HTML Document Structure in Depth

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

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)

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!

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
- Many Search engines and people rely on titles

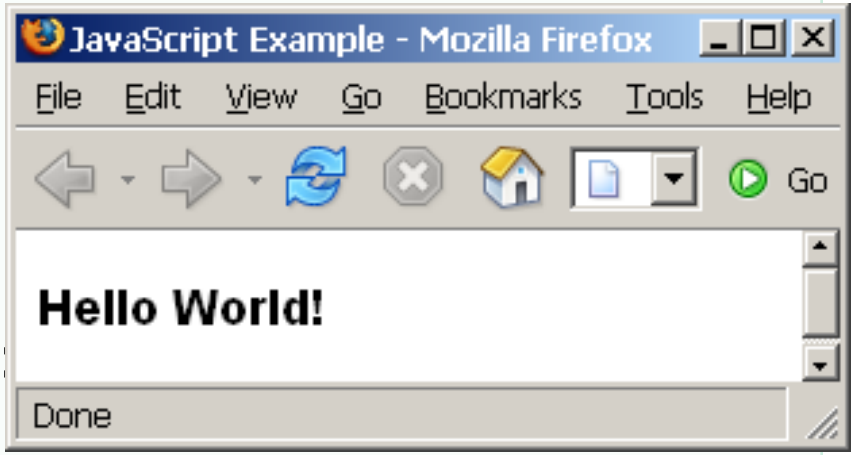
<head> Section: <script>

- 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
  - VBScript

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>
```

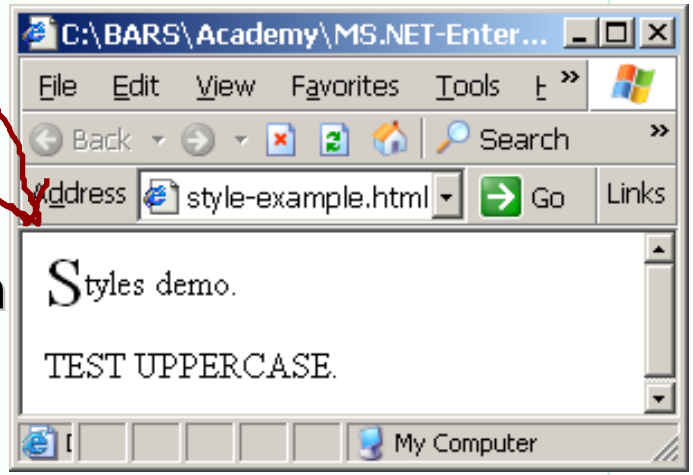


<head> Section: <style>

- 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</spa
    </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">
...
```

<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>
```

Text Formatting

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

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

Text Formatting – Example

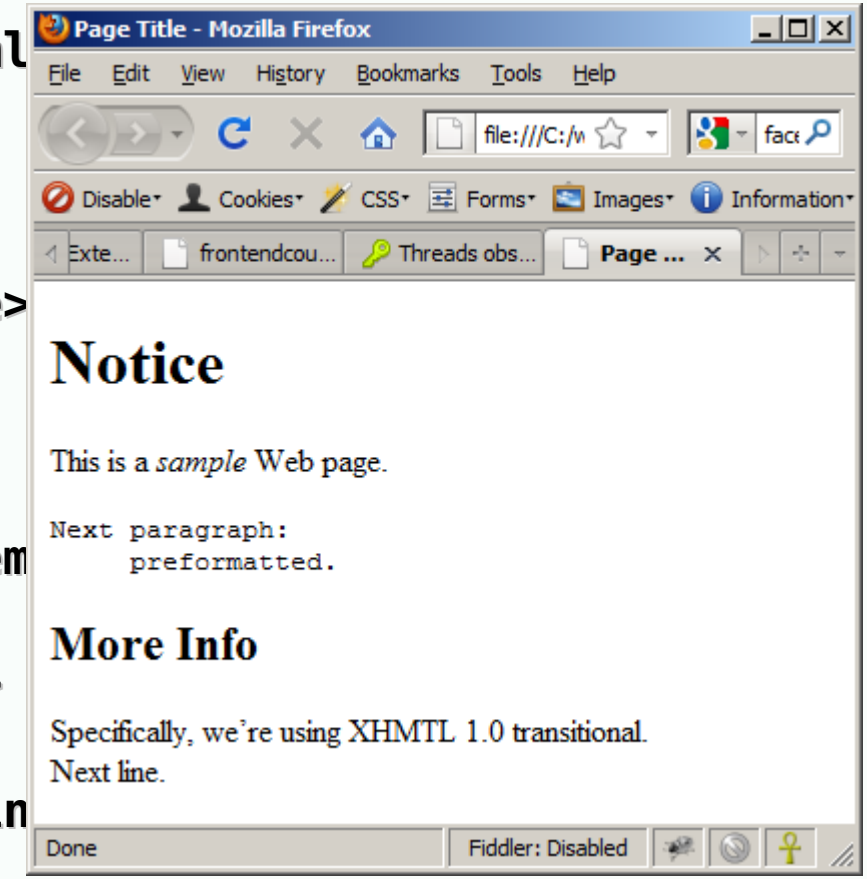
text-formatting.html

```
<!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/xhtml
transitional.dtd">
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
  <h1>Notice</h1>
  <p>This is a <em>sample</em>
  <p><pre>Next paragraph:
     preformatted.</pre></p>
  <h2>More Info</h2>
  <p>Specifically, we're usin
/>
    Next line.</p>
  </body>
</html>
```



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>
```

Hyperlinks: <a> Tag (2)

- Link to an external Web site:

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

  - Always include "http://", not just "www.somesite.com"
  - Using the target="_blank" attribute opens the link in a new window

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>
```

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>
```

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" />
```

Miscellaneous Tags

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

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

- : Deprecated!

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

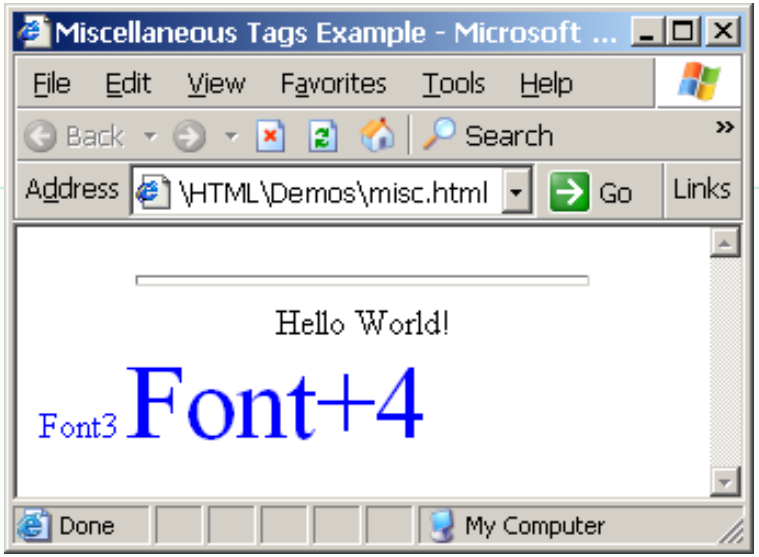
- : Deprecated!

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

Miscellaneous Tags – Example

misc.html

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

1. Apple
2. Orange
3. Grapefruit

A. Apple
B. Orange
C. Grapefruit

a. Apple
b. Orange
c. Grapefruit

I. Apple
II. Orange
III. Grapefruit

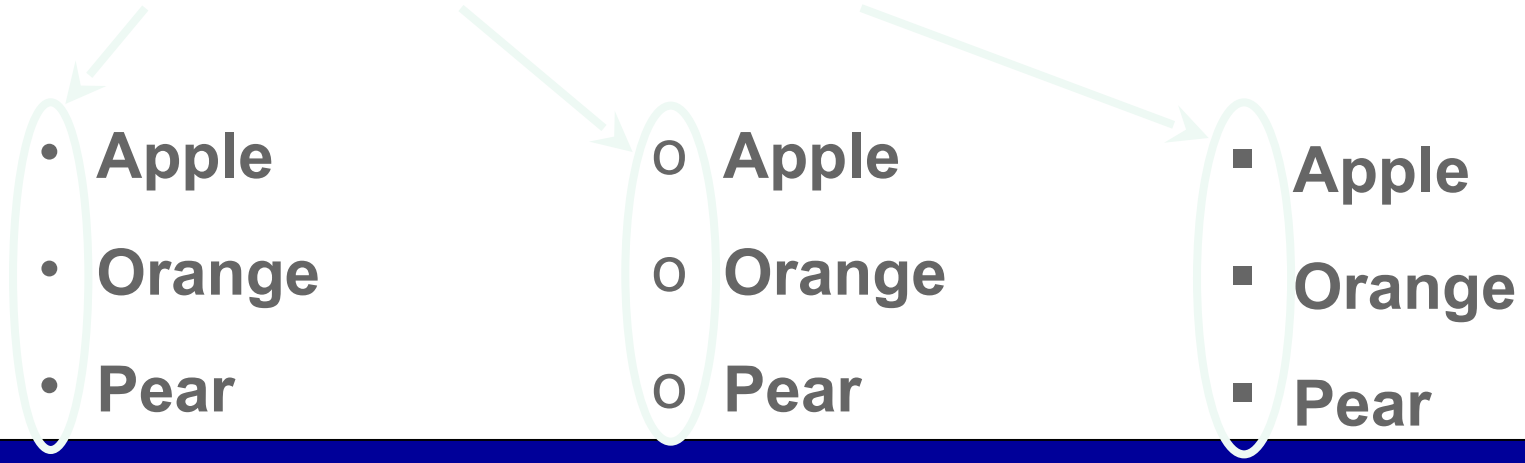
i. Apple
ii. Orange
iii. Grapefruit

Unordered Lists: <ul> Tag

- Create an Unordered List using `<ul></ul>`:

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

- Attribute values for `type` are:
  - `disc`, `circle` or `square`



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

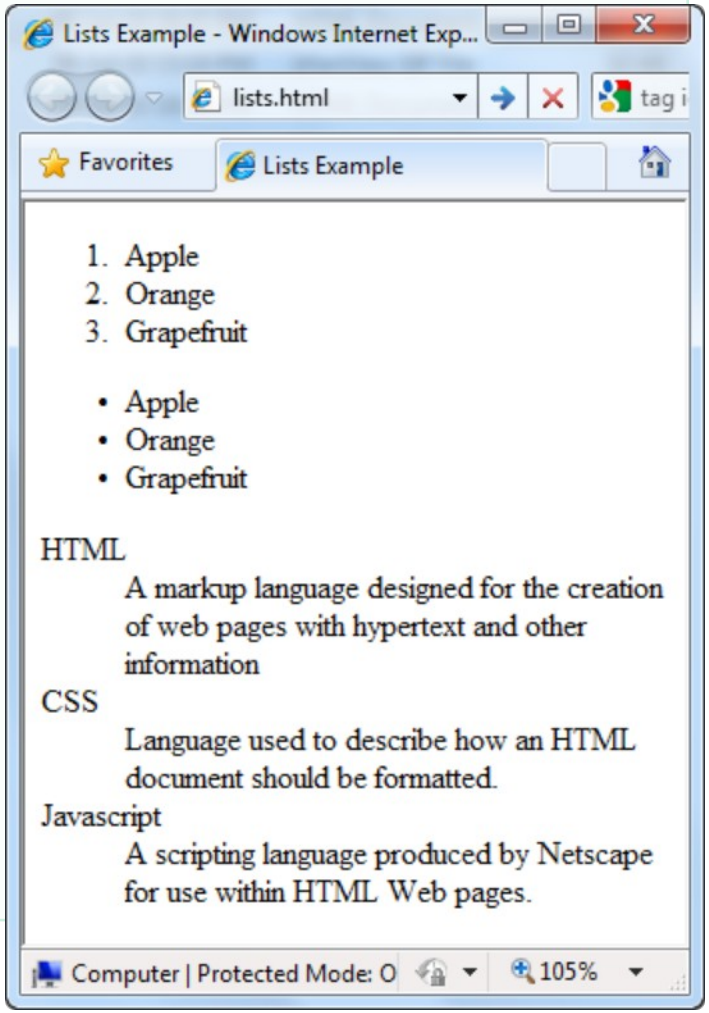
Lists – Example

```
<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	&copy;	©
Registered Trademark Sign	&reg;	®
Trademark Sign	&trade;	™
Less Than	&lt;	<
Greater Than	&gt;	>
Ampersand	&amp;	&
Non-breaking Space		
Em Dash	&mdash;	—
Quotation Mark	&quot;	"
Euro	&#8364;	€
British Pound	&pound;	£
Japanese Yen	&yen;	¥

Special Characters – Example

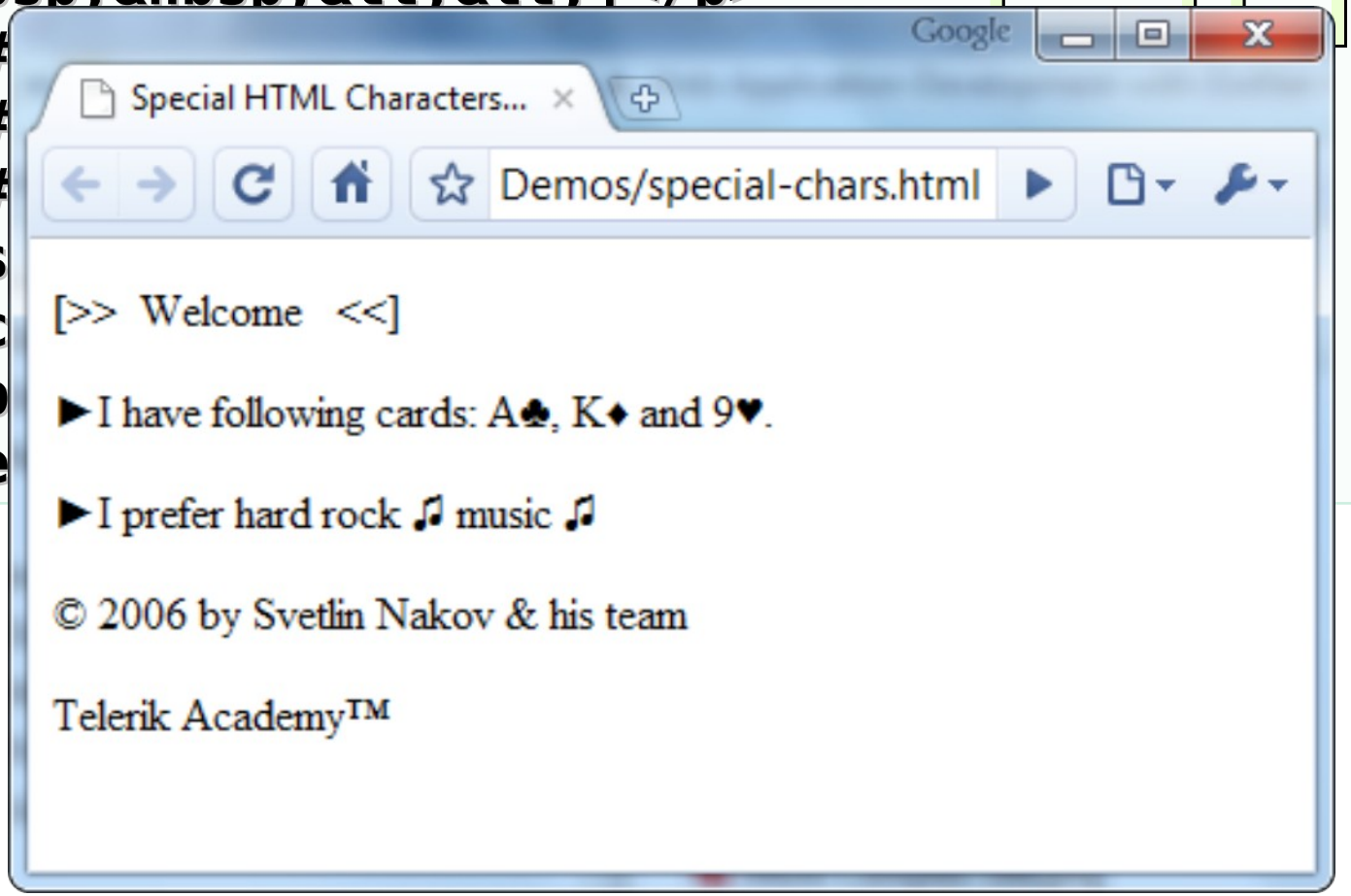
special-chars.html

```
<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)

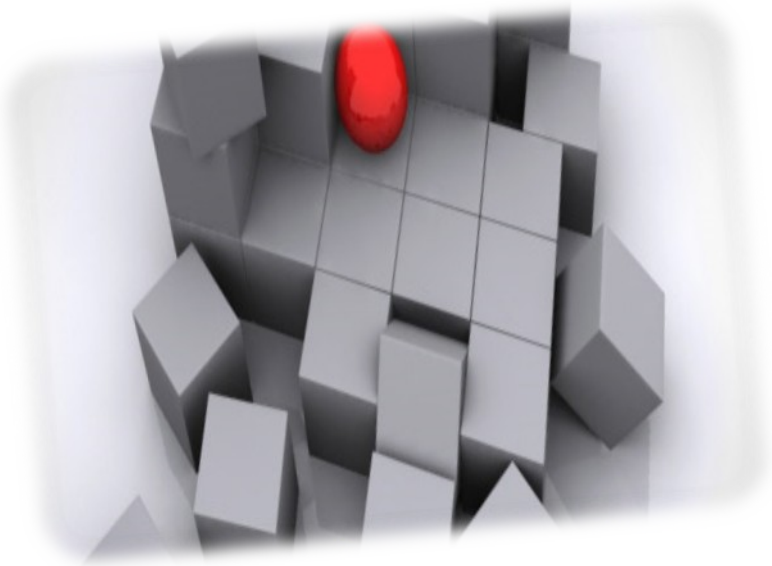
```
<p>[&gt;&gt;  Welcome
      &lt;&lt;]</p>
  <p>&#
    A&#
  <p>&#
    mus
  <p>&c
team</p
  <p>Te
```

special-



```
<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 p
separate license to use the OpenCube
```



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

Block and Inline Elements

- Block elements add a line break before and after them
  - 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>

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>
```

US time	European date (D/M/Y) & time	Y-M-D date & time	Dollar	Chinese money	IP addresses	Names	Numbers
	29/10/1965	83-03-24		YMB 4	98.176.35.80		26.32 E +03
Fri Mar 22 21:48:49 UTC+0200 1957		1967-08-22 06:07:16 PM		YMB -81.38	162.117.253.34	dyse chidi	
Thu, 14 Feb 2002 04:24:20 UTC	06/07/99 06:46:01 AM	81-02-04 09:09:54 AM		YMB -108.83	122.205.50.6	bochai dychai	-191.45E-05
Monday, May 30, 1994 4:47:31 PM	06/09/05 05:11:16 AM			YMB 33.16		dydy balie	-131.20E+01
09/28/2000	24/11/1957		$-38.77	YMB 112.42	15.192.151.209		
		97-08-13 00:01:33 AM	$14.5	YMB -1.75	99.93.147.150	dychai tonchai	-187.28E-05
Mon, 29 Oct 1979 00:44:03 UTC		87-10-16	$14.66	YMB 61.14		chite malie	- 125.19 E -03
Sat, 9 Jan 1982 05:45:06 UTC	04/06/68	74-10-20	$20.47		121.169.225.22	dyma bama	138.11E+02
04/05/75		2000-03-20	$68.84	YMB 88.19	239.133.227.68	made liete	195.44 E +03
Monday, July 15, 2002 1:05:02 AM	01/02/1961 09:40:16 AM		$97.9	YMB 44.28	223.66.228.116	mava sete	-107
this is footer	row	number	ONE!	adsf	adsf	adsf	adsf

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

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

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

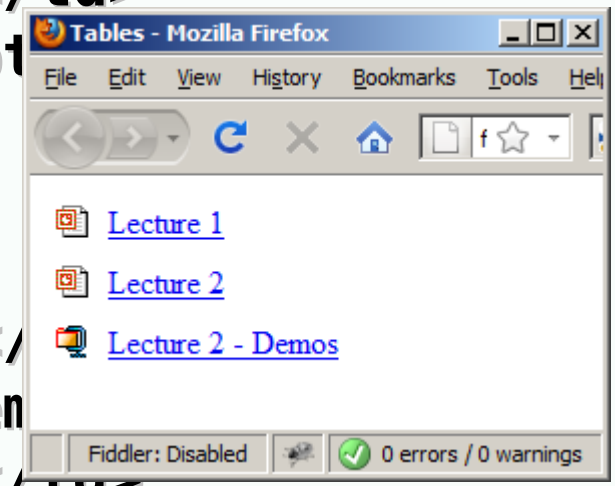
```
<td> ... </td>
```

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
2</a></td>
  </tr>
  <tr>
    <td><img src="zip.gif"></
    <td><a href="lecture2-dem
      Lecture 2 - Demos</a></td>
  </tr>
</table>
```



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
  <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
</tbody>
</table>
```

columns

header

th

footer

Last comes the body (data)

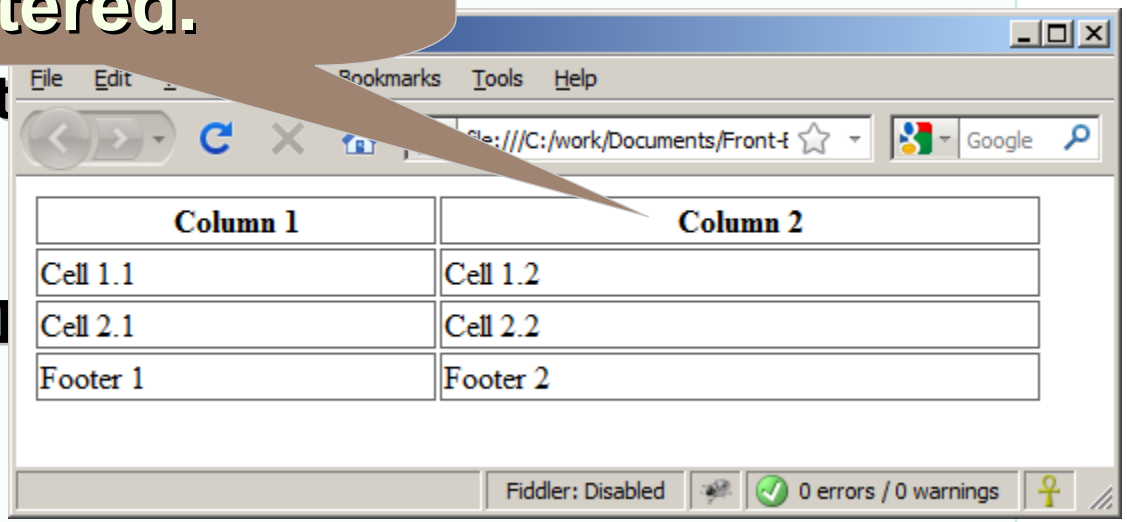
Complete HTML Table:

e (2)

table-full.html

```
<table
<colgr
  <col style="widt
</colgroup>
<thead>
  <tr><th>Column 1
</thead>
<tfoot>
  <tr><td>Footer 1</td><td>Footer 2</td></tr>
</tfoot>
<tbody>
  <tr><td>C                                    </td></tr>
  <tr><td>C                                    </td></tr>
</tbody>
</table>
```

By default, header text is bold and centered.



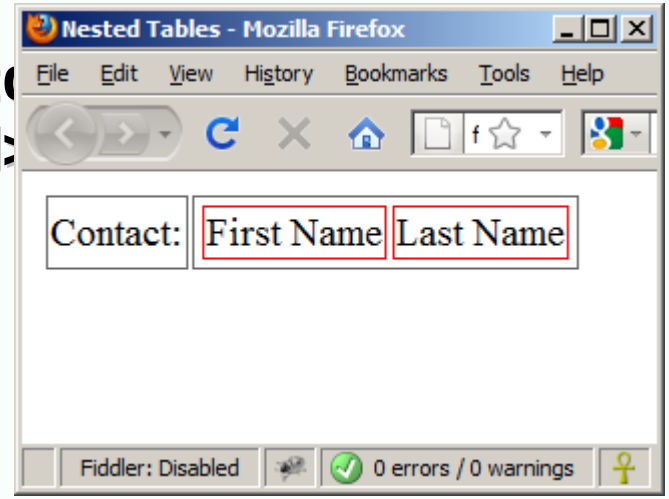
Although the footer is before the data in the code, it is displayed last

Nested Tables

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

nested-tables.html

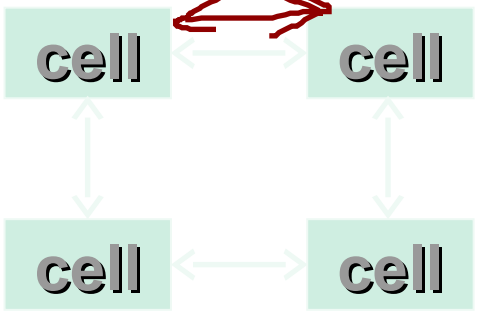
```
<table>
  <tr>
    <td>Contact:</td>
    <td>
      <table>
        <tr>
          <td>First Name</td>
          <td>Last Name</td>
        </tr>
      </table>
    </td>
  </tr>
</table>
```



Cell Spacing and Padding

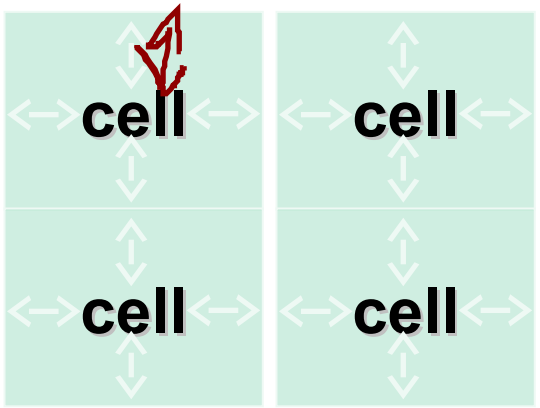
- Tables have two important attributes:

- **`cellspacing`**



- **Defines the empty space between cells**

- **`cellpadding`**



- **Defines the empty space around the cell content**

Cell Spacing and Padding – Example

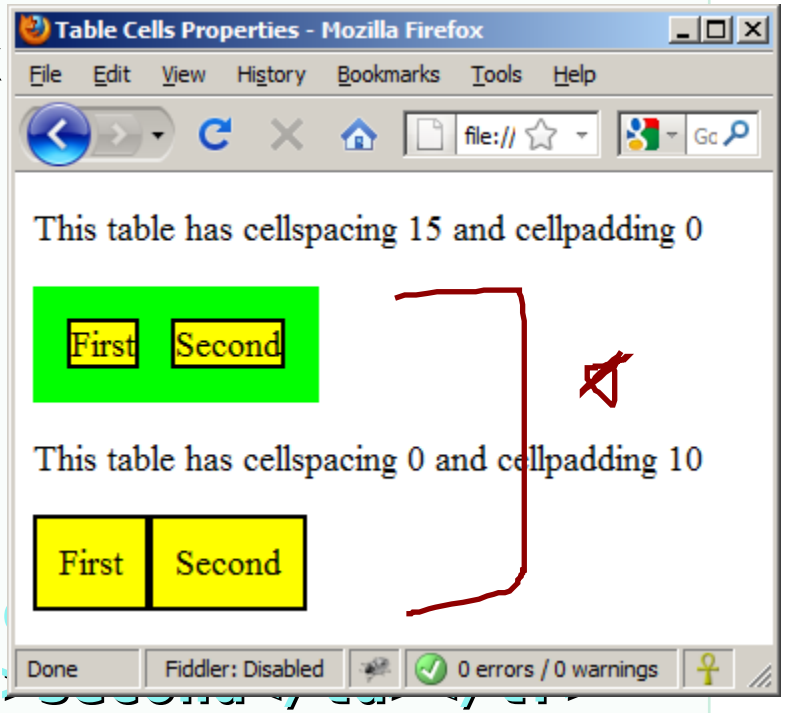
table-cells.html

```
<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)

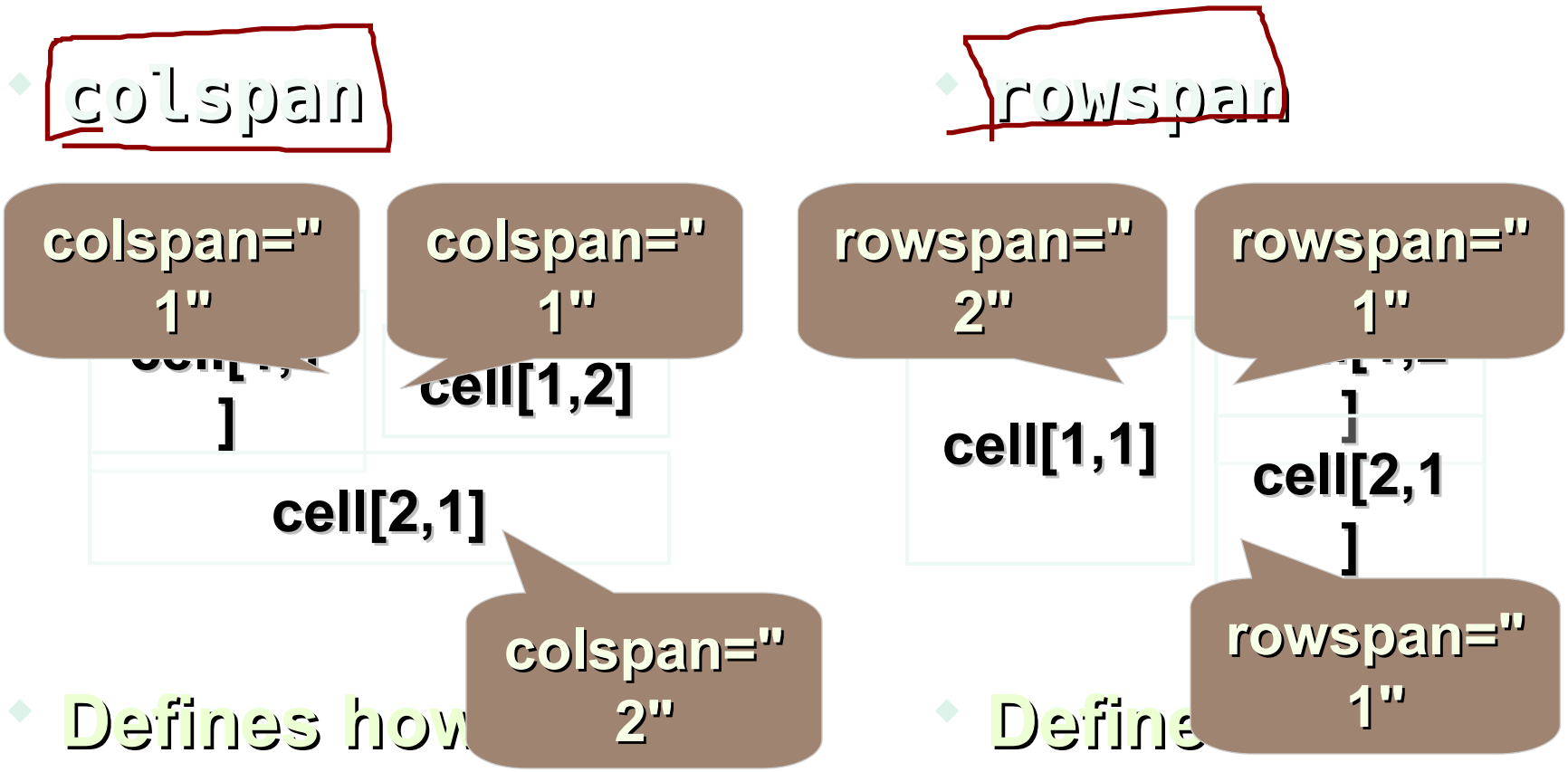
table-cells.html

```
<html>
  <head><title>Table Cells</title>
<style>
table, th, td {
    border: 1px solid black
</head>
  <body>
    <table cellspacing="15"
      <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
  - rowspan



  - Defines how many columns the cell occupies
  - Defines how many rows the cell occupies

Column and Row Span – Example

table-colspan-rowspan.html

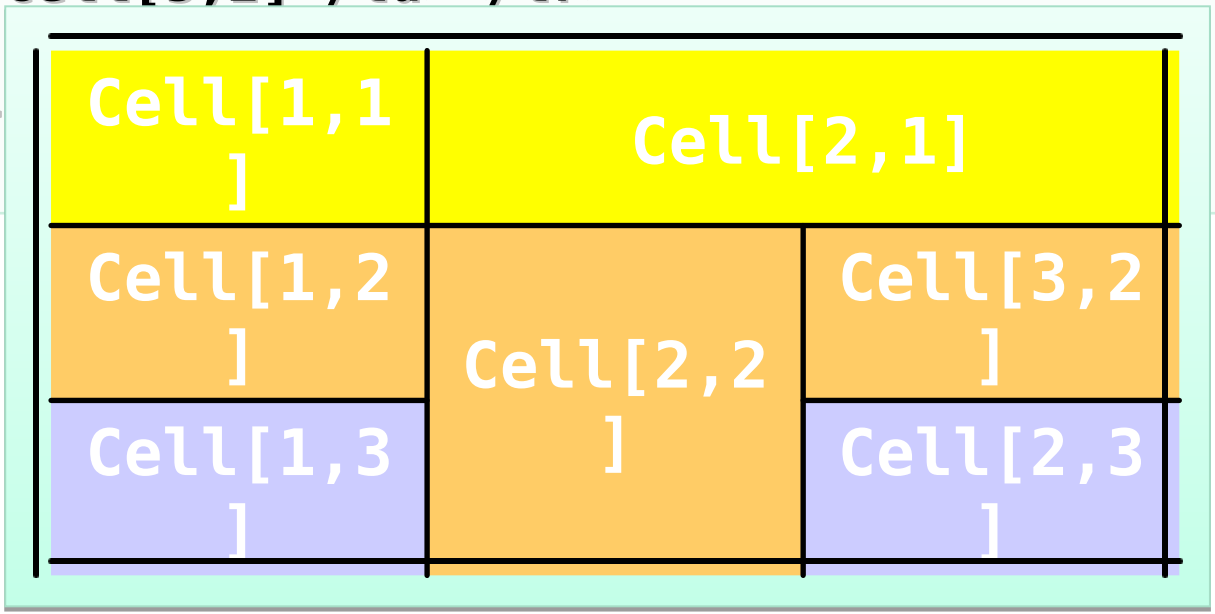
```
<table cellspacing="0">
    <tr class="1"><td>Cell[1,1]</td>
        <td colspan="2">Cell[2,1]</td></tr>
    <tr class="2"><td>Cell[1,2]</td>
        <td rowspan="2">Cell[2,2]</td>
        <td>Cell[3,2]</td></tr>
    <tr class="3"><td>Cell[1,3]</td>
        <td>Cell[2,3]</td></tr>
</table>
```

What is default colspan & rowspan va

Column and Row Span – Example (2)

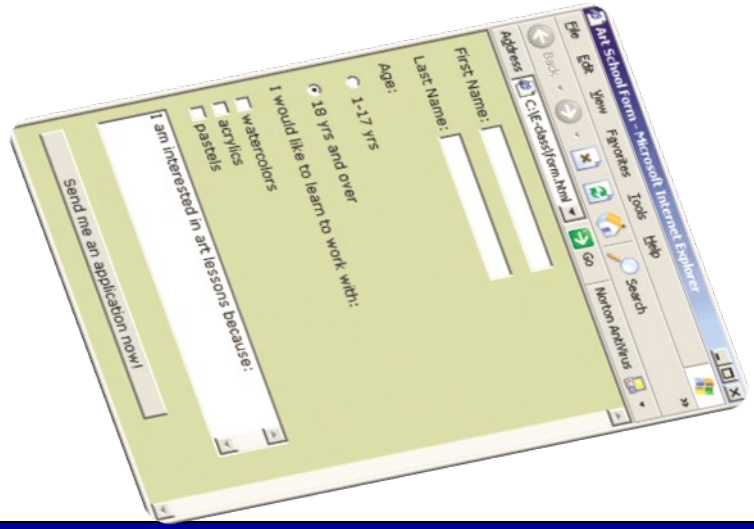
table-colspan-rowspan.html

```
<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 clas
        <td>
</table>
```



HTML Forms

Entering User Data from a Web Page



HTML Forms

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

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

- Example:

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

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

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

Form Fields

- Single-line text input fields:

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

- Multi-line textarea fields:

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

- Hidden fields contain data not shown to the user:

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

- Often used by JavaScript code

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.

Form Input Controls

- Checkboxes:

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

- Radio buttons:

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

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

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

Other Form Controls

- Dropdown menus:

```
<select id="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" id="submitBtn"
value="Apply Now" />
```

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

```
<button type="button"
onclick="myFunction()">Try it</button>
```

Other Form Controls (3)

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

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

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

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

Other Form Controls (4)

- File input – a field used for uploading files

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

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

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

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.

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" />
```

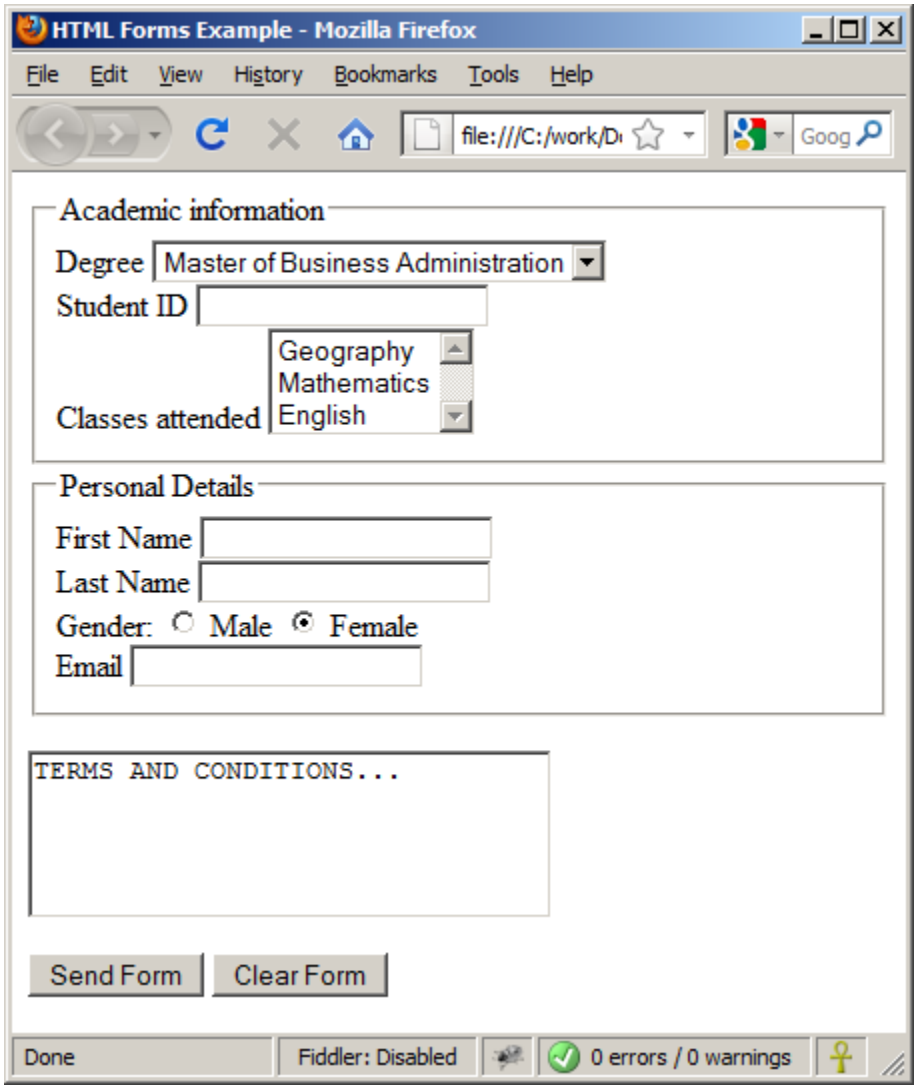
HTML Forms – Example (2)

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>
```

HTML Forms – Example (3)

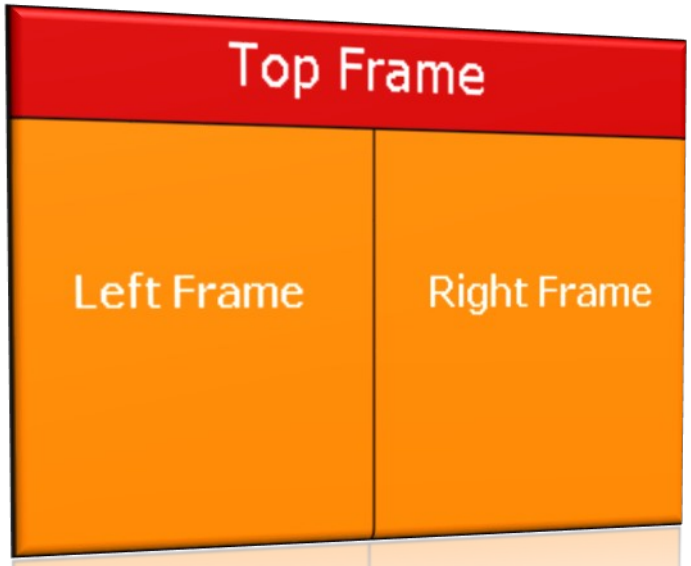
form.html (continued)

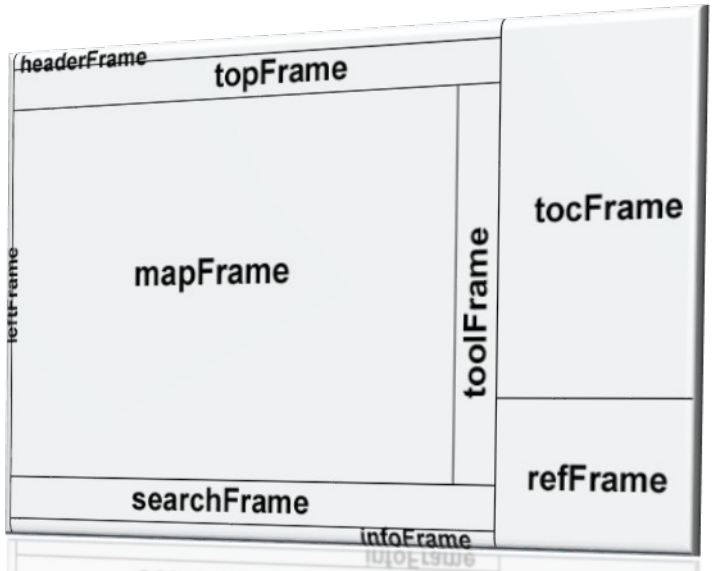


TabIndex

- The tabindex HTML attribute controls the order in which form fields and hyperlinks are focused when repeatedly pressing the TAB key
  - tabindex="0" (zero) - "natural" order
  - If X > Y, then elements with tabindex="X" are iterated before elements with tabindex="Y"
  - Elements with negative tabindex are skipped, however, this is not defined in the standard

```
<input type="text" tabindex="10" />
```





HTML Frames

<frameset>, <frame> and <iframe>

HTML Frames

- 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 are not supported by all user agents (browsers, search engines, etc.)
  - A <noframes> element is used to provide content for non-compatible agents.

HTML Frames – Demo

frames.html

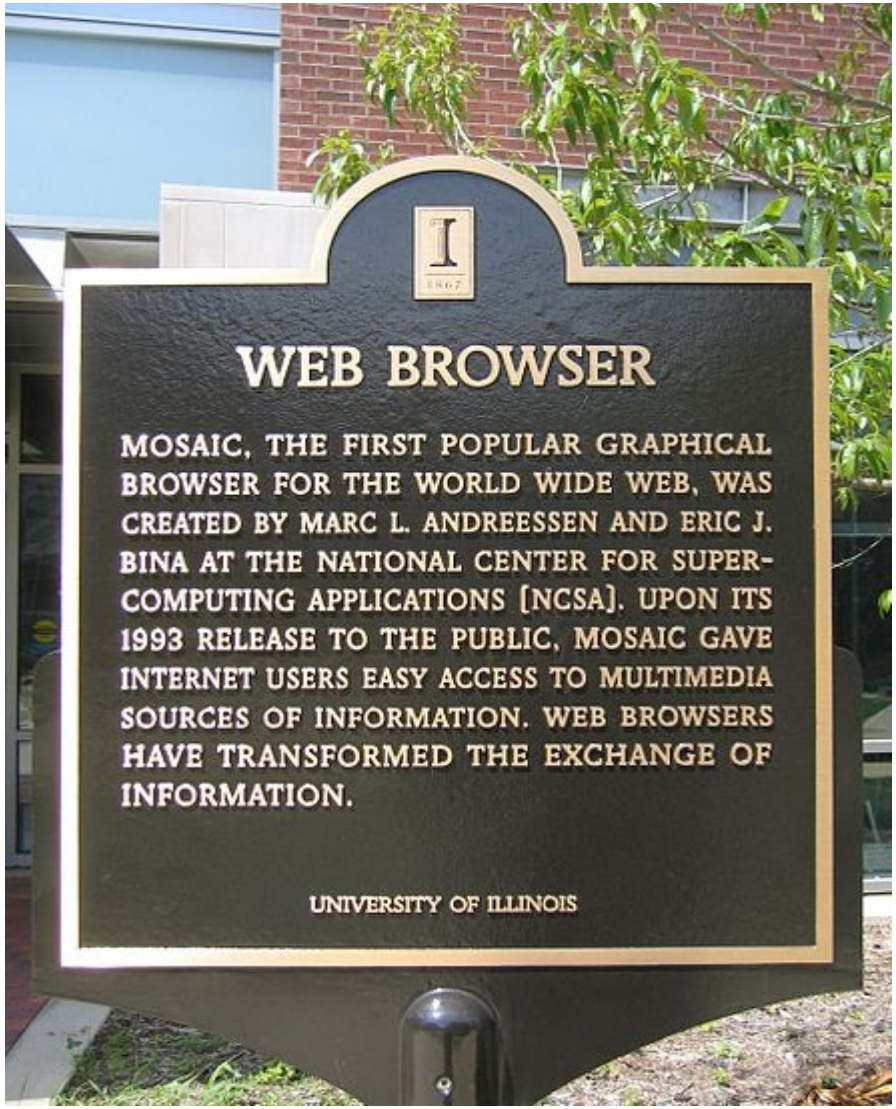
```
<html>
  <head><title>Frames Example</title></head>
  <frameset cols="180px,*,150px">
    <frame src="left.html" />
    <frame src="middle.html" />
    <frame src="right.html" />
  </frameset>
</html>
```

- **Note the target attribute applied to the <a> elements in the left frame.**

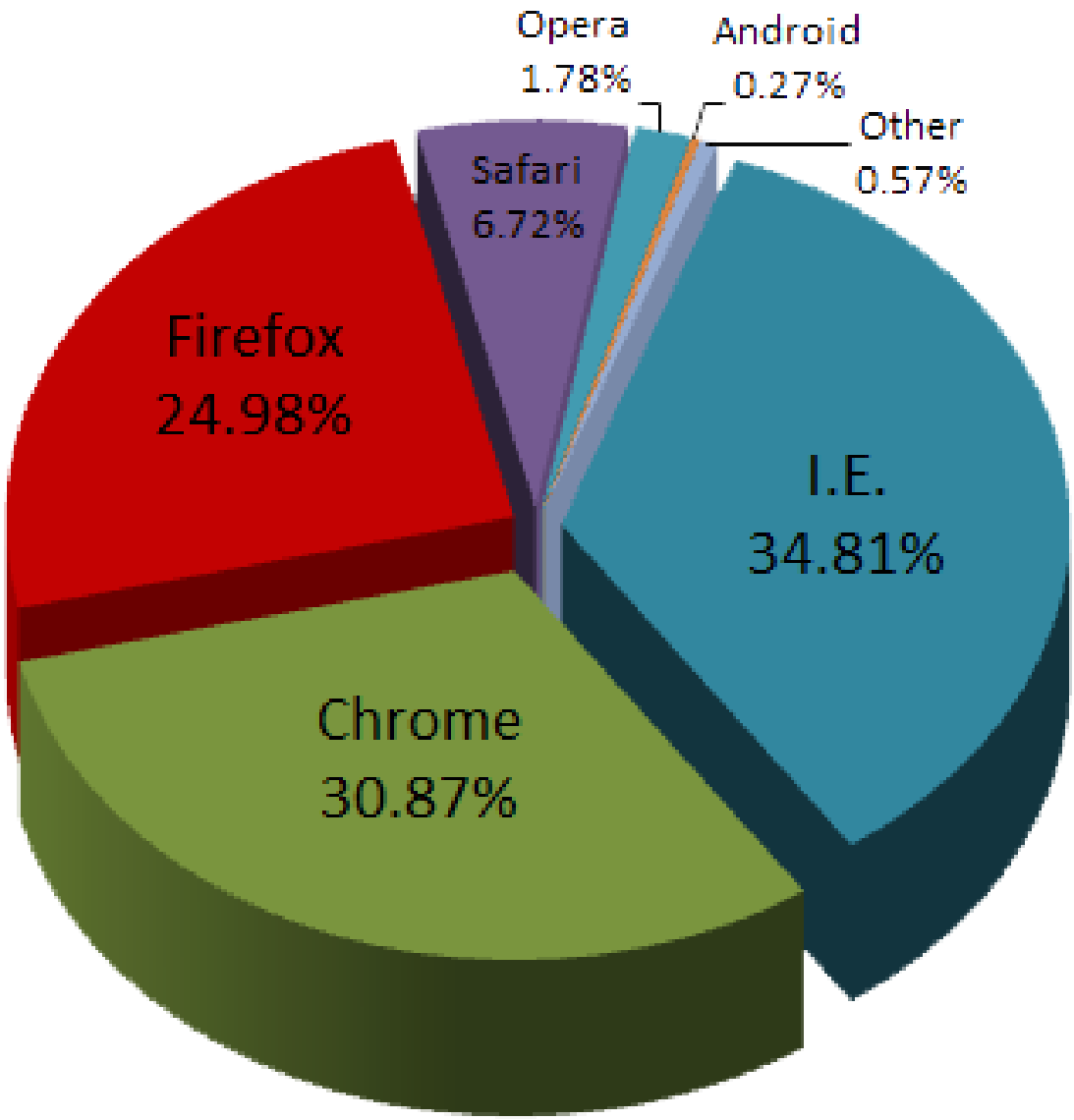
First Popular Web Browser

Mosaic was developed at the National Center for Supercomputing Applications (NCSA)at the University of Illinois Urbana-Champaign beginning in late 1992.

Mosaic was the web browser which led to the Internet boom of the 1990s.



Web Browser usage -2012



Internet / Protocols / WWW

- What is the Internet?
  - a network of networks – an inter-network, or Internet
- What are Internet protocols?
  - the rules for transferring information between programs
  - HTTP - hypertext transfer protocol
  - FTP - file transfer protocol
  - SMTP – simple mail transfer protocol
- What is the World Wide Web?
  - a set of HTML pages accessible using the HTTP protocol

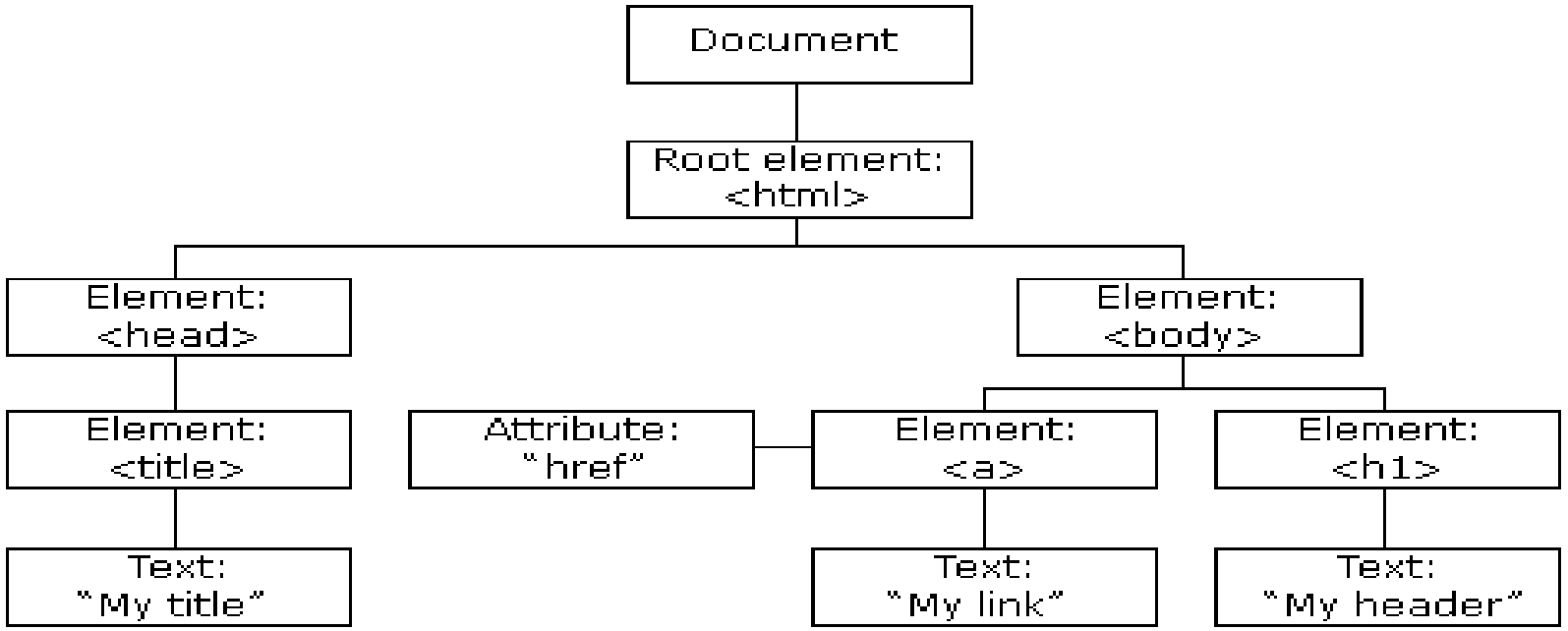
Java Script

What is JavaScript?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a **scripting** language
- A scripting language is a **lightweight programming** language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an **interpreted language** (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without **purchasing a license**

HTML DOM (Document Object Model)

- **The HTML DOM (Document Object Model)**
- 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**:



HTML DOM (Document Object Model)

With the HTML DOM, JavaScript can access all the elements of an HTML document.

With a programmable object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can react to all the events in the page

Finding HTML Elements

Finding HTML Elements

- Often, with JavaScript, you want to manipulate HTML elements.
- To do so, you have to find the elements first.
- Finding HTML elements by id

Finding HTML Elements by Id

```
var x=document.getElementById("intro");
```

Programming Constructs

- Variables
  - Named elements that can change value
- Data types
  - Integer, floating-point, Boolean, string
- Operators
  - Assignment, comparison, arithmetic, Boolean, string, special
- Control statements
  - Conditions, loops
- Keywords
  - Reserved words with special meaning

JavaScript Variables

Variables are "containers" for storing information.

JavaScript Variables

- As with algebra, JavaScript variables are used to hold values or expressions.
- A variable can have a short name, like x, or a more descriptive name, like carname.

Rules for JavaScript variable names:

- Variable names are case sensitive (y and Y are two different variables)
- Variable names must begin with a letter or the underscore character.

Note: Because JavaScript is case-sensitive, variable names are case-sensitive.

Java script

```
<html>
<body>

<h1>What Can JavaScript Do?</h1>

<p id="demo">JavaScript can change HTML content.</p>

<button type="button"
onclick="document.getElementById('demo').innerHTML = 'Hello
    JavaScript!'">
Click Me!</button>

</body>
</html>
```

Javascript Cont...

```
<html>
<body>
<p>This example calls a function which performs a calculation, and returns the
    result:</p>
<p id="demo"></p>
<script>
function myFunction(a, b) {
   return a * b;
}
document.getElementById("demo").innerHTML = myFunction(4, 3);
</script>

</body>
</html>
```

Javascript Cont...

For Loop Example

```
<html>
<body>
<p id="demo"></p>
<script>
cars = ["BMW", "Volvo", "Saab", "Ford"];
text = "";
var i;
for (i = 0; i < cars.length; i++) {
    text += cars[i] + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

Javascript Cont...

While / do while loop:

```
while (i < 10) {
        text += "The number is " + i;
        i++;
    }
```

```
do {
        text += "The number is " + i;
        i++;
    }
    while (i < 10);
```

Javascript Cont...

Array Demo:

```
<html>
<body>

<p id="demo"></p>

<script>
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[0];
</script>

</body>
</html>
```

Javascript Cont...

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

Javascript Cont...

Operator	Description
&&	and
||	or
!	not

Javascript Cont...

```
<html>
<body>
<p>Click the button to display what day it is today:</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction() {
    var day;
    switch (new Date().getDay()) {
        case 0:
            day = "Sunday";
            break;
```

Javascript Cont...

```
case 1:
        day = "Monday";
        break;
    case 2:
        day = "Tuesday";
        break;
    case 3:
        day = "Wednesday";
        break;
    case 4:
        day = "Thursday";
        break;
```

Javascript Cont...

```
case 5:
        day = "Friday";
        break;
    case  6:
        day = "Saturday";
        break;
  }
  document.getElementById("demo").innerHTML = "Today is " + day;
}
</script>

</body>
</html>
```

JavaScript Try...Catch Statement

JavaScript - Catching Errors

- When browsing Web pages on the internet, we all have seen a JavaScript alert box telling us there is a runtime error and asking "Do you wish to debug?". Error message like this may be useful for developers but not for users. When users see errors, they often leave the Web page.

```
try
  {
  //Run some code here
  }
catch(err)
  {
  //Handle errors here
  }
```

Ex: Try Catch

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

<p>Please input a number between 5 and 10:</p>

<input id="demo" type="text">
<button type="button" onclick="myFunction()">Test Input</button>
<p id="message"></p>
```

```
<script>
function myFunction() {
    var message, x;
    message = document.getElementById("message");
    message.innerHTML = "";
    x = document.getElementById("demo").value;
    try {
        if(x == "")  throw "empty";
        if(isNaN(x)) throw "not a number";
        x = Number(x);
        if(x < 5)    throw "too low";
        if(x > 10)   throw "too high";
    }
```

```
catch(err) {
        message.innerHTML = "Input is " + err;
    }
}
</script>
</body>
</html>
```

CSS: Cascading Style Sheets

CSS:-Adding Style to HTML

There are **three** ways of providing styling information for the Web browsers.

- External style sheet
- Internal style sheet
- Inline style

Benefits:

- Authors and Web site managers may share style sheets across a number of documents (and sites).
- Authors may change the style sheet without requiring modifications to the document.
- User agents may load style sheets selectively (based on media descriptions).

External (Linking) style sheet

- You can separate style sheets from HTML documents. Style sheet files are imported to HTML documents by **<link>.**

[example.html]

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

External (Linking) style sheet

- **"myStyle.css"**

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

h1 {
    color: navy;
    margin-left: 20px;
}
```

Internal style sheet

- You can put style sheet rules in the head of the document by **<style>**.

[example.html]

```
<head>
   <style>
   p { color: red; font-size:120%; }
    </style>
 </head>
   <body>
   <p>This is a paragraph</p>
</body>
```

Inline style sheet

- The start tags can contain style sheet rules directly in HTML documents by **the style attribute**.

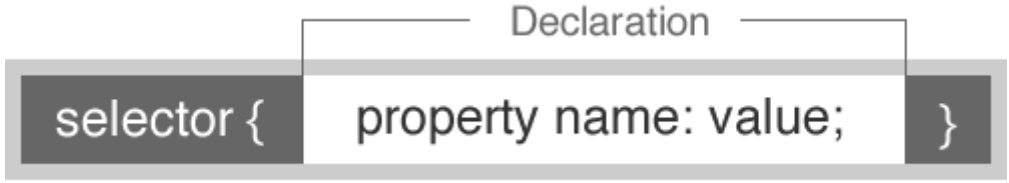
[example.html]

<p **style="color: red; font-size:120%; "**>
This is a paragraph</p>

CSS Syntax

Point of the syntax

- This syntax has two parts, the selector and the declaration.



Selector: Specifies the target of styling.
Declaration: Specifies the property and value.

- Declaration is contained between {" ... "}.
- Declaration end with a semicolon.

```
p{ color: red; }
```

Selectors

Selectors are specify the target of styling. Selectors may range from simple element names to rich contextual representations.

Kind of selector

- **Type selector**
- **Class selector**
- **ID selector**
- **Grouping**

Type selector

A type selector is the name of HTML Tag.

[index.html]

- This is a paragraph
- This is a paragraph
- This is a paragraph

[style.css]

- p{ color: red; font-size: 12px; }

Class selector

Class selector is used for one or more elements. It is described the value of class attribute of HTML document with **".(period)"**.

[index.html]

```
<p class="red">This is a paragraph</p>
<p class="blue">This is a paragraph</p>
<p class="red">This is a paragraph</p>
<p class="blue">This is a paragraph</p>
```

[style.css]

```
p{ font-size: 12px; }
.red{ color: red; }
.blue{ color: blue; }
```

ID selector

- ID selector is used for unique element. It is described the value of ID attribute of HTML document with **"#"**.

[index.html]

- This is a paragraph
- This is a paragraph
- <p class="red" **id="small"**>This is a paragraph

[style.css]

- p{ font-size: 12px; }
- .red{ color: red; }
- .blue{ color: blue; }
- **#small**{ font-size: 9px; }

Grouping

- A comma-separated list of selectors represents the union of all elements selected by each of the individual selectors in the list.

[index.html]

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
<h4>This is a heading</h4>
```

[style.css]

```
h1, h2, h3, h4{ color: red; font-size: 12px; }
```

DHTML

Dynamic HTML

Dynamic HTML, or DHTML, is an umbrella term for a collection of technologies used together to create interactive and animated web sites by using a combination of :

- a static markup language (such as **HTML**),
- a client-side scripting language (such as **JavaScript**),
- a presentation definition language (such as **CSS**), and the **Document Object Model**.

DHTML

- DHTML is not a technology in and of itself; rather, it is the product of three related and complementary technologies: HTML, Cascading Style Sheets (CSS), and script.

- To allow scripts and components to access features of HTML and CSS, the contents of the document were represented as objects in a programming model known as the Document Object Model (DOM).

Differences between HTML and DHTML

- **HTML**

1. It is referred as a static HTML and static in nature.
2.A plain page without any styles and Scripts called as HTML.
3.HTML sites will be slow upon client-side technologies.

DHTML

1.It is referred as a dynamic HTML and dynamic in nature.
2.A page with HTML, CSS, DOM and Scripts called as DHTML.
3.DHTML sites will be fast enough upon client-side technologies.

Short Questions

1. By which property you can select one Radio Button among multiple?
2. Give the names of four Button control present in Standard Toolbox?
3. Write code for a html table that will have one row and one column?
4. How to add JavaScript file reference in the web page?
5. What is external Style Sheet? How do you link it with a HTML Page?
6. How to write bulleted point in HTML using tag?
7. What is the alt property in <img> tag and give the full syntax of this tag ?

Long Questions

1. How a web page runs on a Client server architecture explain?
2. What are HTML Tags and write the steps of add CSS.
3. What is Java Script? Why it is helpful to add JS code in a web form.
4. What is HTML Page? Explain all Important Tags with a practical example.
5. Design the HTML website for a company with some linked pages and explain all pages with their use?
6. What is Cascading Style Sheets and describe all Selectors with their use and examples?
7. What is DHTML. Explain the role of it.

Research Problem

Combinatorial pattern matching in images and audio.

The signal processing community has traditionally addressed the problem of measuring the similarity between two images or audio segments (or parts thereof) despite of slight differences due to scale, orientation, lighting, stretching, etc. (in the first case) or timing, volume, tone, noise, etc. (in the second case). They have used an approach where the object is seen as a continuous signal to be processed.

Research Problem Contd…

- A recent alternative approach to pattern matching in audio and images relies on combinatory rather than on signal processing. The audio or image is seen as a one or two dimensional text, where one or two dimensional patterns are sought. Several results on searching images permitting rotations, scaling, pixel differences and stretching have been obtained, in many of which we have been involved. The same has happened in searching music files, using techniques derived from the large body of knowledge acquired in the field of pattern matching of biological sequences.
- Ref: http://www.cwr.cl/areas.html

Resources: Online

W3C Website

1. http://www.w3.org/Style/CSS/ (CSS)
2. http://www.w3.org/DOM/ (DOM)
3. http://www.w3.org/MarkUp/ (HTML/XHTML)

css-discuss listserv

1. http://www.css-discuss.org/

css-discuss Wiki

1. http://css-discuss.incutio.com/

JavaScript Message Board

1. http://www.aspmessageboard.com/forum/jscript.asp

XMLHttpRequest() Information

1. http://developer.apple.com/internet/webcontent/xmlhttpreq.html
2. http://www.xml.com/lpt/a/2005/02/09/xml-http-request.html

References

TEXT BOOKS:

1. K. K. Sharma, “Web Technology”, A.B. Publication Delhi, First Edition, 2008.
2. **Stephen Walther, “ASP.NET 4”, Pearson Education.**
3. Ethan Cerami, “Web Services”, O'Reilly Media, 2002.
4. Achyut S Godbole and Atul Kahate, “Web Technologies”, Tata McGraw Hill.
5. Heith Morneau, “Active Server Pages “, Vikas Publishing House.

REFERENCES:

1. Raj Kamal , “Internet and Web Technologies”, TMH.
2. **Deitel, “Internet & World Wide Web , How to Program”, PHI.**
3. V. K. Jain, “Advanced programming in Web Design”, Cyber tech.
4. Rick Dranell, “HTML4 unleashed”, Techmedia Publication, 2000.