

Markup Languages

HTML basics

Outline

- Definitions – WWW, HTTP, HTML
- The concept of Markup Language
- What is SGML?
- Introduction to HTML
 - Creating and publishing a Web page
 - Validating a document
 - Main HTML elements
 - Block-level, Text-level HTML elements
 - Creating hypertext links, adding images to documents
 - Building tables, Using Frames and Forms
- Introduction to XML
 - DTD, MathML, XHTML

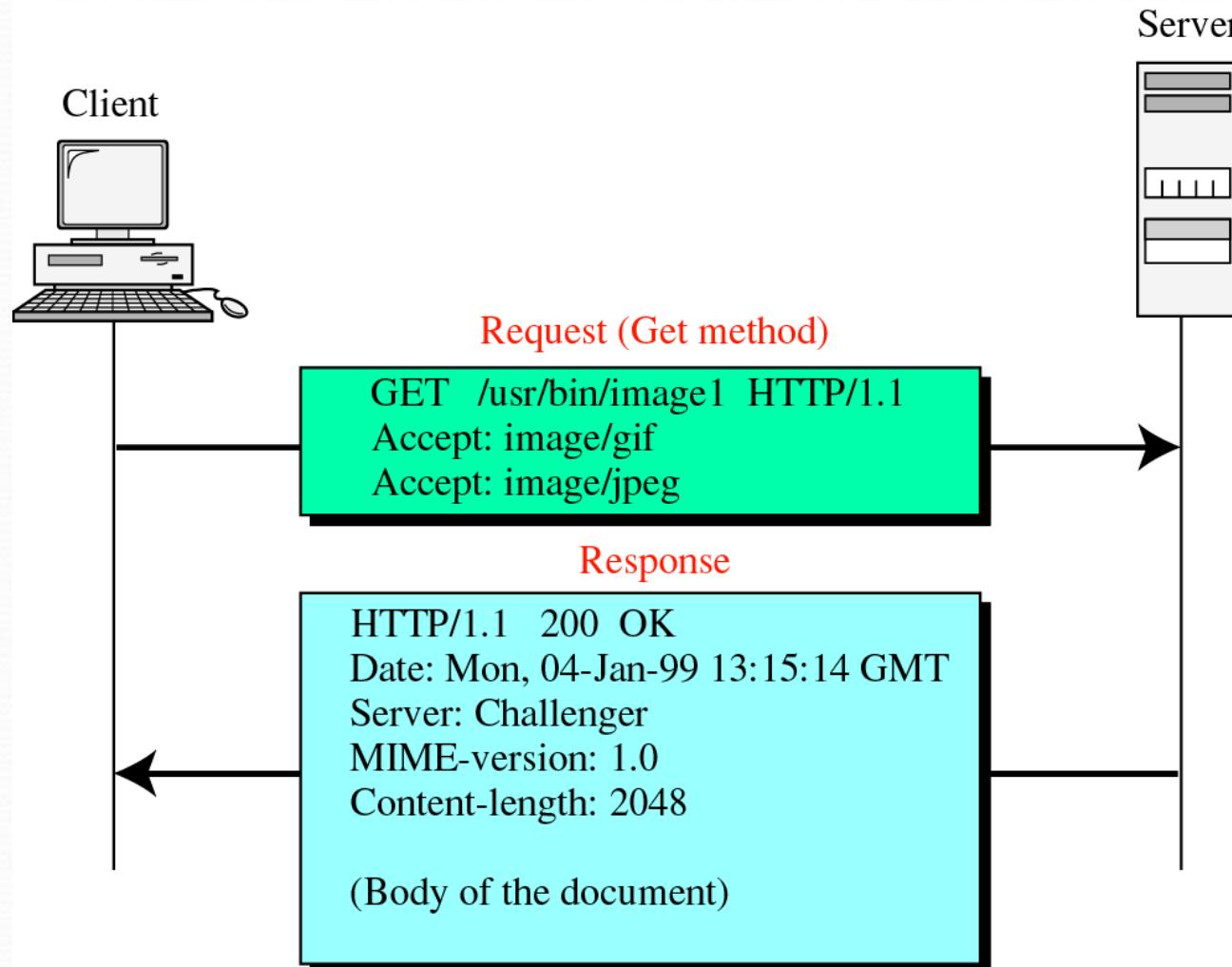
The World Wide Web

- Definitions
 - The World Wide Web (WWW)
 - The set of computers on the Internet that support HTTP
 - Not a separate network
 - HTTP
 - The HyperText Transfer Protocol
 - The language used by a WWW client (e.g. Netscape, IE) to request documents from a WWW server (i.e. the program running at Web sites like amazon.com or yahoo.com)
 - HTML
 - The HyperText Markup Language
 - The language used to design web pages

HTTP

- HTTP (Hypertext Transfer Protocol)
 - protocol used to access data on the WWW.
 - uses one TCP connection on well-known port 80.
 - two types of http messages: *Request*, *Response*
 - transfer data in the form of plain text, hypertext, audio, video, and so on.

HTTP



View Request / response HTTP headers - Chrome

- In Chrome, visit a URL, right click , select Inspect to open the developer tools.
- Select Network tab.
- Reload the page, select any HTTP request on the left panel, and the HTTP headers will be displayed on the right panel.

Elements Console Sources Network Timeline Profiles Application > ⋮ X

Filter Regex Hide data URLs

All | XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Headers	Preview	Response	Timing
me				
me				
me/				
css?family=Open+Sans%3A300it...				
dashicons.min.css?ver=4.0				
admin-bar.min.css?ver=4.0				
style.css?ver=4.0				
cropped-wordpress-blog-header...				
8fa38dd829fc3389ce6fb883825d...				
admin-bar.min.js?ver=4.0				
0122b7145e73054a414318b33c9...				
0122b7145e73054a414318b33c9...				
wordpress.png				
cJZKeOuBrn4kERxqtaUH3VtXRa8...				
data:application/x-...				
ad516503a11cd5ca435acc9bb652...				
ad516503a11cd5ca435acc9bb652...				

General

Request URL: <http://alphapeeler.sf.net/me>
Request Method: GET
Status Code: 301 Moved Permanently
Remote Address: 216.34.181.96:80

Response Headers [view source](#)

Cache-Control: max-age=172800
Connection: keep-alive
Content-Length: 245
Content-Type: text/html; charset=iso-8859-1
Date: Sun, 22 Jan 2017 20:36:32 GMT
Expires: Tue, 24 Jan 2017 20:36:32 GMT
Location: <http://alphapeeler.sourceforge.net/me>
Server: nginx

Request Headers [view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8,es;q=0.6
Connection: keep-alive
Host: alphapeeler.sf.net
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36

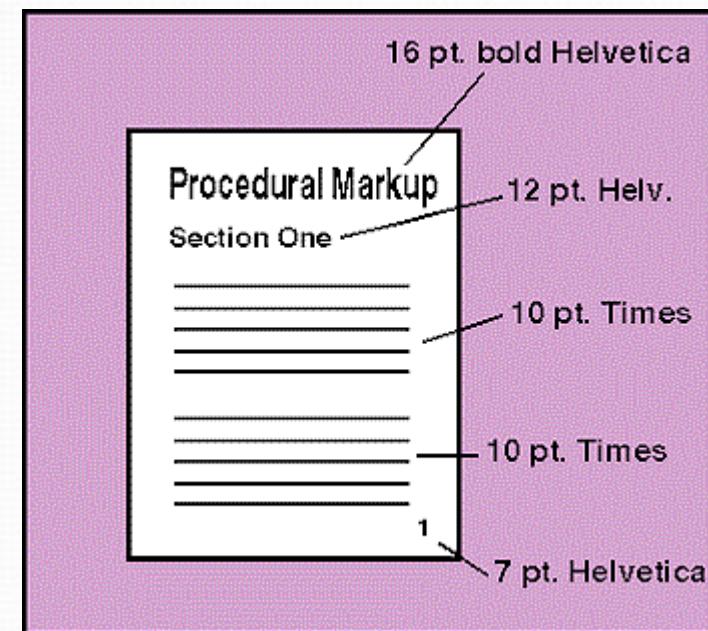
17 requests | 58.8 KB transferred ...

The Concept of Markup Language

- Markup language
 - originates from the book publishing industry.
 - SGML was originally designed to enable the sharing of machine-readable large-project documents in government, law, and industry.
 - allows us to embed formatting instructions in the document.
- Two types of markup:
 - Procedural markup
 - Descriptive markup or Generic markup

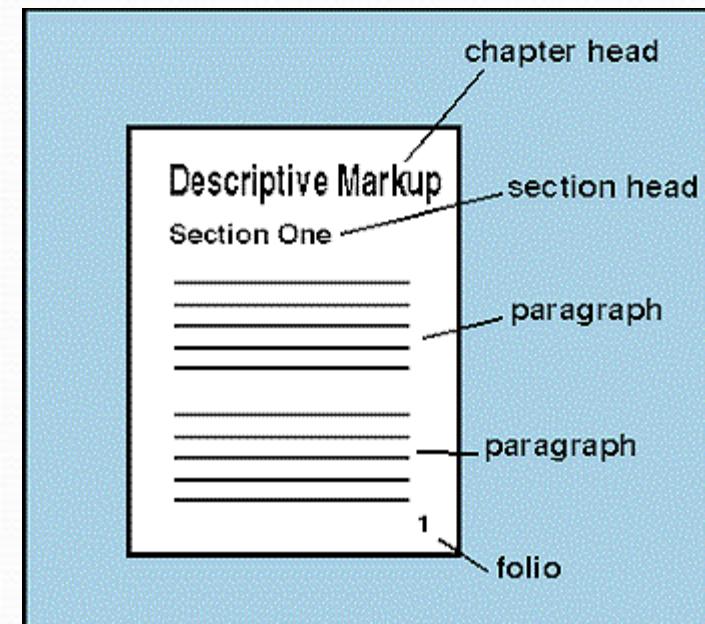
Procedural Markup

- formatting codes are mixed in with the text of the document.
- typically unique to a specific software package such as Word Perfect, Word Star, Microsoft Word etc.
- Disadvantages:
 - a single way of presenting the information, such as a printed page
 - provide no capability to define appearance for other media, such as CD-ROM and Internet.



Descriptive Markup

- Basic concept: **document content should remain separate from style.**
 - based on the **structure of a document** and identifies elements within that structure -- such as a chapter, a section, or a table of contents -- using notations that describe what the element is, **not how it appears.**
- allows multiple presentations of the same information.
- e.g. publications on paper, on CD-ROM and on the Web can come from one set of source files.



SGML

- SGML: Standard Generalized Markup Language
 - an international standard (ISO 8879) published in 1986.
 - prescribes a standard format for embedding descriptive markup within a document.
 - specifies a standard method for describing the structure of a document.
 - allows you to set up hierarchical models for each type of document you produce.
 - supports different document structures for different information types, e.g. technical manuals, parts catalogs, design specifications, reports and memos.

SGML

- SGML documents are
 - ASCII format: human and machine-readable
 - independent of any specific hardware or software
 - portable
- Can be broken into three components:
 - structure
 - content
 - style

SGML

- Structure:
 - defined in a file called the Document Type Definition (DTD)
 - describes the structure of a document, much like a database schema describes the types of information it handles and the relationships between fields
 - e.g. the address information should contain street, city, country fields
 - specifies rules for the relationships between elements
 - e.g. the street field must go first before the city field.
 - e.g. the country field is compulsory and can not be blanks.

SGML

- Content:

- content is the information itself
- identify the content's position within the DTD structure by using "tagging."
 - These tags mark the beginning and end of each part of the structure.
 - e.g. <street> 276 Castle Peak Road </street>
 - "<street>" indicates the start of a street name, and "</street>" indicates the end
- Nesting of tags reveal the structure of a document.

<address>

<street> street name </street>

<city> city name </city>

</address>

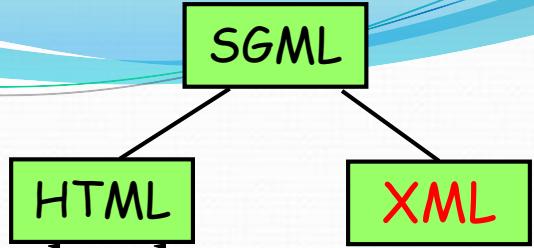
SGML

- Style:
 - Define how different fields should be displayed.
 - e.g. The style defines that “city field in the address should be all in upper case and in red colored font”.
 - Content: <city> hong kong </city>
 - Display: **HONG KONG**
 - SGML itself has nothing to do with setting standards for style, so most systems still rely on proprietary methods.

SGML

- Benefits of SGML:
 - Increased productivity
 - Reusability
 - Information longevity
 - Improved data integrity
 - Better data control
 - Shareability
 - Portability of information
 - Flexibility beyond traditional publishing

SGML



- Two implementations of SGML standard:
 - HTML: Hypertext Markup Language
 - XML: Extensible Markup Language

	HTML	XML
Structure	Not defined	Defined in DTD files
Content	<ul style="list-style-type: none">▪ Limited number of tags▪ Tags are defined in HTML specifications▪ Tags are not case sensitive	<ul style="list-style-type: none">▪ Unlimited number of tags▪ Tags are defined in DTD▪ Tags are case sensitive
Style	Not defined HTML4, CSS-HTML5	Defined in XSL (style) files

SGML

- HTML tags are limited, so we need to write content using available HTML tags
 - e.g. writing address as

<P> 276 Castle Peak Road </P>

<P> Hong Kong </P>
- With XML, we can define new tags for writing address fields.
- e.g. it will be more descriptive to write

<street> 276 Castle Peak Road </street>

<city> Hong Kong </city>

HTML History & its Future

- HTML 2.0
- HTML 3.2
- HTML 4.0
 - All formatting is separated into a style sheet.
- HTML 4.01
 - Makes the future upgrade from HTML to XHTML in a simple process.
- XHTML – sometimes referred to as HTML 5
 - The future of HTML standard
 - Almost identical to HTML 4.01
 - 1.0, 1.1 and 2.0

W3C XHTML Validation Service

The screenshot shows a web browser window with the URL <https://validator.w3.org/#validate-by-input>. The page has three tabs at the top: "Validate by URI" (selected), "Validate by File Upload", and "Validate by Direct Input". The main content area is titled "Validate by URI" and contains the following fields:

- "Address:" input field (empty)
- "More Options" link
- "Character Encoding" dropdown set to "(detect automatically)" with a "Only if missing" checkbox
- "Document Type" dropdown set to "(detect automatically)" with a "Only if missing" checkbox
- Radio buttons for "List Messages Sequentially" (selected) and "Group Error Messages by Type"
- Checkboxes for "Show Source", "Clean up Markup with HTML-Tidy", "Show Outline", "Validate error pages", and "Verbose Output"
- A large "Check" button at the bottom

Validating an XHTML document. (Courtesy of World Wide Web Consortium (W3C).)

<https://validator.w3.org/>

W3C XHTML Validation Service

→ C Secure | <https://validator.w3.org/check>



Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Jump To:

Notes and Potential Issues

Validation Output

Errors found while checking this document as HTML 4.01 Transitional!

Result:	12 Errors, 5 warning(s)
Source :	<pre>Stallings, William, Stallings, Dr. Bill Stallings, Bill, Ibn Khaldun Syste SSUET, S.S.U.E.T., Sir, Syed, Sir Syed, Sir Syed University, Sir Syed University of Engine University of Engineering & Technology, Sir Syed University of Engg & Tech, Sir Syed Unive Syed University of Engg. and Tech., Sir Syed University of Engg. & Tech., Sir Syed Univers University of Engg, Sir Syed University of Engg. , Pakistan, Karachi, Research, Developmer Research & Development, R&D"> </head> <BODY> <center> <OBJECT CLASSID="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000" WIDTH="800" HEIGHT="620" CODEBASE="http://active.macromedia.com/flash3/cabs/swflash.cab#version=5,0,0,0"> <PARAM NAME=movie VALUE="WebPageIntro.swf"> <PARAM NAME=_wmode VALUE="transparent"></pre>
Encoding :	utf-8
Doctype :	HTML 4.01 Transitional
Root Element:	html

XHTML validation results. (Courtesy of World Wide Web Consortium (W3C).)

W3C XHTML Validation Service

The screenshot shows a web browser window with the URL <https://validator.w3.org/check>. The title bar says "Validation Output: 12 Errors". The main content area displays two error messages:

Line 3, Column 1: no document type declaration; implying "<!DOCTYPE HTML SYSTEM>"

The checked page did not contain a document type ("DOCTYPE") declaration. The Validator has tried to validate with a fallback DTD, but this is quite likely to be incorrect and will generate a large number of incorrect error messages. It is highly recommended that you insert the proper DOCTYPE declaration in your document -- instructions for doing this are given above -- and it is necessary to have this declaration before the page can be declared to be valid.

Line 5, Column 1898: cannot generate system identifier for general entity "D"

... Research, Development, Research and Development, Research & Development, R&D ">

An entity reference was found in the document, but there is no reference by that name defined. Often this is caused by misspelling the reference name, unencoded ampersands, or by leaving off the trailing semicolon (;). The most common cause of this error is unencoded ampersands in URLs as described by the [WDG](#) in "[Ampersands in URLs](#)".

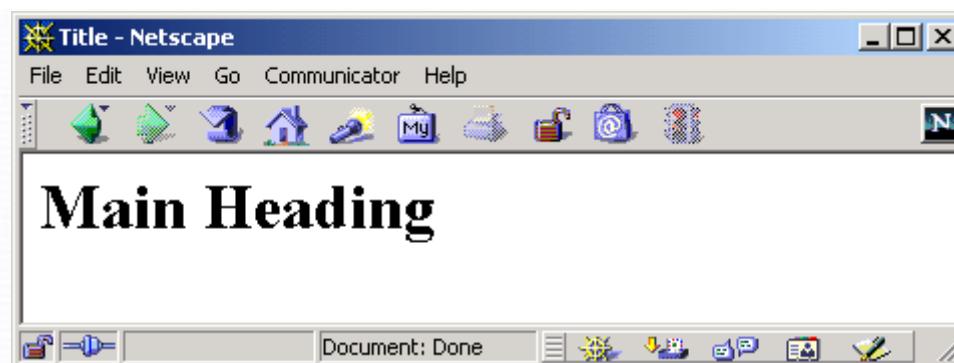
Entity references start with an ampersand (&) and end with a semicolon (;). If you want to use a literal ampersand in your document you must encode it as "&" (even inside URLs!). Be careful to end entity references with a semicolon or your entity reference may get interpreted in connection with the following text. Also keep in mind that named entity references are case-sensitive; &Aelig;

XHTML validation results. (Courtesy of World Wide Web Consortium (W3C).)

HTML Document Template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Title</TITLE> ← Goes on browser's title bar
  May not appear in printouts
</HEAD>

<BODY>
<H1>Main Heading</H1> ← Main heading. Often used as title
  Appears in printouts
<!-- Rest of page goes here --> ← HTML comment
  Replace with body of
  WWW page
</BODY>
</HTML>
```



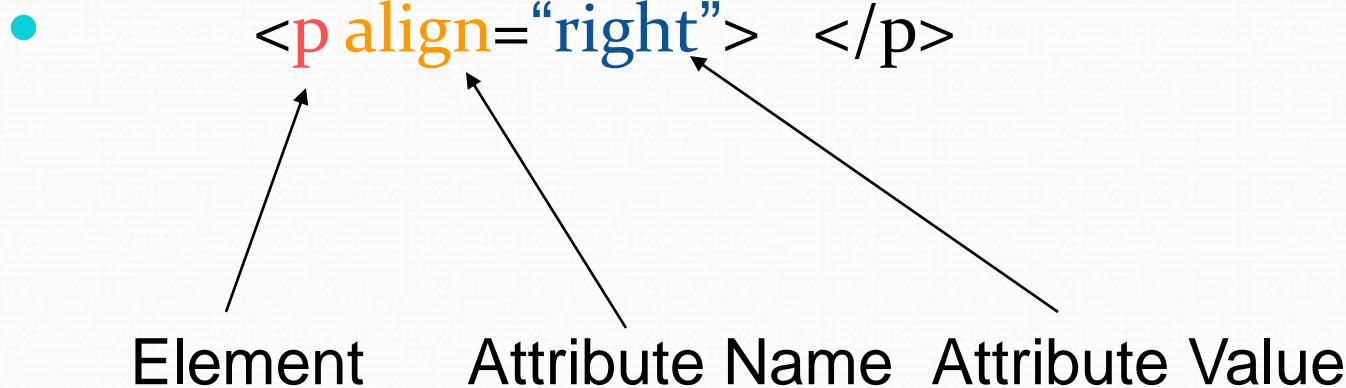
HTML <META> Tag

- Can record document information
- Forward and refresh pages

```
<HEAD>
  <TITLE>Why You Should Buy Windows 2000</TITLE>
  <BASE HREF="http://www.microsoft.com/windows2000/">
  <META NAME="author" CONTENT="Bill Gates">
  <META NAME="keywords"
        CONTENT="Windows,Advocacy,OS,Operating Systems">
  <META NAME="description"
        CONTENT="A summary of the advantages of Windows 2000.">
</HEAD>
```

```
<HEAD>
  <TITLE>Why You Should Buy Windows 2000</TITLE>
  <META HTTP-EQUIV="Refresh" CONTENT="30">
</HEAD>
```

HTML Element / Tag



- You have to understand the important terms related to HTML.
- Not case-sensitive.

Main HTML Elements

1. DOCTYPE

1. HTML 4 : `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">`
2. HTML 5: `<!DOCTYPE HTML>`

2. HTML

3. HEAD

- TITLE element required
- Optional elements:
 - BASE
 - META
 - BG SOUND
 - SCRIPT, NOSCRIPT
 - STYLE
 - LINK

Main HTML Elements

4. BODY Element

- <BODY BGCOLOR="YELLOW">
- HTML Attributes and Attribute Values
 - BACKGROUND
 - BGCOLOR
 - TEXT
 - LINK, VLINK, ALINK
 - OnLoad, OnUnload, OnFocus, OnBlur

5. Elements inside BODY element

- <BODY>
Remaining HTML elements
</BODY>

```
<BODY onBlur="alert('Body onBlur alert!');"  
onError="alert('Body onError alert!');"  
onFocus="alert('Body onFocus alert!');"  
onLoad="alert('Body onLoad alert!');"  
onUnload="alert('Body onUnload alert!');">
```

META Element

- Records document information, forwards and refreshes pages
 - NAME="author"
 - NAME="keywords"
 - NAME="description"
 - HTTP-EQUIV="refresh"

Example:

```
<meta http-equiv="refresh" content="30;url=index.html">
<meta http-equiv="refresh" content="30;url=new.html">
```

META Element, Example

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
  <TITLE>News Headlines</TITLE>
  <META HTTP-EQUIV="REFRESH" CONTENT="30">
</HEAD>

<BODY>
<H1 ALIGN="CENTER">News Headlines</H1>

<H2>National News</H2>
Blah, blah, blah.
<H2>International News</H2>
Yadda, yadda, yadda.

</BODY>
</HTML>
```

Block-Level Elements

- **Headings**
 - H1 ... H6
 - ALIGN
- **Basic Text Sections**
 - P
 - ALIGN
 - PRE
 - WIDTH
 - ADDRESS
 - BLOCKQUOTE
- **Lists**
 - OL
 - LI
 - UL
 - LI
 - DL
 - DT
 - DD
- **Frames and Forms**
- **Misc.**
 - HR
 - DIV
 - CENTER
 - MULTICOL (Netscape only)

Headings

- Heading Types
 - <H1 ...> ... </H1>
 - <H2 ...> ... </H2>
 - <H3 ...> ... </H3>
 - <H4 ...> ... </H4>
 - <H5 ...> ... </H5>
 - <H6 ...> ... </H6>
- Attributes: ALIGN
 - Values: LEFT (default), RIGHT, CENTER
- Nesting tags
 - Headings and other block-level elements can contain text-level elements, but *not* vice versa

Headings, Example

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
    <TITLE>Document Headings</TITLE>
</HEAD>
<BODY>
Samples of the six heading types:
<H1>Level-1 (H1)</H1>
<H2 ALIGN="CENTER">Level-2 (H2)</H2>
<H3><U>Level-3 (H3)</U></H3>
<H4 ALIGN="RIGHT">Level-4 (H4)</H4>
<H5>Level-5 (H5)</H5>
<H6>Level-6 (H6)</H6>
</BODY>
</HTML>
```

Headings, Result

A screenshot of Microsoft Internet Explorer version 4.0 or later. The title bar reads "Document Headings - Microsoft Internet Explorer". The menu bar includes File, Edit, View, Favorites, Tools, and Help. The toolbar contains standard icons for Back, Forward, Stop, Home, Search, and others. The main content area displays the following text:

Samples of the six heading types:

Level-1 (H1)

Level-2 (H2)

Level-3 (H3)

Level-4 (H4)

Level-5 (H5)

Level-6 (H6)

The browser window has scroll bars on the right and bottom. The status bar at the bottom shows "Done" and "My Computer".

P – The Basic Paragraph

- Attributes: ALIGN
 - LEFT (default), RIGHT, CENTER. Same as headings.
 - Whitespace ignored (use
 for line break)
 - Consecutive <P>'s do not yield multiple blank lines
 - End Tag is Optional:

```
<BODY>
<P>
Paragraph 1
</P>
<P>
Paragraph 2
</P>
<P>
Paragraph 3
</P>
</BODY>
```

Fully-Specified

```
<BODY>
Paragraph 1
<P>
Paragraph 2
<P>
Paragraph 3
</BODY>
```

Equivalent with Implied Tags

Preformatted Paragraphs

- The <pre> tag in HTML is used to define the **block of preformatted text** which preserves the text spaces, line breaks, tabs, and other formatting characters which are ignored by web browsers.

- The PRE Element

- <PRE> ... </PRE>

- Problem: Special Characters

```
<PRE>
if (a &lt; b) {
    doThis();
} else {
    doThat();
}
</PRE>
```

```
<pre>
AlphaPeeler
A Computer Science Portal
For NU Students
</pre>
```

Desired Character	HTML Required
<	<
>	>
&	&
"	"
Non-breaking space	

HTML Character Entities

- The “<“ character has special meaning in HTML documents
 - We cannot use it directly in the text.
 - We have to use a character entity.
 - We must use “<” or “<”
 - Note, they are case sensitive.
 - The most common one is the non-breaking space – “ ”

The most common entities

Display	Description	Name
	Non-breaking space	
<	Less than	<
>	Greater than	>
&	Ampersand	&
“	Quotation mark	"
‘	Apostrophe	'

HTML Lists

- Unordred Lists
- Ordered Lists
- Definition Lists

OL: Ordered (Numbered) Lists

- OL Element
 -
 ...
 ...
 ...

 - Attributes: TYPE, START, COMPACT
- List entries: LI
 - <LI ...> ... (End Tag Optional)
 - Attributes: (When inside OL) VALUE, TYPE

A sample list:

```
<OL>
    <LI>List Item One
    <LI>List Item Two
    <LI>List Item
Three
</OL>
```

A sample list:

```
1. List Item One
2. List Item Two
3. List Item Three
```

```
<OL TYPE="I">
<LI>Headings
<LI>Basic Text Sections
<LI>Lists
    <OL TYPE="A">
        <LI>Ordered
            <OL TYPE="1">
                <LI>The OL tag
                    <OL TYPE="a">
                        <LI>TYPE
                        <LI>START
                        <LI>COMPACT
                    </OL>
                <LI>The LI tag
            </OL>
        <LI>Unordered
            <OL TYPE="1">
                <LI>The UL tag
                <LI>The LI tag
            </OL>
        <LI>Definition
            <OL TYPE="1">
                <LI>The DL tag
                <LI>The DT tag
                <LI>The DD tag
            </OL>
        </OL>
    <LI>Miscellaneous
</OL>
```

Nested Ordered Lists

- I. Headings
- II. Basic Text Sections
- III. Lists
 - A. Ordered
 - 1. The OL tag
 - a. TYPE
 - b. START
 - c. COMPACT
 - 2. The LI tag
 - B. Unordered
 - 1. The UL tag
 - 2. The LI tag
 - C. Definition
 - 1. The DL tag
 - 2. The DT tag
 - 3. The DD tag
- IV. Miscellaneous

UL: Unordered (Bulleted) Lists

- UL Element
 -
 - ...
 - ...
 - ...
 -
- Attributes: TYPE, COMPACT
 - TYPE is DISC, CIRCLE, or SQUARE
- List entries: LI (TYPE)
 - TYPE is DISC, CIRCLE, or SQUARE

A sample list:

```
<UL>
  <LI>List Item One
  <LI>List Item Two
  <LI>List Item Three
</UL>
```

A sample list:

```
● List Item One
● List Item Two
● List Item Three
```

```
<UL TYPE="DISC">
<LI>The UL tag
    <UL TYPE="CIRCLE">
        <LI>TYPE
            <UL TYPE="SQUARE">
                <LI>DISC
                <LI>CIRCLE
                <LI>SQUARE
            </UL>
        <LI>COMPACT
    </UL>
<LI>The LI tag
    <UL TYPE="CIRCLE">
        <LI>TYPE
            <UL TYPE="SQUARE">
                <LI>DISC
                <LI>CIRCLE
                <LI>SQUARE
            </UL>
        <LI>VALUE
    </UL>
</UL>
```

UL: Custom Bullets

Unordered Lists

- The UL tag
 - TYPE
 - DISC
 - CIRCLE
 - SQUARE
 - COMPACT
- The LI tag
 - TYPE
 - DISC
 - CIRCLE
 - SQUARE
 - VALUE

DL: Description Lists

- A description list, with terms and descriptions:
- DL – description list
- DT - term
- DD - descriptions
- <dl>

```
<dt>Coffee</dt>
<dd>Black hot drink</dd>
<dt>Milk</dt>
<dd>White cold drink</dd>
</dl>
```

Coffee

Black hot drink

Milk

White cold drink

Text-Level Elements

- Physical Character Styles

- B, I, TT, U, SUB, SUP, SMALL, BIG, STRIKE, S, BLINK (Obsolete)
- TT => teletype text (!HTML5)
- S : <S>Mark up text that is no longer correct </S>
- FONT
 - SIZE
 - COLOR
 - FACE
- BASEFONT (!HTML5)
- SIZE

```
<head>
<basefont color="red" size="5">
</head>
```

- Logical Character Styles

- EM, STRONG, CODE, samp, KBD, DFN, VAR, CITE



Run »

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

<em>Emphasized text</em><br>
<strong>Strong text</strong><br>
<code>A piece of computer code</code><br>
<samp>Sample output from a computer program</samp><br>
<kbd>Keyboard input</kbd><br>
<var>Variable</var><br>
<cite>citation</cite><br>
<dfn>HTML</dfn><br>
</body>
</html>
```

Emphasized text

Strong text

A piece of computer code

Sample output from a computer program

Keyboard input

Variable

citation

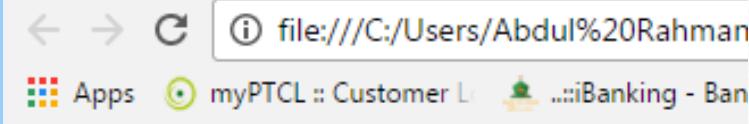
HTML

Text-Level Elements

- Hypertext Links
 - A
 - HREF, NAME, TARGET, ...
- Images
 - IMG
 - SRC (required), ALT, ALIGN, WIDTH, HEIGHT, HSPACE, VSPACE, BORDER, USEMAP, ISMAP
- Misc. Text-Level Elements
 - BR (Explicit line break)
 - AREA (Client-side image maps)
 - APPLET (Java)
 - ...

Physical Character Styles, Example

```
...
<H1>Physical Character Styles</H1>
<B>Bold</B><BR>
<I>Italic</I><BR>
<TT>Teletype (Monospaced)</TT><BR>
<U>Underlined</U><BR>
Subscripts: f<SUB>0</SUB> + f<SUB>1</SUB><BR>
Superscripts: x<SUP>2</SUP> + y<SUP>2</SUP><BR>
<SMALL>Smaller</SMALL><BR>
<BIG>Bigger</BIG><BR>
<STRIKE>Strike Through</STRIKE><BR>
<B><I>Bold Italic</I></B><BR>
<BIG><TT>Big Monospaced</TT></BIG><BR>
<SMALL><I>Small Italic</I></SMALL><BR>
<FONT COLOR="GRAY">Gray</FONT><BR>
<DEL>Delete</DEL><BR>
<INS>Insert</INS><BR>
...
...
```



Physical Character Styles

Bold
Italic
Teletype (Monospaced)
Underlined
Subscripts: $f_0 + f_1$
Superscripts: $x^2 + y^2$
Smaller
Bigger
Strike Through
Bold Italic
Big Monospaced
Small Italic
Gray
Delete
Insert

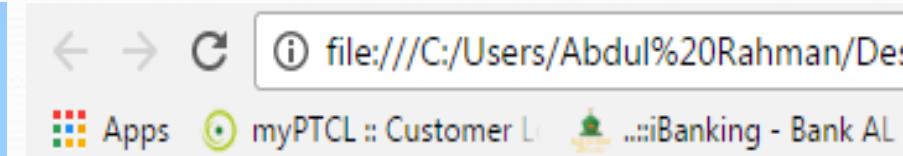
Text Formatting

	Bold text
<big>	Big text
	Emphasized text
<i>	Italic text
<small>	Small text
	Strong text
<sub>	Subscripted text
<sup>	Superscripted text

Logical Character Styles,

Example

```
...  
<H1>Logical Character  
Styles</H1>  
<EM>Emphasized</EM><BR>  
<STRONG>Strongly  
Emphasized</STRONG><BR>  
<CODE>Code</CODE><BR>  
<SAMP>Sample Output</SAMP><BR>  
<KBD>Keyboard Text</KBD><BR>  
<DFN>Definition</DFN><BR>  
<VAR>Variable</VAR><BR>  
<CITE>Citation</CITE><BR>  
<EM><CODE>Emphasized  
Code</CODE></EM><BR>  
<FONT COLOR="GRAY"><CITE>Gray  
Citation</CITE></FONT><BR>  
<ACRONYM TITLE="Java Development  
Kit">JDK Acronym</ACRONYM>  
...
```



Logical Character Styles

Emphasized
Strongly Emphasized
Code
Sample Output
Keyboard Text
Definition
Variable
Citation
Emphasized Code
Gray Citation
JDK Acronym

HTML Links

- <a> to create a link to another document.
- The target attribute
 - xxx
 - Open the document in a new browser window.
- The name attribute
 -
 - Useful text

Hypertext Links

- Links can contain images and other text-level elements (i.e., <A HREF...> ...)
- Link to **Absolute URL**
 - Use a complete URL beginning with http://
Java is discussed in
<A HREF="<http://host/path/chapter2.html>">
Chapter 2.
- Link to **Relative URL**
 - Use a filename or relative path to filename
 - Interpreted wrt location of current file
Java is discussed in
<A HREF="<chapter2.html>">Chapter 2.

Hypertext Links

- Link to Section
 - Use a section name (see below) preceded by #
Images are discussed in
`Section 2.`
- Link to Section in URL
 - Use absolute or relative URL, then #, then section name
Images are discussed in
`
Sec. 2 of Chap. 1.`
- Naming a Section
 - Use `` and do not include the pound sign
`<H2>Images</H2>`

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html >
3
4
5  <!-- Fig. 4.5: links.html -->
6  <!-- Introduction to hyperlinks -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9    <head>
10      <title>Internet and WWW How to Program - Links</title>
11    </head>
12
13  <body>
14
15    <h1>Here are my favorite sites</h1>
16
17    <p><strong>Click on a name to go to that page </strong></p>
18
19    <p><a href = "http://www.deitel.com">Deitel</a></p>
20
21    <p><a href = "http://www.prenhall.com">Prentice Hall</a></p>
22
23    <p><a href = "http://www.yahoo.com">Yahoo!</a></p>
24
25    <p><a href = "http://www.usatoday.com">USA Today</a></p>
26
27  </body>
28 </html>
```

Text between **strong** tags will appear bold.

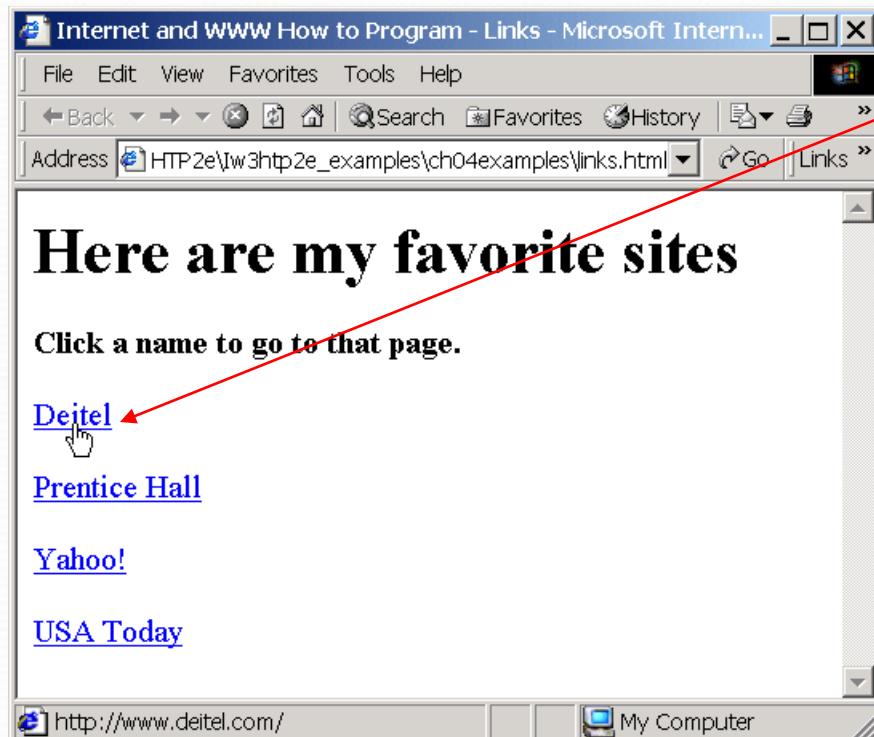
Linking is accomplished in XHTML with the anchor (**a**) element.

The text between the **a** tags is the anchor for the link.

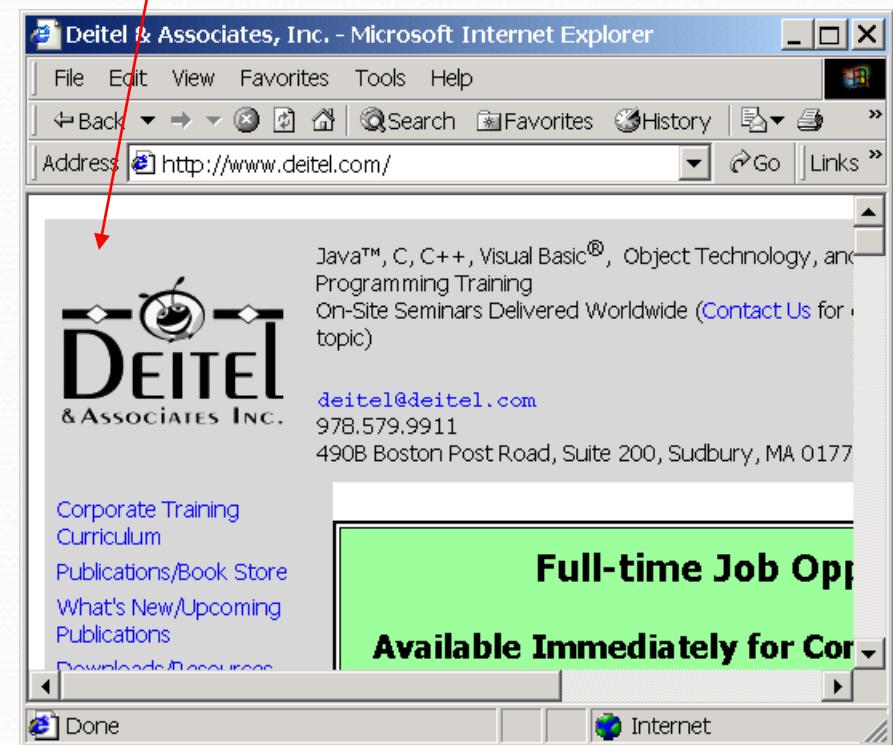
The anchor links to the page that's value is given by the **href** attribute.

Elements placed between paragraph tags will be set apart from other elements on the page with a vertical line before and after it.

Hypertext Links



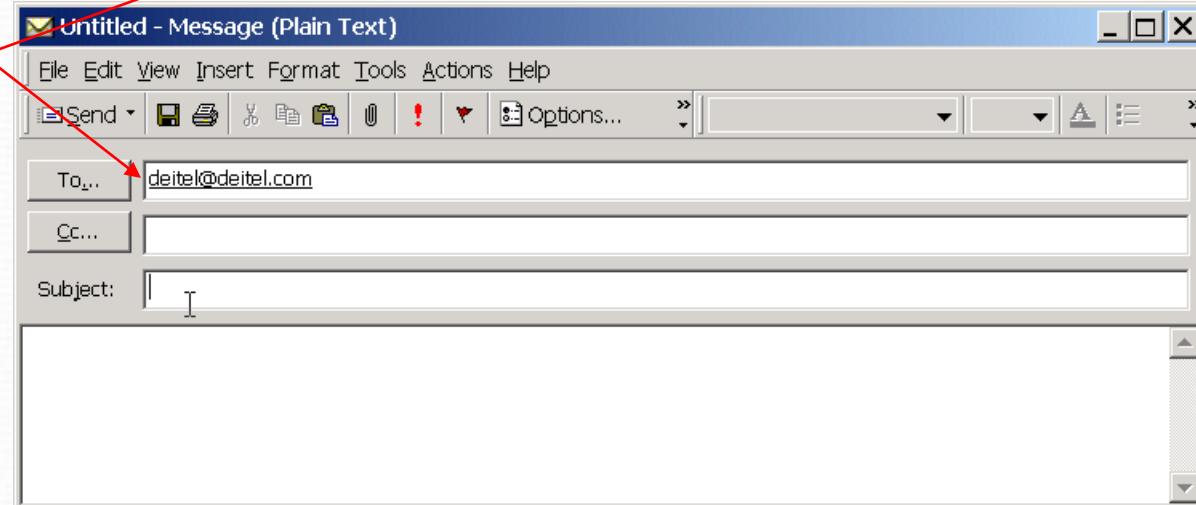
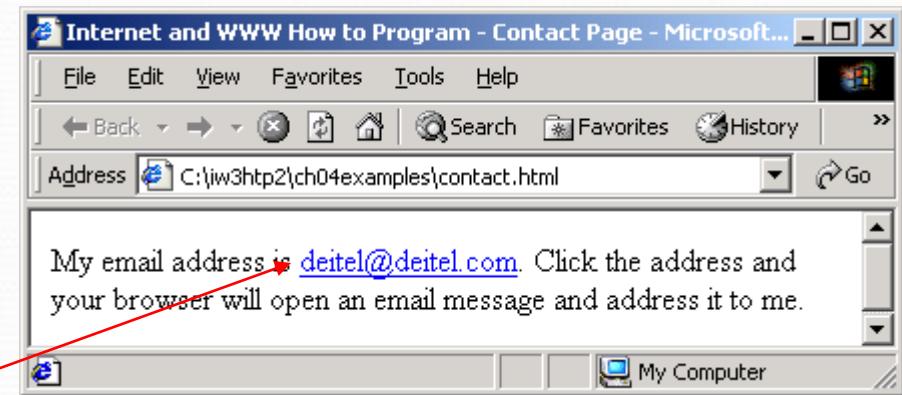
Clicking on the “**Deitel**” link will open up the Deitel homepage in a new browser window.



```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html >
3
4
5 <!-- Fig. 4.6: contact.html -->
6 <!-- Adding email hyperlinks -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Contact Page
11   </title>
12 </head>
13
14 <body>
15
16   <p>My email address is
17   <a href = "mailto:deitel@deitel.com"> deitel@deitel.com
18   </a>. Click the address and your browser will open an
19   email message and address it to me.</p>
20
21 </body>
22 </html>
```

When a user clicks on an email link, an email message addressed to the value of the link will open up.

To create an email link include “mailto:” before the email address in the href attribute.



IMG: Embedding Images

- Example

```
<IMG SRC="SomeFile.gif" ALT="My Dog"  
      WIDTH=400 HEIGHT=300>
```

- Attributes:

- SRC (required)
- ALT (technically required)
- ALIGN (see <BR CLEAR="ALL">)
- WIDTH, HEIGHT
- HSPACE, VSPACE
- BORDER
- USEMAP, ISMAP

Image Alignment, Example

```
<!DOCTYPE HTML >
<HTML>
<HEAD><TITLE>Image Alignment</TITLE></HEAD>
<BODY>
<H1 ALIGN="CENTER">Image Alignment</H1>
<TABLE BORDER=1>
  <TR><TH>Alignment
    <TH>Result
  <TR><TH><CODE>LEFT</CODE>
    <TD><IMG SRC="rude-pc.gif" ALIGN="LEFT"
          ALT="Rude PC" WIDTH=54 HEIGHT=77>
          This positions the image at the left side,
          with text flowing around it on the right.
  <TR><TH><CODE>RIGHT</CODE>
    <TD><IMG SRC="rude-pc.gif" ALIGN="RIGHT"
          ALT="Rude PC" WIDTH=54 HEIGHT=77>
          This positions the image at the right side,
          with text flowing around it on the left.
  ...
</TABLE>
</BODY>
</HTML>
```

Image Alignment, Result

Image Alignment - Netscape

File Edit View Go Communicator Help

Image Alignment

Alignment	Result
LEFT	 This positions the image at the left side, with text flowing around it on the right.
RIGHT	This positions the image at the right side, with text flowing around it on the left. 
TOP	 Here, the image runs into the paragraph and the line containing the image is aligned with the image top.
BOTTOM	 Here, the image runs into the paragraph and the line containing the image is aligned with the image bottom.
MIDDLE	 Here, the image runs into the paragraph and the line containing the image is aligned with the image center.

Document: Done

usemap

```


<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" href="sun.htm"
alt="Sun">
  <area shape="circle" coords="90,58,3" href="mercur.htm"
alt="Mercury">
  <area shape="circle" coords="124,58,8" href="venus.htm"
alt="Venus">
</map>
```

Online image maps



imagemap.org



Create HTML image maps. Choose an image from your computer and highlight areas to create the HTML required for your image.

SELECT IMAGE



<http://www.google.com>

google

<http://alphapeeler.sf.net/me>

alphapeeler

Generated HTML Output

```

<map name="image_map">
  <area alt="google" title="google" href="http://www.google.com"
coords="149,-43,41" shape="circle">
```

How Does it Work?

With the help of our generator creating html imagemaps is free and easy. Simply start by selecting an image from your pc, or load one directly from an external website. Next up create your hot areas using either rectangle, circle or polygon shapes. Creating these shapes is as easy as pointing and clicking on your image. Don't forget to enter a link, title and target for each of them. Then once you're finished simply click Show Me The Code!



Active	Shape	Link	Title	Target	
<input type="radio"/>	Circle	<input type="text" value="http://www.google.com"/>	<input type="text" value="google"/>	<input type="text" value="_top"/>	<input type="button" value="X"/>
<input type="radio"/>	Rect	<input type="text" value="http://alphapeeler.sf.net"/>	<input type="text" value="alphapeeler"/>	<input type="text" value="_top"/>	<input type="button" value="X"/>
<input checked="" type="radio"/>	Poly	<input type="text" value="http://www.cnn.com"/>	<input type="text" value="cnn"/>	<input type="text" value="_top"/>	<input type="button" value="X"/>

Show Me The Code!

X

Generated Image Map Output

```
<!-- Image Map Generated by http://www.image-map.net/ -->


<map name="image-map">
  <area target="_top" alt="google" title="google" href="http://www.google.com" coords="148,68,39" shape="circle">
  <area target="_top" alt="alphapeeler" title="alphapeeler" href="http://alphapeeler.sf.net" coords="384,95,627,262" sha
  <area target="_top" alt="cnn" title="cnn" href="http://www.cnn.com" coords="807,-1,775,21,747,41,726,54,707,67,695
</map>
```



Close

Tables

- Template

```
<TABLE BORDER=1>  
  <CAPTION>Table Caption</CAPTION>  
  <TR><TH>Heading1</TH>      <TH>Heading2</TH></TR>  
  <TR><TD>Row1 Col1 Data</TD><TD>Row1 Col2 Data</TD></TR>  
  <TR><TD>Row2 Col1 Data</TD><TD>Row2 Col2 Data</TD></TR>  
  <TR><TD>Row3 Col1 Data</TD><TD>Row3 Col2 Data</TD></TR>  
</TABLE>
```

Table Caption	
Heading1	Heading2
Row1 Col1 Data	Row1 Col2 Data
Row2 Col1 Data	Row2 Col2 Data
Row3 Col1 Data	Row3 Col2 Data

TABLE Element Attributes

- **ALIGN**
 - The ALIGN attribute gives the horizontal alignment of the table as a whole
 - Legal values are LEFT, RIGHT, and CENTER, with LEFT being the default
- **BORDER**
 - This specifies the width in pixels of the border around the table
 - This is in addition to the border around each cell (the CELLSPACING).
 - The default is zero, which also results in the visible 3D divider between cells being turned off
- **CELLSPACING**
 - This gives the space in pixels between adjacent cells. Drawn as a 3D line if BORDER is nonzero, otherwise empty space in the background color is used
 - The default is usually about 3

TABLE Element Attributes

- **CELLPADDING**
 - CELLPADDING determines the empty space, in pixels, between the cell's border and the table element
 - The default is usually about 1
- **WIDTH**
 - This specifies the width of the table, either in pixels (<TABLE WIDTH=250>) or as a percentage of the current browser window width (<TABLE WIDTH="75%">)
- **BGCOLOR**
 - Specify the background color of the table TABLE (also legal for TR, TD, and TH)
- **BORDERCOLOR, BORDERCOLORDARK, BORDERCOLORLIGHT**
 - *Non standard attributes supported by IE to specify the colors to user for the borders*

TABLE Element Attributes

- **BACKGROUND**
 - This nonstandard attribute supported by IE gives an image file that will be tiled as the background of the table
 - You might want to use style sheets instead.
- **RULES**
 - The rules attribute specifies which parts of the inside borders that should be visible.
 - All are drawn if this attribute is omitted
 - Legal values are NONE, ROWS, COLS, and ALL
 - not supported in HTML5, Use CSS instead.
- **FRAME (used for borders)**
 - Specifies which outer borders are drawn
 - All four are drawn if this attribute is omitted
 - Legal values are BORDER or BOX (all), VOID (none), ABOVE (top), BELOW (bottom), HSIDES (top and bottom, despite the somewhat confusing name), VSIDES (left and right), LHS (left), and RHS (right)

Table CAPTION

- Attribute
 - ALIGN (Values: TOP, BOTTOM)
- Usage
 - An enclosing borderless table may give more flexibility than the built-in CAPTION.

Table Caption	
Heading1	Heading2
Row1 Col1 Data	Row1 Col2 Data
Row2 Col1 Data	Row2 Col2 Data
Row3 Col1 Data	Row3 Col2 Data

TR: Table Row

- TR is used to define each row in the table
- Each row will then contain TH and/or TD entries
- ALIGN
 - ALIGN (legal values LEFT, RIGHT, or CENTER) is used to set the default horizontal alignment for table cells
- VALIGN
 - VALIGN (legal values TOP, MIDDLE, or BOTTOM) is used to set the default vertical alignment for table cells
- BGCOLOR
 - Sets the color for the table row, overriding any values set for the table as a whole via the BGCOLOR attribute of TABLE
- BORDERCOLOR, BORDERCOLORDARK,
- BORDERCOLORLIGHT
 - Supported only by Internet Explorer, these specify the colors to use for the row borders

Table Cells: TH and TD

- COLSPAN
 - COLSPAN defines a heading or cell data entry that spans multiple columns

```
<TABLE BORDER=1>
<TR><TH COLSPAN=2>Col 1&2 Heading
    <TH>Col3 Heading
<TR><TD>Col1 Data
    <TD>Col2 Data
    <TD>Col3 Data
</TABLE>
```

Col 1&2 Heading	Col3 Heading
Col1 Data	Col2 Data

Table Cells: TH and TD

- ROWSPAN
 - ROWSPAN defines a heading or cell data entry that spans multiple rows; similar to COLSPAN
- ALIGN
 - LEFT, RIGHT, CENTER, JUSTIFY and CHAR.
 - E.g., the following aligns entries on a decimal point

```
<TD ALIGN="CHAR" CHAR=".">"
```
- VALIGN
 - TOP, BOTTOM, MIDDLE
- WIDTH, HEIGHT
 - Values in pixels only (no percentages officially allowed)
- NOWRAP
 - Use with caution
- BGCOLOR, BACKGROUND
 - Same as for TABLE and TR

HTML Hex-Colors

	White	#FFFFFF
	Red	#FF0000
	Green	#00FF00
	Blue	#0000FF
	Cyan	#00FFFF
	Yellow	#FFFF00
	Black	#000000
	Dim Grev	#545454
	Forest Green	#238E23
	Grey	#C0C0C0
	Midnight Blue	#2F2F4F
	Orange	#FF7F00
	Spring Green	#00FF7F
	Turquoise	#ADEAEA

Basic HTML - Summary

- A DOCTYPE is required to validate the document
- HTML document should have an enclosing HTML element, a HEAD (TITLE is required) and a BODY
- Hypertext links,
 - can be absolute or relative
 - A link to a named section is denoted by #*section*
- Tables are composed of main table element, <TABLE>; rows, <TR>; table headers, <TH>; and table data, <TD>
 - Use BGCOLOR to give background colors to tables, rows, or cells
 - Use ROWSPAN or COLSPAN to join cells

HTML Frames - Outline

- Advantages and disadvantages of frames
- FRAME template
- Defining rows and cols in a FRAMESET
- Common FRAME and FRAMESET attributes
- Nested frames
- Targeting a document to a named FRAME cell

HTML Frames

- Vertical frameset
- Horizontal frameset
- You can display more than one HTML document in the same browser.
- Webmaster should keep track of more HTML documents.
- Difficult to print the entire page
 - Print friendly page.

Frame Advantages

- Certain parts of the interface (e.g., a TOC) are always on the screen
- Can avoid retyping common sections of multiple Web pages
- Consistent use across a large site sometimes simplifies user navigation
- A convenient way to mix text-oriented HTML with Java applets
- Image maps are more convenient if the map image remains on screen and only the results section changes

Frame Disadvantages

- The meaning of the “Back” and “Forward” buttons can be confusing
- Poorly designed frames can get the user lost
- Hard to find real URL of a page you want
 - Printing problems!
- Hard to bookmark "configuration"
- Some very old browsers do not support frames
- Security
 - Hackers can insert frame cells into your pages in some circumstances, perhaps stealing information intended for your site

Frame Template

```
<!DOCTYPE HTML>
<HTML>
<HEAD><TITLE>Document Title</TITLE></HEAD>

<FRAMESET . . .>
  <!-- FRAME and Nested FRAMESET Entries -->
<NOFRAMES>
  <BODY>
    <!-- Stuff for non-Frames browsers -->
  </BODY>
</NOFRAMES>
</FRAMESET>
</HTML>
```

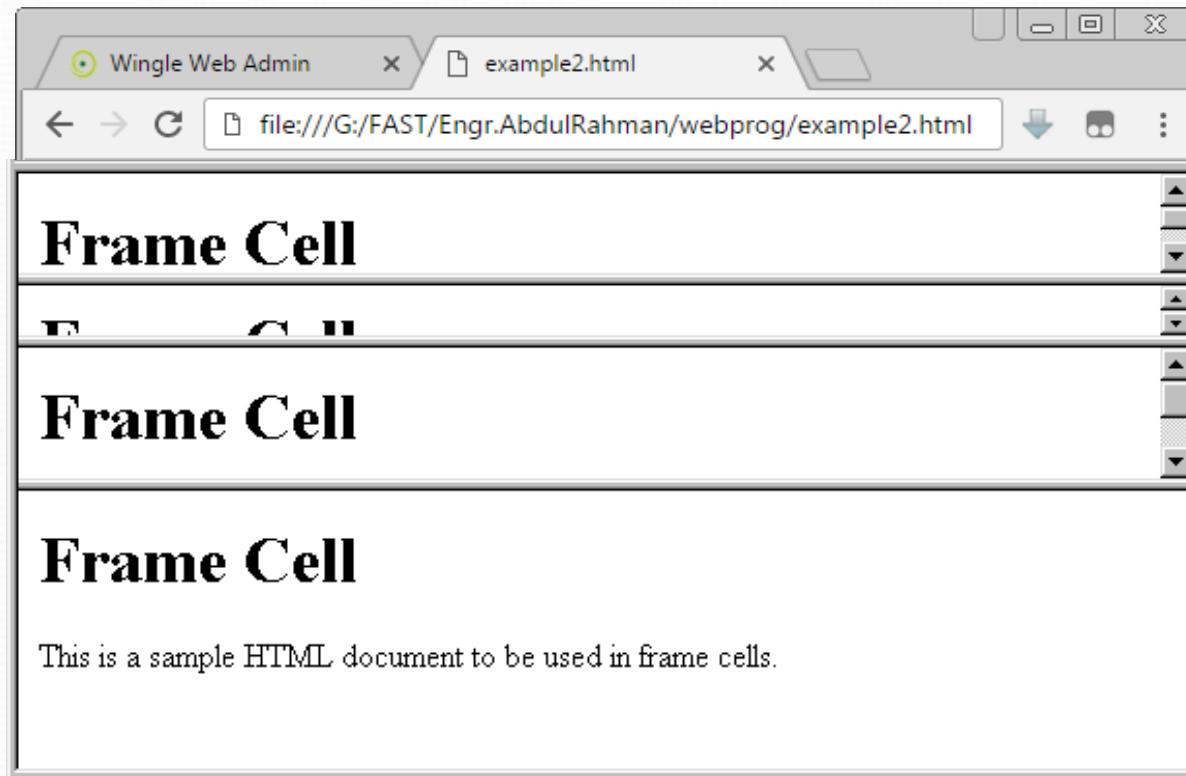
FRAMESET Attributes

- COLS, ROWS
 - A comma-separated list of pixel values, percentages, and weighted remainders
 - FRAMESET entries should *always* specify at least two rows or columns. Netscape problems if not!
 - Examples

```
<FRAMESET ROWS="50,10%,*,2*">
  ...
</FRAMESET>
```

```
<FRAMESET COLS="25%,*,*">
  ...
</FRAMESET>
```

FRAMESET ROWS, Example



```
<FRAMESET ROWS="50,10%,*,2*">
```

...

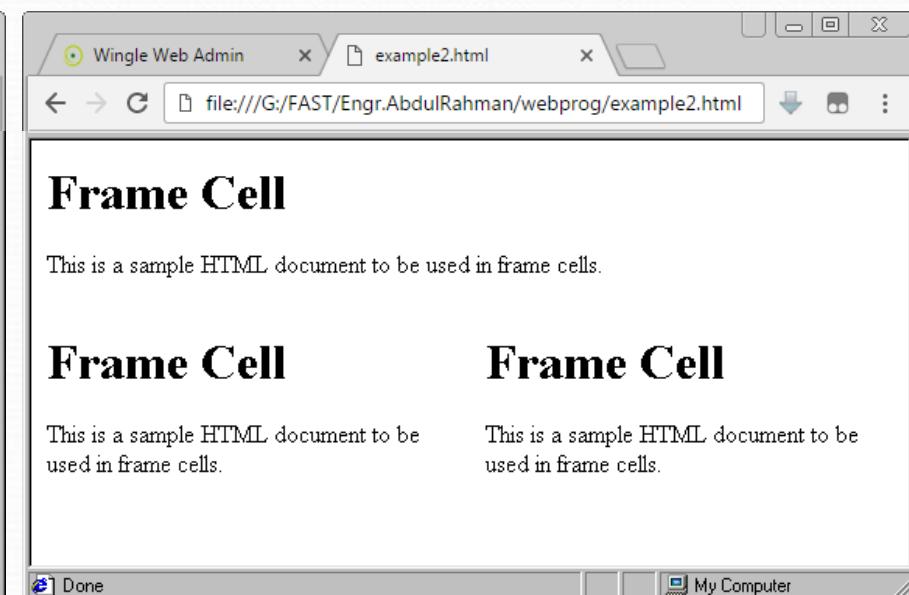
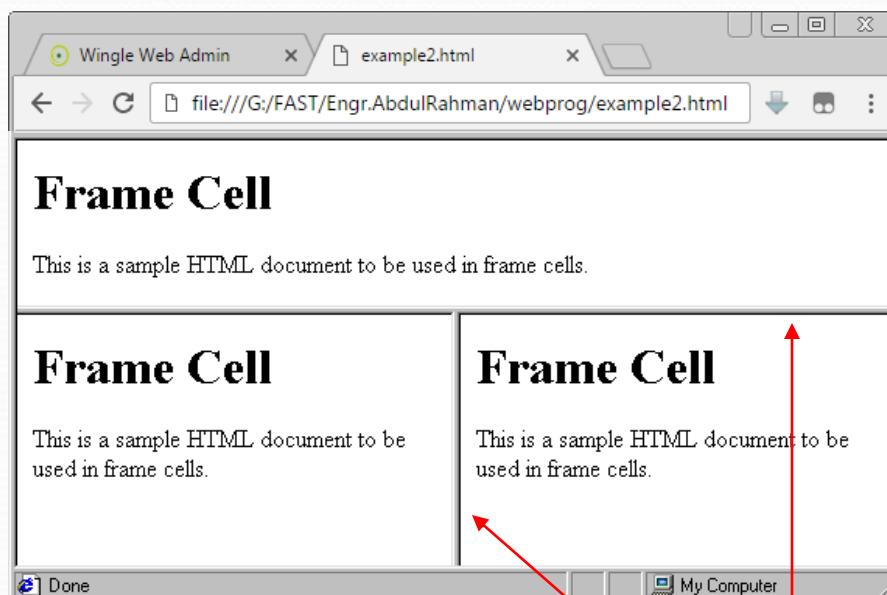
```
</FRAMESET>
```

50 pixels, 10%, 1/3, 2/3

FRAMESET Attributes

- FRAMEBORDER
 - Indicates whether borders will be drawn *between* frame cells
 - YES or 1 specifies borders; NO or 0 specifies no border
 - Can be overridden by FRAMEBORDER settings in individual FRAME entries
 - Often used in conjunction with BORDER=0 and FRAMESPACING=0
- BORDER (Netscape), FRAMESPACING (IE)
 - Specify the thickness of the border between cells
 - Apply to outermost FRAMESET only
- BORDERCOLOR
 - Sets the color of the border between cell, using either a hex RGB value or color name

Frame Border, Examples



Frame Border

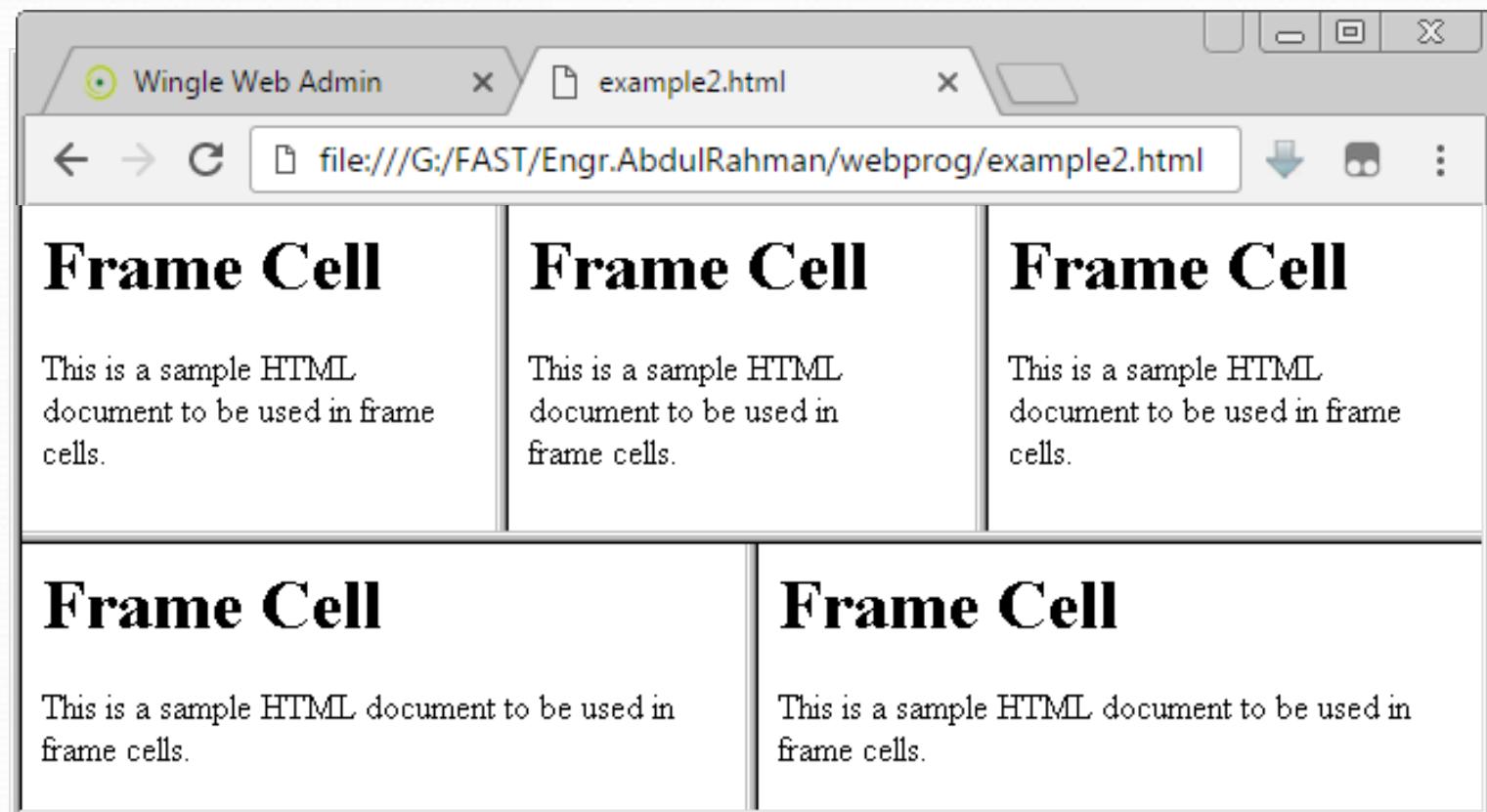
FRAME: Specifying Content of Frame Cells

- SRC
 - URL of the document to place in the frame cell
- NAME
 - Supplies destination for TARGET attribute of hypertext links
- FRAMEBORDER, BORDERCOLOR
- MARGINWIDTH, MARGINHEIGHT
 - Specifies the left/right and top/bottom cell margins, respectively
- SCROLLING
 - Indicates whether cells should have scrollbars
- NORESIZE
 - Disables the ability to resize the frame cells

Frame Example 1

```
<!DOCTYPE HTML >
<HTML>
<HEAD><TITLE>Frame Example 1</TITLE></HEAD>
<FRAMESET ROWS="55%, 45%">
    <FRAMESET COLS="*, *, *">
        <FRAME SRC="Frame-Cell.html">
        <FRAME SRC="Frame-Cell.html">
        <FRAME SRC="Frame-Cell.html">
    </FRAMESET>
    <FRAMESET COLS="*, *">
        <FRAME SRC="Frame-Cell.html">
        <FRAME SRC="Frame-Cell.html">
    </FRAMESET>
    <NOFRAMES>
        <BODY>
            Your browser does not support frames. Please see
            <A HREF="Frame-Cell.html">non-frames version</A>.
        </BODY>
    </NOFRAMES>
</FRAMESET>
</HTML>
```

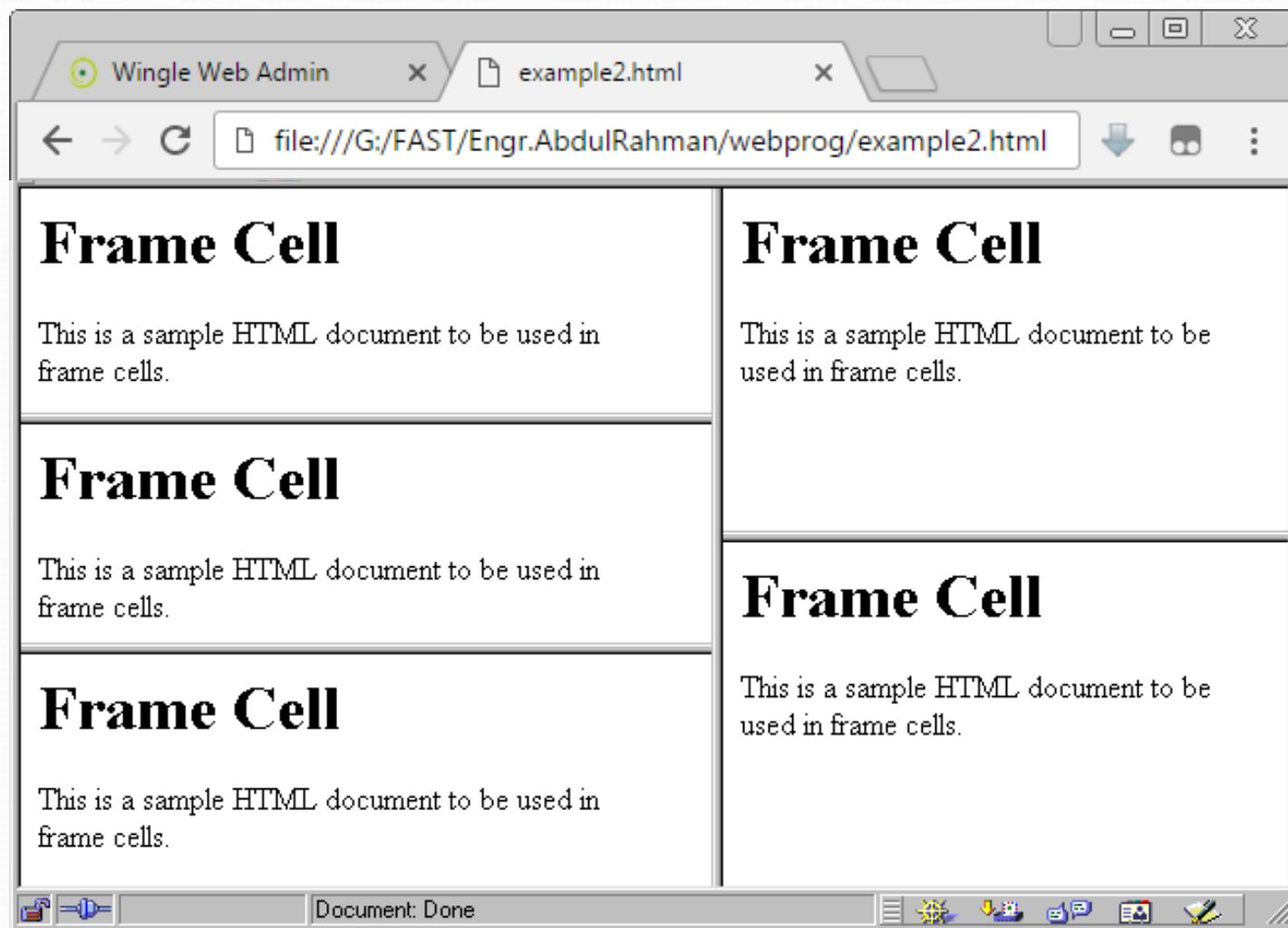
Frame Example 1, Result



Frame Example 2

```
<!DOCTYPE HTML>
<HTML>
<HEAD><TITLE>Frame Example 2</TITLE></HEAD>
<FRAMESET COLS="55% , 45%">
    <FRAMESET ROWS="*,*,*">
        <FRAME SRC="Frame-Cell.html">
        <FRAME SRC="Frame-Cell.html">
        <FRAME SRC="Frame-Cell.html">
    </FRAMESET>
    <FRAMESET ROWS="*,*">
        <FRAME SRC="Frame-Cell.html">
        <FRAME SRC="Frame-Cell.html">
    </FRAMESET>
<NOFRAMES>
    <BODY>
        Your browser does not support frames. Please see
        <A HREF="Frame-Cell.html">nonframes version</A>.
    </BODY>
</NOFRAMES>
</FRAMESET>
</HTML>
```

Frame Example 2, Result



Targeting Frame Cells

- Specify the cell in which to place a page referenced by a hyperlink
- The NAME Attribute of FRAME

```
<FRAME SRC="..." NAME="cellName">
```

- The TARGET Attribute of A HREF

```
<A HREF="..." TARGET="cellName">
```

Targeting Example

[Introduction](#) [Potential](#) [Investing](#) [References](#)

Introduction

A new breakthrough in cold fusion technology!

A ripe opportunity for the lucky investor!

Accepted by the scientific community! Documented in *The Journal of Irreproducible Results*.

Document Done

Cold-Fusion.html

```
<!DOCTYPE HTML>
<HTML>
<HEAD>
  <TITLE>Investing in Cold Fusion</TITLE>
</HEAD>
<FRAMESET ROWS="75,*">
  <FRAME SRC="TOC.html" NAME="TOC">
  <FRAME SRC="Introduction.html" NAME="Main">
<NOFRAMES>
  <BODY>
    This page requires Frames. For a non-Frames version,
    <A HREF="Introduction.html">the introduction</A>.
  </BODY>
</NOFRAMES>
</FRAMESET>
</HTML>
```

TOC.html

```
<!DOCTYPE >
<HTML>
<HEAD>
    <TITLE>Table of Contents</TITLE>
</HEAD>
<BODY>
<TABLE WIDTH="100%">
    <TR><TH><A HREF="Introduction.html" TARGET="Main">
        Introduction</A></TH>
    <TH><A HREF="Potential.html" TARGET="Main">
        Potential</A></TH>
    <TH><A HREF="Investing.html" TARGET="Main">
        Investing</A></TH>
    <TH><A HREF="References.html" TARGET="Main">
        References</A></TH></TR>
</TABLE>
</BODY>
</HTML>
```

Targeting Example, Results

The screenshot shows a web browser window with a light blue header and a white content area. At the top, there is a horizontal menu bar with four items: Introduction, Potential, Investing, and References. Below the menu, the main content area features a large, bold, black title: **Investing in Cold Fusion Technology**. To the right of the title is a vertical scroll bar. Below the title, there is a paragraph of text: "To take advantage of this **once in a lifetime** opportunity, send check or money order to:". Underneath this, there is an address: "Cold Fusion
PO Box 255". At the bottom of the window, there is a toolbar with several icons: a magnifying glass, a mail icon, a question mark, and other standard browser controls. The status bar at the bottom displays the text "Document Done".

Predefined Frame Names

- _blank
 - Opens the linked document in a new window or tab
- _top
 - Causes the linked document to take up the whole browser window
 - Document will not be contained in a frame cell
- _parent
 - Places document in the *immediate* FRAMESET parent
 - Same as _top if no nested frames
- _self
 - Place document in current cell
 - Only necessary to override a BASE entry

HTML Frames - Summary

- Frames require a Frameset DOCTYPE for validation
- A FRAMESET can be divided either into columns or rows
 - To create both rows *and* columns use nested FRAMESETS
- By giving a FRAME a name, documents can be targeted to the named frame cell
 - <FRAME ... NAME="...">
 -
- There are four predefined frame names
 - _blank, _top, _parent, and _self

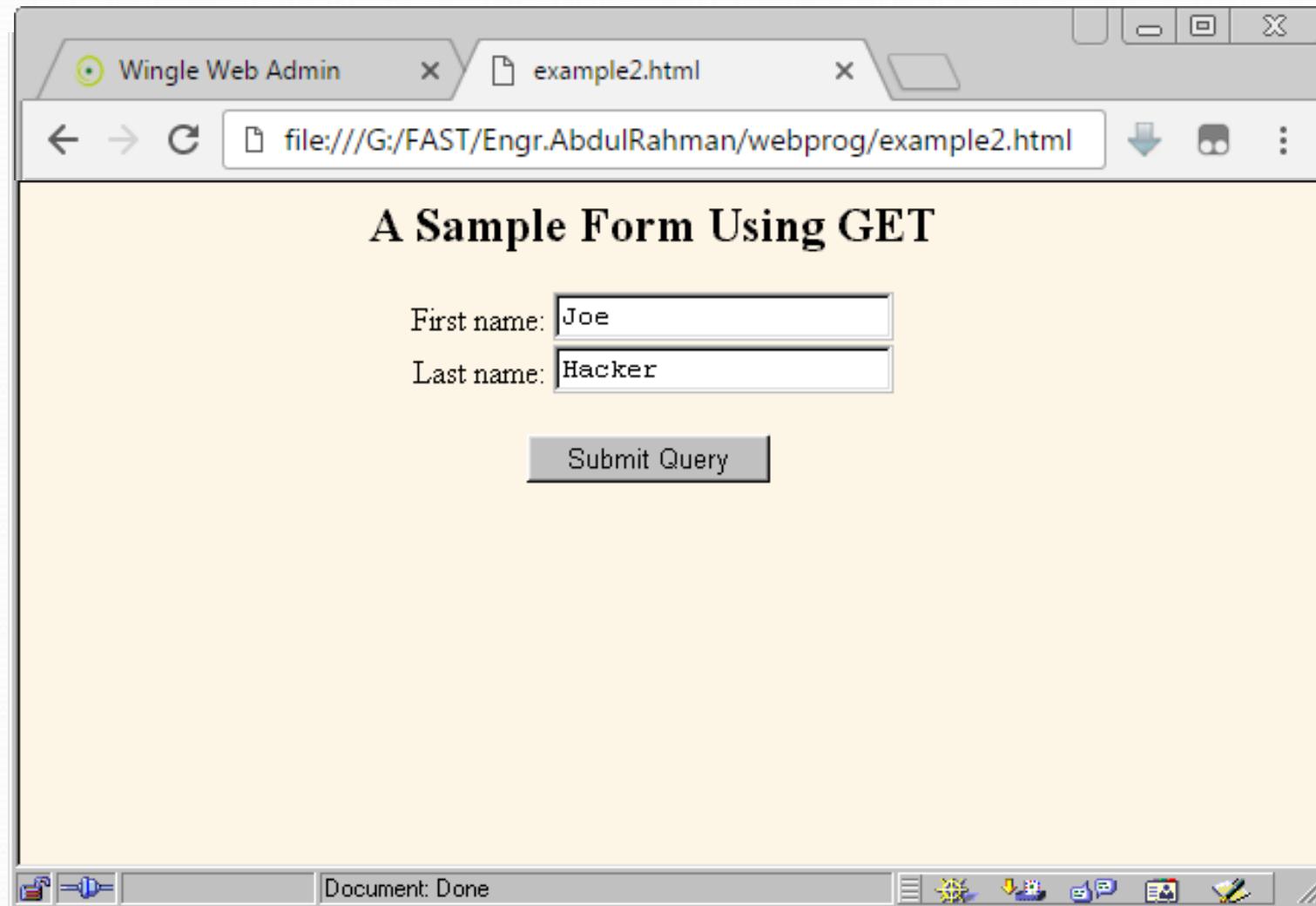
Sending Data with GET

```
...
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using GET</H2>

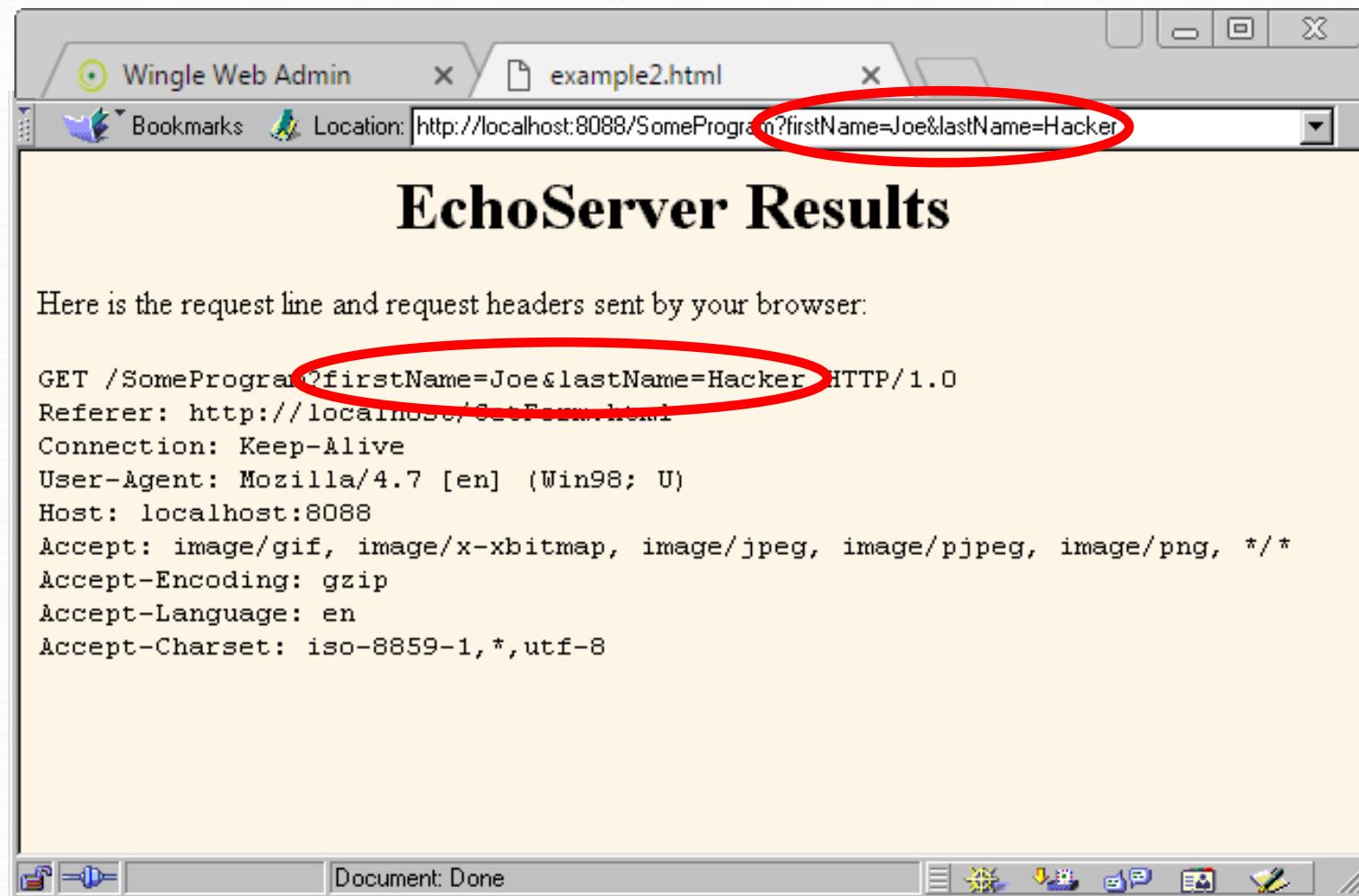
<FORM ACTION="http://localhost:8088/SomeProgram">
<CENTER>
First name:
<INPUT TYPE="TEXT" NAME="firstName" VALUE="Joe"><BR>
Last name:
<INPUT TYPE="TEXT" NAME="lastName" VALUE="Hacker"><P>
<INPUT TYPE="SUBMIT">
</CENTER>
</FORM>

</BODY></HTML>
```

Initial Result



Submission Result



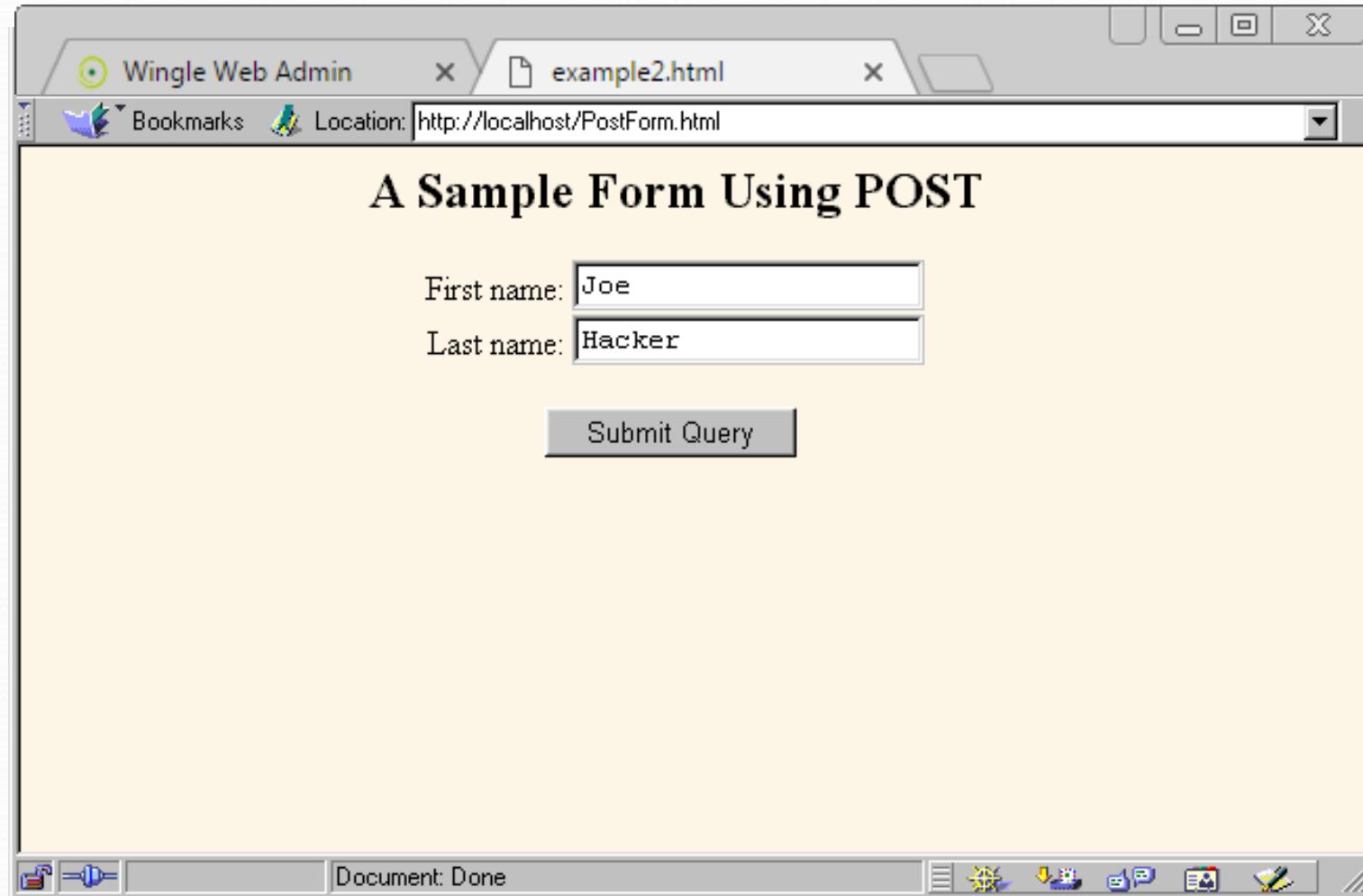
Sending Data with POST

```
...
<BODY BGCOLOR="#FDF5E6">
<H2 ALIGN="CENTER">A Sample Form Using POST</H2>

<FORM ACTION="http://localhost:8088/SomeProgram"
      METHOD="POST">
  <CENTER>
    First name:
    <INPUT TYPE="TEXT" NAME="firstName"
           VALUE="Joe"><BR>
    Last name:
    <INPUT TYPE="TEXT" NAME="lastName"
           VALUE="Hacker"><P>
    <INPUT TYPE="SUBMIT">
  </CENTER>
</FORM>

</BODY></HTML>
```

Initial Result



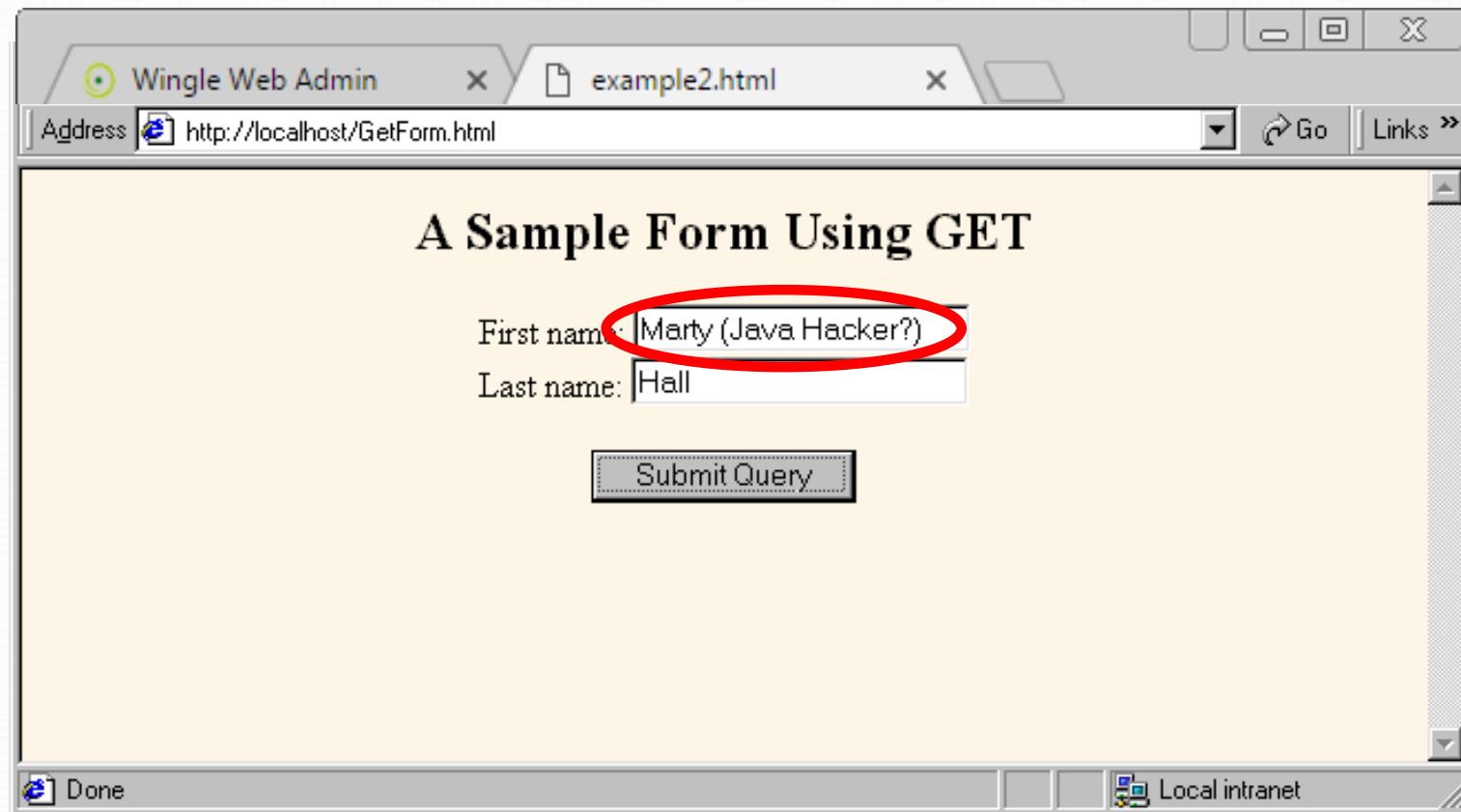
Submission Result

The screenshot shows a Windows-style web browser window. The title bar says "Wingle Web Admin" and the active tab is "example2.html". The location bar shows "http://localhost:8088/SomeProgram". The main content area displays the heading "EchoServer Results" and a message: "Here is the request line and request headers sent by your browser:". Below this, a large block of text represents the HTTP request. A red oval highlights the URL-encoded parameters "firstName=Joe&lastName=Hacker" at the bottom of the list.

```
POST /SomeProgram HTTP/1.0
Referer: http://localhost/PostForm.html
Connection: Keep-Alive
User-Agent: Mozilla/4.7 [en] (Win98; U)
Host: localhost:8088
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1, *, utf-8
Content-type: application/x-www-form-urlencoded
Content-length: 29

firstName=Joe&lastName=Hacker
```

URL Encoding: Original Form



URL Encoding: Result



Text Controls

- Textfields
 - `<INPUT TYPE="TEXT" NAME="..." ...>`
 - `VALUE` can give original value

- Password Fields

- `<INPUT TYPE="PASSWORD" NAME="..."`
 - *Always use POST*



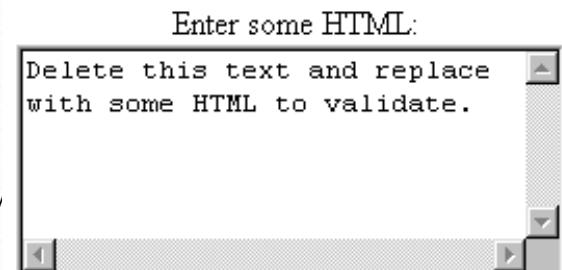
- Text Areas

- `<TEXTAREA NAME="..." ROWS="..." COLS="...">`

...

`</TEXTAREA>`

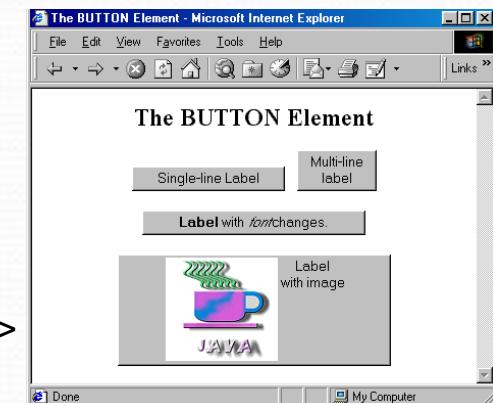
- Interpretation of regular HTML tags turned off between `</TEXTAREA>`



Push Buttons

- Submit Buttons
 - <INPUT TYPE="SUBMIT" ...>
 - Use NAME if you have multiple buttons
 - Use VALUE to change button's label
- Reset Buttons
 - <INPUT TYPE="RESET" ...>
 - Use VALUE to change button's label
- JavaScript Buttons
 - <INPUT TYPE="BUTTON" onClick="someJavaScriptFunction()" ...>
- Fancy Buttons
 - <BUTTON TYPE="SUBMIT" ...>HTML</BUTTON>
 - Internet Explorer and Netscape 6 only

Submit Query



Using Multiple Submit Buttons

```
<CENTER>
Item:
<INPUT TYPE="TEXT" NAME="Item" VALUE="256MB
SIMM"><BR>
<INPUT TYPE="SUBMIT" NAME="Add"
      VALUE="Add Item to Cart">
<INPUT TYPE="SUBMIT" NAME="Delete"
      VALUE="Delete Item from Cart">
</CENTER>
```

Item: 256MB SIMM

Add Item to Cart

Delete Item from Cart

Check Boxes

- Format
 - <INPUT TYPE="CHECKBOX" NAME="..." ...>
 - The CHECKED attribute makes it initially checked
 - Name/value pair sent only if checkbox is checked when form is submitted

- Example code

```
<p>
<input type="checkbox" name="noEmail" checked>
Check here if you do <i>not</i> want to
get our email newsletter
```

- Example result

Check here if you do *not* want to get our email newsletter

Radio Buttons

- Format
 - <INPUT TYPE ="RADIO"
NAME="..." VALUE="..." ...>
 - All radio buttons in a group should have same NAME
 - Only one button in a group can be pressed; pressing a different one causes previous one to pop out
- Example

```
<DL>
  <DT>Credit Card:
  <DD><INPUT TYPE="RADIO" NAME="creditCard"
          VALUE="visa">
    Visa
  <DD><INPUT TYPE="RADIO" NAME="creditCard"
          VALUE="mastercard">
    Master Card
  ...
</DL>
```

Credit Card:

Visa

Master Card

Java Smart Card

American Express

Discover

Combo Boxes

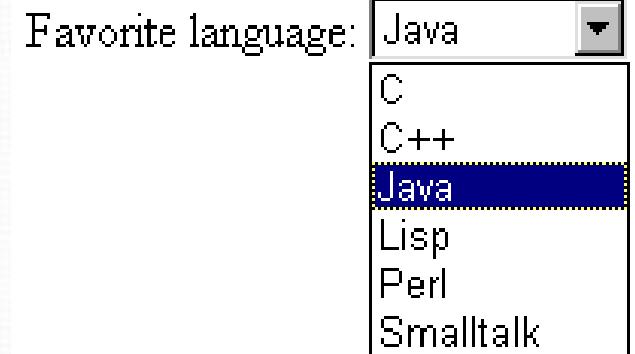
- Format
 - SELECT gives NAME
 - OPTION gives VALUE



- Example

Favorite language:

```
<SELECT NAME="language">  
  <OPTION VALUE="c">C  
  <OPTION VALUE="c++">C++  
  <OPTION VALUE="java" SELECTED>Java  
  <OPTION VALUE="lisp">Lisp  
  <OPTION VALUE="perl">Perl  
  <OPTION VALUE="smalltalk">Smalltalk  
</SELECT>
```



List Boxes

- Format
 - Identical to combo boxes, but specify MULTIPLE
- Example

Languages you know:


```
<SELECT NAME="language" MULTIPLE>
  <OPTION VALUE="c">C
  <OPTION VALUE="c++">C++
  <OPTION VALUE="java" SELECTED>Java
  <OPTION VALUE="lisp">Lisp
  <OPTION VALUE="perl" SELECTED>Perl
  <OPTION VALUE="smalltalk">Smalltalk
</SELECT>
```

Languages you know:



Other Controls and Options

- File upload controls
 - Lets user select a file and send it to the server
- Server-side image maps
 - User clicks on an image and form gets submitted.
 - Form data gets sent as *name.x=x-pos&name.y=y-pos*
- Hidden fields
 - Preset NAME and VALUE sent with form submission..
- Grouping Controls
 - FIELDSET lets you visually group forms.
 - Internet Explorer and Netscape 6 only.
- Tab order control
 - TABINDEX (Internet Explorer and Netscape 6 only)

HTML Form - Example

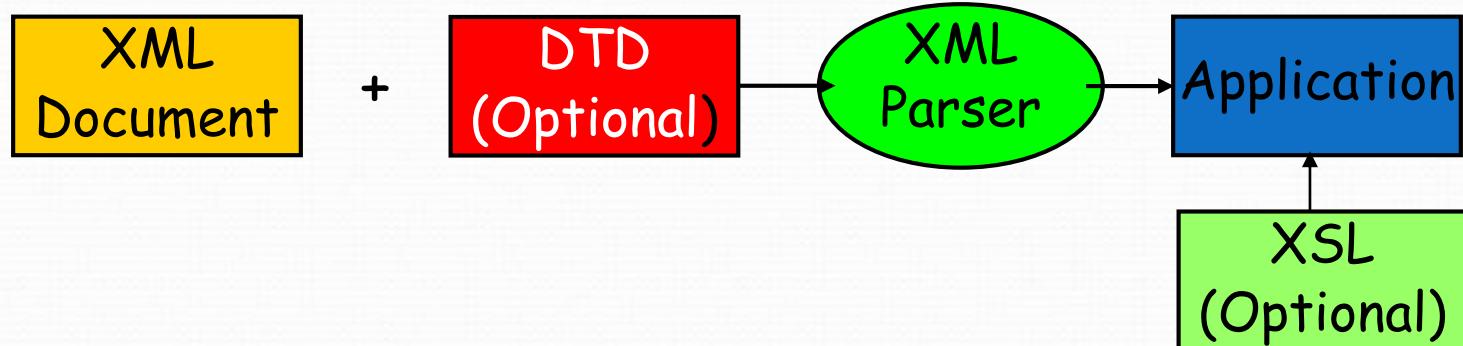
The screenshot shows a web browser window with the title "Wingle Web Admin" and the tab "example2.html". The URL bar displays "file:///G:/FAST/Engr.AbdulRahman/webprog/example2.html". The main content area is titled "Feedback Form" and contains the following text:
Please fill out this form to help us improve our site.
Name: Tem Nieto
Comments:
I really like your Web site. It has improved my HTML skills by leaps and bounds.
Email address: [REDACTED]
Things you liked:
Site design Links Ease of use Images Source code
How did you get to our site?:
Search engine Links from another site Deitel.com Website Reference in a book Other
Rate our site (1-10): 8
Buttons: Submit Your Entries | Clear Your Entries
Status Bar: Done | Local intranet

HTML Forms - Summary

- General process
 - FORM uses ACTION to specify base URL
 - Input elements each have a NAME
 - User enters values
 - When form submitted, URL is
baseURL?name1=value1&name2=value2&...
 - For POST requests, name/value pairs sent on separate line (not part of URL)
- Textfields
 - <INPUT TYPE="TEXT" ...>
- Submit Buttons
 - <INPUT TYPE="SUBMIT" ...>

XML

- XML: Extensible Markup Language
 - was developed in 1996 by the World Wide Web Consortium (W3C) XML working group
 - Designed to carry data
 - Has all 3 components of SGML:
 - structure (DTD)
 - content (XML)
 - style (XSL)



XML

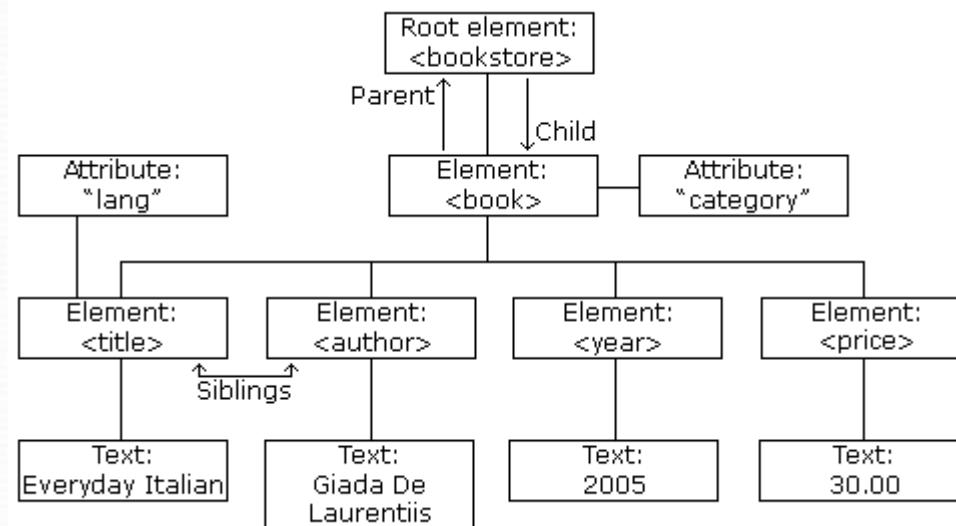
- XML documents contain only data
 - Applications decide how to display the data
 - Files end in the `.xml` extension
 - Highly portable
 - Language for creating markup languages
 - Can create new tags
- Document Type Definition (DTD) files
 - Defines grammatical rules for the document
 - Used to check the XML document structure against
- Extensible Style Language (XSL) files
 - Defines additional information for rendering the document
 - Possible to search, sort, manipulate and render XML using Extensible Style Language (XSL)

An Example XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="web">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

XML prolog

XML Tree Structure



XML Syntax Rules

- XML prolog
- Tags are Case Sensitive
- XML Attribute Values Must be Quoted
- INCORRECT:

```
<note date=28/01/2017>
  <to>Fahad</to>
  <from>Aziz</from>
</note>
```

- CORRECT:

```
<note date="28/01/2017">
  <to>Fahad</to>
  <from>Aziz</from>
</note>
```

XML Syntax Rules

- **Entity References:**
 - This will generate an XML error:
 - <message> salary < 1000 </message>
 - To avoid this error, replace the "<" character with an **entity reference**:
 - <message>salary < 1000</message>
- **Comments in XML**
 - <!-- This is a comment -->
- **White-space is Preserved in XML**
 - XML does not truncate multiple white-spaces (HTML truncates multiple white-spaces to single white-space):
 - XML: Hello Aziz
 - HTML: Hello Aziz

XML Syntax Rules

- XML Stores New Line as LF
- Well Formed XML
 - XML documents that conform to the syntax rules above are said to be "Well Formed" XML documents.
 -

XML Elements are Extensible

- <note>
 <to>Aziz</to>
 <from>Fahad</from>
 <body>Let's goto Airport at 9 AM</body>
 </note>

MESSAGE To: Aziz
From: Fahad
Let's goto Airport at 9 AM

- Assume that we created an app that extracted the <to>, <from>, and <body> elements from XML to produce the output.

- Now Assume that we added some extra info to it:

- <note>
 <to>Aziz</to>
 <from>Fahad</from>
 <heading> Reminder</heading>
 <body>Let's goto Airport at 9 AM</body>
 </note>