



COSC 2206 Internet Tools

Brief Survey of HTML and XHTML
Document Structure
Formatting



W3C HTML Home page

- W3C is the World Wide Web Consortium and their home page has lots of information, links, and a short history of HTML

www.w3.org/MarkUp/



The HTML 4.01 Specification

The World Wide Web Consortium (W3C) is responsible for the HTML specification.

www.w3.org/TR/html4/



The XHTML 1.0 Specification

- This is the latest specification of HTML that conforms to the XML specification: every XHTML document is also an XML document

www.w3.org/TR/2001/WD-xhtml1-20011004/



Web Design Group reference

- This is a comprehensive reference for HTML 4.01 including a complete description of every HTML tag and its attributes.

www.htmlhelp.com/

local link

<wdghtml40\index.html>



HTML and XHTML Tutorials

- Interactive tutorials on HTML and XHTML from w3schools

www.w3schools.com/



web monkey reference

- Lots of information here on HTML and web design

www.webmonkey.com/



Document structure (1)

- Originally HTML was intended as a markup language to describe the logical structure of a document (page description language)
- The browser would be entirely responsible for the the physical layout and formatting.
- This didn't happen:
 - Microsoft and Netscape included tags such as `` and `<size>` which describe physical formatting
 - HTML became an undesirable mixture of logical structure and formatting.



Document structure (2)

- HTML 3.2 mixed the two concepts together so there are a lot of web pages that contain both logical and physical formatting.
- Fortunately HTML 3.2 has been replaced by HTML 4 and CSS (cascading styles sheets).
- Now all physical formatting can be specified by the style sheet.
- An entire web site can be given a different look just by changing a style sheet.



Document Structure (3)

- All tags that describe physical formatting have been deprecated in HTML 4.
 - Example: `` and `<size>`
- This means that you shouldn't use them in new documents.
- Document Type Definitions (DTD's) are used to specify a documents type (grammar)
- For HTML 4 there is the transitional type and the strict type



Document Structure (4)

- The transitional type permits the use of the deprecated tags (elements)
- The strict type does not allow the use of deprecated tags.
- Software exists that can validate a document using its DTD.
- Later we will show how to use it.



HTML and XML

- Recently XML (Extensible Markup Language) has been developed for describing data with user defined tags and attributes.
- To make HTML compatible with XML, HTML 4 has been slightly modified so that an HTML document is also an XML document
- The new HTML is called XHTML 1.0
- Version 1.1 now exists too.

HTML versions

W3C

(World Wide Web Consortium)
is responsible for web
standards

www.w3.org

HTML , HTML +

HTML 3.2

RIP

HTML 4

XML

XHTML 1.X



W3C Quote on XHTML

- "The XHTML family is the next step in the evolution of the internet. By migrating to XHTML today, content developers can enter the XML world with all of its attendant benefits, while still remaining confident in their content's backward and future compatibility."



Structure of an HTML4 doc

- The first line of an HTML 4.01 document should be the document type definition: the DTD.
- It should be one of the following types
 - Strict
 - Transitional
 - Frameset



Strict DTD for HTML4

```
<!DOCTYPE HTML PUBLIC  
    "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">
```




Transitional DTD for HTML4

```
<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/1999/REC-html401-  
19991224/loose.dtd">
```



Frameset DTD for HTML4

```
<!DOCTYPE HTML PUBLIC  
"-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/1999/REC-html401-  
19991224/frameset.dtd">
```

Frames are not part of the XHTML 1.1 specification.
Frames are better done using CSS (cascading style sheets).



HTML 4.01 Template

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
  "http://www.w3.org/TR/html4/strict.dtd">
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>My first HTML document</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
  <P>Hello world!
```

```
</BODY>
```

```
</HTML>
```

tags must be closed
in XHTML

tags must be
lowercase in XHTML

htmlTemplate.html



Structure of an XHTML doc

- The first line of an XHTML 1.0 document should be the document type definition: the DTD.
- It should be one of the following types
 - Strict
 - Transitional
 - Frameset



Strict DTD for XHTML 1.0

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



Transitional DTD XHTML 1.0

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



Frameset DTD XHTML 1.0

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



XHTML 1.0 Template

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

DOCTYPE declaration goes here

```
<html xmlns="http://www.w3.org/1999/xhtml"
```

```
    xml:lang="en" lang="en">
```

```
<head>
```

```
<title>My first XHTML document</title>
```

```
</head>
```

```
<body>
```

```
    <p>Hello world!</p>
```

```
</body>
```

```
</html>
```

[xhtmlTemplate.html](#)



XHTML 1.1 Template

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.1/EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="en" lang="en">
<head>
<title>My first XHTML document</title>
</head>
<body>
    <p>Hello world!</p>
</body>
</html>
```

[xhtmlTemplate1.1.html](#)



HTML to XHTML (1)

- Use the XHTML template to make the HTML document into an XML document.
- Document must be well-formed:
 - illegal nesting is not allowed
 - e.g. `text` is illegal and must be replaced by `text`
 - closing tags must not be omitted
 - e.g. `<p>paragraph 1<p>paragraph 2` is illegal and must be replaced by `<p>paragraph 1</p><p>paragraph 2</p>`



HTML to XHTML (2)

- Tags that are self-closing (empty tags) such as `
`, `<hr>`, `<link>` and `` must be written in a special form:
 - e.g. `
` must be written as `
`
 - e.g. `` must be written as ``
- All tags must be written in lowercase since XML is case sensitive:
 - e.g. `<BODY>` must be replaced by `<body>`



HTML to XHTML (3)

- Attribute values must always be quoted
 - e.g. `<table rows = 3>` is illegal and must be replaced by `<table rows = "3">`
- Attribute minimization is not supported
 - e.g. `<dl compact>` is illegal and must be written as `<dl compact="compact">`
- There are a few other rules (see the W3C working XHTML 1.0 draft for details)

www.w3.org



W3C Validator

- The W3C has a program that can validate HTML and XHTML based on the document type.
- You can upload one of your HTML files and have it validated.
- W3C Validation Service:

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



Kinds of tags in HTML

- tags with a begin and end tag: In XHTML the ending tag must always be present and the attribute values must always be in quotes
 - `<tag attribute="..." ... attribute="..."> ... </tag>`
- self ending tags:
 - `<tag attribute="..." ... attribute="..."> in HTML`
 - `<tag attribute="..." ... attribute="..." /> in XHTML`



HTML tag summary (1)

- `<html> . . . </html>`
 - create an HTML document
- `<head> . . . </head>`
 - comes before body to specify tags such as `<title>`, `<meta>`, and `<link>`
- `<title> . . . </title>`
 - title for browser title bar and favorites list
- `<meta> . . . </meta>`
 - describes page content and other document info



HTML tag summary (2)

- Meta tag example

```
<meta http-equiv=Refresh content=
"5; url=http://www.cs.laurentian.ca/fred/">
```

- Here the page is automatically redirected after 5 seconds to the new home page



HTML tag summary (3)

- `<body> . . . </body>`
 - delimits document content
- `<h1> . . . </h1> . . . <h6> . . . </h6>`
 - hierarchy of heading levels
- ` . . . `
 - bold text (better to use css font-weight)
- `<i> . . . </i>`
 - italic text (better to use css font-style)



HTML tag summary (4)

- **<tt> . . . </tt>**
 - inline monospace text (typewriter font)
 - same as **<code> ... </code>**
- **<cite> . . . </cite>**
 - creates a citation in italic text
- ** . . . **
 - emphasize text: also ** . . **
- ** . . . **
 - emphasize text



HTML tag summary (5)

- ` ... `
 - hypertext link
- ` (XHTML form)`
 - create a target location within a document
- ` ... `
 - make a link to target location specified by "NAME" in an HTML document
- Can also link to a location in another page
 - Ex: `"http://www.c.com/doc.html#NAME"`



HTML tag summary (6)

- `<p> . . . </p>`
 - define a paragraph
- `
` (**XHTML form**)
 - generate a line break
- `<blockquote> . . . </blockquote>`
 - indent displayed text at both margins
- `<dl> . . . </dl>`
 - define a definition list



HTML tag summary (7)

- `<dt> . . . </dt>`
 - define a name in list to be defined
- `<dd> . . . </dd>`
 - specify the definition
- ` . . . `
 - define an ordered list (numbered list)
- ` . . . `
 - define an unordered list (bullet list)

HTML tag summary (8)

- ` ... `
 - define a list item in an ordered or unordered list
- `<div class="..."> ... </div>`
 - define a custom block level division
- ` .. `
 - define a custom inline text
- ` (XHTML form)`
 - define an image



HTML tag summary (9)

- `<hr />` (XHTML form)
 - define a horizontal rule. Attributes are deprecated in HTML 4. Use style sheets to specify them.
- `<table> . . . </table>`
 - define a table



HTML tag summary (10)

- `<tr> . . . </tr>`
 - define a table row (many attributes)
- `<th> . . . </th>`
 - define a table header cell (many attributes)
- `<td> . . . </td>`
 - define a table data cell (many attributes)
- `<frameset> . . . </frameset>`
 - define a set of frames



HTML tag summary (11)

- **<frame> . . . </frame>**
 - define a frame in a frameset
- **<noframes> . . . </noframes>**
 - for browsers that don't support frames
- **<form> . . . </form>**
 - define an interactive form
- **<select> . . . </select>**
 - define a menu



HTML tag summary (12)

- `<option />`
 - define a menu option (**XHTML form**)
- `<textarea> . . . </textarea>`
 - define a multi-line text input area
- `<input type="checkbox" />`
 - define a check box (**XHTML form**)
- `<input type="radio" />`
 - define a radio button (**XHTML form**)



HTML tag summary (13)

- `<input type="text" />`
 - define a text input box (**XHTML form**)
- `<input type="submit" />`
 - define a form submit button (**XHTML form**)
- `<input type="image" />`
 - define submit button with image (**XHTML form**)
- `<input type="reset" />`
 - define a form reset button (**XHTML form**)



Comments in HTML

- Comments begin with

`<!--`

and end with

`-->`

- Example

`<!-- this is an HTML comment -->`

- **NEVER USE `--` INSIDE A COMMENT.
NETSCAPE WILL GET VERY CONFUSED**



Character entities

- To use some characters in literal form it is necessary to use character entities.

< **< ;** **< ;**

> **> ;** **> ;**

" **" ;** **" ;**

& **& ;** **& ;**

- There are many others: search for "character entities" online.



HTML Colors

- see

http://www.w3schools.com/html/html_colors.asp

- color names are not supported by the W3C standards even though IE and Netscape understand many color names.
- Correct HTML should use the Color hex codes such as **#F0A03F**
- there is also a set of web safe colors that should display properly on any system.



The meta tag (1)

- Used to specify a character encoding in HTML
 - `<meta http-equiv="Content-Type" content="text/html; charset=utf-8">`
- Used to specify client-pull (browser automatically requests a new document from server)
 - `<meta http-equiv="refresh" content="15">`
 - `<meta http-equiv="refresh" content="1; url=http://...">`



The meta tag (2)

- Used to specify when a document should be considered expired
 - `<meta http-equiv="expires" content="wed 5 Dec 2001 10:00:00 EST">`
- Used to specify document language
 - `<meta http-equiv="content-language" content="fr">`



The meta tag (3)

- Inserting hidden information and data for search engines
 - `<meta name="..." content="...">`
 - `<meta name="description" content="...">`
 - `<meta name="keywords" content="...">`
 - `<meta name="robots" content="noindex, nofollow">`



HTML tables

- Tables were included in HTML for the presentation of data in tabular form.
- Tables are also used for page layout.
 - Example: two column layout



Examples of tables (1)

- A simple table with two rows and two columns

```
<table border="0">
```

```
<tr>
```

```
    <td>Cell 1</td><td>Cell 2</td>
```

```
</tr>
```

```
<tr>
```

```
    <td>Cell 3</td><td>Cell 4</td>
```

```
</tr>
```

```
</table>
```

Cell 1 Cell 2

Cell 3 Cell 4

Examples of tables (2)

- Previous example with a border

Cell 1	Cell 2
Cell 3	Cell 4

```
<table border="1">  
<tr>  
    <td>Cell 1</td><td>Cell 2</td>  
</tr>  
<tr>  
    <td>Cell 3</td><td>Cell 4</td>  
</tr>  
</table>
```

Examples of tables (3)

- Previous example with a border, cellspacing, and cellpadding

```
<table border="10" cellspacing="20" cellpadding="5">
```

```
<tr>
```

```
    <td>Cell 1</td><td>Cell 2</td>
```

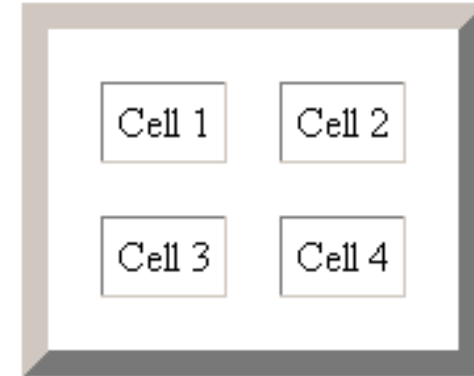
```
</tr>
```

```
<tr>
```

```
    <td>Cell 3</td><td>Cell 4</td>
```

```
</tr>
```

```
</table>
```



Cell 1	Cell 2
Cell 3	Cell 4



Examples of tables (4)

- Spanning multiple columns using colspan

```
<table border="1">
```

```
<tr>
```

```
    <td colspan="2">Cell 1 and Cell 2</td>
```

```
</tr>
```

```
<tr>
```

```
    <td>Cell 3</td><td>Cell 4</td>
```

```
</tr>
```

```
</table>
```

Cell 1 and Cell 2	
Cell 3	Cell 4

Examples of tables (5)

- Spanning multiple rows using rowspan

```
<table border="1">
```

```
<tr>
```

```
    <td rowspan="2">Cell 1 and Cell 3</td>
```

```
    <td>Cell 2</td>
```

```
</tr>
```

```
<tr>
```

```
    <td>Cell 4</td>
```

```
</tr>
```

```
</table>
```

Cell 1 and Cell 3	Cell 2
	Cell 4



Examples of tables (6)

- The directory
`html/examples/tables/`
contains several examples.

[examples/tables/examples.html](#)



Forms

- Forms provide interactive input on a web page
- JavaScript can process a form on the client-side
- Forms can be submitted to a program that runs on the server and processes the information entered in the form (e.g. a Perl or PHP script).



The form tag

- Structure of a form that uses get

```
<form action="..." method="get"
                                target="...">
    define form components here
</form>
```

- Structure of a form that uses post

```
<form action="..." method="post"
                                target="...">
    define form components here
</form>
```



Form components (1)

- `<input type="button" name="..." value="..." />`
 - creates a button that can be used to trigger functions created with JavaScript. Data for this form element is never sent to the server
- `<input type="checkbox" checked="..." name="..." value="..." />`
 - creates a check box that will be checked by default if the checked attribute is specified. The value specifies whether the check-box is checked or not using "off" and "on" (default)



Form components (2)

- `<input type="file" name="..." accept="..." />`
 - can be used to upload a local file to the server using a browse button
- `<input type="hidden" name="..." value="..." />`
 - Associates hidden values with the form that are not visible to the user but are sent to the server when the form is submitted.



Form components (3)

- `<input type="image" ... />`
 - replaces a submit button with an image
- `<input type="password" ... />`
 - Creates a text input field in which a password can be typed. The characters typed are not displayed. The password sent to the server is not encrypted.
- `<input type="reset" />`
 - Creates a reset button that can be used to clear a form to its default values.



Form components (4)

- `<input type="radio" checked="..." name="..." value="..." />`
 - creates a radio button that can be turned on or off. In a group only one button can be on. Data from the on button is submitted
- `<input type="submit" name="..." value="..." />`
 - Creates a button used to submit the form data to the server for processing.



Form components (5)

- `<input type="text" size="..."
maxlength="..." name="..."
value="..." />`
 - creates a text input box of a given size for a maximum number of characters.
- `<select>...</select>`
 - Defines multiple choice menu or scrolling list. Each element in the list is specified by `<option>` tag
- `<option> ... </option>`
 - defines an option for a select list



Form components (6)

- `<textarea name="..." cols="..."`
 `rows="..." wrap="...">`
 default text goes here
 `</textarea>`
- creates a multi-line text input area with a specified number of rows and columns.



Text field example (1)

```
<h2>Please enter your first and last names</h2>
<form action="" method="">
<table border="0" cellspacing="5">
<tr><td>First name:</td>
    <td><input type="text" size="20"
        name="firstName"></td></tr>
<tr><td>Last name:</td>
    <td><input type="text" size="20"
        name="lastName"></td></tr>
</table>
<input type="submit"><input type="reset">
</form>
```



Text field example (2)

Please enter your first and last names

First name:

Last name:

Submit Query

Reset

[examples/forms/form1.html](#)



List box example (1)

```
<h2>Choose your province</h2>
<form action="" method="">
<select name="province" size="1">
<option value="BC">British Columbia</option>
<option value="AB">Alberta</option>
<option value="SA">Saskatchewan</option>
<option value="MA">Manitoba</option>
<option value="ON">Ontario</option>
</select>
<input type="submit"><input type="reset">
</form>
```

List box example (2)

Choose your province

Before choosing

British Columbia ▼	Submit Query	Reset
--------------------	--------------	-------

Choose your province

After choosing

Saskatchewan ▼	Submit Query	Reset
<div>British Columbia</div> <div>Alberta</div> <div>Saskatchewan</div> <div>Manitoba</div> <div>Ontario</div>		

<examples/forms/form2.html>



check box radio button (1)

```
<form action="/cgi-bin/formPost.pl" method="post">
<p>Which of the following types of wine do you
like?</p>
<input type="checkbox" name="like"
  value="Shiraz" checked="checked">Shiraz
<input type="checkbox" name="like"
  value="Merlot">Merlot
<input type="checkbox" name="like"
  value="CabernetSauvignon">Cabernet Sauvignon
<input type="checkbox" name="like"
  value="Burgundy">Burgundy
<input type="checkbox" name="like"
  value="Zinfindel">Zinfindel
```



check box radio button (2)

```
<p>What is your favourite type of wine?</p>
<input type="radio" name="favourite"
      value="Shiraz" checked="checked">Shiraz
<input type="radio" name="favourite"
      value="Merlot">Merlot
<input type="radio" name="favourite"
      value="CabernetSauvignon">Cabernet Sauvignon
<input type="radio" name="favourite"
      value="Burgundy">Burgundy
<input type="radio" name="favourite"
      value="Zinfandel">Zinfandel
<input type="submit"><input type="reset">
</form>
```



Check box radio button (3)

Which of the following types of wines do you like?

☒ Shiraz ☒ Merlot ☒ Cabernet Sauvignon ☐ Burgundy ☐ Zinfandel

What is your favourite type of wine?

☒ Shiraz ☐ Merlot ☐ Cabernet Sauvignon ☐ Burgundy ☐ Zinfandel

Submit Query

Reset

<examples/forms/form3.html>

Note: Later we will process forms like this on server side with PHP)



Frames

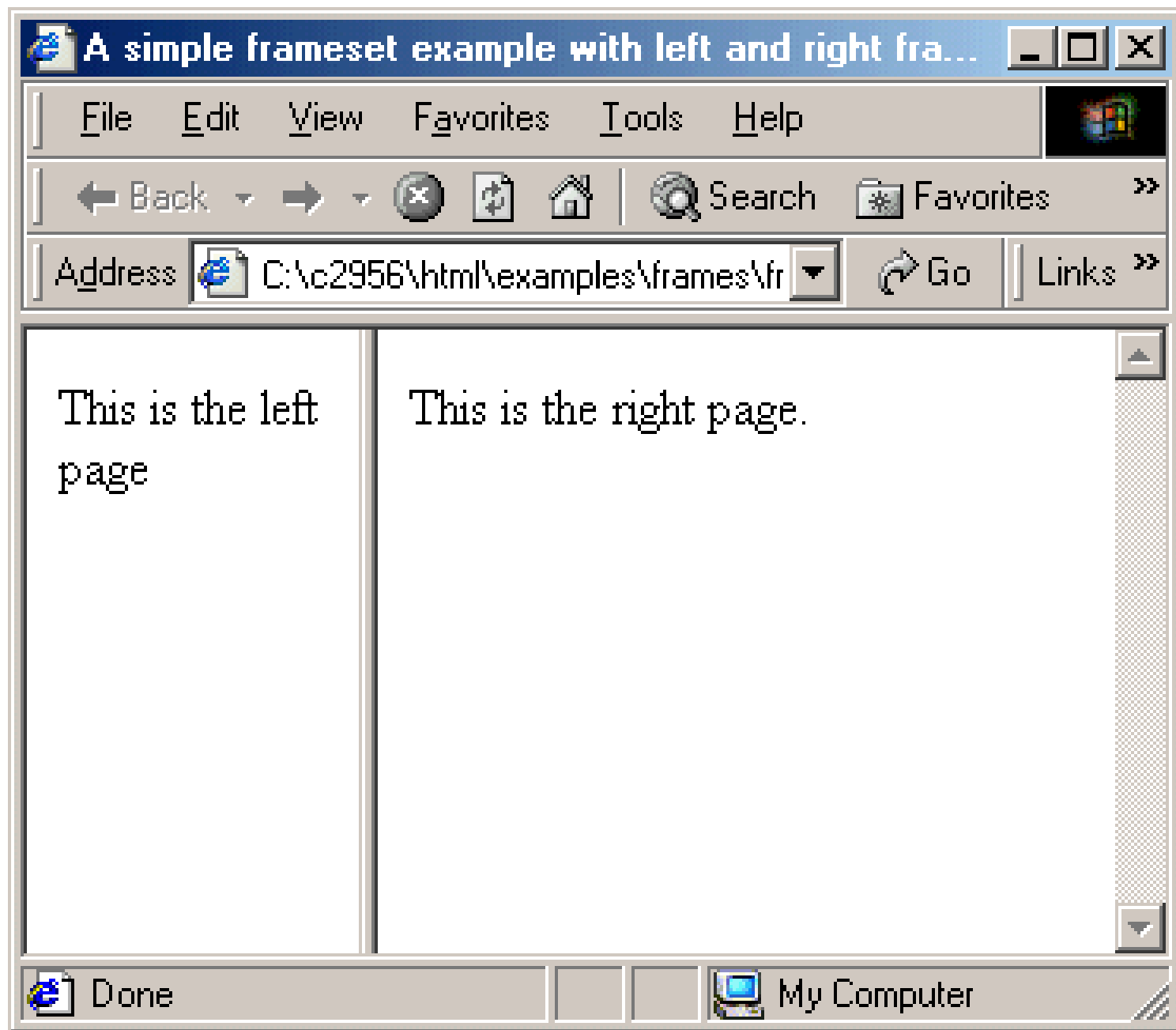
- Frames divide the browser window into several frames, each of which can display a different document.
- Frames are controversial and can be difficult to navigate.
- A common use is to have a non-scrolling area at the top of the browser window which could be used for a navigation bar or an advertising banner.



Side by side frames

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
<html>
<head><title>...</title></head>
<frameset cols="30%,70%">
    <frame src="left.html" name="left">
    <frame src="right.html" name="right"
        scrolling="yes">
</frameset>
</html>
```

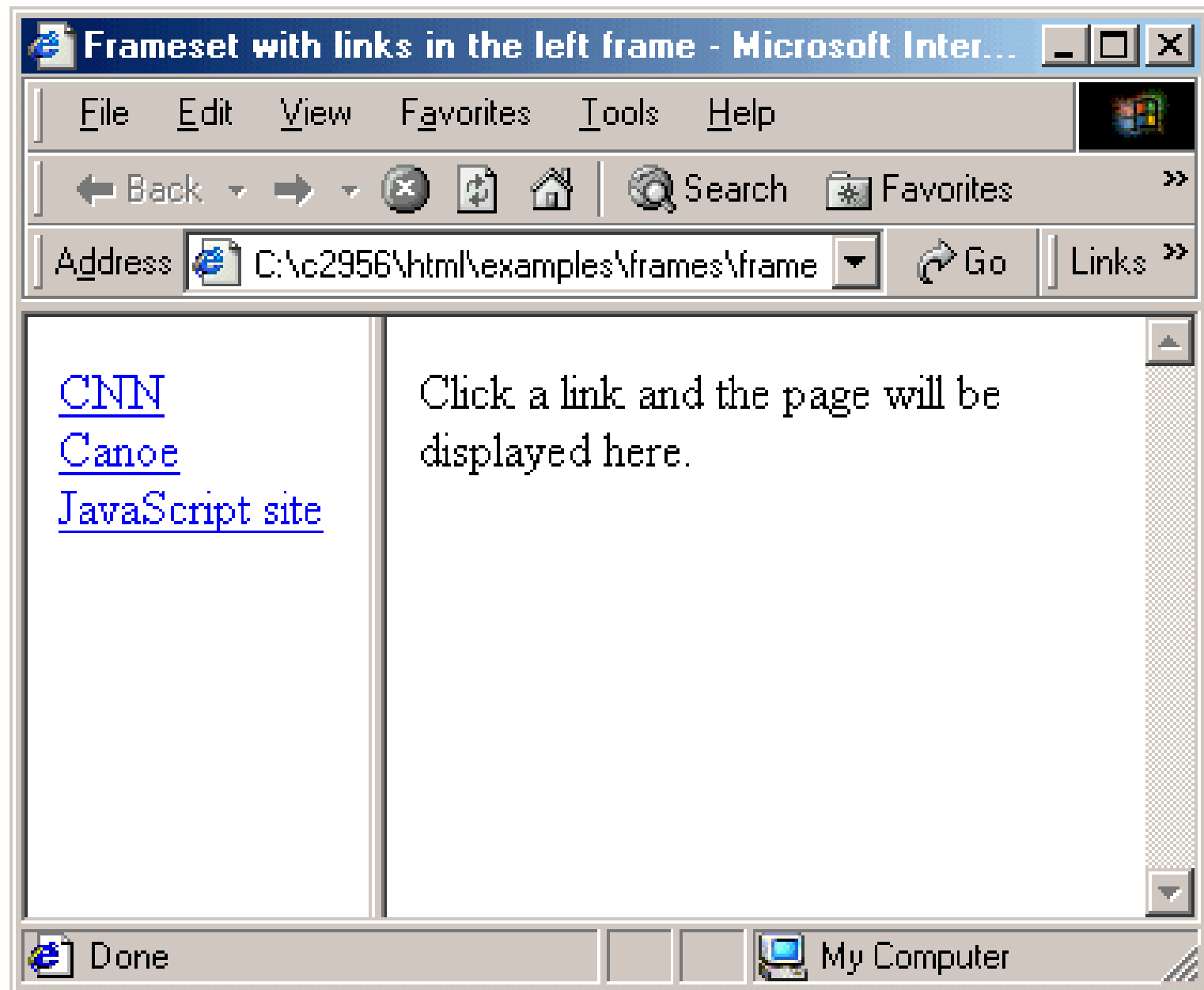
<examples/frames/frameSet1.html>





A frame with links (1)

```
<frameset cols="30%,70%">  
  <frame src="links.html" name="links">  
  <frame src="content.html" name="content"  
    scrolling="yes">  
</frameset>
```





The links page

```
<a href="http://www.cnn.com" target="content">CNN</a><br>
```

```
<a href="http://www.canoe.ca" target = "content">Canoe</a><br>
```

```
<a href="http://www.javascript.com" target = "content">JavaScript site</a>
```

Note how the target refers to the name defined in the frame that should display the document

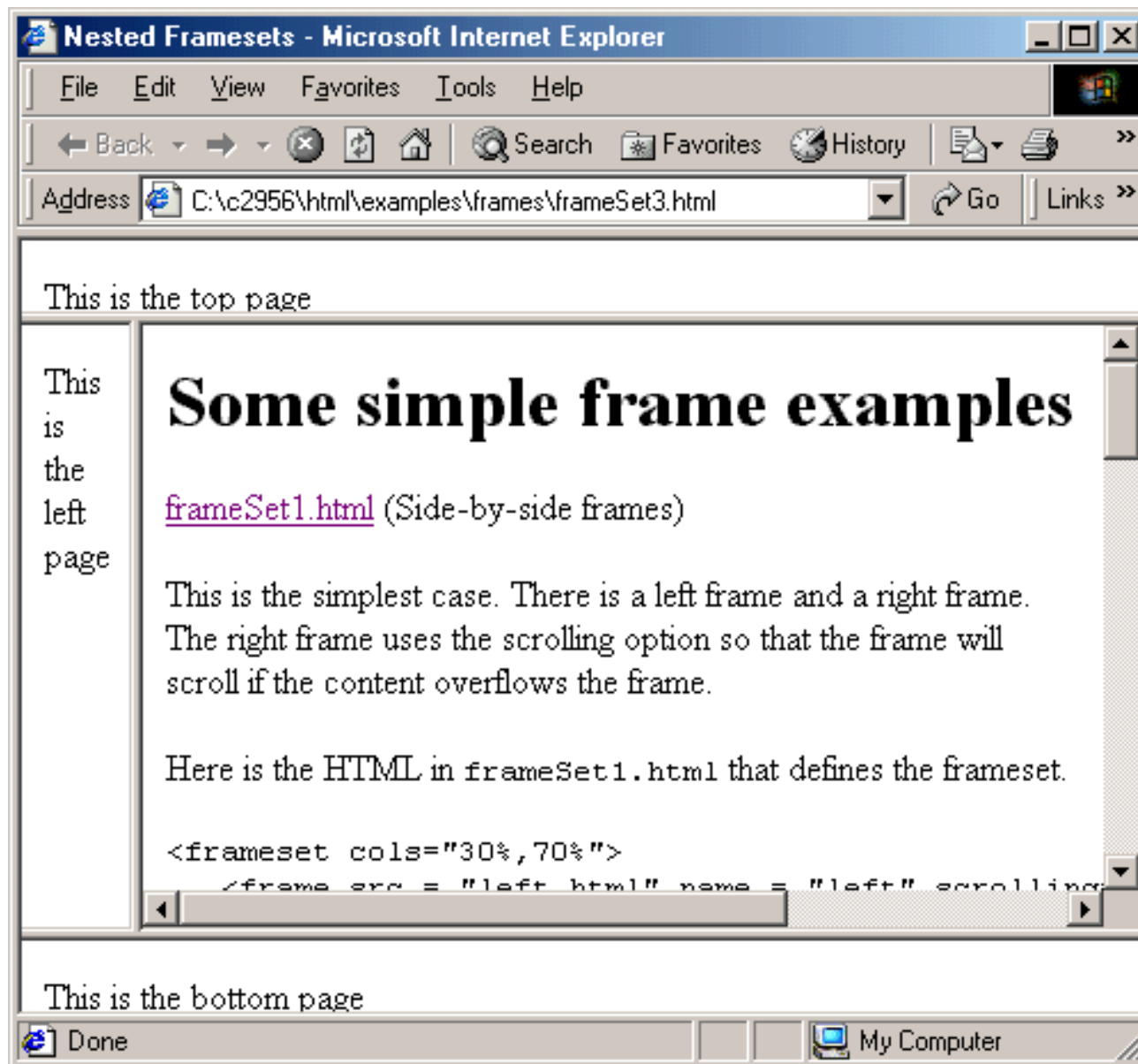
<examples/frames/frameSet2.html>



Nested frame sets

```
<frameset rows="10%,*,10%">
  <frame src="top.html" name="top"
    scrolling="no">
  <frameset cols="10%,*">
    <frame src="left.html" name="left">
    <frame src="right.html" name="right"
      scrolling="yes">
  </frameset>
  <frame src="bottom.html" name="bottom"
    scrolling="no">
</frameset>
```

<examples/frames/frameSet3.html>



Targeting frames

- The frame name can be referred to in a hypertext link

page to load when
link is followed

- `... `

- `<frame src="..." name="frameName">`

default page



Online books (1)

- Frames are useful for displaying books online
- A left frame can hold the table of contents as links
- A right frame can display the content pages
- Alternatively the table of contents can be in a top frame and a bottom frame can be used to display the content pages.



Online books (2)

- Chapter 11 of Intro to CS text book:
 - side by side frames with style sheet

<examples/javaChapter11/chapter11Frames1.html>

- top and bottom frames with style sheet

<examples/javaChapter11/chapter11Frames2.html>

- without style sheet

<examples/javaChapter11/chapter11Frames3.html>



Online books (3)

■ The side by side frameset

```
<frameset cols="20%,70%">  
    <frame src = "toc.html" name = "toc">  
    <frame src = "chapter11.html" name = "page">  
</frameset>
```

■ The top and bottom frameset

```
<frameset rows="20%,70%">  
    <frame src = "toc.html" name = "toc">  
    <frame src = "chapter11.html" name = "page">  
</frameset>
```



Online books (4)

- Each chapter can be in a separate html file. Anchors can be used to give names to specific parts of a document
- For example the following anchor can be places just before section 1
 - ``
- It can be referenced by a hypertext link in the same document using
 - `link text`



Online books (5)

- Anchor example from chapter11

```
<a name="section1">
```

```
<h1>1 Introduction</h1>
```

```
...
```

```
<a name="section2">
```

```
<h2>2 The basic structure ... </h2>
```

```
...
```



Online books (6)

- Then in the table of contents (toc.html) we would use

```
<li>
```

```
<a href="chapter11.html#section1"  
    target="page">1 Introduction</a>
```

```
</li>
```

```
<li>
```

```
<a href="chapter11.html#section2"  
    target="page">2 The basic ...</a>
```

```
</li>
```



Online books (7)

- It can be referenced in another document such as chapter7.html using the hypertext link

- `link text`

<examples/javaChapter11/example.html>