

(X)HTML

(eXtensible HyperText Markup Language)



Representation and Management of Data on the Internet, CS Department, Hebrew University, 2007

W3Schools [XHTML tutorial](#), [HTML tutorial](#)

1

What is HTML?

- A Markup Language for representing documents
 - text (data)
 - structure
 - appearance
 - functionality
- Designed for writing Web pages
- Traditionally rendered by Web browsers (nowadays also by help components, about windows, messengers, e-mail clients... anything for which an HTML renderer is powerful enough – replaced RTF in many of these)

2

Capabilities of HTML

- Content presentation
 - Structures, e.g., paragraphs, lists, tables, etc.
 - Decorations, e.g., fonts, images, etc.
- Declaration of meta information
 - e.g., the page title, language, etc.
- Linkage to other pages
 - i.e., attaching links to components

3

Capabilities of HTML (cont)

- Management of user input
 - e.g., for searching, making reservations, ordering products
- Directions for browsers
 - e.g., refresh, redirect, caching control, etc.

4

A simple HTML page

```
<html>
<head>
  <title>An HTML Page</title>
</head>
<body>
  <h1 id="hdr1">Hello World Wide Web!</h1>
</body>
</html>
```

Filename ends with **.htm** or **.html** [Open hello.html](#)

5

HTML Version History

- HTML 1.0 (first draft) – 1992
- HTML 2.0 (proposed standard) – September 1995
 - From this point on - W3C recommendations
- HTML 3.2 – January 1997
 - added *tables*, *applets*, ...
- HTML 4.0 – December 1997
 - improved *tables*, *forms*, ...

6

HTML Version History (cont)

- **HTML 4.01** – December 1999
 - slightly different from 4.0
- **XHTML 1.0** – January 2000 *Our Examples*
 - reformulation of HTML 4.01 as an **XML** application
 - stricter and cleaner syntax, formatting moved to CSS.
- **XHTML 1.1** – May 2001
 - “Modularization of XHTML”

[Read more about The History of HTML](#)

7

(X)HTML Support in Real Life

- Non-standard / mixed HTML will, most of the time, display, more-or-less as expected, on most browsers, but its appearance may vary between browser types and between browser versions.
- Even when using completely valid and standard HTML, always check your pages on more than one browser type (at least IE & some Mozilla).
- in this course the exercises will only be checked on Mozilla SeaMonkey.

8

Basic HTML Syntax

- (X)HTML contains *text*, separated by *tags*
- Tags come in **pairs**: an *opening* tag and a *closing* tag
- Tags can have *attributes*, which have *values*

```
<html>
<head><title>An HTML Page</title></head>
<body>
<h1 id="hdr1">Hello World Wide Web!</h1>
</body>
</html>
```

9

Basic HTML Syntax (cont)

- An HTML page is surrounded by the **html** tag
- 2 Basic parts:
 - **head**: general information about the document (e.g., title – shown on the browser bar)
 - **body**: the content of the document

Actually a tree structure is created

```
<html>
<head><title>An HTML Page</title></head>
<body>
<h1 id="hdr1">Hello World Wide Web!</h1>
</body>
</html>
```

10

XHTML Stricter Syntax

- Element and attribute names must be in *lower case*
 - HTML allows any case combination, e.g., <Body></BODY>, which conveniently helped distinguish tags from the body text.
- All tags must have corresponding *closing tags*
 - Use
 as a shorthand for
</br>
- Elements should be *properly nested*
 - e.g., hello world is illegal!
- Attribute values must be quoted
 - e.g., <td rowspan="3"> and not <td rowspan=3>

Line break

11

Document Type Definitions

- You should specify which XHTML standard you are using.
- Put a *document type definition* (DTD) at the first line of your file (before the **html** tag).
- For an example, see the next slide.

12

Document Type Definitions (cont)

- XHTML - *strict*

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

- XHTML - *transitional* (loose)

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

Transition `html` → `xhtml`:
allows some legacy
formatting outside CSS

- XHTML - *frameset* (for using frames)

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

13

Text: Structures and Decoration

14

Basic Structures

- In principle, all body elements must reside in **structures** such as the ones below
- Heading: `<h1>`, ..., `<h6>`
 - `h1` is the most important
- Paragraph: `<p>`
- Quotation block: `<blockquote>`
 - separated and indented block

15

Structure Example

```
<h1>This is the main header</h1>
```

```
<h2>And this is a secondary header</h2>
```

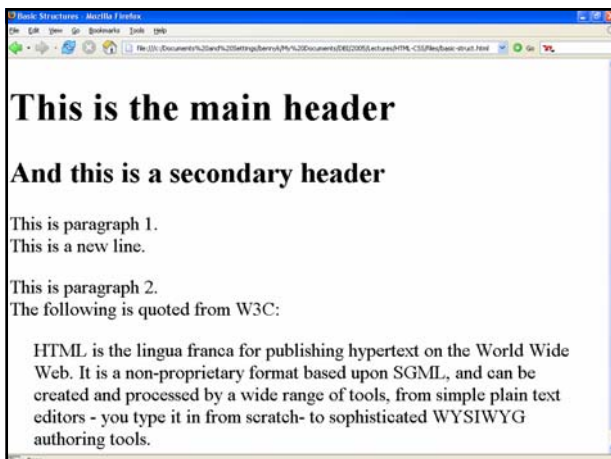
```
<p>This is paragraph 1.<br/>This is a new line.</p>
```

```
<p>This is paragraph 2. <br/>
The following is quoted from W3C:</p>
```

```
<blockquote> <p>...</p>
</blockquote>
```

[Open basic-struct.html](#)

16



Lists

- Ordered (numbered) list: ``
- Unordered list: ``
- List item: ``
- Lists can be nested
 - That is, one list can reside in an item of another

18

A List Example

```

<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
<ul>
  <li>Item A
    <ul><li>First inner item</li>
      <li>Second inner item</li></ul>
  <li>Item B</li>
  <li>Item C</li>
</ul>

```

Open lists.html

19

Tables

Many people use tables for page reformatting

- Define a table using the `<table>` tag
 - Use attributes to specify the border weight, cell-padding, etc.
- A table contains rows: `<tr>`
- Each row contains data cells: `<td>`
 - May contain text, images, lists, tables, etc.
- For heading cells use: `<th>`

21

A Table Example

```

<table border="1">
  <tr>
    <td>Item A</td>
    <td>Item B</td>
  </tr>
  <tr>
    <td>Item C</td>
    <td/>
  </tr>
</table>

```

empty cell

Item A	Item B
Item C	

Open simple-table.html, complex-table.html

22

Text Formatting

- Formatted text:
 - `bold`
 - `<i>italic</i>`
 - `<big>big</big>` `<small>small</small>`
- Preserve spaces and line breaks: `<pre>...</pre>`
 - Outside this tag, any positive amount of white space characters is considered as one space character.
 - Text consisting solely of white spaces is completely ignored.
- Line break: `
`
- Horizontal ruler: `<hr/>`

23

An Example

```

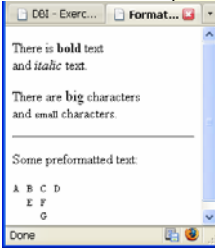
<html>
<head><title>Formatting Example</title></head>
<body>
  <p>
    There is <b>bold</b> text <br/> and
    <i>italic</i> text. <br/></p>
  <p>
    There are <big>big</big> characters<br/>
    and <small>small</small> characters. </p>

```

24

An Example (cont)

```
<hr/>
<p>Some preformatted text:</p>
<pre>
A B C D
E F
G
</pre>
</body>
</html>
```



Open format.html, format2.html

25

Hyperlinks

26

Basic Links

- Basic form:

```
<a href = "target-url">linked-fragment</a>
```

- For example:

```
<a href = "http://www.google.com">Google</a>
```

- The *linked fragment* should contain only *inline* elements

- e.g., text, images, , <i>, etc.

27

Link Targets

When should we use complete paths?

When should we use relative paths?

- The target URL can be absolute

```
<a href = "http://www.google.com">google</a>
```

- Or relative to the folder of the page

```
<a href = "assignments/ex1.html">Exercise 1</a>
```

- Or relative to the server

```
<a href = "/~dast/">DAST</a>
```

What will be the target URL if the link is at <http://www.cs.huji.ac.il/~dbi/index.html>?

28

Anchors / References

- Using anchors, you can define hyperlinks between components in the same page

- To define an anchor, use:

```
<a id = "anc_name">anchored-fragment</a>
```

- name="anc_name" will also work

- To link to an anchor, use:

```
<a href = "#anc_name">linked-fragment</a>
```

Of course you can add an (absolute or relative) URL before the anchor

29

More Hyperlinks

```
<a href = "mailto:dbi@cs.huji.ac.il">Email</a>
```

```
<a href = "news:local.course.dbi">Newsgroup</a>
```

```
<a href = "telnet://aleph@ram0.huji.ac.il">
```

Connect to the Library

This is not a closed list! Many protocols are constantly emerging (e.g., ed2k://, which links to an interesting technology which we will not discuss) or fading away (e.g., gopher://, which lost the battle against http/html for being too strict and lacking in graphics support

30

Forms

31

Forms

- Forms enable users to pass parameters to applications on the WEB.
- For example, search queries, credit-card numbers for online commerce, etc.



32

Registration Form

Of course, all submitted information is kept in the strictest confidence. You always retain control of your information with D. W. Simpson & Co. Our services are always at no cost to you.

Name:

Address:

City: State/Province:

Zip Code:

Country:

Phone: (H) Fax:

Phone: (W) (if able to speak freely)

Email:

Company Name: (optional)

Desired Region: Other Other:

Professional designations earned:

College Student Actuarial Student ACAS FCAS ASA
 FSA EA CFA MBA MAIAA FIA FCIA
 FIAA Financial Engineer M.S. Ph.D. Data Mining Quant

Other Designations:

33

Forms

- A form has the following structure:

```
<form method="method" action="URL">
  HTML with Form Widgets
  (assignments to form parameters)
</form>
```

- The **method** is either **get** or **post** (remember the difference? Hint: request content).
- The **URL** specifies the application that processes the form parameters.

34

Forms (cont)

- Each widget is associated with a parameter (which is the widget's *name*), the value of which the user can determine
- One of the form widgets is a *submission* button
- When this button is pressed, all parameter values are sent to *URL*
- Other submission mechanisms can be used
 - e.g., pressing the "enter" key, choosing an *item*, etc.

35

An Example: Search Google

```
<form method="get"
action="http://www.google.com/search">
  <p> Type the search term:
    <input name="q" type="text"/>
    <input type="submit" value="search"/>
  </p>
</form>
```

The sent data will be something like :
q=blabla



36

Input Widgets

- Text area:

Favorite course:

```
<input type="text" name="fcourse" value="DBI"/>
```

Favorite course: DBI

- Password area:

Password:

```
<input type="password" name="pass" size="5"/>
```

Password: ****

The password is hidden while typing, but is sent as unencrypted plain text. If the "get" method is used, it is also visible in the URL box!

Default value

Checkboxes

```
<input type="checkbox" name="incoffee" value="milk" checked="checked"/>milk<br/>
```

```
<input type="checkbox" name="incoffee" value="sugar"/>sugar<br/>
```

```
<input type="checkbox" name="incoffee" value="cocoa"/>cocoa<br/>
```

Unchecked options are not sent

milk
 sugar
 cocoa

In our case, all of the checked options be assigned as values of the same variable

Sent data is something like: incoffee=sugar& incoffee=milk&...

38

Radio Buttons

Gender:

```
<input type="radio" name="gen" value="f" checked="checked"/>female
```

```
<input type="radio" name="gen" value="m"/>male
```

Only a single option CAN be checked. If there is a default option, then EXACTLY one option MUST be checked.

Gender:
 female male

Sent data is something like : gen=f

39

Menus

The number of visible items in the drop-down list

```
<select name="fruit" size="1">  
<option value="a">apples</option>  
<option value="b" selected="selected">bananas</option>  
</select>
```

bananas

bananas
apples
bananas

Only a single item can be selected

Sent data is something like : fruit=a

40

Menus

```
<select name="vegetable" size="2" multiple="multiple">  
<option value="tomato">tomato</option>  
<option value="cucumber">cucumber</option>  
<option value="lettuce">lettuce</option>  
<option value="carrot">carrot</option>  
</select>
```

tomato
cucumber

Sent data is something like : fruit=tomato&fruit=carrot

Multiple choices can be made

41

Text Areas

Write a story:


```
<textarea name="story" rows="3" cols="20">Default text...</textarea>
```

Write a story:
Default text...

42

Sending Entire Files

```
<form action="http://www.cs.huji.ac.il/~nolace"
method="post" enctype="multipart/form-data">
  <p>Upload your image:<br/>
  <input type="file" name="image"/>
  ...other form widgets...
  <input type="submit" value="send"/></p>
</form>
```

Did you know that most browsers send to the server not only the file but its COMPLETE absolute path on your system? Why is this bad? RFC1867 probably only meant for relative file names to be sent – why is it a good idea to send any part of the file name at all?

Upload your image:

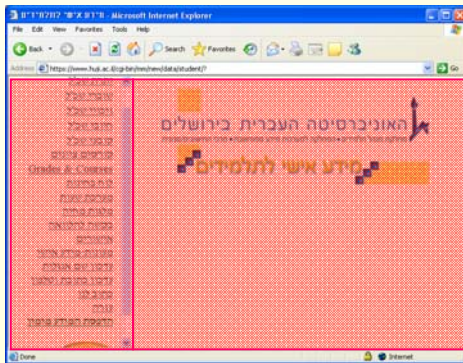
Open forms.html

43

Frames

44

A Common Usage of Frames



45

Framesets

- Instead of a “body”, the document has a “frameset” element
- Size and number of frames is determined by the attributes “cols” and “rows”
- Size can be specified relatively (e.g., 50%), by number of pixels (e.g., 70), or by “remaining size” (*)

We'll see an example later...

46

Frames

- Within frameset elements, there can be frame elements, nested frameset elements and a noframes element
- A frame can have the attributes:
 - src=“url”: the url to be displayed in the frame
 - name=“window_name”: name – used for targeting
 - scrolling=“yes/no/auto”: auto is default
- In a noframes element, put alternative content for browsers that don't support frames (how do these (old) browsers “know” they should display this content?)

47

A Frameset Example

```
<html>
  <head><title>Frames Example</title></head>
  <frameset cols="20%,*">
    <frameset rows="200,*">
      <frame name="frame1" src="left_top.html"/>
      <frame name="frame2" src="left_bottom.html"/>
    </frameset>
    <frame name="main" src="main_frame.html"/>
  </frameset>
</html>
```

48

Links in Frames

- In a link, the target attribute can specify where the new page will be opened

```
<a href="url" target="target">
```

- target="frame-name": in the specified frame
- target="_self": in the frame where the link is
- target="_top": in the whole window
- target="_blank": in a new window of the navigator
- This attribute is disallowed in the strict DTD! (But do not despair! there is a corresponding feature in JavaScript which, for some reason, does not seem to be on the way towards deprecation).

Open frameset.html 49

Inline Frames

- Inline frames are embedded in regular HTML content
- In transitional XHTML, this is done using the iframe tag
- In strict XHTML, we use the object tag

```
<object data="target" height="y" width="x" type="text/html">
```

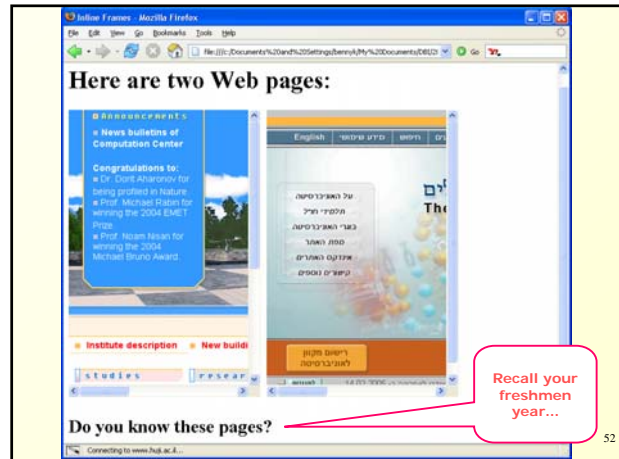
50

Inline Frame Example

```
<body>
<h1>Here are two Web pages:</h1>
<p>
<object data="http://www.cs.huji.ac.il" height="450"
width="300" type="text/html">
Your browser does not support this object!</object>
&nbsp;
<object data="http://www.huji.ac.il" height="450"
width="300" type="text/html">
Your browser does not support this object!</object>
</p>
<h2>Do you know these pages?</h2>
</body>
```

Non-breaking space

Open inline-frames.html 51



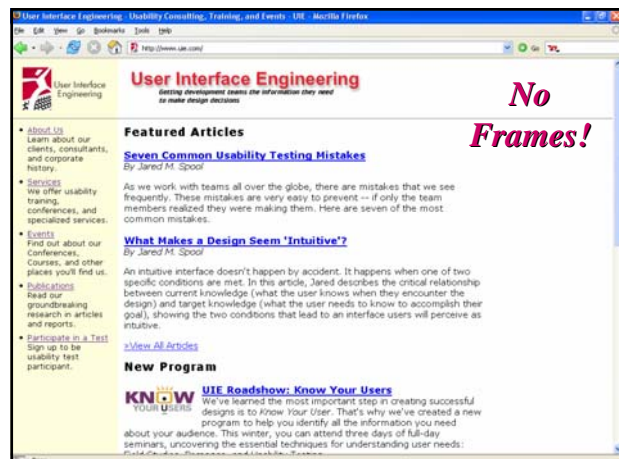
Recall your freshmen year...

52

Avoid Using Frames!

- Usage of frames poses some serious problems
 - Unidentified Web pages
 - Troubles to search engines
 - They take up a lot of screen "real estate"
 - Considered "old-fashioned" web design (like 3D tables as opposed to tables with no borders/rounded corners, etc...)
 - and more
- Hence, frames are considered wrong page design
- Instead, use CSS layout properties (next lecture)

53



No Frames!

Miscellaneous Issues

55

Entities

- There are entities that replace special symbols:

- Space: ` `;

- `<`: `<`;

- `>`: `>`;

- `&`: `&`;

```
<p>1&lt;2 &amp; 3&gt;2.</p>
```



```
1<2 & 3>2.
```

Why are these entities defined?

56

Used by browsers, search engines and other applications

The `<meta>` Tag

- The `meta` tag adds information about the page
- Examples:

```
<meta name="Author" content="Snoopy" />
```

```
<meta http-equiv="Expires" content="Tue, 20 Aug 2006 14:25:27 GMT" />
```

Reload and don't use cache if expired...

```
<meta http-equiv="refresh" content="10; url=http://www.cs.huji.ac.il/~mynewpage" />
```

```
<meta http-equiv="content-type" content="text/html; charset=iso-8859-8-1" />
```

Some other popular tags include robots exclusion tags (pros? cons?)

57

Images

- Images can be added to a page using the `img` tag.

```

```

- An image can be used as a link

A short description of the image – required! (is this good? bad?)

```
<a href="monkeys.html"></a>
```

58

Image Maps

- A picture is divided into several areas, each area linking to a different place:

Order matters! When a pixel is clicked, the data in the first area containing it is used!
See also: the `nohref` attribute

```

<map id="monky-map">
  <area shape="rect" coords="8, 8, 49, 34" href="home.html"/>
  <area shape="poly" coords="42, 87, 47, 66, 40, 87" href="information.html"/>
  <area shape="circle" coords="50, 34, 12" href="#contact"/>
</map>
```

Left, top, right, bottom

centerX, centerY, radius

X1,y1,x2,y2,...
Either clockwise or counter-clockwise.

59

Remarks / Comments

- Remarks are totally ignored during rendering.
- Use `<!-- remark -->` to insert a remark.
- Remarks can include other tags:
 - e.g., `<!-- bold but remarked -->`
- Do not use `--` in remarked text! (why?)

60

Page Validation

- You can validate the compliance of your page with the specifications using the validation service of W3C at validator.w3.org.
- Do not forget to add the appropriate **DTD** declaration and the **content-type** meta header
- If you pass validation, you will have earned the privilege of adding the precious **W3C** icon to your page.



61