Internet technologies

HTML / XHTML

Paul Rajba

pawel@cs.uni.wroc.pl
http://itcourses.eu/

Table of Contents

Introduction

HTML / XHTML is a language for presenting content

The organization of the World Wide Web Consortium (W3C): *http://www.w3c.org/*

- HTML and CSS: *http://www.w3.org/standards/webdesign/htmlcss*
- "Best practices" for HTML: *http://www.w3.org/standards/techs/htmlbp#w3c all*
- HTML Validator: *http://validator.w3.org/*

Noteworthy resources: *http://www.w3schools.com/*

XHTML application of HTML in XML

- You can use the tools for XML

Differences between HTML and XML

- Syntactical interpretation (eg. Vertical centering content)

Page 4

Browsers

Problems with browsers: created page may look like different in every browser

- Theory: created pages should look the same in every browser
- Practice: not in any browser, not all versions (the issue of the costs of establishing and maintaining the service)

The so-called. leading browsers:

- Microsoft Edge, Microsoft Internet Explorer
- Mozilla Firefox
- Google Chrome
- Opera
- Safari

Browsers support the creation of pages:

- Developer Tools in Chrome
- Tools for Web Development in Firefox
- Developer tools in IE

Page 5

Basics of syntax

To build the structure are markers

- Even, for example. <div> </ div>
- odd, for example. <br />

Tags can be parameterized attributes

- eg. <image src = "img.jpg" alt = "Image" />

A few rules regarding tags and attributes:

the name written in small letters
tags must always be closed

tags must be properly nested

Attributes are always form name = "value"

attribute values must always be enclosed in quotation marks

Introduction The Basics Heading Formatting and structure elements fonts

Page 6

Fundamentals of XHTML syntax

Attributes available for all (almost) tags [1]:

class, id, style, title

Attributes language accessible to all (almost) tag [2]:

dir = "ltr | rtl "lang

[1] Not available for tags base, head, html, meta, param, script, style and title
[2] Not available for base tags, br, frame, frameset, hr, iframe, par am and script

Introduction The Basics Heading Formatting and structure elements fonts

Page 7

The structure of the document

```
<! DOCTYPE ...>
<html>
<head>
    <title> ... </ title>
    <! - Document header ->
</ head>
<body>
    <! - Text Document ->
</ body>
</ html>
```

Introduction The Basics Heading Formatting and structure elements fonts

Page 8

document types

XHTML 1.0 Strict

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

XHTML 1.0 Transitional

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

XHTML 1.0 Frameset

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

Introduction The Basics Heading Formatting and structure elements forms

Page 9

The contents of the header

In the header we put tags:

- title - the title of the page
- meta - meta
- link - the link between documents
- base - the base URL for relative references party

Introduction The Basics Heading Formatting and structure elements forms

Page 10

The contents of the header

The link tag

Typical arguments:

href: address resource

type: the type of content

rel: type of the pointed document

selected types: Alternate, StyleSheet, Start, Next, Prev, Index, Content, Glossary, Copyright, Appendix, Help

Typical use:

```
<link rel = "stylesheet" type = "text / css" href = "style.css" />
```

More to read:

http://www.w3schools.com/TAGS/tag link.asp

Marker base, typical usage:

<base href = "http://pawel.ii.uni.wroc.pl/">

Page 11

The contents of the header

meta tag

attributes name and content - information about the document

http-equiv attributes and content - HTTP headers

Examples of the use of meta tag

```
<meta name = "Author" content = "Paul Rajba">
<meta name = "Keywords" content = "xhtml, css, php">
<meta name = "Description" content = "Home">
<meta http-equiv = "Content-Type"
    content = "text / html; charset = utf-8">
<meta http-equiv = "Content-Language" content = "en">
<meta http-equiv = "Refresh" content = "10">
<meta http-equiv = "Refresh" content = "10;
    URL = http: //www.onet.pl/ ">
```

Page 12

The contents of the header and search engines

To control the behavior of search engines have two mechanisms:

The robots.txt file in the root directory service

The corresponding entries in the document header

Page 13

The contents of the header and search engines

The robots.txt file

Keywords:

User-Agent - determines search engine
Disallow - sets out the resource

A simple example:

```
User-agent: Googlebot
User-agent: slurp
Disallow: / js /
Disallow: / webservices /

User-agent: *
Disallow: /
```

For those interested in addresses of resources:

http://www.seoconsultants.com/robots-text-file/

http://tools.seobook.com/robots-txt/generator/

Page 14

The contents of the header and search engines

Robots header in the document

Possible arguments: *index, follow, noindex, follow, index, nofollow, noindex, nofollow, all*
The most common use:

```
<meta name = "robots" content = "noindex">
<meta name = "robots" content = "nofollow">
<meta name = "robots" content = "noindex, nofollow">
```

(needless to *index* and *follow,* because it is the default the behavior of the search engine)
To read:
http://www.seoconsultants.com/meta-tags/robots/

Header *revisit-after*

It is not worth it to say, because it has no meaning
An article on this topic:
http://www.seoconsultants.com/meta-tags/revisit-after

Page 15

elements grouping

The <div> - element type *block*

The <span> - element *inline*

Page 16

formatting text

Specifying fonts

<em> <strong>, <dfn>, <code>

<tt> <i> <b> <u> <big>, <small>

Indices

<sub> <sup>

citations

<blockquote> - type *block*

<q> - *inline*

both can be defined attribute cite = "URL"

(a Firefox-ie, you can preview it by properties)

Page 17

formatting text

block elements

<h1> - <h6>

<address>

<p>

<pre>

<br />,

<hr /> (noshade attributes, size, width)

<ins> <del> (attribute cite = "URL" datetime = "datetime")

Page 18

references

Created using the tag <a>; attributes

href = "address"

target = "blank" | "Parent" | "Self" | "Top"

Anchor - a mechanism to navigate inside

document

We can write <a href="d.html#kotwica"> Anchor </a>

or <a href="#kotwica"> Anchor </a>

Anchor is then any element with *id = "anchor",* for example.

<a name="kotwica"> </a> or <div id = "anchor"> </ div>

addressing

relative, for example. <a href="../index.html"> Home </a>

absolute, for example. <a href="http://home.pl/"> Home </a>

Page 19

Pictures and maps

The image is placed with the <img> tag; attributes:

src = "URI" alt = "description" name = "name" height = "140"

width = "200" usemap = "# map"

What is a map?

Create a map using the tag:

<map id = "name">

The map includes one more area that

We define the <area>; attributes

shape = "rect | circle | poly | default"

coords = "1,2,3,4" alt = "text"

href = "URI" nohref = "nohref"

Page 20

Pictures and maps

Coords attribute depends on the shape attribute:

rect - left-x, top-y, right-x, bottom-y

circle - the center-x, center-y, radius

poly - x1, y1, x2, y2 ... xN, yN

Example:

```
<img src = "navbar.gif" alt = "navigation" usemap = "# map" />
<map id = "map">
<area href = "guide.html" alt = "Access Guide"
      shape = "rect" coords = "0,0,118,28" />
<area href = "search.html" alt = "Search"
      shape = "rect" coords = "184,0,276,28" />
<area href = "shortcut.html" alt = "Go"
      shape = "circle" coords = "184,200,60" />
<area href = "top10.html" alt = "Top Ten" shape = "poly"
      coords = "276,0,276,28,100,200,50,50,276,0" />
</ map>
```

Page 21

Letters

We have three kinds of lists

- numbered list
- list of unnumbered
- list of definitions

[Introduction](#) [The Basics](#) [Heading](#) [Formatting and structure elements](#) [forms](#)

Page 22

list of unnumbered

- We create the <ul>; attributes:
  - type = "disc" | "Circle" | "Square"
  - compact - a greater degree of packing
- List items create a tag <li>

Example:

```
<ul>
<li> Warsaw </ li>
<li> Wroclaw </ li>
<li> Krakow </ li>
</ ul>
```

[Introduction](#) [The Basics](#) [Heading](#) [Formatting and structure elements](#) [forms](#)

Page 23

numbered list

- We create the <ol>; attributes:
  - start = "number"
  - type = "1" | "A" | "A" | "I" | "AND"
  - compact - a greater degree of packing
- List items create a tag <li>

We also attribute value

Example:

```
<ol start = "5" type = "A">
<li> processor </ li>
<li> Memory </ li>
</ ol>
```

Page 24

list of definitions

We create the <dl> <dt> <dd>

Example:

```
<dl>
<dt> Aphrodite </ dt>
<dd> goddess of love and beauty </ dd>
<dt> Nemesis </ dt>
<dd> wrath of gods and punishment falling on people
      exceeding their assigned boundaries </ dd>
<dt> Poseidon </ dt>
<dd> sea god, guardian of sailors and fishermen; son of Cronus and Rhea,
      brother of Zeus and Hades, the husband of Amphitrite </ dd>
</ dl>
```

Page 25

Tabels

We create with the use of tags:

<table>, <th> <tr> <td> <caption>, <colgroup>

<col>, <thead>, <tbody> <tfoot>

Table structure is as follows:

The table consists of rows,

line consists of columns

The root tag is <table>; attributes:

summary = "text"

width = "50%" | "500"

border = "2" cellpadding = "4" cellspacing = "1"

(more at: *http://www.w3.org/TR/html401/struct/tables.html#margins)*

Page 26

A simple example

```
<table>
<tr> <th> No. index </ th> <th> Rating </ th> </ tr>
<tr> <td> 91,044 </ td> <td> 5.0 </ td>
<tr> <td> 91,057 </ td> <td> 5.0 </ td>
<tr> <td> 91088 </ td> <td> 5.0 </ td>
<tr> <td> 91,092 </ td> <td> 5.0 </ td>
</ table>
```

Introduction The Basics Heading Formatting and structure elements forms

Page 27

Combining rows and columns

Cell fusion is carried out in <th> and <td>

Serve to connect attributes

colspan = "3"

rowspan = "2"

Teacher creates the code for the following table:

Introduction The Basics Heading Formatting and structure elements forms

Page 28

Grouping rows

We have three types of groups: *head, body* and *foot*

Each group must have at least 1 row

Section *tfoot* should be before the tbody,

The <tbody> is mandatory, unless there alone

(if it can be omitted)

Table Template using groups:

<table>

```
<thead> .... </thead>
<tfoot> .... </tfoot>
<tbody> ... </ tbody>
<tbody> ... </ tbody>
...
</ table>
```

Introduction The Basics Heading Formatting and structure elements forms

Page 29

Grouping columns

Implemented by the <colgroup>; attributes

span = "4"

width = "50" (for each column in the group)

To apply the common format is useful

the <col> (important: no groups of columns)

attributes span and width

Introduction The Basics Heading Formatting and structure elements forms

Page 30

Grouping columns

Example:

```
<table>
<colgroup>
<col width = "30" />
</ colgroup>
<colgroup>
<col width = "30" />
<col width = "0 *" />
<col width = "2 *" />
</ colgroup>
<colgroup align = "center">
<col width = "1 *" />
<col width = "3 *" />
</ colgroup>
<thead>
... further part of the table ...
</ table>
```

Introduction The Basics Heading Formatting and structure elements forms

Page 31

The lines in the table

We have two attributes of the <table>

frame - defines the border; value:

void - with no side

above, below - at the top, bottom

LHS, RHS - left and right

hsides - at the top and bottom,

vsides - left and right

Box border - on each side

rules - determines how internal lines

values: none, all, groups, rows, cols

Page 32

frames

First of all, they should not be used

We talk about them, because a lot of the parties uses them

Implemented by tags: <frameset> <frame> and

<noframes>

The window (frame) represents the <frame>; attributes

name = "name" src = "URI" frameborder = "1 | 0"

marginwidth = "pixels", marginheight = "pixels"

scrolling = "yes | no | auto"

Page 33

frames

Example:

```
<! DOCTYPE HTML PUBLIC
"- // W3C // DTD HTML 4.01 Frameset // EN">
<HTML>
<HEAD> <TITLE> An example of a document with frames </ TITLE> </ HEAD>
<Frameset cols = "20%, 80%">
<Frameset rows = "100, *">
<FRAME src = "frame1.gif">
<FRAME src = "frame2.html">
</ Frameset>
<FRAME src = "frame3.html">
<NOFRAMES>
<P> This document includes:
<UL>
<LI> <IMG src = "frame1.gif" alt = "image">
<LI> <A href="frame2.html"> Document 2 </A>
<LI> <A href="frame3.html"> Document 2 </A>
</ UL>
```

</ NOFRAMES>
</ Frameset>
</ HTML>

Introduction The Basics Heading Formatting and structure elements forms

Page 34
floating frames

What is a floating frame?

To create the frame we use the <iframe>

The attributes of the <iframe>

name, src, frameborder, marginwidth, marginheight, scrolling,

height, width, align = "left | right | middle | top | bottom"

Example:

```
<iframe src = "ramka.html" width = "400" height = "500" scrolling = "auto"
            frameborder = "1" align = "right">
Your browser currently does not show frames. The contents of this frame
You can view the page at <a href="'ramka.html'"> this </a> address.
</ iframes>
```

Introduction The Basics Heading Formatting and structure elements forms

Page 35
forms

What is the use of forms?

Indicators by which we can build forms

(in brackets marker that is used to create control)

buttons (<button>, <input>)

checkbox-y (<input>)

Radio buttons (<input>)

list (<select> + <option> + <optgroup>)

text boxes (<input>, <textarea>)

select a file (<input>)

hidden field (<input>)

Introduction The Basics Heading Formatting and structure elements forms

Page 36
Creation and form properties

We create a form tag <form>; attributes that

tag:

id = "ID" name = "name"

action = "URI"

method = "POST | GET" (default GET)
enctype = "content type" (makes sense if Methods POST); value:
application / x-www-form-urlencoded (default)
multipart / form-data (for sending files)

Attributes common to most controls:

name = "name" (mandatory)
readonly = "readonly"
disabled = "disabled"
tabindex = "3"

Page 37

text fields

By using the <input>

The <input> then assumes the attributes:

type = "text" or type = "password"
size = "20" maxlength = "40"

Example:

```
<input id = "txtLoginName" type = "text" size = "20" maxlength = "50" />
```

Using the tag <textarea>

The <textarea> has attributes

rows = "10" cols = "40"

Example:

```
<textarea id = "txtDesc" rows = "20" cols = "80">
The first line of the initial text.
The second line of the initial text.
</ textarea>
```

Page 38

Letters

Created using the <select> and <option>

(optional tag <optgroup>

The attributes of the <select>

size = "3"
multiple = "multiple"

The attributes of the <option>

selected = "selected"
value = "value"
label = "text"

The attributes of the <optgroup>

label = "description"

Page 39

Letters

Examples

```
<select id = "city" name = " city ">
<option value = "0"> Wroclaw </ option>
<option value = "1"> Krakow </ option>
<option value = "2"> Poznań </ option>
</ select>
<select id = "Linux" name = " Linux ">
<option selected = "selected" value = "none"> None </ option>
<optgroup label = "SUSE">
<option value = "opensuse"> openSUSE 10.3 </ option>
<option value = "SLES"> SUSE Linux Enterprise Server 10 </ option>
</ optgroup>
<optgroup label = "Ubuntu">
<option value = "ubuntud71"> Ubuntu 7.10 Desktop </ option>
<option value = "ubuntus71"> Ubuntu Server 7.10 </ option>
</ optgroup>
</ select>
```

Page 40

Checkbox and Radio

We create them using the <input>

The tag takes time attributes:

name = "name"

checked = "checked"

value = "value"

In the case of *radio,* the group of elements, which can be selected

Only one element is common attribute name (but

id attribute values must be different)

Page 41

Checkbox and Radio

Examples:

```
<input name = "c1" type = "checkbox"
        checked = "checked" value = "0" /> Sport
<input name = "c2" type = "checkbox"
```

<input name = "c3" type = "checkbox" checked = "checked" value = "1" /> Music
checked = "checked" value = "2" /> Politics

<input type = "radio" checked = "checked" name = "sex" value = "m" /> Man
<input type = "radio" name = "sex" value = "k" /> Woman

Page 42

buttons

We can be created in several ways:

<input type = "submit" value = "string" />

Pressing will send the form data

<input type = "image" src = "przycisk.jpg" />

Pressing will send the form data

Additionally, they sent the coordinates of the click in picture

<input type = "button" value = "string" />

Pressing will not send the form data

Page 43

buttons

<input type = "reset" value = "default" />

Pressing the will to form controls values default

<button> </ button>

The attributes of the <button>

value = "value" (sent to the server)

type = "button | submit | reset"

Example:

<button name = "reset" type = "reset"> <img src = "/ icons / oops.gif"
alt = "oops" /> Reset </ button>

Page 44

others

Box file

```
<input name = "filename" type = "file" size = "30" />
```

The value of the hidden

```
<input type = "hidden" name = "viewstate"

value = "adsfasf" />
```

Border

Implemented by the <fieldset> and <legend>

Example:

```
<fieldset>
<legend> Details </ legend>
The contents of the form
</ fieldset>
```