

6

XHTML

Copyright © 2005, Binnur Kurt

Content

- ▶ Introduction to XHTML
- ▶ Why XHTML
- ▶ XHTML vs HTML
- ▶ XHTML Syntax
- ▶ XHTML DTD
- ▶ XHTML Modules
- ▶ XHTML Attributes

XHTML 6

What is XHTML?

- ▶ XHTML stands for **EX**tensible **HyperText Markup Language**
- ▶ XHTML is aimed to **replace** HTML
- ▶ XHTML is almost **identical** to HTML 4.01
- ▶ XHTML is a **stricter and cleaner** version of HTML
- ▶ XHTML is HTML defined as an **XML application**
- ▶ XHTML is a W3C Recommendation

XHTML is a W3C Recommendation

- ▶ XHTML 1.0 became a W3C Recommendation January 26, 2000.
- ▶ A W3C Recommendation means that the specification is stable, that it has been reviewed by the W3C membership, and that the specification is now a Web standard.
- ▶ W3C defines XHTML as the latest version of HTML. XHTML will gradually replace HTML.

All New Browsers Support XHTML

- ▶ XHTML is compatible with HTML 4.01.
- ▶ All new browsers have support for XHTML.

Why XHTML?

- ▶ We have reached a point where many pages on the WWW contain "bad" HTML.
- ▶ The following HTML code will work fine if you view it in a browser, even if it does not follow the HTML rules:

```
<html>
<head>
<title>This is bad HTML</title>
<body>
<h1>Bad HTML
</body>
```
- ▶ XML is a markup language where everything has to be marked up correctly, which results in "well-formed" documents.
- ▶ XML was designed to describe data and HTML was designed to display data.

Why XHTML?

XHTML 6

- ▶ Today's market consists of different browser technologies, some browsers run Internet on computers, and some browsers run Internet on mobile phones and hand helds.
- ▶ Therefore - by combining HTML and XML, and their strengths, we got a markup language that is useful now and in the future - XHTML.
- ▶ XHTML pages can be read by all XML enabled devices AND while waiting for the rest of the world to upgrade to XML supported browsers, XHTML gives you the opportunity to write "well-formed" documents now, that work in ALL browsers and that are backward browser compatible

How To Get Ready For XHTML

XHTML 6

- ▶ XHTML is the next generation of HTML, but it will of course take some time before browsers and other software products are ready for it.
- ▶ In the meantime there are some important things you can do to prepare yourself for it.
- ▶ XHTML is not very different from HTML 4.01, so bringing your code up to 4.01 standards is a very good start.
- ▶ In addition, you should start NOW to write your HTML code in lowercase letters, and NEVER make the bad habit of skipping end tags like the `</p>`

The Most Important Differences

- ▶ XHTML elements must be **properly nested**
- ▶ XHTML documents must be **well-formed**
- ▶ Tag names must be in **lowercase**
- ▶ All XHTML elements must be **closed**

Elements Must Be Properly Nested

- ▶ In HTML some elements can be improperly nested within each other like this:

`<i>This text is bold and italic</i>`

- ▶ In XHTML all elements must be properly nested within each other like this:

`<i>This text is bold and italic</i>`

Quiz

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  <li>Milk</li>
</ul>
```

Answer

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

Documents Must Be Well-formed

- ▶ All XHTML elements must be nested within the <html> root element.
- ▶ All other elements can have sub (children) elements.
- ▶ Sub elements must be in pairs and correctly nested within their parent element.
- ▶ The basic document structure is:

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

Tag Names Must Be In Lower Case

- ▶ XHTML documents are XML applications.
- ▶ XML is case-sensitive.
- ▶ Tags like
 and
 are interpreted as different tags.
- ▶ This is wrong:

```
<BODY>  
<P>This is a paragraph</P>  
</BODY>
```

- ▶ This is correct:

```
<body>  
<p>This is a paragraph</p>  
</body>
```

All XHTML Elements Must Be Closed

- ▶ Non-empty elements must have an end tag.
- ▶ This is wrong:
`<p>This is a paragraph`
`<p>This is another paragraph`
- ▶ This is correct:
`<p>This is a paragraph</p>`
`<p>This is another paragraph</p>`

Empty Elements Must Also Be Closed

- ▶ Empty elements must either have an end tag or the start tag must end with `/>`.
- ▶ This is wrong:
This is a break`
`
Here comes a horizontal rule:`<hr>`
Here's an image ``
- ▶ This is correct:
This is a break`
`
Here comes a horizontal rule:`<hr/>`
Here's an image ``

Important Compatibility Note

- ▶ To make your XHTML compatible with today's browsers, you should add an extra space before the "/" symbol like this: `
`, and this: `<hr />`.

Some More XHTML Syntax Rules

- ▶ Attribute names must be in **lower case**
- ▶ Attribute values must be **quoted**
- ▶ Attribute minimization is **forbidden**
- ▶ The id attribute **replaces** the name attribute
- ▶ The XHTML DTD defines **mandatory** elements

Attribute Names Must Be In Lower Case

- ▶ This is wrong:

```
<table WIDTH="100%">
```

- ▶ This is correct:

```
<table width="100%">
```

Attribute Values Must Be Quoted

- ▶ This is wrong:

```
<table width=100%>
```

- ▶ This is correct:

```
<table width="100%">
```

Attribute Minimization Is Forbidden

- ▶ This is wrong:
 - <input **checked**>
 - <input readonly>
 - <input disabled>
 - <option selected>
 - <frame noresize>
- ▶ This is correct:
 - <input checked="checked" />
 - <input readonly="readonly" />
 - <input disabled="disabled" />
 - <option selected="selected" />
 - <frame noresize="noresize" />

Attribute Minimization Is Forbidden

- ▶ Here is a list of the minimized attributes in HTML and how they should be written in XHTML:

HTML	XHTML
compact	compact="compact"
checked	checked="checked"
declare	declare="declare"
readonly	readonly="readonly"
disabled	disabled="disabled"
selected	selected="selected"
defer	defer="defer"
ismap	ismap="ismap"
nohref	nohref="nohref"
noshade	noshade="noshade"
nowrap	nowrap="nowrap"
multiple	multiple="multiple"
noresize	noresize="noresize"

The id Attribute Replaces The name Attribute

- ▶ HTML 4.01 defines a name attribute for the elements a, applet, frame, iframe, img, and map. In XHTML the name attribute is deprecated. Use id instead.
- ▶ This is wrong:
``
- ▶ This is correct:
``
- ▶ To interoperate with older browsers for a while, you should use both name and id, with identical attribute values, like this:
``

The Lang Attribute

- ▶ The **lang** attribute applies to almost every XHTML element. It specifies the language of the content within an element.
- ▶ If you use the **lang** attribute in an element, you must add the **xml:lang** attribute, like this:
`<div lang="no" xml:lang="no">Heia Norge!</div>`

Mandatory XHTML Elements

- ▶ All XHTML documents must have a DOCTYPE declaration.
- ▶ The html, head and body elements must be present, and the title must be present inside the head element.
- ▶ This is a minimum XHTML document template:

```
<!DOCTYPE Doctype goes here>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title goes here</title>
</head>

<body>
Body text goes here
</body>

</html>
```

Mandatory XHTML Elements

- ▶ The **DOCTYPE** declaration is not a part of the XHTML document itself. It is not an XHTML element, and it should not have a closing tag.
- ▶ The **xmlns** attribute inside the <html> tag is required in XHTML. However, the validator on w3.org does not complain when this attribute is missing in an XHTML document.
- ▶ This is because "**xmlns**=http://www.w3.org/1999/xhtml" is a fixed value and will be added to the <html> tag even if you do not include it.

The <!DOCTYPE> Is Mandatory

- ▶ An XHTML document consists of three main parts:
 - the DOCTYPE
 - the Head
 - the Body
- ▶ The basic document structure is:

```
<!DOCTYPE ...>
<html>
<head>
<title>... </title>
</head>
<body> ... </body>
</html>
```
- ▶ The DOCTYPE declaration should always be the first line in an XHTML document.

An XHTML Example

- ▶ This is a simple (minimal) XHTML document:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>simple document</title>
</head>
<body>
<p>a simple paragraph</p>
</body>
</html>
```

An XHTML Example

- ▶ The DOCTYPE declaration defines the document type:

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

- ▶ The rest of the document looks like HTML:

```
<html>  
<head>  
<title>simple document</title>  
</head>  
<body>  
<p>a simple paragraph</p>  
</body>  
</html>
```

The 3 Document Type Definitions

- ▶ DTD specifies the syntax of a web page in SGML.
- ▶ DTD is used by SGML applications, such as HTML, to specify rules that apply to the markup of documents of a particular type, including a set of element and entity declarations.
- ▶ XHTML is specified in an SGML document type definition or 'DTD'.
- ▶ An XHTML DTD describes in precise, computer-readable language, the allowed syntax and grammar of XHTML markup.

The 3 Document Type Definitions

- ▶ XHTML 1.0 specifies three XML document types that correspond to three DTDs:
 - Strict
 - Transitional
 - Frameset

XHTML 1.0 Strict

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

- ▶ Use this when you want really clean markup, free of presentational clutter.
- ▶ Use this together with Cascading Style Sheets.

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 6

► Use this

- when you need to take advantage of HTML's presentational features
- when you want to support browsers that don't understand Cascading Style Sheets.

XHTML 1.0 Frameset

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

XHTML 6

- Use this when you want to use HTML Frames to partition the browser window into two or more frames.

Why XHTML Modularization?

- ▶ XHTML is a simple but large language, containing most of the functionality a web developer will need.
- ▶ For some purposes XHTML is too large and complex, and for other purposes it is much too simple.
- ▶ By splitting XHTML into modules, the W3C (World Wide web Consortium) has created small and well-defined sets of XHTML elements that can be used separately for simple devices as well as combined with other XML standards into larger and more complex applications.

Why XHTML Modularization?

- ▶ With modular XHTML, product and application designers can:
 - Choose the elements to be supported by a device using standard XHTML building blocks.
 - Add extensions to XHTML, using XML, without breaking the XHTML standard.
 - Simplify XHTML for devices like hand held computers, mobile phones, TV, and home appliances.
 - Extend XHTML for complex applications by adding new XML functionality (like MathML, SVG, Voice and Multimedia).
 - Define XHTML profiles like XHTML Basic (a subset of XHTML for mobile devices).

W3C has split the definition of XHTML into 28 modules:

XHTML Modules

Module name	Description
Applet Module	Defines the deprecated* applet element.
Base Module	Defines the base element.
Basic Forms Module	Defines the basic forms elements.
Basic Tables Module	Defines the basic table elements.
Bi-directional Text Module	Defines the bdo element.
Client Image Map Module	Defines browser side image map elements.
Edit Module	Defines the editing elements del and ins.
Forms Module	Defines all elements used in forms.
Frames Module	Defines the frameset elements.
Hypertext Module	Defines the a element.
Iframe Module	Defines the iframe element.
Image Module	Defines the img element.

Image Module	Defines the img element.
Intrinsic Events Module	Defines event attributes like onblur and onchange.
Legacy Module	Defines deprecated* elements and attributes.
Link Module	Defines the link element.
List Module	Defines the list elements ol, li, ul, dd, dt, and dl.
Metainformation Module	Defines the meta element.
Name Identification Module	Defines the deprecated* name attribute.
Object Module	Defines the object and param elements.
Presentation Module	Defines presentation elements like b and i.
Scripting Module	Defines the script and noscript elements.
Server Image Map Module	Defines server side image map elements.
Structure Module	Defines the elements html, head, title and body.
Style Attribute Module	Defines the style attribute.
Style Sheet Module	Defines the style element.
Tables Module	Defines the elements used in tables.
Target Module	Defines the target attribute.
Text Module	Defines text container elements like p and h1.

XHTML Standard Attributes

- ▶ XHTML tags can have attributes.
- ▶ The special attributes for each tag are listed under each tag description.
- ▶ The attributes listed here are the core and language attributes that are standard for all tags (with a few exceptions).

Core Attributes

- ▶ Not valid in base, head, html, meta, param, script, style, and title elements.

Attribute	Value	Description
class	<i>class_rule</i> or <i>style_rule</i>	The class of the element
id	<i>id_name</i>	A unique id for the element
style	<i>style_definition</i>	An inline style definition
title	<i>tooltip_text</i>	A text to display in a tool tip

Language Attributes

- ▶ Not valid in base, br, frame, frameset, hr, iframe, param, and script elements.

XHTML 6

Attribute	Value	Description
dir	ltr rtl	Sets the text direction
lang	<i>language_code</i>	Sets the language code

Keyboard Attributes

Attribute	Value	Description
accesskey	<i>character</i>	Sets a keyboard shortcut to access an element
tabindex	<i>number</i>	Sets the tab order of an element

XHTML 6

XHTML Event Attributes

- ▶ New to HTML 4.0 was the ability to let HTML events trigger actions in the browser, like starting a JavaScript when a user clicks on an HTML element. Below is a list of attributes that can be inserted into HTML tags to define event actions.
- ▶ We will learn more about programming with these events in studying JavaScript and DHTML.

Window Events

- ▶ Only valid in body and frameset elements

Attribute	Value	Description
onload	<i>script</i>	Script to be run when a document loads
onunload	<i>script</i>	Script to be run when a document unloads

Form Element Events

► Only valid in form elements.

Attribute	Value	Description
onchange	<i>script</i>	Script to be run when the element changes
onsubmit	<i>script</i>	Script to be run when the form is submitted
onreset	<i>script</i>	Script to be run when the form is reset
onselect	<i>script</i>	Script to be run when the element is selected
onblur	<i>script</i>	Script to be run when the element loses focus
onfocus	<i>script</i>	Script to be run when the element gets focus

Keyboard Events

► Not valid in base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title elements.

Attribute	Value	Description
onkeydown	<i>script</i>	What to do when key is pressed
onkeypress	<i>script</i>	What to do when key is pressed and released
onkeyup	<i>script</i>	What to do when key is released

Mouse Events

- ▶ Not valid in base, bdo, br, frame, frameset, head, html, iframe, meta, param, script, style, and title elements.

Attribute	Value	Description
onclick	<i>script</i>	What to do on a mouse click
ondblclick	<i>script</i>	What to do on a mouse doubleclick
onmousedown	<i>script</i>	What to do when mouse button is pressed
onmousemove	<i>script</i>	What to do when mouse pointer moves
onmouseover	<i>script</i>	What to do when mouse pointer moves over an element
onmouseout	<i>script</i>	What to do when mouse pointer moves out of an element
onmouseup	<i>script</i>	What to do when mouse button is released

Quiz



Question # 1

- ▶ What does XHTML stand for?
 - a. EXtensible HyperText Markup Language
 - b. EXtreme HyperText Markup Language
 - c. EXtensible HyperText Marking Language
 - d. EXtra Hyperlinks and Text Markup Language

XHTML 6

Question # 2

- ▶ XHTML is a Web standard
 - a. True
 - b. False

XHTML 6

Question # 3

- ▶ XML and HTML will be replaced by XHTML
- a. False
- b. True

XHTML 6

Question # 4

- ▶ HTML will be replaced by XHTML
- a. False
- b. True

XHTML 6

Question # 5

► What is the correct XHTML for a paragraph?

- a. `</p><p>`
- b. `<P></P>`
- c. `<P></p>`
- d. `<p></p>`

XHTML 6

Question # 6

► What is a correct XHTML tag for a line break?

- a. `
`
- b. `
`
- c. `<break/>`

XHTML 6

Question # 7

- ▶ What is the correct XHTML for an attribute and its value?
 - a. WIDTH="80"
 - b. width=80
 - c. WIDTH=80
 - d. width="80"

XHTML 6

Question # 8

- ▶ All elements in XHTML must be closed
 - a. True
 - b. False

XHTML 6

Question # 9

- Is this correct XHTML?

```
<ul>
<li>Coffee</li>
<li>Tea
<ul>
<li>Black tea</li>
<li>Green tea</li>
</ul>
<li>Milk</li>
</ul>
```

- No
- Yes

XHTML 6

Question # 10

- The DOCTYPE declaration has no closing tag

- True
- False

XHTML 6

Question # 11

- ▶ Which elements are mandatory in an XHTML document?
 - a. doctype, html, head, body, and title
 - b. doctype, html and body
 - c. doctype, html, head, and body

XHTML 6

Question # 12

- ▶ XHTML documents must be "well-formed"
 - a. True
 - b. False

XHTML 6

Question # 13

- ▶ What XHTML code is "well-formed"?
- a. `<p>A <i>short</i> paragraph</p>`
- b. `<p>A <i>short</i> paragraph`
- c. `<p>A <i>short</i> paragraph</p>`

XHTML 6

Question # 14

- ▶ Which of the following is the right use of the lang attribute?
- a. `<div language="en">Hello World!</div>`
- b. `<div xml:language="en">Hello World!</div>`
- c. `<div lang="en" xml:lang="en">Hello World!</div>`

XHTML 6

Question # 15

- ▶ Which attribute replaces the name attribute?
 - a. None
 - b. The src attribute
 - c. The id attribute

XHTML 6

Question # 16

- ▶ Is attribute minimization allowed in XHTML?
 - a. Yes
 - b. No

XHTML 6

Question # 17

- ▶ What are the different DTDs in XHTML?
 - a. Strict, Transitional, Frameset
 - b. Strict, Transitional, Loose
 - c. Strict, Transitional, Loose, Frameset

XHTML 6

Question # 18

- ▶ Do all XHTML documents require a doctype?
 - a. Yes
 - b. No

XHTML 6

Question # 19

- ▶ What is the most common XHTML DTD?
 - a. Loose
 - b. Normal
 - c. Frameset
 - d. Transitional

XHTML 6

Question # 20

- ▶ All XHTML tags and attributes must be in lower case
 - a. False
 - b. True

XHTML 6