

CHAPTER 6.1 – XHTML



INTRODUCTION

- XHTML stands for **EX**tensible **Hyper Text Markup Language**.
- It is the next step in the evolution of the internet.
- The XHTML 1.0 is the first document type in the XHTML family.
- This is a cleaner and stricter version of HTML 4.01.
- If you already know HTML then you need to give little attention to learn this latest version of HTML.



ADVANTAGES

- XHTML documents are XML conforming as they are readily viewed, edited, and validated with standard XML tools.
- XHTML documents can be written to operate better than they did before in existing browsers as well as in new browsers.
- XHTML documents can utilize applications such as scripts and applets that rely upon either the HTML Document Object Model or the XML Document Object Model.



WHY XHTML?

- XHTML has stricter syntax rules in comparison to HTML.
- Since XHTML is an official standard of the World Wide Web Consortium (W3C), your website becomes more compatible with many browsers and it is rendered more accurately.
- XHTML combines strength of HTML and XML. Also, XHTML pages can be rendered by all XML enabled browsers.
- XHTML defines quality standard for web-pages and if you follow that, then your web pages are counted as quality web pages and are certified by W3C with their quality stamp.



XHTML - SYNTAX

- XHTML syntax is very similar to HTML syntax and almost all the valid HTML elements are valid in XHTML as well.
- Some important points to consider at the time of writing XHTML pages or converting HTML to XHTML document.
 - You need to write a DOCTYPE declaration at the start of the XHTML document.
 - You must write all XHTML tags and attributes in lower case only.
 - You need to close all XHTML tags properly.
 - You must quote all the attribute values.
 - Attribute minimization is forbidden.
 - The **id** attribute replaces the **name** attribute.
 - The **language** attribute of the **script** tag is deprecated.
 - You need to nest all the tags properly.



DOCTYPE DECLARATION

- All XHTML documents must have a DOCTYPE declaration at the start.
- There are three types of DOCTYPE declarations: Strict, Transitional and Frameset
- Here is an example of using DOCTYPE:
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`



CASE SENSITIVITY

- XHTML is case sensitive markup language.
- All the XHTML tags and attributes need to be written in lower case only.
- <!-- This is invalid in XHTML -->
XHTML Tutorial
<!-- Correct XHTML way of writing this is as follows →
XHTML Tutorial



CLOSING TAGS

- Each and every XHTML tag should have an equivalent closing tag, even empty elements should also have closing tags. E.g.
- **Invalid way**
- `<!-- This is invalid in XHTML -->`
`<p>This paragraph is not written according to XHTML syntax.`
`<!-- This is also invalid in XHTML -->`
``
- **Valid way**
- `<p>This paragraph is not written according to XHTML syntax.</p>`
`<!-- This is also valid now -->`
``



ATTRIBUTE QUOTES

- All the values of XHTML attributes must be quoted.
- Otherwise, your XHTML document is assumed as an invalid document. Here is the example showing syntax:
- `<!-- This is invalid in XHTML -->`
``
`<!-- Correct XHTML way of writing this is as follows -->`
``



ATTRIBUTE MINIMIZATION

- XHTML does not allow attribute minimization.
- It means you need to explicitly state the attribute and its value.
- The following example shows the difference:
- `<!-- This is invalid in XHTML →
<option selected>
<!-- Correct XHTML way of writing this is as follows -->
<option selected="selected">`



THE ID ATTRIBUTE

- The **id** attribute replaces the **name** attribute.
- Instead of using `name="name"`, XHTML prefers to use `id="id"`. The following example shows this difference:
- `<!-- This is invalid in XHTML -->`
``
`<!-- Correct XHTML way of writing this is as follows -->`
``



THE LANGUAGE ATTRIBUTE

- The **language** attribute of the script tag is deprecated. The following example shows this difference:
- <!-- This is invalid in XHTML -->
- <script language="JavaScript" type="text/JavaScript">
document.write("Hello XHTML!");
</script>
<!-- Correct XHTML way of writing this is as follows -->
<script type="text/JavaScript">
document.write("Hello XHTML!");
</script>



NESTED TAGS

- All the XHTML tags must be nested properly otherwise your document will be assumed as an incorrect XHTML document. The following example shows the syntax:
- `<!-- This is invalid in XHTML -->`
- `<i> This text is bold and italic`
- `</i> <!-- Correct XHTML way of writing this is as follows -->`
`<i> This text is bold and italic</i>`



ELEMENT PROHIBITIONS

- The following elements are not allowed to have any other element inside them.
- This prohibition applies to all depths of nesting, i.e. it includes all the descendant elements.

Element	Prohibition
<a>	Must not contain other <a> elements.
<pre>	Must not contain the , <object>, <big>, <small>, <sub>, or <sup> elements.
<button>	Must not contain the <input>, <select>, <textarea>, <label>, <button>, <form>, <fieldset>, <iframe> or <isindex> elements.
<label>	Must not contain other <label> elements.
<form>	Must not contain other <form> elements.



A MINIMAL XHTML DOCUMENT

- Following example shows you a minimum content of an XHTML 1.0 document.
- ```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1" xml:lang="en"
lang="en">
<head> <title>Every document must have a title</title> </head>
<body> ...your content goes here... </body> </html>
```



# XHTML DOCUMENTS MUST BE WELL-FORMED

- Essentially this means, all the elements must have closing tags and you must nest them properly.
- **CORRECT** : nested elements
- `<p>Here is an emphasized <em>paragraph</em>.</p>`
- **INCORRECT** : overlapping elements
- `<p>Here is an emphasized <em>paragraph.</p></em>`





# XHTML DOCUMENTS MUST BE WELL-FORMED

- **Elements and attributes must be in lower case**
- XHTML documents must use lower case for all HTML elements and attribute names.
- This difference is necessary because XHTML document is assumed to be an XML document and XML is case-sensitive.
- For example, `<li>` and `<LI>` are different tags.
  
- **End tags are required for all elements**
- In HTML, certain elements are permitted to omit the end tag. But XML does not allow end tags to be omitted.
- **CORRECT** : terminated elements
- `<p>Here is a paragraph.</p><p>here is another paragraph.</p><br/><hr/>`
- **INCORRECT** : unterminated elements
- `<p>Here is a paragraph.<p>here is another paragraph. <br><hr>`



# XHTML DOCUMENTS MUST BE WELL-FORMED

- **Elements and attributes must be in lower case**
- XHTML documents must use lower case for all HTML elements and attribute names.
- This difference is necessary because XHTML document is assumed to be an XML document and XML is case-sensitive.
- For example, `<li>` and `<LI>` are different tags.
  
- **End tags are required for all elements**
- In HTML, certain elements are permitted to omit the end tag. But XML does not allow end tags to be omitted.
- **CORRECT** : terminated elements
- `<p>Here is a paragraph.</p><p>here is another paragraph.</p><br/><hr/>`
- **INCORRECT** : unterminated elements
- `<p>Here is a paragraph.<p>here is another paragraph. <br><hr>`



# XHTML DOCUMENTS MUST BE WELL-FORMED

- **Attribute values must always be quoted**
- All attribute values including numeric values, must be quoted.
- **CORRECT** : quoted attribute values
- `<td rowspan="3">`
- **INCORRECT** : unquoted attribute values
- `<td rowspan=3>`
  
- **Attribute Minimization**
- XML does not support attribute minimization.
- Attribute-value pairs must be written in full.
- Attribute names such as compact and checked cannot occur in elements without their value being specified.
- **CORRECT** : non minimized attributes
- `<dl compact="compact">`
- **INCORRECT** : minimized attributes
- `<dl compact>`



# XHTML DOCUMENTS MUST BE WELL-FORMED

- **Attribute values must always be quoted**
- All attribute values including numeric values, must be quoted.
- **CORRECT** : quoted attribute values
- `<td rowspan="3">`
- **INCORRECT** : unquoted attribute values
- `<td rowspan=3>`
  
- **Attribute Minimization**
- XML does not support attribute minimization.
- Attribute-value pairs must be written in full.
- Attribute names such as compact and checked cannot occur in elements without their value being specified.
- **CORRECT** : non minimized attributes
- `<dl compact="compact">`
- **INCORRECT** : minimized attributes
- `<dl compact>`



# XHTML DOCUMENTS MUST BE WELL-FORMED

- **The <html> Element is a must**
- All XHTML elements must be nested within the <html> root element.
- All other elements can have sub elements which must be in pairs and correctly nested within their parent element.
- The basic document structure is:
  - <!DOCTYPE html....>
  - <html>
  - <head> ... </head>
  - <body> ... </body>
  - </html>



# XHTML - ATTRIBUTES

- There are few XHTML/HTML attributes which are standard and associated to all the XHTML/HTML tags.
- These attributes are listed here with brief description:
- Core attributes
- Not valid in base, head, html, meta, param, script, style, and title elements.

Attribute	Value	Description
class	class_rule or style_rule	The class of the element
Id	id_name	A unique id for the element
style	style_definition	An inline style definition
Title	tooltip_text	A text to display in a mouse tip



# XHTML - EVENT

- We can write our event handlers in JavaScript or VBScript and can specify these event handlers as a value of event tag attribute.
- The XHTML 1.0 has a similar set of events which is available in HTML 4.01 specification.
- The <body> and <frameset> level events
- There are only two attributes which can be used to trigger any JavaScript or VBScript code, when any event occurs at document level.

Attribute	Value	Description
onload	Script	Script runs when a XHTML document loads
onunload	Script	Script runs when a XHTML document unloads



# XHTML - EVENT

- The <form> level events

Attribute	Value	Description
onchange	Script	Script executes when the element changes
onsubmit	Script	Script executes when the form is submitted
onreset	Script	Script executes when the form is reset
onselect	Script	Script executes when the element is selected
onblur	Script	Script executes when the element loses focus
onfocus	Script	Script runs when the element gets focus





# XHTML - EVENT

- Keyboard Events

Attribute	Value	Description
onkeydown	Script	Script executes on key press
onkeypress	Script	Script executes on key press and release
onkeyup	Script	Script executes key release



# XHTML - EVENT

- Other Events

Attribute	Value	Description
onclick	Script	Script executes on a mouse click
ondblclick	Script	Script executes on a mouse double-click
onmousedown	Script	Script executes when mouse button is pressed
onmousemove	Script	Script executes when mouse pointer moves
onmouseout	Script	Script executes when mouse pointer moves out of an element
onmouseover	Script	Script executes when mouse pointer moves over an element
onmouseup	Script	Script executes when mouse button is released

