

HTML ----> XHTML



A Brief Introduction to XHTML

Extensible HyperText Markup Language (XHTML) is a reformulation of HTML 4.0 to make it XML based. This tutorial deals with the changes to be made to convert HTML documents to valid XHTML. The article is prepared with a view to help and guide you through the conversion process.

The W3C, which is the organisation that co-ordinates standardisation of Web protocols, has defined three types of XHTML documents. This is based on the XML Document Type Definition (DTD) that is used by the document. The XHTML DTDs are:

1. Strict: Used when the XHTML document is devoid of all formatting tags like and Cascading Style Sheets (CSS) are used for controlling all presentation aspects.
2. Transitional: This XHTML DTD allows use of presentation tags in the document. This is a safer mode since most of our pages contain many presentation elements.
3. Frameset: Used for XHTML documents that describes frames.

The document type declaration

for transitional XHTML documents is:

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

This should be the next line after the processing instruction.

The declarations for the other XHTML DTDs are:

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

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

XML requires that there must be one and only one root element for a document. Hence, in XHTML, all tags should be enclosed within the <html> tag, ie., <html> should be the root element for the document.

The starting tag <html> should be modified to include namespace information. The modification is as:

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

Attribute xmlns is the XML namespace with which we associate the XHTML document. The value of the attribute lang is the code for the language of the document as specified in RFC1766.



tags in XHTML

- All XHTML tag elements should be in lower case. That means `<HTML>` and `<Body>` are wrong. They should be rewritten as `<html>` and `<body>` respectively.
- All XHTML tags should have their end tags. In HTML it is common for paragraphs to have only the starting `<p>` tag. In XHTML this is not allowed. You need to end a paragraph with the `</p>` tag.
Example: `<p>Hello` is wrong; it should be written as `<p>Hello</p>`.
- Empty XHTML tags should be ended with `</>` instead of `>`. The commonly used empty tags in XHTML are:
 - `<meta />`: for meta information (contained in the head section)
 - `<base />`: used to specify the base URI and also the target frame for hyperlinks (contained in the head section)
 - `<basefont />`: used to specify a base font for the document. Note that attribute 'size' is mandatory
 - `<param />`: parameters for applets and objects.
 - `<link />`: to specify external stylesheets and other references.
 - ``: to include images. Attributes 'src' for the source URI and 'alt' for alternate text are mandatory.
 - `
`: used for forced line break.
 - `<hr />`: for horizontal rules.
 - `<area />`: used inside image maps. Attribute 'alt' is mandatory.
 - `<input />`: used inside forms for input form elements like buttons, textboxes, textareas, checkboxes and radio buttons.

Example: `<br clear="all">` is wrong;
it should be rewritten as `<br clear="all" />`.
`` is wrong;
it should be ``

- Proper nesting of tags is compulsory in XHTML.
Example: `<i>This is bold italics<i>` is wrong.
It should be rewritten as `<i>This is bold italics</i>`.

Rules for XHTML Attributes

- All XHTML attribute names should be in lower case.
Example: `Width="100"` and `WIDTH="100"` are wrong;
only `width="100"` is correct.
Similarly `onMouseOut="javascript:myFunction();"` is wrong;
it should be rewritten as `onmouseout="javascript:myFunction();"`.
- All attribute-value pairs should be quoted.
Example: `width=100` is wrong; it should be `width="100"` or `width='100'`.
- The name attribute is deprecated and will be removed in a future version of XHTML and the id attribute will take its place. So, for HTML tags that need the name attribute, an id attribute should also be specified with the same value as that for name.
Example: `<frame name="myFrame" >` becomes
`<frame name="myFrame" id="myFrame" >`



- HTML supports certain attributes which have no values. Examples are `noshade` which appears in the `<hr noshade>` tag.

XHTML does not allow such empty or compact attributes. The compact attributes generally found in HTML are `compact`, `nowrap`, `ismap`, `declare`, `noshade`, `checked`, `disabled`, `readonly`, `multiple`, `selected`, `noresize` and `defer`.

They should always have a value. In XHTML this is done by giving the attribute name itself as the value!

Example: `noshade` becomes `noshade="noshade"`
`checked` becomes `checked="checked"`

- All `&` (ampersand) characters in the source code have to be replaced with `&`, which is the equivalent character entity code. This change should be done in all attribute values and URIs.

Example: `Bee&Nee` will result in an error if you try to validate it; It should be written as `Bee&Nee`.

`Go` is wrong;

it should be coded as `Go`.

XHTML Tables

- For `<table>` tag, attribute `height` is not supported in XHTML 1.0. Only the `width` is supported. The `<td>` tag does support the `height` attribute.
- The `<table>`, `<tr>` and the `<td>` tag does not support the attribute `background` which is used to specify a background image for the table or the cell. Background images will have to be specified either using the `style` attribute or using external stylesheet. The attribute `bgcolor` for background color is however supported by these tags.

XHTML Images

- The `alt` attribute is mandatory. This value of this attribute will be the text that has to be shown in older browsers, text-only browsers (like lynx), and in place of the image when it is not available.

Note that `` is an empty tag.

Example: ``

XHTML and Javascript

- The `type` attribute is mandatory for all `<script>` tags. This value of `type` is `text/javascript` for Javascript.

- The use of external scripts is recommended.

Example:

```
<script type="text/javascript" language="javascript" src="functions.js">
</script>
```



- If you are using internal scripts, enclose it within the starting tag `<![CDATA[` and the ending tag `]]>`. This will mark it as unparsed character data. Otherwise characters like `&` and `<` will be treated as start of character entities (like ` `;) and tags (like ``) respectively.

Example for XHTML Javascript:

```
<script type="text/javascript" language="Javascript">
<!--
<![CDATA[
document.write('Hello World!');
]]>
//-->
</script>
```

XHTML and Stylesheets

- The `type` attribute is mandatory for `<style>` tag. The value of `type` is `text/css` for stylesheets.

- The use of external stylesheets is recommended.

Example: `<link rel="stylesheet" type="text/css" href="screen.css" />`

- Enclose internal style definitions within the starting tag `<![CDATA[` and the ending tag `]]>` to mark it as unparsed character data.

Example:

```
<style type="text/stylesheet">
<![CDATA[
.MyClass { color: #000000; }
]]>
</style>
```

Otherwise the `&` and `<` characters will be treated as start of character entities (like ` `;) and tags (like ``) respectively.

Element Prohibitions in XHTML

The W3C recommendation also prohibits certain XHTML elements from containing some elements. Those are given below:

- `<a>` cannot contain other `<a>` elements.
- `<pre>` cannot contain the ``, `<object>`, `<big>`, `<small>`, `<sub>`, or `<sup>` elements.
- `<button>` cannot contain the `<input>`, `<select>`, `<textarea>`, `<label>`, `<button>`, `<form>`, `<fieldset>`, `<iframe>`, or `<isindex>` elements.
- `<label>` cannot contain other `<label>` elements.
- `<form>` cannot contain other `<form>` elements.

XHTML Resources on the Web



- **The W3C Pages on XML**
The W3C are the people who work for the formulation and standardisation of Web technologies including XHTML. They are the best place to go.
- **Download the XHTML 1.0 Transitional DTD**
The DTD (Document Type Definition) is used to define an XML application. XHTML is also a XML application and all the rules can be found in this well documented DTD.
- **HTML-Tidy**
Written by Dave Reggett, this tool can will accept any (bloated or rotten) HTML and make it to adhere to standards. It can also be used to accelerate conversion of HTML to XML or XHTML.
- **Chami's HTML-Kit**
An excellent HTML editor (not visual, but supports previewing) which supports XHTML. It supports the HTML-Tidy as a plugin. Recommended.
- **The W3C Online Validator for XHTML**
XHTML documents can be validated online with this W3C Service. Recommended.
- **RFC1766**
This RFC defines the two-letter tags for the Identification of Languages.