



From XHTML To XHTML5

Table Of Contents

- [Introduction](#)
- [Background](#)
- [HTML 5 Improvements](#)

Introduction

This tutorial is for those who know native XHTML (eXtensible HyperText Markup Language) and are interested in upgrading to XHTML5.

It begins with a little background of why XHTML5 came about, then we have the steps for upgrading the old XHTML to the existing subset of XHTML5 that is already supported in native XHTML web browsers.

Background

Since HTML (HyperText Markup Language), written in SGML (Standard Generalised Markup Language), was upgraded to XHTML (eXtensible HyperText Markup Language), written in XML (eXtensible Markup Language), and then upgraded to a modular language; various realities have dawned after field testing XHTML.

As HTML, XHTML, was written in theory and it had to be modified from the specification in order to implement it into programs like web browsers. Although XML provides strict rules on how to handle datatypes and errors which proves a more rock solid insurance of well-formed documents, authors were continuously frustrated with one small mistake and the entire webpage or website throws XML errors at everyone. Plus certain breed of web browsers had no (and seems never will) support native XHTML.

Web Authors began rejecting native XHTML for their websites and either returned to XHTML 1.0 following its HTML-Compatible Guidelines or creating their own custom XHTML using the modules and including support for HTML-Compatibility as the HTML parser handles errors in a more graceful manor.

The World Wide Web Consortium (W3C) were taking the XHTML specification down a road to XHTML Version 2 where developers of web browsers and other programs found it impossible to implement with a massive learning curve for both browser developer and web author.

HTML5 (featuring an XHTML 5 version) is a proposal provided by the WHATWG (Web Hypertext Application Technology Working Group). This working group was founded by Mozilla Foundation, Opera Software and Apple in 2004 after a W3C workshop on XHTML. WHATWG didn't like where W3C was taking XHTML and HTML. Some concepts were an improvement but they do not address recent web features such as Web Applications and non-document-based webpages.

Since, the W3C has recognized HTML 5 and the WHATWG is part of the new W3C HTML Working Group, together they are developing HTML 5. A W3C XHTML2 Working Group continues the development of Version 1.1 of Modular XHTML and the development of XHTML2.

HTML 5 is currently a Draft but basic parts of the specification is modelled on the existing implementation of HTML, XHTML and browser extensions so parts of XHTML 5 is already supported in browsers.

HTML 5 Improvements

HTML (HyperText Markup Language) version 5 is an abstract language, a single vocabulary for webpages that both old HTML 4.01 and XHTML 1.1 and under can upgrade to. For HTML 4.01 and under and HTML-Compatible XHTML 1.0, they can upgrade to the custom HTML of HTML 5. For native XHTML 1.0 and higher, they can upgrade to XHTML 5.

XHTML 5 is based on XML as older XHTML are but additionally is modelled on the physical implementation of XHTML in applications such as web browsers.

An XHTML 5 document is XML 1.0 or 1.1 based not because of the Doctype or the syntax used but because of the MIME Media Types of [text/xml](#), [application/xml](#) or [application/xhtml+xml](#).

All native XHTML web browsers such as Mozilla Firefox, SeaMonkey, Netscape 6 to 9, Konqueror 3.x, Opera and Apple Safari support the basics of XHTML5.

Some simple notes to upgrade from native XHTML to XHTML 5:

1. Block level elements are now categorised as Flow Content and Inline level elements are categorised as Phrasing Content;
2. Elements like ``, `<iframe></iframe>` and `<object></object>` are categorised as Embedding Content and are also Phrasing Content;
3. The `id` attribute still replaces the `name` attribute for most elements in regards to identifying parts of the document for scripting and styling purposes and link targets. Only `<iframe></iframe>` and `<object></object>` will retain the `name` attribute for compatibility with some web browsers but use the `id` attribute as well for other and future web browsers. `<meta/>` and form controls continue to use `name` as usual. Plus `xml:id` is available but don't use both on the same element as you can only have one ID type attribute per element;
4. The `language` attribute in `<script language="JavaScript"> </script>` elements has been dropped completely and replaced permanently with the global standard `type` attribute as `<script type="text/javascript"> </script>`;
5. `document.write()` and `document.writeln()` still doesn't work at all and is old hat. Instead you use `innerHTML` or other features from HTML Document Object Model Level 5 to manipulate the existing markup;
6. `<noscript></noscript>` has been dropped because it actually has no effect in native XML environments;

From XHTML To XHTML5 - Legend Scrolls

7. There is no XHTML 5 Document Type Definition as it is described in terms of the DOM, so named entities beyond XML's five, cannot be guaranteed. Unless you write your own – then you can only use it in environments that actually support DTDs (Most web browsers do not);
8. It is best to move from using character sets like ISO-8859-1 to the most supported global standard and Unicode supporting character set UTF-8;
9. Like version 1.1 of Modular XHTML, the `<script></script>`, `<style></style>` and `<pre></pre>` elements do not need `xml:space="preserve"` attribute to preserve leading and trailing spaces and any multiple spaces within words at the XML parser level;
10. Like Modular XHTML you only use the `xml:lang` attribute for stating the spoken language;
11. You can still use `xml:base` attributes instead of the `<base/>` Empty Element.

As for the namespace, it has the same URI:

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

Some deprecated elements have changed semantics (become more meaningful) such as `<i></i>` and `` are allowed as 'offset from the normal text without any other meaning' such as screen readers would use a different voice or pitch etc. But these two elements are only to be used as an absolute last resort: there are far more meaningful elements at your disposal.

A few existing elements have also improved semantics such as `<small></small>` is for small print, `` is for importance. Multiple nested `` elements will convey stronger emphasis and multiple nested `` will convey stronger importance.

For each initial `<dfn></dfn>` term on the webpage, its expanded value (usually in a `title` attribute) should now be within the surrounding text as should the expanded value of each initial `<abbr></abbr>` abbreviation.

To accompany the `style` attribute (to state style properties only) for local styling on the element, (X)HTML5 adds 'scoped stylesheets' as the first children of elements that can have Flow Content within them such as the `<div></div>` element.

Like a normal `<style></style>` element but has a `scoped="scoped"` attribute. You also are free to use selectors, media At-Rules and other typical code in CSS stylesheets (the default styling language in HTML5 and XHTML5 is CSS (text/css)). Scoped stylesheets provide this local styling for the parent element (such as the `<div></div>`) and its other children. But current web browsers ignore the `scoped="scoped"` attribute but to be valid you need to keep it in.

From XHTML To XHTML5 - Legend Scrolls

For specifying the character set you use the XML Declaration's `encoding` attribute as usual. Any meta instances such as `<meta http-equiv="content-type" content="text/html; charset=UTF-8"/>` and text/html form of HTML 5's `<meta charset="UTF-8">` are ignored and forbidden in XHTML5.

For those still using XHTML 1.0, `ids` are replacing the `name` attribute identifying an element for scripting purposes. Even part of the Document Object Model (DOM) there is a method called `getElementById()` with the parameter as the value of an `id` attribute. This method obtains the element that has that `id` attribute value.

Ids can be used to attach styles too using the hash or sharp character (`#`) as:

```
div#navigation {
    width: 98%;
    background-color: aqua;
}
```

Attaching the width and background colour styles to a div element with an `id="navigation"` on it.

Also it replaces `` elements as fragment identifiers and results of URIs such as `mypage.xhtml#fourthParagraph`. As it is an attribute on an element, any element with an `id` can be the target of such a URI.

`Id` is a unique identifier and so the value must be unique throughout the document. `id` attributes can only start with an underscore (`_`), a colon (`:`) or a letter and then any number of underscores, colons, dashes (`-`), letters or numbers.

As mentioned above, XHTML 5 has no specific Doctype. In environments that do fully support XML DTDs, you may write your own. The XHTML 5 compatible Doctype has these forms:

```
<!DOCTYPE html PUBLIC "-//YourCompany//Your XHTML 5 DTD Name and
Version//EN" "http://me.example.com/path/to/myxhtml5.dtd">
```

or

```
<!DOCTYPE html "http://me.example.com/path/to/myxhtml5.dtd">
```

But most web browsers do not validate webpages. To validate an XHTML 5 document you need to write your own DTD and use an XML DTD enabled validator service or use a HTML 5 validator such as the one at <http://html5.validator.nu/>.

Several elements and attributes have been dropped such as `cellspacing`, `cellpadding` and `summary` attributes on `<table></table>` are replaced by CSS margins, padding and XHTML's `<caption></caption>` element.

From XHTML To XHTML5 - Legend Scrolls

Also `frameborder`, `border`, `hspace`, `vspace`, `leftmargin`, `rightmargin`, `topmargin`, `bottommargin`, `valign` and `align` are replaced by CSS's `border`, `margins`, `padding`, `vertical-align` and `text-align` properties.

Only ``, `<embed/>`, `<object></object>`, `<canvas></canvas>`, `<video></video>` may have the `width` and `height` attributes; others can use the CSS `width` and `height` properties.

A few non-common attributes from `<object></object>` and `language` from `<script></script>` have also been dropped.

``, `<noframes></noframes>`, `<frameset></frameset>`, `<frame/>` and `<noscript></noscript>` are all dropped.

`<embed></embed>` is only supported as `<embed/>` for compatibility with ancient plugins. You should use `<object></object>` these days.

This is just the subset of XHTML 5 that is currently supported by all native XHTML web browsers. But new markup is being introduced and tested addressing sectioning elements (`<section></section>`, `<article></article>`, `<aside></aside>`, `<nav></nav>`, `<header></header>` and `<footer></footer>`); improved forms with Web Forms 2 such as `<input/>` types including `email`, `url`, `datetime`, `time`, `week`, `number`, `range`. Plus markup repetition templates, combo boxes; `<canvas></canvas>` for dynamically drawing bitmap images and animating them; `<audio></audio>` and `<video></video>` for improved embedded multimedia including a multimedia API (Application Programming Interface) for dynamically creating your own controls. And address' Web Applications including client-side data storage.

Brand new and upcoming versions of web browsers are starting to add support for some of these 'beyond the subset' features.

End.

From XHTML To XHTML5 - Legend Scrolls

This quick tutorial is available online at:

<http://www.legendscrolls.co.uk/articles/xhtml12xhtml5/>

Author: Peter Davison from [Legend Scrolls](#).

Copyright ©2008 Legend Scrolls and Peter Davison.

All rights reserved.