



FREE eBook

LEARNING xhtml

Free unaffiliated eBook created from
Stack Overflow contributors.

#xhtml

Table of Contents

About	1
Chapter 1: Getting started with xhtml	2
Remarks.....	2
Versions.....	2
Examples.....	2
Installation or Setup.....	2
Full XHTML and JavaScript Example.....	4
XHTML5 and Boolean Attributes.....	5
Adding XML to an XHTML Application and Retrieving XML from an XHTML Application.....	5
Credits	7

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [xhtml](#)

It is an unofficial and free xhtml ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official xhtml.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with xhtml

Remarks

This section provides an overview of what xhtml is, and why a developer might want to use it.

It should also mention any large subjects within xhtml, and link out to the related topics. Since the Documentation for xhtml is new, you may need to create initial versions of those related topics.

Versions

Version	Release Date
XHTML 1.0	2000-01-26
XHTML 1.1	2001-05-31
XHTML 2.0	2006-08-31
XHTML 5	2014-10-28

Examples

Installation or Setup

XHTML Relation to HTML

XHTML is simply the serialized version of HTML. XHTML was originally intended to clean up HTML markup to better improve standards support. Unfortunately the W3C's work on XHTML 2.0 was not intuitive and detracted from where the industry as a whole was headed with HTML5. XHTML5 is not a standard though you can use XHTML5 (XML serialized HTML5) with minor modifications. Examples are provided below to ease the adoption of XHTML5 and clarify ambiguities in the HTML5 standard where XML is not taken in to consideration.

XHTML Benefits

Because XHTML uses a browser's XML parser it is significantly (though not absolutely) less prone to common programming errors. If a developer loads an XHTML application with malformed XML most browser rendering engines will render *up to* the malformed XML while the Gecko rendering engine (used in Firefox) will show a yellow page. In all cases an error message will be displayed with the line and column numbers of where the *first* encountered XML parsing error was encountered. Many developers have made simple mistakes such as missing a quote from an HTML attribute leaving them spending days trying to determine why one or two browsers are not rendering the HTML page as expected, using XHTML can dramatically decrease development time over HTML. XHTML may have a positive reinforcement on learning to code valid HTML for

people learning web development as it does *not* allow the XHTML application to render in full with malformed code and by giving an explicit error message that new developers can search for online for working solutions. Since XHTML is a subset of XML is therefore has a very high compatibility with software that works with XML in a very wide array of industrial, commercial and residential uses. Lastly due to it's strict requirements working XHTML code is automatically compatible with HTML and XML (subjective to absence multiple same `id` attribute values) while HTML is *not* inherently compatible with other bodies of HTML and XML as it is not immediately apparent that the HTML code may be malformed.

XHTML Drawbacks

Due to the much stricter XML parser rules XHTML is not as easy, initially, for less seasoned developers. XHTML is not taken in to consideration often by various groups, contains ambiguity in regards to poorly written parts of the HTML5 standard and there is no explicit validator that declares support for XHTML5. XHTML is less compatible with "lazier" parts of JavaScript such as `innerHTML` which does not properly serialize new parts of the DOM however this is more of a benefit to dedicated developers learning XHTML as it requires stricter and more interchangeable code.

Using XHTML

XHTML is the combination of HTML and the use of the XML parser which is a much stricter version of the HTML parser; all modern browsers have HTML parsers for HTML and XML parsers for XML (and subsequently XHTML). XHTML does not require an installation of software (beyond using any modern browser) however an (X)HTML application is only XHTML when it is correctly *served* to the browser with a header sent by the server declaring the `application/xhtml+xml` mime type; you can verify if your XHTML application was actually served as this by viewing the media type/mime/type in the network requests panel of the web developer tools in your browser. A page served as `text/html` is *not* XHTML and will instead be parsed by browser's HTML parser.

Loaded from a Server

When loading an XHTML parser from the server a header *must* be set, using a `meta` element serves no useful purpose. In example in combination with PHP:

```
<?php
if (isset($_SERVER['HTTP_ACCEPT']) &&
    strstr($_SERVER['HTTP_ACCEPT'],'application/xhtml+xml'))
{
    // Client header isset` and explicitly declares XHTML parser support.
    header('Content-Type: application/xhtml+xml; charset=UTF-8');
    echo '<?xml version="1.0" encoding="UTF-8"?>'. "\n";
    echo '<!DOCTYPE html>'. "\n";
    echo '<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">'. "\n";
}
else
{
    //Some browsers do not declare support, IE9 shamelessly uses `*/` in example.
    echo '<!DOCTYPE html>'. "\n";
    echo '<html>'. "\n";
}
?>
```

Loaded from a File

If you are doing testing without loading a page from a server path (e.g. localhost, 127.0.0.1, 192.168.xxx.yyy, etc) then the only way to make a browser load an XHTML application and use the XML parser is to give the file an `.xhtml` extension; `example.xhtml`.

XHTML / XML Parser Browser Support

Browsers with XHTML support via XML parsers include Internet Explorer 9+ (XML parser support in older versions do in a very convoluted fashion support `application/xml`), Mozilla Suite 0.8+ (all Firefox versions), Opera 7+, and early versions of KHTML (Konqueror and therefore all versions of Safari and by further extension Chromium/Chrome).

Full XHTML and JavaScript Example

The following is a full example of using XHTML with JavaScript served by PHP as a single-file.

```
<?php
if (isset($_SERVER['HTTP_ACCEPT']) &&
    strstr($_SERVER['HTTP_ACCEPT'],'application/xhtml+xml'))
{
    // Client header isset` and explicitly declares XHTML parser support.
    header('Content-Type: application/xhtml+xml; charset=UTF-8');
    echo '<?xml version="1.0" encoding="UTF-8"?>'. "\n";
    echo '<!DOCTYPE html>'. "\n";
    echo '<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">'. "\n";
}
else
{
    //Some browsers do not declare support, IE9 shamelessly uses `*/` in example.
    echo '<!DOCTYPE html>'. "\n";
    echo '<html>'. "\n";
}
?>
<head>
<style type="text/css">
* {border: 0; margin: 0; padding: 0;}
</style>
<script type="application/javascript">
//
window.onload = function(event)
{
    alert('This JavaScript alert will load once the page has loaded.');</pre></div><div data-bbox="57 961 255 980" data-label="Page-Footer"><hr/><p><a href="https://riptutorial.com/">https://riptutorial.com/</a></p></div><div data-bbox="933 962 953 978" data-label="Page-Footer"><p>4</p></div>
```

XHTML5 and Boolean Attributes

HTML5 defines some HTML attributes as boolean; a boolean can only be `true` or `false`. The specification simply states that the *presence* of a boolean attribute implies that the attribute is set to `true`. In example using a `disabled` attribute in the following example disables the button input element:

```
<input disabled type="button" value="HTML Button">
```

XML, and therefore XHTML by extension, *must* have a valid attribute and value. Because HTML5 is not written in a fashion to clarify such things (ambiguity in past standards has led to differing browser implementations) HTML5 attributes when served in an XHTML application should always use a `true` value, at least until a future specification removes the unnecessary ambiguity.

```
<input disabled="true" type="button" value="XHTML Button" />
```

Adding XML to an XHTML Application and Retrieving XML from an XHTML Application

Using XHTML you should avoid methods such as `document.write` and `innerHTML` as they treat XML as text and do not properly serialize code; an `id` attribute in HTML is essentially dumped in to the DOM and the `id` attribute is *not* serialized which means when trying to reference to it with something such as `document.getElementById('example')` the browser will not "see" the id. The following examples get "get" XHTML code *from* and "set" XHTML code *to* the XHTML application.

Adding XHTML *to* and Retrieving XML *from* the DOM

```
<?php
if (isset($_SERVER['HTTP_ACCEPT']) &&
    strstr($_SERVER['HTTP_ACCEPT'],'application/xhtml+xml'))
{
    // Client header isset` and explicitly declares XHTML parser support.
    header('Content-Type: application/xhtml+xml; charset=UTF-8');
    echo '<?xml version="1.0" encoding="UTF-8"?>'. "\n";
    echo '<!DOCTYPE html>'. "\n";
    echo '<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">'. "\n";
}
else
{
    //Some browsers do not declare support, IE9 shamelessly uses `*/` in example.
    echo '<!DOCTYPE html>'. "\n";
    echo '<html>'. "\n";
}
?>
<head>
<style type="text/css">
* {border: 0; margin: 0; padding: 0;}
</style>
<script type="application/javascript">
//
function xml_get(target)
{</pre></div><div data-bbox="57 961 254 980" data-label="Page-Footer"><p><a href="https://riptutorial.com/">https://riptutorial.com/</a></p></div><div data-bbox="932 962 953 979" data-label="Page-Footer"><p>5</p></div>
```

```

return new XMLSerializer().serializeToString(target)
}

function xml_set(xml,position,target)
{
  if (typeof target=='string' && document.getElementById(target)) {target =
document.getElementById(target);}

  if (!target) {alert('Error: target element was not found in the DOM.')}
  else if (position=='after')
  {
    if (target.nextSibling && target.nextSibling!='[object Text]')
{target.insertBefore(xml.getElementsByTagName('*')[0],target.nextSibling);}
    else {target.parentNode.appendChild(xml.getElementsByTagName('*')[0]);}
  }
  else if (position=='before')
{target.parentNode.insertBefore(document.importNode(xml.getElementsByTagName('*')[0],true),target);}

  else if (position=='inside')
{target.appendChild(document.importNode(xml.getElementsByTagName('*')[0],true));}
  else if (position=='replace')
{target.parentNode.replaceChild(document.importNode(xml.getElementsByTagName('*')[0],true),target);}

  else {alert('Error: unknown position to import data to: '+position+'.')}
}

window.onload = function(event)
{
  var xml_after = new DOMParser().parseFromString('<p xmlns="http://www.w3.org/1999/xhtml">XML
string for <em>after</em> the h1[0] element!</p>','application/xml');
  var xml_before = new DOMParser().parseFromString('<p xmlns="http://www.w3.org/1999/xhtml">XML
string for <em>before</em> the h1[0] element!</p>','application/xml');
  var xml_inside = new DOMParser().parseFromString('<p xmlns="http://www.w3.org/1999/xhtml">XML
string for <em>inside</em> inside the element with the id
<code>example</code>!</p>','application/xml');
  var xml_replace = new DOMParser().parseFromString('<p
xmlns="http://www.w3.org/1999/xhtml">XML string for <em>replace</em>
example!</p>','application/xml');
  xml_set(xml_after,'after',document.getElementsByTagName('h1')[0]);
  xml_set(xml_before,'before',document.getElementsByTagName('h1')[0]);
  xml_set(xml_inside,'inside','example');
  xml_set(xml_replace,'replace','example_replace');

  alert(xml_get(document));
}
//]]>
</script>
</head>

<body>

<h1><span>XHTML and JavaScript Simple Demonstration</span></h1>
<div id="example"></div>
<div id="example_replace"></div>

</body>
</html>

```

Read Getting started with xhtml online: <https://riptutorial.com/xhtml/topic/9045/getting-started-with-xhtml>

Credits

S. No	Chapters	Contributors
1	Getting started with xhtml	Community , John , Ray