**EBook Gratis**

# APRENDIZAJE xhtml

Free unaffiliated eBook created from
**Stack Overflow contributors.**

#xhtml

# Tabla de contenido

# Acerca de

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.



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

# Capítulo 1: Empezando con xhtml

## Observaciones

Esta sección proporciona una descripción general de qué es xhtml y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande dentro de xhtml, y vincular a los temas relacionados. Dado que la Documentación para xhtml es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

## Versiones

| Versión | Fecha de lanzamiento |
|---|---|
| XHTML 1.0 | 2000-01-26 |
| XHTML 1.1 | 2001-05-31 |
| XHTML 2.0 | 2006-08-31 |
| XHTML 5 | 2014-10-28 |

## Examples

### Instalación o configuración

**Relación XHTML a HTML**

XHTML es simplemente la versión serializada de HTML. XHTML fue pensado originalmente para limpiar el marcado HTML para mejorar el soporte de estándares. Desafortunadamente, el trabajo del W3C en XHTML 2.0 no fue intuitivo y le restó valor a la industria en su conjunto con HTML5. XHTML5 no es un estándar, aunque puede utilizar XHTML5 (HTML5 serializado XML) con pequeñas modificaciones. A continuación se proporcionan ejemplos para facilitar la adopción de XHTML5 y aclarar las ambigüedades en el estándar HTML5, donde XML no se toma en consideración.

**Beneficios de XHTML**

Debido a que XHTML utiliza el analizador XML de un navegador, es significativamente (aunque no absolutamente) menos propenso a errores de programación comunes. Si un desarrollador carga una aplicación XHTML con XML con formato incorrecto, la mayoría de los motores de representación del navegador se mostrarán *en* formato XML incorrecto, mientras que el motor de representación de Gecko (utilizado en Firefox) mostrará una página amarilla. En todos los casos, se mostrará un mensaje de error con los números de línea y columna donde se encontró el *primer*

error de análisis XML encontrado. Muchos desarrolladores han cometido errores simples, como faltar una cita de un atributo HTML, lo que hace que pasen días tratando de determinar por qué uno o dos navegadores no están mostrando la página HTML como se esperaba, el uso de XHTML puede disminuir considerablemente el tiempo de desarrollo sobre HTML. XHTML puede tener un refuerzo positivo en el aprendizaje de código HTML válido para las personas que aprenden el desarrollo web, ya que *no* permite que la aplicación XHTML se muestre completa con el código incorrecto y da un mensaje de error explícito que los nuevos desarrolladores pueden buscar en línea para encontrar soluciones que funcionen. Dado que XHTML es un subconjunto de XML, por lo tanto, tiene una compatibilidad muy alta con el software que funciona con XML en una amplia gama de usos industriales, comerciales y residenciales. Por último, debido a sus estrictos requisitos, el código XHTML es automáticamente compatible con HTML y XML (subjetivo a la ausencia de los mismos valores de atributo de `id` ), mientras que HTML *no* es inherentemente compatible con otros cuerpos de HTML y XML, ya que no es evidente de forma inmediata que el código HTML pueda estar mal formado

**Inconvenientes XHTML**

Debido a las reglas de analizador de XML mucho más estrictas, XHTML no es tan fácil, inicialmente, para desarrolladores menos experimentados. XHTML no es tomado en consideración a menudo por varios grupos, contiene ambigüedad con respecto a partes mal escritas del estándar HTML5 y no hay un validador explícito que declare el soporte para XHTML5. XHTML es menos compatible con partes "más perezosas" de JavaScript, como `innerHTML` que no serializa correctamente nuevas partes del DOM; sin embargo, esto es más beneficioso para los desarrolladores dedicados que aprenden XHTML, ya que requiere un código más estricto e intercambiable.

**Usando XHTML**

XHTML es la combinación de HTML y el uso del analizador XML, que es una versión mucho más estricta del analizador HTML; todos los navegadores modernos tienen analizadores HTML para analizadores HTML y XML para XML (y posteriormente XHTML). XHTML no requiere una instalación de software (más allá de usar cualquier navegador moderno) sin embargo, una (X) HTML sólo es XHTML cuando se *sirve* correctamente en el navegador con un encabezado enviado por el servidor declarando la `application/xhtml+xml` de tipos MIME; puede verificar si su aplicación XHTML se sirvió de esta manera al ver el tipo de medio / mime / tipo en el panel de solicitudes de red de las herramientas de desarrollo web en su navegador. Una página servida como `text/html` *no* es XHTML y en su lugar será analizada por el analizador HTML del navegador.

**Cargado desde un servidor**

Cuando se carga un analizador XHTML desde el servidor, se *debe* configurar un encabezado, el uso de un elemento `meta` no sirve para ningún propósito útil. En ejemplo en combinación con PHP:

```
<?php
if (isset($_SERVER['HTTP_ACCEPT']) &&
stristr($_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";
}
?>
```

---

**Cargado desde un archivo**

Si está realizando pruebas sin cargar una página desde una ruta del servidor (por ejemplo, localhost, 127.0.0.1, 192.168.xxx.yyy, etc.), la única forma de hacer que un navegador cargue una aplicación XHTML y use el analizador XML es presentar una extensión `.xhtml` ; `example.xhtml` .

---

**Compatibilidad con el explorador XHTML / XML Parser**

Los navegadores con soporte XHTML a través de analizadores XML incluyen Internet Explorer 9+ (el soporte de analizador XML en versiones anteriores lo hace en una `application/xml` soporte de moda muy compleja `application/xml` ), Mozilla Suite 0.8+ (todas las versiones de Firefox), Opera 7+ y versiones anteriores de KHTML (Konqueror y, por tanto, todas las versiones de Safari y, por otra extensión, Chromium / Chrome).

## Ejemplo completo de XHTML y JavaScript

El siguiente es un ejemplo completo del uso de XHTML con JavaScript servido por PHP como un solo archivo.

```
<?php
if (isset($_SERVER['HTTP_ACCEPT']) &&
stristr($_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">
```

```
//<![CDATA[
window.onload = function(event)
{
 alert('This JavaScript alert will load once the page has loaded.');
}
//]]>
</script>
</head>

<body>

<h1><span>XHTML and JavaScript Simple Demonstration</span></h1>

</body>
</html>
```

## XHTML5 y atributos booleanos

HTML5 define algunos atributos HTML como booleanos; Un booleano solo puede ser `true` o `false` . La especificación simplemente establece que la *presencia* de un atributo booleano implica que el atributo se establece en verdadero. En el ejemplo, el uso de un atributo `disabled` en el siguiente ejemplo deshabilita el elemento de entrada de botón:

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

XML, y por lo tanto XHTML por extensión, *debe* tener un atributo y valor válidos. Debido a que HTML5 no está escrito de manera que aclare tales cosas (la ambigüedad en estándares anteriores ha llevado a implementaciones de navegador diferentes) los atributos HTML5 cuando se sirven en una aplicación XHTML siempre deben usar un valor `true` , al menos hasta que una especificación futura elimine la ambigüedad innecesaria.

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

## Adición de XML a una aplicación XHTML y recuperación de XML de una aplicación XHTML

Usando XHTML debe evitar métodos como `document.write` e `innerHTML` ya que tratan el XML como texto y no serializan el código correctamente; un atributo de `id` en HTML es esencialmente volcado en el DOM y el atributo de `id` *no* está serializado, lo que significa que cuando se intenta hacer referencia a él con algo como `document.getElementById('example')` el navegador no "verá" la identificación. Los siguientes ejemplos obtienen el código XHTML "get" *de* y el código XHTML "set" *en* la aplicación XHTML.

### Añadiendo *a* XHTML y recuperación de XML *desde* el DOM

```
<?php
if (isset($_SERVER['HTTP_ACCEPT']) &&
stristr($_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">
//<![CDATA[
function xml_get(target)
{
 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>
```

Lea Empezando con xhtml en línea: https://riptutorial.com/es/xhtml/topic/9045/empezando-con-xhtml

# Creditos

| S. No | Capítulos | Contributors |
|---|---|---|
| 1 | Empezando con xhtml | Community, John, Ray |