

# Introducing XHTML: HTML's Future

---

In Chapter 6, you learned about XML. In this chapter, you'll see how XHTML, or XML with a specific Document Type Definition (DTD), is serving as the jumping point for future revisions of HTML.

## What Is XHTML?

XHTML is an XML application. The DTDs, or Document Type Definitions, used with XML are actually close copies of HTML 4.01, close enough that XHTML can be viewed as simply a stricter enforcement of HTML. It doesn't offer any exciting new tags. Instead, it has been created with two specific goals — extensibility and portability.

Cross-  
Reference

Chapter 6 provides a detailed explanation of XML.

## Extensibility

Previously, a new version of HTML was introduced every so often, with new elements and attributes that were probably already in use. The major browsers often leapfrogged the HTML specification with new goodies they thought were beneficial, and then Microsoft and Netscape would go to the W3C and campaign for their newly defined tags to be included in the next new HTML specification. This method of defining new tags led to complications we discuss throughout this book. But what it really led to were more headaches for HTML authors who just wanted their pages to look good in any browser.



### In This Chapter

What is XHTML?

Advantages over HTML — extensibility and portability

Differences between HTML and XHTML



XML elements must be well formed, which will be discussed shortly. To change or introduce new HTML elements, the language itself must be changed. But in XML, all that is required is for the new elements to be internally consistent and well formed before they can be added to a DTD. This makes adding new elements much simpler.

## Portability

We are moving rapidly away from a time when the Web was only accessed via computer. New electronic devices, such as cellular phones and personal digital assistants (PDAs), are Web-enabled. In the past, sheer horsepower from the computer could be used to fill in sloppy or ill-formed HTML code, often invisibly to the user.

New devices don't have this kind of spare processor power. Instead of interpreting the bad code, these new Web devices may simply sit there and wait to be told what to do. By more rigidly enforcing the HTML DTD, this type of code error is better prevented, and is thus less likely to be the source of PDA panic.

## Differences Between HTML and XHTML

Current browsers correctly interpret omissions in HTML instructions. In contrast, XHTML code must be well formed by adhering closely to established standards. To give an example, if there is a missing `</table>` end tag, Internet Explorer will fill it in. In XHTML, this would be an example of ill-formed code and it would probably result in a failure to show the table.

There are a number of considerations in writing well-formed XHTML code. Keep in mind that some of these provisions are a stricter enforcement of things that should have been done in HTML 4, but you could get away with errors and the browser would correct many of them. This is no longer true in XHTML.

## Required tags

In XHTML, the head and body tags must be included. Since including them is part of well written HTML code anyway, this shouldn't be a problem.

## Tags must be properly nested

Tags must be closed in the proper order. In HTML, if you were using the italic tags in a paragraph, it didn't matter which tag you closed first, and which tag was closed second. In XHTML, tags need to be closed in reverse order — the last tag opened is the first closed. The following code is an XHTML example:

```
<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```

        "DTD/xhtml11-transitional.dtd" >
<html  xmlns   = "http://www.w3.org/1999/xhtml">
<head>
    <title>Quick Example</title>
</head>
<body>
<h1>   Quick Example
</h1>
<a    href    = "http://validator.w3.org/check/referer">
<img  src     = "http://validator.w3.org/images/vxhtml10"
      height  = "31"
      width   = "88"
      border  = "0"
      hspace  = "16"
      align   = "left"
      alt     = "Valid XHTML 1.0!"
    /></a>
<p>    Note that the layout (with tabs and alignment) is
      purely for readability - XHTML doesn't require it.
</p>
</body>
</html>

```

**This example is code that wouldn't pass XHTML requirements:**

```

<!DOCTYPE html PUBLIC
    "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "DTD/xhtml11-transitional.dtd" >
<html  xmlns   = "http://www.w3.org/1999/xhtml">
<head>
    <title>Quick Example</title>
</head>
<body>
<h1>   Quick Example
</h1>
<a    href    = "http://validator.w3.org/check/referer">
<img  src     = "http://validator.w3.org/images/vxhtml10"
      height  = "31"
      width   = "88"
      border  = "0"
      hspace  = "16"
      align   = "left"
      alt     = "Valid XHTML 1.0!"
    /></a>
<p>    Note that the layout (with tabs and alignment) is
      purely for readability - XHTML doesn't require it.
</body>
</p>
</html>

```

Note

Notice the last closing paragraph tag comes after the closing body tag in this example; this one occurrence of swapped tags makes this invalid XHTML.

## Lowercase tag and attribute names

In HTML, tags are often written in uppercase to make the code easier to follow. This is no longer acceptable. While attribute *values* are still case-insensitive, elements and attributes must be written in lowercase.

## Empty elements are not allowed

Empty elements are those that have no closing tag, like `<hr>`, `<br>`, or `<img src>`. These elements must be closed using the XML syntax for empty tags, which requires a forward slash (/) immediately before the closing bracket, also known as the greater than sign. For example, `<hr>` becomes `<hr />`. An image reference changes from `` to ``. The following code is an example of closing empty elements.

```
<img      src      = "http://validator.w3.org/images/vxhtml10"
         height   = "31"
         width    = "88"
         border   = "0"
         hspace   = "16"
         align    = "left"
         alt      = "Valid XHTML 1.0!"
       /></a>
```

Note

Notice the closing bracket for `img src` has a forward slash (/).

## Nonempty elements have to be closed

Many elements in use don't currently require end tags. For example, in HTML 4 the `<li>` tag is used to list elements, and since the next element is marked by another `<li>` tag; this tag effectively closes the previous list element. But not any more. XHTML requires that nonempty elements be closed. This affects tags such as `<li>`, `<p>`, `<body>`, and many table-related tags.

```
<p>      Note that the layout (with tabs and alignment) is
         purely for readability - XHTML doesn't require it.
</p>
</body>
</html>
```

Note

Notice the presence of the closing paragraph tag, `</p>`.

## Attribute values must be quoted

All attribute values, whether alphabetic or numeric, must now be enclosed in quotation marks. For example, when setting attributes such as height and width for an image element, the numbers must be surrounded by quotation marks.

```
<img src = "http://validator.w3.org/images/vxhtml10"
      height = "31"
      width = "88"
      border = "0"
      hspace = "16"
      align = "left"
      alt = "Valid XHTML 1.0!"
/>
```

Note

Notice the double quotation marks before and after the attribute values, for example, "31".

## Attribute values must be expanded

Attributes are considered minimized when there is only one possible value. XML doesn't allow this type of minimization, so attributes must be expanded. For example, `<ul compact>` becomes `<ul compact="compact">`.

## From Here

Cross-Reference

Proceed to Chapter 8 to learn about HTML development software.

Not ready to take the XHTML plunge? Jump to Chapter 24 to find out about testing and validating your HTML.

## Summary

HTML 4.01 is the final version of HTML. In order to allay portability concerns and address new extensions to the language for specific needs, an XML-based framework, XHTML, has been developed. While it requires a bit more attention to writing well-formed code, these habits will pay off in easier adoption of new elements and better support across many new platforms. XHTML 1.0 takes over where HTML 4.01 left off.



