

What is XHTML?

XHTML is the next generation of markup language to be read by Web browsers, following HTML 4.01 as the standard. Extensible Hypertext Markup Language uses the structure of XML (Extensible Markup Language – used for tagging data) and the tags and elements of HTML (Hypertext Markup Language) to add structure, flexibility, and expandability to Web development.

Why do we need flexibility and expandability? HTML has a fixed set of elements making it limited in its ability to evolve and change. With the introduction of devices other than computer browsers that will be displaying Web data, like wireless handhelds, XHTML provides consistent accessibility across platforms.

XHTML is similar to HTML, it just applies some stricter standards that must be followed:

- Tag elements and attribute names must be in lowercase
- Elements require end tags
- Attribute values must always be in quotes
- No attribute minimization (all attributes must have a value)
- Document Type Definition, XML version, and XML namespace are required at top of document -

```
<?xml version="1.0" encoding="iso-8859-1"?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">
```

This lets the browser know that it is an XHTML document and how to interpret

- You will find some tags deprecated or made obsolete by XHTML. For example, the font tag which is used in HTML to control the font appearance of text, is no longer necessary when using CSS. The theory behind XHTML is to use XHTML code for formatting your document and CSS for styling.

Cascading Stylesheets (CSS) allow you to create styles based on tag element names to perform page design. By separating the structuring of the document (XHTML) from the styling of the document, you gain greater control of the appearance and maintenance of your Web site. More on CSS later.

Block versus inline elements

A block element creates a new block of content with a clear beginning and ending. For example, the `<p>` element not only identifies a paragraph, but it also provides spacing above or below.

An inline element does not create a new block of content. It can be placed within a block element. For example, the tag `` can be placed within a paragraph. No spacing before or after is accomplished by the tag.

FYI

HTML5 and CSS3 are the next revisions of each. The combo will allow for lots of new functionality and flexibility in coding. HTML5 will also specify application programming interfaces (APIs) There will be some new elements and attributes (`<nav>` `<footer>` `<audio>` `<video>`) and some elements will be deprecated. Not all browsers support these standards yet, but will in the near future. For more info visit <http://net.tutsplus.com/tutorials/html-css-techniques/html-5-and-css-3-the-techniques-youll-soon-be-using/> and http://en.wikipedia.org/wiki/HTML_5