

XHTML VS HTML

http://www.tutorialspoint.com/xhtml/xhtml_vs_html.htm

Copyright © tutorialspoint.com

Due to the fact that XHTML is an XML application, certain practices that were perfectly legal in SGML-based HTML 4 must be changed. You already have seen XHTML syntax in previous chapter, so differences between XHTML and HTML are very obvious. Following is the comparison between XHTML and HTML.

XHTML Documents Must be Well-Formed

Well-formedness is a new concept introduced by XML. Essentially, this means all the elements must have closing tags and you must nest them properly.

CORRECT: Nested Elements

```
<p>Here is an emphasized <em>paragraph</em>.</p>
```

INCORRECT: Overlapping Elements

```
<p>Here is an emphasized <em>paragraph.</p></em>
```

Elements and Attributes Must be in Lower Case

XHTML documents must use lower case for all HTML elements and attribute names. This difference is necessary because XHTML document is assumed to be an XML document and XML is case-sensitive. For example, and are different tags.

End Tags are Required for all Elements

In HTML, certain elements are permitted to omit the end tag. But XML does not allow end tags to be omitted.

CORRECT: Terminated Elements

```
<p>Here is a paragraph.</p><p>here is another paragraph.</p>  
<br><hr/>
```

INCORRECT: Unterminated Elements

```
<p>Here is a paragraph.<p>here is another paragraph.  
<br><hr>
```

Attribute Values Must Always be Quoted

All attribute values including numeric values, must be quoted.

CORRECT: Quoted Attribute Values

```
<td rowspan="3">
```

INCORRECT: Unquoted Attribute Values

```
<td rowspan=3>
```

Attribute Minimization

XML does not support attribute minimization. Attribute-value pairs must be written in full. Attribute names such as compact and checked cannot occur in elements without their value being specified.

CORRECT: Non Minimized Attributes

```
<dl compact="compact">
```

INCORRECT: Minimized Attributes

```
<dl compact>
```

Whitespace Handling in Attribute Values

When a browser processes attributes, it does the following –

- Strips leading and trailing whitespace.
- Maps sequences of one or more white space characters *including linebreaks* to a single inter-word space.

Script and Style Elements

In XHTML, the script and style elements should not have “<” and “&” characters directly, if they exist; then they are treated as the start of markup. The entities such as “<” and “&” are recognized as entity references by the XML processor for displaying “<” and “&” characters respectively.

Wrapping the content of the script or style element within a CDATA marked section avoids the expansion of these entities.

```
<script type="text/JavaScript">
  <![CDATA[
    ... unescaped VB or Java Script here...
  ]]>
</script>
```

An alternative is to use external script and style documents.

The Elements with *id* and *name* Attributes

XHTML recommends the replacement of *name* attribute with *id* attribute. Note that in XHTML 1.0, the *name* attribute of these elements is formally deprecated, and it will be removed in a subsequent versions of XHTML.

Attributes with Pre-defined Value Sets

HTML and XHTML both have some attributes that have pre-defined and limited sets of values. For example, **type** attribute of the **input** element. In HTML and XML, these are called **enumerated attributes**. Under HTML 4, the interpretation of these values was case-insensitive, so a value of **TEXT** was equivalent to a value of **text**.

Under XHTML, the interpretation of these values is case-sensitive so all of these values are defined in lower-case.

Entity References as Hex Values

HTML and XML both permit references to characters by using hexadecimal value. In HTML these references could be made using either **&#Xnn;** or **&#xnn;** and they are valid but in XHTML documents, you must use the lower-case version only such as **&#xnn;**.

The <html> Element is a Must

All XHTML elements must be nested within the <html> root element. All other elements can have sub elements which must be in pairs and correctly nested within their parent element. The basic document structure is –

```
<!DOCTYPE html....>
```

```
<html>  
  <head> ... </head>  
  <body> ... </body>  
</html>
```

Loading [Mathjax]/jax/output/HTML-CSS/jax.js