***Syntax, doctype, metadata, tags, attributes etc***

1. *HTML SYNTAX*

The list below provides a quick reference to the rules that will ensure your markup is well-formed and valid.

**The Document Tree**

The elements in a web page are contained in a tree structure in which html is the root element that splits into the head and body elements. An element may contain other nested elements (although this very much depends on what the parent element is; for example, ap element can contain span, em, or strong elements, among others). Where this occurs, the opening and closing tags must be symmetrical.

**Case Sensitivity**

In HTML, tag names are case insensitive, but in XHTML they’re case sensitive. As such, in HTML, you can write the markup in lowercase, mixed case, or uppercase letters. So <p>this is a paragraph</p>, as is <P>this example</P>, and even <P>this markup would be valid</p>. In XHTML, however, you must use lowercase for markup: <p>This is a valid paragraph in XHTML</p>.

**Opening and Closing Tags**

In HTML, it’s possible to omit some closing tags (check each element’s reference to see whether an HTML closing tag is required), so this is valid markup: <p>This is my first paragraph.<p>This is my second paragraph.<p>And here’s the last one.

In XHTML, all elements must be closed. Hence the paragraph example above would need to be changed to: <p>This is my first paragraph.</p><p>This is my second paragraph.</p><p>And here’s the last one.</p>

For empty elements such as [img](http://reference.sitepoint.com/html/img" \o "specifies an inline image element), XHTML (that is not served with the application/xhtml+xml) requires us to use the XML empty element syntax:<elementname attribute="attributevalue"/>

**Readability Considerations**

A browser doesn’t care whether you use a single space to separate attributes, ten spaces, or even complete line breaks; it doesn’t matter, as long as some space is present.

In XHTML all attribute values must be quoted, so you’ll need to writeclass="gallery" rather than class=gallery. It’s valid to omit the quotes from your HTML, though it may make reading the markup more difficult for developers revisiting old markup.

1. *DOCTYPE*

According to HTML standards, each HTML document requires a document type declaration. The “DOCTYPE” begins the HTML document and tells a [validator](http://htmlhelp.com/tools/validator/index.html.en) which version of HTML to use in checking the document’s syntax. The following DOCTYPEs are commonly used:

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"**

**"http://www.w3.org/TR/html4/strict.dtd">**

This declares the document to be **HTML 4.01 Strict**. HTML 4.01 Strict is a trimmed down version of [HTML 4.01](http://www.w3.org/TR/html401/) that emphasizes structure over presentation. Deprecated elements and attributes (including most presentational attributes), frames, and link targets are not allowed in HTML 4 Strict. By writing to HTML 4 Strict, authors can achieve accessible, structurally rich documents that easily adapt to style sheets and different browsing situations. However, HTML 4 Strict documents may look bland on very old browsers that lack support for stylesheets.Newer browsers such as Internet Explorer 5 for Mac, Netscape 6, and [Mozilla](http://www.mozilla.org/) use a standards-compliant rendering for HTML 4 Strict documents. These browsers use a “quirks” mode for most other document types to emulate rendering bugs in older browsers.

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"**

**"http://www.w3.org/TR/html4/loose.dtd">**

This declares the document to be **HTML 4.01 Transitional**. HTML 4 Transitional includes all elements and attributes of HTML 4 Strict but adds presentational attributes, deprecated elements, and link targets.

Newer browsers such as Internet Explorer 5 for Mac, Netscape 6, and [Mozilla](http://www.mozilla.org/) use a standards-compliant rendering for HTML 4.01 Transitional documents that include the URI of the DTD in the DOCTYPE. These browsers use a “quirks” mode to emulate rendering bugs in older browsers if the URI is omitted:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"**

**"http://www.w3.org/TR/html4/frameset.dtd">**

This declares the document to be **HTML 4.01 Frameset**. HTML 4 Frameset is a variant of HTML 4 Transitional for documents that use frames.

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"**

**"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">**

This declares the document to be **XHTML 1.0 Strict**. [XHTML 1.0](http://www.w3.org/TR/xhtml1/) Strict is an XML version of HTML 4 Strict.

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

**"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">**

This declares the document to be **XHTML 1.0 Transitional**. [XHTML 1.0](http://www.w3.org/TR/xhtml1/) Transitional is an XML version of HTML 4 Transitional.

**<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"**

**"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">**

This declares the document to be **XHTML 1.0 Frameset**. [XHTML 1.0](http://www.w3.org/TR/xhtml1/) Frameset is an XML version of HTML 4 Frameset.

**<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">**

This declares the document to be **HTML 3.2**. HTML 3.2 is well supported by most browsers in use. However, HTML 3.2 has limited support for style sheets and no support for HTML 4 features such as frames and internationalization.

**<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">**

This declares the document to be **HTML 2.0**. [HTML 2.0](http://www.w3.org/MarkUp/html-spec/) is widely supported by browsers but lacks support for tables, frames, and internationalization, as well as many commonly used presentational elements and attributes.

1. Metadata

HTML lets you specify metadata - additional important information about a document in a variety of ways. The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| Name | Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc. |
| content | Specifies the property's value. |
| scheme | Specifies a scheme to interpret the property's value (as declared in the content attribute). |
| http-equiv | Used for http response message headers. For example http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie. |

You can use <meta> tag to specify important keywords related to the document and later these keywords are used by the search engines while indexing your webpage for searching purpose.

<meta name="keywords" content="HTML, Meta Tags, Metadata" />

You can use <meta> tag to give a short description about the document. This again can be used by various search engines while indexing your webpage for searching purpose.

<meta name="description" content="Learning about Meta Tags." />

You can use <meta> tag to give information about when last time the document was updated. This information can be used by various web browsers while refreshing your webpage

<meta name="revised" content="Tutorialspoint, 3/7/2014" />

A <meta> tag can be used to specify a duration after which your web page will keep refreshing automatically.

<meta http-equiv="refresh" content="5" />

You can use <meta> tag to redirect your page to any other webpage. You can also specify a duration if you want to redirect the page after a certain number of seconds.

<meta http-equiv="refresh" content="5; url=http://www.tutorialspoint.com" />

You can use <meta> tag to store cookies on client side and later this information can be used by the Web Server to track a site visitor.

<meta http-equiv="cookie" content="userid=xyz; expires=Wednesday, 08-Aug-15 23:59:59 GMT;" />

You can use <meta> tag to specify character set used within the webpage.

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

1. *Tags*

HTML tags are **keywords** (tag names) surrounded by **angle brackets**:

<tagname>content</tagname>

* HTML tags normally come **in pairs** like <p> and </p>
* The first tag in a pair is the **start tag,** the second tag is the **end tag**
* The end tag is written like the start tag, but with a **slash** before the tag name

HTML **tags** mark the beginning and end of HTML **elements** and may contain **attributes**. They are used as follows:

<tagname>Element content</tagname>

<tagname attributename="attributevalue">Element content</tagname>

*<!-- or, for the few attributes without values, simply: -->*

<tagname attributename>Element content</tagname>

*<!-- or, for the few elements without content, simply: -->*

<tagname>

<tagname attributename="attributevalue">

1. *Attributes*

Attributes provide additional information about HTML elements.

* HTML elements can have **attributes**
* Attributes provide **additional information** about an element
* Attributes are always specified in **the start tag**
* Attributes come in name/value pairs like: **name="value"**

<html lang="en-US"> - The language is declared in the **lang** attribute.

<p title="About W3Schools">Something</p> -When you move the mouse over the element, the title will be displayed as a tooltip.

<a href="http://www.w3schools.com">This is a link</a> - HTML links are defined with the **<a>** tag. The link address is specified in the **href** attribute.

<img src="w3schools.jpg" width="104" height="142"> - The filename of the source (**src**), and the size of the image (**width** and **height**) are all provided as **attributes.**

<img src="w3schools.jpg" alt="W3Schools.com" width="104" height="142"> - The **alt** attribute specifies an alternative text to be used, when an HTML element cannot be displayed. The value of the attribute can be read by "screen readers".