Quick Start Guide to HTML

An introduction to HTML and tutorial on the basic structure of web documents. HTML-5.com is a great guide for web developers. TV Series & Actors and Actresses. Follow TV Series and HTML 5 on Google+.

HTML-5.com ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">HTML 5</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb<span itemprop="title">HTML Tutorials</span> ➢ itemscopehttp://data-vocabulary.org/Breadcrumb **Quick Start Guide to HTML**

Quick Start Guide to HTML

Tutorial on Basic Page Structure of HTML Documents

HTTP Headers for HTML 5

This section describes the HTTP headers for HTML 5 and how to set them. (Setting the HTTP headers requires a bit of technical know-how. You may want to skip directly to the Basic HTML Code section. Bookmark this page in your Favorites so you can come back to it later.)

Basic HTML Code AKA skeleton code or boilerplate

Every HTML document should include an <?xml?> declaration, <!DOCTYPE> declaration, **html** element, **head** section and **body** section.

HTML Content Models

The <dfn>content model</dfn> of an HTML element determines what type of content may be coded inside it.

HTML Serialization

This section explains <dfn>HTML Serialization</dfn> and why the `Content-Type` header is set to `application/xhtml+xml`.

back to top

HTTP Headers for HTML 5

How to set the Content-Type Header for HTML

(Setting the parameters to send the proper HTTP headers from a web server requires a bit of technical know-how. You may want to skip directly to the Basic HTML Code section. You will want to come back later if your site shows a "torn page" (broken page) icon in the address bar of Internet Explorer so bookmark this page in your Favorites to find it easily.

<dfn>HTTP headers</dfn> are sent to the client (browser) in the response from the web server before the document itself. The HTTP headers which control how an HTML 5 document is displayed might look like this:

```
Content-Type: application/xhtml+xml; charset=UTF-8
Cache-Control: max-age=120
X-UA-Compatible: IE=9
```

It is highly recommended that the `charset` attribute specifying the character encoding of the HTML page be included in the `Content-Type` header for non-XML user agents as well as in the xml declaration for XML parsers.

For static web pages, it may be necessary to add the MIME Type for HTML 5 <dfn>Polyglot Documents</dfn> to the HTTP web server configuration to send the appropriate Content-Type header. With the Apache HTTP Server, for example, the HTML 5 MIME Type can be added to the .htaccess file(s):

```
  DirectoryIndex index.html
  ErrorDocument 404 /error.html
  AddType application/xhtml+xml;charset=UTF-8 html
```

For a detailed explanation of why the `Content-Type` header is set to `application/xhtml+xml`, see the HTML Serialization section below.

When a program or server-side scripting is generating HTML, the language probably has an API to send the proper HTTP headers.

- How to Set Content-Type via .htaccess (recommended)
- How to Set Content-Type in ASP .NET
- How to Set Content-Type in JavaScript
- How to Set Content-Type in Perl CGI
- How to Set Content-Type in PHP
- How to Set Content-Type in Ruby on Rails

back to top

HTML Boilerplate

Basic HTML Code

The code for a simple HTML page is shown below. This is the HTML equivalent of a "Hello World!" program, but web designers can copy this code and use it as a skeleton for developing much more extensive web pages.

```
<a mode="pre" href="../tags/xml-declaration/"><?xml version="1.0" encoding="UTF-8"?></a>
  <a mode="pre" href="../tags/doctype-declaration/"><!DOCTYPE html></a>
  <a mode="pre" href="../tags/html-tag/index.html#" title="<html> tag"><html xmlns="http://www.w3.org/
     <a mode="pre" href="../tags/head-tag/"><head></a>
        <a mode="pre" href="../tags/title-tag/"><title></a>  Example Only</title>
     </head>
     <a mode="pre" href="../tags/body-tag/"><body></a>
        <a mode="pre" href="../tags/p-tag/"><p></a>  This is only an example. For more information, se
           <a mode="pre" href="../tags/a-tag/"><a href</a>  ="<a mode="pre" href="http://www.ExampleOn
        </p>
     </body>
  </html>
```

(By the way, there is a real "Hello World!" example of the HTML 5 <canvas> tag also.)

back to top

HTML Element Content Models

An HTML element's <dfn>content model</dfn> specifies what is allowed inside the element.

Content Model of <head> Tag

The expected content of the <head> tag consists of HTML elements categorized as metadata content elements. (In the HTML tag list, click on an HTML tag to see the content model of that element.)

Content Model of <body> Tag

The expected content of the <body> tag consists of HTML elements categorized as flow content elements. (In the HTML tag list, click on an HTML tag to see the content model of that element.)

Flow Content vs. Phrasing Content

When the expected content of an HTML element is flow content, you can code either flow content elements or phrasing content elements, with some occasional restrictions.

In other words, flow content elements can *only* be used where flow content is expected.

When the expected content of an HTML element is phrasing content, you cannot code any flow content elements, *only* phrasing content elements, with some occasional additional restrictions.

In other words, phrasing content elements can be used where either flow content or phrasing content is expected.

back to top

HTML Serialization

Serialization of HTML 5 Documents

(Whether a web page is being processed as HTML, xHTML or pure XML is a technical issue for advanced users. You may want to skip directly to the Basic HTML Code section.)

The type of serialization of an HTML document refers to the syntax used when converting the HTML from an internal document model to a stream of bytes to be stored or transmitted or vice-versa. The HTML 5 specifications allow coding HTML documents in either the HTML style, based on 1997 HTML 4 and earlier specifications, or the xHTML style, based on the 1998 XML, 1999 Namespaces and 2000-2001 XHTML 1.x W3C recommendations. The xHTML style of code has a number of advantages, including:

- support for tags with non-HTML namespaces, such as rdf:RDF, and
- it is required for <dfn>Polyglot Documents</dfn>
- the same coding rules can be used for including HTML code in other XML-based document formats, such as in a `<description>` element in an RSS feed, where the HTML serialization would *not* be valid

HTML 5 has been designed to be backward compatible with both the 1997-1999 HTML 4 standards and the 2000-2001 XHTML 1.x W3C recommendations. The XML serialization of HTML 5 merges these two standards, and is already understood by virtually all web browsers including xHTML-based mobile browsers.

Processing HTML Code

HTML code can be processed in at least three different ways:

1. as an HTML serialization of HTML, by web browsers and other software that process HTML documents from that serialization format
2. as an XHTML-compatible XML-based serialization of HTML (xHTML), by web browsers and other software that process HTML documents from that serialization format
3. as pure XML, by XSLT and other software that process documents as XML

Polyglot HTML documents are HTML documents that are coded in a manner that allows them to be read in any of those three ways. This avoids having to limit the documents to only the parsers that process one serialization or the other or having to code the same content in two or three different ways.

Polyglot documents can be delivered as:

1. `text/html` to traditional web browsers,
2. `application/xml` to XML parsers or
3. `application/xhtml+xml`, the combination of both, which works with web browsers on cell phones and other handheld devices as well as on desktop computers

If you start creating polyglot documents now your web pages will be well positioned for both current and future HTML browsers and mobile devices.

back to top

THE END