HTML Quick Reference

The HyperText Markup Language (HTML) is composed of a set of elements that define a document and guide its display. This document presents a concise reference guide to HTML, listing the most commonly used elements from Versions 1 and 2 of HTML, and giving a brief description of those elements.

Users should be aware that HTML is an evolving language, and different World-Wide Web browsers may recognize slightly different sets of HTML elements. For general information about HTML including plans for new versions, see http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html.

Introduction

An HTML element may include a name, some attributes and some text or hypertext, and will appear in an HTML document as

```
<tag_name> text </tag_name>
<tag_name attribute_name=argument> text </tag_name>, or just
<tag_name>
```

For example:

```
<title> My Useful Document </title>
```

and

```
<a href="argument"> text </a>
```

An HTML document is composed of a single element:

```
<html> . . . </html>
```

that is, in turn, composed of head and body elements:

```
<head> . . . </head>
```

and

```
<body> . . . </body>
```

To allow older HTML documents to remain readable, `<html>`, `<head>`, and `<body>` are actually optional within HTML documents.

Elements usually placed in the head element

`<isindex>`
: Specifies that the current document describes a database that can be searched using the index search method appropriate for whatever client is being used to read the document. For example, a Lynx user will use the "s" keyboard command.

`<title> . . . </title>`
: Specify a document title. Note that the title will not appear on the document as is customary on printed documents. It will usually appear in a window bar identifying the contents of the window. HTML header tags perform the functions usually reserved for titles.

`<base href="URL">`
: Specify the name of the file relative to which partially qualified pathnames in URLs should be interpreted. If not otherwise specified the URL containing the document being displayed is used as the base.

`<link rev="RELATIONSHIP" rel="RELATIONSHIP" href="URL">`
: The link tag allows you to define relationships between the document containing the link tag and the document specified in the "URL". The `rel` attribute specifies the relationship between the HTML file and the Uniform Resource Locator (URL). The `rev` attribute (for "reverse") specifies the relationship between the URL and the HTML file. For example, `<link rev="made" href="URL">` indicates that the file maker or owner is described in the document identified by the URL. (Note that link tags are not displayed on the screen as part of the document. They define static relationships, not hypertext links.)

Elements usually placed in the body element

The following sections describe elements that can be used in the body of the document.

Text Elements

`<p>`
: The end of a paragraph that will be formatted before it is displayed on the screen.

`<pre> . . . </pre>`
: Identifies text that has already been formatted (preformatted) by some other system and must be displayed as is. Preformatted text may include embedded tags, but not all tag types are permitted. The `<pre>` tag can be used to include tables in documents.

`<listing> . . . </listing>`
: Example computer listing; embedded tags will be ignored, but embedded tabs will work. This is an archaic tag.

`<xmp> . . . </xmp>`
: Similar to `<pre>` except no embedded tags will be recognized.

`<plaintext>`
: Similar to `<pre>` except no embedded tags will be recognized, and since there is no end tag, the remainder of the document will be rendered as plain text. This is an archaic tag. Note that some browsers actually recognize a `</plaintext>` tag, even though it is not defined by the standard.

`<blockquote> . . . </blockquote>`
: Include a section of text quoted from some other source.

Hyperlinks or Anchors

`<a name="anchor_name"> . . . </a>`
: Define a target location in a document

`<a href="#anchor_name"> . . . </a>`
: Link to a location in the base document, which is the document containing the anchor tag itself, unless a base tag has been specified.

`<a href="URL"> . . . </a>`
: Link to another file or resource

`<a href="URL#anchor_name"> . . . </a>`
: Link to a target location in another document

`<a href="URL?search_word+search_word"> . . . </a>`
: Send a search string to a server. Different servers may interpret the search string differently. In the case of word-oriented search engines, multiple search words might be specified by separating individual words with a plus sign (+).

An anchor must include a `name` or `href` attribute, and may include both. There are several optional attributes, but they are rarely encountered.

The structure of a Uniform Resource Locator (URL) may be expressed as:

resource_type:additional_information

where the possible resource types include: `file`, `http`, `news`, `gopher`, `telnet`, `ftp`, and `wais`, among others, and each resource type relates to a specific server type. Since each server performs a unique function, each resource type requires different `additional_information`. For example `http` and `gopher` URLs will have a structure like:

resource_type://host.domain:port/pathname

The colon followed by an integer TCP port number is optional, and is used when a server is listening on a non-standard port.

Strictly speaking, the `anchor_name` and `search_word` information included in the `name` and `href` attributes in the examples above are part of the URL. They are presented as separate entities for simplicity. A more complete description of URLs is presented in http://www.w3.org/hypertext/WWW/Addressing/Addressing.html

Headers

`<h1>...</h1>` Most prominent header
`<h2>...</h2>`
`<h3>...</h3>`
`<h4>...</h4>`
`<h5>...</h5>`
`<h6>...</h6>` Least prominent header

Logical Styles

`<em>...</em>`	Emphasis
`<strong>...</strong>`	Stronger emphasis
`<code>...</code>`	Display an HTML directive
`<samp>...</samp>`	Include sample output
`<kbd>...</kbd>`	Display a keyboard key
`<var>...</var>`	Define a variable
`<dfn>...</dfn>`	Display a definition (not widely supported)
`<cite>...</cite>`	Display a citation

Physical Styles

`<b>...</b>`	**Boldface**
`<i>...</i>`	*Italics*
`<u>...</u>`	Underline
`<tt>...</tt>`	`Typewriter font`

Definition list/glossary: <dl>

```
<dl>
<dt> First term to be defined
<dd> Definition of first term
<dt> Next term to be defined
<dd> Next definition
</dl>
```

The `<dl>` attribute `compact` can be used to generate a definition list requiring less space.

Present an unordered list:

```
<ul>
<li> First item in the list
<li> Next item in the list
</ul>
```

Present an ordered list:

```
<ol>
<li> First item in the list
<li> Next item in the list
</ol>
```

Present an interactive menu: <menu>

```
<menu>
<li> First item in the menu
<li> Next item
</menu>
```

Present a directory list of items: <dir>

```
<dir>
<li> First item in the list
<li> Second item in the list
<li> Next item in the list
</dir>
```

Items should be less than 20 characters long.

Entities

`&keyword;`
: Display a particular character identified by a special keyword. For example the entity `&` specifies the ampersand (&), and the entity `<` specifies the less than (<) character. Note that the semicolon following the keyword is required, and the keyword must be one from the lists presented in: http://www.w3.org/pub/WWW/MarkUp/html-spec/html-spec_9.html#SEC9.7

`&#ascii_equivalent;`
: Use a character literally. Again note that the semicolon following the ASCII numeric value is required.

HTML Forms Interface

The HTML forms interface allows document creators to define HTML documents containing forms to be filled out by users. When a user fills out the form and presses a button indicating the form should be "submitted," the information on the form is sent to a server for processing. The server will usually prepare an HTML document using the information supplied by the user and return it to the client for display.

The following tags implement the forms interface:

- `<form>...</form>`
- `<input>`
- `<select>...</select>`
- `<option>`
- `<textarea>...</textarea>`

The last four tags can only be used within a `<form>...</form>` element.

Define a form

`<form>...</form>`

Defines a form within an HTML document. A document may contain multiple `<form>` elements, but `<form>` elements may not be nested. Note that non-form tags can be used within a `<form>` element. Attributes and their arguments:

`action="URL"`:
: The location of the program that will process the form.

`method=data_exchange method`
: The method chosen to exchange data between the client and the program started to process the form: One of `get` or `post`. `post` is preferred for most applications.

Example:
: `<form action="http://www.ku.edu/cgi-bin/register" method=post>...</form>`

Define an input field

`<input>` (there is no ending tag)

Defines an input field where the user may enter information on the form. Each input field assigns a value to a variable which has a specified `name` and a specified data `type`. Attributes and their arguments:

`type="variable_type"`
: Specifies the data type for the variable, where:

  - `type="text"` and `type="password"` fields accept character data
  - `type="checkbox"` fields are either selected or not
  - `type="radio"` fields of the same name allow selection of only one of the associated values
  - `type="submit"` defines an action button that sends the completed form to the query server
  - `type="reset"` defines a button that resets the form variables to their default values

- `type="hidden"` defines an invisible input field whose value will be sent along with the other form values when the form is submitted. This is used to pass state information from one script or form to another.
- `type="image"`defines an image map within a form and returns the coordinates of a mouse click within the image.

`name="textstring"`
: where `textstring` is a symbolic name (not displayed) identifying the `input` variable as in:
`<input type="checkbox" name="box1">`

`value="textstring"`
: where the meaning of `textstring` depends on the argument for `type`.

- For `type="text"` or `type="password"`, `textstring` is the default value for the `input` variable. Password values will not be shown on the user's form. Anything entered by the user will replace any default value defined with this attribute.
- If `type="checkbox"` or `type="radio"`, `textstring` is the value that will be sent to the server if the checkbox is "checked".
- For `type="reset"` or `type="submit"`, `textstring` is a label that will appear on the submit or reset button in place of the words "`submit`" and "`reset`".

`checked`
: No arguments. For `type="checkbox"` or `type="radio"`, if `checked` is present the `input` field is "checked" by default.

`size="display_width"`
: where `display_width` is an integer value representing the number of characters displayed for the `type="text"` or `type="password" input` field.

`maxlength="string_length"`
: where `string_length` is the maximum number of characters allowed within `type="text"` or `type="password"` variable values. This attribute is only valid for single line `"text"` or `"password"` fields.

Define a select field

`<select>...</select>`

Defines and displays a set of optional list items from which the user can select one or more items. This element requires an `<option>` element for each item in the list. Attributes and their arguments:

`name="textstring"`
: where `textstring` is the symbolic identifier for the `select` field variable.

`size="list_length"`
: where `list_length` is an integer value representing the number of `<option>` items that will be displayed at one time.

`multiple`
: No arguments. If present, the `multiple` attribute allows selection of more than one `<option>` value.

Define a select field option

`<option>`

Within the `<select>` element the `<option>` tags are used to define the possible values for the `select` field. If the attribute `selected` is present then the `option` value is selected by default. In the following example all three options may be chosen but bananas are selected by default.

```
<select multiple>
<option>Apples
<option selected>Bananas
<option>Cherries
</select>
```

Define a text area

`<textarea>` . . . default text . . . `</textarea>`

Defines a rectangular field where the user may enter text data. If "default text" is present it will be displayed when the field appears. Otherwise the field will be blank. Attributes and their values:

`name="textstring"`
: `textstring` is a symbolic name that identifies the `<textarea>` variable.

`rows="num_rows"` and `cols="numcols"`
: Both attributes take an integer value which represents the lines and number of characters per line in the `<textarea>` to be displayed.

Miscellaneous

`<!--` text `-->`
: Place a comment in the HTML source

`<address>` . . . `</address>`
: Present address information

`<img src="URL" alt="Alternate Text">`
: Embed a graphic image in the document. Attributes:
: `src`
: Specifies the location of the image.
: `alt`
: Allows a text string to be put in place of the image in clients that cannot display images.
: `align`
: Specify a relationship to surrounding text. The argument for `align` can be one of `top`, `middle`, or `bottom`.
: `ismap`
: If `ismap` is present and the image tag is within an anchor, the image will become a "clickable image". The pixel coordinates of the cursor will be appended to the URL specified in the anchor if the user clicks within the ismap image. The resulting URL will take the form "URL?m,n" where m and n are integer coordinates, and the URL will specify the location of a program that will examine the pixel coordinates, and return an appropriate document.

`<br>`
: Forces a line break immediately and retains the same style.

`<hr>`
: Places a horizontal rule or separator between sections of text.

Additional Information

For a tutorial introduction to HTML see:
http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html.

For an introduction to forms within HTML see: An Instantaneous Introduction to CGI Scripts and HTML Forms.

For general information about HTML, see
http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

The current URL is /~acs/docs/other/HTML_quick.shtml
This file was last modified Tuesday, 03-Jul-2001 09:55:56 CDT.
Questions about computing to question@ku.edu.
Problems, comments about this Web site to acsweb@ku.edu.