A Beginner's Guide to HTML

Introduction

This document is a *primer* for writing documents in **HTML** (HyperText Markup Language), the markup language used in the World Wide Web project and the NCSA Mosaic networked information browser. This is not a complete overview of HTML, but covers enough ground to have you creating full−featured HTML documents within an hour or two.

This guide contains the following sections:

Basics of HTML

HTML is a very simple SGML−based markup language −− it is complex enough to support *basic* online formatting and presentation of hypermedia documents, but no more complex. In fact, if you are familiar with LaTeX, TeX, troff, or Texinfo, you can breathe a sigh of relief at this point, since HTML is quite a bit simpler than any of those.

HTML documents use *tags* to indicate formatting or structural information. A tag is simply a left angle bracket (<) followed by a *directive* and zero or more parameters followed by a right angle bracket (>). The remainder of this document explains the various HTML directives.

A Beginning Example

For people who prefer to learn by doing, here is an example of a simple HTML document:

```
<title>Simple example of an HTML document.</title>
<h1>A simple example.</h1>

This is a simple HTML document.  This is the first
paragraph. <p>

This is the second paragraph.  This is a word in
<i>italics</i>.  This is a word in <b>bold</b>.
Here is an inlined GIF image: <img src="myimage.gif">.
<p>

This is the third paragraph.  Here is a hypertext
link from the word <a href="subdir/myfile.html">foo</a>
to a document called "subdir/myfile.html". <p>

<h2>A second−level header.</h2>

Here is a section of text that should show up in a
fixed−width font (as if it were a computer listing
or a verse of poetry): <p>

<pre>
    The cat in the hat
    fell to the ground and went splat.
</pre>

This is a bulleted list with two items: <p>

<ul>
<li> First item goes here.
<li> Second item goes here.
</ul>
```

```
This is the end of my example document. <p>

<address>John Bigbooty</address>
```

Note that any HTML document from anywhere on the net that you access with Mosaic can be easily used as an example; just use the **Document Source** option in Mosaic's **File** menu to call up a window that will show you the HTML for the current document being viewed.

Titles and Headers

Every HTML document should have a *title*: about half a dozen words that declare the document's purpose. Titles are not displayed as part of the document text, but are rather displayed separately from the document by most browsers (at the top of the window in NCSA Mosaic) and used for document identification in certain other contexts. (The title of this document is "A Beginner's Guide to HTML".)

The title generally goes on the first line of the document. Here is an example title:

```
<title>This is my document's title.</title>
```

Notice that the *directive* for the title tag is, appropriately enough, `title`. Note also the fact that there are both *starting* and *ending* title tags, and that the ending tag looks just like the starting tag except a slash (/) precedes the directive. (This is also a good time to note that HTML is not case sensitive: both <title> and <TITLE> mean the same thing.)

Headers are displayed within the document, generally using larger and/or bolder fonts than normal document text. There are six levels of headers (numbered 1 through 6), with 1 being the largest. (Usually only levels 1 through 3 are used with any frequency.)

Here is an example level 1 header:

```
<h1>This is a level 1 header.</h1>
```

Here is an example level 2 header:

```
<h2>This is a level 2 header.</h2>
```

Most documents use the same five or six words both for the title and for the initial (level 1) header; for example, the first two lines of the HTML source for this document are:

```
<title>A Beginner's Guide to HTML</title>
<h1>A Beginner's Guide to HTML</h1>
```

Paragraphs and Formatting

Since HTML is a markup language for creating formatted documents, a basic assumption is that newlines and whitespace aren't significant in normal text, and that word wrapping can occur at any place. Therefore, terminating a paragraph with a single blank line, for example, is not sufficient: **each paragraph should be terminated by a paragraph tag**. The HTML paragraph tag is <p>.

Here is an example paragraph, complete with terminating paragraph tag:

```
This is my first sentence.  This is my
second sentence.  This is my third sentence.
This is the end of the paragraph. <p>
```

Special Characters

Three characters out of the entire ASCII (or ISO 8859) character set are *special* and cannot be used "as–is" within an HTML document. These characters are left angle bracket (<), right angle bracket (>), and ampersand (&).

Why is this? The angle brackets are used to specify HTML tags (as shown above), while ampersand is used as the *escape mechanism* for these and other characters:

- `<` is the *escape sequence* for <
- `>` is the escape sequence for >

- & is the escape sequence for &

Note that "escape sequence" only means that the given sequence of characters represents the single character in an HTML document: the conversion to the single character itself takes place when the document is formatted for display by a reader.

Note also that there are additional escape sequences that are possible; notably, there are a whole set of such sequences to support 8−bit character sets (namely, ISO 8859−1); for example:

- ö is the escape sequence for a lowercase o with an umlaut: ö
- ñ is the escape sequence for a lowercase n with an tilde: ñ
- È is the escape sequence for an uppercase E with a grave mark: È

Many such escapes exist; a canonical list is here.

Basic Special Effects

Individual words or sentences in paragraphs can be put in **bold**, *italic*, or fixed−width styles. Correspondingly, you should know about the following three directives:

- <i>text</i> puts text in italics (the result of the example would be *text*).
- text puts text in bold (the result of the example would be **text**).
- <code>text</code> puts text in a fixed−width font (the result of the example would be text).

Inlined Images

A value−added feature of NCSA Mosaic is that images (in X bitmap or GIF formats) can be displayed inside documents, right in the middle of document text. For example, here's a picture of Elvis: 

Here's how that image was inlined into the document text above:

```
<img align=top src="elvis-small.gif">
```

Note in particular the align=top parameter −− this directs the document viewer to align adjacent text with the top of the image (rather than the bottom, as is the default). So if you just say <img src="elvis−small.gif">, you'll get this effect:  This default behavior is especially suited for using an image at the beginning of a paragraph (see the next paragraph as an example).

 Multiple instances of the img tag can be scattered through the document, but note that each such image takes time to process and thus slows down the initial display of the document. (Using a particular image multiple times in a document causes no performance hit compared to using the image only once, though.)

(Note that the img tag is an HTML extension that is currently only understood by NCSA Mosaic and not by most other World Wide Web browsers.)

Hypertext Links

Since the whole point behind HTML's existence is to allow networked hypertext, it's about time we get to that part of the language. There is a single hypertext−related directive, and it's a, which stands for *anchor* (which is a common term for one end of a hypertext *link*).

An anchor is commonly used to point *to* somewhere from the current document. Here's how that works:

- Start by opening the anchor with the leading angle bracket and the anchor directive: <a
- Name the document that's being pointed to, by giving the parameter href="document.html", and follow that with the closing angle bracket: >.
- Give the text that should show up in the current document as the hypertext link (i.e. the text that will be in a different color and/or underlined, to indicate that clicking on it follows the hyperlink).
- End by giving the ending anchor tag:

So, an example hypertext reference looks like this:

```
<a href="subdir/document.html">some text</a>
```

.......which causes "some text" to be the hyperlink to the document named "subdir/document.html".

Note that inlined images (explained above) can serve as the contents of anchors. For example, the following picture of Elvis is a hyperlink to the NCSA Mosaic documentation:  -- so when you click on Elvis, you get the Mosaic docs. The HTML for that was:

```
<a href="http://machine.name/subdir/file.html">
<img src="elvis-small.html"></a>
```

Another Use For Anchors

Anchors can also be used to say "hey, point to me". If you want to point to a specific location in a document, you can put a *named anchor* in the document at that location and then point to that named anchor as part of a hyperlink reference.

Here's an example. In document A, I have a traditional hyperlink, but the hypertext reference (href) gives not only the filename ("document-b.html") but also the name of a named anchor in the referenced document ("foobar"), with those two things separated by a hash mark ("#"):

```
This is my <a href="document-b.html#foobar">link</a>.
```

Meanwhile, in document B, I have a lot of other text, and then the following:

```
Here's <a name="foobar">some random text</a>.
```

Therefore, the link in document A points directly at the words "some random text" in document B, and following the link from document A will not only jump the reader to document B but will position document B in the window such that "some random text" is immediately visible no matter where in document B it's located. (In Mosaic, the window will be scrolled far enough down so "some random text" will be on the top line of the viewable region of the window, if possible.)

An offshoot of this technique is that you can have hyperlink cross-references *within* a single document: to point to a named anchor with name "blargh" in the current document, just give "#blargh" as the href for the hyperlink (omitting a filename):

```
I'm pointing to the named anchor "blargh" in this
document with this <a href="#blargh">link</a>.
```

Bulleted and Numbered Lists

A basic *bulleted list* can be produced as follows:

- Start with an opening <ul> tag.
- Give the items one at a time, each preceded by a <li> tag. (There is no closing tag for list

items.)
- End with a closing `</ul>` tag.

So, here's an example two−item list:

```
<ul>
<li> First item goes here.
<li> Second item goes here.
</ul>
```

For a *numbered list*, do the same thing except use the `ol` directive rather than the `ul` directive. For example:

```
<ol>
<li> First item goes here.
<li> Second item goes here.
</ol>
```

Lists can be arbitrarily nested: any list item can itself contain lists. Also note that no paragraph separator (or anything else) is necessary at the end of a list item; the subsequent `<li>` tag (or list end tag) serves that role. (One can also have a number of paragraphs, each themselves containing nested lists, in a single list item, and so on.)

An example nested list follows:

```
<ul>
<li> This item includes a nested list.
  <ul>
  <li> First item of nested list.
  <li> Second item of nested list.
  </ul>
<li> Second item goes here.
  <ul>
  <li> Only item of second nested list.
  </ul>
</ul>
```

This is displayed as:

- This item includes a nested list.
  - First item of nested list.
  - Second item of nested list.
- Second item goes here.
  - Only item of second nested list.

Description Lists

A *description list* usually consists of alternating "description titles" (dt's) and "description descriptions" (dd's). Think of a description list as a glossary: a list of terms or phrases, each of which has an associated definition.

Here's an example description list:

```
<dl>
<dt> This is the first "title".
<dd> This is the first "description", followed by
     a lot of completely meaningless text intended to
     make sure that at least one line wrap will occur
     for a reasonable window width, and if you don't
     have a window width wide enough to cause at least
     a single line wrap, you should narrow your window
     at this point, otherwise this example is pretty
     much pointless and here I sit getting carpal
     tunnel syndrome typing in all this verbage all
     for nothing.
<dt> This is the second "title".
<dd> This is the second "description".
</dl>
```

......which comes out looking like this:

This is the first "title".
: This is the first "description", followed by a lot of completely meaningless text intended to make sure that at least one line wrap will occur for a reasonable window width, and if you

don't have a window width wide enough to cause at least a single line wrap, you should narrow your window at this point, otherwise this example is pretty much pointless and here I sit getting carpal tunnel syndrome typing in all this verbage all for nothing.

This is the second "title".

This is the second "description".

Titles and descriptions can contain arbitrary items: multiple paragraphs (separated by paragraph tags), lists, other description lists, or whatever.

Preformatted Text

To put whole sections of text in a fixed−width font and to also cause spaces, newlines, and the like to be significant (e.g., for program listings, or plaintext dumps of numerical spreadsheets) you can use the pre tag ("pre" stands for preformatted). For example, the following HTML:

```
<pre>
column 1      column 2      column 3
--------      --------      --------
   133.0         115.0         332.5
 + 556.0       + 332.6       + 229.3
 = 689.0       = 447.6       = 561.8
</pre>
```

.......will result in exactly this:

```
column 1      column 2      column 3
--------      --------      --------
   133.0         115.0         332.5
 + 556.0       + 332.6       + 229.3
 = 689.0       = 447.6       = 561.8
```

No surprises there. (You should be aware that you can also embed hypertext references inside pre sections without losing the formatted effects, which is good. This capability is used, for example, in the manual page interfaces provided through Mosaic.)

In general, you should try to avoid using pre whenever possible under the principle that the final results will be much less flexible, and attractive, than full HTML. (Most people seem to think that preformatted, fixed−width text −− an artifact of the typewriter and primitive computer era −− looks pretty baroque compared to formatted text.)

Troubleshooting

- While certain HTML constructs can be nested (for example, you can have an anchor within a header), they cannot be overlapped. For example, the following is invalid HTML:

  ```
  <h1>This is <a name="foo">invalid HTML.</h1></a>
  ```

  Since many HTML parsers aren't very good at handling invalid HTML, it is always good to avoid doing bad things like overlapping constructs.
- When an img tag points at an image that does not exist or cannot be otherwise obtained from whatever server is supposed to be serving it, the NCSA logo will be substituted in place. For example, doing <img href="doesNotExist.gif"> (where "doesNotExist.gif" does not exist) causes the following to be displayed:

  If this happens to you, first make sure that the referenced image does in fact exist, then make sure the remote server (if any) can actually serve it, then make sure the image file is uncorrupted (and that your server is not corrupting it −− the NCSA httpd doesn't corrupt images, but certain other common http servers do).

For More Information

The official HTLM spec exists here.

The in−development HTML RFC is here.

A description of SGML, the Standard Generalized Markup Language on which HTLM is based, is here.

A simple overview of Universal Resource Locators (the extended filename references used in hypertext links and in the `src` part of an `img` tag) is here; this overview is still incomplete and will improve in the future.

The URL specification itself is here.

A style guide for online hypertext document structures can be found here.