

Introduction to HTML

What is HTML?

Technically speaking, HTML stands for HyperText Markup Language. HTML is an SGML application. [SGML](#) or Standardized Generalized Markup Language is a system of text rules that had its roots in the late 1960's and was pioneered by a man named [Dr. Charles Goldfarb](#).

All systems of working with text have a basic set of rules. These rules can be referred to as a markup language. To try to ease into this difficult concept lets take a simple letter as an example and attempt to describe its markup language.

If we look at this sample letter:

```
November 3, 1996

John Q. Student
12345 UTD Drive
Richardson, TX 75000

Dear John,

I am sorry that it has to end this way, but I have met a brilliant Internet expert who I wish to follow to the ends of Cyberspace. I am afraid that I can no longer be your E-mail girlfriend.

Things I like about him:
1. He is kind of funny (looking).
2. He is tall.
3. He has a computer

Please don't bother to write back as I will never speak to you again.

Yours always,

Suzy Q. Byte
Your ex-girlfriend

encl: The floppy disk you gave me for our anniversary.
```

We could break the letter down into logical sections:

```
<Date>
<Address>
<Salutation>
<Body>
<Closing>
```

Introduction to HTML

And within each section we could further subdivide:

```
<Date>

<Address>
  <Name>
  <Street>
  <Town, State, Zip>

<Salutation>

<Body>
  <Paragraph>
  <List>
  <Paragraph>

<Closing>
  <Name>
  <Title>
  <Enclosures>
```

Now, if we were to define a set of rules for each section we would have a simple markup language. For example:

1. Print the date in the format: Month XX, year: in the font Times New Roman at 12 point.
2. Print the address in font Times New Roman at 12 point.
3. Print the salutation in font Times New Roman at 12 point.
4. Print the body in font Arial at 12 point.
5. Print the closing in font Times New Roman at 12 point.

That is really all there is to a markup language. It is concerned with the exact layout and formatting of a document as it is to occur in its final output.

It seems simple enough, however there is one glaring problem with this method of defining a document. It is very platform dependent. This means that it can only be used on identical systems. For example, what would happen if this document and its related markup language were transplanted to another system that did not have the font Times New Roman? Something would definitely not work right.

This is exactly the problem that Dr. Goldfarb was interested in addressing. He reasoned that a better markup language would not be quite as specific and would hence be **platform independent**. In other words, we are not interested that the address is Times New Roman 12 point, but rather that the address appears at that particular spot in the document. This way, a document could be moved to different systems and each system could use its own built in set of rules on how to display the corresponding general markup.

In 1986, the International Organization for Standards (ISO) in Geneva adopted SGML as "Information Processing - Text and Office Systems - Standard Generalized Markup Language.

So HTML is a SGML based markup language... meaning that it is not concerned with the specifics of how a document is rendered, merely where each part of the document begins and ends.

Introduction to HTML

Using Standard Tags in HTML

Since HTML is a standardized markup language, we use a limited set of elements or “**tags**” as identifiers to define the structure of a document. A tag is always contained within the “<” and “>” symbols to denote that it is an identifier, and often contains a closing tag which is contained within the “</” and “>” symbols. (The “/” differentiates it as an ending tag.)

An interesting note about HTML tags is that they may be, and quite often are contained within one another. For example, the text attribute BOLD is represented by the set of tags:

```
<b>Everything within is rendered in bold</b>
```

And the text attribute ITALICS is represented by the set of tags:

```
<i>Everything within is rendered in italics</i>
```

If a bold, italics font were desired, the tags would simply be used in conjunction with one another to produce:

```
<b><i>Render in bold, italics</i></b>
```

This use of multiple tags within one another is known as **nesting**. There is one little thing to remember about nested tags, internal tags should be closed before external ones. In the preceding example, the bold tags completely surround the italics tags.

If the closing Italics tag had been outside of the bold tag, the page might still display *OK in this particular case*, but the author would certainly look like an amateur. In addition, if some tags are not nested properly they may destroy the way the document is displayed.

Format of an HTML document

All HTML documents must follow a very basic format. It is quite simple, but must be strictly adhered to. There are five specific tags that must be contained within each document and their format is as follows:

```
1.) <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
2.) <HTML>

3.) <HEAD>
4.) <TITLE>The title of the document goes here</TITLE>
   ...other heading elements go here.
   </HEAD>

5.) <BODY>
   The BODY of the document goes here.
   ...
   </BODY>
</HTML>
```

The 5 required tags are explained as follows:

1. The **DOCTYPE** definition or DTD as it is known for short simply tells which version of HTML is being used to write the document. Just as books and software have version numbers, HTML is also being constantly revised so the DTD tells browsers what kinds of things they can expect to be contained within a document.
2. **HTML** denotes the beginning and end of every HTML document. ALL other tags must be contained within the HTML tags.

Introduction to HTML

3. **HEAD** denotes the beginning and end of the Heading section of the document. It has a required element **TITLE**.
4. **TITLE** defines the title of the document. This is a required element.
5. **BODY** denotes the beginning and end of the **BODY** section of the document. This section is usually much larger than the Heading section, and must always follow it.

For more information regarding the use of any HTML tag visit the Web Design Group's site at <http://www.htmlhelp.com>. There are online and offline references available with complete definitions and usage examples.

Other Commonly Used Tags

Tags used in the HEAD section:

META is used to supply information such as a site description, keywords, and even the document author's name. Sample META tag usage would include:

```
<meta name="author" content="John Pozadzides">
<meta name="description" content="Short site description...">
<meta name="keywords" content="No, more, than, 20, keywords, go, here">
```

STYLE is used to define Cascading Style Sheets (this is an entire lecture on its own!)

SCRIPT is used to contain small programs that run within a web application.

Tags used in the BODY section:

The **BODY** of a document consists of multiple block level elements. If plain text is found inside the body, it is assumed to be inside a paragraph **P**.

Block level elements:

Block level elements are elements that imply a paragraph break, or the start of a new line. Some Block level elements are:

- H1...H6 - headings

Headings are used for relative emphasis of sub-areas of a document. The main heading for a page would be contained in `<H1></H1>`, sub-headings off the main would be in `<H2></H2>`, subheadings within it would be `<H3></H3>`, and so on. Rarely would an author ever get all the way down to an `<H5>` or `<H6>`.

- P - paragraphs

The Paragraph tag denotes the beginning and ending of a paragraph of text. It should be noted that the paragraph is the default for text that is not marked up. In other words, if an author mistakenly includes a line of text without any tags, it will be assumed to be in a Paragraph.

- UL - unordered list

Introduction to HTML

An unordered list will create a bulleted list of all items contained within. A sample unordered list would look like:

```
<UL>
  <LI>Walk the Car
  <LI>Polish the Dog
  <LI>Wash the Cat
</UL>
```

- OL - ordered list

An ordered list will create a numbered list of all items contained within. A sample ordered list would look like:

```
<OL>
  <LI>Walk the Car
  <LI>Polish the Dog
  <LI>Wash the Cat
</OL>
```

- CENTER - centers all enclosed elements

The center tag is a block level element that can contain multiple other elements and provide for them to be centered on the page. A sample of the center tag in action would be:

```
<CENTER>
  <IMG SRC="foo.gif">
  This is a picture of Foo.
  <A HREF="mailto:foo@mail.com">Send mail to Foo.</A>
</CENTER>
```

It should be noted that if only a Heading or a Paragraph needs centering, they have a built in attribute, which may be assigned to them to take care of this. A sample of a centered level 1 heading would look like:

```
<H1 ALIGN=CENTER>Heading goes here</H1>
```

- HR - horizontal lines

The HR tag will insert a horizontal line anywhere it is used. The thickness of the line may be controlled with the attribute *SIZE="#*" and the width of the tag may be controlled with the attribute *WIDTH="##%"*. A sample horizontal line statement could be:

```
<HR SIZE="4" WIDTH="75%">
```

- TABLE - contains rows and columns

Tables are often used to present data in a spreadsheet format, where rows of data need to be maintained in neat columns. Tables are a bit more complex, but you can find more information on them at <http://www.htmlhelp.com>.

Text level elements:

Text level elements are contained within Block level elements and are used for non-breaking markup such as:

- I - italics

Displays all text within in Italics. This tag is not used merely for emphasis. That is what the EMPHASIS tag is for.

Introduction to HTML

- B - bold
Displays all text within in bold type. This tag is not used merely to emphasize text, that is what the STRONG tag is for
- EM - emphasis
Used to emphasize enclosed text.
- S - strong
This tag is used to mark up sections of text that should be displayed with STRONG emphasis.
- BIG - larger text
Will increase the font size by one size increment. This tag may be nested with itself to create incrementally larger sizes of text.
- SMALL - smaller text
Will decrease the font size by one size increment. This tag may be nested with itself to create incrementally smaller sizes of text.

Using Anchor Elements in HTML

Anchors are used to create hyperlinks to other documents or media within Web pages. The A element denotes an *anchor*--a hypertext link or the destination of a link.

```
<A HREF=" " >...</A>
```

The HREF attribute specifies a hypertext link to another resource, such as an HTML document or a JPEG image. Examples:

```
<A HREF="album.html">My photo album</A>  
  
<A HREF="./images/me.jpg">Picture of me</A>  
  
<A HREF="mailto:liam@htmlhelp.com" TITLE="Feedback on HTML 4.0  
Reference">liam@htmlhelp.com</A>
```

The value of the HREF attribute is the URI of the link. The TITLE attribute can be used to briefly describe the contents of the link and is rendered as a "tooltip" by some visual browsers. With mailto links, some browsers use the TITLE attribute value as a subject for the e-mail message.

The content of an A element used as a link should be as context-free as possible. In other words, a user should be able to pull all A elements from a document and still have an idea what lies behind each link. Link text that contains **Click here** or simply **here** is extremely bad form.

```
<A NAME=" " >...</A>
```

The NAME attribute defines a destination for a link. For example, a document containing

Introduction to HTML

```
<H1><A NAME="foo">My Heading</A></H1>
```

defines a link destination named "foo" at the indicated heading. One could then use HREF="#foo" in an A element within the same document or HREF="somedoc.html#foo" from within another document.

An A element cannot contain another A element, so one must be careful that named anchors do not contain link anchors. Authors can use both the NAME and HREF attributes in a single A element to avoid this problem.

HTML 4.0's ID attribute is intended to eliminate the need for A NAME. The ID attribute can be used with almost any element to define a link destination, so that the following could be used in place of the previous example:

```
<H1 ID=foo>My heading</H1>
```

However, browser support for ID link destinations is very poor, so A NAME will be needed for quite awhile.

NAME and ID values must be unique in any document, and different values must differ by more than just the case. Values must begin with a letter in the range A-Z or a-z, and may be followed by A-Z, a-z, 0-9, hyphens, underscores, colons, or periods. When linking to a named anchor, the name *is treated as case sensitive*.

Good Web Design Principles to Remember

- Never use heading tags (e.g., <h4>, <h6>) to achieve a formatting effect.
- Never use "click here" because not everyone uses a mouse to follow links! Instead make links seem incidental, a phrase like "select this link" is preferable to "click here."
- E-mail links should also include a visible mailing address as an alternative for browsers that may have trouble with the mailto: feature.
- Break pages into useable sizes. Nobody likes to scroll and scroll with a topic that seems to ramble on and on.
- When referring to a sites URL, include the trailing slash so the server doesn't need to redirect requests. (".../~johnpoz/" vs. ".../~johnpoz").
- If you link to a file always give the size of the download.

HTML Validation

Why Validate My HTML?

One of the important maxims of computer programming is:

'Be conservative in what you produce; be liberal in what you accept.'

Browsers follow the second half of this maxim by accepting Web pages and trying to display them even if they're not legal HTML. Usually this means that the browser will try to make educated guesses about what you probably meant.

Introduction to HTML

The problem is that different browsers (or even different versions of the same browser) will make different guesses about the same illegal construct; worse, if your HTML is *really* pathological, the browser could get hopelessly confused and produce a mangled mess, or even crash.

That's why you want to follow the first half of the maxim by making sure your pages are legal HTML. The best way to do that is by running your documents through one or more HTML validators.“

Two of the most reputable HTML validators can be found at:

- <http://validator.w3.org/>
- <http://htmlhelp.com/tools/validator/>

More Information

Once you've mastered this document there is much greater depth to be found at <http://htmlhelp.com/reference/html40/>.

Additionally, there are volunteers staffing the HTMLHelp.com Forums at <http://htmlhelp.com/forums/> who will happily help anyone making a genuine attempt to learn HTML.

Introduction to HTML

Good Web Design Principles

The first principle of Web authoring is to convey information to the reader about a particular topic. The manner in which it is conveyed should work as intended with all Web browsers. Remember, information is only useful if it can be interpreted. As a means to that end, here is a list of very important edicts to follow when designing for the Web.

- Don't recommend that a certain browser be used.
- If you use a background color, set all other colors so they don't interfere with default settings.
- Always Validate!
- Use consistent headers to remind users that they are at the same site. It's easy to get lost in Cyberspace.
- Construct pages that load quickly and are easy to navigate.
- Images should enhance the page rather than detract. If it doesn't do that, it shouldn't be there.
- Never use heading tags (e.g., <h4>, <h6>) to achieve a formatting effect.
- Never use "click here" because not everyone uses a mouse to follow links! Instead make links seem incidental, a phrase like "select this link" is preferable to "click here."
- Break pages into useable sizes. Nobody likes to scroll and scroll with a topic that seems to ramble on and on.
- Don't steal someone else's graphics or text!
- If you use background images, keep them small.
- Periodically check links to make sure they still work.
- Always include height and width attributes for images.
- When referring to a sites URL, include the trailing slash so the server doesn't need to redirect requests. (".../~johnpoz/" vs. ".../~johnpoz").
- If you link to a file or another image, always give the size of the download.
- Make the title very descriptive in case someone wishes to add the site to their hotlist.

It just so happens that strict adherence to these rules will not only benefit your visitors, they will help ensure excellent search engine results via Google and other major providers.