

# *CLARION* *5.5*

# **HTML Basics**

**COPYRIGHT 2000 by SoftVelocity Inc.  
All rights reserved.**

This publication is protected by copyright and all rights are reserved by SoftVelocity Inc. It may not, in whole or part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from SoftVelocity Inc.

This publication supports Clarion 5.5. It is possible that it may contain technical or typographical errors. SoftVelocity Inc. provides this publication “as is,” without warranty of any kind, either expressed or implied.

**SoftVelocity Inc.**

2769 East Atlantic Boulevard  
Pompano Beach, FL 33062  
Phone: 954-785-4555  
Fax: 954-946-1650  
E-mail: [info@softvelocity.com](mailto:info@softvelocity.com)

**Trademark Acknowledgements:**

Clarion™ is a trademark of SoftVelocity Inc.

Microsoft® Windows®, Windows 95®, Windows 98®, Windows NT® , and Windows 2000® are registered trademarks of Microsoft Corporation.

Portions Copyright © 1994-2000 [W3C](#)® ([Massachusetts Institute of Technology](#), [Institut National de Recherche en Informatique et en Automatique](#), [Keio University](#)), All Rights Reserved.

All other products and company names are trademarks of their respective owners.

# Contents

<b>HTML BASICS</b>	<b>4</b>
<b>Introduction</b>	<b>4</b>
The Clarion Connection .....	4
<b>HTML Basics</b>	<b>6</b>
Tags .....	6
Document Structure .....	7
<b>Headings .....</b>	<b>7</b>
<b>Horizontal Ruled Lines .....</b>	<b>8</b>
<b>Paragraphs .....</b>	<b>8</b>
<b>Text Formatting Properties .....</b>	<b>8</b>
<b>Alignment Attributes .....</b>	<b>8</b>
<b>The Line Break .....</b>	<b>9</b>
<b>Preformatted Text .....</b>	<b>9</b>
<b>Links and Images .....</b>	<b>10</b>
<b>In-line Images .....</b>	<b>10</b>
<b>Image Links.....</b>	<b>11</b>
<b>Other Miscellaneous tags .....</b>	<b>11</b>
<b>HTML Maintenance .....</b>	<b>12</b>
<b>HTML Comments .....</b>	<b>12</b>
<b>HTML Lists .....</b>	<b>12</b>
<b>More Fonts and Colors .....</b>	<b>14</b>
<b>Navigation within the HTML Document .....</b>	<b>15</b>
<b>Advanced Techniques with HTML Graphics .....</b>	<b>15</b>
<b>Tables .....</b>	<b>17</b>
<b>Adding Sounds to HTML Documents .....</b>	<b>18</b>
<b>Using Frames in HTML .....</b>	<b>19</b>
<b>Defining Frame Columns .....</b>	<b>20</b>
<b>Defining Frame Rows .....</b>	<b>21</b>
<b>The TARGET attribute .....</b>	<b>22</b>
<b>Meta Tags .....</b>	<b>22</b>
<b>Auto-refreshing .....</b>	<b>23</b>
<b>Forms .....</b>	<b>23</b>
For futher reading .....	26
<b>INDEX</b>	<b>27</b>

# HTML BASICS

## Introduction

The Internet has revolutionized the computer and communications world like nothing before. The combined technologies of the telegraph, telephone, radio, and computer have set the stage for this unprecedented integration of capabilities. The Internet, simply defined, is a Global Information Infrastructure. It combines:

- ◆ A world-wide low-cost broadcasting capability
- ◆ A mechanism for information dissemination
- ◆ A medium for collaboration and interaction between individuals and their computers without regard for geographic location.

The Internet is also a model case-study of the benefits of long term investment and commitment to research and development of an information infrastructure. From the dawn of packet switching, the government, industry and universities have been partners in evolving and deploying this exciting new technology. Today, terms like [KewlGuy@domain.com](mailto:KewlGuy@domain.com) and <http://www.softvelocity.com> are now proliferating into every kind of media imaginable.

## The Clarion Connection

---

When first released in version 4, there was only one set of templates to make an Internet enabled application, called *Internet Connect* (aka *CWIC*). This template could be used with ABC or the Legacy template set. It used many client-side Java applets.

Later with the new Wizatron technology, the idea about using a set of classes that can build a web-enabled application without Java emerged. This went under the name of *iBuild*. iBuild was never released as it was discovered that the design and implementation would not work.

However, much of the work that did go into iBuild, was later converted to a series of object classes and a new template set called, *Web Builder*. It is also Java free. Many of the iBuild designs are in the Web Builder classes.

This approach offers some distinct advantages:

- ◆ No Java.
- ◆ Everything is in classes. This allows easy updating without

breaking existing code.

- ◆ ABC only.
- ◆ 32-bit only.
- ◆ Fast (connection speed and net traffic are the only speed bumps).
- ◆ Everything is contained in the APP file.
- ◆ Those using CWIC will find many of the interfaces and embeds are still there. Learning curve is kept small.
- ◆ A few other new features, dicussed later.

You still build your applications as you always have. However, if you plan on deploying your application to the web, you must keep some addition design considerations in the back of your head.

This document covers some of these issues. These templates and technology are new for many developers.

One of the common questions that comes up is how does Clarion maintain your web site? *It doesn't*. That is *not* the purpose of these templates. They are there to *enable* an application to run *on* a web site. That is a big difference bewteen this technology and those that build and maintain web sites. There are many web site tools. Use the one that fits your skills and designs the best.

Another common question is what type of server - can it run under Linux? Currently, you need a web site that runs under Windows NT/2000. If you have a Linux (or other non-NT) web server, you need to point to an NT server to run a Clarion web-enabled application. While non-NT web servers may be in the cards, you *must* run Clarion web-enabled applications on Windows NT/2000. The *development* of such applications can be on any Windows9x or better environment. You then deploy the application (and supporting files) to the NT server.

SoftVelocity has published documents on the exact steps you need to do this. There are other documents targetting areas you need to study.

This document covers only HTML and the basics. It is intended for those not familiar with HTML to ease them into this language. When you make Web applications, some HTML is required as there are embeds for HTML code in your applications.

This document is not a complete reference for HTML, nor will it ever be one. The sole intention is to introduce you to the language so that it is not an unknown factor in your designs.

References for more data and tutorials are mentioned at the end of this document.

## HTML Basics

HyperText Mark-Up Language, or **HTML**, is the default formatting language used in nearly all WWW documents. The Clarion Web templates create HTML files at runtime when the client connects to the server.

These HTML files are removed from the server when the client terminates the connection. There are available code embed points for every control in your web-enabled application with the express purpose of embedding HTML. Using template derived properties, you can place conditional code that only executes when your Web Server is active.

For these reasons stated above, it is critical for all Clarion developers to quickly familiarize themselves with the HTML Basics. We will begin with the fundamental syntax rules, and add additional functionality in the following pages.

**Note:** The Web templates will eliminate the need to use a lot of custom HTML code in your applications. There will be occasions where its use can come in handy.

## Tags

---

HTML documents look a lot like word-processing documents. You can have **bold** and *italicized*, **Larger** and Smaller, or `type-written` text. The HTML code for this looks a little like this:

```
You can have <b>bold</b> and <i>italicized</i>, <font size=+2>Larger</font> and <font size=-2>Smaller</font>, or <tt>type-written</tt> text.
```

Notice the “<” and “>” characters. When you place text and symbols within these brackets you are creating a *tag*. For example the <b> tag is saying to start bold text, and the </b> tag is saying to stop bold text. The tag with the slash (/) is known as the *closing tag*. Many opening tags require a following closing tag, but not all do. Tags make up the entire structure of an HTML document.

```
<b>This Text is Bold</b> ^^^-Opening Tag ^^^-Closing Tag
```

Here are two pieces of HTML code, the second of the two has an error in it, what is it?

```
#1 - Bob jumped OVER the fence.
#1 - Bob jumped <b>OVER</b> the fence.
#2 - Bob jumped UNDER the fence.
#2 - Bob jumped <b>UNDER<b> the fence.
```

Did you notice that the second code is missing a slash (/) in the tag after the word UNDER, which causes the web browser to interpret the code as leaving the bold face on! This is a common error, so be careful of it!

## Document Structure

---

HTML files are just normal text files... they usually have the extension of .htm, or .html. HTML documents have two (2) parts, the head and the body. The body is the larger part of the document, as the body of a letter you would write to a friend would be. The head of the document contains the document's title and similar information, and the body contains most everything else.

Example of basic HTML document Structure...

```
<html>
<head><title>Title goes here</title></head>
<body>Body goes here</body>
</html>
```

**Note:** Extra spaces and line breaks (blank lines) are ignored when the HTML is interpreted... so add them if you wish to do so.

Whatever falls between the TITLE tags will be the title of the document, when the page is viewed. It is found in the title bar on your browser window. [Note: You may NOT use other tags within the TITLE tags (Example: You cannot have the code read: <title><b>title goes here</b></title>.)]

Example of how titles are viewed...

In Netscape Navigator...

Netscape - [Title goes here] OR Title goes here - Netscape [depending on version]

In Microsoft Internet Explorer...

Title goes here - Microsoft Internet Explorer

Whatever you place between the BODY tags will fall into the major area of the document window, and therefore it is the largest part of your HTML document.

### Headings

Headings are some of the most important tags within the BODY of your HTML document. You will usually use a heading to tell what the following section of your page is about. Think of the heading as those used in most word processors. The opening tag for a heading is <h> and the closing tag

is `</h>` with **y** being the size of the heading... from 1 to 6. (1 being largest, and 6 being smallest)

## Horizontal Ruled Lines

Horizontal Ruled Lines are used to separate different areas of a web page. The tag for a horizontal ruled line is `<hr>`. The horizontal ruled line DOES NOT have a closing tag. You may also add certain attributes to the `<hr>` tag, such as:

WIDTH=*n* (for fixed pixel width)

WIDTH=*n*% for a certain percentage of the screen wide

SIZE=*n* to make the line a certain pixel amount thick

NOSHADE to turn the line's shading off.

A plain `<hr>` with no attributes will make the line the full width of the screen. **Note:** Since CWIC does not support line controls, use `<HR>` in the embedded code areas. (more on this later)

## Paragraphs

You will often use paragraphs in HTML, just as you do when you write standard documents. The opening tag for a paragraph is `<p>`, and the closing tag is `</p>`. The closing tag for a paragraph is not always needed, but recommended to make your HTML more readable.

## Text Formatting Properties

If you had an entire web page without formatted text, it would look rather dull and boring. This is why we use text-formatting tags. Some common text formatting tags are:

`<b>` and `</b>` for bold

`<i>` and `</i>` for italics

`<u>` and `</u>` for underlined

The `<font size=n>` and `</font>` tags are also useful.

## Alignment Attributes

Many tags support three types of alignment:

Left Align

```
<h1 align=left>Left Align</h1>
```

Center Align

```
<h1 align=center>Center Align</h1>
```

## Right Align

```
<h1 align=right>Right Align</h1>
```

Note that the ALIGN attribute is placed inside the opening tag before the '>'.

## The Line Break

An HTML document will normally word-wrap at the end of a line. If you want to have the text break (go to another line) you will use the <br> tag. This tag has no closing tag. This tag is similar to the “soft return” in a word processing doc.

## Preformatted Text

If you wish to have text line up properly (fixed width text) that will include line breaks without the use of the <br> you may find the <pre> and </pre> tags helpful. This is especially useful when extra line breaks are necessary.

Example of text WITHOUT pre-formatting:

### HTML Source

```
The cat ran after the dog. ^ ^-verb ^noun ^-noun
```

### Browser Output:

```
The cat ran after the dog. ^ ^-verb ^noun ^-noun
```

HTML ignores extra line breaks, so the text above does not line up properly.

Example of text WITH pre-formatting:

### HTML Source:

```
<pre>
The cat ran after the dog.
    ^   ^-verb       ^noun
    ^-noun
</pre>
```

### Browser output:

```
The cat ran after the dog.
    ^   ^-verb       ^-noun
    ^-noun
```

## Links and Images

Links are probably one of the more important HTML tags to understand. Without them, navigation from one HTML document to another would be difficult (if not impossible). Links are what adds the “hyper” to hypertext! To add a link, use the `<a href="location">` opening tag and `</a>` closing tag.

This is called an *anchor* tag. The `<a>`/`</a>` are the opening and closing tags. The anchor tag has attributes and *href* is one of them.

Whatever appears between these two tags will become underlined and colored, and if you click on the underlined text it will send the browser to the location within the quotes.

Example HTML:

```
Visit <a href="http://www.finatics.com/">Jim's Dolfan Site</a>!
```

Browser Output:

Visit Jim's Dolfan Site!

Although Links are usually used to send people to other web pages, you may also use it to send email to a specific address by using a location of **mailto:user@host**.

Example HTML:

```
Send email to <a href="mailto:jim-d@topspeed.com">WebMaster Jim</a>!
```

Browser Output:

Send email to WebMaster Jim!

## In-line Images

You may also add images (pictures) to your web page, as long as the image is in the .gif or .jpg (or .jpeg) file formats.

The basic tag for in-line images in ``.

The IMG tag has no closing tag. It is recommended to add HEIGHT and WIDTH attributes to the IMG tag, which will allow the image to take proper proportions on a browser that is not currently viewing images. It is also recommended to use **ALT="what picture is"** to tell a person what picture is active in case it is still loading or they are not able to view the image.

HTML example of a basic in-line image...

```

```



### Image Links

Many times you may want to have an image that is linked to another HTML document. You only need to place the IMG tag within the A HREF tags. You may also use the ALIGN tags with images.

HTML example:

```
<a href="http://www.westie.com/"></a>  
<a href="http://www.westie.com/"></a>
```

### Other Miscellaneous tags

There are additional tags available for

- ◆ Center Alignment <center> text</center>

· BODY attributes

- ◆ Background Image BACKGROUND="location\_of\_image"
- ◆ Background Color BGCOLOR="#hex color code here"
- ◆ Color for Links LINK="# hex color code here"
- ◆ Color for Links Visited VLINK="# hex color code here"
- ◆ Color Code for Text Color TEXT="# hex color code here"

**Tip:** If you specify one color on your page, you should specify all colors.

## **HTML Maintenance**

As with good Clarion programming, there are a few rules in HTML programming that you should follow to avoid unnecessary problems and to enhance the overall clarity.

1. Avoid adding special characters into your code, but instead type their numeric equivalent.
2. Many characters should NEVER be typed directly into HTML code. (example: "<", ">", "@", "&", and " itself. Use the following format instead:

**&escape\_code;** (Ampersand, Escape Code for Character, then a semicolon).

3. Use quotes around values in attributes... For example, if you want a horizontal rule that is half of the screen width, type `<hr width="50%">` rather than `<hr width=50%>`, or if you want one that is size 5 type `<hr size="5">` rather than `<hr size=5>`.
4. Don't overlap tags. Overlapping occurs when Tag A starts, Tag B starts, Tag A closes, then Tag B closes. This will cause errors in sensitive browsers (Netscape purposefully built error checking into the browser so developers could catch errors)

Which line contains overlaps?

- a. `<font size=+1><b>This is Bold and One Font Size Bigger</font></b>`
- b. `<font size=+1><b>This is Bold and One Font Size Bigger</b></font>`
- c. `<a href="here.html"><i>This link is italicized</i></a>`
- d. `<a href="here.html"><i>This link is italicized</a></i>`

5. Test in more than one browser! Your users can have a favorite browser, but the developer should have "both".

## **HTML Comments**

Most programming languages contain a symbol to designate remarks or comments to help document a difficult section of code. HTML is no exception. Comments will not appear in a web browser when the page is displayed, and it is only visible when the source code is viewed.

HTML example:

```
<!--This text should not appear in your browser-->
```

## **HTML Lists**

There are three types of HTML Lists.

### **1. Unordered**

The unordered list is probably the most common list you will use.

Example:

•pencils •pens •erasers •paper •glue

Note the bullet before each list item. Here is the HTML Code for the list shown above:

```
<ul>
<li>pencils</li>
<li>pens</li>
<li>erasers</li>
<li>paper</li>
<li>glue</li>
</ul>
```

The <ul> tag begins the unordered list. It is closed with the </ul> tag. Between these two tags you place LIST ITEMS, each one having an individual <li> opening tag, and an optional </li> closing tag. There is no limit to the number of List Items you may have in a single list.

## 2. Ordered

The ordered list is also known as the numbered list. The opening tag for the list is <ol> instead of <ul>, and the closing tag is </ol> instead of </ul>. List Items within the list still use the same tags.

Example:

1.pencils 2.pens 3.erasers 4.paper 5.glue

HTML Code for the ordered list above:

```
<ol>
<li>pencils</li>
<li>pens</li>
<li>erasers</li>
<li>paper</li>
<li>glue</li>
</ol>
```

## 3. Definition

The definition list is a little more complicated, but still is very easy to create. The list starts with a <dl> opening tag, and ends with a </dl> closing tag. Two other tags are used: <dt> for Definition Term, and <dd> for Definition. These two tags do not need closing tags.

Example of a Definition List:

Merit

The organization that initially managed NSFNET.

MILNET

The Department of Defense network.

MIME

An extension to Internet mail that allows for the inclusion of non-textual data such as video or zip files in e-mail.

Here is the HTML code for the definition list on the previous page:

```
<dl>
<dt>Merit
<dd>The organization that initially managed NSFNET. The organization
that initially managed NSFNET.
<dt>MILNET
<dd>The Department of Defense network.
<dt>MIME
<dd> An extension to Internet mail that allows for the inclusion of non-
textual
data such as video or zip files in e-mail.
</dl>
```

## More Fonts and Colors

The newest version of many browsers support extended fonts, in which you can choose to have the document fonts be other than the normal one. This is accomplished by adding the `FACE="font_name"` attribute to the `<FONT>` tag. The most commonly supported fonts are Arial, Helvetica, Impact, Comic Sans MS, (IE3.0) and a few others. It is not recommended to make your page font dependent, because the older versions of many browsers don't support this feature and the new HTML 4.x standard recommends you use as little as possible.

Examples:

```
<font size=+2 face="Arial">Arial</font>
Arial
<font size=+2 face="Helvetica">Helvetica</font>
Helvetica
<font size=+2 face="Impact">Impact</font>
Impact
<font size=+2 face="Comic Sans MS">Comic Sans MS</font>
Comic Sans MS
```

You can also change the color of the text by setting the `COLOR="font_color"` attribute in the `<FONT>` tag. The Color is usually set by using the hexadecimal system (#000000 black to #FFFFFF white) but can also be set in newer browsers by using the simple word of the color. (Red for Red, Blue for Blue, etc.)

Examples:

```
<font color="Blue">This should be Blue!</font>
This should be Blue!
<font size=+2 face="Impact" color="Green">Green Impact Font!</font>
Green Impact Font!
```

## **Navigation within the HTML Document**

It is possible (and very convenient in large HTML documents) to be able to click on a link and move to another area within the same page. Use the normal anchor tag (<A HREF>) and reference a named portion of the document, instead of placing another page in the quotes. To name the part of the document, go to the area you want to name, and place <a name="name\_of\_area">text</a>. To call a link to that place from somewhere else in the document, use <a href="#name\_of\_area">text</a>

Example:

```
<a href="#hotsection">Go To Hot Section</a><br>
text<br>
more text<br>
etc.<br>
etc.<br>
etc.<br>
etc.<br>
<a name="hotsection">Welcome to the Hot Section!</a>
```

Once you have the section identified, you can call it from other documents... for example, if you named a section in index.html called section2, you could call it from bookmarks.html using <a href="index.html#section2">. The "name" shown above is also known as a bookmark. Think of this as a target to which a link will jump.

You can also add a "MIDPAGE" anchor to your hyperlink, allowing a user to jump to a specific area on another page.

Example:

```
<a href="http://www.finatics.com/jim.htm#MIDPAGE">
```

## **Advanced Techniques with HTML Graphics**

This section covers a variety of topics associated with HTML graphics

### **1. Aligning images with text**

Use the ALIGN attribute for precise positioning of text and graphics

For example, to have the text start at the top next to the image:

```
SoftVelocity, Inc.<br>
Mastery Class
```



SoftVelocity, Inc.

### Mastery Class

If you want a horizontal line to fall below the image, you should add the `<br clear=all>` tag between the image and the horizontal rule. This is also useful in other instances, such as when you want the text to fall below the image.

## 2. Transparent GIFs

GIF files support a feature called transparency, in which one color out of the 256 colors is set to be transparent. If the background color of the file is transparent, this will let the web page's background show through. In many graphics programs you will find an option to save the background color as transparent, which allows you to make your own transparent gifs.

## 3. Interlaced vs. Non-Interlaced

When a graphic is downloaded into a web page, it is loaded one of two ways. Interlacing loads every other line the first time, then the other lines during the second pass. Non-interlacing loads each line in sequential order.

Each type takes the about same time to load, so it is more or less an aesthetic decision. This feature is not determined by a tag, but by the format you saved the file in.

## 4. Animated GIFs

GIF version 89a supports an advanced technology that allows animation. Inside a single .gif file many picture frames are stored, along with an index telling how long each frame should be displayed. Another option can be stored to determine if they should repeat after the sequence is complete.

If you wish to create your own animated gifs, several software packages are available, like the Gif Construction Set from Alchemy Mindworks and the MS GIF animator (free). Animated .gifs are treated as a standard image files, so they are loaded with the standard `<IMG>` tag.

## 5. A summary reference of the standard IMG tag

Here is the complete syntax of the IMG tag and associated attributes:

```

Name Of File-^      pixels-^ pixels-^ border-^ text-^
                wide      deep      size      description
```

The alt tag should always be used, to allow the users to view “something” while the image is loading.

## Tables

A table, simply defined, is an organized data array. Tables are used to make data important to the user easier to locate and analyze.

The next few paragraphs document how an HTML table is constructed. By using the Internet Connect, you probably will never have a need to create your own tables. It is nevertheless useful to know how tables work.

Tables are normally one of the more challenging things to code with HTML. They start with the <table> tag, and usually contain a border=n attribute within the opening tag. If border=0, then the table’s border is invisible. A missing border attribute will make the table border invisible. This can be particularly useful when you want to align text in rows and columns, but don’t want a table border around it. The lower the border number, the thinner a border will appear. A table must end with a </table> tag.

### Example of HTML Tables:

```
<table border=4>
<tr>
<th>What are tables used for?</th>
</tr>
<tr>
<td>Tables are used to make data easier to interpret or to just give your
document more impact.</td>
</tr>
</table>
<table border=0>
<tr>
<td>This table has a border of 0.</td>
</tr>
</table>
<table border=3>
<tr>
<td>This table has a border of 3.</td>
</tr></table>
```

Each row within the table is defined by the opening <tr> tag and the optional </tr> closing tag. Within each table row are table cells, which are defined by the <td> opening and </td> closing tags. Most table rows contain more than one cell. Many times, you will need a heading for a column of cells of the first row. To do this, you will use the <th> opening and </th> closing tag. The table heading tag automatically makes the text in that cell bold and centered.

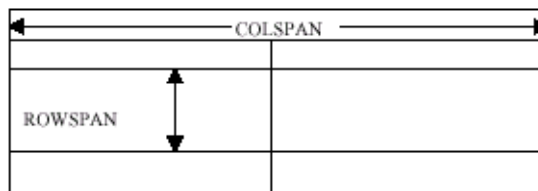
Example:

```

<table border=2>
<tr>
<th>Heading A</th><th>Heading B</th><th>Heading C</th>
</tr>
<tr>
<td>Cell A</td><td>Cell B</td><td>Cell C</td>
</tr>
<tr>
<td>Cell D</td><td>Cell E</td><td>Cell F</td>
</tr>
</tr>
</table>
<table border=2>
<tr>
<th>Heading A</th><th>Heading B</th><th>Heading C</th>
</tr>
<tr>
<td rowspan=2>Cell A & D</td><td>Cell B</td><td>Cell C</td>
</tr>
<tr>
<td>Cell E</td><td>Cell F</td>
</tr>
</tr>
</table>

```

Notice the *rowspan* attribute above. This will allow that cell to span two rows. If you need a cell to span more than one column, use the *colspan*=*n* attribute. Also, you may wish to use the *ALIGN* and *VALIGN* attributes to align the contents of cells. If you wish to change the horizontal alignment of the contents of a certain cell, add *ALIGN*=*LEFT*, *ALIGN*=*CENTER*, or *ALIGN*=*RIGHT* to the opening *<td>* tag. If you wish to change the vertical alignment of the contents of a cell, use the *VALIGN*=*TOP*, *VALIGN*=*MIDDLE*, or *VALIGN*=*BOTTOM* attributes. You may also want to try out the *WIDTH*=*n*% attribute, to change the width of a table or a cell.



## **Adding Sounds to HTML Documents**

The most compliant way to add a background sound to your HTML documents is with the *<BGSOUND>* Tag. The *SRC* attribute indicates the physical location of the file on the server. The *LOOP* attribute determines the number of times that the audio file is played. *LOOP*=*INFINITE* will play the file repeatedly. Valid sound files are .WAV, .MID, or .AU formats.

Example HTML:

```
<BGSOUND SRC='JAZZ.WAV' LOOP=INFINITE>
```

The <BGSOUND> tag is only compliant with Internet Explorer, Mosaic and compatible browsers. If you are using Netscape, its unique <EMBED> tag can also be used to play .wav, .mid, and .au files. Here is the basic structure of the tag:

```
<EMBED SRC="filename.ext" width=x height=x autostart=x hidden=x loop=x
volume=x>
```

If you want the file to automatically play when the web page is loaded, set autostart=true. If you want the control box to be hidden, use the hidden=true attribute. If you want repeat play, use the attribute loop=true. You can set the volume setting with the volume attribute, a value from 1 to 100 (default is 50).

You may use multiple embed tags within one page, but it is recommended to never set more than one on autostart. It is also never recommended to use .wav or .au files in the EMBED tag, because they tend to be very large and can take as much as few minutes to download. Remember that for compatibility with all major browsers, you will probably have to include both <BGSOUND> and <EMBED> tags.

## **Using Frames in HTML**

Frames are one of the later features of HTML, and are only implemented on newer browsers (Netscape 2.0 and higher, MS Internet Explorer, and many others). They simply allow a single browser window to be divided into multiple sections, each with an independent HTML page loaded inside it. These HTML pages can also interact with each other. Each page is loaded within each section of the frames window as a separate HTML document.

Example:

```
<html>
<head><title>Frame Test...</title></head>
<frameset cols="30%,70%">
<frameset rows="75%,25%">
<frame src="test.html" name="indexbar">
<frame src="http://www.topspeed.com/image/logo.gif">
</frameset>
<frameset rows="*">
<frame src="main.html" name="main">
</frameset>
</frameset>
<noframes>
This page requires a frame-capable browser... please get one!
</noframes>
</html>
```

The frame page itself normally does not actually contain any content. Instead, the content is contained on the separate HTML pages loaded within each frame. The frames page acts as a guide, defining which page to be loaded into each frame, and the frame attributes themselves. A frame page

closely resembles a normal **HTML page**, except that the **body** is **replaced with frameset**, and an additional **noframes** tag is added.

The FRAMESET tags are used to define the characteristics of the frames, and the NOFRAMES tags define what a not frames-enabled browser will see.

Because a non-frame browser ignores the FRAMESET tags, anything within the NOFRAMES tags will be considered a normal HTML page. The NOFRAMES content will not be seen by a frame-enabled browser unless they choose 'View >> Source' in the Browser Menu.

The FRAMESET tag will be used multiple times throughout a single frames page.

The first FRAMESET tag is used to define the number of separate frame columns within the browser window, and what width each of those windows will be. The next set of FRAMESET tags will be used to define how many frame rows will be in the browser window, and each one's height.

The row attribute is set separately for each column, for example, your first column may have 4 frames, and your second column may have 2 frames, etc. There is no specific limit on the number of frames you may have within a single browser window, but you must realize that your page will be viewed in different resolutions, from 640 x 480 pixels to 1024 x 768 pixels and greater. A good rule is to limit your page to no more than 4 frames within a single browser window.

### **Defining Frame Columns**

The first FRAMESET tag should use the following format:

```
<FRAMESET cols="SIZE_OF_COLUMN_1,SIZE_OF_COLUMN_2,ETC">.
```

This first tag will be closed with a </FRAMESET> tag, but only after the FRAMESET rows for each column are also defined.

The SIZE\_OF\_COLUMN can be one of three numbers:

- x% - each column is set by a percent of the available browser window. (x is a number from 1 to 100)
- x - each column is set by a fixed pixel amount. (not recommended, because pixel widths vary depending on the viewer's resolution) (x is any number)
- \* - the \* tells the browser to use all available space that is left for this column.

A FRAMESET reading <FRAMESET cols="20%,80%"> would mean that the first column is the browser window will take up 20% of the

browser window's width, and that the second column will take up 80% of the total browser's width. `<FRAMESET cols="120,*">` will set the first column to 120 pixels wide, and the second column takes up all remaining width. Only one column is required, and there is no limit on how many columns can be defined.

## Defining Frame Rows

Within the column defining the FRAMESET opening and closing tag are the row defining FRAMESET tags. The number of row defining FRAMESET tags is directly dependent on the number of column defined FRAMESET tags. For example, if there are two columns defined, there will be two separate row defining FRAMESET tags.

The row FRAMESET tag should use the following format:

```
<FRAMESET rows="SIZE_OF_ROW_1,SIZE_OF_ROW_2,ETC">.
```

The `SIZE_OF_ROW` is defined almost identically to the `SIZE_OF_COLUMN`... Rows are defined from top to bottom, and Columns are defined from left to right.

## Frame Attributes

- The simplest frame tag reads `<frame src="document_to_load.html">`, where **document\_to\_load** is the name of the web page that is to be loaded in that frame. Note that the frame tag has no closing tag.
- `SCROLLING="yes|no|auto"` - This defines if the frame will use a scroll bar. By default the frame sets scrolling to auto, which means the browser determines if a scroll bar is needed. If set to yes, the scroll bar is always present. If set to no, the frame will never have a scroll bar.
- `NORESIZE` - this attribute states that the user should not be able to resize the frame. By default the user is able to resize all frames within the browser window.
- `NAME="x"` - this attribute defines the name of the frame, which is used to target pages to be loaded in that frame.

Example:

```
<html>
<head><title>Testing Complex Frames</title></head>
<FRAMESET cols="33%,33%,33%">
<FRAMESET rows="*,100">
<frame src="page1.html" NAME="index">
<frame src="page2.html" NORESIZE>
</FRAMESET>
<FRAMESET>
<frame src="main.html" NAME="main">
</FRAMESET>
<FRAMESET rows="50%,50%">
```

```

<frame src="page3.html">
<frame src="page4.html" SCROLLING="no">
</FRAMESET>
</FRAMESET>
<noframes><body>
This page requires a frames-enabled browser!
</body></noframes>
</html>

```

## **The TARGET attribute**

If you have a document in one frame containing a link, but need the link to load information into a different frame that you have defined, use the TARGET attribute. The TARGET attribute is part of the <a href> tag, and is formatted as follows:

```
<a href="mypage.html" TARGET="target_frame">text</a>
```

**mypage.html** is the name of the file which should be loaded in the frame.

**target\_frame** is the defined name of the “target” frame

**text** is the anchored text of the link.

Linking without a target will load the page into the current frame. There are other target attributes which can be used:

- TARGET=“\_blank” – a link is loaded into a new blank browser window.
- TARGET=“\_self” – a link is loaded into frame that link was clicked in.
- TARGET=“\_top” – the link is loaded into current full browser window, and all frames disappear, leaving the new page to have the entire window.

## **Meta Tags**

A Meta tag is used to indicate special instructions for the HTML document. The tag should be placed into the heading of the HTML document. There are several uses for this flag. One is to improve search engine results when your page is located.

When a search engine finds your page, it needs to index it (that is, add it to its searchable database) with some information extracted from the page. Many search engines now support the <META> tags, which allow you to give keywords and a description to your page. This gives you more control over how your page will show up during a search, and will often cause more traffic to your page.

```

<meta name="description" content="page description goes here">
<meta name="keywords" content="keywords go here">

```

When a user searches a search engine that supports meta tags and they query a phrase (search for a keyword) related to your page, your page may show up in the list of results. Your page will be listed by its Title, and then underneath its title will be the first hundred or so characters of the description you placed in the meta tag.

It is recommended that you keep the description content to no more than 200 characters. Although the keywords content is not seen by the user when searched, it is recommended to keep this less than 1000 characters, because if you have more the search engine will either ignore the rest or delete you from the index. (Commas are not needed to separate keywords)

#### HTML Example:

```
<html>
<head>
<title>Jim Defabia's DolFan Page</title>
<meta name="description" content="Jim's tailgate gang at E12">
<meta name="keywords" content="tailgate Dolphins football party">
</head>
<body>
Tailgate stuff goes here
</body>
</html>
```

### Auto-refreshing

Meta Tags are also used to provide an auto refresh feature for your HTML documents. Once one page loads, you can set a certain number of seconds and then force the browser to load another page automatically. The basic syntax is as follows:

```
<meta HTTP-EQUIV=REFRESH CONTENT=#seconds;URL="http://www.nextpage.com/nextpage.html">
```

- URL is the page you want it to refresh to
- CONTENT is the number of seconds you want it to wait before refreshing.
- HTTP-EQUIV=REFRESH identifies this as the refresh meta tag

The CWIC templates use this refresh meta tag for any Splash procedures you have defined.

### Forms

Forms are online data entry sheets supported by most of the World Wide Web browsers. The Clarion Internet Connect, in most cases, will eliminate the need to create forms in embedded HTML. The Form Procedure in the CW Application Generator is converted by the Internet templates into the appropriate HTML format. For this reason, we will only focus on a few basics in this section.

Forms begin with <FORM> and end with </FORM>.

When the user clicks on the Submit button, the POST method sends the form content to the appropriate URL (POST is case sensitive):

```
<form method=POST action="mailto:username@domain.com">
```

There are countless Form Processor programs scattered around the Internet that translate the data posted from a form. Of course, using the Internet Connect, this step is eliminated, as data is updated directly into a data file located on the server.

Here are some other supported form attributes:

```
<input type="text" name="first name" size=20>
```

Similar to the entry field in Clarion.

```
<input type="hidden" name="whatever" value="whatever">
```

Not seen by the user, but is sent to the owner with name=value in the response. .

```
<input type="radio" name="whatever" value="whatever">
```

Similar to the radio buttons found in Clarion.

```
<input type="checkbox" name="whatever" value="whatever">
```

Similar to the checkbox found in Clarion.

```
<input type="password" name="whatever" value="whatever">
<textarea name="textarea" rows=3 cols=40>
```

Similar to the text box found in Clarion.

```
<select name="select" size= 1>
<option> Option #1
<option> Option #2
<option> Option #3
</select>
```

Similar to the Clarion Drop Down List

```
<input type="submit" value="Submit!">
```

This is the default button the users will click on to send the form in whatever way you have set up. You can change the value (text) to whatever you like.

```
<input type="reset" value="Clear This Form">
```

Clears all items in the form structure

Now that you know the commands, let's put it all together with the following HTML example:

```
<html>
<p>
<hr width=450>
<p>
<d1>
<form method=POST action=../../www.softvelocity.com/cgi-bin/class/
formmail.pl-subject=class
Feedback&recipient=class Feedback>
```

```

<input type=hidden name=subject value=Class Feedback>
<input type=hidden name=recipient value=reggen@softvelocity.com>
What is your name? <input type=text name=realname size=40><br>
What is your email address? <input type=text name=username size=30><br>
How old are you? <input type=text name=age size=5><br>
Where are you from? City: <input type=text name=city size=20> State:
<input type=text
name=state size=3><br>
<dt>How did you find out about our class?<br>
<dd><input name=find me type=radio value=friend> A friend <br>
<input name=find me type=radio value=web> The SoftVelocity Web Page<br>
<input name=find me type=radio value=mail> A mailing <br>
<input name=find me type=radio value=not known> I don't remember!<br>
<dt>How would you rate this class?
<select name=rating>
<option value=5 stars>5 Stars (This is awesome!)
<option value=4 stars>4 Stars (This is great!)
<option value=3 stars>3 Stars (This class is above average!)
<option value=2 stars>2 Stars (This class is OK, Had moments.)
<option value=1 star>1 Star (The instructor was over my head!)
<option value=0 stars> No Stars (The instructor should be shot!)
</select>
<dt>What other SoftVelocity classes have you attended?
<dd><input name=classes type=checkbox value=Mastery OOP> The Mastery OOP
class<br>
<input name=classes type=checkbox value=Mastery Templates> Mastery
Template class<br>
<input name=classes type=checkbox value=Mastery SQL> Mastery SQL
class<br>
<input name=classes type=checkbox value=Mastery Web> Mastery Web
class<br>
<input name=classes type=checkbox value=Mastery Wizatrons> Mastery
Wizatron class<br>
<input name=classes type=checkbox value=ReportWriter> Report Writer
class<br>
<input name=classes type=checkbox value=Foundations> Foundations
class<br>
<input name=classes type=checkbox value=Essentials> Essentials class<br>
<input name=classes type=checkbox value=none> None - I haven't been
anywhere<br>
<dt>Now, tell me what you think of this class. (Please be kind!)<br>
<textarea name=opinion rows=3 cols=40 wrap=virtual></textarea>
<p>
<input type=submit value=Send it on in!>
<input type=reset value=Forget it!>
</form>
</dl>
<p>
<hr width=450>
<p>
</html>

```

Loading the HTML code in the browser of your choice should produce an output similar to this:

What is your name?

What is your email address?

How old are you?

Where are you from? City:  State:

How did you find out about our class?

- ☐ A friend
- ☐ The SoftVelocity Web Page
- ☐ A mailing
- ☐ I don't remember

How would you rate this class?

What other SoftVelocity classes have you attended?

- ☐ The Mastery OOP class
- ☐ Mastery Template class
- ☐ Mastery SQL class
- ☐ Mastery Web class
- ☐ Mastery Wustan class
- ☐ Report Writer class
- ☐ Foundations class
- ☐ Essentials class
- ☐ None - I haven't been anywhere

Now, tell me what you think of this class. (Please be kind)

## For further reading

This document is designed to get one familiar with HTML code and some basic commands. It is not intended as a complete guide to HTML code.

The Education Dept. at SoftVelocity strongly recommends that you visit the World Wide Web Consortium. The URL is <http://www.w3c.org>. You can continue with their easy to follow tutorials start with <http://www.w3.org/MarkUp/Guide/>.

Also, <http://www.willcam.com/cmat/html/crossname.html> for a superb reference of commands and attributes.

# INDEX

<b>A</b>		
Adding Sounds to HTML Documents .....	18	
Advanced Techniques with HTML Graphics .....	15	
A summary reference of the standard IMG tag .....	16	
Aligning images with text .....	15	
Animated GIFs .....	16	
Interlaced vs. Non-Interlaced .....	16	
Transparent GIFs .....	16	
Align .....	8	
Alignment Attributes .....	8	
ALT .....	10	
anchor .....	10	
Auto-refreshing .....	23	
CONTENT .....	23	
HTTP-EQUIV .....	23	
URL .....	23	
<b>B</b>		
BODY attributes .....	11	
BACKGROUND .....	11	
BGCOLOR .....	11	
LINK .....	11	
TEXT .....	11	
VLINK .....	11	
<b>C</b>		
center .....	11	
<b>D</b>		
Defining Frame Columns .....	20	
Defining Frame Rows .....	21	
Document Structure .....	7	
<b>F</b>		
form attributes .....	24	
checkbox .....	24	
hidden .....	24	
password .....	24	
radio .....	24	
reset .....	24	
submit .....	24	
text .....	24	
Forms .....	23	
Frame Attributes .....	21	
NAME .....	21	
NORESIZE .....	21	
SCROLLING .....	21	
Frames .....	19	
frameset .....	19	
<b>H</b>		
Headings .....	7	
Horizontal Ruled Lines .....	8	
HTML .....	6	
HTML Comments .....	12	
HTML Lists .....	12	
Definition .....	13	
Ordered .....	13	
Unordered .....	12	
HTML Maintenance .....	12	
<b>I</b>		
Image Links .....	11	
img .....	10	
In-line Images .....	10	
<b>L</b>		
Line break - br .....	9	
Links and Images .....	10	
<b>M</b>		
Meta Tags .....	22	
<b>N</b>		
Navigation within the HTML Document .....	15	
NOSHADE .....	8	
<b>O</b>		
Other Miscellaneous tags .....	11	
<b>P</b>		
Paragraphs .....	8	
POST .....	24	
Preformatted Text .....	9	

**S**

SIZE ..... 8

**T**

Tables ..... 17

Tags ..... 6

Text Formatting Properties ..... 8

The Line Break ..... 9

The TARGET attribute ..... 22

**W**

WIDTH ..... 8