



RHI Consulting

Quick Reference Guide

HTML & CGI

***RHI*CONSULTING®**
Information Technology Professionals

RHI Consulting is...

...The world's leader in information technology staffing. Finding the right information technology support for your organization can be difficult. Experience, personality, technical skills and cost must be weighed and factored into your decision.

Through ProSCANSM, our specialized evaluation process, *RHI Consulting* guarantees that our consultants possess the knowledge and skills necessary to get the job done right. And, our InfinitySM  Advanced Technical Training Program assures that our consultants maintain leading-edge skills for the IT industry.

Call today to be connected with your nearest *RHI Consulting* location or contact us online for more information.

RHICONSULTING[®]

Information Technology Professionals

1-800-793-5533 U.S. • 1-800-268-1948 Canada

www.rhic.com

TABLE OF CONTENTS

This quick reference guide has been developed to provide programmers and developers with "at-your-fingertips" information on creating dynamic Web pages using HTML and CGI scripting.

Since its creation, the World Wide Web has become one of the most important aspects of the Internet, experiencing phenomenal growth and development over the past few years. To enhance the static nature of HTML documents, CGI was developed as a standard communications protocol between Web clients and server-based applications.

This guide explains the various scripting elements and specifications available for HTML Level 2.0 and CGI/1.1. Please note that while the information contained herein is complete at the time of writing, World Wide Web specifications are constantly evolving. (*For the most current specifications, refer to the various online resources listed on page 32.*)

World Wide Web	2
Sample Web Page	3
CGI Overview	4
Passing Data to a CGI Script	5
CGI Environment Variables	5
HTTP Request Headers	7
Sample HTML/CGI Form Script	9
Returning Data from a CGI Script	10
Server Directives for Parsed Headers	10
HTTP Response Headers	11
Sample CGI Response Script	13
HTML Elements	14
Online Resources	32

© Robert Half International Inc. All registered trademarks, trademarks, and service marks mentioned in this guide are the property of their respective companies.

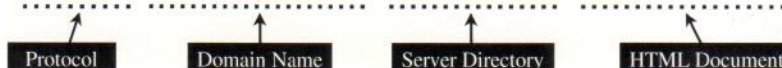
World Wide Web

The **World Wide Web** (the "Web") was created in 1989 at the European Laboratory for Particle Physics (CERN) as a way of easily accessing cross-referenced documents that existed elsewhere on the Internet. This was accomplished by creating **Hypertext Markup Language (HTML)**. HTML is a simple scripting language that is a subset of **Standard Generalized Markup Language (SGML)**. **HTML documents** (or "**Web pages**") contain hyperlinks that load a requested file from the server where it resides to the local client. Programs known as **Web Browsers** were then developed to graphically display HTML documents.

The World Wide Web itself is driven by **Hypertext Transfer Protocol (HTTP)** which manages the hypertext links that are used to navigate the Web. This protocol ensures that the various types of information contained in Web pages (text, graphics, hyperlinks) are correctly processed from a Web server to a Web client and that they are displayed in the correct format. Web browsers such as **Netscape** and **Mosaic** rely primarily on HTTP to process and display Web pages and content. Other primary Internet protocols include **Finger**, **FTP**, **NNTP**, **SMTP**, **TCP/IP**, and **WhoIs**.

Every Web document has a unique address known as **Uniform Resource Locator (URL)**. Each URL begins with a protocol, followed by either the unique domain name for the Web server or its **Internet Protocol (IP)** address. A specific file name can also follow the domain name or IP address. If a file name is not specified, the requesting Web server automatically looks for a file named **INDEX.HTML** in the root or specified directory. If **INDEX.HTML** does not exist, the Web server displays a list of files that exist in the root or specified directory.

Sample URL: `http://www.sandia.gov/sci_compute/html_ref.html`

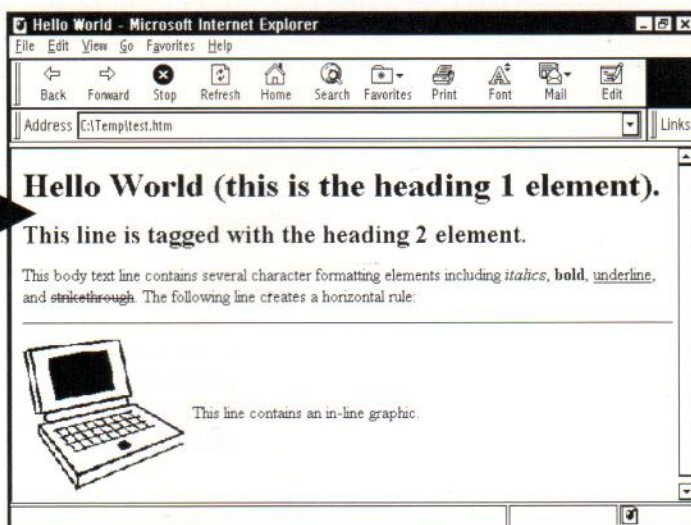


Sample Web Page

HTML Source Document

```
<HTML>
<TITLE>Hello World</TITLE>
</HEAD>
<BODY>
<H1>Hello World (this is the heading 1
element).</H1>
<H2>This line is tagged with the
heading 2 element.</H2>
This body text line contains several
character formatting elements including
<I>italics</I>, <B>bold</B>, <U>underline</U>,
and <STRIKE>strikethrough</STRIKE>. The
following line creates a horizontal rule:<BR>
<HR>
<IMG SRC="laptop.gif" ALIGN=middle>This line
contains an in-line graphic.
</BODY>
</HTML>
```

Web Browser



The **Common Gateway Interface** (CGI) is a simple protocol used to communicate between Web pages and Web server-based applications. Its primary function is to initiate a Web server-based program, then pass and receive environment variables to and from it. A Web server-based application in this instance is referred to as a **CGI script** (or "**CGI program**") and can be designed to perform any of a multitude of functions. CGI scripts can be created with any programming or scripting language that can read environment variables, read from **standard input** (STDIN), write to **standard output** (STDOUT), and that is capable of running on your Web server. Common languages used to create CGI scripts include scripting languages such as **AppleScript**, **PERL**, and **TLC**, and programming languages such as **C**, **Basic**, and **Pascal**. Note that the terms CGI script and CGI program should not be confused with CGI protocols themselves.

CGI scripts are normally launched with HTML elements contained in a Web page, although CGI scripts can also be called directly by appending a script name to a URL. In addition, CGI scripts are usually placed within the **cgi-bin directory** on a Web server, with an extension of ".CGI". Most Web servers can be configured to automatically recognize files with this file extension that reside in the cgi-bin directory as CGI scripts. Consult your server documentation for complete details on this feature.

CGI scripts are commonly used to process information entered into HTML forms. Various HTML form elements are used to pass information that is entered into a form to CGI environment variables. These variables are then sent to a CGI script on a Web server. The CGI script performs some sort of action (e.g., a query) and sends a response back to the Web browser (e.g., the query results). CGI scripts can either send a response back to the requesting Web page or generate a new HTML document. (See *Sample HTML/CGI Form Script*, page 9, and *Form Elements*, page 20, for more information.)

Passing Data to a CGI Script

Data is passed to a CGI script using environment variables and standard input (STDIN). The specific mode in which data is passed is determined by the **METHOD="GET"** and **METHOD="POST"** attributes used with the HTML **<Form>** element. When **METHOD="GET"** is used, data is sent to the Web server using the **QUERY_STRING** environment variable. When **METHOD="POST"** is used, data is sent to the Web server using the **NAME** attributes that are assigned by the various HTML Form elements. Note that even when **METHOD="POST"** is selected, environment variables are still set. The **CONTENT_LENGTH** variable should always be checked since the Web server does not send an **end-of-file** notification at the end of standard input data. (See *Form Elements*, page 20, for more information.)

When data is passed to a Web server, it is sent as **URL-encoded name/pair values**, regardless of whether it was sent through the **QUERY_STRING** variable or through standard input. The URL-encoded name/pair values can be decoded manually or by using one of the many public domain routines available on the Web. (See *Online Resources*, page 32, for more information.)

In addition to the environment variable and standard input procedures, the command line is also used to pass data to a CGI script, but only in the case of an **ISINDEX** query (note that the command line is not used with HTML forms). In the case of an **ISINDEX** query, the Web server should search the **QUERY_STRING** environment variable for a non-encoded equal sign (=). If a non-encoded equal sign is found, the command line is not to be used. (See the **QUERY_STRING** listing under **CGI Environment Variables**, page 6, and the **ISINDEX** listing under **Document Structure Elements**, page 15, for more information.)

CGI Environment Variables

The following environment variables are set by the Web server whenever a CGI script is executed:

Variable	Contains
GATEWAY_INTERFACE	The CGI specification being used. The current specification is CGI/1.1.
SERVER_NAME	The domain name or IP address of the Web server.
SERVER_SOFTWARE	The name and version of the Web server software executing the CGI script.

Passing Data to a CGI Script (continued)

The following environment variables are set by the Web server whenever a CGI script is executed:

Variable	Contains
AUTH_TYPE	The protocol-specific authentication method used to validate the user, provided the Web server supports user authentication and the script is protected.
CONTENT_LENGTH	The length of the user-provided content from the Web page requesting the CGI script.
CONTENT_TYPE	The content type of the data for queries which have attached information, such as HTTP POST and PUT.
PATH_INFO	Additional path information for the CGI script, as given by the Web client. Scripts can be accessed by their virtual path-name, followed by extra information at the end of the path. This information should be decoded by the server if it comes from a URL before it is passed to the CGI script.
PATH_TRANSLATED	The Web server translated version of PATH_INFO, which takes the path and performs any virtual-to-physical mappings.
QUERY_STRING	The user-provided information which follows the ? in the URL which referenced the CGI script. It should not be decoded in any fashion. This variable should always be set when there is query information, regardless of command line decoding.
REMOTE_ADDR	The IP address of the Web client requesting the CGI script.
REMOTE_HOST	The domain name of the Web client requesting the CGI script. If the Web server cannot retrieve this information, it should set REMOTE_ADDR and leave this unset.
REMOTE_IDENT	The user's login name, provided the Web server supports RFC 931 identification.
REMOTE_USER	The user name of the Web browser, provided the Web server supports user authentication and the script is protected.
REQUEST_METHOD	The request method used to call the CGI script. For HTTP, this is "GET", "HEAD", "POST", etc.
SCRIPT_NAME	The virtual path and name of the CGI script being executed. Used for self-referencing URLs.
SERVER_PORT	The port number of the Web server from which the Web browser requested the CGI script.
SERVER_PROTOCOL	The name and protocol being used to request the CGI script (usually the HTTP protocol).

Passing Data to a CGI Script (continued)

HTTP Request Headers

Any request headers received from a Web browser are also placed in environment variables with the prefix **HTTP_** followed by the header name. Note that the Web server may exclude headers that have already been processed. The Web server may also exclude any or all of these headers if including them would exceed any system environment limits.

Variable	Contains
HTTP_ACCEPT	A semicolon-separated list of media types the browser can accept in response from the Web server. To allow Web browsers to receive content types that they are not aware of, an asterisk "*" wildcard may be used in place of either the second half of the content-type value, or both halves. If no Accept field is present, then it is assumed that text/plain and text/html are accepted. Examples: text/plain, text/html; audio/basic; image/gif.
HTTP_ACCEPT_ENCODING	The Content-Encoding types the Web browser can receive in response from the Web server. Examples: x-compress; x-zip.
HTTP_ACCEPT_LANGUAGE	Language value the Web browser can receive in response from the Web server. A response in an unspecified language is allowed. Example: en_UK for the English of the United Kingdom.
HTTP_AUTHORIZATION	Authorization information from the Web browser. The format of this field is in extensible form. The first word is a specification of the authorization system in use.
HTTP_CHARGE_TO	Account information for the costs of the application of the method requested. The format is TBS. The format of this field must be in extensible form.

Passing Data to a CGI Script (continued)

HTTP Request Headers (continued)

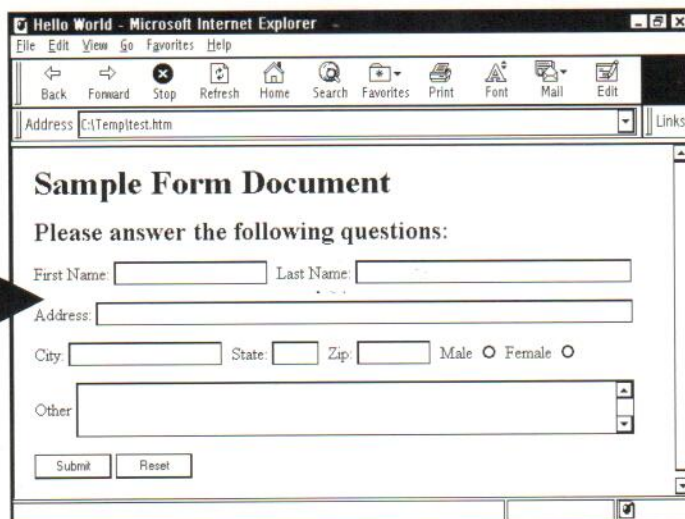
Variable	Contains
HTTP_FROM	The name of the requesting user as supplied by the Web browser in an Internet mail (e-mail) format. Example: DonGosselin@compuserve.com.
HTTP_IF_MODIFIED_SINCE	A value specified in a valid ARPANET date standard. Can be used in conjunction with the GET method to return the requested document only if it has changed since the date specified. Examples: Weekday, DD-Mon-YY HH:MM:SS TIMEZONE.
HTTP_PRAGMA	The value of any special directives for the Web server. Pragma directives should be understood by servers to which they are relevant, e.g. a proxy server. Currently only the no-cache pragma is defined. When present, the proxy should not return a document from the cache even though it has not expired, but it should always request the document from the actual server.
HTTP_REFERER	Contains the Uniform Resource Identifier (URI) of the document that contained the link to the currently requested document. Example: HTTP_REFERER: http://www.w3.org/hypertext/DataSources/Overview.html.
HTTP_USER_AGENT	The name of the Web browser software used by the Web client. Example: Mozilla/2.0 (Win95; I), which would be the user agent for the Netscape 2.0 browser for Windows 95.

Sample HTML/CGI Form Script

HTML Source Document

```
<HTML>
<TITLE>Sample Form Document</TITLE>
</HEAD>
<BODY>
<H1>Sample Form Document</H1>
<H2>Please answer the following questions:</H2>
<FORM METHOD="POST" ACTION="http://
www.domainname.com/sample.htm">
<P>First Name: <INPUT NAME="firstname" size="25">
Last Name: <INPUT NAME="lastname" size="48">
<P>Address: <INPUT NAME="address" size="97">
<P>City: <INPUT NAME="city" size="25"> State:
<INPUT NAME="state" size="5"> Zip: <INPUT
NAME="zip" size="10"> Male <INPUT NAME="gender"
TYPE=RADIO VALUE="male"> Female <INPUT
NAME="gender" TYPE=RADIO VALUE="female">
<P>Other <TEXTAREA NAME="other" cols=105 rows=2>
</textarea>
<P><INPUT TYPE=SUBMIT><INPUT TYPE=RESET>
</FORM>
</HTML>
```

Web Browser



Hello World - Microsoft Internet Explorer

File Edit View Go Favorites Help

Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit

Address C:\Temp\test.htm

Sample Form Document

Please answer the following questions:

First Name: Last Name:

Address:

City: State: Zip: Male ☐ Female ☐

Other

Submit Reset

Returning Data from a CGI Script

Data that is returned from a CGI script can be sent to the Web server or directly to the requesting Web browser through standard output (STDOUT). If the data is to be returned to the Web server, it must be preceded by a parsed header. This parsed header can contain any of the CGI environment variables listed on page 5, and is in the same format as an HTTP header. Note that parsed headers must always be followed by a blank line and that any lines that are not directives to the server are sent to the Web browser as part of the HTTP response header. If the data is to be returned directly to the requesting Web browser, it must be preceded by a nonparsed header containing proper HTTP response headers.

Server Directives for Parsed Headers

Directive	Function
Content-type	Specifies to the Web server the MIME type of the data being returned by the CGI script.
Location	Specifies to the Web server the URL or virtual path of a document rather than the document itself. If the argument is a URL, the server will issue a redirect to the Web client. If the argument is a virtual path, the Web server will retrieve the document as if the Web client had requested that document originally.
Status	Returns to the Web server an HTTP/1.0 status line that is sent to the Web client. The format is nnn xxxxx, where nnn is the 3-digit status code, and xxxxx is the reason string, such as "Forbidden".

Returning Data from a CGI Script (continued)

HTTP Response Headers

Header	Contains
ALLOWED	The request methods that the Web client is allowed. If this header line is omitted, the default request method allowed is GET HEAD.
CONTENT-ENCODING	The encoding method used.
CONTENT-LANGUAGE	The ISO code for the language in which the document is written. This field should be omitted if the language is not known.
CONTENT-LENGTH	The byte size of the data being returned.
CONTENT-TRANSFER-ENCODING	Encoding method of the data between the Web server and the Web browser.
CONTENT-TYPE	The type of data sent between the Web server and the Web browser. Examples: text/html; image/gif.
COST	Will contain the cost of retrieving the requested object. Header type is not yet specified.
DATE	The creation date of the requested object in a valid ARPANET format.
DERIVED-FROM	A version number for the requested object.
EXPIRES	The date the requested information becomes invalid and should be retrieved again. Formatted in an ARPANET date format.
LAST-MODIFIED	The last modification date of the requested object. Formatted in an ARPANET date format.
LINK	Used to specify information about the requested object such as the inclusion of another URL within a document or the creator of a returned object.

Returning Data from a CGI Script (continued)

HTTP Response Headers (continued)

Header	Contains
MESSAGE-ID	A unique identifier for an HTTP message.
PUBLIC	The request methods that the Web client is allowed. If this header line is omitted, the default request method allowed is GET.
TITLE	The title of the requested document. This field is not part of the document. For HTML files, this is equivalent to the value of the <TITLE></TITLE> elements.
URI	The Universal Resource Identifier (URI) where the requested object can be found. This will point to an object that should be the same as the one being returned, with some variation. This will not always be the URL the user entered in the Web browser requesting the returned object.
VERSION	Defines the version of a changing object. Header type is not yet specified.

Sample CGI Response Script

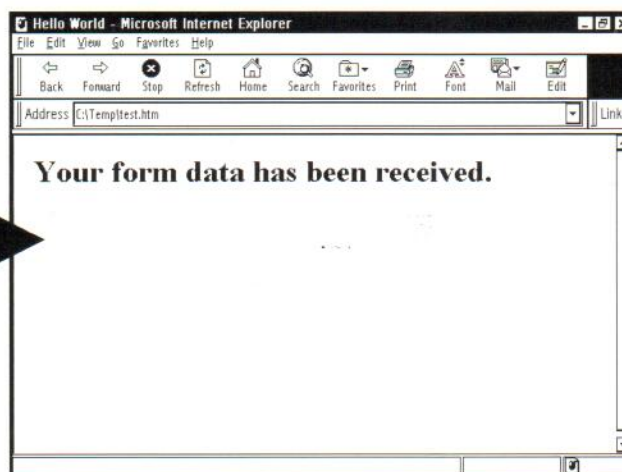
The following is a simple program written in PERL that demonstrates how an HTML document can be generated by a CGI script as a response.

PERL Script

```
# !/usr/local/bin/perl
# formdata.pl-- "Form Data Received" program
#
# The following line prints the CGI response header that
# is required for HTML output. Each \n sends
# a blank line (response headers must be followed
# by a blank line).
print "Content-type: text/html\n\n" ;

# The following lines print the HTML response
# page to STDOUT:
print <<EOF ;
<html>
<head><title>Form Data Received</title></head>
<body>
<h1>Your form data has been received.</h1>
</body>
</html>
EOF
exit ;
```

Web Browser



HTML Elements

The following listing includes the HTML 2.0 specifications, the draft HTML 3.0 specifications, and extensions made popular by various browsers, including **Netscape Navigator** and **Microsoft Internet Explorer**. The various options and syntax structure for each element are listed in the **Attributes** column. Please note that this listing is not complete due to the size and constantly evolving nature of the HTML language. (For the most current specifications, refer to the various online resources listed on page 32.)

Document Structure Elements

Element	Description	Attributes
<!-- COMMENTS -->	Includes programming-style comments in code of an HTML document.	--
<BASE>	Contains the absolute URL base to be used for any relative URL links in a document. It must be a complete file name, and is usually the original URL of a document.	<BASE HREF="..." TARGET="...">
<BODY>... </BODY>	Contains all the content of the document, as opposed to the HEAD, which contains information about the document. All displayable elements should be within the content of the BODY.	<BODY LANG="..." DIR=ltr rtl ID="..." CLASS="..." BACKGROUND="..." BGCOLOR="..." BG-PROPERTIES=fixed TEXT="#rrggbb" LINK="#rrggbb" VLINK="#rrggbb" ALINK="#rrggbb"> </BODY>
<HEAD>... </HEAD>	Contains general information about the document. Note that none of the contents of the HEAD are displayed.	<HEAD LANG="..." DIR=ltr rtl> </HEAD>

HTML Elements (continued)

Document Structure Elements (continued)

Element	Description	Attributes
<HTML>... </HTML>	Contains the entire HTML text in the document. All other HTML elements are inside the start and end of the HTML element.	<HTML VERSION="..." LANG="..." DIR=ltr rtl> </HTML>
<ISINDEX>	Notifies a Web server that the document is an index document.	<ISINDEX LANG="..." DIR=ltr rtl ACTION="..." PROMPT="...">
<LINK>	Indicates a relationship between this document and other documents or objects.	<LINK HREF="..." REL="..." REV="..." LANG="..." DIR=ltr rtl CHARSET="..." URN="..." TITLE="..." METHODS="..." >
<META>	Used within the HEAD element to embed document meta-information not defined by other HTML elements. This information may be extracted by servers or browsers.	<META HTTP-EQUIV="..." NAME="..." CONTENT="..." URL="..." >
<TITLE>... </TITLE>	The title of the document. Often used by a browser to label the display window. Some Web search engines only search the title of Web pages.	<TITLE LANG="..." DIR=ltr rtl>characters... </TITLE>

HTML Elements (continued)

Style Formatting Elements

Element	Description	Attributes
<ABBREV>... </ABBREV>	Changes the character rendering of the contents of the element to logically represent an abbreviation.	<ABBREV LANG="..." DIR=ltr rtl ID="..." CLASS="..."> characters... </ABBREV>
<ACRONYM>... </ACRONYM>	Changes the character rendering of the contents of the element to logically represent an acronym.	<ACRONYM LANG="..." DIR=ltr rtl ID="..." CLASS="..."> characters... </ACRONYM>
<AU> ... </AU>	Changes the character rendering of the contents of the element to logically represent the name of an author.	<AU LANG="..." DIR=ltr rtl ID="..." CLASS="..."> characters... </AU>
...	Changes the physical rendering of the contents of the element to a bold font.	<B LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters...
<BASEFONT SIZE>	Change the document base font size to one of the seven defined sizes. The default is 3.	<BASEFONT SIZE=1 2 3 4 5 6 7>
<BIG>...</BIG>	Changes the physical rendering of the contents of the element to a bigger font than normal text, if practical.	<BIG LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </BIG>
<BLINK>...</BLINK>	Changes the physical rendering of the contents of the element to a blinking font.	<BLINK>characters... </BLINK>
<CITE>...</CITE>	Changes the character rendering of the contents of the element to logically represent a citation.	<CITE LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </CITE>
<CODE>...</CODE>	Changes the character rendering of the contents of the element to logically represent computer code. It is intended for short words or phrases.	<CODE LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </CODE>

HTML Elements (continued)

Style Formatting Elements (continued)

Element	Description	Attributes
...	Changes the character rendering of the contents of the element to logically represent deleted text, as in modifications to legal documents.	<DEL LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters...
<DFN>...</DFN>	Changes the character rendering of the contents of the element to logically represent a defining instance of a term.	<DFN LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </DFN>
...	Changes the character rendering (usually to italics) of the contents of the element to logically emphasize the text.	<EM LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters...
...	Changes the font size to one of the seven defined sizes, or plus or minus from the document BASEFONT size.	 ...
<I>...</I>	Changes the physical rendering of the contents of the element to an italic font.	<I LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </I>
<INS>...</INS>	Changes the character rendering of the contents of the element to logically represent inserted text, as in modifications in legal documents.	<INS LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </INS>
<KBD>...</KBD>	Changes the character rendering of the contents of the element to logically represent text entered as keyboard input.	<KBD LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </KBD>

HTML Elements (continued)

Style Formatting Elements (continued)

Element	Description	Attributes
<code><LANG> ... </LANG></code>	Changes the default <code>LANG</code> context for subsequent elements from the current default.	<code><LANG ID="..." CLASS="...">characters... </LANG></code>
<code><PERSON>... </PERSON></code>	Changes the character rendering of the contents of the element to logically represent the name of people to allow to automatically extract the document with indexing programs.	<code><PERSON LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </PERSON></code>
<code><Q>...</Q></code>	Changes the character rendering of the contents of the element to logically represent a short quotation.	<code><Q LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </Q></code>
<code><S> ... </S></code>	Changes the physical rendering of the contents of the element to a font with a knockout line through the letters. This element replaces the <code>STRIKE</code> element described below.	<code><S LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </S></code>
<code><SAMP>...</SAMP></code>	Changes the character rendering of the contents of the element to logically represent a sequence of literal characters.	<code><SAMP LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </SAMP></code>
<code><SMALL>... </SMALL></code>	Changes the physical rendering of the contents of the element to a smaller font than normal text, if practical.	<code><SMALL LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </SMALL></code>
<code><STRIKE>... </STRIKE></code>	Changes the physical rendering of the contents of the element to a font with a knockout line through the letters.	<code><STRIKE>characters... </STRIKE></code>

HTML Elements (continued)

Style Formatting Elements (continued)

Element	Description	Attributes
<code>... </code>	Changes the character rendering of the contents of the element to a strong emphasis style.	<code><STRONG LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </code>
<code><SUB>...</SUB></code>	Changes the physical rendering of the contents of the element to a subscripted position.	<code><SUB LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </SUB></code>
<code><SUP>...</SUP></code>	Changes the physical rendering of the contents of the element to a superscripted position.	<code><SUP LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </SUP></code>
<code><TT>...</TT></code>	Changes the physical rendering of the contents of the element to a fixed-width teletype font.	<code><TT LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </TT></code>
<code><U> ... </U></code>	Changes the physical rendering of the contents of the element to an underlined font.	<code><U LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </U></code>
<code><VAR>...</VAR></code>	Changes the character rendering of the contents of the element to logically represent a variable name.	<code><VAR LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </VAR></code>

HTML Elements (continued)

Hypertext Elements

Element	Description	Attributes
<A>...	Contains a piece of text (and/or image) which is identified as a hypertext link. This element must have either an HREF attribute or a NAME attribute.	characters...
<AREA>	Specifies a single area of an image which, if selected, will link to the hyperlink identified by HREF.	<AREA COORDS="..." SHAPE=rect circle polygon HREF="..." NOHREF ALT="...">
<MAP>... </MAP>	Used to name and describe a client-side image map.	<MAP NAME="..."></MAP>

Form Elements

Element	Description	Attributes
<FORM> ... </FORM>	Defines data input form within an HTML document.	<FORM ACTION="..." METHOD=GET POST ENCTYPE="..." LANG="..." DIR=ltr rtl ACCEPT-CHARSET="..." SCRIPT="..."> </FORM>
<INPUT>	Defines form fields and their attributes. Resides between the <Form> and </Form> tags.	<INPUT TYPE="text password checkbox radio submit reset hidden image file range scribble jot" LANG="..." DIR=ltr rtl ID="..." CLASS="..." NAME="..." VALUE="..." SRC="..." CHECKED SIZE="..." MAXLENGTH=number ALIGN=top middle bottom left right ACCEPT="..." DISABLED ERROR="..." MIN=number MAX=number MD="...">

HTML Elements (continued)

Form Elements (continued)

Element	Description	Attributes
<OPTION> ... </OPTION>	Specifies choices that will be included within a <Select> element. This element can only occur within a <Select> element.	<OPTION SELECTED VALUE="..." LANG="..." DIR=ltr rtl ID="..." CLASS="..." DISABLED ERROR="..." SHAPE="...">characters... </OPTION>
<SELECT> ... </SELECT>	Creates drop-down and scrolling lists. Used with the <Option> element to create a list of choices.	<SELECT NAME="..." SIZE=value MULTIPLE LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=top middle bottom left right DISABLED ERROR="..." SRC="..." MD="..." WIDTH=value HEIGHT=value UNITS=pixels em>characters... </SELECT>
<TEXT AREA> ... </TEXT AREA>	Creates a multi-line input field.	<TEXTAREA NAME="..." ROWS="..." COLS="..." LANG="..." DIR=ltr rtl ID="..." CLASS="..." WRAP=off virtual physical ALIGN=top middle bottom left right DISABLED ERROR="..."> characters... </TEXTAREA>

Structural Formatting Elements

Element	Description	Attributes
<ADDRESS>... </ADDRESS>	Defines a separated multi-line set of text to be rendered for address information.	<ADDRESS LANG="..." DIR=ltr rtl ALIGN=center left right justify ID="..." CLASS="..." CLEAR=left right all "..." NOWRAP>characters... </ADDRESS>

HTML Elements (continued)

Structural Formatting Elements (continued)

Element	Description	Attributes
<BLOCKQUOTE>... </BLOCKQUOTE>	Defines a separated multi-line set of text to be rendered as quoted text.	<BLOCKQUOTE LANG="..." DIR=ltr rtl ALIGN=center left right justify> </BLOCKQUOTE>
<BQ>	Defines a multi-line set of text to be rendered as quoted text.	<BQ LANG="..." DIR=ltr rtl ID="..." CLASS="..." CLEAR=left right all "..." NOWRAP> </BQ>
 	New line break.	<BR CLEAR=left right all "..." ID="..." CLASS="...">
<CENTER>... </CENTER>	Centers the contents of this element between the left and right margins.	<CENTER>characters... </CENTER>
<DIV>...</DIV>	Proposed to be used with the CLASS attribute to represent different kinds of divisions (e.g. chapter, section, abstract, or appendix).	<DIV LANG="..." DIR=ltr rtl ALIGN=left center right justify ID="..." CLASS="..." NOWRAP CLEAR=left right all "...">characters... </DIV>
<FN>...</FN>	Logically identifies text to be presented as a footnote.	<FN LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </FN>
<H1>...</H1>	Level 1 heading.	<H1 LANG="..." DIR=ltr rtl ALIGN=left center right justify ID="..." CLASS="..." CLEAR=left right all "..." SEQNUM=nnn SKIP=nnn DINGBAT=entity-name SRC="..." MD="..." NOWRAP>characters... </H1>

HTML Elements (continued)

Structural Formatting Elements (continued)

Element	Description	Attributes
<H2>...</H2>	Level 2 heading.	<H2 LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=left center right justify CLEAR=left right all "..." SEQNUM=nnn SKIP=nnn DINGBAT=entity-name SRC="..." MD="..." NOWRAP>characters... </H2>
<H3>...</H3>	Level 3 heading.	<H3 LANG="..." DIR=ltr rtl ALIGN=left center right justify ID="..." CLASS="..." CLEAR=left right all "..." SEQNUM=nnn SKIP=nnn DINGBAT=entity-name SRC="..." MD="..." NOWRAP>characters... </H3>
<H4>...</H4>	Level 4 heading.	<H4 LANG="..." DIR=ltr rtl ALIGN=left center right justify ID="..." CLASS="..." CLEAR=left right all "..." SEQNUM=nnn SKIP=nnn DINGBAT=entity-name SRC="..." MD="..." NOWRAP>characters... </H4>
<H5>...</H5>	Level 5 heading.	<H5 LANG="..." DIR=ltr rtl ALIGN=left center right justify ID="..." CLASS="..." CLEAR=left right all "..." SEQNUM=nnn SKIP=nnn DINGBAT=entity-name SRC="..." MD="..." NOWRAP>characters... </H5>
<H6>...</H6>	Level 6 heading.	<H6 LANG="..." DIR=ltr rtl ALIGN=left center right justify ID="..." CLASS="..." CLEAR=left right all "..." SEQNUM=nnn SKIP=nnn DINGBAT=entity-name SRC="..." MD="..." NOWRAP>characters... </H6>

HTML Elements (continued)

Structural Formatting Elements (continued)

Element	Description	Attributes
<HPn> ... </HPn>	HP set of elements, where n=1,2, etc. This element provides a mechanism to highlight the characters in a phrase with one of a set of browser-defined highlight mechanisms.	<HPn>characters... </HPn>
<HR>	Creates a horizontal ruler.	<HR DIR=ltr rtl ALIGN=left right center justify ID="..." CLASS="..." CLEAR=left right all "..." SRC="..." MD="..." SIZE=number WIDTH=number percent NOSHADE>
<NEXTID>	Provides the next available identifier for use by automatic hypertext editors.	<NEXTID N="..." >
<NOBR> ... </NOBR>	Disallows line breaks between the contents of an NOBR element.	<NOBR>characters... </NOBR>
<NOTE>...</NOTE>	Changes the rendering of the contents of the element to logically represent separated notational text.	<NOTE LANG="..." DIR=ltr rtl ID="..." CLASS="..." CLEAR=left right all "..." SRC="..." MD="..."></NOTE>
<P>...</P>	Paragraph break.	<P ALIGN=center left right justify indent WRAP=on off NOWRAP CLEAR=left right all "..." LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </P>
<PLAINTEXT> ... </PLAINTEXT>	Defines a separated multi-line set of text to be rendered as it exists in the source document with the same line breaks.	<PLAINTEXT WIDTH="..." LANG="..." DIR=ltr rtl>characters... </PLAINTEXT>
<PRE>...</PRE>	Defines a separated multi-line set of text to be rendered as it exists in the source document with the same line breaks and spaces.	<PRE WIDTH=number LANG="..." DIR=ltr rtl ID="..." CLASS="..." CLEAR=left right all "...">characters... </PRE>

HTML Elements (continued)

Structural Formatting Elements (continued)

Element	Description	Attributes
 ... 	Sets language characteristics for its content.	characters...
<TAB>	Horizontal tab.	<TAB ID="..." INDENT=ens TO="..." ALIGN=left center right decimal DP="...">characters... </TAB>
<WBR>	Force a word break in a no-break section.	<WBR>
<XMP> ... </XMP>	Defines a separated multi-line set of text to be rendered as it exists in the source document with the same line breaks.	<XMP WIDTH="..." LANG="..." DIR=ltr rtl>characters... </XMP>

Image Elements

Element	Description	Attributes
<CAPTION>... </CAPTION>	Labels a table or figure.	<CAPTION ALIGN=top bottom left right LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </CAPTION>
<CREDIT>... </CREDIT>	Names the source of a block quotation or figure	<CREDIT LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </CREDIT>
<FIG> ... </FIG>	Advanced form of the IMG element to define an image, with optional overlays, text elements and "hotzones".	<FIG SRC="..." LANG="..." DIR=ltr rtl ID="..." CLASS="..." CLEAR=left right all "..." NOFLOW MD="..." ALIGN=left right center justify bleedleft bleedright WIDTH=value HEIGHT=value UNITS="..." IMAGEMAP="..."></FIG>

HTML Elements (continued)

Image Elements (continued)

Element	Description	Attributes
	Incorporates an in-line graphic into an HTML document.	
<OVERLAY>	Overlays one or more images on top of a FIG image.	<OVERLAY SRC="..." MD="..." UNITS=pixels en X=value Y=value WIDTH=value HEIGHT=value IMAGE MAP="...">

List Elements

Element	Description	Attributes
<BDO> ... </BDO>	Directional override.	<BDO DIR=ltr rtl LANG="...">characters... </BDO>
<DD>...</DD>	Identifies the separated multi-line definition item in a DL definition list.	<DD LANG="..." DIR=ltr rtl>characters... </DD>
<DIR>...</DIR>	Defines an unordered list consisting of a number of single-line elements.	<DIR COMPACT LANG="..." DIR=ltr rtl ALIGN=center left right justify> </DIR>
<DL>...</DL>	Defines a definition list.	<DL COMPACT LANG="..." DIR=ltr rtl ID="..." CLASS="..." CLEAR=left right all "..."> </DL>

HTML Elements (continued)

List Elements (continued)

Element	Description	Attributes
<DT>...</DT>	Identifies the separated term item in a DL definition list.	<DT LANG="..." DIR=ltr rtl>characters... </DT>
<LH> ... </LH>	Defines a list header used as a title for a list.	<LH LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </LH>
...	Used within the context of the <DIR> element to define a list item.	<LI LANG="..." DIR=ltr rtl ALIGN=center left right justify within UL TYPE=disk circle square within OL TYPE=A a I i i within OL VALUE=n>characters...
<LISTING> ... </LISTING>	Defines a separated multi-line set of text to be rendered as it exists in the source document with the same line breaks.	<LISTING WIDTH="...">characters... </LISTING>
<MENU>...</MENU>	Defines an unordered list consisting of a number of separated multi-line elements which may or may not be marked by a bullet or similar symbol.	<MENU COMPACT LANG="..." DIR=ltr rtl ALIGN=center left right justify> </MENU>
...	Defines an ordered list consisting of a number of separated multi-line elements, ordered numerically in some way.	<OL COMPACT LANG="..." DIR=ltr rtl ALIGN=center left right justify ID="..." CLASS="..." CLEAR=left right all "..." CONTINUE SEQNUM=value START=value TYPE=A a I i 1>
...	Defines an unordered list consisting of a number of separated multi-line elements, usually marked by a bullet or similar symbol.	<UL COMPACT LANG="..." DIR=ltr rtl ALIGN=center left right justify ID="..." CLASS="..." CLEAR=left right all "..." PLAIN SRC="..." MD="..." DINGBAT="..." WRAP=vert horiz TYPE=disk circle square>

HTML Elements (continued)

Table Elements

Element	Description	Attributes
<CAPTION>... </CAPTION>	Defines a table caption.	<CAPTION ALIGN=top bottom left right LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </CAPTION>
<COL>	Specifies column based defaults for table properties.	<COL LANG="..." DIR=ltr rtl ID="..." CLASS="..." SPAN=nn WIDTH="..." ALIGN=left center right justify char CHAR="..." CHAROFF="..." VALIGN=top middle bottom baseline>characters... </COL>
<COLGROUP> ... </COLGROUP>	Defines a group of one or more columns and specifies the defaults for all the columns in this group.	<COLGROUP LANG="..." DIR=ltr rtl ALIGN=left center right justify char ID="..." CLASS="..." CHAR="..." CHAROFF="..." VALIGN=top middle bottom baseline>characters... </COLGROUP>
<TABLE>... </TABLE>	Defines a series of rows of table cell elements.	<TABLE LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=left right center justify bleedleft bleedright WIDTH="..." COLS=number BORDER="..." FRAME=void above below hsides lhs rhs vsides box border RULES=none basic rows cols all CELLS=number all CELLSPACING="..." CELL-PADDING="..." CLEAR=left right all "..." NOFLOW UNITS= en relative pixels COLSPEC="..." DP="..." NOWRAP></TABLE>
<TBODY> ... </TBODY>	Encloses a series of table row definitions and specifies the defaults for all the rows in this group.	<TBODY LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=left right center justify char CHAR="..." CHAROFF="..." VALIGN=top middle bottom baseline>characters... </TBODY>

HTML Elements (continued)

Table Elements (continued)

Element	Description	Attributes
<TD>... </TD>	Defines a data cell as part of a TABLE construct.	<TD LANG="..." DIR=ltr rtl ID="..." CLASS="..." AXIS="..." AXES="..." NOWRAP ROWSPAN=value COLSPAN=value ALIGN=left right center justify char decimal CHAR="..." CHAROFF="..." DP="..." VALIGN=top middle bottom baseline WIDTH=value BGCOLOR="...">characters... </TD>
<TFOOT> ... </TFOOT>	Encloses a series of table row definitions and specifies the defaults for all the rows in this group.	<TFOOT LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=left right center justify char CHAR="..." CHAROFF="..." VALIGN=top middle bottom baseline>characters... </TFOOT>
<TH>... </TH>	Defines a header cell as part of a TABLE construct.	<TH LANG="..." DIR=ltr rtl ID="..." CLASS="..." AXIS="..." AXES="..." NOWRAP ROWSPAN=value COLSPAN=value ALIGN=left right center justify char decimal CHAR="..." CHAROFF="..." DP="..." VALIGN=top middle bottom baseline WIDTH=value BGCOLOR="...">characters... </TH>
<THEAD> ... </THEAD>	Encloses a series of table row definitions and specifies the defaults for all the rows in this group.	<THEAD LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=left right center justify char CHAR="..." CHAROFF="..." VALIGN=top middle bottom baseline>characters... </THEAD>
<TR>	Defines a table row in one of the three sections of a TABLE construct	<TR LANG="..." DIR=ltr rtl ID="..." CLASS="..." ALIGN=left right center justify char decimal CHAR="..." CHAROFF="..." VALIGN=top middle bottom baseline DP="...">

HTML Elements (continued)

Frame Elements

Element	Description	Attributes
<FRAME>	Defines a single frame in a frameset.	<FRAME SRC="..." NAME="..." MARGINWIDTH="..." MARGINHEIGHT="..." SCROLLING=yes no auto NORESIZE>
<FRAMESET> ... </FRAMESET>	Used instead of the BODY element in an HTML document whose sole purpose is to define the layout of the sub-HTML documents, or frames, that will make up the page.	<FRAMESET ROWS="..." COLS="...">characters... </FRAMESET>
<NOFRAMES> ... </NOFRAMES>	Defines content within FRAMESET that is to be ignored by browsers that can define frames.	<NOFRAMES>characters... </NOFRAMES>

Dynamic Elements

Element	Description	Attributes
<!--#...>	Server Side Includes. Allows interactive, real-time querying and updating of databases, echoing current time, conditional executions, e-mail, and other features, without programming or CGI scripts.	<!--# ECHO VAR="..." INCLUDE virtual file="..." FSIZE virtual file="..." FLASTMOD virtual file="..." EXEC CMD CGI="..." CONFIG errmsg timefmt sizefmt cmdecho cmdprefix cmdpostfix onerr="..." ODBC debug connect statement format="..." EMAIL fromhost tohost fromaddress toaddress message subject sender replyto cc inreplyto id="..." IF="..." GOTO="..." LABEL="..." BREAK --->
<APPLET> ... </APPLET>	Identifies and invokes a Java application.	<APPLET CODE="..." WIDTH="..." HEIGHT="..." CODEBASE="..." ALT="..." NAME="..." ALIGN=left right top texttop middle absmiddle baseline bottom absbottom VSPACE="..." HSPACE="...">characters... </APPLET>

HTML Elements (continued)

Dynamic Elements (continued)

Element	Description	Attributes
<BANNER> ... </BANNER>	Proposed for corporate logos, navigation aids, disclaimers and other information that should not be scrolled with the rest of the document.	<BANNER LANG="..." DIR=ltr rtl ID="..." CLASS="...">characters... </BANNER>
<BGSOUND>	Plays an audio file as background sound in a document.	<BGSOUND SRC="..." LOOP="...">
<EMBED> ... </EMBED>	Defines a container that allows the insertion of arbitrary objects directly into an HTML page.	<EMBED SRC="..." HEIGHT="..." WIDTH="..." attribute_1="..." attribute_2="..." ...>characters... </EMBED>
<MARQUEE> ... </MARQUEE>	Defines an area in which visual scrolling is used to display the content of the element.	<MARQUEE ALIGN=top middle bottom BEHAVIOR=scroll slide alternate BGCOLOR=#rrggbb color-name DIRECTION=left right HEIGHT=number number% HSPACE=number LOOP=number INFINITE -1 SCROLLAMOUNT=number SCROLLELAY=number VSPACE=number>characters... </MARQUEE>
<NOEMBED> ... </NOEMBED>	Defines content within EMBED content that is ignored by browsers that can activate the EMBED plug-in application.	<NOEMBED>characters... </NOEMBED>
<PARAM>	Define general purpose parameters passed to <APPLET> applications.	<PARAM NAME="..." VALUE="...">

Online Resources

CGI

Name	URL
CGI Manual of Style	http://www.mcp.com/zdpress/features/3970/
Common Gateway Interface	http://hoohoo.ncsa.uiuc.edu/cgi/
Common Gateway Interface Specification	http://hoohoo.ncsa.uiuc.edu/cgi/interface.html
Matt's Script Archive	http://www.worldwidemart.com/scripts/
Selena Sol's Public Domain CGI Script Library	http://www.eff.org/~erict/Scripts/

HTTP

Name	URL
HTTP Protocol As Implemented In W3	http://www.w3.org/hypertext/WWW/Protocols/HTTP/AsImplemented.html
Object Header lines in HTTP	http://www.w3.org/hypertext/WWW/Protocols/HTTP/Object-Headers.html
Protocols used in the World-Wide Web	http://www.w3.org/hypertext/WWW/Protocols/RelevantProtocols.html
Status codes in HTTP	http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTRESP.html
Uniform Resource Locators	http://www.w3.org/hypertext/WWW/Addressing/URL/Overview.html

HTML

Name	URL
A Beginner's Guide to HTML	http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html
HTML Quick Reference	http://www.cc.ukans.edu/info/HTML_quick.html
HTML Quick Reference (Wilson)	http://sdcc8.ucsd.edu/~m1wilson/htmlref.html
HTML Reference Manual	http://www.sandia.gov/sci_compute/html_ref.html
HTML3 Manual of Style	http://www.mcp.com/zdpress/features/3520/
HyperText Markup Language	http://www.w3.org/pub/WWW/MarkUp/

RHI Consulting is...

...The world's leader in information technology staffing. Finding the right information technology support for your organization can be difficult. Experience, personality, technical skills and cost must be weighed and factored into your decision.

Through ProSCANSM, our specialized evaluation process, *RHI Consulting* guarantees that our consultants possess the knowledge and skills necessary to get the job done right. And, our InfinitySM  Advanced Technical Training Program assures that our consultants maintain leading-edge skills for the IT industry.

Call today to be connected with your nearest *RHI Consulting* location or contact us online for more information.

RHICONSULTING[®]

Information Technology Professionals

1-800-793-5533 U.S. • 1-800-268-1948 Canada

www.rhic.com