# Sample Web Page

Original source:[http://www.columbia.edu/%7Efdc/sample.html](#http://www.columbia.edu/%7Efdc/sample.html)

NOTE: This was originally created by Frank da Cruz as an HTML 4 document. It has been converted to an HTML5 document but I didn't change the actual examples from HTML 4 to HTML5.

  
*A random photo, maximize your browser to enlarge.*

Frank da Cruz  
 Sat Jan 17 12:07:32 2004

### CONTENTS

[Creating a Web Page](#basics)

[HTML Syntax](#syntax)

[Special Characters](#chars)

[Converting Plain Text to HTML](#convert)

[Effects](#effects)

[Lists](#lists)

[Links](#links)

[Tables](#tables)

[Installing your Web Page on the Internet](#install)

[Where to go from here](#more)

### 1. Creating a Web Page

This page was typed by hand. Anybody can do this, you don't need any special "web creation" tools or HTML editors, and the pages you make can be viewed from any browser. To see how this page was made, choose**View Source** (or View Page Source, or View Document Source) in your browser's menu. A simple web page like this one is just plain text with HTML commands (markup) mixed in. HTML commands themselves are plain text.

When you're just learning and want to experiment, you can do everything on your PC. Create a new directory ("folder") for your website, and then put the web-page files (HTML plus any pictures) in it. Use NotePad or other plain-text editor (not word processor) on your PC to create anindex.html file, which you can view locally with your Web browser. (You can also use word processors such as Word or WordPad if you save in "plain text", "text", "text document", or "text document MS-DOS format".)[Later](#install) I'll explain how you can install your web site on the Internet.

***Example for Windows:***

Move your mouse cursor to the desktop. Right-click, choose*New*, then choose*Folder*, and a "New Folder" appears, with its name highlighted. Type "Web" to change the folder's name to Web (or anything else you want). Then*Start* -> *Run* and type "notepad" to start NotePad. Now you can create your first Web page.

### 2. HTML Syntax

Web pages are written in Hyper Text Markup Language (HTML). HTML has three special characters:<,&,>. An HTML command is enclosed in<...>, for example<p>, which is a paragraph separator, or<b> ("begin bold") and</b> ("end bold"). So the following HTML text:

This sentence contains <b>bold</b> text.

produces:

This sentence contains**bold** text.

A Web page starts with a series of HTML commands, and ends with a few more. The contents go in between:

*(Contents go here)*

</body> </html>

The first line (DOCTYPE) specifies which language the page uses; just copy this line1. The next line,<html>, starts the page, and is matched by the last line,</html>, which closes the page.<head>, starts the heading, which contains a title to be displayed on the browser's title bar and a declaration of the character set (usually US-ASCII, ISO-8859-1, or UTF-8).</head> closes the heading.

The<body> statement starts the body of the document and can contain specifiers for color and other attributes. This one chooses black writing on a white background. The body is closed by</body> statement.

As you can see, most HTML commands (but not all!) come in begin-end pairs:<b>...</b>,<head>...</head>, etc. The closing part of the command has a slash (/) between the< and the first letter of the command.

Blank lines and line breaks are ignored. The browser automatically "flows" your text into lines and paragraphs that fit in its window. Paragraphs must be separated by<p>. Line breaks can be forced by<br>.

***Example for Windows:***

Use the mouse to copy the HTML above into NotePad. Then save the file (*File* -> *Save As...*) in your Web directory asindex.html. Suppose your Windows username is Olga. Then (depending on which version of Windows you have) this might be:

C:\Documents and Settings\Olga\Desktop\Web\index.html

Now to see your new web page, just double-click on the Web folder and then double-click onindex.html.

Now you're ready to start adding "content" to your web page. Go back to NotePad and replace the title and "(Contents go here)" with whatever you want. Any time you want to see the result, use*File* -> *Save* in NotePad and then click the Reload button on your browser.

The next sections tell how to achieve different kinds of effects.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HTML 4.01 is chosen because it supports tables (new to HTML 3.2) as well as a couple other desirable features, such as the ability to specify an image width as a percentage of the Browser's window width, and a bunch more character entity symbols such as&euro;(new to 4.01). Otherwise we would stick with the oldest available verifiable version, which is 2.0.

### 3. Special Characters

HTML special "character entities" start with ampersand (&) and end with semicolon (;), like "&euro;" = "". The ever-popular "no-break space" is&nbsp;. There are special entity names for accented Latin letters and other West European special characters such as:

|  |  |  |
| --- | --- | --- |
| &auml; | a-umlaut |  |
| &Auml; | A-umlaut |  |
| &aacute; | a-acute |  |
| &agrave; | a-grave |  |
| &ntilde; | n-tilde |  |
| &szlig; | German double-s |  |
| &thorn; | Icelandic thorn |  |

Examples:

**ForSpanish you would need:**

&Aacute; (),&aacute; (),&Eacute; (),&eacute; (),&Iacute; (),&iacute; (),&Oacute; (),&oacute; (),&Uacute; (),&uacute; (),&Ntilde; (),&ntilde; ();&iquest; ();&iexcl; ().  
 Example: Aorarn =A&ntilde;orar&aacute;n.

**ForGerman you would need:**

&Auml; (),&auml; (),&Ouml; (),&ouml; (),&Uuml; (),&uuml; (),&szlig; ().  
 Example: Gre aus Kln =Gr&uuml;&szlig;e aus K&ouml;ln.

CLICK HEREfor a complete list. Cyrillic, Arabic, Hebrew, Greek. Japanese, etc, are a bit harder, butdoable.

If you want to include<,&, or> literally in text to be displayed, you have to write&lt;,&amp;,&gt;, respectively.

### 4. Converting Plain Text to HTML

If you have a plain text file that you want to convert to HTML, load the file into a plain-text editor and then follow these steps.

Change all occurrences of "&" to "&amp;".

Change all occurrences of "<" to "&lt;".

Change all occurrences of ">" to "&gt;".

Change any accented letters to HTML entity names (previous section).

Put "<p>" between each paragraph.

Insert the standard prolog at the top, substituting an appropriate title.

Insert the standard epilog at the bottom.

Save the result as*xxx*.html, where*xxx* is the part of the original file's name before the dot. For example, if the original file wasletter.txt, save the edited version asletter.html.

If you are aKermituser, you can find a script to convert plain text to HTMLHERE.

If the text contains lists, tables, or other structures, read on.

If you have a Microsoft Word document you want to convert to HTML, and your copy of Word does not allow the file to be "Saved As" HTML, then save it as plain text and follow the same instructions. In this case you lose the "richness" (bold, italics, font changes, etc) when you save the file, and will have to put the effects back by hand (next section).

### 5. Effects

The rest of this document shows what you can do with simple HTML commands, but I don't explain how to do it. To see that, just tell your browser to**View Source** and compare the HTML in the source window with the result in the original window.

**This sentence is bold.***This sentence is in italics.****This sentence is in bold italics.***This sentence is in typewriter font. This sentence hasunderlined words and**underlined bold words**.Thissentencehascoloredwords. This sentence hasbig words. This one hasvery big words. This one hasvery small words.

This is a "blockquote", which is like a regular paragraph, but it has bigger margins. Begin a blockquote with<blockquote> and end it with</blockquote>. Environments such as blockquotes, lists, etc, that have a beginning and an end always use paired commands like<*blah*>...<*/blah*>.

This is a blockquote inside another blockquote, which shows how HTML environments can be "nested".

Here we are back in the first blockquote again.

And here we are back outside of the first blockquote.

### 6. Lists

Here is an Unordered (bullet) List (<ul>>..</ul>):

This is a List Item (<li>).

This is another item.

This is yet another item.

Here is an Ordered (numbered) List (<ol>>..</ol>):

This is a List Item (<li>).

This is another item.

This is yet another item.

And here is a Description List (<dl>). using Kermit commands as an example:

**SET FILE TYPE BINARY**

This command tells Kermit to transfer files in binary mode. In other words, don't mess with the file, just send it as-is. The result on the receiving computer should be identical to the original.

**SET FILE TYPE TEXT**

This command tells Kermit to transfer files in text mode. This should be used with plain-text files, especially when transferring them between computers with different file formats or operating systems, such as VMS and Unix, or Unix and Windows. It converts the file's format and character-set (if necessary) so the received file is usable on the destination computer.

You can have lists within lists:

A gromet

A widget

A framus, which consists of the following components:

A doohickey.

A veeblefetzer -- these come in several colors:

Purple

Purple

Purple

A whatchamacallit.

A doodad.

And you can have ordered lists that use letters instead of numbers:

Pennies

Nickels

Dimes

Quarters

### 7. Links

Links can be internal within a Web page (like to the[Table of Contents](#contents) at the top), or they can be to external web pages or pictures on the same website, or they can be to websites, pages, or pictures anywhere else in the world.

Here is a link to theKermit Project home page.

Here is a link to[Section 5](#lists) of this document.

Here is a link toSection 4.0of theLC-Kermit for Unix Installation Instructions.

Here is a link to a picture:[CLICK HERE](#./Sample Web Page_files/picture-of-something.jpg) to see it.

### 8. Tables

Here's a simple table with some headings and a few rows:

|  |  |  |
| --- | --- | --- |
| Heading A | Heading B | Heading C |
| Cell 1A | Cell 1B | Cell 1C |
| Cell 2A | Cell 2B | Cell 2C |
| Cell 3A | Cell 3B | Cell 3C |

Same table again but with borders:

|  |  |  |
| --- | --- | --- |
| Heading A | Heading B | Heading C |
| Cell 1A | Cell 1B | Cell 1C |
| Cell 2A | Cell 2B | Cell 2C |
| Cell 3A | Cell 3B | Cell 3C |

And again but with Column C right-adjusted:

|  |  |  |
| --- | --- | --- |
| Heading A | Heading B | Heading C |
| Cell 1A | Cell 1B | Cell 1C |
| Cell 2A | Cell 2B | Cell 2C |
| Cell 3A | Cell 3B | Cell 3C |

### 9. Installing Your Web Page on the Internet

How to put your web page on the Internet depends on your Internet Service Provider (ISP). At Columbia University, each user has a "shell account" on the central server, which runs a Unix-based operating system, and which you can access with a Telnet, SSH, or modem program such as[Kermit](#http://www.columbia.edu/kermit/). Here's an example that applies to Columbia University's web server, showing how to upload your files from Windows:

*There are easier ways to do this than what I describe below, but they require add-on software. The following method should work for everybody who has Windows and an Internet connection.*

If you create apublic\_html subdirectory of your login directory, give it world read and search permission, and then create anindex.html file in that directory and give it world read permission, you'll have a home page. In this example "$" is the shell prompt (yours might be different), and what you type is underlined. CAUTION: the directory name ispublic\_html but the underscore might be obscured the underline in the examples below. Whenenever typing "public\_html" always include the underscore.

$cd*(Change to your login directory)* $mkdir public\_html*(Create* public\_html*subdirectory)* $chmod 755 public\_html*(Give it world read/search permission)* $cd public\_html*(Enter the* public\_html*subdirectory)*

You only have to do this part once.

Let's assume you have created a website in the Web folder on your PC. Here's an example of how to upload your Web files to yourpublic\_html directory on Columbia University's Cunix server using FTP (File Transfer Protocol). First start the FTP program:

*Start* -> *Run*

and type "ftp" in the box. An FTP window opens and an "ftp>" prompt appears. Type the underlined commands at the "ftp>" prompt (substituting your own user ID, etc):

ftp>lcd Desktop Local directory now C:\Documents and Settings\olga\Desktop. ftp>lcd Web Local directory now C:\Documents and Settings\olga\Desktop\Web. ftp>open cunix Connected to cunix.cc.columbia.edu. 220 Cunix FTP server (Version 5.60) ready. User (cunix.cc.columbia.edu:(none)):olga 331 Password required for olga. Password:*(type your password here)* 230 User olga logged in. ftp>cd public\_html*(That's* "public\_html"*in case you can't see the underscore)* ftp>binary ftp>put index.html 200 PORT command successful. 150 Opening ASCII mode data connection for index.html. 226 Transfer complete. ftp: 285 bytes sent in 0.00Seconds 285000.00Kbytes/sec. ftp>site chmod 664 index.html 200 CHMOD command successful ftp>bye

This sends theindex.html file to yourpublic\_html directory on the server. You can send any other file by subsituting its name for "put index.html. If you want to send all the files in your Web folder, replace "put index.html with "put \*" (asterisk, meaning "all files" in this directory). Always use binary mode unless you know what you're doing.

If the "site chmod" command failed (this service is not supported by some FTP servers), you have one more step. Before others can see your web files, you have to give them "world read" permission. Again, log in to the server using a terminal emulator (Telnet, SSH, Kermit, whatever), and:

$cd ~/public\_html*(Enter the* public\_html*subdirectory)* $chmod 664 \**(Make all files publicly readable)*

Now you have a home page. If you were at Columbia and your login ID was "olga", the address (URL) of your home page would be:

[http://www.columbia.edu/~olga/](#http://www.columbia.edu/~olga/)

If you want to add pictures to your Web page, you can upload those too (also with Kermit or FTP), and you also have to "chmod 664" all the files to make them readable by everybody. Every time you add new files to yourpublic\_html directory, you have to "chmod 664" them so they are accessible, either in the FTP session itself (as shown previously), or by logging in to the host and:

$cd ~/public\_html*(That's* "public\_html"*in case you can't see the underscore)* $chmod 664 \*

Pictures should be in JPG or GIF format. To include a picture ("image") in your page, include a statement like this at the desired spot:

<img src="*filename*" alt="brief description" width="100%">

Replace*filename* by the name of the file (e.g.skyline.jpg). You can omit or experiment with thewidth= clause. 100% means "make the picture the same width as the browser window."

Now you have your own home page on the Web, and your own URL (Uniform Resource Locator, or Web address). In this example, the URL is:

[http://www.columbia.edu/~olga/](#http://www.columbia.edu/~olga/)

Of course, if you prefer, you can also do all the Web-page editing directly on the server, using an a server-based text editor like EMACS, Pico, or Vi while logged in to the Unix shell. In that case you don't need to upload anything (except maybe photos), but then you also need to be more familiar with the server's Unix environment and commands and utilities.

### 10. Where to go from here

Most Web pages are created by hideous bloated "Web Authoring" tools, which are designed to force readers to use a Web browser (such as IE) that is made by the same company that made the Web-authoring tool (e.g. Front Page). If all you want is text with some pictures and links and maybe some tables, as opposed to spinning blinking popup holograms with streaming video and sound effects, it's best to keep it simple and do it yourself. This is how the Web started off in the HTML 1.0 days of the early 1990s. The ingenious thing was that it was self propogating. If you saw a web page with a certain effect and wanted to know how it was done, you could simply "view source" to get the "source code" and then adapt it to your own page. You can still do that with pages that look like this one, but since most Web pages are no longer made by hand, you'll often see tons of incomprehensible gibberish (the more special effects, the more gibberish).

If you have mastered the simple techniques shown in this page, you know the basics. Which is more than can be said of many "web designers" who only know how to use prepackaged software to create web pages by picking things from menus and moving things around with a mouse.

To learn more, follow these links:

The HTML 4.01 specification

You can check the validity of your web page at the[W3C Markup Validation Service](#http://validator.w3.org/).

You can find lots more Web pages in the same simple style as this one at the[Kermit Project](#http://www.columbia.edu/kermit/) website.

Don't worry that the HTML 4.01 specification says that everything from HTML 1.0 is "deprecated" and we're supposed to be using XHTML or XML and "cascading style sheets" or whatever. They're not allowed to take the original simple HTML syntax away.

If your web pages doesn't look the way you expected, try using the[Validation Service](#http://validator.w3.org/). If you have errors in it, this will tell you what they are. The most common errors are unclosed environments (such as<blockquote> without a matching</blockquote>) and missing doublequotes inside commands, e.g.:

<a href="http://www.columbia.edu/kermit>

which must be:

<a href="http://www.columbia.edu/kermit">

For more about "special characters" and character sets, see:

[HTML 4 Symbolic Character Entities](#http://www.columbia.edu/~fdc/entities.html)

A[Unicode 4.0 Table](#http://www.columbia.edu/kermit/utf8-t1.html) (all but Chinese, Japanese, and Korean)

More[character set tables](#http://www.columbia.edu/kermit/csettables.html)

A[UTF-8 sampler](#http://www.columbia.edu/kermit/utf8.html) (text in many languages)

[Unicode Character Charts](#http://www.unicode.org/charts/) (official)

*(End)*