

Making web pages usable

© 1996-06-17
Charles A. Poynton
tel +1 416 486 3271
fax +1 416 486 3657
poynton@poynton.com
www.poynton.com/~poynton

This note describes issues of information design that you should understand if you wish to make web pages that are a pleasure for your visitors to use.

To put pages up on the World Wide Web (www), you need to have access to a web server (perhaps run by an Internet service provider). You need to use conversion or authoring tools to create the HTML files of the web. Finally, you need to access the machinery to transfer the files and make them accessible through your web server. Those tasks are complicated, but they are all more or less mechanical. They are described well elsewhere, and I assume that you are familiar with them.

In order for the information that you provide to be useful and pleasurable for your web visitors, you must write well, and if you use images, you must present them well. These are aesthetic issues, and I cannot tell you too much about them here.

There is an intermediate area between the mechanical and the aesthetic that you must also consider in making web pages: the area of information design. This domain is fairly well understood for the production of books, posters, maps and so on, but is rather undeveloped in the electronic domain. Information

design for the web involves technical elements put into service to convey information. The mechanics will get easier as HTML tools become available, but respecting a carefully-chosen set of content conventions will remain important, and that work will never be done by machines.

Don't say, "Click here!"

The style guides deprecate this phrase, but it is so rampant on the web that it deserves special mention: It is a mistake to make a link that is labelled click here.

There are several reasons to avoid this phrase. First, some web browsers do not have mice: they activate links through means other than clicking. Second, web pages are often printed, or saved text-only! The notion of clicking on a phrase in its printed form is quite absurd; the absurdity reflects on the author. Third, it is good information design and human interface design to put in a link an indication of what is *under* the link. A sodapop machine doesn't have a button labelled *Press here*, it has a button labelled *7-up!*

Say, *You can access a list of articles, or A list of articles is available.* These phrases make perfect sense, even when printed or displayed in text-only form without hypertext links.

Don't say, "Under Construction."

The web is so naturally dynamic that it seems to me redundant at best, and cute at worst, to draw attention to this. So please, abolish those silly *Under Construction* icons. If it weren't under construction, it wouldn't be the web.

Don't say, "Coming soon."

Your visitor will be very frustrated to access a link that says, *Application Notes are available*, only to be presented with a content-free page that says *Coming soon*. Put the *Coming soon* or the *not yet available* one level back, on the page with the link, so as to avoid wasting your visitor's time accessing a useless page.

Strive for visual consistency

Give your pages a consistent design, so your visitor maintains a sense of continuity while at your site. Apply typographic wisdom: if you center some elements on a page, center all the elements.

Don't put blue text on a black background

Remember the ransom-note days of desktop publishing, when people thought that because they had access to forty fonts, they had to use all of them in a single document? Thankfully those days have passed in DTP, but Netscape has implemented codes that allow an HTML document to specify the colors it wants to be displayed in. Don't succumb to ransom-note color choice. If you are not competent to choose colors, don't choose: let them default.

If you choose your own colors, choose a light background color and dark text: the larger the differential between these two, the better the contrast ratio, and the more legible your page. I consider it a mistake to use a black background, because color CRT displays have much higher contrast ratio for black text on white than for white text on black. You must choose link colors carefully: many people who set Netscape to use black background and white text neglect to change the link colors. Their links display dark blue, which renders them virtually invisible against the black.

You should be aware that web technology cannot yet guarantee accurate color reproduction across different platforms, so you have no guarantee of consistent color.

If you decide to use a background image, you should do so only with a good understanding of graphic issues. A poorly-chosen background image can destroy the readability of your pages.

Restrain yourself from using `<BLINK>`. None of us will benefit if we turn the web into a poor imitation of Las Vegas, and your visitor is unlikely to be impressed by a page that is reminiscent of those *TAKE A COUPON!* blinking lights in sleazy supermarkets.

Compose to HTML, not to a particular browser

Netscape browsers dominate the web at the moment, and it is tempting to compose HTML that looks good when viewed with Netscape. You may find that Netscape's Heading 1 lines, in `<H1>` style, are displayed too large. You may choose Heading 3 `<H3>` instead. I have fallen victim to this temptation, but it's a bad idea.

HTML is designed to encapsulate the structure of a document, leaving the presentation to the browser. If you tune a document to a particular browser, your page is almost certain to appear a mess to a different browser. Even if your visitor is using the same browser that you use, if he has customized the fonts and sizes in his browser, your document is likely to be poorly presented.

We can expect browser capability to improve, but it is unlikely that you will be inspired to go back and retune your pages. If you stick to the standard

HTML structure, your pages will look no worse today than anyone else's, and they will look better and better as browsers improve. If you tune your pages, today they will look better some of the time and worse some of the time, and they will age very poorly as browsers improve.

If you have a document that begs to be presented typographically, consider distributing it in Acrobat PDF format instead of – or in addition to – HTML. Acrobat Reader is freely available for the major platforms: Windows, MS-DOS, Mac, and UNIX. Acrobat Exchange integrates well with the web.

Use 7-bit character set and “escaped entities”

The assignment of *glyphs*– or shapes – to character codes between 0 and 127 is established by the ISO 646 standard, which is essentially the international version of ANSI X3.64 (*ASCII*). This standard guarantees that *7-bit* codes produce the same glyphs on different platforms.

The ISO 8859-1 *Latin-1* standard conforms to ISO 646 for codes 0 to 127, but assigns additional glyphs – mainly accented characters – to codes in the range 128-255. The Macintosh and Windows operating systems do not respect the ISO 8859 standard, so codes in the range 128 to 255 produce different glyphs when transported between these platforms. Most applications pay no special attention to character sets, and inherit the character set native to the underlying operating system.

Web technology allows transport of *8-bit* characters coded according to the ISO 8859-1. Web browsers implement platform-dependent translation so that 8-bit characters received in a web page are displayed correctly. Some browsers have an option

setting to enable the translation; Netscape 1.1 for the Mac comes with a setting that is not ISO Latin-1. Set your browser to conform to the standards of the web: Set its character set to ISO Latin-1.

Few text editors implement the ISO 8859-1 character set directly, so creation of web pages using characters in the range 128-255 is difficult. If you make a web page using a text editor that allows insertion of codes in the range 128-255, you have two options: You must either take care to avoid or remove characters in that range, or you must arrange to have those characters translated.

If you remove characters in the range 128-255 by *stripping the eighth bit*, the result is guaranteed to comprise just 7-bit ASCII characters. But in stripping the eighth bit, you may inadvertently turn characters into ASCII codes that you don't intend. On a Macintosh, if your document uses a *bullet* character (•, Option-8), it will turn into a percent sign. It is a better idea to translate, and many utilities are available to translate from a platform's native character code to ISO 8859-1.

Although 8-bit characters are handled well by the web browsers, transport of 8-bit characters by other means – e-mail, ftp and physical media – remains problematic. In HTML there is provision to convey accented characters and other characters of ISO 8859-1 using *escaped entities* that comprise an ampersand, a short sequence of 7-bit ASCII letters, and a terminating semicolon. I recommend that instead of translating to 8-bit ISO 8859-1 you translate to 7-bit ASCII with the escaped entities. This will assure that your pages are transported easily and displayed correctly on any conformant browser.

A few important characters are not accommodated by ISO 8859-1. The most glaring omission is typographic (*curly*) quotes. Your translator will turn these into straight quotes. The trademark sign (™) is absent from ISO 8859-1. Provision has been made in HTML 3.0 for an escaped entity *™*; but most browsers in use today do not conform to HTML 3.0 and would display *™* instead of the symbol that you want. Write that one out, (tm).

A handful of escaped entities are not handled properly by Macintosh browsers: avoid the superior figures, fractions, *y*-acute, *thorn*, *eth*, and the so-called *times* symbol. If you don't know what these are, you're probably not using them!

Establish a context for the reader (and for the robots)

Make sure that the first several lines of text on your page describe the content of that page.

You can include in a web page a link to any other page on the web; part of the power of the web lies in jumping from site to site. But the flip side of this situation is that your page may be accessed from places different from what you anticipate. By providing a short outline of the content of your page, you establish the context for a visitor who has come to your page from somewhere else.

There is another reason for the description to be short, and to be located at the top of the page: Many automatic programs – the crawlers, wanderers, robots, harvesters and spiders – traverse the web, extracting and indexing pages. Many of these programs index all of the words in a page, but save only the first several lines for display in a search result. In order for the user of a search service to recognize your page as useful when it is returned as

a search result, you need a useful description in the first few lines.

You will find many web pages that have adopted cutesy elements like spaces between the letters of the page title. People do this in an attempt to create a distinctive look, and sometimes it succeeds in attracting the viewer's attention. On the other hand, it defeat the robots' attempts to index the page. If potential readers never access the page, what good is a distinctive style?

Provide a page title

Include a title – the <TITLE> element – on every page. Limit your title to about 40 characters, to avoid overflowing the your visitor's screen width. Help your visitor to navigate by making the structure of your titles consistent among your pages. The search engines usually display the page title along with a search result. If your page has no title, it will be displayed alongside a message like *No Title Provided*, which makes you look unprofessional.

Plan for an international audience

The first *w* in WWW is for *world*. Expect the audience for your web pages to be international.

If you write a date in the form 08/04/50, will your visitor think it April or August? In the next century, will 01/02/03 be the first, second or third day of the month? Banish this confusion for once and for all by writing dates in the ISO/IEC 8824 form, 1996-06-17.

To respect my international colleagues, in front of any telephone number I place a plus sign and the country code, for example, +81 for Japan or +44 for the UK. The country code +1 indicates the United States, Canada, Mexico, and parts of the Caribbean.

Delimit the area code – or, in other parts of the world, the city code – using spaces instead of parentheses. Parentheses are not particularly computer-friendly, and many people handle telephone numbers using computers.

Write out only digits that are properly part of the telephone number: Omit local access codes. In particular, do not indicate the access code 0 in front of a city code used in Europe: people who need it know to dial it, but if a person unfamiliar with the convention dials the zero, his call will fail. In some countries, such as Russia, the digit zero can be a legitimate part of a city code, and must be dialed.

Keep web pages small

Many people use the web through dialup modem connections capable of transfer rates of only 1000 or 2000 characters per second. If you link to an exceptionally large page, larger than 50 KB or so, you should provide at the point of a link an indication of the size of the referenced object. Your home page, including its images, should be no larger than this.

If a link accesses an ftp file, then provide at the point of the link an indication of the format of the file and the size of the file (no matter how small). This indicates to the reader that accessing the link will transfer the file. Avoid notations like download here and download now, for the same reasons that you avoid click here.

If you link to an ftp *directory*, as opposed to a file, include a trailing slash at the end of the URL. This indicates to your visitor – and to his web browser or ftp client – that the item is a directory.

Code images correctly

Include WIDTH and HEIGHT information in image (IMG) links. This allows a browser to complete page layout before accessing the image, and avoids flashing due to re-layout. Choose the WIDTH and HEIGHT of the actual image file; do not arbitrarily choose them expecting the browser to scale the image, because not all browsers have that capability, and in any case a scaled bitmap reproduces poorly.

If your image forms part of a link, include an ALT tag describing the image in words. You will be thanked by visitors without image display capability, and by visitors who have disabled image display (perhaps for reasons of speed).

If you have an inline image, make it small (10 KB or less), and save it in GIF format (until PNG format is widespread). If you want to provide for your visitor an image larger than that, make a small GIF version of it – a *proxy* – and place the proxy on your page. Make the proxy a link to the large image. If the large image is full color or continuous-tone, save it in JPEG/JFIF format.

You can process a GIF bitmapped image so as to make some of its pixels transparent. The opaque pixels will then be displayed against the background color that was chosen by a preference set in your visitor's browser. If your visitor has a modern browser and you have specified the appropriate codes in your HTML, it will display against a background that you have chosen. If you choose to specify transparency, be aware that the less-sophisticated browsers will display your image entirely opaque. Choose a background color appropriate for those browsers, say [192, 192, 192] for a light gray. If you use a custom background color or image, be

aware that it will be ignored by less-sophisticated browsers.

Sign your work

It is frustrating to find a page on the web whose authorship is unknown, especially when there are no other links on the page to establish where it lives or what it relates to. Sign your pages.

If a user comes to a page from a foreign link, give him the opportunity to explore your home page or the rest of your pages: make your signature a link, direct or indirect, to your home page.

At the bottom of every page, my signature is a link *up* within my tree of pages. For a page other than the *index.html* file in a directory, I place a signature that names the directory and a link to *index.html* in that directory. At the bottom of each *index.html* file I refer to the title of the next level up, and place a link to *../index.html*. This enables my visitor to ascend the whole tree back to my home.

At the bottom of my home page, my signature is a MAILTO link. If my visitor hasn't discovered the information he wants in his traversal of my pages, this invites him to send e-mail to me.

I include at the bottom of every page the date that I last modified the page.

Conform to server conventions

Learn from your Internet service provider how to make your files accessible to his web server.

If you use a UNIX server, include the lowercase *L* at the end of the *.html* extension when you transfer, even if your local filenames are limited by MS-DOS

or Windows. Use UNIX (LF) line ends in text files (including HTML) stored at a UNIX server.

Make sure every directory has a file *index.html*. If you do not do this, then a visitor who manually enters the path to a directory will be presented with a list of all of the files in that directory, perhaps including some files that you do not want to advertise.

Your HTML pages include whatever file names and paths you need for your links. The robots and wanderers will harvest filenames from your HTML code, and add these referenced files to their indices. If you want a file to be indexed, you should include its name in another file that is indexed already: The robots will eventually find your new page!

You can place in your web directory a file whose name is *not* referenced in any of your pages. The robots will not discover this file. But if a visitor guesses a name, *index.bak* or *index.old* for example, there is no method to prevent the visitor from retrieving that file. The only way to be absolutely certain that a visitor will not have access to a file is to remove that file from your web directory.

Maintain the hierarchy

My home page is accessible at the URL `<http://www.inforamp.net/~poynton/>`. If I wish to direct someone to a page other than my home page, say by e-mail, I specify the full URL of that page: my page of Macintosh information is located at `<http://www.inforamp.net/~poynton/Poynton-mac.html>`. However, within my home page, I use a relative pathname such as `Poynton-mac.html`. Using a relative pathname makes it easier for you to maintain pages and links, and makes it easier for your

visitor to make local copies of your pages while maintaining the function of the links.

If you have created a hierarchy of pages, the easiest way to manually create a new page is to copy, then edit, a page at the same level of the hierarchy.

Choose filenames that are mnemonic. When a visitor decides to save one of your pages, the name you choose will be presented as his default name.

Once you've chosen the name of a file (or page or directory), stick to it. Other sites may have made links to your page (or directory). If you change a name, you will break those links.

Test locally

Your pages will be no pleasure for your visitor if they do not work as you yourself intend. Make sure that your pages work for you before you subject someone else to them!

Test your pages locally, use the *Open File* capability of your favourite browser. Use two or three different browsers, to see how they present things differently. Test your pages in black-and-white, to preview how they will appear to a user who has only black-and-white display capability.

Validate

When you have finished making a page, that you run it through an HTML validation service to ensure that it conforms to the technical requirements of HTML. If you do not do this, you cannot be sure that it will work reliably on other browsers and other platforms than yours.

If you have manually created your HTML, you can fix it by hand. If you have used automated conversion tools, you may have little scope to repair failures in validation. In this case, take the validation report to the provider of your conversion tools.

Provide alternate access paths

The online services such as AOL and CIS are slow to adapt to new technology, and millions of their subscribers have no access to the web. If you want to make information available to online service subscribers you must use *file transfer protocol (ftp)*. Modern web browsers accommodate ftp access, so your files will be available not only to those subscribers, and to ftp users on the Internet, but also to web users on the Internet. You will need your Internet access provider's help to set up an ftp directory at his site.

My technique is to store the bulk of my information in my ftp directory, and to use the web as an attractive, functional and interactive interface to that information. All of the common web browsers include ftp capability as a subset, so your web visitor will see no break in continuity but your information is stored only once, and CIS and AOL visitors have full access. My web pages simply contain links that use ftp protocol to access directories and files. You can view the source (HTML) of my pages to see how these links work.

When I create new information to publish, I store the files themselves in my ftp directories. Then I create a web page with the descriptions and links. I open this page in my web browser and Save As ASCII text-only format. I use that text file as the README file for the ftp directory.