**Website Checklist**

So your new web-site looks wonderful. Nice colours, gorgeous images, lots of interesting content. And the Javascript is finally debugged. So it’s ready for public viewing? Well, based on a lot of the web-sites I’ve seen, probably not yet!

Here is a checklist of things to run though before going public with your web-site. Most of them are things which seem to have been overlooked by large numbers of web developers.

This checklist is oriented towards matters which can be – more or less – objectively determined. It does not attempt to cover matters which are very subjective, such as clarity of layout or text, though these are of course just as important. It covers only matters which affect the readability of your site – not those which affect how easy it will be for you to maintain it.

A useful resource to help with further checking of your site, including the more subjective matters, is the alt.html.critique Usenet newsgroup.

**The Checklist**

The links in the checklist lead to a discussion of each issue.

1. Validation
   1. [Validate the HTML](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#html)
   2. [Validate the CSS](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#css)
   3. [Check for broken links](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#links)
2. Flexibility
   1. [Try varying window sizes](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#windowsize)
   2. [Try varying font sizes](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#fontsize)
3. Speed
   1. [Access the site via a modem](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#modem)
   2. [Check image size specifications](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#imagesize)
4. Accessibility
   1. [Test accessibility](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#access)
   2. [View in text browser](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#text)
5. Browser independence
   1. [Try different browsers](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#browsers)
   2. [Check printed pages](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#print)
   3. [Switch Javascript off](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#javascript)
   4. [Switch plug-ins off](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#plugins)
   5. [Switch images off](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#images)
6. Other checks
   1. [Check non-reliance on mailto](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#mailto)
   2. [Check no orphan pages](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#orphan)
   3. [Check sensible page titles](http://www.xs4all.nl/~sbpoley/webmatters/checklist.html#title)

**Validation**

**Validate the HTML**

The first stage of checking a web-site is to make sure that you have valid HTML (or XHTML). This can be done with a validator such as the [W3C validator](http://validator.w3.org/) or [WDG validator](http://www.htmlhelp.com/tools/validator/). Your own browser may ignore certain errors, but there is a significant risk that markup errors will result in display problems in some browser or other.

There are sometimes reasons for including invalid HTML. For example some pages on my own site use the non-standard NOINDEX element, for the benefit of my site search engine. But you should only tolerate HTML validation errors if there is a clear need for the non-standard markup, and if you have tested the result in a variety of browsers.

**Validate the CSS**

CSS can be validated with for example the [W3C CSS validator](http://jigsaw.w3.org/css-validator/validator-uri.html). The considerations here are much the same as HTML validation. It may sometimes be necessary to use something non-standard to get Internet Explorer to work, but such rules can be placed in a separate CSS file and hidden in an [Internet Explorer conditional comment](http://msdn.microsoft.com/workshop/author/dhtml/overview/ccomment_ovw.asp), where they won’t bother other browsers or a validator.

**Check for broken links**

Obviously you don’t want broken links on your site. There are various tools available to help find these, such as the [Link Valet](http://valet.webthing.com/link/) (which is convenient for checking a few pages) or [Xenulink](http://home.snafu.de/tilman/xenulink.html) (convenient for checking a whole site).

**Flexibility**

**Try varying window-sizes**

A very important aspect of web design is coping with different window sizes. Window widths may vary from about 200 pixels on a web-enabled telephone to 2000+ pixels on a technical workstation in full-screen mode. While providing a readable site to the very smallest screens is something of a challenge, your site should at least be readable on a wide variety of sizes. As of mid 2006, nearly 20% of all readers are still using screens of 800x600 pixels or smaller, and if the reader wishes to compare the contents of your site with another document, it is entirely possible that he/she may want to use a window-width of around 400 pixels.

Fortunately, at least as far as the text goes, this is not very difficult – just refrain from specifying sizes in pixels or points and you are most of the way. See my [flexible design](http://www.xs4all.nl/~sbpoley/webmatters/flexdesign.html) page for further thoughts.

It is obviously easy to test window-sizes smaller than your own screen-size. Testing larger window-sizes might seem impossible, but you can do a rough simulation using the zoom facility of the Opera browser – zoom down to 50% and you get a screen twice the size. It may not be very readable, but any major layout errors should be obvious.

Incidentally don’t worry too much about the very long lines of text that appear at large screen sizes when small fonts are used. If the reader doesn’t use a large font, he can always reduce the window size to something comfortable – that is, after all, half the point of having windowing user interfaces in the first place. But if you wish to, you can also use the CSS2 ‘max-width’ property to limit column width, just as this page does. (Only discerning readers are currently able to benefit from this, as IE does not support it. Rumour has it that it will finally appear in IE 7 – I suppose eight years late is better than never.)

**Try varying font sizes**

Some people use large screen fonts because they have a large screen with a very fine resolution; other people have to use large screen fonts because of declining eyesight. On the other hand, some people like to use microscopic fonts with their nose pressed against the screen (it takes all sorts...)

So while doing the above activity, adjust the default text-size in your browser, and check that all the text scales appropriately. (You *did* specify the text size in ems or %, didn’t you? If not, see my [font-size page](http://www.xs4all.nl/~sbpoley/webmatters/fontsize.html).)

One other aspect to consider is that users may impose a minimum font size to make sure that all text is readable for them. This means that font sizes show a smaller range than you had in mind. If you have a complex page with a wide range of font sizes, it would be worth imposing a minimum size larger than your smallest font (this can be done in e.g. Opera or Firefox) and checking that this does not make parts unreadable.

**Speed**

**Access the site via a modem**

So you think you have a great site, full of beautiful images. Put your site on the server, then dial in via a modem (a real modem – not an ADSL gateway, which is sometimes erroneously referred to as a modem). Does it look so good half a minute later? Or are you still staring at a pretty-much blank screen?

If the opening page of your site takes more than about 15 seconds to appear, then you are losing visitors fast. (Broadband users generally expect a still faster response.) Don’t overload it. If you have to include large objects on your site – perhaps it revolves around high-resolution reproductions of fine art – put them on later pages and tell (or warn?) your users what is coming.

Web developers generally have broadband access, and they sometimes forget that the majority of the world's internet connections still run via a modem. Yes, more and more people are getting broadband, but the number of people without it is still large, and many people in rural areas, let alone developing countries, simply can’t yet get broadband. By the way, don’t make the mistake of assuming that you don’t have to worry about bandwidth issues if your site is mainly aimed at companies. I have more than once worked at companies which did have a broadband connection, but had it shared between so many users and applications that the net result was scarcely faster than a dial-up modem.

**Check image size specifications**

While doing the test above, check that at least the text of the page appears quickly. If it doesn’t (or if it dances all over the place while the page is loading) it is probably because you have forgotten to specify HEIGHT and WIDTH attributes on some of your images. Specifying these enables the browser to work out where the text needs to go, and to display it before the images have finished downloading.

**Accessibility (for the disabled)**

Since I first wrote this page, it has been pointed out to me that actually most of the page is about accessibility: that is, ensuring that a web page is accessible under a wide range of browsing conditions. However accessibility is often used in the narrower sense I use in this section.

**Test accessibility**

This is mainly important for handicapped users, but also relevant for e.g. people who use a text-only browser, or disable images, because of a slow connection. See the [Web Content Accessibility Guidelines](http://www.w3.org/TR/WCAG10/).

Automated accessibility checking does need to be taken with a pinch of salt. Many aspects of the guidelines require human judgement to say whether a page is accessible or not – for example whether HTML Heading tags are used correctly. And even when the guidelines are unambiguous, you don’t need to follow them slavishly. For example the absence of a caption on a table is unimportant if the previous paragraph explained what the table is about. Nonetheless it is well worth running a few pages through a checker such as [Bobby](http://bobby.watchfire.com/bobby/html/en/index.jsp) or [Accessibility Valet](http://valet.webthing.com/access/) in order to familiarise yourself with the issues involved. You can then make the necessary improvements.

**View in text browser**

It is also worth running pages through a text-only browser, or [text-browser emulator](http://www.delorie.com/web/lynxview.html) to see what e.g. a blind person using a text-to-speech converter will encounter. It will help you pick up on badly-chosen or missing [ALT texts](http://www.xs4all.nl/~sbpoley/webmatters/alt.html) for example. It also shows you the site pretty much as a search engine will see it. Incidentally the Opera browser has a built-in text-browser emulator.

**Browser independence**

Your site may be viewed in a large variety of situations: different browsers, different operating systems, different features enabled or disabled. It is important that your site stands up well in these different situations. The first point of attention here is validation – described separately above. Then there are the following points.

**Try different browsers**

Almost all web developers (ahem! – perhaps that should read “quite a lot of web developers”) are aware of the need to check how their site looks in a variety of browsers. How far you go obviously depends on the resources available – not everyone is in a position to check Windows, Mac, Unix and Linux platforms. The minimum test would probably be:

1. Firefox, as that has the best standards compliance and is the second most-used browser;
2. Internet Explorer for Windows – currently the most widely used browser. It is essential to check both versions 6 and 7 as version 7 fixed quite a lot of bugs in 6 but introduced a new set of its own. (Microsoft is however still kicking developers in the teeth by not making it possible to install both versions on the same computer; you will either need two computers or one of the work-arounds available on the net.) Version 5 should preferably also be checked; as of spring 2008 the number of users is not yet negligible. However it is now uncommon enough that you needn’t worry about cosmetic issues; as long as the site is readable that should be sufficient.
3. Opera – growing in popularity due to its speed and pretty good standards compliance.

For some time I also recommended checking Netscape 4 as well, as it often produces radically different results from any other browser, and was very popular for a long time. However the number of users of this bug-ridden browser is now so small (under 0.1% and decreasing) that it can now probably safely be ignored.

**Check printed pages**

Print some of the pages on a normal printer (i.e. with a paper size of A4 or Letter) and check that they appear sensibly. Due to the somewhat limited formatting options available for printing, you probably can’t achieve an appearance comparable to a document produced by a word-processor, but you should at least be able to read the text easily, and not have lines running off the right-hand side of the page. It is truly extraordinary how many site authors fail to think of this most elementary of operations.

You should also consider using CSS to adjust the appearance of the page when printed. For example you could – probably should – suppress the printing of information which is not relevant to the printed page, such as navigation bars. This can be done using the “@media print” or “@import print” CSS features.

Some sites provide separate “printer friendly” versions of their pages, which the user can select and print. While this may occasionally be necessary as a last resort, it significantly increases the amount of work needed to maintain the site, is inconvenient for the reader and shouldn’t usually be needed.

**Switch Javascript off**

There are unfortunately quite a number of Internet sites which abuse Javascript by, for example, generating unwanted pop-ups and irritating animations. There are also a number of Javascript-related security holes in browsers, especially Internet Explorer. As a result a lot of readers switch Javascript off – indeed I often do myself. (I have a page giving the [reasons](http://www.xs4all.nl/~sbpoley/webmatters/whatnojs.html) in more detail.) Some organisations even block the usage of Javascript completely. Furthermore few, if any, search engines support Javascript.

It is therefore important to check that your site still functions with Javascript disabled. A lot of sites rely – quite unnecessarily – on Javascript for navigation, with the result that the lack of Javascript renders the site unusable.

Clearly if you need it for essential content, that functionality will be lost. But there is no reason why the basic text of the site should be unavailable.

Avoid nearly-meaningless messages like “Javascript needed to view this site”. If you have something worth showing, tell the user what it is, e.g. “enable Javascript to see animation of solar system”.

**Switch plug-ins off**

The considerations for plug-ins (such as Flash or Java) are very similar to those for Javascript above. Check the site with any plug-ins disabled. The basic text and navigation should still work.

Interest the reader sufficiently, and he might just go to the trouble of down-loading the plug-in. Greet him with a blank screen or a “You need Flash to read this site” message and he will probably go away, never to return.

**Switch images off**

If scanning a number of sites quickly for information, many readers (including myself) switch images off, for quick loading. Other people cannot view images. So switch images off and check that the site is readable and navigable. This means, in particular, checking that sensible ALT texts have been provided for images. (This check is similar to using a text browser, but not quite the same).

**Other checks**

**Check non-reliance on mailto**

In order to give readers a chance to contact them, web authors often include a link of the form “mailto:abc@xyz.org”. However this unfortunately does not work for anything like all browser/e-mail client combinations. And people in e.g. an Internet cafe cannot use this type of link. Indeed the figure is sometimes circulated that as many as 50% of readers cannot use mailto links. While I doubt whether the true figure is quite this high, it is a substantial number. Many readers prefer a form anyway: most of the responses I get to this site come via the contact form rather than the mailto link.

Therefore the best thing is to provide a contact page which has both a mailto link and a contact form; the user can then choose which to use. See my own [contact page](http://www.xs4all.nl/~sbpoley/webmatters/contact.html) as an example.

**Check for orphan pages**

An orphan page is one that contains no links back up to its ancestors – i.e. to pages higher in the hierarchy of the site. Once one arrives at an orphan page, the only way to get to the rest of the site is via the ‘back’ button. Which is fine until people arrive at the page via a search engine, or via a link that someone else gave to them. They cannot then visit the rest of the site. (True, they *may* be able to get up the hierarchy by lopping off the end of the URL, but this depends on how the site is built, and is in any case not reader-friendly.) So ensure all pages include a link back up the hierarchy.

Orphan pages are particularly easy to overlook in sites with frames. Remember that when one addresses the page directly the other frames are absent.

**Check sensible page titles**

Check that the page titles (i.e. the contents of the <TITLE> elements) are sensible. Page titles are important, as for example they show up prominently in search-engine results, in bookmarks and also on the tabs of multi-tab browsers such as Opera. Generally speaking, each page of a site should have a unique title. They seem to be somewhat prone to the dreaded cut & paste disease. The “grep” tool is convenient for quickly checking the titles in all your page source files.

A quick web search reveals over a million pages with the illuminating title “New Page 1”, while “Untitled” pages run into several millions.