

Best Practices for Web Development

The following guidelines are based, in part, on notes that I compiled for orientation sessions for new developers at Navigata Communications. These guidelines are certainly not exhaustive and are definitely subject to change.

General

Use valid (X)HTML

Do not use deprecated tags and avoid the use of browser-specific features. Where a justification can be made for using browser-specific tags or features, document appropriately. Please avoid using tables for layout. The use of `<td rowspan= whatever >` is grounds for a savage beating. Before going live, please check your HTML for validity.

Further reading:

<http://validator.w3.org>

<http://www.w3.org/TR/REC-html40/index/elements.html>

http://www.w3schools.com/html/html_reference.asp

Use descriptive directory names in URLs, not filenames

Directories provide a means of grouping information and contribute to better site architecture. This approach also allows you to migrate and combine technologies without advertising this fact. An exception to this rule is when you are presenting non-HTML information (i.e. PDFs). In this case, feel free to use filenames. In directory names, a hyphen is preferable to an underscore for separating words.

Poor: www.example.org/products.html

Better: www.example.org/products/

Further reading:

<http://www.w3.org/Provider/Style/URI.html>

<http://www.alistapart.com/articles/slashforward/>

<http://forums.seochat.com/t9981/s.html>

Test on multiple browsers

At a minimum, please test on the following platforms:

Windows: MSIE 6.0+, latest versions of Opera and Mozilla

Mac: latest version of MSIE and Safari

Further reading:

http://www.w3schools.com/browsers/browsers_stats.asp

Do not use `click here`

`Click here` is bad linking practice as it impedes the user from visually scanning the page for keywords and themes. It also presupposes that your users do not understand how to follow a link.

Poor: [Click here](#) for more information on Widget Co. Hoodingies

Better: [More information on Widget Co. Hoodingies](#)

Further reading:

<http://www.cs.tut.fi/~jkorpela/www/click.html>

Never hard-code variable dates

A surefire way to make a site look out-of-date is to have hardcoded dates in things like copyright notices, ranges for credit card expiry, etc. Whether done it server-side or client-side, a date that can change should be set to change automatically.

Do not use graphical representations of text, use text instead

This is particularly important for site navigation, as adding or modifying navigation elements should be a text-only exercise. By presenting text as graphics, you are potentially affecting accessibility, usability, and search engine indexes. An exception applies when the text is part of a promotional or header image. In such a case, the main points put forward in the image should be repeated as plaintext somewhere on the rest of the page.

Further reading:

<http://www.w3.org/TR/WCAG10-CORE-TECHS/#text-equivalent>

Do not create separate, printer-friendly pages

A printed Web page does not require header and footer elements such as navigation. A well-built page should use CSS to suppress the printing of these elements.

Further reading:

<http://www.alistapart.com/articles/goingtoprint/>

Programming

Create functions that return stand-alone blocks of HTML

If a div or table is opened in a function, it should be closed in that same function. Exceptions apply for example, a div opened in a header can be closed in a footer.

Pass objects, not object members

By passing an object, you get access to its members and methods. This encapsulation is one of the major advantages of object-oriented design. By passing an object's member data, you break encapsulation, which makes Baby Jesus cry.

Poor: `$someList->getBySomeObj($obj->id);`

Better: `$someList->getBySomeObj($obj);`

Further reading:

<http://www.hamsterific.com/JAVA/aaa-TIJ3-distribution/TIJ-3rd-edition-html/TIJ320.htm>

Never trust user data

Always validate data provided by a user, whether via POST, GET, or cookies. Check length and type of data and escape it before adding to a database.

Further reading:

<http://www.microsoft.com/mspress/books/sampchap/5612a.asp>

Avoid the use of sessions, maintain state via database records instead

Sessions do not scale well across multiple Web servers. Storing an object in a session does not make sense if that object's member data (i.e. objects responsible for error handling or database connectivity) are intended to be stateless.

Further reading:

http://www.acros.si/papers/session_fixation.pdf

<http://www.fawcette.com/Archives/premier/mgznarch/vbj/1998/02feb98/id0298.pdf>

Recognize differences between front-end and back-end development

Front-end development, where possible, should feature a scripting language embedded within HTML. Back-end development, which should be standardized and make use of auto-generated code, can feature HTML embedded in a scripting language.

Make error messages meaningful and do not use exclamation points

Your users are already confounded and annoyed by the fact that your application isn't working for them. Do you honestly think that returning "Can't open file!!" is going to make their day better?