From EULER Project

PmWiki: Pm Wiki Philosophy

This page describes some of the ideas that guide the design and implementation of PmWiki. Patrick Michaud doesn't claim that anything listed below is an original idea; these are just what drive the development of PmWiki. You're welcome to express your disagreement with anything listed below. PmWiki.Audiences also describes much of the reasoning behind the ideas given below.

1. Favor writers over readers

At its heart, PmWiki is a collaborative authoring system for hyperlinked documents. It's hard enough to get people (including Pm) to contribute written material; making authors deal with HTML markup and linking issues places more obstacles to active contribution. So, PmWiki aims to make it easier to author documents, even if doing so limits the types of documents being authored.

2. Don't try to replace HTML

PmWiki doesn't make any attempt to do everything that can be done in HTML. There are good reasons that people don't use web browsers to edit HTML--it's just not very effective. If you need to be writing lots of funky HTML in a web page, then PmWiki is not what you should be using to create it. What PmWiki does try to do is make it easy to link PmWiki to other "non-wiki" web documents, to embed PmWiki pages inside of complex web pages, and to allow other web documents to easily link to PmWiki.

This principle also follows from the "favor writers over readers" principle above--every new feature added to PmWiki requires some sort of additional markup to support it. Pretty soon the source document looks pretty ugly and we'd all be better off just writing HTML.

Another reason for avoiding arbitrary HTML is that ill-formed HTML can cause pages to stop displaying completely, and arbitrary HTML can be a security risk--more so when pages can be created anonymously. See http://www.cert.org/advisories/CA-2000-02.html(approve sites) for more information.

3. Avoid gratuitous features (or "creeping featurism")

In general PmWiki features are implemented in response to specific needs, rather than because someone identifies something that "might be useful". In any sort of useful system, it's hard to change a poorly designed feature once people have built a lot of structure based on it. (Need an example? Look at MS-DOS or Windows.) One way to avoid poor design is to resist the temptation to implement something until you have a clearer idea of how it will be used.

4. Support collaborative maintenance of public web pages

Although this wasn't at all the original intent of PmWiki, it became quickly obvious that WikiWikiWeb principles could be used to make it easier for groups to collaboratively design and maintain a public web site presence. PmWiki allows individual pages to be password protected, and a couple of local customizations makes it easy to protect large sections of PmWiki pages. Furthermore, in many ways PmWiki provides "style sheets on steroids": you can quickly change the headers, footers, and other elements on a large group of pages without ever having to touch the individual page contents. Finally, it's relatively easy to add custom markup for specialized applications.

5. Be easy to install, configure, and maintain

With a gzip-compressed file size of just around 200K, uploading PmWiki to your server is a speedy operation. Do a chmod or two, update a few settings in config.php and you should be up and running. PmWiki stores all data in flat files, so there is no need for MySQL or other utilities. Upgrading is usually a simple matter of copying the latest version's files over the files of your existing PmWiki installation. (One of the biggest reasons for the creation of PmWiki was that other wiki engines at the time required modifications to the distribution files, so admins ended up losing their customizations on every upgrade.)