- [What is a skin?](#)
- [Where do I get skins?](#)
- [How do I use a skin?](#)
- [How can I modify an existing skin?](#)
- [How can I make a skin?](#)

A skin changes the look and feel of a [Pm Wiki](#) page, Group of pages, or the entire wiki.
To see this try some skins out using the links below.

- [BeeblebroxNetGila](#)
- [JHSkin](#)
- [GeminiSkin](#)
- [LightSkin](#)
- [Monobook Skin](#)
- [Simple Skin](#)
- PmWiki (default) Skin

As you see, all skins show the same page contents, but the other elements such as the sidebar, header, and footer, have changed.
For example, different skins may display the sidebar on the left, on the right, or even not at all.
Some skins have action links and features that others do not, especially if they were designed to take advantage of particular [cookbook recipes](#).

So, a skin is just the set of files that determine how pages are displayed in Pm Wiki. Skins are stored as subfolders of *pub/skins/*. For example you might create the *example* skin in *pub/skins/example/*. Each skin typically has one or more of the following kinds of files:

- A template file, such as *skin.tmpl* or *example.tmpl*. The template is written in HTML or XHTML, and is the skeleton for the skin. It contains special markers that tell Pm Wiki where to insert the page's contents.
- CSS stylesheet files, which can control the skin's appearance, such as *pmwiki.css* or *example.css*.
- Image files, for decorating a page with images.
- PHP files, such as *skin.php* or *example.php*. These let skins provide extra customization setting or capabilities that HTML and CSS alone cannot.
- Documentation files for the administrator, usually with names like *readme.txt*, *install.txt* or *skinname.txt*. These usually give you information about any special installation steps or nifty features the skin has.

Skins are available in the [Cookbook:Skins](#) collection.
The skins in the collection have been contributed by many Pm Wiki administrators for all to use, and typically have their own set of customization possibilities.
When you find a skin you like, follow the link to download the skin package.
You can also [make your own](#) skin.

Most skin packages are .zip, .tgz, or .tar.gz files. You should be able to unpack these with most archiving software.

1. Unpack the skin to *pub/skins/* inside your pmwiki folder. Most well-designed skin packages will create a subfolder in *pub/skins/* named after the skin.
   º If the skin did not make a folder of its own, create one and move the skin files to it.

- Open up your *local/config.php* file, and add a line like

```
$Skin = 'example';
```

where *example* is the name of the skin's folder.

Reload a page from your wiki in the browser, and you should be able to see the difference.

If you'd like to let your site's visitors choose one skin from a selection of skins you've installed, look at the [Cookbook:Skin Change](#) recipe. (That's what we used for the demo above.)

There are a number of ways to further customize the appearance of a skin, including

- adding statements to /local/config.php that are compatible with your chosen skin;
- adding css files to /pub/css/, such as local.css (for your entire wiki) and [My Group?](#).css (for [My Group?](#)); and
- directly editing the skin's files.

If the skin is updated regularly, you probably will want to avoid editing the files in the skin's folder. Check the skin's page in the Cookbook for specific suggestions.

If you want to modify the default pmwiki or print skins included with the Pm Wiki distribution, you should copy the pub/skins/pmwiki/ and pub/skins/print/ directories to another name and then use those skins instead of the default ones. While the name of the skin.tmpl and skin.css files don't usually matter, the optional skin.php file MUST match the name of the skin.

The best way to make your first skin is to modify a copy of Pm Wiki's default skin.

1. Make a copy of the folder *pub/skins/pmwiki* and name it whatever your new skin should be named.
2. In your *local/config.php* file, set $Skin to be the name of your new skin.
3. Modify the template and CSS files to suit you.
4. Test your new skin.
5. Repeat steps 3 and 4 until you're happy with the results.

The reason we recommend starting with the default Pm Wiki skin is that it's quite a simple skin, much more so than many of the skins you'll find in [Cookbook:Skins](#). The starting point is the template (.tmpl) file, which

provides the overall layout of the page. Inside of the template file are a number of special substitutions and directives that provide places for Pm Wiki to insert the data relevant to the current page being displayed. Skin Templates describes the format and directives in more detail. There are also skin guidelines available on pmwiki.org.

It's beyond the scope of this page to explain how to write HTML (hypertext markup language), XHTML (extensible HTML, which is a bit newer) or CSS (cascading style sheets), but there are many good tutorials on the web for all three of them. One caution: if you run into an HTML tutorial that explains about how to use or <blink> tags, or spacer gifs, it's at least five years out of date, so skip it and find another one.

You should test your skin on a variety of browsers -- ideally as many as you can, on as many different platforms as you can -- but at minimum you should be testing on Internet Explorer 8, Firefox 3, and Chrome, since those are the most common, and have different bugs, it is also useful to test on Opera and Safari. Don't forget to do things like resize windows and change text size during your testing.

Print Skins

By default your new skin will use the standard /pub/skins/print/ skin.

To over-ride this add the following to local/config.php:

```
$ActionSkin['print'] = 'yourprintskin';
```

This says to use 'yourprintskin' for ?action=print instead of the default.

Tools that you'll need

There are good examples of all these programs available for free.

HTML and CSS editor(s). There are two types of editors: graphical (WYSIWYG, or "what you see is what you get"), and hand-coding or programmer's editors. Graphical editors are less intimidating to novices, but you won't learn as much, or know your code as intimately as you will by using a hand-coding editor. Whichever you choose, get one that has syntax highlighting for the code, because it will help you spot mistakes. Also, live preview features are not that helpful when writing a Pm Wiki skin, because Pm Wiki does stuff that the live preview won't, such as substitute values for variables and insert sidebar content.

Test wiki. You don't want to be wreaking havoc on your skin while visitors can see your site. It's a better idea to set up a test wiki, either on your real webserver or on your own machine. Linux or MacOS computer owners may have webservers and PHP already running on their machines, but Windows users often don't. If that describes you, then you might want to take a look at the Cookbook:Standalone recipe, which runs Pm Wiki without needing a complex webserver, or Cookbook:InstallOnIIS. Or, you can find many local server packages which install a webserver, PHP, and other stuff (e.g. MySQL), all configured to work together. Try to get a package that has the same software and versions as used on your live setup, since then there will be less to go wrong when the site goes live.

FTP client to transfer files to your webserver. You probably had one of these already.

Color picker. Your editor might include one, or you could pick up a standalone application. Extremely helpful

for creating and saving color palettes.

See also

- [Skin Templates](#)
- [Cookbook:Skins](#)
- [Cookbook:Skin Guidelines](#)
- [Cookbook:Standalone](#)
- [Installation](#)

How do I change the Wiki's default name in the upper left corner of the Main Page?

Put the following config.php

```
$WikiTitle = 'My Wiki Site';
```

The *docs/sample-config.php* file has an example of changing the title.

How can I embed Pm Wiki pages inside a web page?

Source them through a PHP page, or place them in a frame.

How do I change the font or background color of the hints block on the Edit Page?

Add a CSS style to pub/css/local.css: `.quickref {background:...; color:... }`. The hints are provided by the [Site.EditQuickReference](#) page, which is in the Pm Wiki or Site wikigroup. Edit that page, and change the "bgcolor" or specify the font "color" to get the contrast you need.