



The Handbook

Codex

[http://codex.wordpress.org/Stepping Into Templates](http://codex.wordpress.org/Stepping_Into_Templates)

Version Date

8 August 2005

Stepping Into Templates

[Template files](#) are the building blocks of your WordPress site. They fit together like puzzle pieces to generate any web page on your site. Some are used repeatedly for every web page, like the header and footer template files, while others are used only under specific conditions.

A traditional web page consists of two files: the **HTML page** to hold the structure of the page and the **style sheet** which holds the presentation styles of the page. In WordPress, the style sheet is still present, but the HTML structure is broken up into various parts and pieces, blocks if you will, that can be put together in many different ways to generate a single web page. These blocks are known as **template files**.

Each [WordPress Theme](#) is made up of a style sheet and the building blocks or template files to generate the different page views found on a WordPress site.

Simple Page Structure

Let's look at a very simple web page structure. Most web pages feature a header, the content, and a footer. Three "blocks" that make up the whole of the page.

Header

Content

Footer

Inside of the **header** block or template file is all the information that needs to be at the *top* of your web page, such as the `doctype`, `meta tags`, links to style sheets, the start of the `body` tag, and the [header information](#) itself which typically includes the title of your site.

The **content** holds the documentation, articles, posts, the *meat* of your site.

The **footer** usually holds information like links to other [Pages](#) or categories on your site in a [navigation menu](#), copyright, contact information, and other details.

Building a Template Foundation

As with all building blocks, you have to start with a good foundation upon which to build. In WordPress, the core template file is the `index.php` found within your [WordPress Theme's](#) folder. The code inside the `index.php` template file begins the *call* to other blocks or template files and the [WordPress Loop](#) to gather information from the database to create the final web page.

In the `index.php` template file, the call for the header and footer looks like this:

```
<?php get_header(); ?>
<?php get_footer(); ?>
```

To add another block to our web page's structure, we could add a **sidebar** that includes more [navigation features](#) and information about your website.

```
<?php get_sidebar(); ?>
```

Header

Content

Sidebar

Footer

These "get" commands are called [template tags](#) which instruct the page to "get" or include these template files in the final page.

Notice that we haven't included a template tag to "get" the content. That is because the content is in the `index.php`. The `index.php` template file is the foundation upon which the building blocks are laid. It compiles the blocks all together around the content instructions in the `index.php`.

Here are the basic steps it takes for a simple layout on a WordPress web page:

1. Initiate `index.php`
2. Get the `header.php`
3. Generate the content within the `index.php`
4. Get the `sidebar.php`
5. Get the `footer.php`
6. Display entire page in the browser

The process of determining where and how these blocks or template files "look" on your final web page are determined within the style sheet file. Inside the style sheet are instructions for the header, sidebar, content, and footer to place the header at the top, put the sidebar on the right or left, and make sure the footer sits below the content. For more information on styling your WordPress Themes and web pages, see [Blog Design and Layout](#).

Template Files Within Template Files

You have seen how WordPress includes template files within the `index.php` template file. You can include multiple template files within template files.

Inside of the `sidebar.php` is typically another template file for getting the `searchform.php` to include a search form box in the sidebar.

```
<?php include (TEMPLATEPATH . '/searchform.php'); ?>
```

While the command structure is different, it does the same thing. Instead of using the WordPress template tags to "get", it uses the [PHP command `include`](#) (<http://us3.php.net/manual/en/function.include.php>) to "include" the template file. It also uses the `TEMPLATEPATH` to find the appropriate template file within your Theme's folder.

This method allows other template files to include other template files as needed to build the structure of the final generated web page. The `get` command template tags call to

standard template files. The `include` PHP command gets template files that are standard AND non-standard template files.

For example, if you are using a query in your [Loop](#) that asks "if displaying a post from category ID 14, use `header2.php` and `sidebar2.php` instead of the normal header and sidebar", you would use the `include` function as [part of that query](#) (<http://www.cameraontheroad.com/?p=623>). While the innards of `header2.php` may be exactly the same as `header.php` except for a slight change in the style sheet link to style posts within category 14 differently from the rest of the category posts, because `header2.php` isn't the "standard" header template file, you need an `include` command.

```
<?php include (TEMPLATEPATH . '/header2.php'); ?>
<?php include (TEMPLATEPATH . '/sidebar2.php'); ?>
```

Header

Content

Comment Form

Sidebar

Search Form

Footer

Most WordPress Themes include a variety of template files within templates to generate the different web pages on the site. For example, within the sidebar template file is a call for the search form template to be included. The following template files are typical for the front page (`index.php`) of a WordPress site:

1. `header.php`
2. The Content (`index.php`)
 1. `wp-comments.php`
3. `sidebar.php`
 1. `searchform.php`
4. `footer.php`

A lot of the blocks are interchangeable, allowing you to move template files around in the structure of your site. Some Themes feature the search form in the header, moving it from the sidebar. Others don't use a footer, leaving the footer template off their design. Other users don't want comments, so they eliminate or [comment out](#) the call for the comments form.

Customized templates can also be used with [Pages](#), allowing these to be uniquely styled and structured. You can also create your own custom templates to add or replace any of the parts and pieces, such as an alternative header, sidebar, feeds, asides, whatever building block you want to add all the time or occasionally to your web pages.

Remember, the styles are in the style sheet, but any structural elements can be put into a template tag and then called with an `include` command.

Using Alternative Template Files

WordPress features two **core page views** of web pages in a WordPress site. The **single post view** showcases the view of a single post web page. The **multi-post view** lists the posts or post summaries of more than one post. These are usually sorted by category, date (archives), author posts, and other ways. While some of the multi-post views can be generated from the `index.php` template file, WordPress' inherent [template hierarchy](#) allows specific templates to be used instead of the `index.php` file. This allows even more fine-tuning and customization.

Header

Single Post

Comment Form

Footer

The use of template files allow users to play with different blocks to build their own pages, choosing which ones they want to use when, and where, they need them. Built into WordPress' core programming is something called the [Template Hierarchy](#). Basically, it answers the following question:

What template file will WordPress use when the _____ (page) is displayed?

WordPress won't generate any special template files, but it will automatically recognize them if they are present and use them, without any effort on your part. Well, that's not true. If they aren't there, you have to provide them.

For instance, in the WordPress Default Theme, when a user clicks on the title of a post, if WordPress detects the `single.php` template file, it will use it to generate the web page view of a single post. In this Theme, the single post view does not include the `sidebar.php`. This is an example of the flexibility of WordPress template files.

You might want to have different information or looks to specific category listings of posts. WordPress automatically looks for the `category.php` template file to generate the list of [category posts](#), and if it isn't found, then generates them using the `index.php`. Using the template hierarchy, if it finds the `category.php` template file, it then looks for a category template file that matches the category ID number being sought, for example, `category-2.php`. If that template file matches, it will use that template file to generate a page view of all the posts in category 2.

WordPress will automatically look for a variety of template files based upon the user's request and you can learn more about creating and styling these and other custom template files at:

- [Template Hierarchy](#)
- [Archive Templates](#)
- [Category Templates](#)
- [Search Templates](#)
- [Sidebar Templates](#)
- [Header Templates](#)
- [Page Templates](#)
- [Form Templates: Comment, Search, etc.](#)

- [404 Page Not Found Templates](#)
- [Author Templates](#)

Template File Tips

Template files include the use of [XHTML](#) tags and [CSS](#) references. They are also packed with [template files](#), queries, and the [WordPress Loop](#) which generates the content within each page view from within the core template files like `index.php`, `single.php`, `category.php`, `archives.php`, and others.

Here are some tips for using WordPress template files:

Tracking Opening and Closing Tags

HTML tags and CSS references can cross template files, beginning in one and ending in another. For example, the `html` and `body` HTML tags begin in the `header.php` and end in the `footer.php`. In some WordPress Themes, the content container may start in the `header.php` and end in the `index.php` or `single.php`. Tracking down where one tag begins and the other one ends can get complicated if you are [developing, designing, or adjusting a Theme](#). We recommend you use [comments](#) to note in the template files where a large container tag opens and when it closes so you can track which `</div>` is which at the end of different sections.

Test Template Files Under Different Views

If you have made changes to the comments, sidebar, search form, or any other template file, make sure you view these template files in all their possible page generations. For example, check out the single post view, and various multi-post views like categories, archives, or search to make sure the changes you've made to template files included in other template files work across all of the various page views.

Comment Deviations

If you are [designing Themes for public release](#), make notes in your template files where you have made dramatic changes from the Default and/or Classic Themes, especially where you have changed style references or added styles outside of the style sheet like in the header or inline with the XHTML tags. Make a note of the styles used outside of the style sheet in the style sheet and include comments within the template files to guide future users.

Close the Tag Door Behind You

If you start a HTML tag or `div` in one template file, make sure you include the closing tag in another template file. The WordPress Forum gets a lot of questions about "what happened to my theme" when they remove the footer template file without closing the tags that began in the header template file. Track down your tags and make sure they are closed.

CSS Styles in Templates

Different core or included template files may feature CSS styles the same as other core template files, or different. For example, the Default Theme features different styles depending upon if the user is viewing a page generated with the `index.php`, `single.php`, `archives.php`, and other template files. While Theme authors may change the style references in their Theme, for a list of the general architecture and style references in the Default and Classic WordPress Themes, see [Site Architecture 1.5](#).

Template File Resources

- [Templates](#)
- [Site Architecture 1.5](#)
- [Stepping Into Template Tags](#)
- [Template Tags](#)
- [The Loop](#)
- [The Loop in Action](#)
- [Conditional Tags](#)
- [Theme Development](#)
- [Designing Themes for Public Release](#)
- [Query Posts Template Tag](#)
- [Creating Multiple Single Posts for Different Categories](#)