

# How to convert any web template into a WordPress theme

Among blogging applications, WordPress probably has the largest number of great-looking themes to use. Still, there are hundreds of free and even open source web templates not yet converted to work with WordPress. Knowing how to make these themes work with WordPress broadens your choice of design to use for your blog.

Converting a web template is fairly easy if you take the time to learn how to do it. I wrote this guide for someone like me a few months back — eager to use a great looking web design and yet not knowing how to start converting it to work with WordPress. If you want to view a video tutorial on how I turned [this open source web design](#) into [this WordPress theme](#), [click here for the blog post](#).

## The Tools

In converting web templates into WordPress themes, I use [Aptana](#) and [Cream](#) (my preferred text editor). I use Aptana to visually edit the template and incorporate the changes I want done. I used to do this with Dreamweaver (I wrote [a tutorial on how to use Dreamweaver to edit WordPress themes](#)) but ever since I tried Aptana, an open source software, I've never gone back to the iconic WYSIWYG tool.

## Where to download templates

I prefer web templates released under an open source license so I usually get my templates at [Open Source Web Design](#) () and [Open Web Design](#).

The advantage of using open source templates is that you are allowed to change and redistribute it. But if you only plan to create a custom WordPress theme for your site and not redistribute it, you can browse through the hundreds of websites offering free web templates.

## Tips on choosing a template

Choose designs done in CSS and XHTML. Most of the free and open source web templates offered for download are now done in CSS and XHTML and authors usually highlight this fact either on the web template itself or in the readme file that comes with it.

Validate the design using website validation service. You can use [validator.W3.org](#). Again, most templates contain a link within the page that would take you to the result of W3's validation of the page.

Choose designs *not using HTML tables* to render the page's columns. Tables are great for presenting tabular data such as statistics but not for page layout. Most competent web designers will tell you never to use tables for laying out your web page.

## Other resources

The [WordPress Codex](#) is an extensive documentation on various aspects of the blogging script. In converting a web template into a WordPress theme, you should check the [template tags section of the WordPress Codex](#) to see what tags you'd need to generate blog items such as the post headline, date, comments link etc. Be sure to read about the [WordPress Loop](#), the code that generates blog post content.

You can simplify the template conversion by keeping as reference a copy of a working WordPress theme. You can just then copy snippets of code from the theme for use in your own template. You might also want to check [these pieces of great WordPress theme code](#) that I've collected.

## Organize theme files

After downloading the template you want to convert, check how the files are organized. If the stylesheet is not named style.css, rename the file. Add style information to the CSS file, including your name for the WordPress theme, just copy what other WordPress theme

authors did by opening their style.css file. Don't forget to edit the CSS filename in the HTML file so that you can still preview the design later when you open it in Aptana.

If the images are saved in the root directory, create an image directory and save all the images there. This isn't really required but it makes for an organized way of doing things. When you do transfer images, make sure that you edit the file locations of all img calls in the HTML and CSS file.

### **Finalize design**

Open the HTML file in Aptana and finalize changes you want done on the design. You might want to play around with the type of font used or its sizes or the dimensions of the web templates' columns. If you want to completely change the template's layout, you can spend time editing various components of the template, including the images that come with it.

It is important, however, that you finalize the design at this stage since it will be cumbersome if you do design changes when the files have been cut up into different WordPress theme .php files.

Most free web templates typically contain several sample posts to show how articles, headlines and author links are formatted. Leave only one article in the template. You will be inserting the WordPress Loop in this part of the template later. Mark this part with `<!-- WordPress loop here -->`.

### **Mark up the template**

In Aptana, go to Code View and mark up the parts of the page that you will be cutting up later into header.php, sidebar.php and footer.php. Just mark it up using `<!-- Section starts here -->` and `<!-- Section ends here -->` tags.

Create a new folder and name it after the WordPress theme name you chose and copy the style.css and image folder into it.

### **Cut design into different WordPress theme files**

Open Cream and then go back to Aptana. Cut and paste the header, sidebar and footer parts of the theme, paste these into Cream and then save it as header.php, sidebar.php and footer.php. Save the remaining code in Aptana as index.php and insert php calls for the header, sidebar and footer: `<?php get_header(); ?>`, `<?php get_sidebar(); ?>` and `<?php get_footer(); ?>`.

Open header.php in Cream and insert elements such as the template tags for the page title, stylesheet, RSS feed, favorites icon. You can simplify this process by copying the codes in the header.php of a working WordPress theme. [K2](#) is a good theme to copy from. You can also read about header.php in the WordPress.codex.

Here's the code I usually add to the header.php of a template I'm converting:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://gmpg.org/xfn/11">
<meta http-equiv="Content-Type" content="<?php bloginfo('html_type'); ?>;
charset=<?php bloginfo('charset'); ?>" />
<title><?php wp_title(''); ?> <?php if ( !(is_404()) && (is_single()) or
(is_page()) or (is_archive()) ) { ?> at <?php } ?> <?php bloginfo('name');
?></title>
<meta name="generator" content="WordPress <?php bloginfo('version'); ?>" />
<!-- leave this for stats -->
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>"
type="text/css" media="screen" />
<link rel="alternate" type="application/rss+xml" title="RSS 2.0" href="<?php
bloginfo('rss2_url'); ?>" />
<link rel="alternate" type="text/xml" title="RSS .92" href="<?php
bloginfo('rss_url'); ?>" />
<link rel="alternate" type="application/atom+xml" title="Atom 0.3"
href="<?php bloginfo('atom_url'); ?>" />
<link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
```

```
<?php wp_get_archives('type=monthly&format=link'); ?>
<?php wp_head(); ?>
</head>
```

Open sidebar.php and then insert sidebar elements you want in your blog. Open footer.php and include information you want to display there.

Open index.php and include the WordPress Loop, the code that publishes your posts. You can read more about the [WordPress Loop](#) here. Again, you can simplify this part by copying pieces of code from a working WordPress theme.

The loop starts with:

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
```

Place this part above the part you marked `<!-- WordPress loop here -->`. Replace the article headline (usually between `<h1></h1>` or `<h2> </h2` tags with `<a href="<?php the_permalink() ?>" rel="bookmark" title='Click to read: "<?php strip_tags(the_title()); ?>"><?php the_title(); ?></a>`. What this does is generate the blog posts headline and link it to its post page. You can remove the link in the post page by using this `<?php the_title(); ?>` in single.php. After all, it doesn't make sense for the title in the blog post page to be clickable and linked to the current page. Replace the dummy text in your web template with `<?php the_content("Continue reading " . the_title(' ', ' ', false) . " "); ?>`. This generates the blog post article.

You can add other post elements such as author data and links or the date of posting. Just check the WordPress codex for their template tags.

End the loop with:

```
<?php endwhile; ?>
```

You can then add previous and next post links after this. Here's the one I use:

```
<?php next_posts_link('&laquo; Older blog posts') ?> <?php
previous_posts_link('&bull; Newer blog posts &raquo;') ?></p>
```

After that, add this:

```
<?php else : ?>
  <h2 class="center">Not Found</h2>
      <p class="center">Sorry, but you are looking for something
that isn't here.</p>
      <?php include (TEMPLATEPATH . "/searchform.php"); ?>
<?php endif; ?>
```

After adding the Loop, save the file and then save it again as page.php (to serve as template for WordPress pages) and single.php (to serve as template for individual blog posts). You can then customize the files if you want, for example, a different presentation for individual blog articles or for the pages.

Add the comments tags in single.php `<?php comments_template(); ?>`.

### **Adding other theme elements**

Add a comments.php to the your theme. What I do is copy an existing comments.php and then change its styling to reflect the stylesheet I'm using. Here's [a guide to comments.php in the WordPress Codex](#).

Add a 404 page template by copying an existing one from a working WordPress theme. Add a searchform.php, here's [a guide from the WordPress codex](#).

You can also add a category template by copying index.php and inserting customizations you might want done. Add an archives.php for your archives. Here's a handy guide from the WordPress Codex. [http://codex.wordpress.org/Creating\\_an\\_Archive\\_Index](http://codex.wordpress.org/Creating_an_Archive_Index)

After you're done with the template files, upload the folder into your WordPress themes folder and then CHMOD the stylesheet and individual PHP files to 666 so that you can continue tweaking and editing it online.  
Activate the theme, check for errors and continue editing and tweaking it.