



The Handbook

Codex

http://codex.wordpress.org/Designing_Themes_for_Public_Release

Version Date

6 August 2005

Designing Themes for Public Release

As open source code, WordPress developers like to keep things loose when it comes to what a user does with WordPress. However, when it comes to designing your styles or Themes for contests and public consumption, there are a few things you should take into consideration. These are not hard and fast rules, they are more guidelines, on how to make your Theme the best it can be before exposing it to the public.

Having the ability to quickly change a site's look is very exciting, and having your Theme or style be one of their choices is even more exciting. To keep all of this excitement going, we need to make sure that under the hood, there are some consistencies and things you should know before releasing your hard work and wonderful design to the public.

Get Familiar With the Process

In the WordPress Codex, the main documentation for using WordPress, you will find tons of information that will answer your questions about Themes, creating Themes, styling CSS, Template Tags, and more. Before you begin the process, we recommend you get familiar with the functions of WordPress.

- [WordPress Blog Design and Layout](#)
- [WordPress CSS Information and Techniques](#)
- [Using WordPress Themes](#)
- [Site Architecture 1.5](#)
- [Developing WordPress Themes](#)
- [Stepping Into Templates](#)
- [Stepping Into Template Tags](#)
- [WordPress Templates](#)
- [Template Tags](#)
- [The WordPress Loop](#)
- [The Loop in Action](#)
- [Lessons: Designing Your WordPress Site](#)
- [Lessons: Using Template Tags](#)
- [Lessons: Template Files](#)

Make a Plan

As you consider your design possibilities, check out [CSS Zen Garden](http://www.csszengarden.com/) (<http://www.csszengarden.com/>) for inspiration. The volunteer designers took one HTML page and created hundreds of various web page designs from it. The HTML doesn't change, only the style sheet changes. The designs are incredibly varied and unique. This shows you not only the power of designing with CSS, but also the power of the imagination to create and manipulate the most simplest of information into works of art. It begins with a plan.

According to NuclearMoose, one of the popular WordPress proponents, start to make your plan with some basic tools that don't involve the computer.

I have some tips...paper, pencil, sticky notes, and pencil crayons. "Whazzat for?" you ask? I'll tell you.

I use the sticky notes to represent a piece, chunk, module, or section of the page. (Use your own appropriate term.) I write "Links" and "Navigation" etc. on them and set them on a blank piece of paper. That way, I can easily move around the elements as I like, finding something that looks balanced. I even cut some of them to size to more closely represent their actual proportion on a page.

After I get something I like, I will sketch it out with more detail. Often I will colour things to try out various colour combos. Actually, I just like colouring, but I don't want my kids to think I'm colouring for fun...this is serious stuff!

I write down all of the things I want to include in the site on a piece of paper. Then I plan out each part of it, making notes about fonts, alignments, plugins that may be needed, background images, or other artwork/graphics that I may want.

At the end of all this process, I have a pretty good handle of how I want things to look, as well as how it should be structured. I can start gathering all of the assets for the site (images, plugins, etc.) and begin to think about coding.

In other words, there's a fair bit of non-computer-related work up front. I've found that this helps me a lot when starting the actual "construction" of the site since I'm not sitting in front of my 'puter with a blank screen in front of me, being taunted by that damn cursor blinking, blinking, blinking...

*Oh yeah, it doesn't hurt to have a CSS pocket reference or some other favourite CSS resource material handy so that you don't do silly things like *cough* use deprecated attributes *cough* on some of your elements.*

Once you get tired of all the work, you can invite your kids to come and help Dad colour the web site and spend some quality time together.

Couldn't have explained it any better. Your plan should include:

- **Structural layout** - Where do all the parts go?
- **Specific Elements** - Will you have a calendar, comments, what parts will you include?
- **Template Modular Elements** - Which templates will you use or add? Site map? [Pages](#)? A distinctly different single post page?
- **Graphics** - What graphics should go where?
- **Colors** - How many, how are they used, do they have a purpose or are just for show?
- **Fonts** - How many and what sizes go where?
- **Space** - Space is an important part of layout so how will you use space?
- **Itinerary** - How and when are you going to do all this?

Know Your Sources

As you design your Theme or style, you will be using HTML and CSS references. Decide to get into the code and add some PHP, you need to have the right resources to dig into and get the information right to make your WordPress website sing. We've put together a [few of the resources you will need](#) to become very familiar with over time to help you get started.

Part of designing a solid WordPress Theme is to know what you are doing. This means, you have to know where to get the answers. Keep that list handy, bookmarked or saved to your hard drive. You'll refer to it often.

Start With The Default or Classic Theme

You can start with someone else's theme or style sheet, that's fine, but if you really want to do this right, start with one of the two default themes that come with WordPress. They are called "Classic" and "Default" aka Kubrick.

Why start here? Because these have been through the presses by WordPress designers and testers, as well as bazillions of users who are more rigorous on these things than the developers. These are solid code, for the most part, and a good starting point. From there, do whatever your imagination desires.

- [Site Architecture 1.5](#)
- [Kubrick](http://binarybonsai.com/kubrick) (<http://binarybonsai.com/kubrick>)
- [Boren's Anatomy of a WordPress Theme](#) (<http://boren.nu/archives/2004/11/10/anatomy-of-a-wordpress-theme/>)
- [WordPress Templates](#)
- [Template Tags](#)

Keep The Core References

Do not remove "default" WordPress CSS references!

Okay, after making it clear that designers are free to do whatever they want as long as it validates, and that these are only guidelines and not rules, we admit it, we lied a little.

User-friendly is the key to designing WordPress Themes. Part of that friendliness is to keep the default elements of code within the [Templates](#) and [CSS](#) files and design these to change their look, but not to remove the code references. You can hide them, but don't remove them.

For example, in the WordPress default Theme, the Theme's author made some decisions about removing the author tag from the post (it's your blog so why would you have to keep telling people you wrote it) and the calendar. If you look at the code for the calendar, it is still there, just [commented out](#), so if the next user wants to feature a calendar, they can. The styles for the calendar are still in the style sheet so it won't look bad if enabled.

Just because you like some detail or not, leave the core coding in to accommodate the various needs of the users.

To ensure smooth transition for [language localization](#), use the `_e()` function for *echoing* titles and headings within the template files. This makes it easier for the translation files to hook in and translate the titles into the site's language.

As you modify the styles, if you tread into the template files, make sure that `<div id="header">` stays that way. Don't change it to `<div id="fred">` just because you think "post" is boring and you always named your header section `fred`. If you are adding a fourth column, or a new division or class identifier, you can name it anything you want, but make a note in the style sheet and template file that these are additions to the default.

When adding references to new elements in the style sheet, it's recommended to isolate them from the default references so they are highlighted or make a note of them if they are grouped with related elements like the structure, helping people identify them more quickly. This also highlights the uniqueness of your design by showing off your style codes.

Again, these are things designers for contests and public consumption need to worry about. Everyone else tweaking their sites for their own use can do anything they want - and will!

Reconsider Plugins

[WordPress Plugins](#) add great extensions and capabilities to a WordPress site. Some plugins require the user to edit their theme and add a line of code that calls their plugin. If a plugin is included with a Theme, users still want the right to use or not use them. Themes should be plugin-independent and not rely on a plugin to be installed in order to function. Unfortunately, many questions come into the WordPress Forums about plugins associated with Themes. People want to know how to turn them on, or off, and "What is this code in my Theme?".

If you choose to include plugin support in your theme, here are some tips:

- Ensure that the plugin has been fully tested and will work under a variety of conditions.
- Include the plugin or a link to the plugin.
- Include specific instructions for installing and uninstalling the plugin to the Theme.
- Include coding around any tags within the template that will allow the Theme to work even when the plugin is turned off (see below).
- Be prepared to support questions about the plugin as well as your Theme.
- Maintain an updated link and information in your Theme and site to the plugin, if that information changes.

Some plugins feature tags inside of the template files. If the plugin is not activated, it will "break" the Theme and it may report errors or fail to load. It is therefore imperative to prevent the plugin from being detected in case it is turned off.

To detect if a plugin is installed, you can use a simple `function_exists()` check. The `if (function_exists())` checks for the plugin, and if it exists, it will use it. If it returns `FALSE` or "not found", it will ignore the plugin tag and continue loading the page.

```
<?php
if (function_exists('FUNCTION_NAME')) {
    FUNCTION_NAME();
}
?>
```

This example plugin uses a function called `jal_get_shoutbox()` to print out its contents.

```
<?php
```

```
if (function_exists('jal_get_shoutbox')) {  
    jal_get_shoutbox();  
}  
?>
```

Style ALL the Template Files

Before WordPress v1.5, it was all about the `index.php` file, forcing it to do all the work for every element of your page. Today's WordPress uses [modular elements](#) to make up an entire page.

The Default Theme for WordPress relies upon the `index.php`, `sidebar.php`, `single.php` (for a single post which looks different from the categories and archives), `comments.php`, `header.php`, and `footer.php`, among others. If you create a theme or style based on only the `index.php`, leaving out the comments, header, footer, or others, you may run into design problems.

If WordPress can't "find" the modular part, such as the comment template, it will look for it in the default folder. Therefore, if the designer hasn't taken this modular template into consideration, the layout and design might be a little messed up. It will work, but it could look weird. You don't have to use all the various bits and pieces, keeping the header and sidebar inside of the `index.php` or `single.php`, but do look at the parts as well as the whole.

- [WordPress Template Hierarchy](#)
- [WordPress Conditional Tags](#)
- [The WordPress Loop](#)
- [WordPress Templates](#)
- [Theme Development](#)

Style Sheet Structure

There is a lot of debate about how to layout the structure of a CSS style sheet. The key to the layout is how to make it easier for users (and designers) to find the information they want and need, for information and to make modifications. Some say that an alphabetical order of the Selectors (names of the style identifiers) makes it easy so that if you are looking for `#post` you just scroll towards the bottom of the page.

Yet, if you are designing a Theme, different elements, like links, may have a different look between sections. So would you naturally look for the `#post a:link` in `post` or under `a:link`? If you didn't know it was in the `post` section, you would look in the latter area.

Some people prefer to group their related selector elements together such as structure, links, lists, headings, and so on. This makes a lot of sense and is helpful to the user. If they are tweaking the structure, for example, any change to the `header` height will impact the `content` and `sidebar`. Instead of bouncing up and down the page from middle to top to bottom, all of the elements would be close to each other, preferably identified as `/* Structure */` in the style sheet. If the user wanted to change the look of all of the links to make them more vibrant, then all the links would be together in one place.

There are no hard and fast rules about this. Consider what works best for you and then what may work best for the end user. Keep your presentation consistent. Make a plan on

how you want to do this early on in your designing so you won't spend so much time rearranging things later.

Consider the Details

There are a lot of details that have to be taken into consideration when designing your WordPress Theme for public release. Here are a few to consider.

Consistent and Standard Fonts

As you work your Theme, [fonts play a critical part](#) in the design. Many inexperienced web page designers present themes and styles with only one font. Not just an overall font style for the whole page but ONE font. In the body style tag it may say:

```
body {margin: 0; font-family: "Trebuchet MS",  
sans-serif; font-size: 1em;...}
```

What happens if the user doesn't have Trebuchet on their computer? It happens every day. If that font isn't on the user's computer, the system default shows up which is often Arial or some similar sans-serif font. If you like the look, great, because just about everyone has it, but if you want Trebuchet and you really want your fonts to have a specific look, then you have a problem.

To increase the chances of a font similar to the one you really want showing up on the page when it displays, add more choices to the font-family like this:

```
font-family: "Trebuchet MS", Verdana, Futura, Arial, Helvetica, sans-serif;
```

If the browser can't find Trebuchet, it looks for Verdana, and if can't find that, it looks for Futura (Mac systems), and so on. That should cover all your bases.

Also remember that while there are some fonts which are common on Windows systems, there are other computer systems out there like Linux and Mac. Consider using fonts that will also work on their machines, too.

As you choose your fonts and design your Theme, make sure you [choose fonts](#) that are readable and easy -to-read, keeping font-sizes large to avoid eye-strain yet sized to fit within your design.

Comment Comment Comment

Have you looked under the hood of a racing car lately? The bells and whistles inside are a nightmare to figure out. Remember the first time you looked at PHP, CSS, or HTML code? Bet that gave you a few moments of hysteria. When you are releasing your Themes and styles to the public, remember the user may take a peek under the hood and run screaming from the room, too.

[Comments are part of code](#) that helps the designer and the user figure out what is what. Comments help users identify DIVs that cross templates, when a template begins and ends, and any changes that indicate you, the designer, has modified something that makes it different from the original code.

Help your users by commenting as much as possible to help them find the different

sections and use your Theme with ease.

Lean and Mean CSS Style Sheet

WordPress stresses that code and style files should validate and be laid out with a lot of tabs so they are easy to read. The [World Wide Web Consortium](http://w3c.org/) (<http://w3c.org/>) and the [Web Standards Organization](http://www.webstandards.org/) (<http://www.webstandards.org/>) stresses that all web page code be compliant with their standards. If you are going to get into this, you should familiarize yourself with the most basic of these standards.

One of those standards is to present a clean and optimized style sheet and HTML code. While WordPress contests tend to be lenient on this subject, other web page design contests don't. Their motto tends to be "a clean and tight code is a work of art, too".

Basically, it means doing [some housekeeping on your code](#) before you release it. Yes, you will need to validate it and test it across browsers, but let's start with some simple cleaning.

Every space, character, and bit in your code and style sheets add up to bytes. That sentence came to about 64 bytes. Each byte of information adds up and the larger they are, the longer they take to load. You do yourself and your users a favor by keeping your file sizes down to byte sized sizes. So where do all these bandwidth wasters hide?

If you have set your code to look pretty with lots of indents, have you checked to see if there are any TAB codes at the end of the line before the line break? I find a lot of these. You don't need a TAB before a line break, only after, but somehow, these sneak into the code.

Using spaces to line up code adds to the size. A TAB is considered one character in most editors, but the five spaces that copy the TAB indent takes up five characters. Using double spaces instead of single spaces in your code and styles adds up, too.

Using a good search and replace capable text editor, you can quickly clean these up, making your styles and code optimized for fast loading.

Condense and Use Shorthand

It's a good idea to use [shorthand for your CSS](#). While it isn't a required standard, it is part of code optimization. Use this technique with caution as some browsers can't handle it, typically "older" browsers. If you are unfamiliar with CSS shorthand, we've put together a [short tutorial for you](#).

Validate! Validate! Validate!

Make sure your codes and styles validate across the board. That means they have to meet the "strict" standards set by the [W3C Organization](http://www.w3c.org/) (<http://www.w3c.org/>) and pass a variety of validations for CSS and HTML. Not all validators check for the same things. Some only check CSS, others HTML, and others for accessibility. If you are sincere in presenting solid code to the public, test your code with several validators.

Validation doesn't just mean putting your pages through some web driven testers. It also means test-driving it with friends, relatives, co-workers, and strangers you meet on the street. Everyone has a little different system, so ask for others to test-drive your styles or

themes before you make them public. [The Your WordPress section](http://wordpress.org/support/forum.php?id=11) (<http://wordpress.org/support/forum.php?id=11>) in the [WordPress Forums](http://wordpress.org/support/) (<http://wordpress.org/support/>) is dedicated to checking out your site by WordPress volunteers while you are working on it, when you have trouble, or just to say ooooh and ahhhhhhh.

If you are unfamiliar with how to validate your web pages and style sheets, we've put together a [list of resources and information](#) to help you.

CSS and HTML Bugs

WordPress Themes and styles are diverse examples of the brilliant and imaginative WordPress users out there. What all of them have in common as they hold the design concept in their head and work long hours to convert it into code that matches their imagination is.....browser bugs.

You work for three days to make it beautiful in Microsoft Internet Explorer, take a look at the same site in Firefox and it looks different. In fact, things don't line up right. Or you designed it to work in Opera, but when you view it in Internet Explorer, the sidebar is now half off the screen.

Before you pull out your hair, remember that others have suffered before you and you are not alone. Check out these resources for help on solving the many CSS, HTML, and browser bugs that are out there.

- [WordPress CSS Information and Techniques](#)
- [CSS: Fixing Browser Bugs](#)
- [CSS: Troubleshooting](#)
- [Positioniseverything.com](http://www.positioniseverything.net/) (<http://www.positioniseverything.net/>)

What Else...?

These are a few of the most common things to take into consideration when presenting your themes or styles to the public, but it is only the tip of the iceberg. Remember, if there is a problem with your theme, users should begin their search for help on your website, but they often turn to the [WordPress Forums](http://www.wordpress.org/support/) (<http://www.wordpress.org/support/>). The more consistent with the default codes and style references, the faster users can get the help they need if there is a problem. We know that you read through this article and followed all of these recommendations to the letter, users will come to the WordPress Forums to brag and show off instead.

The better designed your Theme is, the more web standard and cross-browser compliant, the more successful your Theme may be. As we said, there are no hard and fast rules for design, only guidelines for the code under the hood. The sky is the limit.

Promoting Your Theme

Once your WordPress Theme has been validated and tested thoroughly, then it is ready for release.

Put all your Theme files, including a readme text file with information and description, in a zip file for easy downloading. If possible, provide two or more file compression types like

RAR, ZIP, GZIP, etc. to maximize user choices. Be sure and include information on any custom tweaks, tips, plugins, or things the user must know in a readme file.

Post a link to a page on your site with the following:

- Demo or Screenshot of various page views
- Description
- Downloadable zip file

Go to the [Using Themes/Theme List](#) and find the appropriate category for your Theme.

Post a link to the Theme information and downloadable file, and add the name and **keyword description** of the Theme, per other examples. The more keyword descriptions you use, the easier it is for people to search through the list to find your Theme.

Post a note about your new Theme on the WordPress Forum under [Themes and Templates](#) (<http://wordpress.org/support/forum/5>) describing the Theme. The more descriptive keywords you use, the more likely people's search for Themes will turn up your request. Include links to your Theme and the downloadable zip file.

Check the other lists for Theme Resources on the [Using Themes](#) article and check there for instructions on adding your Theme to their resource list.

It is preferable that you provide support for your Theme on your site with a link in the Theme and readme file for the support page, to help users. Also check the Forum frequently to see if people are having problems with your Theme and address their issues as much as possible and update your Theme if necessary. If you update, please add a note of the update with the version number in the Forum and on the Theme List in the Codex.