

NAMING CONVENTIONS

It's important to understand the role that naming conventions play in writing code regardless of what platform with which you're working. Ultimately, naming conventions - regardless of if they are for classes, functions, variables, attributes, or arguments - should help to explicate the purpose that they serve.

By that, I mean that class names should typically be nouns, functions should typically be verbs, and variables, attributes, and arguments should explain the purpose that they serve within the context of the class or function in which they are to be defined.

Class Names

When it comes to working with WordPress, you're not likely to encounter classes unless you're doing one of two things:

- Writing a custom library to work alongside a theme or application
- Writing an OOP-based plugin

Function Names

As mentioned earlier, if classes are nouns that ideally represent a single idea or single purpose, then their methods should be the actions that they are able to take. As such, they should be verbs - they should indicate what action will be taken whenever they are called.

Variable Names

Variable names actually aren't much different from function names other than they represent a single value or a reference to a particular object. The naming conventions still follow what you'd expect:

- Lower case (versus camelCase)
- Separate spaces with underscores

TEMPLATE METHOD

The process enables you to make small changes to your pages with relative ease. However, it can become unwieldy if you're trying to make a lot of alterations. Instead, you may want to create your own custom page templates. Let's walk through how to do that in just four steps.

1. Protect Your Site From Potential Mishaps

The first thing you'll want to do is take a few precautionary measures. This is important because you'll be making changes within your core WordPress files. That means if you aren't careful, you can end up breaking something by adding or deleting the wrong code and have a hard time fixing your mistake.

2. Copy and Customize the Default Page Template

Now, it's time to create your page template. While you *could* build one from scratch, it's easier to copy and modify an existing template. Therefore, you'll need to log into your site via File Transfer Protocol (FTP) to access the file. In the bottom right quadrant, scroll through the files until you find *page.php*. Make a copy of it and save it somewhere on your computer.

3. Upload the Template to Your WordPress Website

Once you're happy with your customizations, you'll need to upload the template to your WordPress site. You'll need to have your FTP client open once again. To add in your custom page template, simply copy and paste it back into the theme folder alongside the original.

4. Assign Your Template to a Page

Your custom page template is now ready to go! However, it won't do anything yet since it isn't associated with any pages. You'll need to link it to the page (or pages) you want it to affect. To do this, simply open up the page you'd like to use and look for the *Page Attributes* field.

You'll see a Template drop-down menu here too, and you can select your custom template and save or update the page. This can be a quicker way to assign pages to a template if you have a lot of them to get through.