



# Building a WordPress Theme From Scratch

By Joseph Erickson

# Building a WordPress Theme from Scratch

## A Step-By-Step Guide to Getting Your Design into WordPress

Joseph Erickson

This book is for sale at <http://leanpub.com/wordpressthemefromscratch>

This version was published on 2014-03-30



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2013 - 2014 Joseph Erickson

# **Tweet This Book!**

Please help Joseph Erickson by spreading the word about this book on [Twitter!](#)

The suggested tweet for this book is:

Learning how to create WordPress Themes with Building a WordPress Theme from Scratch

The suggested hashtag for this book is [#wpthemefromscratch](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#wpthemefromscratch>

# Contents

<b>Introduction</b> . . . . .	<b>i</b>
<b>WordPress Data</b> . . . . .	<b>1</b>
Template Tags . . . . .	1
Information About the Website . . . . .	1
Information About the Page . . . . .	2

# Introduction

There are a million and one articles, blog posts and wiki pages on building themes for WordPress, so why did I write a book about it?

If you're looking at this book, you probably know the answer; none of them actually help you build a WordPress theme from scratch.

In this book, I will show you how to build a WordPress theme with zero prior knowledge of WordPress or even PHP programming in general. Starting from nothing, we will build a WordPress theme step by little step. By the end of the book you will have the basic knowledge you need to turn one of your own designs into a basic WordPress theme.

A couple of assumptions I'm making about you:

1. You don't know how to program in PHP, or any language for that matter. If you do, all the better, but it's not required.
2. You already understand HTML and CSS. You can try following along if not, but this book is about getting your designs into WordPress and if you can already build a design, thus won't help you much. I'll also be working in HTML5, but you can really use any flavor of HTML if you get the overall concepts.
3. You have access to a WordPress installation. This is key since this book is extremely hands on. You won't really learn this unless you do it along with me. You can install it on a web host, which will be easiest, or on your computer using the XAMPP software available at <http://www.apachefriends.org/en/xampp.html>.

You can also follow along with everything I'm doing by going to:

1 <https://github.com/firstclown/wordpressthemefromscratch>

From there, you can check out the code and follow along. Every chapter is there as a git branch and builds off the previous chapter. If you know nothing about `git` or version control, this will be a great place to start to learn this valuable skill. GitHub has a lot of valuable help documents that can help get you started.

At the end of this book, you're going to have a firm grasp on how themes work in WordPress, how WordPress itself handles its data and how you can get your designs into WordPress sites. Without further ado, let's get started.

# WordPress Data

This will be a short chapter just to get you up to speed on the *very basics* you need to get started in working with WordPress. Once we get through this, we'll dive right into the basics of making a theme.

The main purpose of your theme is getting data from WordPress and displaying it to the user. WordPress provides a number of ways for your theme to get that data. Let's go over the most useful functions now.

## Template Tags

Data is obtained and shown in WordPress via what are called Template Tags. Template Tags will look something like this:

### Sample Template Tag

---

```
1 <?php bloginfo('name') ?>
```

---

If you know anything about PHP, this will look familiar, but just understand that this is what a Template Tag will look like when putting it into your theme files.

When WordPress sees a Template Tag in your theme, it will replace everything from the `<?php` to the `?>` with the appropriate data. For instance, if I use the above Template Tag like so:

### Sample Template Tag Use

---

```
1 <title><?php bloginfo('name') ?></title>
```

---

Then when WordPress shows the site to the user, it will look like this:

### Sample Template Tag Display

---

```
1 <title>My Blog</title>
```

---

Let's look at some of the most important pieces that will comprise your theme files.

## Information About the Website

Most information about a website can be obtained via the `bloginfo()` Template Tag. `bloginfo()` will get details about the site itself from the WordPress database and let you display those on the site. The name is a bit misleading since the site might not be strictly a blog, but understand that WordPress was originally designed to just be a blogging platform even though it is now used for applications and regular websites. The name `bloginfo()` just kind of stuck around for the ride.

Commonly used options for `bloginfo()` are:

## ‘name’

Using this will display the ‘Site Title’ that was set in Settings > General of the WordPress Admin.

Example Usage of `bloginfo(‘name’)`

```
1 <h1><?php bloginfo('name') ?></h1>
```

## ‘description’

This will display the ‘Tagline’ that was set in Settings > General of the WordPress Admin.

Example Usage of `bloginfo(‘description’)`

```
1 <h2><?php bloginfo('description') ?></h2>
```

## Information About the Page

There is also a lot of information about the page that the user is currently looking at, but before we get into that, we need to look at a core WordPress concept called The Loop.

## The Loop

When a user visits a WordPress website, they will do so at a certain URL, like `http://jerickson.net/wp-books/`. WordPress’s first job on every request is to look at that URL and decide what information the user is looking for. In this case, it will see that I have a page with the slug `wp-books` and will make that page available for your theme to show the user. If the user asked for `http://jerickson.net/category/books/`, WordPress will get every post that is under the category `books` and give all of them to your theme. So sometimes you will get one page to show and sometimes you will get several “pages” to show.

In order to handle this, WordPress has a concept called The Loop. The Loop, in essence, is how you will go through each piece of content, either a post or a page, that WordPress gives to your theme to show to the user. And this is important, **you must always assume that WordPress is giving you more than one piece of content.**



### Keep In Mind...

Unless otherwise stated, you have to handle more than one post or page in each call to your theme even if you think you won’t have to. In other words, always use The Loop.

Here’s an example use of The Loop:

### Example Usage of The Loop

```
1 <ul>
2     <?php while ( have_posts() ) : the_post() ?>
3         <li><?php the_title() ?></li>
4     <?php endwhile ?>
5 </ul>
```

In the above example, the theme using this code will show an unordered list of the title of every piece of content that WordPress has given us. If we look at this in detail, we see that `while` we have `have_posts()` available, we're going to load `the_post()` and make it available to us. We then show `the_title()` of the current post we're working with in `<li>` tags. Once we get to the `endwhile` tag, we start over and as long as we have `have_posts()` to work through, we continue the loop.

If we only get one piece of content, like a single page, it will only show one bullet point. If we get more than one, like a category page with many posts on it, it will have multiples.

### Result of the Above Example

```
1 <ul>
2     <li>Title of first post</li>
3     <li>Title of second post</li>
4     <li>Title of third post</li>
5 </ul>
```

Anything within The Loop (in between the `while` and `endwhile` tags) will be shown for each piece of content, like the `<li>` tags above. We will be getting into very concrete examples soon, so don't worry if this isn't completely clicking right at the moment.

## Template Tags in The Loop

There are a number of Template Tags used inside The Loop to get details from our content. In fact, we used one above for our examples.

### **the\_title()**

Using `the_title()` will display the title of the piece of content.

### Example Usage of `the_title()`

```
1 <h3><?php the_title() ?></h3>
```

### **the\_content()**

This will display the content entered into the rich text portion for this page or post.



### Example Usage of `the_content()`

---

```
1 <?php the_content() ?>
```

---

### `the_date()`

This will display the date that the page or post was published.

### Example Usage of `the_date()`

---

```
1 Published: <?php the_date() ?><br>
```

---

### `the_author()`

This will display the name of the user that published this post.

### Example Usage of `the_author()`

---

```
1 Written By: <?php the_author() ?><br>
```

---

We will see many more examples of Template Tags we can use throughout this book, but these should be enough to get us started. So, let's get working on our first template.