Justin Tadlock

Life, Blogging, and WordPress

Navigation

- Home
- About
    - Contact
    - Movies
    - TV Series
- Archives
    - Asides
    - Galleries
    - Tags
    - Topics
- Writing
    - Essay
    - Musing
    - Novel
    - Poem
    - Short Story
- Book
- Subscribe

Coding the WordPress Loop

Have you ever downloaded a WordPress theme with sloppy code?

There's probably a good chance you have. Either more than 90% of themes have sloppy code or by some astronomically improbable chance, I only ever happen to download themes with sloppy code. I'm hoping to open the eyes of at least a small percentage of theme authors and help usher in a new era of neater code.

I'm going to break down the basics of the WordPress Loop in this post and help you write cleaner themes. I'll explain each line of code along the way and point out references to the outside information.

I know. I know. A post about The Loop doesn't sound too exciting, but maybe this is just what we need — a journey back to the basics. Take a look at the example file to see what The Loop should look like. The important thing here is not to focus on the code but to notice how the code is formatted. See the plentiful whitespace? I'd like to see more of that in themes.

- Example Loop (use to follow along with the post)

> More seasoned theme authors might have slightly different methods for displaying content than in this guide. This should serve as a reference, or a starting point, for using correct WordPress template tags and proper code formatting.

What is The Loop?

The Loop is what makes WordPress go. It's how all of the WordPress magic happens. It's the most important thing. Everything else is secondary.

To be more specific, it cycles through posts, allowing us to display them in any manner we want.

The first step: Opening The Loop

Before we display our posts, we have to open The Loop. Most theme authors know how to get this right, which at least shows potential.

```php
<?php if(have_posts()) : while(have_posts()) :
```

Notice that before the loop starts, we are actually checking if there are any posts. The `if(have_posts())` checks if there are any posts available. If there are posts, we should **loop** through them.

The `while(have_posts())` begins the loop. It will begin a continuous cycle, looping through each post available for display.

`the_post()` calls up information about the post that we'll be using and sets the global `$post` variable.

That's quite a lot for one line of code, right? It's pretty important stuff.

Resources:

- The Loop
- The Loop in action
- WP Query: Methods

Displaying the post

This next step is all about a simple line of code that is commonly referred to as the *post div*. We want to wrap all of our post content into one `div` to kind of hold it all together.

```
<div id="post-<?php the_ID(); ?>" <?php post_cl
```

`the_ID()` gives us the ID of the post. Each post has a unique ID.

`post_class()` is a WordPress 2.7+ function that provides post-specific classes for use in your stylesheet. This is a neat feature because you can display posts with specific tags, categories, and so on differently than other posts.

Resources:

- [the_id()](#)

The post title

There are a few different ways you could code the post title, but I'll show you my favorite method.

```php
<?php the_title('<h2 class="entry-title"><a hre
```

`the_title()` is the WordPress template tag for showing the post title. Pretty simple, right? It takes up to three optional parameters: `before` (before the title), `after` (after the title), and `display` (whether to display or return it for use in a function).

The `get_permalink()` template tag gives us the post URL. It's the permanent link to the post.

`the_title_attribute()` is a template tag I don't see

often enough. Anytime you put a post title in the `title` attribute of a hyperlink, don't use `the_title()`. Use `the_title_attribute()`. This takes care of those pesky posts with things such as quotation marks, leaving us with valid XHTML.

Alternately, you can use `single_post_title()` for displaying a post title in the `single.php` and `page.php` templates.

Resources:

- the_permalink()
- the_title()
- the_title_attribute()
- single_post_title()

The byline

Byline is a term often used in journalism that denotes the writer of a story. Blogs are essentially like newspapers or magazines in this way. You could also go as far as adding a *dateline*, but I like to add my post author, post time, and edit link all within the byline paragraph before the post text.

```
<p class="byline">
        <span class="author vcard"><?php the_au
        <abbr class="published" title="<?php th
        <?php edit_post_link(__('Edit', 'exampl
</p>
```

The first template tag we see in this block of code is
`the_author_posts_link()`. This displays a link to the
author's archive page. Alternately, you could use
`the_author` if you only want to display the author's
display name.

`the_time()` displays when the post was published.
There are several formats you can use for this to get
just the right look. You may be wondering why there
are two instances of `the_time()` in my code. The first
is set in an `abbr` element's `title` attribute. I've done
this so that when a reader hovers over the time
display, they'll see an expanded version with more
details.

The `edit_post_link()` displays a link to edit the post
to only logged-in users that have the capability to
edit posts. Otherwise, the link is not shown. Clicking
on it will take you to the edit post screen in the
WordPress dashboard.

Resources:

- the_author_posts_link()
- the_author()
- Author template tags
- the_time()
- Date and time formatting
- edit_post_link()
- Byline
- Dateline

Displaying the post content

There are two different methods for displaying post content. You can display the entire post or an excerpt. The first method I'll show you is of displaying the entire post. You should use always use this in `single.php` and `page.php`. Most themes will use this method in `home.php` or `index.php` as well.

```
<div class="entry-content">
        <?php the_content(__('Continue reading'
        <?php wp_link_pages('before=<p class="p
</div>
```

Note that I'm wrapping the post content in its own `div` here. This gives us better separation and more styling options.

`the_content()` displays the entire post. It doesn't get much simpler than that. *Continue reading* is the text shown in the link when the post author cuts off the post with the `<!--more-->` tag.

`wp_link_pages()` is a piece of code that's not added to enough themes. This is a vital piece of code! When an author uses the `<!--nextpage-->` quicktag to break his or her posts into more than one page, we need to display the links to each page.

If you only want to display the excerpt of the post, follow along. Displaying the excerpt is useful for any archive-based template files, such as `category.php` and `date.php`, and `search.php`.

```
<div class="entry-summary">
        <?php the_excerpt(); ?>
</div>
```

This will display either the first 55 words of a post or the text entered into the *Excerpt* textarea when on the *Write Post* panel in the WordPress dashboard.

Resources:

- Advanced post editing (i.e. writing excerpts)
- WordPress definition of excerpt
- the_excerpt()
- the_content()
- Customizing the read more
- Write Post Subpanel: Quicktags (`<!--more-->` and `<!--nextpage-->`)
- wp_link_pages()

Showing the post metadata

After showing the post content, I generally display the post metadata, which is additional information

about the post. I like to put the category, tags, and comments link in this section. You can combine this with the byline and/or mix and match the two however you want.

```
<p class="entry-meta">
        <span class="categories"><?php _e('Post
        <?php the_tags('<span class="tags"> <sp
        <span class="sep">|</span> <?php commer
</p>
```

`the_category()` shows your post categories. I'm using a comma as the separator in the code above. You can use anything you like there to separate multiple categories.

`the_tags()` displays all of your post tags. Notice how it looks a bit different than the display of the categories though — everything is tucked neatly away inside of `the_tags()`. Doing it this way assures that the tags or any additional text you add will only display if the user tags the post. This keeps users that don't use tags from getting frustrated with seeing *Tagged* followed by nothing.

The last part of the post metadata that we'll show is a link to the comments of the post. The template tag for this is `comments_popup_link()`. It's important to use this because it serves two purposes: links to a pop-up window for users that are using pop-up comments or links to the post comments for users

that aren't using pop-up comments. Note that this won't work on `single.php` and `page.php` because you'll already be on the page with the comments list.

Resources:

- the_category()
- the_tags()
- comments_popup_link()
- Metadata

Closing The Loop

The tough part is over. Now we must close this thing off to make it all work correctly. We must also display a message in case no posts were found.

```php
        </div>

        <?php endwhile; ?>

    <?php else : ?>

        <p class="no-posts"><?php _e('Sorry, no

    <?php endif; ?>
```

Localization

The more observant of you may have noticed the

several instances of text with the word `example` in it, like these lines:

```
__('Pages:','example')
_e('Sorry, no posts matched your criteria', 'ex
```

This process is referred to as *localization*, which makes your WordPress theme ready for translation. I'll cover this in a later post in more detail. Making your theme ready to use in any language should be something to consider.

Resources:

- [Translating WordPress](#)
- [Localizing plugins and themes](#)

Start writing cleaner loops

Now that I've shown you the proper template tags and have pointed out loads of resources, I expect you theme authors to start making cleaner themes.

Fancy gizmos and gadgets don't impress me much. Some cool JavaScript feature isn't so cool if your code isn't understandable.

Think about your theme users. I know most of us theme developers don't like for users to muddle in our code, but the honest truth is that end users

tinker with the code. Of course, there are countless ways to do things without diving into the templates, but they'll continue doing it anyway. The best thing we can do for them is make the code as readable as possible.

Clean code also helps others learn from what you've done. Why not help a new theme designer looking at your theme code by making things a little neater?
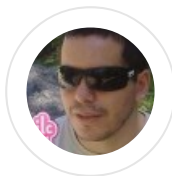
Tagged WordPress, WordPress Tutorials

46 Responses
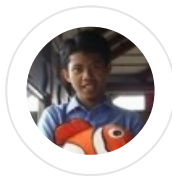
Agus MU
JANUARY 23, 2009 *at* 8:59 PM | Permalink

It's very complete! You always write a complete tutorials. I like it. Thanks Justin…

Reply

Elliot
JANUARY 23, 2009 *at* 11:48 PM | Permalink

Is there a reason behind using endwhile + endif? why not use braces?

Billyboylindien

JANUARY 24, 2009 *at* 2:08 AM | Permalink

Verry simple and complete doc.
Merci beaucoup

Pierre K.

JANUARY 24, 2009 *at* 2:21 AM | Permalink

Hello Justin
I have in fact read many articles about the loop, but yours is a model of precision and clarity. And I didn't even know about the_title_attribute()...
"Bravo !" as we say in french.

Justin Tadlock

JANUARY 24, 2009 *at* 3:33 AM | Permalink

Agus MU — One of my goals for this blog is to

start writing more in-depth tutorials, explaining each step along the way. In the past, some of them were not as complete, so I'm trying to change that and be as thorough as possible.

[Elliot](#) — I simply prefer `endwhile` and `endif` over braces. It's more understandable and readable to me. I also think it would be easier for a new user (someone that doesn't understand PHP) to read. You are certainly welcome to use braces though.

[Billyboylindien](#) — Thank you. I was hoping to keep it simple.

[Pierre K.](#) — `the_title_attribute()` is something I actually only learned about a few months ago. If you notice in some of my older themes, that I wasn't using it. It's now one of my favorite template tags to bring up in WP discussions.

Reply

Hugo Baeta

Just wanted to commend you for writing such an amazing and precious article – oh, and written fantastically well!

I've been playing with WordPress for a while (since 2.0), and I feel this article is perfect to rewind and review your coding – something worth doing!

Thank you so much for this and for the great website.
Cheers!

AC

Justin, thanks so much for the clarity of this document. I've only recently been trying out this powerful feature on my own, and I've put a little something together on my website in the Superior Software List page. I plan to do much more and this explanation will be my go to source for learning the details. Right now, I've just cut and pasted snips of code from your themes to put my page together and it's working. But like i said, I'd like to do more.

It's so great how powerful wordpress is. Someone like me could not have put together a page as nice as I currently have 10 years ago without learning a lot of code and spending a lot of time on it. Thanks again.
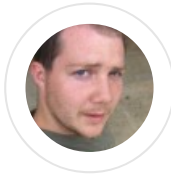
Justin Tadlock

Hugo Baeta — Thank you. Reevaluating how we code is an important part of coding. Every now and then a refresher on the basics is just what we need.

AC — The best thing when learning to code your own themes is to start with the basic foundation. You not only need to know what code to use, you need to understand it. So, just keep this page bookmarked and return to it anytime you need to see how The Loop works.

Reply

Justin Tadlock on properly coding the WordPress Loop — WPCandy — WordPress Themes, Plugins, Tips, and Tricks

Stefano

Hi Justin,

great tutorial!
It's a "must read" for everyone that wants to write (and maybe distrubute) its own theme.

Can I translate it to italian and post to my reader at WordPressMania.it obviously with linkback and so on… Still too many italians can't read english…

Stefano

Reply

BloggerSavvy

JANUARY 28, 2009 *at* 10:12 AM | Permalink

I like that you break this down, statement by statement. In particular, paying attention to detail is what makes this post more valuable. I've been able to use it to do a bit of tweaking. I'm not a programmer, but I can edit or change existing code, so this is a great "lesson". This post is definitely "in the loop" 😃
Bookmarked!
Thanks!

Reply
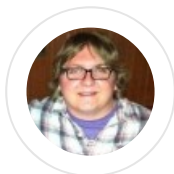
Tommy Day

JANUARY 28, 2009 *at* 10:59 AM | Permalink

Justin,

Great article, I feel like I finally understand the loop now, when before I just felt that I was copying and pasting it every time.

Thanks!

Reply

[Richard](#)

JANUARY 28, 2009 *at* 12:43 PM | [Permalink](#)

Just like to say thanks for writing such an in depth, concise and easy to understand tutorial. I'm currently developing my first complete wordpress theme and have searched high and low for a proper breakdown on the loop. This has been the best explanation to date, just popped you into my RSS reader for future tutorials.

Thanks

Reply

[Weekend Links - Jan 29, 2009 | OMNINOGGIN](#)

JANUARY 31, 2009 *at* 2:25 PM |

[Permalink](#)

Andrew

I really do think that many of the theme designers out there need to go to 'WP Theme Coding 101' to learn the basics about coding WP Themes.

When helping people troubleshoot problems with their themes, I'm often pulling apart poor code that makes finding what I want difficult.

Anyways, very good post, and I have bookmarked it for the time that I need to create a theme using very good coding standards.

Reply

WP Thoughts » Blog Archive » Coding the WordPress Loop - Part 1
FEBRUARY 8, 2009 *at* 5:15 PM |
Permalink

juliet

can you show me a code that the output is the

multiplication table..?

from 1-12...

hope you will grant my wish..tanx..!!!

Vitty Library

MARCH 4, 2009 *at* 4:46 PM | Permalink

Thanks for the guide i had developed my first theme thanks a lot

WordPress Loop | Theme Heven

APRIL 23, 2009 *at* 10:27 PM | Permalink

Marlice

MAY 25, 2009 *at* 4:14 AM | Permalink

Hello Justin,

just read a tutorial in the mag "Web designer" (http://www.webdesignermag.co.uk) on how to create a wordpress theme by Dan Philibin and Michael Castilla. They reference to your great article. Thank you very much for your great work and greetings from Germany 😃

Marlice

Adi

Is this about setting the single post template..

Kevin Lucius

Thanks a lot! I am just learning wordpress and php and this was very easy to follow and helped a lot. You really took time to explain every little bit. Thanks again

Alex

Thanks for this tutorial. It's well written and I appreciated that you offered an example to look at before diving in. Great job! This was really helpful.

kronos

AUGUST 28, 2009 *at* 12:56 PM | Permalink

Yea, this is a real nice blog. Thanks you

Codelobster vs YouCMSAndBlog WYSIWYG Vs Aptana IDE for PHP, CSS, HTML, CMS (WordPress, Joomla, etc) « Sadhas Blog

JANUARY 27, 2010 *at* 1:16 AM | Permalink

Никита
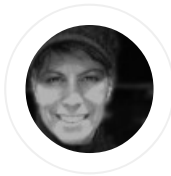
FEBRUARY 2, 2010 *at* 12:09 PM | Permalink

I feel like I finally understand the loop now !! $)

WP: Loop de Loop « Pubmedia Commons

MARCH 19, 2010 *at* 7:03 AM | Permalink

Suzanne

Justin I'm experimenting with a custom content type that would be similar to a Facebook status update or tweet – short blurbs I don't want in my blog entry stream. I display these blurbs on their own index page using a standard loop, but unlike a normal index loop, I want comments to show under each entry so it looks similar to the Facebook wall. The comment form does NOT need to be there but I'd like the comments to show. Is this even possible within the loop?

Reply

WP: Loop de Loop | PubMedia Commons

David A. Kennedy | 31 Resources for Powering a Website with Wordpress

stephen

This article helped me understand how to use the loop in a nice, clean fashion. I hate sloppy code and trying to learn from themes wasn't getting me anywhere. Thanks so much!

Richard

Hi Justin
Thanks for your good explanation of the basic WP loop.

I am presently struggling trying to find how to add 'Permalink' link at the end of WP3 loop in my theme anIMass!
It is somewhere in here! But what & where & I have spent a lot of today trying to get it working?!

```php
<?php edit_post_link( __( 'Edit', 'anIMas
        |


                <!-- .entry-utility -->
        <!-- #post-## -->
```

Oh the joy & pain of jolly old WordPress!

Reply
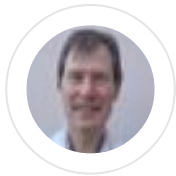
Lu
NOVEMBER 11, 2010 *at* 7:22 AM | Permalink

Hi Justin. Thank you so much for this tutorial. I've hunted high and low for a simple tutorial that explains to me what The Loop is all about, what each bit is for, why it's necessary and where it goes. You words are simple and concise. You can't imagine how useful it has been! 😃

Reply

Manoj
JANUARY 10, 2011 *at* 9:12 AM | Permalink

Thanks much Justin. On my main / home page, I have a pic and excerpt of the most recent post. I want to add a previous / next link and when user clicks just display the excerpt and picture of the previous / next post. Any tips on how to accomplish this. This is what I have in my loop currently:

```
<a href="" rel="bookmark">
        Category:  | Published by: |
<a href="" rel="bookmark" title="Permanen
```

Reply

Designing a Wordpress Theme

JANUARY 15, 2011 *at* 6:57 PM |

Permalink

FC Barcelona

SEPTEMBER 5, 2011 *at* 1:28 AM | Permalink

Hi, its a good post, maybe you could also make a tutorial about using the loop twice on a page, what to avoid and the proper way of doing it?

FC

Reply

Liam

OCTOBER 20, 2011 *at* 11:41 AM | Permalink

Hey, thanks for the tutorial, its done me wonders so far, I was wondering however if its possible to control an image within the entry

content as well as the writing .. Im trying to position my post images to the left of the text, but for some reason I can't seem to get it to work ? Any ideas anyone ?? Thanks very much !!

Reply

[WordPress ile ilgili takip edilesi 10 muhteşem blog | Blog Wolkanca](#)
JANUARY 4, 2012 *at* 2:49 PM | **Permalink**

[Tim](#)
MARCH 6, 2012 *at* 2:19 PM | **Permalink**

Thanks! You really simplified the Codex.

Reply

Loren Helgeson
APRIL 12, 2012 *at* 12:19 AM | **Permalink**

Wow! Back to basics, indeed. I've used most the standard loop a lot in my three years of working with Wordpress. Recently, I've been trying to breakdown the loop one item at a time, since I do a lot of custom themes and I'm always trying to make each new theme better

than the last.

I was never got a proper indoctrination into the importance of a clean loop. So, thank you for laying this out in as complete a detail as you have.

drwain

It's been a few years since you posted these instructions but for a web builder incorporating WordPress for the first time this is exactly what I was looking for.
Thank you so much for you precise instructions and code samples.
I had searched the web for a while and only your instructions gave me the look I was after.
Cheers 😀

William Mosley

Thank you for the post. I found it very useful. I was also looking to have more control over the styling of the loop and posts in it. Your code

provides that.

Gaurav Ramesh

AUGUST 26, 2012 *at* 9:32 PM | Permalink

Hey Justin.. Thanks for such nice articles. I've just started with Wordpress development and I'm lucky to have found your blog right in the beginning. This article and "sidebar in wordpress" are awesome so far.

Create Your WordPress Loop In 5 Easy Steps | Simplified Scripting

MAY 16, 2013 *at* 1:48 PM | Permalink

deWEBlopers – Cómo crear un loop WordPress perfecto

JUNE 16, 2013 *at* 9:18 PM | Permalink

Cómo crear un loop WordPress perfecto | CURSO WORDPRESS

JUNE 17, 2013 *at* 5:13 AM | Permalink

Leave a Reply

Name *

Email *

Website

Comment

Post Comment

By submitting a comment here you grant this site a perpetual license to reproduce your words and name/Web site in attribution.

Please use your real name or a pseudonym (i.e., pen name, alias, *nom de plume*) when commenting. If you add your site name, company name,

or something completely random, I'll likely change it to whatever I want.