



WordPress conditional tags

One of the things I like the most about working in WordPress code, although it can be really frustrating at times, it's the large amount of conditional tags that are available. This post will explore what they are, and some examples of how we use them.

What they are

Conditional tags are pre-packaged PHP functions that allow you to customize your WordPress theme based on whether the current item matches a condition or not. Conditional Tags usually work with PHP if /else Conditional Statements.

The if/else code checks if a statement is true or false. The example uses the `is_single()` conditional tag.

If the statement is found to be true, the first set of code is executed and we echo `This is a single post!`.

If it's false, we skip the first block of code, and execute the else block and echo `This is not a single post!` instead.

```
<?php
if ( is_single() ):
    echo 'This is a single post!';
    echo 'This is not a single post!';
endif;
'This is not a single post!'?>
```

How do they work?

Expanding on the example on the previous section we can use one or more conditional tags and one or more logical operators in either a template or in a function that hooks to a trigger or action for later execution.

We'll walk through some examples as a means to explain what they do.

Are we in the home page?

WordPress makes a difference between the front page (the main page of the site) and the main page (the index page for the blog). They may be the same or they may not; in this particular case we only want to output something if the

```
<?php if ( ( is_front_page() && is_home() ) ) :  
    echo "This" "This is front and home page" cho "do something cool";  
endif  
"do something cool"?>
```

Likewise, WordPress make conditional tags for different types of content like single posts or pages. The next example will branch out and check if the user is visiting a single post or a page and echo different output depending on what we visited.

```
<?php if ( is_single() ) :  
    echo "Visi" "Visiting Single Post" eif ( is_page() ) :  
    echo "Visiting Page";  
endif;  
"Visiting Page"?>
```

The final example is to check whether a user is logged in and display different content to them than to anonymous visitors.

```
<?php if ( is_user_logged_in() ):  
    echo "Show" "Show secret for logged in users" e :  
    echo "Shhh, user is not logged in!";  
endif;  
"Shhh, user is not logged in!"?>
```

We've used echo statements in the example; in real production code we'd use more complex code, including WordPress specific functions.

The last example will work through actual WordPress code to, hopefully, fully illustrate how conditional tags work.

The code uses conditional tags in two places:

1. The first one is to check if we're in the site's front page **and** the blog home page (they are one and the same)
 1. IF the condition matches we use h1 for the title
 2. If not then we use p elements for the title since we want to style it differently
2. Check if we have a description of the blog (another name for the tag line) **or** we're looking at the customizer view of the page
 1. If neither condition are met we don't show the description

```
<div class="site-branding">
  <?php the_custom_logo(); ?>
  <?php if ( is_front_page() && is_home() ) : ?> <!-- 1 -->
    <div class="site-branding__text">
      <h1 class="site-title"><<div class="site-branding__text">_PHP3___</div>
      ___PHP4___
      <p class="site-title"><a href="___PHP5___" rel="home">___PHP6___</a>
      ___PHP7___ </a><!-- 2 --> <p class="site-description">___PHP8___<
      ___PHP9___
    </div> <!--<p class="site-description"><!-- .site-branding__text -->
  </div><!-- .site-branding -->
```

There is a huge number of conditional tags available. If you want to see the full list, check the [List of Conditional Tags](#) at wordpress.org.

Links and Resources

- [Conditional Tags](#)
- [List of Conditional Tags](#)
- [How to Use WordPress Conditional Tags](#)
- [The Ultimate Guide to WordPress Conditional Tags](#)