# Mastering WP\_Query: Properties and Methods

掌握WP\_Query：WP\_Query类的属性和方法

Welcome to yet another part of our "Mastering WP\_Query" series. How are we doing so far? I hope you're enjoying these tutorials as much as we're enjoying writing them.

欢迎大家来到“掌握WP\_Query” 系列新的章节。到目前为止你学的怎么样？我希望大家能够在学习这些教程的过程中去享受它，就像我写这系列的教程的时候一样的去享受写的过程。

In this tutorial, you're going to learn about the properties and methods of theWP\_Query class. But first, I think it's appropriate to talk about what "properties" and "methods" are in a class.

在这篇教程中，我们将要学习WP\_Query类的属性和方法。但是，首先我想先讲一下在类中的“属性”和”方法”是什么。

Shall we?

坐稳了，可以开始了吗？

**What Are Properties and Methods, Exactly?**

属性和方法到底是什么?

In PHP, there's a way to write code in a *cleaner* way: object-oriented programming, or OOP. With OOP, we use "classes" as blueprints—I took the term from one of the *classic* posts of Code Tuts+, "[Object-Oriented PHP for Beginners](https://code.tutsplus.com/tutorials/object-oriented-php-for-beginners--net-12762)" by Jason Lengstorf:

在PHP中，有写一个更清洁的方式编码方式：面向对象编程，或OOP。随着OOP中，我们使用“类”作为模板——从Jason Lengstorf帖子“为初学者准备的面向对象的PHP教程”中取了经典的代码例子TUTS+：

*A class, for example, is like****a blueprint for a house****. It defines the shape of the house on paper, with relationships between the different parts of the house clearly defined and planned out, even though the house doesn't exist.*

(And remember that WP\_Query is an essential class of the WordPress core.)

类，举个例子，就像一个房子的模板：即使不存在这个房子，这个模板定义了房子的形状，并定义了房子的不同部分之间的关系明确的计划——就像是建房子的蓝图。

When you understand the notion of PHP classes, "properties" and "methods" become extremely easy to understand since the words are just synonyms for "variables" and "functions". Yep, *properties* are variables of a PHP class and *methods* are functions of a PHP class.

当你了解PHP类的概念，“属性”，“方法”变得非常容易理解。就像是“变量”和“功能”只是同义词。在PHP中，属性是一个PHP类的变量，方法是一个PHP类的功能，这样去理解。

Now we've covered what they are, let's get to know each of those properties and methods.

现在，我们已经介绍了“属性”，“方法”是什么，下面，让我们来认识每个属性和方法。

*Warning:* It wouldn't be wise to change the properties directly. [As the Codex says](http://codex.wordpress.org/Class_Reference/WP_Query#Methods_and_Properties), you should *interact* with them using the methods of WP\_Query.

警告：直接更改属性不是明智的。根据官方文档，你应该学会它们与WP\_Query的方法联合使用。

**Properties of the WP\_Query Class**

**WP\_Query类的属性**

Let's begin with the properties, or the variables, of the WP\_Query class.

让我们开始学习WP\_Query类的属性吧，或者说是它的变量！

**The Query String: $query**

This property stores the query passed to the $wp\_query object.

查询字符串：$query

这个属性储存了传递给$wp\_query对角的查询相关变量。

**The Array of Query Variables: $query\_vars**

This property stores an associative array of the variables (and their values) of$query.

查询变量数组: $query\_vars

这个属性储存了相关联的变量和变量的值，集合成数组，传递给$query。

**The Queried Object: $queried\_object**

This property stores the currently queried object, like the $post object if it's a post query or the $author object if it's an author query.

查询对象：$queried\_object

这个属性储存了查询的对象，就像$post对象是针对文章查询的，$author对象是针对作者查询的。

**The ID of the Queried Object: $queried\_object\_id**

This property stores the ID of the queried object.

被查询对象的ID：$queried\_object\_id

很好理解，这个就是查询对象的ID。

**The Posts Returned From the Query: $posts**

This property stores the posts returned from the query.

从文章查询返回的结果：$posts

这个属性储存了从查询返回回来的文章。

**The Number of Posts Being Displayed: $post\_count**

This property stores the number of posts for the current query.

显示文章的数量：$post\_count

这个属性储存了当前查询的文章的数量。

**The Number of Posts Returned from the Query: $found\_posts**

This property stores the number of posts without the LIMIT clause of the SQL query.

查询返回的文章的数量$found\_posts

这个属性储存了没有LIMIT子句的SQL查询的文章数量。

**The Number of Pages: $max\_num\_pages**

This property stores the number of pages—it's calculated by dividing $found\_postsby $posts\_per\_page.

分页数量:$max\_num\_pages

这个属性储存了文章的数量——它是由$ posts\_per\_page除以$ found\_posts计算而来的。

**The Index of the Current Post: $current\_post**

This property stores the index number of the current item in the Loop. For example, it's -1 if the Loop has just started and it's incremented by the next\_post() method.

当前文章索引：$current\_post

这个属性储存了当前Loop循环中的文章索引。例如，当Loop循环刚刚开始的时候，$current\_post被next\_post()方法设置成-1。

**The Current Post: $post**

This property stores, well, the current post.

当前文章：$post

这个属性不用多作解释，储存了当前文章。

**Conditional Tag Booleans: $is\_{conditional}**

The following properties are stored as booleans, giving information about the current post's status:

条件标签：$is\_{conditional}

下面的属性是布尔类型的，是为了标定当前文章的状态

* $is\_single: Checks if it's a single post of any post type (except "attachment" and "page" post types) or not.
* $is\_page: Checks if it's a page or not.
* $is\_archive: Checks if it's an archive page or not.
* $is\_preview: Checks if it's a post preview or not.
* $is\_date: Checks if it's a date-based archive page or not.
* $is\_year: Checks if it's a year-based archive page or not.
* $is\_month: Checks if it's a month-based archive page or not.
* $is\_time: Checks if it's a time-based (hourly, minutely or secondly) archive page or not.
* $is\_author: Checks if it's an author's archive page or not.
* $is\_category: Checks if it's a category archive page or not.
* $is\_tag: Checks if it's a tag archive page or not.
* $is\_tax: Checks if it's a taxonomy archive page or not.
* $is\_search: Checks if it's a "search results" page or not.
* $is\_feed: Checks if it's a feed or not.
* $is\_comment\_feed: Checks if it's a comment feed or not.
* $is\_trackback: Checks if it's a trackback or not.
* $is\_home: Checks if it's the main blog page or not.
* $is\_404: Checks if it's a 404 error page or not.
* $is\_comments\_popup: Checks if it's a comments popup window or not.
* $is\_admin: Checks if it's the administration panel or not.
* $is\_attachment: Checks if it's an attachment or not.
* $is\_singular: Checks if it's a single post of any post type (including "attachment" and "page" post types) or not.
* $is\_robots: Checks if it's a query for the robots.txt file or not.
* $is\_posts\_page: Checks if it's the "Posts Page" (set on the "Reading Settings" page in the admin panel) or not.
* $is\_paged: Checks if it's a paginated query and it's not the first page.
* $is\_single: 检查是否为单页的某类型文章（除了“附件”和“页”文章类型）
* $is\_page: 检查是否的是页面。
* $is\_archive: 检查是否是分类网页。
* $is\_preview: 检查是否是预览页面。
* $is\_date: 检查是否是一个基于日期的分类页面。
* $is\_year: 检查是否是一个基于年份的分类页面。
* $is\_month: 检查是否是一个基于月份的分类页面。
* $is\_time: 检查是否是一个基于时间（小时，分钟，秒）的分类页面。
* $is\_author: 检查是否是一个基于某作者的分类页面。
* $is\_category: 检查是否是一个类型的分类页面。
* $is\_tag: 检查是否是一个标签的分类页面。
* $is\_tax: 检查是否是存档的分类页面。
* $is\_search: 检查是否是查询返回结果的页面。
* $is\_feed: 检查是否是订阅的页面。
* $is\_comment\_feed: 检查是否是订阅评论的页面。
* $is\_trackback: 检查是否是引用的页面。
* $is\_home: 检查是否是主页。
* $is\_404: 检查是否是404页面。
* $is\_comments\_popup: 检查是否是评论弹出窗口的页面。
* $is\_admin: 检查是否是管理员页面。
* $is\_attachment: 检查是否有附件。
* $is\_singular: 检查是否为单页的某类型文章（包括“附件”和“页”文章类型）。
* $is\_robots: 检查是否是robots.txt查询
* $is\_posts\_page: 检查是否是发布文章页面（在管理员面板中设置“阅读设置”页面中）
* $is\_paged: 检查是否进行了标页，并且不是第一页的标页

**Methods of the WP\_Query Class**

**WP\_Query类的方法**

Now that we're done with the properties, let's move on to the methods (functions) of the WP\_Query class.

当我们学习完属性之后，让我们学习WP\_Query类的方法吧。

**init()**

This method simply initializes the object, setting all properties to NULL, 0 or FALSE.

初始化方法，主要是初始化对象的的属性值为Null，0或者false。

**parse\_query( $query )**

This method uses the $query property to parse the query and populate all other properties (except $posts, $post\_count, $post and $current\_post).

**parse\_query( $query )**

这个方法使用$query属性传递给查询其相应的属性（除了$posts,$post\_count,$post和$current\_post）

**parse\_query\_vars()**

This method reparses the query variables.

这个方法重新传递变量给查询。

**get( $query\_var )**

This method fetches a given query variable.

这个方法获取给定的查询变量。

**set( $query\_var, $value )**

This method sets the given query variable to a specific value.

这个方法设置给定查询变量为一个特定值。

**&get\_posts()**

This method returns the posts requested by the query and populates the $posts and$post\_count properties.

这个方法返回的查询请求的文章，并填充$post和$ post\_count属性。

**next\_post()**

This method increments the $current\_post index and advances to the next post in$posts, returning the current post object. (This method must be used inside a Loop to work properly.)

这个方法$ current\_post索引递增，并推送文章到$posts变量，返回当前post对象（此方法必须使用在循环中）。

**the\_post()**

This method sets the global $post variable with the next post's data. (This method must be used inside a Loop to work properly.)

这个方法设置了下一篇文章的全局变量$post（此方法必须使用在循环中）。

**have\_posts()**

This method checks if there are any posts left to work on, and returns FALSE if there aren't. (This method must be used *before* a Loop to work properly.)

这个方法检查是否任何文章有需要做的后续工作，并如果没有返回FALSE。 （此方法必须使用在循环之前）

**rewind\_posts()**

This method simply resets the $current\_post and $post properties.

这个方法重置了$current\_post和$post属性。

**&query( $query )**

This method calls two of its sibling methods, parse\_query() and get\_posts(), and returns the result of get\_posts().

这个方法调用了同级方法parse\_query()和get\_posts()，并且返回get\_posts()方法的结果。

**get\_queried\_object()**

This method returns the queried object. (It sets $queried\_object if it's not already set.)

这个方法返回查询的对象（如果$queried\_object没有被设置，则设置）

**get\_queried\_object\_id()**

This method, similar to the one above, returns the queried object's ID ($queried\_object\_id).

*Quick Tip:* If there's an ampersand before a method, it means that the method [returns by reference](http://php.net/manual/en/language.references.return.php).

这个方法，就像上一个一样，返回的是查询对象的ID（$queried\_object\_id）

小提示：如果有方法之前有符号&，这意味着该方法通过引用返回。

**Wrapping Up for Today**

I hope I managed to make it clear for you what "properties" and "methods" mean for classes. And if you understood the purpose of the properties and methods of theWP\_Query class, I can say that I've done a decent job!

Do you have anything to add to this article? Share your thoughts with us in the Comments section below. And if you liked the article, don't forget to share it with your friends.

See you in the next part of the series!

今天就到这里。

我希望我设法让大家明白什么是类的“属性”和“方法”。如果你通过本文了解到WP\_Query类的属性和方法，我就觉得已经做了很了不起的事情了。

对这篇文章，想吐槽吗？请在下面的评论部分，与我们分享您的想法。如果你喜欢这篇文章，不要忘了分享给你的朋友。

系列文章下个部分见。