**总的来说，翻译的还可以，但是有几点要说下：**

1. **相同的内容，最好上下的翻译要一致（这里说的不仅是当前文章，而且整个系列都要统一），比如标题，你翻译“精通 wp\_query”，但是下面的第一段，你翻译的却是“掌握 ……”；还有 achieve 有翻译为 存档和归档 ……**
2. **有极少数地方翻译不够准确，希望以后可以自己检查下**

**原文：**[**http://code.tutsplus.com/tutorials/mastering-wp\_query-an-introduction--cms-23023**](http://code.tutsplus.com/tutorials/mastering-wp_query-an-introduction--cms-23023)

**译者：痴空见观 链接：http://wayearn.com**

**Mastering WP\_Query: An Introduction**

**精通 WP\_Query : 使用方法大全**

*by*[Rachel McCollin](http://tutsplus.com/authors/rachel-mccollin) *4 Sep 2015* [*9 Comments*](http://code.tutsplus.com/tutorials/mastering-wp_query-an-introduction--cms-23023#disqus_thread)

**作者：Rachel McCollin 发表日期：2015年 4月**

This post is part of a series called [Mastering WP\_Query](http://code.tutsplus.com/series/mastering-wp_query--cms-818).

[Mastering WP\_Query: Using the Loop](http://code.tutsplus.com/tutorials/mastering-wp_query-using-the-loop--cms-23031)

**这篇是《掌握WP\_Query》系列的一篇博文，下一篇：掌握WP\_Query：学会使用Loop循环**

By default, WordPress will run a query on each page that your site displays, the nature of which will be determined by the type of page being viewed. So if a static page is being viewed, WordPress runs a query to display the page with the relevant ID, while if an archive page is being viewed, the query will retrieve all posts in that archive.

**通常来说，Wordpress会在你所有展现的网页上运行query（查询）功能，查询不同的属性决定了你所看到页面。因此，如果你正在查看一个静态页面，WordPress的运行查询相关ID来显示页面，而如果存档页面被浏览，查询将检索该归档的所有文章。**

But sometimes you want to do things a bit differently. Either in your page's main content or elsewhere on the page (such as in the sidebar or footer), you might want to display some specific content which isn't output by the standard query.

**但有，有的时候我们需要定制内容。这可能是你的文章内容部分或其他页面上的部分（如：侧边栏或页脚），显示一些特定的内容，而不是WP默认查询输出。**

The great news is that WordPress makes that possible with the [WP\_Query](http://codex.wordpress.org/Class_Reference/WP_Query) class. This class provides you with a large number of parameters you can use to define what content you want to output (which needn't be limited to posts) and then write a loop which will show the content exactly as you want to.

**有了WP\_Query类，这一切都好办了。WP\_Query类提供了大量的参数，你可以去设置这些参数来决定什么样的内容将被展示出来（这不仅仅局限于文章页面）。然后，它会运行一个循环函数，来循环展示你所想要展示的内容。**

In this nineteen-part series on Mastering WP\_Query, [Baris Unver](http://tutsplus.com/authors/baris-unver) and I will take you through the ins and outs of WP\_Query so that once you've completed the series, you'll be able to use it in a variety of scenarios and fine tune the way WordPress queries the data in your site's database.

（这一部分我没有终于原文翻译）

**《掌握WP\_Query》本系列的学习，共19篇博文。完成本系列的学习，你会从基础入门到精通WP\_Query，学会在各种场景下使用WP\_Query，轻松使用其完成对Wordpress网站的数据库数据检索/修正功能。**

In this introduction I'll cover the following:

What is WP\_Query?

Why use WP\_Query?

Potential problems / what to be aware of.

**在此文我们将介绍以下内容：**

**什么是WP\_Query？**

**为什么要使用WP\_Query？**

**潜在的问题/有什么要注意的。**

## What Is WP\_Query?

WP\_Query is a class provided by WordPress. The fact that it's a class means that by using it you can quickly access the variables, checks and functions which have been coded into that class in WordPress core, without having to worry about writing all that code yourself. This makes your code more efficient and reliable.

**什么是WP\_Query？**

**WP\_Query是由WordPress提供的一个类。通过WP\_Query，你可以轻松的使用已经集成到WordPress核心代码中的变量、判断和功能，而不必担心自己编写所有的代码。这使你的代码更高效，更可靠。**

If you want to understand exactly how WP\_Query works under the hood, you can see its code in the [includes/query.php](https://core.trac.wordpress.org/browser/tags/4.1/src/wp-includes/query.php#L0) file.

**如果你是个处女座，想知道WP\_Query干了嘛，你可以在includes/query.php中看到它的原码。（么担心啦，无病毒啦~~~全世界那么多同志在使用，嘻嘻）**

WP\_Query consists of four main elements:

* the arguments for the query, using parameters which will be covered in detail in this series
* the query itself
* the loop, which will output post content, titles or whatever you want to display
* finishing off: closing if and while tags and resetting post data

**WP\_Query包括四个部分：**

* **需要查询的参数，或者参数集合。**
* **开始查询。**
* **循环输出——这将输出文章内容、标题或任何你想显示的内容。**
* **查询结束——通过标签判断、重置请求数据。**

In practice this will look something like the following:

**下面我们来举个例子：**

<?php

$args = array(

**// 用于查询的参数或者参数集合**

);

**// 自定义查询**

$query = new WP\_Query( $args );

**// 判断查询的结果**

if ( $query->have\_posts() ) {

**// 通过查询的结果，开始主循环**

    while ( $query->have\_posts() ) {

        $query->the\_post();

**//查询的内容放在这里。**

    }

}

**// ~~保存请求数据~~重置请求数据**

wp\_reset\_postdata();

?>

You could define your arguments when writing the query itself, but I prefer to define those first, as it keeps things neat.

**你可以在查询的时候去定义需要查询的参数，但是我们不建议这样，我们建议：在查询之前准备好需要查询的参数。保持代码的简洁、易读。**

### Resetting Post Data

In the example above I've added wp\_reset\_postdata() after each query. This is important because it resets the query back to the main query being run on that page.

For example, if you're using WP\_Query to run a query in the sidebar, usingwp\_reset\_postdata() effectively tells WordPress that we're still on whatever page is being viewed and that it should work with the default query for that page.

If you don't do this, any other queries being run on the page (including the default query) might break, and any conditional tags checking for what kind of page is being viewed won't work.

**重置请求数据**

**在例子程序中，我们在每次查询之后添加了wp\_reset\_postdata()函数，它可以把本次查询的结果传递到页面上主查询中去。**

例如，如果你使用WP\_Query去展现一组查询数据到侧边栏里，使用wp\_reset\_postdata()去告诉WordPress哪个页面是我们刚刚访问的页面，并且传递参数至页面的主查询。

如果你不这样做，其他任何页面上运行的查询（包括默认查询）都可能会中止，任何判断页面类型的条件标签都不会正常工作。

## Why Use WP\_Query?

If you haven't used the WP\_Query class before, you may be wondering why you should start. Here I'm going to focus on two aspects of this: why use WP\_Query over other methods of writing custom queries, and scenarios where you might want to use WP\_Query.

为什么要使用WP\_Query?

如果你没有使用过WP\_Query类，你可能会想为什么要学习它。这里我们将从两个方面进行回答：为什么使用WP\_Query 而不使用其他自定义的查询方法；哪些地方你可能会想到用WP\_Query。

### Why Use WP\_Query Over Other Methods?

WP\_Query isn't the only method for creating a custom query. There are four more:

* pre\_get\_posts
* get\_posts()
* get\_pages()
* query\_posts() (which you should avoid, as I'll explain)

I'm not going to go into a lot of detail as to how each of these work, but it's helpful to know when they're used:

为什么使用WP\_Query 而不使用其他自定义的查询方法？

WP\_Query不是仅有自定义查询方法。还有以下四种：

* pre\_get\_posts
* get\_posts()
* get\_pages()
* query\_posts()  （不推荐）

**以上的方法我们不过多的介绍，但是你有时间去了解一下，也是很有用处的。**

* [pre\_get\_posts](http://codex.wordpress.org/Plugin_API/Action_Reference/pre_get_posts) is a hook which modifies the main query. You can use it with a conditional tag to check if a certain type of page is being viewed (for example the home page) and then use it to amend the query that's run (for example to remove the most recent three posts, if you're displaying them elsewhere on the page). It's a very efficient way of modifying the main query and should be your first port of call if that's what you want to do. However you can't use it to create a completely new query.

**Pre\_get\_post 是一个修改主查询的hook（挂钩）。通过条件标签去判断是否有一个确切类型的页面已经展示了出来（如：主页），然后使用pre\_get\_post去修改查询（如：删除在某个地方展示的最近的3篇博文）。这是非常高效的方式去修改主查询，也是首先应该想到的。但是，不能使用它去创建一个新的查询。**

* [get\_posts()](http://codex.wordpress.org/Template_Tags/get_posts) and [get\_pages()](http://codex.wordpress.org/Function_Reference/get_pages) are very similar, with the main difference being obvious from their names. These template tags actually use the WP\_Queryclass, so they are another way of doing the same thing, but add an extra step because they call the WP\_Query class instead of you doing it directly. You can only use them to query either posts or pages, while WP\_Query itself is more powerful and lets you query almost anything held in your database.

**get\_post() 和 get\_paget() 除了名字不同，功能类似。这些模板类拓展了WP\_Query类，用不同的方式做了同一件事件，当你在调用WP\_Query类之前，额外的插入了一步，而非直接插入到wp\_query中去。当且仅当查询文章与页面的时候使用get\_post和get\_paget()，而强大的wp\_query，可以包含数据库所有的查询情况。**

* [query\_posts()](http://codex.wordpress.org/Function_Reference/query_posts) modifies the main query but shouldn't be used in plugins or themes. This is because it throws out the main query and starts all over again, replacing the main query with a new query. It is also prone to errors, particularly with pagination, and is inefficient and will affect your page load times. If you need to modify the main query, use pre\_get\_posts instead, and if you want to create a completely new query, use WP\_Query.

**Query\_post() 可以修改主查询（不能在插件和主题中使用）。Query\_post()会跳出主查询并重新开始，把主查询给替换掉。很容易报错，特别是标记页码的时候，它会很低效并且使页面加载时间变长。如果需要修改主查询，使用pre\_get\_post吧。如果你是想重新新建一个查询，请使用wp\_query。**

The diagram below, released by [Andrey “Rarst” Savchenko](http://www.rarst.net/wordpress/wordpress-query-functions/) under Creative Commons license, makes some sense of this:

### When You Might Use WP\_Query

There are many scenarios when WP\_Query will come in useful, and I can't cover them all here, but here's an overview:

* To add a list of related posts under the current post—for example a list of all posts in the same category.
* To create two loops on the same page: for example an [FAQ page](http://code.tutsplus.com/tutorials/creating-an-faq-page-using-two-loops--cms-20444) with the question titles at the top and the content beneath.
* To create a custom list of recent posts in the sidebar or footer of your site, when the Recent Posts widget doesn't do what you need (or you'd rather code it yourself).
* To create [custom queries for taxonomies](http://code.tutsplus.com/tutorials/taxonomy-archives-list-posts-by-a-second-taxonomys-terms--cms-20450), using more than one taxonomy to define what's displayed.
* To query post types which aren't output by the default query, such as[attachments](http://code.tutsplus.com/tutorials/advanced-use-of-attachments-in-wordpress-creating-categorised-gallery-pages--cms-21007).
* To create custom pages with a number of queries for different content types, as I've done in this [example site](http://type-academy.co.uk/temperament-intro/) for a client.

**什么时候你可能需要用到WP\_Query？**

**有很多种情况下，使用WP\_Query会很有效率。下面尽可能多给大家展示：**

1. **在当前文章的后面添加一个相关文章的列表——例如：添加一个相关分类的所有文章的列表。**
2. **在同一页面新建2个循环，例如：在FAQ页面，设置在问题标题处和内容的下面。**
3. **当小工具（近期文章）显示不出你所需要的内容（或者你是大牛，自己来写的除外^\_^），那就用WP\_Query来新建一个自定义近期文章列表，放到侧边栏或者网站页脚。**
4. **创建分类自定义查询，使用多个分类区分要显示的内容。**
5. **查询文章类型不能够由默认查询而输出的，如：附件生成。**
6. **由很多种查询生成自定义的内容多样的页面，正如示例程序：**

## Some Caveats

The WP\_Query class is great. I create a lot of sites needing custom queries, so I use it frequently. But it doesn't come without its downsides. Here are some things you should be aware of:

* If all you want to do is change the way posts are being displayed for a particular content type or archive, don't use WP\_Query. Instead, simply create a template file for that archive or content type, and modify the loop in that template file.
* If you want to display a few more or less posts than would normally be shown on an archive page (for example not showing a particular category), don't useWP\_Query to create a whole new query. Instead use pre\_get\_posts to modify the main query, along with a conditional tag to single out where exactly you want to do it.
* Beware of running too many queries on a page. You could theoretically create a page with hundreds of custom queries, but just think of the server load. If you need more than four or five queries on a page, you might want to consider creating extra pages.

**注意事项**

**WP\_Query很强大，但是它也有缺点：**

1. **如果是为了改变文章显示成为特点的类型或者分类，不要用WP\_Query。而是，简单的为新的内容或者分类新建一个模板文件，在这个模板文件中设置主循环。**
2. **如果是在一个归档中展示一小部分文章（例如：不展示特定的分类文章），不要用wp\_query去生成一个总的查询。而是用pre\_get\_posts去修改主查询，利用一些条件标签去定位出你想要修改的内容。**
3. **一定要警惕在页面上使用过多的查询。理论上，可以在一个页面上使用成百上千个查询，但是请考虑一下你的服务器的负载吧。如果说某个页面需要4到5个查询工作，那么就考虑做一个新的页面模板吧。**

## Summary

The WP\_Query class is a powerful and extremely useful tool for creating custom queries and making your WordPress site behave exactly as you want it to. As we've seen, there are times when you would use other methods to create custom queries, but it does have a wide range of uses.

In the rest of this series, we'll take you through the details of how to use WP\_Queryand get the most from it.

**小结**

**WP\_Query类是一个强大的丰富的工具，它可以让你自定义页面上的查询，使你的Wordpress网站展现成你所需要的内容。同样，我们也可以使用其他的查询方法，如文章中所介绍的。**

**~~在余下的系列中~~在该系列的剩余部分，我们将介绍如何使用wp\_query，并且熟悉掌握它。**