超详细 WordPress 常用函数

WordPress 是目前十分流行的独立博客程序，因傻瓜化安装和使用，其在网民中的应用已近乎普及。但也因为很多新入门的用户几乎对 WordPress 程序没有任何了解，造成使用中碰到问题无法解决，求助也十分不易。而且，根据 月夜 的经验，WordPress 用户学习了解并掌握一些基本的 WordPress 知识尤其是 WordPress 中功能强大使用方便的函数会极大地方便自己的应 用，从而定制一个自己心仪的独立博客。

在这一系列文章中，月夜试着将自己在长期的 WordPress 生涯中摸索积累的 WordPress 函数以自己的语言与朋友们分享，希望这一系列文章能够为朋友们使用 WordPress 程序带来方便。

在讲述下面的 WordPress 函数之前，我们需要明确这样一点，所有的 WordPress 函数都是已经定义好的 PHP 函数，它们都需要写在 PHP 语句中（<?php ?>）才能执行；而且，所有这些函数在 PHP 语句中都应以分号”;”结尾。其实，为了用好这些函数，为你的 WordPress 服务，你最好具有 PHP 语言的基础。

1.bloginfo()

顾名思义，该函数主要用来显示博客信息；而且根据参数的不同，可以用来显示博客信息中的不同部分。常用的有以下几种：

bloginfo(’ name’) 显示博客题名，如“月夜”；默认（不写参数）输出该项；

bloginfo(’ description’) 显示博客描述部分，如“分享网络知识●享受快乐生活”；

bloginfo(’ url’) 输出博客 URL 地址，如 http://www.yueye.org;

bloginfo(’ rss2_url’) 显示博客的 RSS2.0 feed 地址，如 http://www.yueye.org/feed;

bloginfo(’ template_url’) 用来获取 WordPress 博客的模板地址;

bloginfo(’ charset’) 显示博客的编码方式，如“UTF-8”;

一种常见的使用 bloginfo()函数的组合如下:

<a href=”<?php bloginfo(’ url’); ?>” title=”<?php bloginfo(’ description’); ?>” ><?php bloginfo(’ name’); ?>

在月夜博客中，如上信息输出形如月夜的样式。这种形式通常会用来添加博客的底部信息，如 Copyright @ 月夜，经常在主题模板中使用。

需要注意的是 bloginfo()函数只能输出显示这些参量，如果你想在 PHP 语句中使用得到的这些值，则需使用 get_bloginfo()函数，该函数和 bloginfo()使用相同的参数，获得相同的结果。

2.wp_title()

该函数用来显示页面的标题，如在文章页面，则显示文章标题;在分类页面，则显示分类名称;等等。

wp_title()函数可以跟三个参数，即 wp_title(’ separator’ ,echo,seplocation)，其中 separator 是 title 和其余部分之间的分割符号，默认是>>; echo 是个 bool 变量，取 true 显示标题，取 false 则将标 题作为一个 PHP 参量返回; seplocation 定义分隔符的位置，取 right 定义分隔符在标题后面，取其他任何值，都表示将分隔符放在标题前面。

比如形如主题文件夹下 header.php 中的一段代码:

```
<title>
<?php wp_title(’ | ‘,true,’ right’ ); ?>
<?php bloginfo(’ name’ ); ?> - <?php bloginfo(’ description’ ); ?>
</title>
```

在博客首页可以显示这样的效果“月夜 - 分享网络知识●享受快乐生活”；在文章页，可以显示这样的效果“用好 WordPress 不可不知的 50 个函数 | 月夜 - 分享网络知识●享受快乐生活”；而在分类页面，则可以显示这样的效果“网站相关 | 月夜 - 分享网络知识●享受快乐生活”；等等。

3.wp_get_archives()

该函数用来获取博客的文章存档，通过设置函数的参数，可以按各种方式获取，如按月，按年等等。

wp_get_archives()函数后面同样可以跟多种参数，只不过所有参数都需要使用&连接，并放在单引号(’)中以字符串方式传递 给函数，形如 wp_get_archives(’type=monthly&format=html& show_post_count=1&limit=10’)。

如上的参数意义描述如下:

type=monthly 表示按月显示文章存档，可以使用 yearly、daily、weekly 等代替 monthly 表示按年、日、以及周显示文章存档；

format=html 表示使用通常的 HTML 中<li>格式化文章列表；

show_post_count=1 表示在文章存档后面显示属于该类别（年、月等）的文章数量，该参量是个 bool 值；

limit=10 表示显示的文章存档的最大数量为 10，超过次数，则超出部分不显示；

尽管参数稍多，显得略为复杂，但其实只需注意 type、show_post_count 以及 limit 等三个参量即可。

4.wp_list_categories()

和 wp_get_archives()函数类似，wp_list_categories()函数用来获取博客文章的分类信 息，并可以通过设置适当的函数参数，将其显示出来。该函数的参数和 wp_get_archives()函数类似，都需要使用&连接，放在单引号 (’)中以字符

串方式传递。形如 wp_get_archives(’ orderby=name&order=ASC& show_count=1&use_desc_for_title=1&feed=订阅&exclude=2,5& number=10 ’)。

如上示例中，函数各参数的意义如下:

orderby=name 表示按照分类名称的字母先后顺序显示分类信息，可以将 name 换为 ID 等;

order=ASC 表示按照分类名称的字母的升序显示分类信息，将 ASC 改为 DESC 表示按降序;

show_count=1 在每个分类名称后面显示属于该分类的文章数;

use_desc_for_title=1 使用该分类的描述信息为每个分类名称超链接添加一个 title 属性;

feed=订阅: 在每个分类信息旁边添加一个名为“订阅”的超链接，提供该分类的 RSS 订阅;

exlude=2,5: 在显示的分类中去除 ID 为 2 和 5 的分类；也可以用 include=2,5 表示只显示 ID 为 2 和 5 的分类;

number=10: 表示只显示最先的 10 个分类。

5.get_the_category()

get_the_category()函数用来返回当前文章所属的类别的若干属性所组成的一个数组，该数组包括以下内容:

cat_ID: 当前类别的 ID(也可以写作’ term_id’);

cat_name: 当前类别的名称(也被写作’ name’);

category_description: 当前分类的描述(也可以写作’ description’);

category_count：属于当前分类的文章数量(也被写作’count’)。

具体的使用方法，我们通过下面的几个句子来说明：

形如 get_the_category()->cat_ID 的语句，返回当前文章所属分类的 ID 号；

形如 get_the_category()->description 的语句，返回当前文章所属分类的描述；等等。

6.the_category()

该函数返回当前文章所属的类别名称，而且是文章类别的超链接形式。

默认的无参数形式 the_category()直接以超链接形式显示类别名称，显示为：精品推介；

可以在函数中跟上分隔符等参数来格式化输出，如 the_category(’-’)，若当前文章属于两个以上分类，可以显示这样的形式：精品推介-经验知识；如只属于一个分类，则显示为这样的形式：精品推介。

7.category_description()

该函数以分类的 ID 为输入，得到该分类的描述。常和 echo、get_the_category()配合使用，将当前分类描述输出：

```
echo category_description(get_the_category()->cat_ID);
```

如上语句，get_the_category()得到保存有当前分类信息的一个数组；cat_ID 为该数组中该分类的 ID；将该 ID 输入给 category_description()函数，即可得到该分类的描述；然后使用 echo 将其输出。

但经月夜试验，使用如下的语句可以实现和上面语句相同的功能：

```
echo category_description();
```

这可能是因为该函数在默认无参数输入的情况下会输出当前分类描述的结果吧。

8.is_home()

is_home()用以判断当前显示的博客页面是否是博客首页，返回的是一个 Bool 值。如果是在首页，则返回 TRUE；否则返回 FALSE。

该函数常用来控制博客侧边栏的显示方式，经常使用如下代码段：

```
<?php
if ( is_home() ) {
//此为在博客首页应该显示的内容
} else {
//此为非博客首页应该显示的内容
}
?>
```

9.is_archive()

is_archive()用以判断当前显示的内容是否是博客存档页面，比如按日期的存档，或者按分类的存档，等等；其和 is_home()函数一样，返回一个 Bool 值。

10.is_page()

is_page()函数判断当前显示的内容是否是博客的独立页面（page），如“月夜私语”、“关于月夜”等页面；它也返回一个 Bool 值。

我们可以在模板中通过该函数判断当前是否是一个独立页面，从而决定是否为当前显示的文章显示发布时间等等。

11.is_paged()

该函数用以判断当前文章是否因为内容过多而分页显示；需要注意的是，如果你在写文章时手动添加了<! -nextpage ->标签，来强制分页的话，该函数并不会因此而返回 TRUE。

12.is_page_template()

is_page_template()函数需要跟一个参数，通常以如下方式使用：

is_page_template(’ guestbook.php’);

藉此判断当前显示的独立页面（page）是否使用了参数所示的模板 guestbook.php；如果不跟参数，函数返回当前独立页面是否使用了模板。

13.is_single()

is_single()用以判断当前显示的页面内容是否是一篇单独的文章。其后面可以跟三种参数，一种是文章 ID；一种是 文章题目(title)；一种是文章名称(slug，文章题目的一种简短说明形式)；或者可以将三种参数组合使用，藉此来判断当前页面内容是否是具体的某 篇文章。

一个简单的例子如下，我们可以通过如下几种方式判断当前显示的内容是否是本文：

is_single(’ 808′);
is_single(’ 用好 WordPress 不可不知的函数（二）’);
is_single(’ functions-must-known-using-wordpress-second’);
is_single(’ 808′,’ 用好 WordPress 不可不知的函数（二）’,’functions-must-known-using-wordpress-second’);

在这里，月夜需要对上述最后一种方式做些说明：该函数后跟三个参数时，有优先级，如果第一个参数符合条件，则返回 TRUE；否则，则返回 FALSE；貌似后面的参数并没有什么意义。

14.is_category()

该函数用以判断当前显示的页面内容是否是一个分类页面，如网站相关；其中无需参数。函数返回一个 Bool 值。

15.is_tag()

is_tag()用以判断当前显示的页面是否是一个标签页面，比如 WordPress；其后也不需要跟参数。该函数同样返回一个 Bool 值。

16.is_date()

此函数用以判断当前显示的内容是否为按时间归档的页面，比如 2009 年四月，或者 2009 年 4 月 8 日，等等。

17.is_day()、is_month()、is_year()

这些函数用以判断当前显示的内容是否为按天、按月、按年份归档的页面。它们和 is_date()类似，只不过将归档时间更具体化而已。

18.is_author()

该函数用以判断当前显示的内容是否为以作者名归档的页面，比如月夜博客的 admin 作者页面。

19.is_admin()

is_admin()函数用以判断当前是否在控制面板页面，或者管理员面板页面。

20.get_bloginfo()

该函数和我们前面的文章用好 WordPress 不可不知的函数（一）中介绍的 bloginfo()函数实现近乎相同的功能。主要用来显示博客的信息；而且根据后跟参数的不同，会输出博客的不同信息。

其后不跟参数时，get_bloginfo()可以显示博客名称，形如“月夜”；

后跟其他参数时，可以显示对应的信息，比如 get_bloginfo（‘description’）用以显示博客描述信息；

其他还可以使用的参数包括 name、url、wpurl 以及 admin_email 等等。但因为其与 bloginfo()函数实现相同的结果，所以，在 bloginfo 能够实现的情况下，月夜不推荐使用 get_bloginfo()函数。

21.query_posts()

query_posts()函数结合适当的参数用来控制哪些文章会在页面上显示。

形如 query_posts(”cat=3,6&cat=-5,-10″)表示取分类 ID 为 3 和 6 的文章显示，不取分类 ID 为 5 和 10 的文章显示；

形如 query_posts(”order=ASC&showposts=10&offset=1&orderby=date&posts_per_page=5″)意义如下：

order=ASC 表示按照升序排列，取为 DESC 则表示按降序；

showposts=10 则表示获取 10 篇文章；

offset=1 表示取最新的文章；

orderby=date 表示将文章按照日期排序；

posts_per_page=5 表示每页显示 5 篇文章。

需要注意的是该函数只是将文章内容从 MySQL 数据库中查询出来，要将其显示，还需要与其他语句配合，比如一个经常在侧边栏中使用的形式如下：

```
<li><h2>最近文章</h2>
<?php query_posts(’ showposts=5&offset=1′); ?>
<ul>
```

```
<?php while (have_posts()) : the_post(); ?>
<li><a href=”<?php the_permalink(); ?>” title=”<?php the_title(); ?>”
><?php the_title(); ?></li>
<?php endwhile;?>
</ul>
</li>
```

如上的这段代码用以在侧边栏的指定位置上显示最新的 5 篇文章。

query_posts()函数后面可以跟众多种类的参数，功能十分强大，在此，我们不进行过多介绍。如果时间和精力允许，月夜会在以后的文章中专门撰文详细地为朋友们做一介绍。

22.get_posts()

该函数和 query_posts()函数功能大体相同，都是用来从数据库中查询并得到符合某条件的文章。不过 get_posts()函数的使用有一个固定的形式，如下：

```
<?php
$lastposts = get_posts(’ numberposts=5′ );
foreach($lastposts as $post) : setup_postdata($post);
?>
<h2><a href=”<?php the_permalink(); ?>” id=”post-<?php the_ID(); ?>”
><?php the_title(); ?></h2>
<?php the_content(); ?>
<?php endforeach; ?>
```

即首先使用 get_posts()函数查询得到文章数据，然后使用形如 foreach($lastposts as $post) : setup_postdata($post);的循环将查询得到的文章内容显示出来。

get_posts()函数使用和 query_posts()函数相同的参数，在此我们也不再作详细介绍。

23.wp_list_cats()

该函数和前文用好 WordPress 不可不知的函数（一）中 wp_list_categories()函数实现相同的功能，不过在最新的 WordPress 版本中，此函数已经被弃用，其功能为 wp_list_categories()函数完全取代。

24.get_calendar()

get_calendar()函数用以在 WordPress 上显示日历，日历样式和使用 widget 显示的效果相同，如下图所示:

其后可以跟一个 BOOL 参数，用以控制日历上方星期的显示样式。但经月夜测试，在中文状态下，使用 TRUE 或 FALSE 参数，日历的显示效果并没有什么区别。

25.wp_list_bookmarks()

该函数用来显示博客的友情链接，并可以使用各种参数来控制显示的数量、种类以及样式等等。

形如 wp_list_bookmarks('title_li=&categorize=0&orderby=rand&include=41,40,37,54');的形式，意义解释如下:

title_li=&categorize=0 是一种通常使用的组合，意为不显示 WordPress 后台控制面板中设置的友情链接标题，但所有友链都按照设置的分类显示出来；单独的一个 title_li=还可以用来设置被显示友链的类别名称；

orderby=rand 设置友链的显示方式为随机顺序，当然，还可以设置为其他的方式，比如 id、url、name 等等；

include=41,40,37,54 表示只显示 ID 为这四个数字的四个友链；与此对应，还支持使用 exclude，表示不显示 ID 为多少的友链。

此外，该函数还经常用到的参数有 before 和 after，用来设置每个链接前后的文字，默认的是
和
标记。

26.get_links()、wp_get_links()

这两个函数实现和 25.wp_list_bookmarks()相同的功能，不过此二函数已在 WordPress 升级的过程中为 25.wp_list_bookmarks()所取代。

27.wp_list_pages()

该函数以页面名称的超链接形式显示 WordPress 博客内的所有页面，经常用来建立顶端导航页面，或用来修饰侧边栏。

形如 wp_list_pages('title_li=&sort_column=menu_order&include=12,25,38,57&depth=1&');的函数调用，各参数意义如下：

title_li=用来设置所有显示页面的一个总名称；后面没有参数值时，表示不显示名称；

sort_column=menu_order 用来设置页面的显示顺序，表示按照 WP 后台设置的各页面顺序显示，其他的常用顺序设置可能还包括 post_title、post_date、ID 等等；

include=12,25,38,57 表示只显示 ID 为这四个数值的四个页面；同样，可以使用 exclude 来排除相应 ID 的页面；

depth=1 表示只显示父页面，对所有子页面不予显示；其他数值还包括默认的 0，表示显示所有页面（子页面有缩进）；-1 显示所有页面（子页面无缩进）；等等。

此外，该函数可能会用到的属性还包括 link_before 和 link_after，用于设置显示的页面链接前后的字符。

28.wp_tag_cloud()

顾名思义，wp_tag_cloud()函数用来显示 WordPress 博客的标签云。

一种形如 wp_tag_cloud('smallest=8&largest=22&number=30&orderby=count');的函数调用，各参数的意义如下:

smallest=8 用来设置标签云中显示出来的所有标签中，计数最少（最少文章使用）的标签的字体大小为 8;

largest=22 用来设置标签云的所有标签中，计数最多（最多文章使用）的标签的字体大小为 22;

number=30 设置标签云中显示的最多标签数量为 30;

orderby=count 设置标签云中标签的排序方式为计数（默认），而不是名称（相应参数为 name，widget 调用时的默认值）。

其他常用的参数还包括 include 和 exclude，用来设置在标签云中是否包含或去除 ID 为某数字的标签。

29.wp_register()

wp_register()函数用以向管理员显示“站点管理”超链接；或者当 WP 博客开放了注册时，向未登陆的用户显示“注册”超链接。

该函数不需要什么参数,唯一可能用到的参数形式如 wp_register('前','后'),可以在如上显示的超链接文字的前后分别显示一个“前”字和一个“后”字。当然，你可以据此发挥想象力来个性化自己网站的管理或注册链接。

30.wp_loginout()

该函数用来在指定位置显示一个“登录”链接；当然，如果你已经登录过了，则会相应地显示一个“退出”链接。此函数后面不使用任何参数，所以无法进行灵活的自定义。

不过如果你想自定义自己的 WP 博客的登录或退出链接文字的话，还是可以使用下面 31 中 月夜 介绍的函数 wp_logout_url()和 wp_login_url()。

31.wp_logout_url()、wp_login_url()

使用如上 30 中的函数尽管可以方便地为 WP 博客设置登录、退出链接，但自定义不够灵活。所以，从 WordPress 2.7 版本开始，提供了这里的两个函数。它们分别用来获取 WP 博客退出或登录超链接，然后，我们使用获得的超链接即可编写如下代码，来对 WP 博客的登录和 退出链接进行灵活设置：

<a href=”<?php echo wp_logout_url(); ?>”>点击这里退出

<a href=”<?php echo wp_login_url(); ?>”>点击这里登录

当然，要想实现完美的效果，还需要对访客的登录状态进行判断，使用一个 if 语句，根据登录状态显示相应的菜单项。

32.wp_meta()

该函数通常会紧跟如上 29、30 中的函数后面，其具体在直观显示上没有什么异样，貌似是WP主题为WP插件留下的API Hook，月夜建议朋友们在如上函数后面跟上这一函数。

33.get_recent_posts()

该函数只有当你安装了中文 WordPress 工具箱之后，才能使用。其作用是用来获取最新日志，函数原型如下：

get_recent_posts($no_posts = 5, $before = ‘
+ ‘, $after = ‘’, $show_pass_post = false, $skip_posts = 0)

可以使用$no_posts 控制显示文章数量，$before 和$after 的意义和前面函数中相同；至于后两个参数，一般不必设置，直接取默认值即可。

不过因为该函数与 WordPress 内置的 get_posts()和 query_posts()函数功能重复，所以通常情况下很少使用。

34.get_recent_comments()

其实安装了如上的中文 WordPress 工具箱之后，最常使用的是这个函数，因为 WordPress 程序本身没有内置获取最新评论的函数。该函数原型如下：

get_recent_comments($no_comments = 5, $before = ‘ ‘, $after = ‘’, $show_pass_post = false)

意义显然，和上面函数类似，月夜此处不再赘言。

35.get_recentcomments()

该函数是在安装了 WP-RecentComments 插件之后才具有的功能，与如上 34 中的函数类似。

该函数原型如下：

get_recentcomments(int num, int size)

num 表示返回的最新评论数量；size 表示返回的评论内容的长度。

36.wp_get_post_tags()

该函数用来在某个文章页面或者根据某篇文章的 ID 来获取该文章的 tag，获取的结果被放置到一个 tag 数组中。一个常见的使用方式如下：

```
if (is_single()){
$keywords = “”;
$tags = wp_get_post_tags($post->ID);
foreach ($tags as $tag ) {
$keywords = $keywords . $tag->name . “,”;
}
echo $keywords;
}
```

首先判断是否是单文章页面，如果是，则据当前文章的 ID（$post->ID）来获取当前文章的 tag，然后取得其 name（$tag->name），并将其组合输出。

37.single_cat_title()、single_tag_title()

如名所言，这两个函数用来获取分类页面和 tag 页面的 title，其通常的使用方式如：

```
<?php
$str = single_cat_title();
echo $str;
?>
```

和

```
<?php
$str = single_tag_title();
echo $str;
?>
```

然而，除此之外，single_cat_title()还可以用来在 tag 页面上获取当前页面的 title；但 single_tag_title()却不可用于获取分类页面的 title。

38.get_settings()、get_option()

此二函数与前文用好 WordPress 不可不知的函数(三)中函数 20.get_bloginfo()类似，使用方法也相同，可以通过后跟各种参数来获取 WordPress 博客的相关信息。

比如如下的调用方式:

get_settings(’ name’)或 get_option(’ name’)

可以用来获取当前 WordPress 博客的标题。

39.wp_head()

该函数与前文用好WordPress不可不知的函数（五）中函数 32.wp_meta()相同，是WP主题为WP插件留下的API Hook。

40.get_header()、get_footer()、get_sidebar()和 comments_template()

这几个函数是用来在 WordPress 主题中获取并包含相应的文件的。比如:

get_header()用来包含当前主题文件夹下的 header.php;

get_footer()用来包含主题文件夹下的 footer.php;

get_sidebar()用来包含主题文件夹下的 sidebar.php;

comments_template()用来包含 comments.php。

需要注意的一点是，如果当前主题文件夹下缺少对应的文件，则函数会使用 wp-content/themes/default/文件夹下的对应文件代替。

此外，以上函数后面都不能跟参数，只有 get_sidebar()例外，因为一个主题中可以使用多个 sidebar。形如 get_sidebar(’ up’)的调用方法可将 sidebar-up.php 侧边栏模板文件包含到主题中。

除了以上几个函数之外，在主题中如果想包含一个具体的文件，还可以使用如下方式：

include(TEMPLATEPATH . '/***.php')

如上的函数形式可以将当前主题文件夹下名为***.php 的文件包含进来；其中 TEMPLATEPATH 是当前主题文件夹地址的一个引用（不含末尾的/，所以需要添加上）。

41.have_posts()、the_post()

这两个函数的使用范围有限，通常在 WordPress 的循环中使用，用以获取所有文章。其固定使用形式如下：

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
此处显示文章
<?php endwhile; ?>
<?php else : ?>
此处显示未找到文章时的信息，比如 404 相关
<?php endif; ?>
```

另一种常见的形式是将如上代码中的前两行组合起来（其他地方不变）：

```
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
```

该形式通常会在模板的 index.php、archive.php 或者 single.php 等页面使用。除此之外的其他地方，我们通常不会看到此二函数的身影。

42.the_title()、the_title_attribute()

the_title()函数主要用来获取当前文章的 title，其后可以跟上三个参数（可全部省略，取默认值），调用形式如下：

<?php the_title(’before’, ‘after’, display); ?>

参数 before 用来设置在获取的 title 前面显示的字符内容；after 用来设置 title 其后显示的内容；而 display 是一个 Bool 值，用于控制获取的 title 是否显示出来。

the_title_attribute()函数与 the_title()类似，其使用方法如下：

the_title_attribute(’before=前&after=后&echo=true’)

其中 before=和 after=分别用于设置 title 前面和后面显示的字符；echo=true 或者 false 用户设置获取的 title 字符串是否显示出来。

形如<?php the_title(’当前文章’, ‘的评论：’, true); ?>或 the_title_attribute(’before=当前文章&after=的评论：&echo=true’)的调用形式将会显示如下的结果：

当前文章用好 WordPress 不可不知的函数（七）的评论：

43.single_post_title()、single_tag_title()、single_cat_title()

这一系列函数用于获取当前文章页面、tag 页面或分类页面的 title 字符串。其后可跟两个参数，调用形式如下：

<?php single_post_title(’前缀’,display); ?>

可以使用形如<?php single_post_title(’当前文章：’,TRUE); ?>或<?php single_post_title(’当前文章：’); ?>的调用形式来显示如下的结果：

当前文章：用好 WordPress 不可不知的函数（七）

此外，也可以像 42 中的函数一样，将显示属性设置为 false，把获取的 title 字符串传给一个变量，以供其他语句使用。此时的调用形式如下：

<?php $tt = single_post_title(’当前文章：’,false); ?>

44.the_ID()

该函数后面不跟任何参数，使用如下所示的调用方式：

```
<?php
$id = the_ID();
echo $id;
?>
```

用于获取并显示当前文章页面的 ID 号。不过需要特别注意的一点是，该函数只能在 WordPress 的大循环内使用，在其他地方使用可能也会显示 ID 号，但显示的内容始终不会随文章而改变。

此外，该函数通常还会在如下所示的CSS结构中使用：

```
<h2 id=”post-<?php the_ID(); ?>” >
<?php the_title(); ?>
</h2>
```

可为博客中不同的作者设置不同的 title 样式，以示区分。

45.get_the_ID()

该函数与 44.the_ID()函数实现完全相同的功能，目前 WordPress 官方也没有提供该函数的使用说明。你可以参阅如上 44 中对 the_ID()函数的介绍。特别提醒一点，该函数与 the_ID()类似，也只能在 WordPress 的大循环中才能正确使用。

46.the_time()、get_the_time()

the_time()用来获取并显示当前文章发布的时间，和上面几个函数类似，此函数也是只能在 WordPress 的大循环中使用。

该函数后面可以跟控制日期或时间格式的参数，常用的参数形式如下：

如<?php the_time(’ F j, Y’); ?>的调用形式显示效果为: 六月 13, 2009 (英文状态下显示 June 13, 2009) ;

如<?php the_time(’ g:i a’); ?>的调用形式显示效果为: 7:09 下午 (英文状态下显示 7:09 pm) ;

如<?php the_time(’ G:i’); ?>的调用形式显示效果为: 19:09。

事实上, 除了使用 the_time()函数之外, WordPress 还提供了一个具有类似功能的 get_the_time()函数。该函数除了不具 有 the_time()函数的显示功能之外, 其余功能二者完全相同。使用 get_the_time()函数时, 如欲将获取的时间显示出来, 需要使用专用语 句。

下面, 我们籍此机会来简单了解一下 WordPress 中时间的格式。在 WordPress 中, 通常用于控制时间格式的有一下字符: l, F, j, S, Y, G, g, i, a 等等, 其详细意义如下:

l (小写 L) 用来显示一周之中每一天的名称, 比如星期六, 或者在英文中显示 Saturday;

F 用来显示月份名称, 比如六月, 或者 June;

j 用来显示一月之中的某一天, 比如 13;

Y 用来以 4 位数字形式显示年份, 使用 y 则以末两位数字显示年份, 比如 2009 或 09;

G, g, i, a 等四个字符通常组合使用, 如前例子, 有两种形式:

g:i a 以形如 7:09 下午或 7:09 pm 的形式显示时间;

G:i 以形如 19:09 的 24 小时进制形式显示时间。

S 通常紧跟在 j 后面，表示是否在一月之中某天之后添加英文后缀（st，nd，th 等）。

免费分享教程，爱心接力分享 ing!

Godaddy支付宝人民币付款购买空间教程图解

IX Web Hosting 信用卡和支付宝(Alipay)购买教程

[教程] 购买Godaddy域名赠送 10G免费空间的开通方法(图)

LunarPages美国虚拟主机购买过程图解（支持国内银行卡付款）

超详细的BlueHost 虚拟主机购买教程图解