



WORD PRESS

WordPress高级教程 (晋级版)

编著：夜阑小雨

网站：www.coagenpl.cn

2009年 8月 26日

目 录

精通 WORDPRESS 主题制作技巧	6
<?php get_bloginfo (''); ?> :	6
<?php bloginfo(''); ?> : 显示博客的信息 (参数同 get_bloginfo)	6
<?php wp_list_pages(''); ?>: 列表显示页面链接.....	6
<?php the_ID(); ?>: 特定内容页 (Post)自动生成的 ID(无参数)	6
<?php the_permalink() ?>: 得到内容页 (Post) URL(无参数)	6
<?php the_title(); ?>: 显示或返回内容页 (Post)标题.....	6
<?php the_time('F j S, Y') ?>: 日志发表的日期 格式: 月日年.....	7
<?php the_author() ?>: 显示日志的作者(无参数)	7
<?php the_content('Read the rest of this entry »'); ?> : 日志的内容.....	7
<?php the_tags('Tags: ', ', ', ' '); ?> : 日志的 tag 标签.....	7
<?php edit_post_link('Edit', '', ' '); ?> : 当用户已登录并具有权限时, 所显示的日志编辑链接	7
<?php comments_popup_link('No Comments »', '1 Comment »', '% Comments »'); ?> : 日志的评论情况.....	7
<?php next_posts_link('« Older Entries') ?> : 较早日志页面的链接.....	7
<?php previous_posts_link('Newer Entries »') ?> : 较新日志页面的链接.....	7
<?php wp_get_archives('type=monthly');?> : 按月显示内容存档.....	7
<?php wp_tag_cloud('');?> : 显示 tag 标签.....	8
<?php get_posts(''); ?> : 得到日志的信息.....	8
<?php wp_list_categories('sort_column=&optioncount=&hierarchical=&.....');?> :按照字符顺序显示分类列表.....	10
<?php wp_list_bookmarks('categorize=&title_after=&title_before=&.....');?> : 显示友情链接	10
<?php wp_list_pages('sort_column=menu_order&depth=3&title_li='); ?> :显示深度到 3 的页面链接.....	10
<?php include(TEMPLATEPATH. '/searchform.php'); ?> : 调用 searchform 的文件生成搜索栏	

.....	10
<?php get_calendar(); ?> : 调用 WordPress 自带的日历.....	10
WORDPRESS 基本模板文件.....	10
一套完整的 WordPress 模板应至少具有如下文件：	10
基本条件判断 Tag.....	11
Header 部分常用到的 PHP 函数.....	11
模板常用的 PHP 函数及命令	12
与模板相关的其他函数.....	12
WORDPRESS 函数技巧	13
Loop 循环 (成功)	13
在 WordPress 的第一篇文章中插入 Google 广告 (成功)	13
改变每个分类页面的日志显示数 (成功).....	14
给特定分类添加不同模板 (成功).....	14
为特定分类文章创建不同外观.....	14
样式化不同分类.....	15
在某一个页面单独调用某一个 php 文件；	15
动态标题以实现更友好的 SEO.....	15
动态显示菜单：根据分类及页面类型显示相应条目：	15
动态显示高亮显菜单：	16
自定义顶部导航.....	16
查询文章.....	17
Query_posts(查询文章?)能被用来显示特定文章，你所需做只是你一个模板文件的 Loop 开始之前调用函数.....	17
query_posts() 函数的一些常见用法.....	17
获取文章.....	17
如何按分类和按月显示一个 Archives 存档页面.....	18
页面模板头部声明.....	18
为不同的分类指定不同的图像.....	18
如何添加幻灯片 BANNER 广告	19

5 个你不知道的 WORDPRESS 函数	22
WORDPRESS 一些常用代码：	23
显示最新文章.....	23
显示最新评论.....	23
显示热评文章.....	24
调用 Gravatar(只适应 2.5以上).....	24
调用前两篇日志，后续 3篇日志列表.....	24
随机文章.....	24
安全的调用 Wordpress插件模板代码.....	25
WORDPRESS 常用代码解释.....	25
1. 最新文章.....	25
2. 随机文章.....	25
3. 最新留言.....	26
调用 TAB 页签菜单：	27
添加 head 头文件：判断后调用 js文件.....	27
添加 css 代码到 样式文件内.....	27
添加 tab代码到需要调用的地方.....	28
保存 js代码到目录下.....	31
所有页面的归档：	32
WORDPRESS 主题中常用代码总结：	34
WORDPRESS 简单企业模板,企业网站完成.....	37
WORDPRESS 搭建企业网站.....	40
一，WordPress 搭建中小企业网站思路.....	40
二，注意事项.....	46
创建漂亮的 WORDPRESS 日期按钮.....	47
第一步：输入代码	47

第二步：建立背景图片.....	48
第三步：为其创建样式.....	48
创建动态的置顶文章	48
查询 WORDPRESS 博客资料的 SQL 命令	50
给 WP 添加 CTRL+ENTER 回复快捷键	51
WORDPRESS 的 10 个杀手级 HACK 技巧.....	52
1.只在搜索引擎结果中显示 AdSense 广告.....	52
2.避免在多循环中出现重复文章.....	53
3.用页码代替“上页”和“下页”.....	54
4.自动获取文章图像.....	55
5.创建“发送到 Twitter”按钮.....	55
6. 使用直引号，避免弯引号.....	56
7. 如无 Referrer 请求则禁止发表评论.....	56
8.在 WordPress 导航条中使用 CSS 滑动门.....	57
9. WordPress 博客百变头图.....	58
10.显示撰写计划.....	58
侧边加入登陆界面：	59
获取单篇文章内的关键字及描述方法：	60

精通 WordPress 主题制作技巧

`<?php get_bloginfo (""); ?>` :

将由 WordPress 函数自动生成的信息传回博客，
然后再在其他地方使用你的 PHP 代码

部分常用参数：

default：默认

name：名称

description：说明

url、home、siteurl：网址

charset：网页的编码方式

stylesheet_url：css 文件地址

version：博客所使用的 WordPress 的版本

`<?php bloginfo(''); ?>`：显示博客的信息（参数同 `get_bloginfo`）

`<?php wp_list_pages(''); ?>`：列表显示页面链接

部分常用参数：

post_title：按标题字母顺序排列（默认）

post_date：按创建时间排序

post_modified：按最后修改时间排序

ID：按照 id 排序

post_author：按照作者 id 排序

`<?php the_ID(); ?>`：特定内容页 (Post) 自动生成的 ID (无参数)

`<?php the_permalink() ?>`：得到内容页 (Post) URL (无参数)

`<?php the_title(); ?>`：显示或返回内容页 (Post) 标题

用法：`<?php the_title('before','after',display); ?>`

before：(字符串) 标题前面的文字，默认情况下不显示，

after：(字符串) 标题后面的文字，默认情况下不显示

display：(布尔) 当布尔值为真时，显示标题（默认），当布尔值为假时，得到标题的值但不显示

`<?php the_time('F j S, Y') ?>`：日志发表的日期 格式：月日年

`<?php the_author() ?>`：显示日志的作者（无参数）

`<?php the_content('Read the rest of this entry »'); ?>`：日志的内容

用法：`<?php the_content('more_link_text', strip_teaser, ""); ?>`

more_link_text：(字符串)，当日志有 more 标记时，链接到日志页面

strip_teaser：(布尔)，日志“更多”链接的隐藏 (true) 或显示 (false) (默认)

`<?php the_tags('Tags: ', ', ', '
'); ?>`：日志的 tag 标签

用法：`<?php the_tags ('before', 'separator', 'after'); ?>`

before：(字符串) 用在标签的前面，默认为“tags：”

separator：(字符串) 分隔符，默认情况下不显示

after：(字符串) 用在标签的后面，默认情况下不显示

`<?php edit_post_link('Edit', '"', '| '); ?>`：当用户已登录并具有权限时，所显示的日志编辑链接

`<?php comments_popup_link('No Comments »', '1 Comment »', '% Comments »'); ?>`：日志的评论情况

`<?php next_posts_link('« Older Entries') ?>`：较早日志页面的链接

`<?php previous_posts_link('Newer Entries »') ?>`：较新日志页面的链

`<?php wp_get_archives('type=monthly');?>`：按月显示内容存档

用法：`<?php wp_get_archives('type=&limit=&format=&before=&after=&show_post_count=')`

参数及值：

type：(字符串) 显示档案的方法

l yearly：年

l monthly (默认)：月

l daily：日

l weekly：周

limit：(字符串) 显示文章数目的限制，默认情况下无限制

format：(字符串) 文档清单的格式

html：添加 HTML 列表 () 标签 (默认)

option：添加选择 (<select>) 或下拉式选项 (<option>) 标记

link：添加链接 (<link>) 标记

custom：自定义列表使用之前和之后的字符串。

before：(字符串) 当有 format 标记时用在 format 前面

after : (字符串) 当有 format 标记时用在 format 后面

show_post_count : (布尔) 是否显示文档里面日志的数目 , 1 显示 0 不显示 (默认)

<?php wp_tag_cloud("):?> : 显示 tag 标签

用法 :

```
<?php wp_tag_cloud('smallest=,
largest=,unit=,number=,format=,orderby=name,order=ASC,exclude=,include='); ?>
```

参数 :

smallest : (整数) 文字的最小尺寸 (unit 给予单位)

largest : (整数) 文字的最大尺寸 (unit 给予加单位)

unit : (字符串) 给 smallest 和 largest 添加的单位

number : (整数) 显示的 tag 的数目

format : (字符串) 标签云显示的格式值 :

- | flat : 由空格分开 (默认)
- | list : 以有序列表排列
- | array : 标签被放在一个数组里面 , 并且函数将标签云返回为调用。注意 : 数组是返回 , 而不显示

orderby : (字符串) 标签的排列的顺序值 :

- | name : 名字 (默认)
- | count : 数字

order : (字符串) 排序的次序 (要大写才有效) 值 :

- | ASC : 升序 (默认)
- | DESC : 降序
- | RAND : 随机

exclude : (字符串) 排除某个特定的标签

include : (字符串) 只包含某个特定的标签

```
<?php $myposts=get_posts('numberposts=10$offset=1');foreach($myposts
as $post):?><li><a href="<?php the_permalink();?>"><?php the_title();?></a></li><?php
endforeach;?> : 显示从第一篇开始取 10 篇日志的列表
```

<?php get_posts(' '); ?> : 得到日志的信息

用法 : <?php get_posts('numberposts=, offset=, category=, category_name=, tag=, orderby= ,.....'); ?> 参数及值 :

numberposts : (整数) 返回日志的数目 , 默认值 : 5

offset : (整数) 最新文章的开始点 , 默认值 : 0

category : (整数) 只显示这一类别的文章 (根据 ID), 默认值 : 无

category_name : (字符串) 只显示这一类别的文章 (根据名称或缩写), 默认值 : 无

tag : (字符串) 只显示这一标签的文章 , 若有多个标签 , 用逗号隔开 , 则一

个不显示，应用空格隔开

orderby : (字符串) 文章的排列顺序 (默认 : post_date)

值 :

- | author : 作者 id
- | category : 分类 id
- | content : 内容 id
- | date : 日期 id
- | ID : 文章本身的 id
- | menu_order : 按菜单命令 (只应用于网页 page)
- | mime_type : 按 MIME 类型 (只应用于带附件的)
- | modified : 最后修改的日期
- | name : 按缩略名
- | parent : 按父文章的 id
- | password : 按密码
- | rand : 随机
- | status : 按状态
- | title : 按标题
- | type : 按类型

order : (字符串) 排序的次序 (要大写才有效) 值 :

- | ASC : 升序
- | DESC : 降序 (默认)

include : (字符串) 显示拥有某些 ID 的文章，多个以逗号、/、空格隔开，默认值：无

exclude : (字符串) 排除拥有某些 ID 的文章，多个以逗号、/、空格隔开，默认值：无

meta_key 和 meta_value : (字符串) 只显示那些有特定键值对的自定义域，(如果要用的话，两个值一起)，默认值：无

post_type : (字符串) 文章的类型值 :

- | post (默认)
- | page
- | attachment
- | any

post_status : (字符串) 查看文章的特殊状态 值 :

- | publish (默认)
- | private
- | draft
- | future
- | inherit

post_parent : (整数) 只显示这个拥有这个 id 的文章的子文章，默认值：无

nopaging : (布尔) 启用或禁用分页，默认值：无

`<?php wp_list_categories('sort_column=&optioncount=&hierarchical=&.....');?>` :
按照字符顺序显示分类列表

参数 show_option_all : (字符串) 当 style 设置为 list 的时候, 会有一个值来显示到所有分类的链接, 默认是不显示所有链接
orderby : (字符串) 排序的依据, 默认值: name
值:
| ID
| name (默认)
| count
order : ASC (升序) 或 DESC (降序)
show_last_update : 是 (真) 否 (假) 显示最后修改时间, 1 为真, 0 (默认) 为假
style : (字符串) 显示分类的方式, 值: list (默认); none
show_count : (布尔) 是 (真) 否 (假) 显示日志的数目, 1 为真, 0 (默认) 为假
hide_empty : (布尔) 是 (真) 否 (假) 隐藏无日志的分类, 1 (默认) 为真, 0 为假
use_desc_for_title : (布尔) 是 (真) 否 (假) 给标题设置分类信息, 1 (默认) 为真, 0 为假
child_of : (整数) 只显示这个分类的子分类, 没有默认值
exclude : 除了
include : 只包括
hierarchical : (布尔) 是 (真) 否 (假) 让子分类显示时缩进, 1 (默认) 为真, 0 为假
number : (整数) 设置显示的数目, 默认无限制
echo : (布尔) 得到的结果是 (真) 否 (假) 存在一个变量中, 1 (默认) 为真, 0 为假
depth : (整数) 显示分类的深度, 0 (默认) 为无限制

`<?php wp_list_bookmarks('categorize=&title_after=&title_before=&.....');?>` : 显示友情链接

`<?php wp_list_pages('sort_column=menu_order&depth=3&title_li='); ?>` : 显示深度到 3 的页面链接

`<?php include(TEMPLATEPATH. '/searchform.php'); ?>` : 调用 searchform 的文件生成搜索栏

`<?php get_calendar(); ?>` : 调用 WordPress 自带的日历

WordPress 基本模板文件

一套完整的 WordPress 模板应至少具有如下文件：

- style.css : CSS(样式表)文件

- index.php : 主页模板
- archive.php : Archive/Category模板
- 404.php : Not Found 错误页模板
- comments.php : 留言 /回复模板
- footer.php : Footer模板
- header.php : Header模板
- sidebar.php : 侧栏模板
- page.php : 内容页 (Page)模板
- single.php : 内容页 (Post)模板
- searchform.php : 搜索表单模板
- search.php : 搜索结果模板

当然，具体到特定的某款模板，可能不止这些文件，但一般而言，这些文件是每套模板所必备的。

基本条件判断 Tag

- is_home() : 是否为主页
- is_single() : 是否为内容页 (Post)
- is_page() : 是否为内容页 (Page)
- is_category() : 是否为 Category/Archive页
- is_tag() : 是否为 Tag存档页
- is_date() : 是否为指定日期存档页
- is_year() : 是否为指定年份存档页
- is_month() : 是否为指定月份存档页
- is_day() : 是否为指定日存档页
- is_time() : 是否为指定时间存档页
- is_archive() : 是否为存档页
- is_search() : 是否为搜索结果页
- is_404() : 是否为 “ HTTP 404: Not Found” 错误页
- is_paged() : 主页 /Category/Archive页是否以多页显示

Header 部分常用到的 PHP 函数

- <?php bloginfo('name'); ?> : 博客名称(Title)
- <?php bloginfo('stylesheet_url'); ?> : CSS 文件路径
- <?php bloginfo('pingback_url'); ?> : PingBack Url
- <?php bloginfo('template_url'); ?> : 模板文件路径
- <?php bloginfo('version'); ?> : WordPress 版本
- <?php bloginfo('atom_url'); ?> : Atom Url
- <?php bloginfo('rss2_url'); ?> : RSS 2.0 Url
- <?php bloginfo('url'); ?> : 博客 Url
- <?php bloginfo('html_type'); ?> : 博客网页 Html 类型

- <?php bloginfo('charset'); ?>：博客网页编码
- <?php bloginfo('description'); ?>：博客描述
- <?php wp_title(); ?>：特定内容页(Post/Page)的标题

模板常用的 PHP 函数及命令

- <?php get_header(); ?>：调用 Header 模板
- <?php get_sidebar(); ?>：调用 Sidebar 模板
- <?php get_footer(); ?>：调用 Footer 模板
- <?php the_content(); ?>：显示内容(Post/Page)
- <?php if(have_posts()) ?>：检查是否存在 Post/Page
- <?php while(have_posts()) : the_post(); ?>：如果存在 Post/Page 则予以显示
- <?php endwhile; ?>：While 结束
- <?php endif; ?>：If 结束
- <?php the_time('字符串') ?>：显示时间，时间格式由“字符串”参数决定，具体参考 [PHP 手册](#)
- <?php comments_popup_link(); ?>：正文中的留言链接。如果使用 comments_popup_script()，则留言会在新窗口中打开，反之，则在当前窗口打开
- <?php the_title(); ?>：内容页(Post/Page)标题
- <?php the_permalink() ?>：内容页(Post/Page) Url
- <?php the_category(', ') ?>：特定内容页(Post/Page)所属 Category
- <?php the_author(); ?>：作者
- <?php the_ID(); ?>：特定内容页(Post/Page) ID
- <?php edit_post_link(); ?>：如果用户已登录并具有权限，显示编辑链接
- <?php get_links_list(); ?>：显示 Blogroll 中的链接
- <?php comments_template(); ?>：调用留言/回复模板
- <?php wp_list_pages(); ?>：显示 Page 列表
- <?php wp_list_categories(); ?>：显示 Categories 列表
- <?php next_post_link(' %link '); ?>：下一篇文章链接
- <?php previous_post_link('%link'); ?>：上一篇文章链接
- <?php get_calendar(); ?>：日历
- <?php wp_get_archives() ?>：显示内容存档
- <?php posts_nav_link(); ?>：导航，显示上一篇/下一篇文章链接
- <?php include(TEMPLATEPATH . '/文件名'); ?>：嵌入其他文件，可为定制的模板或其他类型文件

与模板相关的其他函数

- <?php _e('Message'); ?>：输出相应信息
- <?php wp_register(); ?>：显示注册链接
- <?php wp_loginout(); ?>：显示登录/注销链接

- <!--next page-->：将当前内容分页
- <!--more-->：将当前内容截断，以不在主页/目录页显示全部内容
- <?php timer_stop(1); ?>：网页加载时间(秒)
- <?php echo get_num_queries(); ?>：网页加载查询量

WORDPRESS 函数技巧

Loop 循环（成功）

```
<?php if(have_posts()) : ?>
<?php while(have_posts()) : the_post(); ?>
// the code inside the loop //插入 Loop 中的代码
<?php endwhile; ?>
<?php else : ?>
<?php endif; ?>
```

在 WordPress 的第一篇文章中插入 Google 广告（成功）

```
<?php while ( have_posts() ): the_post(); $count++;?>
```

```
<?php if ($count == 1) : ?>
// Insert your Google AdSense code here
<?php endif; ?>
```

```
<?php endwhile; ?>
```

你同样可以改变 count 值来让广告比如放在不同的日志后面，比如改成 count == 2，则把 Google 广告显示在第二篇日志后面：

改变每个分类页面的日志显示数。（成功）

你将需要打开 category.php 文件并且找到下面的这行,

在<?php if (have_posts()) : ?>

下面添加内容：

```
<?php if (is_category('1')) {$posts = query_posts ($query_string.'&showposts=1'); } ?>
<?php if (is_category('16')) {$posts = query_posts($query_string.'&showposts=2'); } ?>
```

“is_category()”就是分类 ID，比如可以写成“is_category('5)”，“showposts=4 里面的数值就是你要显示的分类日志数。

给特定分类添加不同模板（成功）

如果你想给 ID 为 7 的分类制作一个和你目前主题不一样的外观，只需在你当前主题文件夹中创建一个名为 category-7.php 的模板文件，然后自定义我喜欢的样式，WordPress 会自动查看并显示它，请注意文件名中的“-”符号。

为特定分类文章创建不同外观

假设你有两个分类，分别名为“News”和“Tutorials”，然后你想让“News”分类里的单篇文章的外观表层为 style1.css，让“Tutorials”分类里的单篇文章外观表层为 style2.css，Lorelle 提供了以下这种比较简单的解决方法：

打开 single.php 文件，并删除里面的所有代码，添加以下代码：

```
<?php
$post = $wp_query->post;
if ( in_category('9') ) {
include(TEMPLATEPATH . '/single2.php');
} else {
include(TEMPLATEPATH . '/single1.php');
}
?>
```

通常情况下，PHP 会自动查询这段代码，如果日志的分类 ID 为 9，则会自动显示 single2.php，如果不是则会显示为 single1.php。

注意：single1.php 和 single2.php 里的样式自定义，系统会自动判断加载不同的样式显示

样式化不同分类

如果你想给特定分类规定不同的样式表，你所需做的只是在 header.php 文件的标签中加入以下代码：

```
<?php if ( is_category('15') ) { ?>
<link rel="stylesheet" href="<?php bloginfo('template_url'); ?>/cat-15.css"
type="text/css" media="screen" />;
<?php } else { ?>
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url'); ?>" type="text/css"
media="screen" />
<?php } ?>
```

在某一个页面单独调用某一个 php 文件；

如果你仅想在首页显示某个文件内容，可在 index.php 文件中写入以下代码：

```
<div> <?php if ( is_home() ) { include (TEMPLATEPATH.'/test.php'); } ?> </div>
<div> <?php if ( is_category(1) ) { include (TEMPLATEPATH.'/test.php'); } ?> </div>
页面显示特定内容：( 取名字到 is_page('*') )
<div> <?php if( is_page('contact')) { include (TEMPLATEPATH.'/test.php'); } ?> </div> //判断是不 contact 页面
    /// is_posted()
```

动态标题以实现更友好的 SEO

替换：header.php 内 <title><?php bloginfo('name'); wp_title(); ?></title>

```
<?php
    if (is_home()) { echo bloginfo('name'); }
    elseif (is_404()) { bloginfo('name'); echo ' - Oops, this is a 404 page'; }
    else if ( is_search() ) { bloginfo('name'); echo (' - Search Results');}
    else { bloginfo('name'); echo (' - '); wp_title(""); }
?>
```

动态显示菜单：根据分类及页面类型显示相应条目：

```
<!--Begin an unordered list the ID is for the bookmark, the class is for the style
      (so I can have other lists with the same styling as seen on RedScrubs.com)-->
<ul id="nav" class="btn">
<!--IF the current page is the 404 page, note the added error class-->
<?php if(is_404()) { ?>
    <li class="current_page_item error"><a href="#">404 Error</a></li>
<?php } ?>
<!--IF the current page is the search results page-->
<?php if(is_search()) { ?>
    <li class="current_page_item"><a href="#">Search Results</a></li>
<?php } ?>

<!--IF the current page is a single post-->
<?php if(is_single()) { ?>
    <li class="current_page_item"><a href="#">Selected Article</a></li>
<?php } ?>
```

动态显示高亮菜单：

动态加载 css 样式：

```
<ul id="nav">
    <li<?php if ( is_home() || is_category() || is_archive() || is_search() || is_single() ||
is_date() ) { echo ' class="current"'; } ?>><a href="#">Gallery</a></li>
    <li<?php if ( is_page('about') ) { echo ' class="current"'; } ?>><a href="#">About</a></li>
    <li<?php if ( is_page('submit') ) { echo ' class="current"'; } ?>><a
href="#">Submit</a></li>
</ul>
```

自定义顶部导航

```
<ul id="nav">
    <li<?php if ( is_home() || is_category() || is_archive() || is_search() || is_single() ||
is_date() ) { echo ' class="current"'; } ?>><a href="<?php bloginfo('url'); ?>">首页</a></li>
    <li<?php if ( is_page('about') ) { echo ' class="current"'; } ?>> <a href="<?php echo
'/index.php/about' ; ?>">关于</a> </li>
    <li<?php if ( is_page('liuyan') ) { echo ' class="current"'; } ?>> <a href="<?php echo
'/index.php/liuyan' ; ?>">留言板</a> </li></ul>
```


查询文章

`Query_posts`(查询文章?)能被用来显示特定文章,你所需做只是你一个模板文件的 `Loop` 开始之前调用函数

只显示某个分类中的文章

如果你想显示某个特定分类的一定目录的文章

```
<?php query_posts('cat=15&showposts=10'); ?> //这里则会显示分类ID为15的最新10篇文章。
```

从首页移除特定分类

```
<?php if (is_home()) { query_posts("cat= -3"); } ?> //从主页上移除分类为3的所有日志。
```

检索某篇文章

```
<?php query_posts('p=3'); ?> 只显示 post-id 为 3 的页面。
```

`query_posts()` 函数的一些常见用法

是 page 而不是 post

```
<?php
query_posts('page_id=1'); // 只显示 id=1 的页面
query_posts('pagename=about'); // 只显示页面名字叫 about 的页面
?>
```

获取文章

`get_posts` 是一个简单的你可利用多循环以不同方式罗列文章的标签,例如你想罗列最新的5个条目,代码可如下:

```
<div>
  <h2> 最新文章</h2>
  <?php
    $previous_posts = get_posts('numberposts=5');
    foreach($previous_posts as $post) :
      setup_postdata($post);
    ?>
    <li><a href="<?php the_permalink(); ?>" id="post-<?php the_ID(); ?>"><?php
```

```
the_title(); ?></a></li>
    <?php /*?><?php the_content(); ?><?php */?>
    <?php endforeach; ?>
</div>
```

上面的代码中变量"numberposts= ?"则告诉 WordPress 在循环里索引和填充多少文章，WordPress 文章格式函数 setup_postdata()将会自动填充所需函数！

如何按分类和按月显示一个 Archvies 存档页面

```
<?php while(have_posts()) : the_post(); ?>
<ul><?php wp_get_archives('type=monthly&show_post_count=1') ?></ul>
<ul><?php wp_list_cats('sort_column=name&optioncount=1') ?></ul>
```

```
<?php endwhile; ?>
```

使用 wp_list_cats 标签显示一个按分类排列的 WordPress 存档列表和 wp_get_archives 显示基于日期的存档列表。

页面模板头部声明

```
<?php/*
Template Name: Gallery
*/
?>
```

为不同的分类指定不同的图像

```
<?php if (is_category('1') ): ?>
<img src='<?php bloginfo('template_url'); ?>/01.bmp' alt='' />
<?php elseif (is_category('16') ): ?>
<img src='<?php bloginfo('template_url'); ?>/16.bmp' alt='' />
<?php endif; ?>
```

is_paged()：主页 /Category/Archive页是否以多页显示：这里是指是否是分页的第 2, 3, 4, , , , 页。

! is_paged() 是指分页的第一页

如何添加幻灯片 banner 广告

在开始之前我们需要先下载 SmoothGallery 2.0

1) 将所需文件放到合适的地方

解压下载到的 smoothgallery

将解压得到的 css 文件夹复制到 wordpress 目录 wp-content/themes/your_theme_name

将 scripts 文件夹复制到 wp-content/themes/your_theme_name

2) 在 header.php 里添加：

```
<script type="text/javascript" src="<?php
bloginfo('template_url'); ?>/huandeng/scripts/mootools.v1.11.js"></script>
<!--<!--JS SmoothGallery-->-->
<script type="text/javascript" src="<?php
bloginfo('template_url'); ?>/huandeng/scripts/jd.gallery.js"></script>
<?php /*?><?php wp_head();?><?php */?>去掉此行，不然不能实现。不知道什么原因？
wp_head();里面是其他插件里面的样式或者是其他调用代码，如果去掉此行的话将无法显示了，所以只能把他们所需要的代码直接写进主题的样式模板内进行统一调用。可以解决问题。
```

3) 新建一个文件 huandeng.php

添加代码：

```
<!-- Initialization of SmoothGallery-->
<script type="text/javascript">
    function startGallery() {
        var myGallery = new gallery($('myGallery'), {
            timed: false
        });
    }
    window.addEvent('domready',startGallery);
</script>
<!-- Creation of the html for the gallery -->
<div class="content">
    <div id="myGallery">
        <div class="imageElement">
            <h3>Item 1 Title</h3>
            <p>Item 1 Description</p>
            <a href="#" title="open image" class="open"></a>
            <img
                                src='
                                <?php
bloginfo('template_url'); ?>/huandeng/images/brugges2006/1.jpg' class="full" />
                                <img
                                src='<?php
bloginfo('template_url'); ?>/huandeng/images/brugges2006/1-mini.jpg' class="thumbnail" />
```

```
</div>
<div class="imageElement">
  <h3>Item 2 Title</h3>
  <p>Item 2 Description</p>
  <a href="#" title="open image" class="open"></a>
  
  
</div>
<div class="imageElement">
  <h3>Item 3 Title</h3>
  <p>Item 3 Description</p>
  <a href="#" title="open image" class="open"></a>
  
  
</div>
<div class="imageElement">
  <h3>Item 4 Title</h3>
  <p>Item 4 Description</p>
  <a href="#" title="open image" class="open"></a>
  
  
</div>
<div class="imageElement">
  <h3>Item 5 Title</h3>
  <p>Item 5 Description</p>
  <a href="#" title="open image" class="open"></a>
  
  
</div>
<div class="imageElement">
  <h3>Item 6 Title</h3>
  <p>Item 6 Description</p>
  <a href="#" title="open image" class="open"></a>
  
  
</div>
```

```

        <div class="imageElement">
            <h3>Item 7 Title</h3>
            <p>Item 7 Description</p>
            <a href="#" title="open image" class="open"></a>
            
            
        </div>
        <div class="imageElement">
            <h3>Item 8 Title</h3>
            <p>Item 8 Description</p>
            <a href="#" title="open image" class="open"></a>
            <img rc="<?php
bloginfo('template_url'); ?>/huandeng/images/brugges2006/8.jpg" class="full" />
            
        </div>
    </div>
</div>

```

- 4) 在需要引用的地方添加以下代码：(不知道什么原因调用 **huandeng.php** 代码后，此区域将脱离文件流，而且没有办法让它归位，只能将其直接写入 **index.php** 文件内部)

```
<?php include(TEMPLATEPATH.'/huandeng.php'); ?>
```

这里面有一个

5) 自定义 gallery 的具体显示效果。

打开文件 wp-content/themes/your_theme_name/css/jd.gallery.css，在这里修改 gallery 的宽和高。(通过修改 jd.gallery.css 完全对这个 slideshow 根据自己的主题进行个性化。^_^)

```

#myGallery, #myGallerySet, #flickrGallery
{
width: 590px;
height: 250px;
z-index:5;
border: 1px solid #000;
}

```

默认的字号对于中文太小了，可以调整 slideshow 下方信息栏的高度及文字的字号，只需要修改

.jdGallery .slideInfoZone (滑动信息栏高度、颜色等参数)

.jdGallery .slideInfoZone h2 (信息栏内标题样式)

.jdGallery .slideInfoZone p (信息栏文本样式)

*上滑动栏宽度;

```
.jdGallery .carousel
{
    position: absolute;
    width: 100%;
    margin: 0px;
    left: 0;
    top: 0;
    height: 90px; /*上滑动栏宽度;*/
    background: #333;
    color: #fff;
    text-indent: 0;
    overflow: hidden;
}
```

你还可以修改 wp-content/themes/your_theme_name/scripts/jd.gallery.js 来改变 gallery 的展示效果 (Smooth Gallery 提供了多种不同的显示效果, 你可以根据需要进行修改)

5 个你不知道的 Wordpress 函数

1. **wp_mail()**基本上是一个超级简单的函数, 允许你通过一些简单的代码轻松地向任何人发送电子邮件。例如:

```
<?php
$to = 'user@example.com';
$subject = 'Hello from my blog!';
$message = 'Check it out -- my blog is emailing you!'

$mail = wp_mail($to, $subject, $message);

if($mail) echo 'Your message has been sent!';
else echo 'There was a problem sending your message. Please try again.';
?>
```

2. **wp_logout()** 主题中显示“登录”链接
3. **clean_url()**: 入 URL并测试它的结构是否正确。如果链接前面缺少 http:// 它可以自动添加, 转换符号为正确的 HTML,
4. **wpautop()**: 这个函数用来转换换行符字符串为
标记, 并把双换行符作为一个新的段落的开始, 在前一段加入 </p>, 并在新的段落加上 <p>

wordpress 一些常用代码：

显示最新文章

```
| <div id="newpost">
| <h2> 最新文章</h2>
| <?php
| $previous_posts = get_posts('numberposts=15');
| foreach($previous_posts as $post) :
| setup_postdata($post); ?>
| <li><a href="<?php the_permalink(); ?>" id="post-<?php the_ID(); ?>"><?php
| the_title(); ?></a></li>
| <?php /*?><?php the_content(); ?><?php */?>
| <?php endforeach; ?>
| </div>
| <div id="newcomment">
```

显示最新评论

```
| <h2>最新评论</h2>
| <div id="newcom">
| <?php
| global $wpdb;
| $sql = "SELECT DISTINCT ID, post_title, post_password, comment_ID,
| comment_post_ID, comment_author, comment_date_gmt, comment_approved,
| comment_type,comment_author_url,
| SUBSTRING(comment_content,1,30) AS com_excerpt
| FROM $wpdb->comments
| LEFT OUTER JOIN $wpdb->posts ON ($wpdb->comments.comment_post_ID =
| $wpdb->posts.ID)
| WHERE comment_approved = '1' AND comment_type = '' AND
| post_password = ''
| ORDER BY comment_date_gmt DESC
| LIMIT 10";
| $comments = $wpdb->get_results($sql);
| $output = $pre_HTML;
| $output .= "\n<ul>";
| foreach ($comments as $comment) { $output .=
| "\n<li>".strip_tags($comment->comment_author) . ":" . "<a href=\"" .
| get_permalink($comment->ID) . "#comment-" . $comment->comment_ID . "\"
| title=\"" . $comment->post_title .
| "\">" . strip_tags($comment->com_excerpt) . "</a></li>"; }
| $output .= "\n</ul>";
```

```
| $output .= $post_HTML;  
| echo $output;?>  
| </div>  
| </div>
```

显示热评文章

```
| <div id="reping">  
| <h2> 最受关注文章</h2>  
| <?php $result = $wpdb->get_results("SELECT comment_count,ID,post_title FROM  
| $wpdb->posts ORDER BY comment_count DESC LIMIT 0 , 10");  
| foreach ($result as $stopten) {  
| $postid = $stopten->ID;  
| $title = $stopten->post_title;  
| $commentcount = $stopten->comment_count;  
| if ($commentcount != 0) { ?>  
| <li><a href="<?php echo get_permalink($postid); ?>" title="<?php echo  
| $title ?>"><?php echo $title ?></a></li>  
| <?php } } ?>  
| </div>
```

调用 Gravatar (只适应 2.5以上)

```
<?php if(function_exists('get_avatar')){ echo get_avatar($comment, '50');} ?>
```

调用前两篇日志，后续 3篇日志列表

```
| <?php query_posts('showposts=2'); ?><?php if (have_posts()) : while (have_posts()) :  
| the_post(); ?>  
| <h2><?php the_title(); ?></h2>  
| Posted on <?php the_time('F jS, Y') ?> in <?php the_category(', '); ?>  
| <?php endwhile; endif; ?><!-- google ad script -->  
| (Google ads here)  
| <?php query_posts('showposts=3&offset=2'); ?>  
| <?php if (have_posts()) : while (have_posts()) : the_post(); ?>  
| <h2><?php the_title(); ?></h2>  
| Posted on <?php the_time('F jS, Y') ?> in <?php the_category(', '); ?>  
| <?php endwhile; endif; ?>
```

随机文章

```
| <div id="rand">  
| <h2>随机文章</h2>  
| <?php  
| $rand_posts = get_posts('numberposts=10&orderby=rand');
```



```
|   foreach( $rand_posts as $post ) : ?>
|   <!--下面是你想自定义的 Loop-->
|   <li><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></li>
|   <?php endforeach; ?></div>
```

安全的调用 Wordpress插件模板代码

```
<?php if ( function_exists('plugin_template_tag_function') ) : ?>
<h2>部件标题</h2>
<?php plugin_template_tag_function(); ?>
<?php endif; ?>
```

Wordpress 常用代码解释

1. 最新文章

Wordpress最新文章的调用可以使用一行很简单的模板标签 `wp_get_archives`来实现。代码如下：

```
<?php get_archives('postbypost', 10); ?> (显示 10篇最新更新文章 )
```

或

```
<?php wp_get_archives('type=postbypost&limit=20&format=custom'); ?>
```

后面这个代码显示你博客中最新的 20篇文章，其中 `format=custom`这里主要用来自定义这份文章列表的显示样式。具体的参数和使用方法你可以参考官方的使用说明 - [wp_get_archives](#)。（`format=custom`也可以不要，默认以 UL列表显示文章标题。）

补充：通过 WP的 [query_posts\(\)](#)函数也能调用最新文章列表，虽然代码会比较多一点，但可以更好的控制 Loop的显示，比如你可以设置是否显示摘要。具体的使用方法也可以查看官方的说明。

2. 随机文章

```
<?php
$rand_posts = get_posts('numberposts=10&orderby=rand');
foreach( $rand_posts as $post ) :
?>
<!--下面是你想自定义的 Loop-->
<li><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></li>
```

```
<?php endforeach; ?>
```

补充：上面提到的 `query_posts` 同样可以生成随机文章列表。

3. 最新留言

下面是我之前在一个 Wordpress 主题中代到的最新留言代码，具体也记不得是哪个主题了。该代码直接调用数据库显示一份最新留言。其中 `LIMIT 10` 限制留言显示数量。绿色部份则是每条留言的输出样式。

```
<?php
global $wpdb;
$sql = "SELECT DISTINCT ID, post_title, post_password, comment_ID,
comment_post_ID, comment_author, comment_date_gmt, comment_approved,
comment_type,comment_author_url,
SUBSTRING(comment_content,1,30) AS com_excerpt
FROM $wpdb->comments
LEFT OUTER JOIN $wpdb->posts ON ($wpdb->comments.comment_post_ID =
$wpdb->posts.ID)
WHERE comment_approved = '1' AND comment_type = '' AND
post_password = ''
ORDER BY comment_date_gmt DESC
LIMIT 10";
$comments = $wpdb->get_results($sql);
$output = $pre_HTML;

foreach ($comments as $comment) {
$output .= "\n<li>".strip_tags($comment->comment_author)
.":" . " <a href=\"" . get_permalink($comment->ID) .
"#comment-" . $comment->comment_ID . "\" title=\"" .
$comment->post_title . "\">" . strip_tags($comment->com_excerpt)
."</a></li>";
}

$output .= $post_HTML;
echo $output;?>
```

调用 TAB 页签菜单：

添加 head 头文件：判断后调用 js文件

```
<?php if ( is_home() ){ ?>
```

```
<script type="text/javascript" src="<?php  
bloginfo('template_url'); ?>/tab/easytabs.js"></script>
```

```
<?php } else { ?>
```

```
<script type="text/javascript" src="<?php  
bloginfo('template_url'); ?>/tab/easytabs1.js"></script>
```

```
<?php } ?>
```

添加 css 代码到 样式文件内

```
| <!--tab 页签 css-->  
| #tab{  
| text-align:left;  
| position:relative;  
| }  
| .menu {  
| background-color:#ececce;  
| color:#272727;  
| border-bottom:1px solid #d7d7d7;  
| height:23px;  
| width:450px;  
| vertical-align:middle;  
| }  
| .menu ul {  
| margin:0px;  
| padding:0px;  
| list-style:none;  
| text-align:left;  
| }  
| .menu li {display:inline;  
| line-height:23px;}  
|  
| .menu li a {color:#000000;  
| text-decoration:none;  
| padding:4px 5px 6px 5px;
```

```
|         border-left:1px solid #ececec;
|         border-right:1px solid #ececec;
|     }
|     .menu li a.tabactive {border-left:1px solid #d7d7d7;
|                             border-right:1px solid #d7d7d7;
|                             background-color:#CCCCCC;
|                             font-weight:bold;
|                             position:relative;}
|
|     #tabcontent1,#tabcontent2,#tabcontent3,#tabcontent4 {
|     border:1px solid #ececec;
|     width:450px;
|     text-align:left;
|     padding:6px 0px;
|     font-size:12px;
|     margin:2px 0 5px 5px;
|     }
|
|
|     #container1 #newcomment,#container #newcomment{
|
|         position: relative;
|         margin-left:30px;
|         width:350px;
|     }
|
|
|     #container1 #reping,#tabcontent1 #reping{
|         position: relative;
|         margin-left:30px;
|         width:350px;
|     }
|
|     #container1 #rand li, #tabcontent2 #rand li{
|         margin-left:30px;
|         padding:0;
|     }
|
|     <!--tab 页签 css-->
```

添加 tab代码到需要调用的地方

```
<div id="tab"><h2>最新关注日志及评论</h2>
```

```
<!--Start of the Tabmenu 1 -->
```

```

<div class="menu">

<ul>

<li><a href="#" onmouseover="easytabs('1', '1');" onfocus="easytabs('1', '1');"
onclick="return false;" title="" id="tablink1">最新评论</a></li>

<li><a href="#" onmouseover="easytabs('1', '2');" onfocus="easytabs('1', '2');"
onclick="return false;" title="" id="tablink2">最受关注文章 </a></li>

<li><a href="#" onmouseover="easytabs('1', '3');" onfocus="easytabs('1', '3');"
onclick="return false;" title="" id="tablink3">随机文章 </a></li>

</ul>

</div>

<!--Start Tabcontent 1 -->

<div id="tabcontent1">

<div id="newcomment">

<?php

global $wpdb;

$sql = "SELECT DISTINCT ID, post_title, post_password, comment_ID,
comment_post_ID, comment_author, comment_date_gmt, comment_approved,
comment_type,comment_author_url,SUBSTRING(comment_content,1,30) AS
com_excerpt FROM $wpdb->comments LEFT OUTER JOIN $wpdb->posts ON
($wpdb->comments.comment_post_ID = $wpdb->posts.ID) WHERE comment_approved =
'1' AND comment_type = '' AND post_password = '' ORDER BY comment_date_gmt DESC
LIMIT 10"; $comments = $wpdb->get_results($sql); $output = $pre_HTML; foreach
($comments as $comment) { $output .=
"\n<li>".strip_tags($comment->comment_author).": " . "<a href=\"" .
get_permalink($comment->ID). "#comment-" . $comment->comment_ID . "\"
title=\"on " . $comment->post_title . "\">" .
strip_tags($comment->com_excerpt)."</a></li>"; } $output .= $post_HTML;

echo $output;?> </div>

</div>

```

```
<!--Start Tabcontent 2-->

<div id="tabcontent2">

<div id="reping">

<?php $result = $wpdb->get_results("SELECT comment_count,ID,post_title FROM
$wpdb->posts ORDER BY comment_count DESC LIMIT 0 , 10");

foreach ($result as $stopten) {

$postid = $stopten->ID;

$title = $stopten->post_title; $commentcount = $stopten->comment_count;

if ($commentcount != 0) { ?>

<li><a href="<?php echo get_permalink($postid); ?>" title="<?php echo
$title ?>"><?php echo $title ?></a></li>
    <?php } } ?>

</div>

</div>

<!--End Tabcontent 2 -->

<!--Start Tabcontent 3-->

<div id="tabcontent3">

<div id="rand">

<?php

$rand_posts = get_posts('numberposts=10&orderby=rand');

foreach( $rand_posts as $post ) : ?>

<!--下面是你想自定义的 Loop--> <li><a href="<?php
the_permalink(); ?>"><?php the_title(); ?></a></li>
    <?php endforeach; ?>

</div>

</div><br/><br/>
```

</div>

保存 js代码到目录下

```
/*

EASY TABS 1.2 Produced and Copyright by Koller Juergen

www.kollermedia.at | www.austria-media.at

Need Help? http://www.kollermedia.at/archive/2007/07/10/easy-tabs-12-now-with-autochange

You can use this Script for private and commercial Projects, but just leave the two credit lines, thank
you.*/

//EASY TABS 1.2 - MENU SETTINGS

//Set the id names of your tablink (without a number at the end)设置链接的 tab 页签名称

var tablink_idname = new Array("tablink")

//Set the id name of your tabcontentarea (without a number at the end)设置 tab 内容区域 id 名称

var tabcontent_idname = new Array("tabcontent")

//Set the number of your tabs//设置 tab 的个数

var tabcount = new Array("2")

//Set the Tab wich should load at start (In this Example:Tab 2 visible on load)

var loadtabs = new Array("1")

//Set the Number of the Menu which should autochange (if you dont't want to have a change menu
set it to 0)

var autochangemenu = 0;

//the speed in seconds when the tabs should change

var changespeed = 3;

//should the autochange stop if the user hover over a tab from the autochangemenu? 0=no 1=yes

var stoponhover = 0;

//END MENU SETTINGS
```

```
/*Swich EasyTabs Functions - no need to edit something here*/
```

```
function easytabs(menunr, active) {if (menunr == autochangemenu){currenttab=active;}if ((menunr == autochangemenu)&&(stoponhover==1)) {stop_autochange()} else if ((menunr == autochangemenu)&&(stoponhover==0)) {counter=0;}menunr = menunr-1;for (i=1; i <= tabcount[menunr]; i++){document.getElementById(tablink_idname[menunr]+i).className='tab'+i;document.getElementById(tabcontent_idname[menunr]+i).style.display = 'none';}document.getElementById(tablink_idname[menunr]+active).className='tab'+active+'tabactive';document.getElementById(tabcontent_idname[menunr]+active).style.display = 'block';}var timer; counter=0; var totaltabs=tabcount[autochangemenu-1];var currenttab=loadtabs[autochangemenu-1];function start_autochange(){counter=counter+1;timer=setTimeout("start_autochange()",1000);if (counter == changespeed+1) {currenttab++;if (currenttab>totaltabs) {currenttab=1}easytabs(autochangemenu,currenttab);restart_autochange();}}function restart_autochange(){clearTimeout(timer);counter=0;start_autochange();}function stop_autochange(){clearTimeout(timer);counter=0;}

window.onload=function(){

var menucount=loadtabs.length; var a = 0; var b = 1; do {easytabs(b, loadtabs[a]); a++; b++;}while (b<=menucount);

if (autochangemenu!=0){start_autochange();} }
```

所有页面的归档：

```
function BX_archive(){

    global $month, $wpdb;

    $now = current_time('mysql');

    $sarcresults = $wpdb->get_results("SELECT DISTINCT YEAR(post_date) AS year, MONTH(post_date) AS month, count(ID) as posts FROM " . $wpdb->posts . " WHERE post_date < " . $now . " AND post_status='publish' AND post_type='post' AND post_password='' GROUP BY YEAR(post_date), MONTH(post_date) ORDER BY post_date DESC");

    if ($sarcresults) {

        foreach ($sarcresults as $sarcresult) {

            $url = get_month_link($sarcresult->year, $sarcresult->month);
```



```
$text = sprintf('%s %d', $month[zeroise($sarcresult->month,2)], $sarcresult->year);

echo get_archives_link($url, $text, "<h3>','</h3>');

$thismonth = zeroise($sarcresult->month,2);

$thisyear = $sarcresult->year;

$sarcresults2 = $wpdb->get_results("SELECT ID, post_date, post_title, comment_status
FROM " . $wpdb->posts . " WHERE post_date LIKE '$thisyear-$thismonth-%' AND post_status='publish'
AND post_type='post' AND post_password='' ORDER BY post_date DESC");

if ($sarcresults2) {

    echo "<ul class='postspermonth'>\n";

    foreach ($sarcresults2 as $sarcresult2) {

        if ($sarcresult2->post_date != '0000-00-00 00:00:00') {

            $url = get_permalink($sarcresult2->ID);

            $sarc_title = $sarcresult2->post_title;

            if ($sarc_title) $text = strip_tags($sarc_title);

            else $text = $sarcresult2->ID;

            echo "<li>".get_archives_link($url, $text, "");

            $comments = mysql_query("SELECT * FROM " . $wpdb->comments . "
WHERE comment_approved='1' and comment_post_ID=" . $sarcresult2->ID);

            $comments_count = mysql_num_rows($comments);

            if ($sarcresult2->comment_status == "open" OR $comments_count > 0)
echo '&nbsp;('.$comments_count.')';

            echo "</li>\n";

        }

    }

    echo "</ul>\n";

} } }
```

Wordpress 主题中常用代码总结：

1. 在 Wordpress主题中显示最热文章的 PHP代码

```
<?php $result = $wpdb->get_results("SELECT comment_count,ID,post_title FROM $wpdb->posts ORDER BY comment_count DESC LIMIT 0 , 10");
foreach ($result as $post) {
    setup_postdata($post);
    $postid = $post->ID;
    $title = $post->post_title;
    $commentcount = $post->comment_count;
    if ($commentcount != 0) { ?>
    <li><a href="<?php echo get_permalink($postid); ?>" title="<?php echo $title ?>">
    <?php echo $title ?></a> (<?php echo $commentcount ?>)</li>
    <?php } } ?>
```

2. wordpress主题 – 相关文章代码

```
<div id="newpost">
    <ul>
        <?php
            $tags = wp_get_post_tags($post->ID);
            if ($tags) {
                $first_tag = $tags[0]->term_id;
                $args=array(
                    'tag__in' => array($first_tag),
                    'post__not_in' => array($post->ID),
                    'showposts'=>10,
                    'caller_get_posts'=>1
                );
                $my_query = new WP_Query($args);
                if( $my_query->have_posts() ) {
                    while ($my_query->have_posts()) : $my_query->the_post(); ?>
                        <li><a href="<?php the_permalink() ?>" rel="bookmark"
                        title="Permanent Link to
                        <?php the_title_attribute(); ?>"><?php the_title(); ?>
                        </a> </li>
                        <?php endwhile; } else {echo 'not realate post';} }else {echo 'not
                        realate post';} ?>
```

3. Wordpress主题使用 PHP代码输出最新文章

```
<?php
$limit = get_option('posts_per_page');
$paged = (get_query_var('paged')) ? get_query_var('paged') : 1;
query_posts('showposts=' . $limit . '&paged=' . $paged);
$wp_query->is_archive = true; $wp_query->is_home = false;
?>
<?php while(have_posts()) : the_post(); if(!($first_post == $post->ID)) : ?>
<ul>
<li><a href="<?php the_permalink() ?>" rel="bookmark" title="Permanent Link to
<?php the_title_attribute(); ?>"><?php the_title(); ?></a></li>
</ul>
<?php endif; endwhile; ?>
```

4. 最新评论：

```
<?php
global $wpdb;
$sql= "SELECT DISTINCT ID, post_title, post_password,
comment_ID,comment_post_ID, comment_author, comment_date_gmt,
comment_approved,comment_type,comment_author_url,
SUBSTRING(comment_content,1,30) AS com_excerpt FROM $wpdb->comments LEFT OUTER
JOIN $wpdb->posts ON($wpdb->comments.comment_post_ID = $wpdb->posts.ID)
WHERE comment_approved = '1' AND comment_type = '' AND post_password = '' ORDERBY
comment_date_gmt DESC LIMIT 10";

$comments = $wpdb->get_results($sql);
$output = $pre_HTML;

foreach ($comments as $comment) {
$output.= "\n<li>". "<a href=\"". get_permalink($comment->ID). "#comment-" .
$comment->comment_ID . "\" title=\"on
\".$comment->post_title . "\">". strip_tags($comment->comment_author). "</a>" .
": ". strip_tags($comment->com_excerpt). "</li>";
}

$output .= $post_HTML;
echo $output;
?>
```

5. wordpress主题 – 相关文章代码

基本 sql 是这样：\$sql = "SELECT p.ID, p.post_title, p.post_date, p.comment_count, count(t_r.object_id) as cnt FROM \$wpdb->term_taxonomy t_t, \$wpdb->term_relationships t_r, \$wpdb->posts p WHERE t_t.taxonomy = 'post_tag' AND t_t.term_taxonomy_id = t_r.term_taxonomy_id AND t_r.object_id = p.ID AND (t_t.term_id IN (\$taglist)) AND p.ID != \$post->ID AND p.post_status = 'publish' GROUP BY t_r.object_id ORDER BY cnt DESC, p.post_date_gmt DESC LIMIT \$limit;";

一点一点的解释：term_taxonomy、term_relationship、posts 这三张表存的什么我不多说，网上一般都可以查到，维基百科貌似都有。把他们连起来，做个 sql，注意 group by 以及 limit，这样就可以提取结果了\$related_post = \$wpdb->get_results(\$sql);

之前要把\$taglist 做好，利用 wp_get_post_tags(\$post->ID);可以将该篇文章的 tag 取到一个数组中，然后再链接就可以了

最后怎么处理输出就看个人爱好了，这个功能我写的大概也就十几行，我比较喜欢简短的说，呵呵。

```
function related_post($limit = 10) {
    global $wpdb, $post;
    $tags = wp_get_post_tags($post->ID);
    $taglist = "" . $tags[0]->term_id. "";
    $tagcount = count($tags);
    if ($tagcount > 1) {
        for ($i = 1; $i < $tagcount; $i++) {
            $taglist .= ", " . $tags[$i]->term_id. "";
        }
    }
    $sql = "SELECT p.ID, p.post_title, p.post_date, p.comment_count, count(t_r.object_id) as cnt
    FROM $wpdb->term_taxonomy t_t, $wpdb->term_relationships t_r, $wpdb->posts p WHERE
    t_t.taxonomy = 'post_tag' AND t_t.term_taxonomy_id = t_r.term_taxonomy_id AND t_r.object_id =
    p.ID AND (t_t.term_id IN ($taglist)) AND p.ID != $post->ID AND p.post_status = 'publish' GROUP
    BY t_r.object_id ORDER BY cnt DESC, p.post_date_gmt DESC LIMIT $limit;";
    $related_post = $wpdb->get_results($sql);
    $output = "";
    foreach ($related_post as $tmp){
        $output .= 这个就看个人爱好了
    }
    if($output == "")$output = "No Related Post Yet";
    echo $output;
}
```

WordPress 简单企业模板,企业网站完成

先在 WordPress 生成一分类，叫[基本资料]，然后分别写[公司简介]和[联系方式]等，再增加分类[新闻中心]。

首页模板里调用 `query_posts('p=1')`，完成公司简介，1 是公司简介这篇文章 ID

`index.php` 代码:

```
<?php
get_header();
?>
<div id="page" class="clearfloat">

<div class="clearfloat">
<div id="header">
<div class="mainheader"><a href="<?php bloginfo('url'); ?>"><?php
bloginfo('name'); ?> </a></div>
<div class="description"></div>

</div>

<ul id="nav" class="clearfloat">
<li><a href="<?php bloginfo('url'); ?>">HOME</a></li>
</ul>

<div id="content">

<br />



<?php global $firstx; ?>
<?php $firstx=0;?>
<?php if ( is_home()) { ?>
<?php $firstx=1;?>
<?php query_posts('p=1'); ?>
<?php } ?>
<?php if(have_posts()) : ?>
<?php while(have_posts()) : the_post(); ?>
<div class="post" id="post-<?php the_ID(); ?>">
<div class=singlepage><a href="<?php the_permalink() ?>"
rel="bookmark"><?php the_title(); ?></a></div>
```

```
<div class="entry">
<?php the_content(); ?>
<?php
// get_the_title()
// get_the_content()
// $ss=get_the_content();
// echo "<p> *****<p>";
// echo $ss;
// echo "<p> *****<p>";
?>

<!--yes kao-->
</div>
</div>

<?php endwhile; else: ?>
<p><?php _e('No Data'); ?></p>
<?php endif; ?>

<p>

</div>

<?php get_sidebar(); ?>
</div>

<?php get_footer(); ?>
```

在 sidebar.php 写死了部分导航 ,其实可以用其它方式的 ,以下是 sidebar.php 的代码:

```
<div id="sidebar">
<div class="sideblock">
<h3>导航</h3>

<ul>
<li><a href='<?php bloginfo('url'); ?>'>首页</a></li>
<?php wp_list_cats ('include=3'); ?>
</ul>
<br />
</div>

<div class="sideblock">
```

```
<h3>新闻中心</h3>
<ul>
<?php query_posts('cat=3&showposts=5'); ?>
<?php if(have_posts()) : ?>
<?php while(have_posts()) : the_post(); ?>
<li><a href="<?php the_permalink() ?>"><?php the_title(); ?></a></li>
<?php endwhile; else: ?>
<li><?php _e('No Data'); ?></li>
<?php endif; ?>

</ul>
<br />
</div>

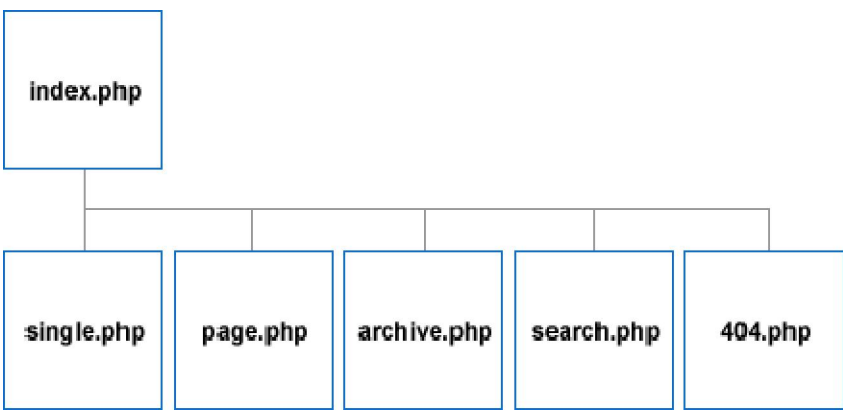
<?php global $firstx; ?>
<?php if ($firstx==1){ ?>
<div class="sideblock">

<h3>联系我们</h3>
<ul>
<?php query_posts('p=12'); ?>
<?php if(have_posts()) : ?>
<?php while(have_posts()) : the_post(); ?>
<?php the_content(); ?>
<?php endwhile; else: ?>
<li><?php _e('No Data'); ?></li>
<?php endif; ?>

</ul>
<br />
</div>
<?php } ?>
</div>
```

WordPress 搭建企业网站

在看该文档之前需要对 WordPress 主题模板的层次架构有一定的了解。简单介绍如图：



更详细的 WordPress 主题模板的结构层次请见：[wordpress 主题结构层次图](#)

一 . WordPress 搭建中小企业网站思路

WordPress 是基于文章的博客程序，而企业或产品网站也是基于一篇篇的文章，所以用 WordPress 搭建企业网站是可行的。用 WordPress 建中小企业站模板的思路如下：

1) 大分类+子分类：

首先，根据需求明确企业网站的整体结构，需要几大类的内容，如分为 1 产品介绍，2 技术中心，3 关于我们，4 试用购买，这些内容可以放在网站的主菜单上。

然后，细化企业网站结构，既明确子分类，如：



由上图可以清晰的了解网站的结构，上图中 **16** 表示该文章分类的 id 号，因为 WordPress 本身有一个未分类和链接占用分类的 1、2 两个 id 号，我们只能从 3 开始，图中 id 为 3、4、5、6 的几个分类为大分类；而 7、8、9、10 为 3 的子分类，其他父子分类关系也如此。子分类可作为网站内页的 side 里的内容。值得一提的是，**post** 表示文章，“关于公司”、“联系我们”、“合作伙伴”三个文章是直接属于大分类 5，这个网站的“关于我们”内页 side 里既有分类又有文章，如何解决？下边会有介绍。现在整体的大思路应该很明确了就是 大分类+子分类。

2) 让每个大分类内页的 side 里显示子分类：

如图：



这是我们要的效果，因为我们每个大分类下面都要显示各自的子分类，所以在 sidebar.php 里的分类列表只写 `<?php wp_list_cats('child_of=5'); ?>` 已经不能满足我们的需求。因此需要写一个 if 判断语句：

这个 if 语句帮我们解决了在每个大分类下面调用其各自子分类。

```
<?php $post = $wp_query->post;

if ( in_category(' 3 ')|in_category('7')|in_category(' 8 ') |in_category(' 9 ')|in_category(' 10 '))
    <!-- 如果文章属于大分类 3 及其所有子分类 7、8、9、10 时-->

    { wp_list_cats('child_of= 3 ');}    <!-- 则显示分类 3 的子分类-->

elseif ( in_category(' 4 ') |in_category(' 11 ')|in_category(' 12 ') |in_category(' 13 '))|in_category(' 14 '))

    { wp_list_cats('child_of= 4 ');}

elseif ( in_category(' 5 ')| in_category('1 5 ')|in_category('1 6 '))

    { wp_list_cats('child_of= 5 ');}
```

```
elseif ( in_category(' 6 '))
```

```
{ wp_list_cats('child_of= 6 ');}
```

?

但上面我们所提到的“关于我们”的 side 里是“文章+子分类”的列表，所以以上代码也不能完全满足，需要在 `{ wp_list_cats('child_of= 5 ');}` 里加入已发文章的静态链接。如下：

 $\{$

?

- [关于公司](http://?php bloginfo('url'); ?>/about/company/ "关于公司")
- [联系我们](<?php bloginfo('url'); ?>/about/contact/ "联系我们")
- [合作伙伴](http://?php bloginfo('url'); ?>/about/partner/ "合作伙伴")

<?

```
wp_list_cats('child_of= 5');
```

}

现在 side 里“文章+子分类”的列表，我们很好的解决了。由此启发，当我们在点击各大分类下 side 列表不一定都用分类，因为如果是分类的话，右边调用的是 `category.php`，显示分类页。如果每个大分类下都是这样显示分类列表的话，看起来还是很像 blog，不像企业网站。所以我们的网站构架要改一下了，如图：



正如图中所示有两种方法实现：一是写一篇该子分类概述的文章，文章里有其他该子分类的链接，把概述文章的链接用刚才所述的方法写在 sidebar.php 里；另外一种是用一个子分类概述 page 页面把链接写在 sidebar 里。这里我们采取第一种方法。因为用 page 的话会导致页面过多不好管理，而且 page 的 url 地址为：`http://域名/页面缩略名/` 的结构，这样就不好识别是哪个分类的内容了。改完的代码如下：

<u| >

```
<?php $post = $wp_query->post;

if (in_category(' 3 ')|in_category(' 7 ')|in_category(' 8 ')|in_category(' 9 ')|in_category(' 10 '))

{

    ?>

    <li><a href="<?php bloginfo('url'); ?>/products/products-all/production/" title="产品概括">产品概括</a></li>

    <li><a href="<?php bloginfo('url'); ?>/products/features/functions-and-features/" title="功能特点">功能特点</a></li>

    <li><a href="<?php bloginfo('url'); ?>/products/strength/strength/" title="核心优势">核心优势</a></li>

    <li><a href="<?php bloginfo('url'); ?>/products/application/report-appdeployment/" title="部署与运用开发">部署与运用开发</a></li>

    <?

}

elseif (in_category(' 4 ')|in_category(' 1 1 ')|in_category(' 1 2 ')|in_category(' 13 ')|in_category(' 14 '))

{ wp_list_cats('child_of=4');} /* 该分类是“技术中心”，需要它为子分类，就直接调用其子分类 */

?>

elseif ( in_category(' 5 ')|in_category(' 15 ')|in_category(' 1 6 '))

{

    ?>

    <li><a href="<?php bloginfo('url'); ?>/about/company/" title="关于公司">关于公司</a></li>

    <li><a href="<?php bloginfo('url'); ?>/about/contact/" title="联系我们">联系我们</a></li>
```

```
<li><a href="<?php bloginfo('url'); ?>/about/partner/" title="合作伙伴">合作伙伴
</a></li>
```

```
<li><a href="<?php bloginfo('url'); ?> /about /news/" title="企业新闻">企业新闻
</a></li>
```

```
<li><a href="<?php bloginfo('url'); ?> /about /cases/" title="成功案例">成功案例
</a></li>
```

```
<?
```

```
}
```

```
elseif ( in_category('6'))
```

```
{
```

```
?>
```

```
<li><a href="<?php bloginfo('url'); ?>/about/download/" title="下载试用">下载试用
</a></li>
```

```
<li><a href="<?php bloginfo('url'); ?>/about/purchase/" title="购买产品">购买产品
</a></li>
```

```
<?
```

```
}
```

```
</ul>
```

至此，sidebar 里的子分类（文章）列表已经改完。

接着，需要在子分类上标上该大分类的名称，这一点我仍是采用了 if 语句将其写死。

```
<h3>
```

```
<?php
```

```
if ( in_category('3')|in_category('7')|in_category('8') |in_category('9')|in_category('10') ) { ?>产品介绍<?php }
```

```
elseif ( (in_category('4')|in_category('11')|in_category('12')|in_category('13')|in_category('14') ) ) { ?>技术中心<?php }
```

```
elseif ( in_category('5')|in_category('15')|in_category('16') ) { ?>关于我们<?php }
```

```
elseif ( in_category('6')) { ?>试用购买<?php }  
  
?>  
  
</h3>
```

至此，sidebar 里的 title 名称也写完了。sidebar 里的子分类（文章）模块内容也全都完成，如若需要可在下面再写点热点链接之类的内容。

最后，single.php、category.php、archive.php（企业网站它的用处不大）、index.php 等都可以调用该 sidebar.php 了。

3) 菜单调用大分类

Sidebar 做好了，下面就让菜单上显示调用的大分类，调用

<?php wp_list_cats('include=3,4,5,6'); ?> 这句函数即可。include=ID：是我们调用的大类，多个大类的 ID 号用英文“,”隔开。

4) 修改 category.php

现在菜单上列出的是大分类调用，这样每个大分类下面就都会显示分类列表，但如果现在有的大分类不想显示分类列表，而显示该分类的第一篇文章，就需要来修改 category.php 文件了。思路和修改 sidebar 差不多。最初 category.php 的结构应该如下：

```
<?php get_header(); ?>  
  
<?php get_sidebar(); ?>  
  
<div id= " content " >  
  
    /*这里是调用分类的内容*/  
  
</div>  
  
<?php get_footer() ?>
```

现在因为需要点击大分类 3、5、6 时，页面上展示的是文章而非分类列表，所以在 <div id= " content " ></div>里添加一个 if 语句：

```
<div id="content">  
<?php if (is_category('3')) { ?><?php q...<?php } /*点击大分类3时，显示的内容*/  
    elseif (is_category('5')) { ?><?php q...<?php } /*点击大分类5时，显示的内容*/  
    elseif (is_category('6')) { ?><?php q...<?php } /*点击大分类6时，显示的内容*/  
    else { ?><?php i...<?php } /*点击大分类4或其他子分类时，这里是调用分类的内容*/  
?>  
</div>
```

把以前的 /*这里是调用分类的内容*/ `else { ?><?php i... <?php }` 的 `<?php i...` 中。

而需要在结构中添加的是大分类 3、5、6 中 `<?php q...` 这一块的内容，代码如下

```
<?php query_posts(' name=production '); ?>

<div class="post" >

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

        <h2><a href="<?php the_permalink() ?>" class="homesidetext1 title="<?php the_title() ?>"> <?php the_title() ?> </a> </h2>

        <div class="entry">

            <?php the_content('Read the rest of this entry »'); ?>

        </div>

    <?php endwhile;?>

</div>
```

query_posts () 是获取文章，其内参数可用 'p=文章 ID' 或 'name=文章缩略名' 来调用想放在大分类下首页的文章；

<h2></h2>里是文章名称；

the_content(); 是调用文章内容。

这样就可以在菜单中列出的大分类上点击，出来想要的文章或分类列表了。

以上几点就是企业级网站全站用 WordPress 搭建的大致思路。

二．注意事项

- 1、 这样的主题模板要基于 WordPress 的数据库，因为在写 side 时用到了分类的 id 号。
- 2、 正是因为第一条，这样的 WordPress 主题模板没有通用的，只能因企业而异。
- 3、 永久链接最好用 /%category%/ %postname%/ （分类+文章缩略名）的形式，这样 url 地址会看起来更有结构性，看起来也更像企业或产品网站。

创建漂亮的 Wordpress 日期按钮

我不知道我的那些日期按钮怎么样，但似乎你们都很喜欢。是的，今天，我们将来学习如何为 WordPress 创建一个漂亮的日期按钮。总共有三步简单的步骤：



第一步：输入代码

(请不要复制粘贴，输入到模板中)

```
<div class=" post-date" >
<div class=" month" ><?php the_time(' M' ) ?></div>
<div class=" day" ><?php the_time(' d' ) ?></div>
</div>
```

为了符合结构格式，在源代码中应该如下图所示：

```
<div class="post-date">
    <div class="month"><?php the_time('M') ?></div>
    <div class="day"><?php the_time('d') ?></div>
</div>
```

这些代码的意义解释：

<div class=" post-date" > – 我正在建立一个盒模型或者一个被命名为 " post-date" 的 DIV(层标签)。当然我也可以将它命名为 " little-potato."。

<div class=" post-month" > – 在 post-month盒模型中，建立一个现实月份的盒模型。

<?php the_time(' M') ?> – 这里我使用了 php函数 **the_time(' M')**，用来调用缩写的月份值。例如：六月 June的英文缩写为 JUN，但在我的日期按钮例子中，并没有使用 JUN代替 JUNE，这是因为我没有调用函数缩写版本的代码。

</div> – 结束月份的盒模型。（即关闭 DIV标签 -译者注）

<div class=" day" > – 建立日期盒模型。

`<?php the_time(' d') ?>` – 使用 PHP函数 `the_time(' d')`，用来调用日期值，会显示成两位数值。例如：03

`</div>` – 结束日期盒模型。

`</div>` – 没有任何内容可以添加到我的日期按钮中了，所以这里我也结束（关闭）了 `post-date`盒模型（或者称 DIV标签），因为我必须按顺序关闭所有打开的标签。

（更多关于 date的显示选项，请访问 php.net）

第二步：建立背景图片

你需要某些设计类软件用来设计你的日期按钮的背景图片，如 Photoshop，下图是我的日期按钮背景图片：



第三步：为其创建样式

这一步需要你了解一点 CSS的知识，样式决定于个人的偏好，因为你的背景图案会与我的不同。但，下面是些基本的内容：

```
.post-date{
float: left;
display: inline;
margin: 0 10px 0 0;
background: url(images/date_button_template.gif) no-repeat;
}
```

图片的名称应该命名为 `date_button_template.gif`。images/是存放图片的文件夹。如果你存储你的图片在其他文件夹，更改 images/为你的图片文件夹。

创建动态的置顶文章

我们的博客，[MaxPower](#)，需要某些文章比其他文章保持在最上面更长的时间。我们发现我们撰写的稿件像慢慢陷入沼泽，不能同其他的稿件区分开来。例如，在这个博客上我们会写关于所有东西的内容，但重点放在我们的旅行经验。问题来了，我们如何才能在首页重点显示这些经验，但并不会忽略所有其他的稿件？这是我想出的：

从 [wordpress help forums](#) 关于多重循环的各种讨论中让我想到一个使用两次循环的方法。一个用来查询并仅显示某个分类中的文章。另一个用来和擦寻在所有分类中的所有的文章。第一个查询是从 [the codex](#) 中复制来的，很容易理解：

```
$my_query = new WP_Query('category_name=frontpage&showposts=1');  
while ($my_query->have_posts()) : $my_query->the_post();  
$do_not_duplicate = $post->ID;
```

上述代码的意思是，变量 `my_query` 赋值为查询分类名为 `frontpage` 的所有文章的结果，并仅得到一篇文章。关键点在最后部分，给变量 `do_not_duplicate` 赋值为返回的单独文章的 ID 值。我们在下一步需要这个值。

下面一段代码是很重要很标准的，可以在许多主题中找到。它能获取所有日志：

```
if (have_posts()) : while (have_posts()) : the_post();  
//do stuff like format each post  
endwhile;
```

这段代码获取所有的日志，并根据 `//do stuff` 这行（依赖于模板）的格式显示。如果仅仅这样编写代码，就会发生你所选的文章（置顶文章）被显示两次的情况（一次在置顶区域，另一次则是在所有其他文章区域）。显示这样并不好，所以这里要使用到 `do_not_duplicate` 函数了。添加到一个 `if` 语句中阻止制定文章显示两次。下面是代码：

```
if (have_posts()) : while (have_posts()) : the_post();  
if( $post->ID == $do_not_duplicate )  
continue;  
//do stuff like format each post  
endwhile;
```

代码中的第二个 `if` 语句的意思是，获取所有日志，且当发现一篇文章的 ID 值与 `do_not_duplicate` 的值相等时中断执行（`continue` 语句），否则显示根据 `//do stuff` 这行的格式显示所有其他的文章。记住，`do_not_duplicate` 函数值所代表的文章已经被显示过了。

就这样当你把所有内容放到一起得到了什么？一个动态的置顶文章！且最棒的部分是当置顶文章被新的一篇代替，之前的文章会在下面所有文章的区域显示出来（取决于你选择了多少文章显示以及文章的发表频率）。

查询 wordpress 博客资料的 SQL 命令

查询 2006年所有的文章的 SQL命令：

```
SELECT COUNT(*)  
FROM `wp_posts`  
WHERE post_date >= ' 2006-01-01'  
AND post_date < '2007-01-01'  
AND post_status = ' publish'
```

查询 2006年所有评论的 SQL命令：

```
SELECT COUNT(*)  
FROM `wp_comments`  
WHERE comment_date >= ' 2006-01-01'  
AND comment_date < '2007-01-01'  
AND comment_approved = ' 1'
```

查询 2006每篇文章平均字数的 SQL命令：

```
SELECT AVG(LENGTH(post_content))  
FROM `wp_posts`  
WHERE post_date >= ' 2006-01-01'  
AND post_date < '2007-01-01'  
AND post_status = ' publish'
```

查询 2006年文章总字数的 SQL命令：

```
SELECT SUM(LENGTH(post_content))  
FROM `wp_posts`  
WHERE post_date >= ' 2006-01-01'  
AND post_date < '2007-01-01'  
AND post_status = ' publish'
```

给 WP 添加 Ctrl+Enter 回复快捷键

首先是在模版的 comments.php 底部加一段 JS代码

```
<script type="text/javascript">
document.getElementById("comment").onkeydown = function (moz_ev)
{
var ev = null;
if (window.event){
ev = window.event;
}else{
ev = moz_ev;
}
if (ev != null && ev.ctrlKey && ev.keyCode == 13)
{
document.getElementById("submit").click();
}
}
</script>
```

最后是找到这段，并修改：

```
<input name="提交 " type="submit" class="button" id="submit"
tabindex="5" value="提交 Ctrl+Enter" />
```

方法二：(厉戈)

在 comments.php 中找到以下代码：

```
<TEXTAREA id=comment tabIndex=4 name=comment rows=10 cols=105
jQuery1250561977609="48"></TEXTAREA>
```

修改为：

```
<TEXTAREA id=comment
onkeydown="if(event.ctrlKey&&event.keyCode==13){document.getElementById("submit")
.click();return false};" tabIndex=4 name=comment rows=10 cols=105></TEXTAREA>
```

总结：相比来说，第二种方法更简单，却实现了相同的效果。

需要注意的是：这里的 id一定要和 js里的 id值一样不然无法处理。

WordPress 的 10 个杀手级 Hack 技巧

2008 年对 WordPress 来说是非同寻常的一年，它的软件多次更新，最终促成重大版本 2.7 的发布，许多网站因此也开始使用 WordPress 软件。一些用户在使用过程中对 WordPress 的代码等作了许多改进，更多 WordPress 用户也因此受益。

这篇文章将向大家介绍 10 个最新最实用的 WordPress hack，它能让你的博客引擎开足马力，挣脱束缚，最大限度地为你服务。此外，每个 hack 还都附有相关说明，这样你不仅能让 WordPress 释放最大力量，还能知道如何操作。

1. 只在搜索引擎结果中显示 AdSense 广告

问题：众所周知，一般访问并不点击广告。广告点击率 90% 是来自搜索引擎的显示结果。另一个问题是 Google 的智能计价。智能定价意味着 CTR 比较低，点击率单价大打折扣。例如，如果一次点击正常获得 1 美元的话，Google 的智能计价只会是 0.1 美元。郁闷吧？但是，以下的代码改进就能解决这个问题了。它只在搜索引擎结果中显示 AdSense 广告，这意味着你能得到更多的点击率，更高的 CTR。

解决方案：

1. 打开主题中的 functions.php 文件。

2. 粘贴进下代码：

```
function scratch99_fromasearchengine(){  
    $ref = $_SERVER['HTTP_REFERER'];  
    $SE = array('/search?', 'images.google.', 'web.info.com', 'search.', 'del.icio.us/search',  
    'soso.com', '/search/', '.yahoo.');
```

3. 把以下代码粘贴到 AdSense 广告显示位置的相应模板中。它只会在搜索引擎结果中显示：

```
if (function_exists('scratch99_fromasearchengine')) {  
    if (scratch99_fromasearchengine()) {  
        INSERT YOUR CODE HERE  
    }  
}
```

代码说明：这个改进以 `scratch99_fromasearchengine()` 函数开头，包括了用来指定搜索引擎的 \$SE 数组变量。这样，只要把新元素加入到数组中就能轻松添加新的搜索引擎。如果使用搜索引擎搜索的访客包含了 \$SE 数组变量，`scratch99_fromasearchengine()` 接着就会返回 true 值。

2. 避免在多循环中出现重复文章

问题：由于“杂志型”主题的流行，现在很多 WordPress 用户在他们的首页上都使用了多个循环，因此他们也迫切需要在第二个循环上避免出现重复文章。

解决方案：使用 PHP 数组就能轻松解决这一问题

1. 创建一个 PHP 数组，并把第一个循环的所有文章 ID 放在其中

```
<h2>Loop n°1</h2>

<?php
$ids = array();
while (have_posts()) : the_post();
the_title();
?>

<br />

<?php $ids[] = $post->ID;
endwhile; ?>
```

2. 对于第二个循环：使用 PHP 函数 `in_array()` 来检查 \$ids 数组中是否包含了某个文章 ID。如果这个文章 ID 没有包含在此数组中，意味着第一个循环中没有此篇文章，那么系统就可以显示它了。

```
<h2>Loop n°2</h2>

<?php
query_posts("showposts=50");
while (have_posts()) : the_post();
if (!in_array($post->ID, $ids)) {
the_title();?>
<br />
<?php }
endwhile; ?>
```

代码说明：执行第一个循环时，其中的所有文章 ID 都会被放到一个数组变量中。因此，第二个循环执行时，我们只要到这个数组中检查当前的文章 ID 是否已显示在第一个循环中就可以了。

3.用页码代替“上页”和“下页”

问题：WordPress 有默认函数来显示上页和下页页面，虽然这比没有这个什么功能好，但是我不明白开发人员为什么不把页面写进核心代码呢。当然，我们可以用插件来生成页码，但是如果能把页码直接插入主题岂不更好？！

解决方案：这里使用 WP-PageNavi 插件把页码直接写入主题

1. 首先下载 WP-PageNavi 插件
2. 在硬盘驱动上解压插件存档，并把 wp-pagenavi.php 和 wp-pagenavi.css 文件上传到主题目录
3. 打开要放置页码的文件（如 index.php, categories.php, search.php,等），找到以下代码：

```
<div class="navigation">
<div class="alignleft"><?php next_posts_link('Previous entries') ?></div>
<div class="alignright"><?php previous_posts_link('Next entries') ?></div>
</div>
```

用以下代码代替以上内容：

```
<?php
include('wp-pagenavi.php');
if(function_exists('wp_pagenavi')) { wp_pagenavi(); }
?>
```

4. 接着请修改插件文件。打开 wp-pagenavi.php 文件并找到以下代码（61 行）：

```
function wp_pagenavi($before = "", $after = "") {
global $wpdb, $wp_query;
修改成：
function wp_pagenavi($before = "", $after = "") {
global $wpdb, $wp_query;
pagenavi_init(); //Calling the pagenavi_init() function
```

5. 最后，我们要把 wp-pagenavi 样式表添加到博客。

打开 header.php 文件，把以下代码添加进去：

```
<linkrel="stylesheet"href="<?php echoTEMPLATEPATH.'/pagenavi.css';?>"type="text/css"
media="screen"/>
```

代码说明：这个代码改进直接在主题文件中加入添加了插件代码。我们另外还调用了 pagenavi_init() 函数以使页码能够正常显示。

4.自动获取文章图像

问题：使用自定义字段来显示和日志相关的图像固然很好，但是许多用户想直接检索并使用文章本身嵌入的图像。

解决方案： 至今为止，还没有这样的插件。值得庆幸的是，以下循环将帮我们解决这一问题：它会搜索文章内容的图像并把它们显示出来

1. 把以下代码粘贴到主题文件任意位置：

```
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
<?php
$szPostContent = $post->post_content;
$szSearchPattern = '~<img [\>]*\ />~';
// Run preg_match_all to grab all the images and save the results in $aPics
preg_match_all( $szSearchPattern, $szPostContent, $aPics );
// Check to see if we have at least 1 image
$iNumberOfPics = count($aPics[0]);
if ( $iNumberOfPics > 0 ) {
// Now here you would do whatever you need to do with the images
// For this example the images are just displayed
for ( $i=0; $i < $iNumberOfPics ; $i++ ) {
echo $aPics[0][$i];
};
};
endwhile;
endif;?>
```

代码说明：以上代码实际上包含了一个 WordPress 循环。使用 PHP 和正则表达式的唯一区别就是前者会自动搜索文章内容中的图像而不是仅仅显示文章。一旦发现图像，系统就会显示。

5.创建“发送到 Twitter”按钮

问题：你是 Twitter 用户吗？如果是，相信你一定了解和朋友在线分享有趣内容的乐趣。那么，为什么不给你的读者也提供一个机会，让他们可以直接把你的文章 URL 发送到 Twitter 以给你带来更多流量呢？

解决方案：这个代码改进非常简单。只要创建一个带有 status 参数的 Twitter 链接就可以了。而对于 WordPress,直接使用 the_permalink()函数就可获取文章 URL 了：

```
<a href="http://twitter.com/home?status=Currentlyreading<?php the_permalink(); ?>" title="ClicktosendthispagetoTwitter!" target="_blank">ShareonTwitter</a>
```

非常简单，对吧？但它同时也非常实用！

6. 使用直引号，避免弯引号

问题：如果你经常在自己的网站上发布代码片段的话，可能会经常碰上这类问题：某用户会说你发布的代码不起作用。这是为什么呢？WordPress 默认情况下会将直引号转为“smart 引号”，而后者会截断代码片段。

解决方案：要避免出现这些弯引号，请按以下操作：

1. 打开主题中的 functions.php 文件。如果不存在该文件的话，请自行创建一个。
2. 粘贴进如下代码：

```
<?php remove_filter('the_content', 'wptexturize'); ?>
```

3. 保存文件。大功告成！

代码说明：wptexturize()函数会自动将直引号转为 smart 引号。而通过使用 remove_filter()函数，我们会告知 WordPress 不要对日志内容使用 wptexturize()函数，问题自然得到解决。

7. 如无 Referrer 请求则禁止发表评论

问题：垃圾评论是让所有博主都头疼的问题。尽管 Akismet 能够帮你应付这些问题，但如果能化被动为主动岂不是更好？以下给出的代码在 wp-comments-post.php 被访问时，会查找 referrer（即 URL 来源页面）。如果此时存在一个 referrer，且为你博客的 URL，那么就可以发表评论。反之，页面将会停止载入，而评论也就无法发表了。

解决方案：要应用此 hack 的话，直接将以下代码粘贴到外观主题的 function.php 文件中即可。如果外观主题不含此文件，请自行创建一个。

```
function check_referrer() {  
    if (!isset($_SERVER['HTTP_REFERER']) || $_SERVER['HTTP_REFERER'] == "") {  
        wp_die( __('Please enable referrers in your browser, or, if you\'re a spammer, bugger off!') );  
    }  
}  
  
add_action('check_comment_flood', 'check_referrer');
```


8.在 WordPress 导航条中使用 CSS 滑动门

问题：虽然内置的 `wp_list_pages()`和 `wp_list_categories()`函数很有用，但它们不允许嵌入 `` 元素。这样一来，就只能使用目前最赞的 CSS 滑动门技术了。而且幸运的是，我们只要稍微借助一下 PHP 和正则表达式就能够在 WordPress 博客中使用超酷的滑动门技术了。

由于有关 CSS 滑动门的教程已经很多了，因此在这里我们就不对其工作方式再做任何说明。如果你想对此技术进行更深入的了解，可以看看[这篇精品文章](#)。想看看实际效果的话，请[点击这里](#)。

1. 创建你所需的图片，并对 WordPress 外观主题中的 `style.css` 文件进行编辑。这里给出例子：

```
#nav a, #nav a:visited {
display: block;
}

#nav a: hover, #nav a: active {
background: url(images/tab-right.jpg) no-repeat 100% 1px;
float: left;
}

#nav a span {
float: left;
display: block;
}

#nav a: hover span {
float: left;
display: block;
background: url(images/tab-left.jpg) no-repeat 0 1px;
}
```

2. 现在让我们来编辑 `header.php` 文件吧。根据自身需要将如下代码复制粘贴进去即可：

```
<ul id="nav">

<li><a href="<?php echo get_option('home'); ?>/"><span>Home</span></a></li>

<?php          echo          preg_replace('@<li([^\>]*)><a([^\>]*)>(.*)\</a>@i',
'<li$1><a$2><span>$3</span></a>',
wp_list_pages('echo=0&orderby=name&exclude=181&title_li=&depth=1')); ?>

</ul>

列出页面：

<ul id="nav">

<li><a href="<?php echo get_option('home'); ?>/"><span>Home</span></a></li>

<?php          echo          preg_replace('@<li([^\>]*)><a([^\>]*)>(.*)\</a>@i',
```

```
'<li$1><a$2><span>$3</span></a>',  
wp_list_categories('echo=0&orderby=name&exclude=181&title_li=&depth=1')); ?>  
</ul>
```

代码说明：此例中，我们在 `wp_list_pages()` 和 `wp_list_categories()` 函数中使用了 `echo=0` 参量，它允许你获取函数结果而无需将其直接打印在屏幕上。接着函数结果将用于 `PHP preg_replace()` 函数，并最终显示出来，此时 `` 和 `<a>` 标签之间已加入了 `` 标签。

9. WordPress 博客百变头图

问题：确切地说这并不是什么问题，但有不少 WordPress 用户希望自己的博客头图能够丰富多变。

解决方案：

1. 当你相中了一些头图图片后，将它们命名为 1.jpg, 2.jpg, 3.jpg，诸如此类。当然，你可以使用任意多的图片。
2. 将图片上传到 `wp-content/themes/yourtheme/images` 目录下。
3. 打开 `header.php` 文件，并将如下代码粘入：

```
$num = rand(1,10); //Get a random number between 1 and 10, assuming 10 is the total  
number of header images you have  
&lt;div id="header" style="background:transparent url(images/.jpg) no-repeat top left;">
```

4. 完成！现在博客各页面或日志页面将会显示随机的头图了。

代码说明：这里没什么好说的。我们直接使用 `PHP rand()` 函数初始化一个 `$num` 变量来随机获取 1 到 10 之间的数字。接着我们将 `$num` 变量的结果连上当前外观主题的路径就 OK 了。

10. 显示撰写计划

问题：许多博主都希望自己拥有更多的读者，有更多的 RSS feed。而把文章撰写计划罗列出来显然能提高访客的好奇心和兴趣。

解决方案：打开任意一个主题文件并粘贴进以下代码：

```
<?php  
$my_query=newWP_Query('post_status=future&order=DESC&showposts=5');  
if($my_query->have_posts()){  
while($my_query->have_posts()):$my_query->the_post();?>  
<li><?phpthe_title();?></li>  
<?phpendwhile;  
}  
?>
```

代码说明：在这个代码改进中，我们使用 WP_Query 类创建了自定义语句以发送数据库查询并提取最近将要撰写的五篇文章。完成后，系统就会使用一个简洁的 WordPress 循环来显示文章标题。

侧边加入登陆界面：

```
<li>    <?php global $user_ID, $user_identity, $user_level ?>
    <?php if ( $user_ID ) : ?>
    <h2>控制面板</h2>
    <ul>
        <li>已登录 <strong><?php echo $user_identity ?></strong>.
        <ul>
            <li><a href="<?php bloginfo('url') ?>/wordpress/wp-admin/">后台</a></li>
            <?php if ( $user_level >= 1 ) : ?>
            <li><a href="<?php bloginfo('url') ?>/wordpress/wp-admin/post-new.php">撰写文章</a></li>
            <?php endif // $user_level >= 1 ?>
            <li><a href="<?php bloginfo('url') ?>/wordpress/wp-admin/profile.php">个人资料</a></li>
            <li><a href="<?php echo wp_logout_url() ?>">注销</a></li>
        </ul>
    </li></ul>
    <?php elseif ( get_option('users_can_register') ) : ?>
    <h2>登录</h2>
    <ul>
        <li>
            <form action="<?php bloginfo('url') ?>/wordpress/wp-login.php" method="post">
                <p>
                    <label for="log"><input type="text" name="log" id="log" value="<?php echo
wp_specialchars(stripslashes($user_login), 1) ?>" size="22" /> 用户</label><br />
                    <label for="pwd"><input type="password" name="pwd" id="pwd" size="22"
/>密码</label><br />
                    <input type="submit" name="submit" value="Send" class="button" />
                    <label for="rememberme"><input name="rememberme" id="rememberme"
type="checkbox" checked="checked" value="forever" /> 记住我</label><br />
                </p>
                <input type="hidden" name="redirect_to" value="<?php echo
$_SERVER['REQUEST_URI']; ?>" />
            </form>
        </li>
        <li><a href="<?php bloginfo('url') ?>/wordpress/wp-register.php">注册</a></li>
        <li><a href="<?php bloginfo('url') ?>/wordpress/wp-login.php?action=lostpassword">找回密码</a></li>
    </ul>
```

获取单篇文章内的关键字及描述方法：

```
<?php if(is_single()){
    $keywords = " ";
    $tags = wp_get_post_tags($post->ID);
    foreach ($tags as $tag ) {
        $keywords = $keywords . $tag->name . ",";
    }
    if (have_posts()) : while (have_posts()) : the_post();
        if(strlen($text) > 150){
            $text = substr($text,0,150);
            $text= mb_convert_encoding($text,"UTF-8");
            $text=preg_replace("/(\s+)/","",$text);
        }
    }
    if ($post->post_excerpt){
        $text = mb_substr(strip_tags($post->post_excerpt),0,150);
    } else {
        $text = mb_substr(strip_tags($post->post_content),0,150);
    }?>

<meta name="keyword" content="<?php echo $keywords; ?>"/>
<meta name="description" content="<?php echo $text; ?>"/>
<?php } else { ?>
<meta name="keyword" content="草根评论，时事经济，网络热点，IT 视点，最新软件，网站 SEO，心情故事"/>
<meta name="description" content="<?php bloginfo('description'); ?>"/>
<?php } ?>
```