

wordpress 代码调用大全更新到_3.0 版

<?php get_calendar() ?> 日历

no-category-parents --->优化路径插件

第三款显示同一分类下面文章的 wordpress 插件：中文 wordpress 工具箱 补充版。

中文 wordpress 工具箱 补充版，顾名思义，就是对 wordpress 中午工具箱的补充，增加了显示同一分类下面文章的功能。

调用同一分类下面文章的函数格式是

:

<?php get_category_post_list(\$num, \$limit, \$order="DESC"); ?>

其中\$num 对应的是分类的 ID; \$limit 对应的是显示文章的数量，\$order 对应的是按顺序显示还是乱序显示文章，具体可以参考源代码。

该插件不太适用于在具体文章后面显示同一类别下的文章，但是非常适用于那些想把 wordpress 打造成 cms 的朋友。自定义一个首页，然后在首页调用不同类别的文章，设定一个非常漂亮的首页，肯定是没有问题的。

Wordpress 模板基本文件

style.css 样式表文件

index.php 主页文件

single.php 日志单页文件

page.php 页面文件

archie.php 分类和日期存档页文件

searchform.php 搜索表单文件

search.php 搜索页面文件

comments.php 留言区域文件(包括留言列表和留言框)

404.php 404 错误页面

header.php 网页头部文件

sidebar.php 网页侧边栏文件

footer.php 网页底部文件

Wordpress Header 头部 PHP 代码

注：也就是位于<head>和</head>之间的 PHP 代码

```
<?php bloginfo( ' name' ); ?> 网站标题
<?php wp_title(); ?> 日志或页面标题
<?php bloginfo( ' stylesheet_url' ); ?> Wordpress 主题样式表文件 style.css 的相对地址
<?php bloginfo( ' pingback_url' ); ?> Wordpress 博客的 Pingback 地址
<?php bloginfo( ' template_url' ); ?> Wordpress 主题文件的相对地址
<?php bloginfo( ' version' ); ?> 博客的 Wordpress 版本
<?php bloginfo( ' atom_url' ); ?> Wordpress 博客的 Atom 地址
<?php bloginfo( ' rss2_url' ); ?> Wordpress 博客的 RSS2 地址
<?php bloginfo( ' url' ); ?> Wordpress 博客的绝对地址
<?php bloginfo( ' name' ); ?> Wordpress 博客的名称
<?php bloginfo( ' html_type' ); ?> 网站的 HTML 版本
<?php bloginfo( ' charset' ); ?> 网站的字符编码格式
```

Wordpress 主体模板 PHP 代码

```
<?php the_content(); ?> 日志内容
<?php if(have_posts()) : ?> 确认是否有日志
<?php while(have_posts()) : the_post(); ?> 如果有，则显示全部日志
<?php endwhile; ?> 结束 PHP 函数” while”
<?php endif; ?> 结束 PHP 函数” if”
<?php get_header(); ?> header.php 文件的内容
<?php get_sidebar(); ?> sidebar.php 文件的内容
<?php get_footer(); ?> footer.php 文件的内容
<?php the_time( ' m-d-y' ) ?> 显示格式为” 10-12-13” 的日期
<?php comments_popup_link(); ?> 显示一篇日志的留言链接
<?php the_title(); ?> 显示一篇日志或页面的标题
<?php the_permalink() ?> 显示一篇日志或页面的永久链接/URL 地址
<?php the_category( ' , ' ) ?> 显示一篇日志或页面的所属分类
<?php the_author(); ?> 显示一篇日志或页面的作者
<?php the_ID(); ?> 显示一篇日志或页面的 ID
<?php edit_post_link(); ?> 显示一篇日志或页面的编辑链接
<?php get_links_list(); ?> 显示 Blogroll 中的链接
<?php comments_template(); ?> comments.php 文件的内容
<?php wp_list_pages(); ?> 显示一份博客的页面列表
<?php wp_list_cats(); ?> 显示一份博客的分类列表
<?php next_post_link( ' %link ' ) ?> 下一篇日志的 URL 地址
<?php previous_post_link( ' %link' ) ?> 上一篇日志的 URL 地址
<?php get_calendar(); ?> 调用日历
<?php wp_get_archives() ?> 显示一份博客的日期存档列表
```

<?php posts_nav_link(); ?> 显示较新日志链接(上一页)和较旧日志链接 (下一页)
 <?php bloginfo(' description'); ?> 显示博客的描述信息

其它的一些 Wordpress 模板代码

/%postname%/ 显示博客的自定义永久链接
 <?php the_search_query(); ?> 搜索表单的值
 <?php _e(' Message'); ?> 打印输出信息
 <?php wp_register(); ?> 显示注册链接
 <?php wp_loginout(); ?> 显示登入/登出链接
 <!-- next page --> 在日志或页面中插入分页
 <!-- more --> 截断日志
 <?php wp_meta(); ?> 显示管理员的相关控制信息
 <?php timer_stop(1); ?> 显示载入页面的时间
 <?php echo get_num_queries(); ?> 显示载入页面查询

----->

使用文章访问统计插件 WP-PostViews 即可实现热门文章的展示, 但是类似可定制性不够强。例如循环前他、翻页阅读等等。

事实上 WordPress 提供的 query_posts() 方法允许按照评论数目进行文章排序, 而且可以良好支持翻页, 甚至是兼容 WP-PostNavi 插件。通常情况下, 例如 index.php 模板中, 主循环的基本格式为

```
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
// insert code here
```

<?php endwhile; else : endif; ?>现在通过过滤机制让其按照评论数排序, 则需要在主循环之前添加以下代码:

```
<?php $paged = (get_query_var('paged')) ? get_query_var('paged') : 1;
query_posts('orderby=comment_count&paged='.$paged); ?>标红的 orderby=comment_count 字段即为排序条件。WP-PostNavi 翻页测试顺利通过。当然也可以通过需要设置, 排序参数的完整列表如下:
```

排序参数:

orderby=author - 按作者排序
 orderby=date - 按发布时间排序
 orderby=title - 按文章标题排序
 orderby=modified - 按修改时间排序
 orderby=menu_order 按菜单顺序排序, 仅支持单页面
 orderby=parent - 这个芒果也不是很清楚...
 orderby=ID - 按文章 ID 排序
 orderby=rand - 随机排序

orderby=meta_value - 通过自定义字段值排序，需要同时给定自定义字段 (meta_key)的名称

orderby=none - 无序排列，版本要求 2.8

orderby=comment_count - 按评论数排序，版本要求 2.9

其实很多功能 WordPress 原生都是支持的，并不需要通过 SQL 查询实

----->

当前分类侧栏显示最近 10 条更新文章:

```
<?php $posts = get_posts( "category= $cat &numberposts=10" ); ?>
<?php if( $posts ) : ?>
<ul><?php foreach( $posts as $post ) : setup_postdata( $post ); ?>
<li><a href="<?php the_permalink() ?>" rel="bookmark" title="<?php the_title(); ?>">
<?php the_title(); ?></a></li>
<?php endforeach; ?></ul>
<?php endif; ?>
```

----->

根据评论数量来判断而调用

```
<UL>
<?php $result = $wpdb->get_results("SELECT comment_count,ID,post_title FROM
$wpdb->posts ORDER BY comment_count DESC LIMIT 0 , 5");
foreach ($result as $post) {
setup_postdata($post);
$postid = $post->ID;
$title = $post->post_title;
$commentcount = $post->comment_count;
if ($commentcount != 0) { ?>
<LI><A title="<?php echo $title ?>" href="<?php echo get_permalink($postid); ?>">
<?php echo $title ?></A> {<?php echo $commentcount ?>}
<?php } } ?>
</LI></UL>
```

----->

用代码实现 WordPress 首页只显示某些分类的文章

2010 年 3 月 28 日 WordPress 1 条评论 如何实现在 WordPress 首页不显示某些分类的文章、或者只显示某些分类的文章呢？这方面的插件很多，但大多不太稳定，容易和其他插件发生冲突。而且不管怎样，插件应该能少用就少用。

其实，用代码就可以很实现这一点，方法如下：

在 index.php 或 home.php 里查找

<?php if (have_posts()) : ?>在之前加入以下代码就可以了。

```
<?php
//只显示某分类的文章
$cat_id='1,5';//the category ID
//不显示某些分类的话 这样写
//$cat_id='-4,-8';//the category ID
$limit = get_option('posts_per_page');
$paged = (get_query_var('paged')) ? get_query_var('paged') : 1;
query_posts('cat=' . $cat_id . '&showposts=' . $limit . '&paged=' . $paged);
$wp_query->is_archive = true; $wp_query->is_home = false;
?>
```

----->

WordPress 得到所有的分类 YY 发表于 2010-06-04 14:32 | 来源: | 阅读 1,215 次点击
WordPress 得到所有的分类已经是一个很简单的事件了，我们用”wp_list_categories()”函数就可以实现，但是如果我们要对每个分类进行操作就是一件很难的事件了。
查询了很多资料，终于找到了一个函数可以用来解决这个问题，”get_categories()”函数，它返回与查询参数匹配的类别 objects 的数组，和”wp_list_categories()”函数非常相像。
好了，下面我们就用个例子来讲解一下这个函数。
下面的代码就是得到所有的分类，并显示出每个分类下最新的 10 篇文章：

```
<?php
// 得到所有分类列表
$categories = get_categories();

// 循环所有分类
foreach ($categories as $cat) {

    // 得到分类 ID
    $catid = $cat->cat_ID;

    // 得到分类下 10 篇最新文章
    query_posts("showposts=10&cat=$catid"); ?>

    <!-- 输出分类标题及链接 -->
    <h4><a href="<?php echo get_category_link($catid);?>"
        title="<?php echo strip_tags(category_description($catid)); ?>">
        <?php single_cat_title(); ?>
    </a></h4>
```

```

<!-- 输出 10 篇最新文章标题及链接 -->
<ul>
  <?php while (have_posts()) : the_post(); ?>
    <li>
      <a href="<?php the_permalink() ?>" title="<?php the_title(); ?>">
        <?php the_title(); ?></a>
      <?php the_time('m-d'); ?>
    </li>
  <?php endwhile; ?>
</ul>
<?php } ?>

```

----->

WordPress 获取当前分类的根分类 id 函数 Wordpress 2010-08-26 15:52:05 阅读 77 评论 0
 字号：大中小 订阅

之前在做这个 WordPress 的 Themes 时，需要实现在文章内容页面（single.php）的循环体外部输出该篇文章所属分类的所有同一级分类，也就是该篇文章所属分类的上一级分类下的子分类。当然，这样只适用于只有二级分类目录的情况！虽然貌似 WordPress 有内置这样的函数，但是都没达到我想要的效果，因为有些 WordPress 的内置函数是要使用在 The Loop 的循环体内的，所以用 WordPress 内置函数无法实现我需要的功能，只有自己动手了，后来使用了以下两个方法解决了问题！

```

function get_category_root_id($cat)
{
  $this_category = get_category($cat); // 取得当前分类
  while($this_category->category_parent) // 若当前分类有上级分类时，循环
  {
    $this_category = get_category($this_category->category_parent); // 将当前分类设为上级分类（往上爬）
  }
  return $this_category->term_id; // 返回根分类的 id 号
}

function fengzx_get_category_ID() {
  $category = get_the_category();
  return $category[0]->cat_ID;
}

```

使用的时候就这样：

----->

wordpress 单个文章页面的侧边栏如果还是显示和首页，分类页面一样的文章就没什么意思了。下面的代码是显示文章所属分类的文章列表，这样可以让访问者可以方便查看相关文章。回头再改改，看看能不能在分类页面上显示最受欢迎的文章。添加下列代码在合时的位置即可

```
<?php
/*
single page?show current category articles
*/
?>
<?php
if ( is_single() ) :
global $post;
$categories = get_the_category();
foreach ($categories as $category) :
    ?>
    <li class="widget widget_recent_entries" id="<?php $category->term_id;?>-posts">
        <h2 class="widgettitle"><?php echo $category->name; ?></h2>
        <ul>
            <?php
            $posts = get_posts('numberposts=5&category='. $category->term_id);
            foreach($posts as $post) :
                ?>
                <li>
                    <a href="<?php the_permalink(); ?>"><?php the_title(); ?></a>
                </li>
            <?php endforeach; ?>
        </ul>
    </li>
<?php
endforeach; endif ; ?>
<?php
/*
end show current category articles
*/
?>
```

----->

1. wordpress 调用最新文章

Wordpress 最新文章的计算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)
或者
```

```
<?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. wordpress 调用随机文章

```
<?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; ?>
```

3. wordpress 调用最新留言

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 = '' 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; ?>
```

4. wordpress 调用相关文章

在文章页显示相关文章

```
<?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="<?php the_title_attribute(); ?>"<?php the_title();?>
<?php comments_number('','(1)','(%)); ?></a></li> <?php endwhile; } } wp_reset_query(); ?>
```

5.wordpress 调用指定分类的文章

```
<?php $posts = get_posts( "category=4&numberposts=10" ); ?><?php if( $posts ) : ?><ul><?php
foreach( $posts as $post ) : setup_postdata( $post ); ?><li><a href="<?php the_permalink() ?>"
rel="bookmark" title="<?php the_title(); ?>"><?php the_title(); ?></a></li><?php
endforeach; ?></ul><?php endif; ?>
```

6.wordpress 去评论者链接的评论输出

```
<?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,1,14) 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;?>
```

7.wordpress 调用含 gravatar 头像的评论输出

```
<?php global $wpdb; $sql = "SELECT DISTINCT ID, post_title, post_password, comment_ID,
comment_post_ID, comment_author, comment_date_gmt,
comment_approved,comment_author_email, comment_type,comment_author_url,
SUBSTRING(comment_content,1,10) 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 comment_author != '郑永' AND
post_password = '' ORDER BY comment_date_gmt DESC LIMIT 10"; $comments =
$wpdb->get_results($sql); $output = $pre_HTML; foreach ($comments as $comment)
{ $output .= "\n<li>".get_avatar(get_comment_author_email('comment_author_email'), 18). " <a
href=\"" . get_permalink($comment->ID) . "#comment-" . $comment->comment_ID . "\" title=\"\" .
$comment->post_title . " 上的评论\">". strip_tags($comment->comment_author) .": " .
"
```

```
strip_tags($comment->com_excerpt) . "</a></li>"; } $output .= $post_HTML; $output =
convert_smilies($output); echo $output; ?>
```

上面代码把 comment_author 的值改成你的 ID，18 是头像大小，10 是评论数量。

8.wordpress 调用网站统计大全

1、日志总数： <?php \$count_posts = wp_count_posts(); echo \$published_posts = \$count_posts->publish;?> 2、草稿数目： <?php \$count_posts = wp_count_posts(); echo \$draft_posts = \$count_posts->draft; ?> 3、评论总数： <?php echo \$wpdb->get_var("SELECT COUNT(*) FROM \$wpdb->comments");?> 4、成立时间： <?php echo floor((time()-strtotime("2008-8-18"))/86400); ?> 5、标签总数： <?php echo \$count_tags = wp_count_terms('post_tag'); ?> 6、页面总数： <?php \$count_pages = wp_count_posts('page'); echo \$page_posts = \$count_pages->publish; ?> 7、分类总数： <?php echo \$count_categories = wp_count_terms('category'); ?> 8、链接总数： <?php \$link = \$wpdb->get_var("SELECT COUNT(*) FROM \$wpdb->links WHERE link_visible = 'Y'"); echo \$link; ?> 9、用户总数： <?php \$users = \$wpdb->get_var("SELECT COUNT(ID) FROM \$wpdb->users"); echo \$users; ?> 10、最后更新： <?php \$last = \$wpdb->get_results("SELECT MAX(post_modified) AS MAX_m FROM \$wpdb->posts WHERE (post_type = 'post' OR post_type = 'page') AND (post_status = 'publish' OR post_status = 'private')");\$last = date('Y-n-j', strtotime(\$last[0]->MAX_m));echo \$last; ?> 9.wordpress 判断语句

is_single()

判断是否是具体文章的页面

is_single(' 2')

判断是否是具体文章 (id=2) 的页面

is_single('Beef Stew')

判断是否是具体文章 (标题判断) 的页面

is_single('beef-stew')

判断是否是具体文章 (slug 判断) 的页面

comments_open()

是否留言开启

pings_open()

是否开启 ping

`is_page()`

是否是页面

`is_page(' 42')`

id 判断，即是否是 id 为 42 的页面

`is_page('About Me')`

判断标题

`is_page('about-me')`

slug 判断

`is_category()`

是否是分类

`is_category(' 6')`

id 判断，即是否是 id 为 6 的分类

`is_category('Cheeses')`

分类 title 判断

`is_category('cheeses')`

分类 slug 判断

`in_category(' 5')`

判断当前的文章是否属于分类 5

`is_author()`

将所有的作者的页面显示出来

`is_author(' 1337')`

显示 author number 为 1337 的页面

```
is_author('Elite Hacker')
```

通过昵称来显示当前作者的页面

```
is_author('elite-hacker')
```

下面是通过不同的判断实现以年、月、日、时间等方式来显示归档

```
is_date()
```

```
is_year()
```

```
is_month()
```

```
is_day()
```

```
is_time()
```

判断当前是否是归档页面

```
is_archive()
```

判断是否是搜索

```
is_search()
```

判断页面是否 404

```
is_404()
```

判断是否翻页，比如你当前的 blog 是 <http://blog.sina.com.cn/gutengfei0707> 显示 <http://blog.sina.com.cn/gutengfei0707?paged=2> 的时候，这个判断将返回真，通过这个函数可以配合 `is_home` 来控制某些只能在首页显示的界面，

例如：

```
<?php if(is_single()):?> //这里写你想显示的内容，包括函数 <?php endif;?>
```

或者：

```
<?php if(is_home() && !is_paged() ):?> //这里写你想显示的内容，包括函数 <?php endif;?>
```

10.wordpress 非插件同步 twitter

```
<?php require_once (ABSPATH . WPINC . '/class-feed.php' ); $feed = new SimplePie();
$feed->set_feed_url(          '          http://feeds.feedburner.com/agting          ');
$feed->set_file_class( 'WP_SimplePie_File' ); $feed->set_cache_duration(600); $feed->init();
$feed->handle_content_type(); $items = $feed->get_items(0,1); foreach($items as $item) { echo
'<a target="_blank" rel="external nofollow" title="Follow Me on Twitter" href="
http://twitter.com/agting" >@郑永</a>: '. $item->get_description(); } ?>
```

代码中的 agting 改成你的 twitter 用户名，郑永改成你的名字。

另一种调用方法需要你的空间是国外主机：

```
<?php // Your twitter username. $username = "wange1228"; // Prefix - some text you want
displayed before your latest tweet. // (HTML is OK, but be sure to escape quotes with backslashes:
for example href="link.html") // Suffix - some text you want display after your latest tweet.
(Same rules as the prefix.) $suffix = ""; $feed = "http://search.twitter.com/search.atom?q=from:" .
$username . "&rpp=1"; function parse_feed($feed) { $stepOne = explode("<content
type=\"html\">", $feed); $stepTwo = explode("</content>", $stepOne[1]); $tweet = $stepTwo[0];
$tweet = str_replace("&lt;", "<", $tweet); $tweet = str_replace("&gt;", ">", $tweet); return
$tweet; } $twitterFeed = file_get_contents($feed); echo stripslashes($prefix) .
parse_feed($twitterFeed) . stripslashes($suffix); ?>
```

ZeroZ 总结了一下这个方法的特点：

- 1、非插件！
- 2、不用验证用户名和密码，也就是说你可以指定调用任何一个人的 tweet！
- 3、可以自定义 tweet 信息后显示的文字，就是 \$suffix = “ ”；这里！
- 4、只能调用最新的一条 tweet，刚好满足我的需求。
- 5、大概只有国外空间才能使用！（经我验证，确实如此）

WP 全自动外链插件

在这里，向大家隆重推荐WP全自动外链插件：[Automatic Backlink Creator](#)插件。

这款软件本人使用过，效果非常不错，所以今天在这里推荐一下，希望可以节省大家做外链的时间和精力！

Automatic Backlink Creator 主要针对 wordpress 程序建立的网站。热衷于 WP 的站长朋友，尤其是针对做外贸的站长朋友们，主做 Google、Yahoo 搜索引擎 SEO，应该是一个非常好的消息！

这款软件类似于 WP 插件，是一个 WP 站外链的完美解决方案！只需要在网站后台轻松安装，就可以采用对搜索引擎有好的方式，让 WP 网站自动增加高权重外链。

最近本软件的官方网站，Automatic Backlink Creator 的价格只需要 37 美元，可以使用信用卡或者 paypal 付款，在国外卖的非常火爆！购买的同时，还赠送 MetaSnatcher 插件，这

款插件可以自动跟踪排在谷歌前几名的竞争对手网站核心关键，并且自动返回到软件，省去大量关键字分析的时间。**Spin Master Pro** 插件。这款插件相当于是 **WP** 离线伪原创和发布插件，安装这款插件之后，可以在自己的电脑进行内容伪原创并离线发布，节省大量时间。同时，本软件提供 60 天不满意退款保证。

[点击查看](#)

这款软件的开发者是一群 SEO 高手，结合谷歌和雅虎的外链算法，开发出了这款强大优秀的外链软件，在外链的 **PR**、**OBL**、**FLAG** 等方面考虑到了极致。并且通过这个系统可以产生稳定的、持续增加的高质量反链，例如.edu、.gov 等等网站的外链。

下载：[最经典的SEO链轮解决方案](#)