

# Wordpress插件开发系列

## 一、Wordpress插件机制浅析

Wordpress 通过插件机制，使一套基于基本的 CMS 系统的博客系统有了很大的扩展能力。

第一、Wordpress 的插件与文件系统的接口为 wp-content\plugins 文件夹。接口主文件可以直接放置在该文件夹下，如果牵涉的文件较多，也可以再建立一个文件夹，放在该文件夹下，该文件夹最好与接口主文件的名字相同：如主接口文件为 myplugin.php，则文件夹名字可以为 myplugin。

插件大体上可划分为两个部分：前面是注释，注释下是代码区。

注释部分大体如下示例：

```
/*
Plugin Name: MyCopyright
Plugin URI: http://www.why100000.com
Version: 0.1

Author: gdfos

Author URI: http://blog.why100000.com

Description: 把字符串"&lt;!--mycopyrighth--&gt;"替换为版权信息 : show copyright once there are
letters match "&lt;!--mycopyrighth--&gt;".
```

不要简单的看待以上的注释，不是随便写的，是固定的格式，可看做是插件的元信息（meta）。冒号前面的形如“Plugin Name”的信息都是有用的，作为插件的相关信息，会出现在系统设置等菜单下，作为本插件的标识信息。冒号后面的信息可以根据需要改写。冒号连接的一行信息类似于表中的一条记录，且不能换行。

再例如，Wordpress2.3.3 有个插件，其头部注释如下：

```
/*
Plugin Name: Hello Dolly
Plugin URI: http://wordpress.org/#
Description: This is not just a plugin, it symbolizes the hope and enthusiasm of an entire generation
summed up in two words sung most famously by Louis Armstrong: Hello, Dolly. When activated
you will randomly see a lyric from <cite>Hello, Dolly</cite> in the upper right of your admin
screen on every page.
```

```

Author: Matt Mullenweg
Version: 1.5
Author URI: http://photomatt.net/
*/

```

第二、代码部分就复杂了。不过一般应该有类似 `add_action('admin_footer', 'hello_dolly');` 的 PHP 代码，也就是至少需要执行 `add_action` 函数，当 Wordpress 执行 `admin_footer` 系统函数的时候，执行插件自定义的 `hello_dolly` 函数。`hello_dolly` 函数中就写有特点插件的逻辑。

大家可以去看插件 `hello.php`，很简单，功能是：当点击 Wordpress 后台管理的菜单项时，在页面右上角随机出现一些类似“毛主席语录”的“口号”（插件的 Description 里准确描述了其功能）。插件里有两处用了 `add_action` 函数：

```

add_action('admin_footer', 'hello_dolly');
add_action('admin_head', 'dolly_css');

```

分别执行了自定义函数 `hello_dolly` 和 `dolly_css`，用于显示信息和定义 CSS 样式。

第三、我模仿网络上的例子，写了一个最简化的 Wordpress 插件，全部代码如下：

```

<?php
/*
Plugin Name: MyCopyright
Plugin URI: http://www.why100000.com
Version: 0.1

Author: 网眼

Author URI: http://blog.why100000.com

Description: 把字符串"&lt;!--mycopyrigh--&gt;"替换为版权信息：show copyright once there are
letters match "&lt;!--mycopyrigh--&gt;".
you should config your copyright information in this file,with this verison.
*/

//WP 执行 the_content 函数时，调用插件自定义函数 why100000_showcopyright，对文章内容进行过滤
add_filter('the_content', 'why100000_showcopyright');
function why100000_showcopyright($content){
    $search = "<!--mycopyright-->";

    $replace = "<font color='red'>本文由网眼原创。转载请注明<a
    href='http://www.why100000.com'>http://www.why100000.com</a>";
}

```

功能是：把文章中的字符串“<!--mycopyrigh-->”替换为版权信息字符串。`add_filter` 其实是一个回调函数，用于在 Wordpress 的 `the_content` 函数执行的时候，处理一些事务。具体意义参见自定义 `why100000_showcopyright` 函数。

用法：保存为一个 PHP 文件，拷贝到插件目录下，或见一个文件夹，拷贝到文件夹下。这样，在插件管理界面里，就可以看见该插件的列表信息，点击“启用”使插件生效。然后，写博客的时候，在源代码里添加<!--mycopyright-->注释，保存博客后，浏览博客，会看到该注释被替换成了红色字体的字符串“本文由网眼原创。转载请注明原创作者：网眼（张庆）”，且包含一个链接。

这个简单的例子可以增加一些额外的功能，以使其更强大。

这仅仅是 WP 插件的应用层的知识。要进一步了解 Wordpress 的插件机制，需要阅读 Wordpress 的源代码。好在是开源的，应该从里面学到更多的经验和技巧。

## 二、Wordpress插件示例之一

这个 Wordpress 插件，除过能够在插件管理面板管理外（可以被开启和禁用），还能够在“设置”菜单下对插件进行配置，使插件的功能可以得到扩展。本插件可以实现对博客文章中的任意字符串进行替换。

以下代码中，why100000\_keyword 和 why100000\_replace 作为选项文本域标识变量，其值被 option.php 取到并保存到数据库的 wp\_options 表中。

get\_option("why100000\_keyword") 和 get\_option("why100000\_replace") 语句用于从 wp\_options 表中取出数据。

```
<?php
/*
Plugin Name: MyCopyright
Plugin URI: http://www.why100000.com
Version: 0.1

Author: 网眼(张庆-陕西西安-开开工作室-http://www.myopenit.cn)
Author URI: http://blog.why100000.com

Description: 把字符串"&lt;!--mycopyrigh--&gt;"替换为版权信息 : show copyright once there are
letters match "&lt;!--mycopyrigh--&gt;".
you should config your copyright information in this file,with this verison.
*/

//加到“设置”主菜单下
add_action('admin_menu', 'why100000_add_options_page');
function why100000_add_options_page(){
    add_options_page('MyCopyright', 'MyCopyright-Config', 8, 'mycopyright.php',
'why100000_myCopyright_mainpage');
}
function why100000_myCopyright_mainpage(){
?>
```

```

MyCopyright is active.<br><br>

<h2>MyCopyright Config ( 版权插件配置 ) </h2>

<div id="msgall">
    <fieldset class="options">

        <legend>keyword ( 原字符串 ) :</legend>

        <p>
            <textarea style="width: 80%;" rows="5"
cols="40" name="why100000_keyword"><?php echo
get_option("why100000_keyword");?></textarea>
        </p>
    </fieldset>
    <fieldset class="options">

        <legend>replacement ( 代替的字符串 ) :</legend>

        <p>
            <textarea style="width: 80%;" rows="5"
cols="40" name="why100000_replace"><?php echo
get_option("why100000_replace");?></textarea>
        </p>
    </fieldset>
</div>
<p class="submit">
    <input type="submit" value="<?php _e('Update Options
&raquo;') ?>" name="update_message"/>
</p>
<input type="hidden" name="action" value="update">
<input type="hidden" name="page_options"
value="why100000_keyword,why100000_replace">
</div>
</form>
<?php
}
add_action('activate_mycopyright/mycopyright.php', 'why100000_copyright_install');

//插件第一次被“启用”时执行，作为初始值保存到表 wp_options 中

function why100000_copyright_install(){
    add_option("why100000_keyword", "<!--mycopyright-->");
    add_option("why100000_replace", "我的版本");
}

//WP 执行 the_content 函数时，调用插件自定义函数 why100000_showcopyright，对文章内容进行过滤

add_filter('the_content', 'why100000_showcopyright');
```

```
}
?>
```

### 三、Wordpress插件示例之二-对示例一的重构

示例一已经完全可以工作了，这个在 Wordpress2.3.3 和 2.9.1 版本中已经得到证实。但示例一有几个缺陷。

由于 Wordpress 的插件众多，全世界有很多程序员包括我们中国的程序员，都在为它开发插件，这些插件往往被一起安装在一套 Wordpress 系统中协调工作，这未免会产生冲突的情况。

为了隔离各个插件的代码，我们需要想一些办法。而把插件代码封装在一个类结构中是个行之有效的方法之一。正确应用一些检测函数，可以保证只要插件类名不冲突，类所包含的成员函数就不会与其它插件冲突。这大大增强了插件代码的健壮性。而且插件类名是否冲突也作为代码容错检测的一部分，使系统不至于出现严重的崩溃。

请参看下面的一个完整的 Wordpress 插件代码，是参考以上考虑，对示例一的代码的完全重构。

```
<?php
/*
Plugin Name: MyCopyright
Plugin URI: http://www.why100000.com
Version: 0.1

Author: 网眼(张庆-陕西西安-开开工作室-http://www.myopenit.cn)
Author URI: http://blog.why100000.com

Description: 把字符串"&lt;!-mycopyrigh--&gt;"替换为版权信息 : show copyright once there are letters
match "&lt;!-mycopyrigh--&gt;".
you should config your copyright information in this file,with this verison.
*/
if (!class_exists("why100000_Plugin")){
    class why100000_Plugin{

        //构造函数

        function why100000_Plugin(){

            //加到"设置"主菜单下

            function why100000_add_options_page(){
                add_options_page('MyCopyright', 'MyCopyright-Config', 8, 'mycopyright.php',
array(&$this, 'why100000_myCopyright_mainpage'));
            }

            function why100000_myCopyright_mainpage(){
```

```

MyCopyright is active.<br><br>

<h2>MyCopyright Config ( 版权插件配置 ) </h2>

<div id="msgall">
    <fieldset class="options">

        <legend>keyword ( 原字符串 ):</legend>

        <p>
            <textarea style="width: 80%;" rows="5" cols="40"
name="why100000_keyword"><?php echo
get_option("why100000_keyword");?></textarea>
        </p>
    </fieldset>
    <fieldset class="options">

        <legend>replacement ( 代替的字符串 ):</legend>

        <p>
            <textarea style="width: 80%;" rows="5" cols="40"
name="why100000_replace"><?php echo get_option("why100000_replace");?></textarea>
        </p>
    </fieldset>
</div>
<p class="submit"><input type="submit" value="<?php
_e('Update Options &raquo;') ?>" name="update_message"/></p>
<input type="hidden" name="action" value="update">
<input type="hidden" name="page_options"
value="why100000_keyword,why100000_replace">
</div>
</form>
<?php
}

//插件第一次被“启用”时执行，作为初始值保存到表 wp_options 中

function why100000_copyright_install(){
    add_option("why100000_keyword", "<!--mycopyright-->");
    add_option("why100000_replace", "我的版本");
}

//WP 执行 the_content 函数时，调用该函数，对文章内容进行过滤

function why100000_showcopyright($content){
    $search = get_option("why100000_keyword"); // "<!--mycopyright-->";

    $replace= get_option("why100000_replace"); //代替的字符串;

    $content= str_replace($search, $replace, $content);

```

```

}
if (isset($MyPlugin)){
    add_action('activate_mycopyright/mycopyright.php', array(&$MyPlugin,
'why100000_copyright_install'));
    add_action('admin_menu', array(&$MyPlugin, 'why100000_add_options_page'));
    add_filter('the_content', array(&$MyPlugin, 'why100000_showcopyright'));
}
?>

```

于是，我们归纳出一个相对完善的 Wordpress 插件代码书写构架：

```

<?php

/* 插件注释部分..... */

/* 插件代码部分： */

/* 插件类定义：*/

if (!class_exists("插件类名")){
class 插件类名
{

    /* 全部插件函数..... */

}
}

/* 实例化插件类： */

if (class_exists("插件类名")){

    $MyPlugin = new 插件类名();

}

```

## 四、在 Wordpress 插件中使用样式表

要让插件功能非凡，难免需要 JavaScript 和 CSS 的参与。本节将举例在 Wordpress 插件中插入 CSS 代码。下一节将试图添加 JS 脚本。

### 1、插件的文件夹结构

插件主文件可以直接放置在 plugin 文件夹下，也可以建立一个文件夹，放置在该文件夹下。由于插件的相关文件可能较多，所以为了清晰的管理插件，建议把插件文件部署于自己的文件夹下。而且 JS、CSS、图片文件也按文件夹分类放置，通常的文件夹可以取名为 js、css 和 images。

一个示例的文件夹结构如下：

```
├─ ~\plugins
│   ├─ css          (样式表文件)
│   ├─ images       (图片文件)
│   ├─ js           (脚本文件)
│   ├─ php          (相关 PHP 脚本)
│   └─ myplugin.php (插件主文件)
```

## 2、示例插件实现的功能

在文章末尾加一段文字：<span id="myalert">弹出警告对话框</span>，用样式表中的样式代码改变其样式。文章末尾添加的文字，可以在插件设置面板里进行定制修改，比较灵活。

## 3、完整代码及其解释

要往文章中加入 CSS 代码，一种是直接打开 Wordpress 的 themes 有关模板文件直接添加，但仅作用于这个模板页面。这样做可以系统的运行效率比较高，但不够灵活。而使用插件来实现，实现的结果对所有的 themes 有效。

在 Wordpress 插件中使用样式表，主要使用了 add\_head 这个 API 事件。

首先编辑 css.css 文件，内容如下：

```
#myalert{
    background-color: #006;
    color: #fff;
    cursor: hand;
}
```

保存于 css 文件夹下。该段样式表可以按需要定制，但要确保不能与其它样式冲突，否则得不到预期的结果。该段样式表代码对 id="myalert" 的标签进行修饰。

然后定义插件类的成员函数 why100000\_addHeaderCode()，输出引用样式表的标签。注意 css 文件的路径。函数完整代码如下：

```
function why100000_addHeaderCode(){
    echo '<link type="text/css" rel="stylesheet" href="'. get_bloginfo('wpurl') .
        '/wp-content/plugins/test-css-js/css/css.css">';
}
```

然后调用以下代码：

```
add_action('wp_head', array(&$MyPlugin_testcssjs, 'why100000_addHeaderCode'), 1);
```

使生成文章页面 head 部分时，添加<link ...>对样式表的调用。



本例的完整代码如下，关键代码有注释：

```
<?php
/*
    Plugin Name: Test_CSS_JS
    Plugin URI: http://www.why100000.com
    Version: 0.1

    Author: 网眼(张庆-陕西西安-开开工作室-http://www.myopenit.cn)
    Author URI: http://blog.why100000.com

    Description: 测试样式表和 JS 脚本：在文章末尾加一段文字，改变样式，点击弹出对话框。
*/
if (!class_exists("why100000_Test_css_js")){
    class why100000_Test_css_js{

        //构造函数

        function why100000_Test_css_js(){}

        //加到“设置”主菜单下

        function why100000_add_options_page(){
            add_options_page('Test_CSS_JS', 'Test_CSS_JS-CFG', 8,
basename(__FILE__), array(&$this, 'why100000_Test_CSS_JS_mainpage'));
        }
        function why100000_Test_CSS_JS_mainpage(){

?>

        <form name="formamt" method="post" action="options.php">
        <?php wp_nonce_field('update-options') ?>
        <div class="wrap">
            Test_CSS_JS is active.<br><br>

            <h2>Test_CSS_JS Config ( 插件配置 ) </h2>

            <div id="msgall">
                <fieldset class="options">

                    <legend>Word ( 字符串 ) :</legend>

                    <p><textarea style="width: 80%;" rows="5" cols="40"
name="why100000_word"><?php echo
get_option("why100000_word");?></textarea></p>
                </fieldset>

            </div>

            <p class="submit"><input type="submit" value="<?php
_e('Update Options &raquo;') ?>" name="update_message"/></p>
```

## 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](#)