

学习Hello Dolly插件

Alex / 2017-10-15 / free_learner@163.com / learning-archive.org

更新于2023-06-04，主要是文字排版上的更新，内容基本保持不变。

一、背景

通过安装插件，可以拓展WordPress网站的功能。Hello Dolly插件是WordPress默认安装的两个插件之一（另一个是Akismet）。Hello Dolly插件的功能是在WordPress后台的每个页面随机显示Hello, Dolly的一句歌词。学习Hello Dolly可以了解如何写一个简单的插件。

二、文件夹和文件

在wp-content/plugins文件夹下，不难找到名为hello-dolly的文件夹，该文件夹中包含两个文件，分别名为hello.php和readme.txt。其中hello.php即为实现插件功能的文件。注意这个PHP文件的名称可以和插件名称不同，但必须是独一无二的，否则会和其他插件产生冲突。

三、头信息

打开hello.php文件，顶部出现的注释内容即为头信息（header），其中Plugin Name一行是必须的，而且插件名称也必须是独一无二的。

```
/*
Plugin Name: Hello Dolly
Plugin URI: https://wordpress.org/plugins/hello-dolly/
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: 1.6
Author URI: http://ma.tt/
Text Domain: hello-dolly
*/
```

四、自定义函数

hello.php中有三个自定义函数，分别是 `hello_dolly_get_lyric` ， `hello_dolly` 和 `dolly_css` 。 `hello_dolly_get_lyric` 的作用是随机抽取一句歌词， `hello_dolly` 的作用是用HTML标记抽取的歌词， `dolly_css` 的作用是设置抽取的歌词CSS样式。

五、Action Hook

WordPress中有两种Hook（Hook本意是钩子，我理解为就是将自定义的代码添加到WordPress的核心代码中），分别是filter和action。hello.php中使用了两个action hook，分别是 `admin_notices` 和 `admin_head` ，代码如下：

```
add_action( 'admin_notices', 'hello_dolly' );
add_action( 'admin_head', 'dolly_css' );
```

`admin_notices` 的作用是在后台页面顶部显示信息（这里即显示抽取的歌词）， `admin_head` 的作用是向后台页面的 `<head>` 元素中添加信息（这里即添加CSS）。

六、在主页顶部显示一句励志的话

这里我简单修改一下Hello Dolly来实现在网站主页顶部随机显示一句励志的话的功能。代码如下：

```

<?php
/*
Plugin Name: Chicken Soup
Description: Display some encouraging words in the header of home page
Author: Alex
Version: 1.0
*/

function chicken_soup_get_words() {
    // Example chicken soup words from Steve Jobs
    $words = 'Stay Hungry, Stay foolish.
If you haven\'t found it yet, keep looking. Don\'t settle.
Your time is limited, so don\'t waste it living someone else\'s life';

    // Here we split it into lines
    $words = explode( "\n", $words );

    // And then randomly choose a line
    return wptexturize( $words[ mt_rand( 0, count( $words ) - 1 ) ] );
}

// This just echoes the chosen line, we'll position it later
function chicken_soup() {
    if (is_home()){
        $chosen = chicken_soup_get_words();
        echo "<p id='chicken-soup'>$chosen</p>";
    }
}

// Now we set that function up to execute when the wp_footer action is called
add_action( 'wp_footer', 'chicken_soup' );

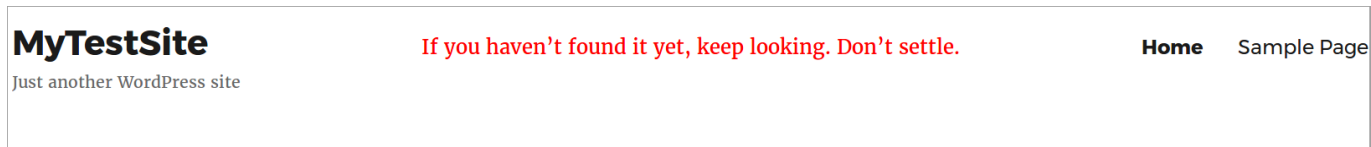
// We need some CSS to position the paragraph
function chicken_soup_css() {
    echo "
<style type='text/css'>
#chicken-soup {
    position: absolute;
    top: 140px;
    left: 50%;
    transform: translateX(-50%);
    color: red;
}
</style>
";
}

add_action( 'wp_head', 'chicken_soup_css' );

?>

```

最终效果如下图：



七、参考

- https://codex.wordpress.org/Writing_a_Plugin
- <https://developer.wordpress.org/plugins/>