

Pasos Para Crear Un Widget Para WordPress



Como hemos visto en anteriores artículos, sobre como [personalizar la pantalla de login de WordPress](#), ó de como [mostrar iconos en el menú de nuestro sitio](#) aun cuando nuestra plantilla no soporte esto, entre otras cosas. Hemos visto la flexibilidad que tenemos al usar WordPress, ademas de las grandes ventajas. En este artículo vamos a hablar de una característica importante donde converge la flexibilidad y ventaja que podemos tener con WordPress, los Widgets. Los **Widgets de WordPress** te dan la posibilidad de editar una cierta parte de la plantilla de este CMS, por lo general en la barra lateral. Usando sus funciones de arrastrar y soltar, es bastante fácil para los principiantes añadir, personalizar o eliminar un widget.



Aunque existe una gran cantidad de **plugins de WordPress** disponibles en la Web, la mejor manera tener el mejor resultado que deseas es programarlo. **Programar un Widget para WordPress**? Este redactor esta loco! Seguramente lo pensarás, al igual que como tu empece como novato y sentí ese miedo de

tocar el código de algo que parecía bastante complejo, y lo es, pero lo bonito de WordPress es que en cada actualización mejoran la disposición para que el usuario (tu, el que está detrás de ti, tu vecino, y todo aquel que use WordPress) pueda modificar lo que desee ;)

Crear un widget para WordPress es como la codificación de un plug-in, pero, mucho más fácil y simple. Aunque los **Widgets de WordPress** brindan funciones bastante sencillas, sin duda alguna proporcionan ciertas opciones útiles, como: llenar formularios, incluir o excluir cierta información de datos, entre otras cosas. Para esto haremos uso del archivo `functions.php`. Para el ejemplo práctico, usaremos un **Widget para mostrar un banner publicitario** en nuestro sitio.

Necesitas Esto Para Continuar El Tutorial

- WordPress Instalado
- Conocimientos básicos sobre los **Widgets de WordPress**. Te invitamos a [conocer su documentación](#).
- Conocimientos básicos sobre la **API de WordPress**. Te invitamos a [conocer su documentación](#).
- Algo de paciencia ;)

Paso 1: Crear La Clase Y Funciones

Los **Widgets de WordPress** son clases y estas son fáciles de duplicar. Una clase ofrece el modelo que puede ser usado varias veces sin problemas. La clase base tiene las funciones que deben ser heredadas para hacer el trabajo de widgets. Para empezar, se debe crear primero la clase.

Cree un nuevo archivo php en la **carpeta plugins** de la carpeta de instalación de WordPress y nombrelo **ads_widget.php**. A continuación, copie el código de la clase siguiente.

[php]

```
<!--?php class Ads_Widgets extends WP_Widget { function
Ads_Widgets() { // procesos actuales del widget } function
form($instance) { // mostrar las opciones desde el dashboard
del admin } function update($new_instance, $old_instance) { //
opciones de los procesos que pueden ser guardado } function
widget($args, $instance) { // muestra el contenido del widget
} } //llamar el registro del widget add_action(
'widgets_init', create_function(», 'return
register_widget("Ads_Widgets");') ); ?-->
```

[/php]

Paso 2: Procesos Del Widget

Dentro de las funciones `Ads_Widgets`, copie el código de abajo. Estos códigos inicializar el widget y se utiliza en toda la función para especificar la instancia actual del widget.

[php]

```
<!--?php function Ads_Widgets() { $widget_ops =  
array('classname' => 'Ads_Widgets', 'description' =>  
'Displays Ads' );  
$this->WP_Widget('Ads_Widgets', 'Advertisement Widget',  
$widget_ops);  
}  
?>
```

[/php]

Paso 3: La

función Form

A continuación, programar el formulario para mostrar las opciones que se verán en el panel de administración del administrador. Copie el código de abajo en el interior de la **función form**.

```
[php]
```

```
function form($instance)
{
$instance = wp_parse_args( (array) $instance, array( 'title'
=> » ) );
$title = $instance['title'];
$message = esc_attr($instance['message']);
$link = esc_attr($instance['link']);
?>

<label for="<?php echo $this->get_field_id('title');
?>">Title: <input class="widefat" id="<?php echo
$this->get_field_id('title'); ?>" type="text" name="<?php echo
$this->get_field_name('title'); ?>" value="<?php echo
attribute_escape($title); ?>" /></label>

<label for="<?php echo $this->get_field_id('message');
?>"><!--?php _e('Advertisement Banner'); ?--></label>
<textarea class="widefat" id="<?php echo
$this->get_field_id('message'); ?>" cols="50" name="<?php echo
$this->get_field_name('message'); ?>" rows="4"><?php echo
($message); ?> </textarea>

<label for="<?php echo $this->get_field_id('link');
?>"><!--?php _e('Ads Link'); ?--></label>
<input class="widefat" id="<?php echo
$this->get_field_id('link'); ?>" type="text" name="<?php echo
$this->get_field_name('link'); ?>" value="<?php echo $link;
```

```
?>" />
```

```
<!--?php } ?-->
```

```
[/php]
```

Paso 4: Actualización De La Función

A continuación, trabajar en la función de actualización. Esta función se llevará a la configuración de usuario y los guardara en la base de datos.

```
[php]
```

```
<!--?php function update($new_instance, $old_instance) {  
$instance = $old_instance; $instance['title'] =  
$new_instance['title']; $instance['message'] =  
$new_instance['message']; $instance['link'] =  
$new_instance['link']; return $instance; } ?-->
```

```
[/php]
```

Paso 5: La Función Widget

Esta función mostrará el contenido del widget en la página principal. El código contendrá argumentos para la plantilla, que puede ser un título, una descripción u otros valores. Copie el código de abajo en el interior de la función widget.

```
[php]
```

```
function widget($args, $instance) {
extract( $args );
$title = apply_filters('widget_title', $instance['title']);
$message = $instance['message'];
$link = $instance['link'];
?>
<!--?php echo $before_widget; ?-->
<!--?php if ( $title ) echo $before_title . $title .
$after_title; ?--></pre>
<ul>
<li><!--?php echo '<a href="'. $link.'" target="_blank"-->"; ?></li>
</ul>
<pre>
<!--?php echo $after_widget; ?-->
<!--?php <br ?--> }
}
```

```
[/php]
```

Paso 6: Agregar Hook Acción y Devolución Aplicaciones

Por último, cargue el widget por la función `widget_init` utilizando un hook de acción. Esto también registrará el nombre del widget para que funcione tanto en el front y back-end. Coloque el código debajo de la clase que se creó anteriormente.

```
[php]add_action( 'widgets_init', create_function(», 'return register_widget("Ads_Widgets");') );[/php]
```

Paso 7:

Activar El Plugin Del Widget

Ahora que el widget está listo, solo faltara activar el plugin. En el panel de administración, vaya al panel Plugins y luego haga clic en el enlace de activación del complemento creado.

<input type="checkbox"/>	Plugin	Description
<input type="checkbox"/>	Advertisement Widget	An Advertisement Banner Widget
	Activate Edit Delete	

A continuación, para mostrar los anuncios, sólo tiene que suministrar los siguientes campos.

Advertisement Widget: Adverti...

Title:
Advertisement

Advertisement Banner
<http://localhost/wpwidgets/wp-content/uploads/2014/06/banner.jpg>

Ads Link
<http://www.1stwebdesigner.com/bl>

[Delete](#) | [Close](#) [Save](#)

Si visitas el sitio podrás ver tu anuncio, en nuestro caso veremos el siguiente ;)

-10% **Hosting Compartido Linux** Desde: **\$70,000/Año**
Alojamiento web fiable, fácil y asequible.

cPanel **Fantastico DeLuxe**   **Joomla!**  **MySQL**  **WORDPRESS** 

Finalmente

Eso es todo! Acabas de crear tu primer **widget de anuncios para tu sitio WordPress**. Seguramente pensarás que esto no es nada, pero no es así. Esta es la base para que puedas mejorarlo y modificarlo según tus necesidades, digamos, mejorar el uso con jQuery, hacerlo mas dinámico. Incluir [anuncios animados con CSS](#), pero seguramente encontrarás alguna otra mejora para este Widget ;)