


10 Funciones De WordPress Que Probablemente No Conozcas

El popular [CMS WordPress](#) está lleno de grandes y útiles  funciones para los desarrolladores. Se pueden usar para realizar cualquier tipo de operación, ya sea el envío de mail, la mejora de la seguridad, conexión con las diferentes redes sociales, o algunas funciones útiles de forma rápida. Si lo que quieres es ahorrar líneas de código, y no volver a inventar la rueda, este artículo te será de gran interés.

Sin embargo, hay un buen número de funciones de WordPress que parecen no se usadas. A continuación, te compartiremos algunas de las **funciones en WordPress** que pocos desarrolladores conocen, y que de seguro te serán de gran utilidad.

1. Antispambot()

`<code> antispambot() </ code>` reemplaza los caracteres de una cadena con las entidades HTML, lo cual es una forma ingeniosa de ocultar las direcciones de correo electrónico de los malignos scrapers que andan por la web.

```
[php]
```

```
$email = 'mymail@mail.com';  
echo 'You can contact me at ' . antispambot( $email ) . ' any  
time'.
```

```
[/php]
```

Si bien esta es una función útil, también es un ejemplo de por qué algunas personas critican WordPress, ya que esta función ha sido horriblemente mal nombrada. Tan solo con leer el nombre, no sabes que es lo que puede llegar a hacer la función.

2. Human_time_diff()

Esta función, da como salida la diferencia entre dos marcas de tiempo, además no es una función nueva, ya que ha estado desde la versión 1.5 (que es el 17 de febrero 2005!). El siguiente fragmento de código, cuánto tiempo hace que se publicó un artículo. Utiliza la fecha de publicación del artículo como primer argumento y la fecha actual como segundo argumento.

```
[php]echo 'This post was published ' . human_time_diff(
get_the_time( 'U' ), current_time( 'timestamp' ) ) . '
ago'; [/php]
```

3. Get_post_meta()

Esta es una función bien utilizada, sin embargo, la forma en que funciona, no es la que se debería de usar. En primer lugar, al omitir el segundo y tercer parámetro, con los cuales puedes obtener todos los metadatos de un artículo.

```
[php]$all_meta = get_post_meta( 14 ); [/php]
```

Incluso si sólo obtiene los datos de una sola palabra principal todo postmeta se obtienen de todos modos. La razón es bastante lógico. Los metadatos se utilizan en múltiples lugares. Si `get_post_meta()` Consulta la base cada vez que se utilizó acabaríamos con demasiadas consultas. En cambio, si se obtiene los metadatos, estos se almacenan en caché y utiliza los valores almacenados en caché en todas las recuperaciones de los metadatos.

4. Wp_upload_bits ()

Esta función se usa para la **carga directa de archivos**. Si bien no se mueve el archivo a la carpeta de archivos y se agrega a

la sección multimedia de WordPress, es bastante conveniente y siempre que se pueda para el resto usar la función `wp_insert_attachment ()` .

```
[php]$upload = wp_upload_bits( $file['name'], null,
file_get_contents( $file['tmp_name'] ) );[/php]
```

Alguna explicación está a la mano para esto: el primer parámetro es el nombre del archivo. El segundo se deprecia por lo que debe establecerse a `null`. El tercer parámetro es el contenido real del archivo.

5. `Get_post_field()`

No necesitaras crear un ciclo o una consulta para obtener el numero de comentarios de un articulo, todo lo que necesitas es la función `<code> get_post_field () </ code>`. Esta función obtiene el valor de un campo para un solo articulo en la base de datos. Se usa justo de la siguiente forma.

```
[php]Este articulo tiene <?php echo get_post_field(
'comment_count', 4124 ) ?> comentarios. [/php]
```

6. `Wpautop()`

Esta función es similar a la nativa de PHP `<code> nl2br </ code>`, pero en lugar de crear nuevas líneas, envuelve su contenido en párrafos. Esto es útil si tienes un cuadro de texto y deseas asegurarte de que cuando los usuarios crean párrafos con saltos de línea dobles, siguen siendo visibles en el front-end también.

```
[php]
```

```
<h2>¿Qué dicen nuestros usuarios?</h2>
<?php echo wpautop( $user_comment ) ?>
```

```
[/php]
```

7. **Wp_is_mobile ()**

Esta función bien detecta cuando un usuario se encuentra desde un dispositivo móvil y te permite mostrar contenido especial para este usuario. Como se trata de una etiqueta condicional devuelve verdadero o falso dependiendo del escenario.

```
[php]
```

```
<?php if( wp_is_mobile() ) : ?>
```

Visite nuestro sitio web desde tu PC para una disfrutar de una mejor experiencia

```
<?php endif ?>
```

```
[/php]
```

8. **DOING_AJAX**

Si bien esto es una constante, no una función (lo siento), no obstante es útil. Puede distinguir entre llamadas AJAX y procesos regulares comprobando si está definida y es verdadera. Esto puede ser útil si estás usando una función común para una tarea que a veces se desencadena por AJAX.

```
[php]
```

```
if (defined('DOING_AJAX') && DOING_AJAX) {  
    echo 'done';  
    die()  
}  
else {  
}
```

```
[/php]
```

9. Wp_redirect()

Esta función muestra otra función ordenada: `wp_redirect()`. Esto se debe utilizar en lugar de la función nativa de PHP `header()`. La función de redirección de WordPress permite configurar una **dirección URL para redirigir**, y también establecer un mensaje de estado, ideal para el manejo de redirecciones permanentes.

```
[php]
```

```
// Para una URL que ya no está en uso
wp_redirect( 'http://website.com/nueva-url/', 301 );
```

```
[/php]
```

10. Paginate_links()

Por defecto, WordPress muestra / anteriores y siguientes en los enlaces al final de su lista de correos. En realidad, esto se puede hacer con un poco de trabajo utilizando la función `paginate_links()`. Cuenta con un buen número de parámetros, así que recomiendo echar un vistazo a su [documentación](#).

El siguiente ejemplo del código, muestra cómo se puede añadir a un bucle por defecto:

```
[php]
```

```
global $wp_query;
$big = 999999999; // need an unlikely integer
echo paginate_links( array(
    'base' => str_replace( $big, '%#%', esc_url( get_pagenum_link(
    $big ) ) ),
    'format' => '?paged=%#%',
    'current' => max( 1, get_query_var('paged') ),
```

```
'total' => $wp_query->max_num_pages  
) );  
  
[/php]
```

Finalmente

Estas son sólo algunas de las funciones que son poco conocidas por la mayoría de **desarrolladores y programadores en WordPress**. Si tienes una función poco conocida y que sea de gran utilidad, tienes la libertad de compartirla en un comentario ;)