

Mejorar El Uso Del Editor TinyMCE En WordPress

Todos los que usan *WordPress* está familiarizados con el **editor TinyMCE**. Es el editor que se usa en este famoso CMS para crear o editar el contenido, es aquel que te ofrece una serie de botones que son de gran utilidad al momento de dar un excelente formato a tus publicaciones.



El editor se basa en un ***sistema de Javascript independiente de la plataforma***, llamada TinyMCE que se utiliza en una serie de proyectos en la Web. Tiene una gran API que WordPress le permite aprovechar para crear tus propios botones y añadirlos de forma fácil en WordPress. Sin duda, es una [parte fundamental en WordPress](#), pero podrías mejorar o añadir sus funciones, y esto lo aprenderás en este artículo.

Añadir Botones

[WordPress](#) usa algunas de las opciones disponibles en TinyMCE para desactivar algunos botones (como superíndice, subíndice y reglas horizontales) con el fin de limpiar la interfaz. Ellos pueden ser agregados nuevamente sin mucho esfuerzo.

El primer paso es crear un plugin. Conoce como puedes [crear un plugin en WordPress](#) en el sitio web oficial. En pocas palabras, tienes que crear una carpeta llamada 'my-mce-plugin' en la carpeta wp-content/plugins. Luego crear un archivo con el mismo nombre, con una extensión de PHP: my-mce-plugin.php, y agregar lo siguiente al archivo creado:

```
[php]
```

```
<?php
```

```
/**
 *Plugin Name: My TinyMCE plugin
 *Description: A plugin for adding custom functionality to the
 WordPress TinyMCE plugin.
 */

[/php]
```

Una vez hecho esto, deberías ser capaz de seleccionar este plugin en WordPress y activarlo. Todo el código a partir de ahora se puede añadir dentro de este archivo. Así que, a continuación está el código necesario para activar los botones ocultos que se han mencionado al inicio del artículo:

```
[php]

add_filter( 'mce_buttons_2', 'my_tinymce_buttons' );
function my_tinymce_buttons( $buttons ) {
$buttons[] = 'superscript';
$buttons[] = 'subscript';
$buttons[] = 'hr';
return $buttons;
}

[/php]
```



Existen otros botones que están ocultos por defecto, puede mirar la [documentación de TinyMCE](#) para conocer más de estos.

Creando Nuestros Propia Botones

Además de habilitar aquellos botones ocultos por defecto, también podemos crear y añadir nuestros propios botones a este editor. Por ejemplo, si deseamos añadir etiquetas para alguna

línea de código, debemos de hacer lo siguiente:

```
[php]<pre><code>$variable = 'value'</code></pre> [/php]
```

Para esto podemos crear un botón, y esto solo se hará en tres pasos. Tendrás que añadir un botón, cargar un archivo javascript y escribir el contenido del archivo de Javascript. Empezamos con agregar el botón y cargar el archivo javascript:

```
[php]

add_filter( 'mce_buttons', 'pre_code_add_button' );
function pre_code_add_button( $buttons ) {
    $buttons[] = 'pre_code_button';
    return $buttons;
}
add_filter( 'mce_external_plugins', 'pre_code_add_javascript'
);
function pre_code_add_javascript( $plugin_array ) {
    $plugin_array['pre_code_button'] =
    get_template_directory_uri() . '/tinymce-plugin.js';
    return $plugin_array;
}

[/php]
```

El siguiente paso es escribir algo de Javascript para implementar las funciones. Esto es lo que utiliza para comenzar y finalizar las etiquetas <pre> y <code> a la vez.

```
[php]

(function() {
    tinymce.PluginManager.add('pre_code_button', function( editor,
    url ) {
        editor.addButton( 'pre_code_button', {
            text: 'Prism',
            icon: false,
            onclick: function() {
```

```
var selected = tinyMCE.activeEditor.selection.getContent();
var content = '<pre><code>' + selected + '</code></pre>';
editor.insertContent( content + "\n" );
}
});
});
})();

[/php]
```

El **método onclick** se usa para cuando el usuario hace click con el mouse. El texto seleccionado puede ser tomado usando **tinyMCE.activeEditor.selection.getContent()**. Entonces, se envuelve los elementos a su alrededor y la inserta, reemplazando el contenido resaltado con el nuevo elemento. También se ha añadido una nueva línea para que pueda comenzar fácilmente la escritura.

También es necesario adjuntar una hoja de estilos en el plugin y luego añadir un estilo sencillo para mostrar el icono. En primer lugar, vamos a añadir a nuestro estilo de WordPress:

```
[php]

add_action( 'admin_enqueue_scripts', 'pre_code_styles' );
function pre_code_styles() {
wp_enqueue_style( 'pre_code_button', plugins_url(
'/style.css', __FILE__ ) );
}

[/php]
```

Es necesario volver al código Javascript y junto al icono en la función addButton, reemplace el valor por «false» con una clase en la que le gustaría usar en el botón (si usas iconos de dashicons). Ahora debes **crear el archivo style.css** en tu directorio de plugins y añadir el siguiente CSS:

```
[css]
```

```
i.mce-i-pre_code_button:before {  
  
content: "\f475";  
}  
  
[/css]
```



Tenga en cuenta que el botón recibirá la clase en `mce-i-[su clase aquí]`, en la cual se puede utilizar para apuntar y añadir estilos. Especifique la fuente como `dashicons` y el contenido utilizando el valor Unicode de antes.

Conclusión

El **editor TinyMCE** es una manera fácil de crear y editar el contenido en WordPress, permitiendo la facilidad de buscar e implementar contenido, agilizando y mejorando el proceso a la hora de añadir contenido de calidad al sitio. Sin duda, al añadir nuevos botones, se puede mejorar en gran manera las funciones de este editor, complementando las nuevas funciones que ha traído la versión 4.1 de WordPress.