Estándares De Programación Para WordPress [Guía]

WordPress ha sido un popular CMS, no solo por la facilidad que brinda a la hora de compartir contenido, sino también por las ventajas a la hora de **modificar y personalizarlo**. Aunque es bien cierto que los programadores de WordPress no cuentan con un estandar, el cual deben seguir al pie de la letra como sucede en Laravel o algún otro Framework de PHP, existen algunas buenas practicas que podemos definir como un estándar a aplicar.

Las **mejores prácticas en WordPress** son difíciles de hacer cumplir, simplemente porque los desarrollos que emplean un mal código pueden funcionar igual de bien (en la superficie), que el código que se ha realizado con buenas practicas.

Siguiendo algunos **estándares en la programación para WordPress**, se puede aprender un poco sobre el desarrollo de WordPress, crear productos más compatibles con WordPress. Con esto desarrollarás código de calidad, lo cual será **beneficioso para la comunidad de WordPress**.

Estándares de WordPress Codificación

En este momento WordPress tiene cuatro guías, una para cada tecnología web que se utiliza: PHP, HTML, Javascript y CSS. Forman parte de un conjunto más amplio de conocimientos, el Manual Core. A continuación se destaca lo mas importante, en lo que **los usuarios fallan al momento de programan para Wordpress**.

PHP

PHP es el **lenguaje principal de WordPress**, y es un lenguaje bastante conocido, lo cual es propicio para ser usado en diferentes proyectos web.

Estilos En Bloques

Llaves de partida inicio deben colocarse al final de las líneas. Si existe una nueva condicional, esta debe añadirse justo después del cierre de la anterior condicional, justo de la siguiente manera:

```
[php]if ( condition ) {
// Do Something
} elseif ( condition ) {
// Do Something
} else {
// Do Something
}  [/php]
```

Uso De Espacio

Reducir el tamaño de los archivos de codigo es algo que beneficia cuestiones de velocidad, aunque esto es util en ocasiones, hay otros escenarios en los que el uso de espacio es bastante util para la lectura del codigo:

```
[php]

function my_function( $complete_array = null, $key_1 = 4, $key_2 = ‘bar’ ) {
if ( null == $complete_array ) {
$final_array = $complete_array;
} else {
$key_1 = (integer) $key_1;
$final_array[0] = ‘this’;
$final_array[ $key_1 ] = ‘is’;
```

```
$final_array[ $key_2 ] = ‘an’;
$final_array[‘last’] = ‘example’;
}
return $final_array;
}[/php]
```

Convenciones De Nomenclatura

Éste puede ser difícil acostumbrarse, especialmente si vienes de otros lenguajes de programación. En pocas palabras:

- Los nombres de variables deben estar en minúsculas, palabras separadas por guiones bajos
- Los nombres de clase deben usar palabras en mayúsculas separadas por guiones bajos. Los acrónimos deben estar en mayúsculas
- Constantes deben estar en mayúsculas, separadas por guiones
- Los nombres de archivo deben estar en minúsculas, separados con guiones

Condiciones Yoda

Escribir condiciones al revés te podria ayudar a prevenir errores de análisis. Puede ser un poco raro, pero es mejor para el código.

```
[php]if ( ‘alex’ === $name ) {
echo ‘Podría escribir un articulo’;
}[/php]
```

HTML

HTML no tiene que bastantes reglas asociadas a él, pude llegar a mucho para hacer las cosas más modular. Sólo hay cinco reglas que usted necesita saber la hora de escribir HTML:

- El código debe validar contra el validador del W3C.

- Las etiquetas HTML de cierre automático deben tener exactamente un espacio antes de la barra diagonal (es una **especificación del W3C**, no un motivo favorito de WordPress)
- Los atributos y etiquetas deben estar en minúsculas. La única excepción es cuando los valores de atributos están destinados para el consumo humano, en cuyo caso deben ser escritos de forma natural.
- Todos los atributos deben tener un valor y deben ser citado (<input disabled> No es la escritura correcta)
- La sangría debe lograrse mediante pestañas y debe seguir la estructura lógica.

CSS

CSS es otro lenguaje de programación bastante flexible para escribir, por lo que hay un montón de trabajo por hacer aquí. Aun así, las buenas practicas son bastante fácil al momento de codificar.

Selectores

Los selectores deben ser tan cualificados como sea necesario, ser humanamente legible, estar en minúsculas con palabras separadas con guiones y selectores de atributos deben utilizar comillas dobles. He aquí un ejemplo justo de lo que debes hacer:

```
[css]input[type="text"],
input[type="password"],
.name-field {
background: #f1f1f1;
}  [/css]
```

Orden de propiedad

Las reglas reconocen la necesidad de un poco de espacio

personal aquí, ya que no prescriben un orden específico de **reglas CSS**. Lo que no dicen es que debes **seguir una estructura semántica que tenga sentido**. Las propiedades del grupo por sus relaciones o agruparlos alfabéticamente, simplemente no las escriben de forma aleatoria.

El agregar las propiedades se basa en la lógica en la que se desee visualizar el contenido:

- visualización
- Posicionamiento
- Modelo de caja
- Los colores y la tipografía
- Otros

```
[css].profile-modal {
display: block;
position:absolute;
left:100px;
top:90px;
background: #ff9900;
color: #fff;
}  [/css]
```

Javascript

Javascript se ha convertido en un lenguaje bastante usado en la web, en **WordPress** no ha sido la excepción para ser usado. Para **usar Javascipt en WordPress**, debes basarte en la guía de jQuery, en la cual se refieren a la longitud de una linea de programación, la cual no debe ser extensa.

Punto Y Coma

Es algo que sucede y es bastante común, tal vez sea la regla más sencilla es la que se pasa por alto con mayor frecuencia. Nunca, jamás, debes omitir un punto y coma, por la sencilla razón que tu código no podría funcionar.

Asegúrate de que tu código tenga un punto y coma al final de cada sentencia valida, recuerda que los corchetes no llevan el punto y coma al final.

Sangría

Las tabulaciones siempre se deben utilizar sangría. También debes añadir espacio en el contenido de un cierre, incluso si el contenido de un archivo está contenido en uno.

Líneas De Ruptura

Limitar hasta donde llega determinada instrucción, nos sirve para conocer en que punto termina y comienza un nuevo bloque de código. Con esto también nos ahorramos el dolor de cabeza que nos podría dar el punto y coma.

Ciclos Con jQuery

De acuerdo con la normas de **iteración con jQuery** (jQuery.each ()) sólo debe usarse en **objetos jQuery**. Puedes usar los típicos ciclos de for/in para iterar otros tipos de datos.

Finalmente

La consistencia es la regla más importante. Es mejor desarrollar bastante de cerca con los estándares, y no tener que cambiar a mitad del desarrollo del proyecto una gran cantidad de lineas de código.