Cargar CSS De Forma Correcta En WordPress

Como saben la mayoría de desarrolladores, en el CSS es el que manda en la partida a la hora del diseño visual. Si se desea desarrollar una plantilla bonita y ademas totalmente personalizable por el usuario, deberás aprender el camino correcto para cargar el CSS en WordPress.

WordPress es actualmente el CMS mas popular en el mundo, y cuenta con decenas de millones de usuarios. Es por eso que, con el fin de hacer una plantilla de éxito, tenemos que pensar en cada usuario de WordPress y tratar de ir por la librería y cargarla correctamente para el uso de la personalización.

Como No Se Debe Cargar El CSS En WordPress

Con los años, **WordPress** ha crecido no solo en su popularidad, sino también en su código con el fin de hacerlo más y más flexible, y el permitir archivos CSS y JavaScript fue una gran movimiento para atraer desarrolladores. Los malos hábitos se mantuvieron por un tiempo, sin embargo. Con saber que

WordPress introdujo la adición de archivos CSS y JavaScript, se continuaba añadiendo este código en nuestros archivos **header.php**:

[html]<link rel = href "stylesheet" = "<php echo get_stylesheet_uri ();?>">[/html]

… o agregamos el código de abajo en nuestros archivos **functions.php**, pensando que era mejor:

[php]

<? php
función add_stylesheet_to_head () {
echo "<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>";}

add_action ('wp_head', 'add_stylesheet_to_head');
?>

[/php]

En los anteriores casos, **WordPress** no puede determinar si los archivos CSS se cargan en la página o no. **Eso podría ser un terrible error!**

Si algún otro plugin utiliza el mismo **archivo CSS**, no sería capaz de verificar si el archivo CSS ya se ha incluido en la página. Entonces, dicho plugin carga el mismo archivo por segunda vez, lo que resulta en código duplicado. Por suerte, **WordPress tiene una solución** muy fácil para este tipo de problemas: el registro y encolado de las hojas de estilo.

La Forma Correcta De Cargar CSS en WordPress

Como se menciono al inicio, **WordPress** ha crecido mucho en los últimos años y tenemos que pensar en todos los usuarios de WordPress del mundo. Además de ellos, tenemos también que tomar en cuenta la gran cantidad de plugins de WordPress que existen. Pero no dejes que estos grandes números te asusten: **WordPress** cuenta con funciones muy útiles que nos sirven para **cargar correctamente los archivos CSS en WordPress**.

Vamos a echar un vistazo.

El registro de los archivos CSS

Si usted va a cargar las hojas de estilo CSS, usted debe registrarse primero con la función () wp_register_style:

[php]

```
<?php
wp_register_style( $handle, $src, $deps, $ver, $media );
?>
```

[/php]

- **$handle** (String, requerido) es el nombre único para su hoja de estilo. Otras funciones usarán este «manejador» para poner en cola e imprimir su hoja de estilos.
- **$src** (String, requerido) hace referencia a la URL de la hoja de estilo. Puede utilizar funciones como **get_template_directory_uri ()** para obtener los archivos de estilo dentro del directorio de su tema.
- **$deps** (array, opcional) maneja los nombres de estilos que son dependientes. Si su hoja de estilos no funcionará, en caso de que algún otro archivo de estilo no se encuentra, utilice este parámetro para establecer las «dependencias».
- **$ver** (String o booleano, opcional) es el número de versión. Usted puede utilizar el número de versión de su tema o inventar uno, si deseas. Si no desea utilizar un número de versión, solo pong null. Por defecto es falso, lo que hace **WordPress** es añadir su propio número de versión.
- $media (string, opcional) es el tipo de hoja de estilo CSS que usaria «screen» ó «handheld» ó «print». Si no estás seguro de lo que necesita para usar esto, no lo use. Su valor predeterminado es «all».

He aquí un ejemplo de la función **wp_register_style()**:

[php]

```
<?php

// ejemplo
wp_register_style(
```

```
‘my-bootstrap-extension’, // nombre del manejador
get_template_directory_uri()  .  ‘/css/my-bootstrap-extension.css’, // la URL de la hoja de estilo
array( ‘bootstrap-main’ ), // un array con las dependencias del estilo
‘1.2’, // numero de la version
‘screen’, // tipo de archivo CSS
);

?>
```

[/php]

El registro de estilos es algo opcional en **WordPress**. Si usted no cree que su estilo va a ser utilizado por cualquier otro plugin o no vas a usar cualquier código para cargar de nuevo, es libre de **poner en cola el estilo sin registrarlo**. A continuación se muestra como hacerlo ;)

Encolar Los Archivos CSS En WorPress

Después de registrar nuestro archivo de estilo, tenemos que «poner en cola» para que sea listo para cargar en la sección <head> de nuestro tema. Esto lo hacemos con la función wp_enqueue_style ():

[php]

```
<?php
wp_enqueue_style( $handle, $src, $deps, $ver, $media );
?>
```

[/php]

Los parámetros son exactamente lo mismo con el wp_register_style (función), por lo que no hay necesidad de repetirlos una vez mas :)

Pero como dijimos la función **wp_register_style () no es obligatorio**, ademas, podemos utilizar wp_enqueue_style () de dos maneras diferentes:

[php]

```
<?php
// si se ha registrado antes el archivo
wp_enqueue_style( ‘my-bootstrap-extension’ );

//si no lo hemos registrado, TENEMOS que configurar el parámetro $src !
wp_enqueue_style(
‘my-bootstrap-extension’,
get_template_directory_uri()  .  ‘/css/my-bootstrap-extension.css’,
array( ‘bootstrap-main’ ),
null,
// ..no le añadimos el tipo de CSS
);
?>
```

[/php]

Tenga en cuenta que si un plugin tendrá que encontrar su hoja de estilo o tiene la intención de cargarlo en varias partes de su tema, que sin duda debe registrarse primero.

Cargando Los Estilos En Nuestro Sitio Web

No podemos utilizar la función **wp_enqueue_style()** en cualquier parte de nuestra plantilla, así que, tenemos que utilizar las «acciones». Hay tres acciones que podemos utilizar para diversos fines:

- **wp_enqueue_scripts** para scripts de carga y estilos en el front-end de nuestra página web,
- **admin_enqueue_scripts** para scripts y estilos en las páginas de nuestro panel de administración de carga,
- **login_enqueue_scripts** para scripts de carga y estilos en la página de inicio de sesión de WordPress.

Aquí están los ejemplos de estas tres acciones:

[php]

```
<?php

//cargar el CSS en el front-end del sitio web
function mytheme_enqueue_style() {
wp_enqueue_style( ‘mytheme-style’, get_stylesheet_uri() );
}
add_action( ‘wp_enqueue_scripts’, ‘mytheme_enqueue_style’ );
```

```
// cargar el CSS en la pagina de administracion
function mytheme_enqueue_options_style() {
wp_enqueue_style(           'mytheme-options-style',
get_template_directory_uri() . '/css/admin.css' );
}
add_action(                 'admin_enqueue_scripts',
'mytheme_enqueue_options_style' );

// cargar el CSS en el login del usuario
function mytheme_enqueue_login_style() {
wp_enqueue_style(           'mytheme-options-style',
get_template_directory_uri() . '/css/login.css' );
}
add_action(                 'login_enqueue_scripts',
'mytheme_enqueue_login_style' );

?>

[/php]
```

Antes de continuar es necesario saber: «Use **wp_enqueue_scripts ()**, no **wp_print_styles ()**«. Ya que esto podria informarle sobre una incompatibilidad en WordPress Version 3.3.

Algunas Funciones

Adicionales

Hay algunas funciones muy útiles para el **CSS en WordPress**: Nos permiten mostrar los estilos en línea, comprobar el estado de puesta en cola de nuestros archivos de estilo, añadimos los metadatos de nuestros archivos de estilo, y eliminar los estilos.

Añadir Estilos Dinámicos En Linea: wp_add_inline_style ()

Si la plantilla tiene opciones para personalizar el estilo del tema, puede utilizar un estilo la función () wp_add_inline_style:

[php]

```
<?php

function mytheme_custom_styles() {
wp_enqueue_style( ‘custom-style’, get_template_directory_uri() . ‘/css/custom-style.css’ );
$bold_headlines = get_theme_mod( ‘headline-font-weight’ ); // asignaremos el valor de "bold"
$custom_inline_style = ‘.headline { font-weight: ‘ . $bold_headlines . ‘; }’;
wp_add_inline_style( ‘custom-style’, $custom_inline_style );
}
add_action( ‘wp_enqueue_scripts’, ‘mytheme_custom_styles’ );

?>
```

[/php]

Rápido y fácil. Recuerde, sin embargo: Tiene que usar el mismo nombre hadle con la hoja de estilo que desee agregar un estilo inline después.

Comprobando El Estado Encolado De La Hoja De Estilos: wp_style_is ()

En algunos casos, es posible que necesitemos la información sobre el estado de un estilo: ¿Está registrado, se le pone en cola, está impreso o en espera de impresión? Se puede determinar con la función () wp_style_is:

[php]

```
<?php

/*
* wp_style_is( $handle, $state );
* $handle – name of the stylesheet
* $state – state of the stylesheet: ‘registered’, ‘enqueued’, ‘done’ or ‘to_do’. default: ‘enqueued’
*/

// wp_style_is() example
function bootstrap_styles() {

if( wp_style_is( ‘bootstrap-main’ ) {

// poner la libreria de bootstrap en cola si ya se ha iniciado

wp_enqueue_style(       ‘my-custom-bootstrap-theme’, ‘http://url.of/the/custom-theme.css’ );
```

```
}

}
add_action( ‘wp_enqueue_scripts’, ‘bootstrap_styles’ );

?>

[/php]
```

Añadiendo Metadatos A La Hoja De Estilos: wp_style_add_data()

Aquí está una función muy útil llamada **wp_style_add_data ()** que le permite añadir metadatos a su hoja de estilos, incluyendo los comentarios condicionales, soporte RTL y mucho más!

Échale un vistazo:

```
[php]

<?php

// ejemplo
function mytheme_extra_styles() {
wp_enqueue_style( ‘mytheme-ie’, get_template_directory_uri() .
‘/css/ie.css’ );
wp_style_add_data( ‘mytheme-ie’, ‘conditional’, ‘lt IE 9’ );
/*
* alternate usage:
*    $GLOBALS[‘wp_styles’]->add_data(    ‘mytheme-ie’,
‘conditional’, ‘lte IE 9’ );
* wp_style_add_data() is cleaner, though.
*/
```

```
}

add_action( ‘wp_enqueue_scripts’, ‘mytheme_ie_style’ );

?>

[/php]
```

Interesante, ¿no?

Desregistrar Archivos De Estilo Con wp_deregister_style()

Si alguna vez tiene que «dar de baja» una hoja de estilo (con el fin de volver a registrar con una versión modificada, por ejemplo), puede hacerlo con wp_deregister_style ().

Vamos a ver un ejemplo:

```
[php]

<?php

function mytheme_load_modified_bootstrap() {
//si bootstrap ya se ha registrado
if( wp_script_is( ‘bootstrap-main’, ‘registered’ ) ) {
// primero se des registra
wp_deregister_style( ‘bootstrap-main’ );
// y ahora registramos el nuestro
wp_register_style( ‘bootstrap-main’, get_template_directory_uri() . ‘/css/bootstrap-main-modified.css’ );
// y lo encolamos
wp_enqueue_style( ‘bootstrap-main’ );
}
```

```
}

add_action( 'wp_enqueue_scripts', 'mytheme_load_modified_bootstrap' );

?>

[/php]
```

Aunque no es obligatorio, siempre debe volver a registrar otro estilo sin dar de baja uno, si no lo hacemos, es posible que creemos un error.

Usando Todo

Felicitaciones, ahora sabe todo acerca de la inclusión correcta de CSS en WordPress! Espero que haya sido de gran utilidad el tutorial.

¿Tienes algún consejo o experiencia que quieras compartir? Comparte lo que sabes con nosotros, deja un comentario ;)