

Conoce Como Funciona El Manejo De Las Imágenes En WordPress Para Un Mejor Diseño Responsive

Uno de los mayores desafíos que enfrentan los desarrolladores web de hoy, es el correcto manejo de las imágenes para el [diseño responsive](#). Cuestiones como el tamaño de las imágenes en las pantallas que se ven, la velocidad de descarga del usuario, si el dispositivo es retina (o, en general, un súper altas pantallas de resolución), y más.

Afortunadamente, el [elemento <picture>](#) recientemente se convirtió en una especificación aceptada, la cual será implementado por los principales navegadores, pero todavía hay mucho trabajo que hacer. Por suerte, la forma de manejar imágenes en [WordPress](#) hace las cosas mucho más fácil el desarrollador.

Como Trabaja El Cargador De

Archivos Multimedia De WordPress

Antes de comprender el como se usa el cargador multimedia, veremos su uso habitual en WordPress. El cargador es accesible desde varias áreas, incluyendo post, páginas, tipos de artículos personalizados que lo soportan (en términos generales, el editor), y el menú Multimedia.

Las únicas diferencias entre el menú Media y el resto, es que una imagen se puede asociar directamente con un artículo o una página si se carga desde el editor.



Una vez que una imagen se ha subido, WordPress por defecto crea 4 tamaños para poder insertar la imagen donde deseemos:

- miniatura (150×150)
- medio (300 máx x 300 máx)
- grande (640 x 640 máx máx)
- completa (el tamaño original de la imagen).

Aunque tambien podemos personalizar el tamaño de la imagen, es decir, aunque hayamos subido una imagen grande, podemos cambiar el tamaño a algo mas pequeño a nuestro gusto. También puede agregar sus propios tamaños de imagen mediante la función [add_image_size \(\)](#). Por ejemplo, si queremos añadir una imagen para un producto que es 700px de ancho por 450px de

alto, haríamos:

```
[php]add_image_size( 'product-img', 700, 450, false );[/php]
```

Esto está diciendo a **WordPress** que cree una nueva imagen con el nombre de «product-img » con nuestras **dimensiones especificadas**. El último argumento es si la imagen puede ser cortada.

Cuando se establece en false, la imagen **será redimensionada proporcionalmente sin recortar**; cuando se establece en true, la imagen se **recortará cualquiera de los lados** o en la parte superior / inferior. Tenga cuidado con esto, sin embargo, los resultados pueden variar de imagen a imagen.

Insertar Una Imagen

Hay dos formas de insertar una imagen en un artículo o una página en **WordPress**: la primera es a través del cargador de archivos multimedia, como se muestra aquí:



La segunda forma es mediante el uso de la opción de **Imagen Destacada**. Puede habilitar imágenes destacadas agregando este código en el archivo **functions.php** de tu plantilla, o en algún lugar cerca del comienzo de sus plugins:

```
[php]<br /><br />if ( function_exists( 'add_theme_support' ) )  
{<br /> add_theme_support( 'post-thumbnails' );<br />}<br /><br />[/php]
```

Esto agregará un cuadro de «Imagen Destacada» a todos los artículos, páginas y tipos de artículos personalizados que lo soporten. También puede enviar un segundo argumento, que es una matriz de lo que debe tener una imagen destacada.

Por ejemplo, si sólo desea usar imágenes destacadas en artículos destacados:

```
[php]<br /><br />if ( function_exists( 'add_theme_support' ) )  
{<br /> add_theme_support( 'post-thumbnails', array('post'  
);<br />}<br /><br />[/php]
```

El último paso es colocar el código dentro del bucle de la plantilla, siempre que sea desea que la imagen aparezca:

```
[php]<br /><br />if ( has_post_thumbnail() ) {<br />  
the_post_thumbnail('large');<br />}<br /><br />[/php]
```

El argumento aceptado es del tamaño de la imagen que desea utilizar. De forma predeterminada, es posterior a la de miniaturas. Una vez conseguido eso, se le añade tu imagen ofrecida a través de esta casilla:



Por último, si se carga un conjunto de imágenes en el mismo puesto o una página, puede insertar como una galería con los **shortcode [gallery], como mas sobre como usarlos [aquí](#).**

Las Mejoras De

La Version 3.9

Con la versión 3.9 de WordPress, y con ella, algunas mejoras muy llamativas para las imágenes y la galería. Además de mejoras en el editor visual, al igual que la capacidad de arrastrar y soltar imágenes, [add_image_size\(\)](#) es conseguir un nuevo argumento para especificar si desea recortar en los lados o la parte superior / inferior.

Entonces, ¿Que tiene que ver todo esto con el **diseño Responsive**? Me alegro de que lo preguntes! ;)

Usando picturefill.js

Existe un archivo [JavaScript](#) creado por el **Filament Group** llamado [picturefill.js](#), que emula la misma función que el elemento `<picture>`.

La sintaxis es la siguiente:

```
[js]<br /><br /><span data-picture data-alt="A giant stone face at The Bayon temple in Angkor Thom, Cambodia"><br />
<span data-src="small.jpg"></span><br /> <span data-src="medium.jpg" data-media="(min-width: 400px)"></span><br /> <span data-src="large.jpg" data-media="(min-width: 800px)"></span><br />
<span data-src="extralarge.jpg" data-media="(min-width: 1000px)"></span><br /><br /> <!-- Fallback content for non-JS browsers. Same img src as the initial, unqualified
```

```
source element. -&gt;<br /> &lt;noscript&gt;<br /> &lt;img  
src="small.jpg" alt="A giant stone face at The Bayon temple in  
Angkor Thom, Cambodia"&gt;<br /> &lt;/noscript&gt;<br  
>&lt;/span&gt;<br /><br />[/js]
```

Note que hay una entrada llamada **data-src** para cada imagen que queremos usar, así como un punto de división mínimo asociado para mostrar esa imagen. Vale la pena señalar que no hay límite para el número de imágenes/puntos de ruptura que pueda tener.

Esta es actualmente una manera en gran medida aceptada para lograr el mismo efecto que el elemento **<picture>**. Esto nos permitirá utilizar mejor y nuevos elementos en [HTML](#) en los navegadores modernos. Con [picturefill.js](#) y el cargador de imagen de **WordPress**, podemos crear **imágenes potentes adaptables al diseño responsive**.

Agrupando Todo

En este momento, no hay manera de que [WordPress](#) lo haga de forma nativa; podemos hacer un par de cosas para automatizar el proceso. En esta sección, vamos a tocar algunas posibilidades.

Tenga en cuenta !: Usted se dará cuenta que se ha usado el **prefijo jlc_** en las funciones. Te recomiendo utilizar esto para evitar conflictos con otras plantillas y plugins.

Reemplazando Las Imágenes

La primera pieza de este rompecabezas, y el más sencillo, es reemplazar las imágenes destacadas con el marcado de **picturefill.js**. Esto se puede lograr con el filtro [post_thumbnail_html](#), que intercambiará el marcado predeterminado por el nuestro.

```
[php]<br /><br />function jlc_script() {<br />wp_register_script(          'picturefill',  
get_stylesheet_directory_uri() . '/js/picturefill.js');<br />wp_enqueue_script( 'picturefill' );<br />}<br />add_action(  
'wp_enqueue_scripts', 'jlc_script' );<br /><br />[/php]
```

Asegúrese de modificar la ruta donde se encuentre su archivo **picturefill.js**. Una vez que esté en su lugar, es el momento de añadir nuestra función de sustitución:

```
[php]<br /><br />function jlc_get_featured_image( $html, $aid  
= false ) {<br /><br /> $sizes= array( 'thumbnail', 'medium',  
'large', 'full' );<br /> $img= '<span data-picture data-  
alt="">'.get_the_title().'">';<br /> $ct = 0;<br /> $aid = (  
! $aid ) ? get_post_thumbnail_id() : $aid;<br /><br /> foreach  
( $sizes as $size ) {<br /><br /> $url =  
wp_get_attachment_image_src( $aid, $size );<br /> $width = (  
$ct < sizeof( $sizes ) - 1 ) ? ( $url[1] * 0.66 ) : (  
$width / 0.66 ) + 25;<br /> $img .= '<span data-src="">'.  
$url[0] .'"';<br /> $img .= ( $ct > 0 ) ? ' data-media="( min-width: ' .  
$width . 'px)"></span>';  
: '></span>';<br /> $ct++;<br /> }<br /><br /> $url=  
wp_get_attachment_image_src( $aid, $sizes[1] );<br /> $img.=
```

```
'<noscript><br /> <br /> </noscript><br />
</span>';<br /><br /> return $img;<br /><br />}<br /><br />
/>[/php]
```

Hay varias tareas que se realizan aquí:

1. Tenemos un arsenal de cada tamaño de la imagen que queremos que se incluya (por su nombre). Usted puede agregar o quitar aquí como desee.
2. Hemos creado el marcado con `picturefill.js`.
3. Con un ciclo nosotros a través de la matriz, obtenemos la URL para cada tamaño y lo agregamos como una entrada a nuestro elemento `picturefill.js`
4. Incluimos algunos cálculos para averiguar donde los puntos de interrupción debe ocurrir sobre la base de los tamaños de imagen
5. Para usuarios sin javascript, se incluye por defecto un tamaño mediano.

El siguiente paso es agregar nuestro filtro, el cual puede añadir justo debajo de la función:

```
[php]add_filter( 'post_thumbnail_html',
'jlc_get_featured_image');[/php]
```

Ahora, la función `jlc_get_featured_imag ()` se utilizará para la salida **`post_thumbnail`** en lugar del marcado predeterminado. Todo se ha envuelto usando un simple plugin llamado [Responsive Featured Image](#). Puede [descargarlo desde Github](#).

Otros Métodos

Como se puede imaginar, el manejo de las imágenes es un asunto bastante complicado cuando se trata el concepto de **diseño responsive**. La funcionará solamente con imágenes destacadas, no con todas las imágenes en el artículo. Si bien existen métodos para hacer esto, incluyendo [este plugin](#), mis pruebas han demostrado problemas de rendimiento y escalado.

También puede utilizar un código corto, pasando una URL o un ID de imagen para generar el código, justo así: `[jlc_picturefill_image aid=x]`. Desafortunadamente sin codificación extensa, esto desactiva el uso de la cargador de imágenes para insertar imágenes en el artículo. Con la funcionalidad de arrastrar y soltar en **WordPress 3.9**, podría causar mas conflictos.

Finalmente

En este artículo, nos centramos bastante en cómo **WordPress** maneja las imágenes y exploramos una manera de integrar **picturefill.js** junto a **WordPress**, y poder hacer uso eficiente de las imágenes adaptables. Es importante saber que esta es un área en constante evolución de su interés. Esperamos haya sido de gran utilidad esta pequeña guía.