

Lectura de imágenes externas

La visualización de imágenes no presenta ningún problema –lo hacemos habitualmente mediante etiquetas HTML– cuando se encuentran en el espacio del servidor, bien sea propio o ajeno.

El problema puede surgir cuando tratemos de almacenar esas imágenes fuera del *root* del servidor (una forma de impedir la accesibilidad desde otras webs) y eso puede conseguirse mediante las funciones que veremos en este capítulo.

El primer paso será establecer la ruta y el nombre de la imagen. Al margen tienes comentados algunos detalles al respecto.

El paso siguiente será *extraer* el formato de la imagen. Lo hice leyendo la parte de la cadena comprendida entre el último punto (.) y el final de la cadena que contiene el nombre de la imagen.

Mediante el **switch** elijo las instrucciones para cada tipo de imagen que son similares pero con *matices* según del formato.

La visualización de la imagen contiene tres instrucciones: *Header*, *imagecreatefrom* e *imageXXX*.

En **Header** hay que incluir un **Content - type** acorde con el tipo de imagen. Puede ser, entre otros valores: **image/jpeg**, **image/png** ó **image/gif** según la imagen tenga formato: **jpg**, **png** ó **gif**.

Con idéntico criterio, la función que **crea** la imagen ha de ser una de estas:

```
$f=imagecreatefromjpeg($i)  
ó  
$f=imagecreatefrompng($i)  
ó  
$f=imagecreatefromgif($i)
```

siendo **\$i** la variable que recoge el *nombre* y el *path* de la imagen original y **\$f** la variable que contiene el resultado de la ejecución de esta función.

La función *image* (la que permite visualizar la nueva imagen) también puede tener una de las tres variantes que ya conocemos de ejemplos anteriores. Pueden ser:
imageieea(\$f), **imageona(\$f)** ó

Lectura de imágenes externas

Uno de los problemas que puede presentarse es la forma de indicar dónde están la imagen a visualizar. Utilizaremos la variable predefinida **\$_SERVER['DOCUMENT_ROOT']** para ubicarnos en el directorio *root* del servidor Apache. De esta forma no tendremos problemas de visualización sa cual fuere el directorio de instalación de servidor web.

```
<?  
# indicar la ruta de la imagen  
$original=$_SERVER['DOCUMENT_ROOT']."/cursophp/images/caballos.jpg";  
  
# extraer el tipo de imagen según su la extensión del fichero  
for($i=strlen($original)-1;$i>0;$i--){  
    if (substr($original,$i,1)=="."){  
        $tipo=substr($original,$i+1);  
        break;  
    }  
}  
# las diferentes opciones dependiendo del formato de la imagen  
switch($tipo){  
    case "jpg":  
        Header("Content-type:image/jpeg");  
        $nueva=imagecreatefromjpeg($original);  
        imagejpeg($nueva);  
        break;  
  
    case "png":  
        Header("Content-type:image/png");  
        $nueva=imagecreatefrompng($original);  
        imagepng($nueva);  
        break;  
  
    case "gif":  
        Header("Content-type:image/gif");  
        $nueva=imagecreatefromgif($original);  
        imagegif($nueva);  
        break;  
}  
ImageDestroy();  
?>
```

[Ver ejemplo .jpg](#)[Ver ejemplo .png](#)[Ver ejemplo .gif](#)

¡Cuidado!

Observa que en las imágenes en formato **png** se visualizan con deficiencias en los bordes de las áreas transparentes.

Con el método que vemos a continuación ese problema se reduce considerablemente.

Lectura y redimensionado de imágenes externas

```
<?  
# indicar la ruta de la imagen  
$original=$_SERVER['DOCUMENT_ROOT']."/cursophp/images/caballos.jpg";
```

imagegif(\$f), donde **\$f** es la variable que recoge el resultado de la función anterior.

Redimensionado de imágenes externas

Tampoco parece *ninguna* utilidad. ¿Verdad? A fin de cuentas con etiquetas HTML podemos asignar el ancho y el alto de una imagen. Pero... ya verás como no es tan *trivial* esta opción.

El proceso es el siguiente:

1º.- Determinamos cuales son las dimensiones de la imagen externa que vamos a utilizar. Para ello, usaremos de la función:

\$dim=getimagesize(\$r)

donde **\$r** es la variable que contiene el path y nombre del fichero que contiene la imagen y **\$dim** es un **array escalar** que contiene las dimensiones de la imagen analizada.

El elemento del array **\$dim[0]** contiene el ancho y **\$dim[1]** el alto, ambos expresados en **pixels**.

¡Cuidado...!

Esta función solo funciona cuando se trata de imágenes externas. Para determinar las dimensiones de imágenes generadas por PHP tendrás que utilizar otra distinta. De nada :-)

2º.- Creamos una *copia* de la imagen original por medio de la función **imagecreate** adecuada al **tipo** de imagen que deseamos importar. Es exactamente lo que hemos visto en el párrafo anterior.

3º.- Creamos una **nueva imagen** -podemos trabajar con varias imágenes, en eso no hay problema- mediante la función:

\$d=imagecreatetruecolor(x,y)

dónde **\$d** es el identificador de la imagen, y **x** y **y** son las dimensiones de esta nueva imagen.

Dado que esta imagen va a ser el **soporte** -una imagen en color verdadero (de ahí lo de *truecolor*) con fondo negro, algo muy similar al papel fotográfico que se usa en los laboratorios- sobre el que se va a *impresionar* esa especie de *negativo* que es la *imagen original* es necesario que sus dimensiones sean las deseadas para la imagen resultante.

4º.-Ahora **toca positivar** la nueva foto. Para hacerlo disponemos de la función

imagecopyresampled() que debe incluir -dentro del paréntesis- un

```
for ($i=strlen($original)-1;$i>0;$i--){
    if (substr($original,$i,1)=="."){
        $tipo=substr($original,$i+1);
        break;
    }
}
# dimensiones del original
$tamano=getimagesize($original);
$orig_Ancho = $tamano[0];
$orig_Alto = $tamano[1];
# factores de ampliación, distintos para provocar una distorsión
# en la imagen resultante
$ampliacion_X=2;
$ampliacion_Y=1.5;
# dimensiones de la imagen resultante. Vamos a dejarla a sangre
# (sin márgenes en blanco) y vamos a reproducir el original
# sin reencuadrar así que las esquinas superiores izquierdas de
# ambas imágenes estarán en 0,0.
$resultado_Ancho=$orig_Ancho*$ampliacion_X;
$resultado_Alto= $orig_Alto*$ampliacion_Y;
# creamos una imagen a partir de la original. Debemos elegir
# la función adecuada al tipo de imagen original
switch($tipo){
    case "jpg":
        $importada=imagecreatefromjpeg($original);
        break;
    case "png":
        $importada=imagecreatefrompng($original);
        break;
    case "gif":
        $importada=imagecreatefromgif($original);
        break;
}
# insertamos la cabecera de la nueva imagen
Header("Content-type:image/jpeg");
# creamos una imagen nueva en color verdadero
$im_base=imagecreatetruecolor($resultado_Ancho,$resultado_Alto);
# aplicamos un color de fondo a la nueva imagen
# para poder visualizar que incluye la transparencia del png o del gif
if($tipo=="png" OR $tipo=="gif"){
    $fondo=imagecolorallocate($im_base,255,255,200);
    imagefill($im_base,0,0,$fondo);
}
# superponemos la imagen importada sobre la que acabamos de crear
imagecopyresampled($im_base,$importada,0,0,0,0,
    $resultado_Ancho, $resultado_Alto,
    $orig_Ancho,$orig_Alto);
# visualizamos la imagen resultante
imagejpeg($im_base);
ImageDestroy();
?>
```

[Ver ejemplo .jpg](#)

[Ver ejemplo .png](#)

[Ver ejemplo .gif](#)

Observa que -tanto en el ejemplo anterior como en el siguiente- solo hemos utilizado la extensión de la imagen original para elegir la función **imagecreatefrom....** En el Header hemos puesto **image/jpeg** y, como es obvio, hemos utilizado la función asociada a este formato (**imagejpeg**). Si sustituimos *ambos valores* por los correspondientes a otro formato (gif, png) obtendríamos resultados similares.

Recortar imágenes externas

```
<?
# obtener la imagen
$original=$_SERVER['DOCUMENT_ROOT']."/cursophp/images/aviones4.jpg";
for ($i=strlen($original)-1;$i>0;$i--){
    if (substr($original,$i,1)=="."){
        $tipo=substr($original,$i+1);
        break;
    }
}
# tamaño del original
```

este orden):

\$d que es el identificador de la imagen destino, es decir el papel fotográfico que hemos *creado* en el paso anterior.

\$f que es el identificador de la imagen original (*negativo*) obtenido en el punto 2°.

X_d e **Y_d** son las coordenadas de un punto situado en la *esquina superior izquierda del papel* a partir del que queremos que se *impresione la fotografía*. Si queremos una *foto a sangre* pondremos **0,0** y, si quieres dejar **márgenes en blanco**, habrá que poner los anchos de esos márgenes (izquierdo y superior) respectivamente.

X_f e **Y_f** nos servirán para *reencuadrar* la foto original *recortando* por la izquierda y por arriba, respectivamente, los anchos que se indiquen aquí en pixels.

D_x e **D_y** indican el ancho y el alto (por este orden) que va a tener la mancha de imagen en el positivo. Ten en cuenta que *no puedes salirte del papel* así que esos valores sumados con los márgenes (izquierdo y superior) no podrán ser mayores que las dimensiones que has elegido para el *papel fotográfico* en el punto 2°.

F_x e **F_y** indican el ancho y el alto de la porción del *original* que tratamos de reproducir. Sumados con **X_f** e **Y_f** no pueden exceder el tamaño del *negativo*.

Con estos parámetros la función ya se encarga de redimensionar la imagen (incluso distorsionarla si no hay proporcionalidad entre los anchos y altos del *original* y del *soporte*).

```
$tamano=getimagesize($original);
$orig_Ancho = $tamano[0];
$orig_Alto = $tamano[1];
# estableceremos un margen en blanco alrededor de la imagen de 5 pixels
# igual por los cuatro lados
$margen=10;
# establecemos recortes para reencuadrar la imagen
$recorte_izq=50;
$recorte_sup=80;
$recorte_der=40;
$recorte_inf=60;
# calculamos las dimensiones para utilizar como parámetros
# en la funcion imagecopyresampled
# ancho y alto original recortado
$Ancho_recortado=$orig_Ancho-$recorte_izq-$recorte_der;
$Alto_recortado=$orig_Alto-$recorte_sup-$recorte_inf;
# factores de ampliación en este caso iguales
# sin distorsión de imagen
$ampliacion_X=1;
$ampliacion_Y=1;
# dimensiones del soporte
$papel_Ancho=$Ancho_recortado*$ampliacion_X+ 2*$margen;
$papel_Alto=$Alto_recortado*$ampliacion_Y+2*$margen;
# dimensiones de la mancha de imagen al positivar
# hay que quitar los márgenes
$resultado_Ancho=$papel_Ancho -2*$margen;
$resultado_Alto=$papel_Alto -2*$margen;
switch($tipo){
  case "jpg":
    $importada=imagecreatefromjpeg($original);
    break;
  case "png":
    $importada=imagecreatefrompng($original);
    break;
  case "gif":
    $importada=imagecreatefromgif($original);
    break;
}
Header("Content-type:image/jpeg");
$im_base=imagecreatetruecolor($papel_Ancho,$papel_Alto);
$fondo=imagecolorallocate($im_base,255,255,200);
imagefill($im_base,0,0,$fondo);
imagecopyresampled($im_base,$importada,$margen,$margen,
  $recorte_izq,$recorte_sup,
  $resultado_Ancho,$resultado_Alto,
  $Ancho_recortado,$Alto_recortado);

imagejpeg($im_base);
ImageDestroy();
?>
```

Ver imágenes original y
resultante

Anterior



Índice



Siguiente

