

Unidad 4

Imágenes en PHP

Creación de imágenes dinámicas

Una imagen dinámica es un fichero PHP que contiene las instrucciones para su creación.

Para visualizar una imagen dinámica desde una página web basta con invocar desde la etiqueta clásica de inserción de imágenes de HTML:

```

```

Primera etiqueta

Una vez conocidos los formatos que soporta nuestra versión ya podemos generar imágenes en cualquiera de esos formatos.

Vimos un script que nos permite saber esto en la unidad anterior. Para el ejemplo, trabajaremos con dos formatos: JPG y PNG.

Cabe aclarar desde el principio, que en algunas versiones de servidor local, se han comprobado problemas al mostrar ciertas imágenes dinámicas en formato PNG, cosa que no sucede una vez colocado on-line el script, sobre todo cuando el PNG no posee color de fondo.

La primera instrucción de un fichero de creación de imágenes dinámicas ha de ser:

```
header("Content-type: image/jpeg")
```

si se trata de crear una imagen JPG o:

```
header("Content-type: image/png")
```

si pretendemos que la imagen tenga formato PNG.

En el fichero de creación de la imagen no se debe insertar más que el código de creación de la imagen y no se deben dejar líneas en blanco entre la etiqueta de apertura de PHP **<?php** y la etiqueta header ya que caso contrario nos da un error.

Definida la etiqueta anterior, tenemos que crear la imagen, dibujarla y luego enviarla al navegador para su visualización y por último (no es imprescindible pero si muy conveniente) borrarla con el fin de liberar la memoria ocupada durante su generación.

Funciones PHP para esos procesos:

`$nombre = imagecreate(anch,alto)`

Con esta función creamos una imagen con el tamaño especificado (anch y alto) en pixeles que es recogida en la variable nombre. Esta función es idéntica para cualquier formato de imagen.

Envío de imágenes al navegador

Para enviar imágenes al navegador (visualización) se usan funciones diferentes dependiendo del tipo de imagen definida en Header.

Si pretendemos que la imagen tenga formato JPG habremos puesto en Header la indicación jpeg (Cuidado que es jpeg y no jpg).

En este caso la función de visualización sería:

`imagejpeg($nombre)`

Si se tratara de una imagen en formato PNG (recuerden que debe estar definido en header) la sintaxis sería:

`imagepng($nombre)`

Eliminar imágenes de la memoria

Independientemente del formato utilizado para borrar imágenes de la memoria (no del navegador) se utiliza la siguiente sintaxis:

`imagedestroy($nombre)`

Como podemos observar en ejemplo_01_imagecreate.php y en el ejemplo ejemplo_01_imagecreate.php la imagen que se genera es un cuadrado con fondo negro en el primer caso y un rectángulo con fondo negro en el segundo, ambos con las dimensiones especificadas en cada script.

Ejemplos:

```
<?php
header("Content-type: image/jpeg");
$im = imagecreate(200,200); imagejpeg($im); imagedestroy($im);
?>
```

```
<?php  
header("Content-type: image/png");  
$im = imagecreate(300,200); imagepng($im); imagedestroy($im);  
?>
```

Colores

PHP permite crear una paleta de colores.

Para ello se pueden crear variables de color (independientemente del formato utilizado) con la siguiente función:

```
$color=imagecolorallocate ($nombre,R,G,B)
```

Donde la variable recoge el color especificado por la mezcla de los colores primarios indicados en R, G y B que serán números enteros comprendidos entre 0 y 255 y que especifican la intensidad de las luces roja, verde y azul utilizadas para la obtención del color.

Se pueden definir tantos colores como se deseen sin más que utilizar nombres del variables distintos para cada uno de ellos.

Aplicar colores de fondo

Para aplicar un color de fondo a una imagen (independientemente del formato) se utiliza la siguiente función:

```
imagefill($nombr,x,y,$col)
```

Aquí \$nombr es la variable en la que se ha creado la imagen, x e y son las coordenadas del punto de la imagen a partir del cual se aplica el relleno y \$col el color (previamente definido) aplicable a la imagen.

Vemos a continuación el ejemplo con JPG.

Ejemplo:

```
<?php
header("Content-type: image/jpeg");
$im = imagecreate(200,200);
$fondo=imagecolorallocate ($im, 0, 0, 200);
imagefill ($im, 0, 0, $fondo); imagejpeg($im);
imagedestroy($im);
?>
```

Vemos a continuación el ejemplo con PNG.

Ejemplo:

```
<?php  
header("Content-type: image/png");  
$im = imagecreate(300,200);  
$fondo=imagecolorallocate ($im, 0, 0, 200);  
imagefill ($im, 0, 0, $fondo); imagepng($im); imagedestroy($im);  
?>
```

Formas

Rectángulos sin relleno

Para dibujar un rectángulo sin relleno (solo las líneas) se utiliza la siguiente función:

```
imagerectangle($nom, x0, y0, x1, y1, $col)
```

Donde \$nom es el nombre de la imagen, x0, y0 las coordenadas del vértice superior izquierdo y x1, y1 las coordenadas del vértice inferior derecho y \$col el color con el que queremos dibujar las líneas del rectángulo.

Cabe recordar siempre que el punto (0,0) siempre es la esquina superior izquierda de la imagen y que si no usamos para las líneas un color distinto al del fondo no se verá nada.

Ejemplo:

```
<?php
header("Content-type: image/jpeg");
$im = imagecreate(200,200);
$fondo=imagecolorallocate ($im, 0, 0, 200);
$blanco=imagecolorallocate ($im, 255, 255, 255);
imagefill ($im, 0, 0, $fondo);
imagerectangle ($im, 10, 10, 190, 190, $blanco); imagejpeg($im);
imagedestroy($im);
?>
```

Rectángulos con relleno

Para dibujar un rectángulo con relleno se utiliza la siguiente función:

```
imagefilledrectangle($nom, x0, y0, x1, y1, $col)
```

Los parámetros son idénticos a los del caso anterior con la única diferencia de que en este caso el rectángulo aparecerá relleno con el color elegido.

Ejemplo:

```
<?php
header("Content-type: image/jpeg");
$im = imagecreate(200,200);
$fondo=imagecolorallocate ($im, 0, 0, 200);
$blanco=imagecolorallocate ($im, 255, 255, 255);
```



```
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagefill ($im, 0, 0, $fondo);
imagerectangle ($im, 10, 10, 190, 190, $blanco);
imagefilledrectangle ($im, 20, 20, 180, 180, $amarillo);
imagejpeg($im);
imagedestroy($im);

header("Content-type: image/jpeg");
$esquinas=array(20,100,100,180,180,100,100,20);
$im = imagecreate(200,200);
$fondo=imagecolorallocate ($im, 0, 0, 200);
$blanco=imagecolorallocate ($im, 255, 255, 255);
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagefill ($im, 0, 0, $fondo);
imagerectangle ($im, 10, 10, 190, 190, $blanco);
imagefilledrectangle ($im, 20, 20, 180, 180, $amarillo);
imagefilledpolygon ($im, $esquinas, 4, $blanco); imagejpeg($im);
imagedestroy($im);
?>
```

Polígonos sin relleno

Su funcionamiento es idéntico al anterior en tanto y cuanto requiere que se defina el array de coordenadas de los vértices y los parámetros de la función son los mismos indicados en el caso anterior. Solo se modifica el nombre de la función que en este caso es:

```
imagepolygon($nom, $vert, nº vert , $col)
```

Ejemplo:

```
<?php
header("Content-type: image/jpeg");
$esquinas=array(20,100,100,180,180,100,100,20);
$im = imagecreate(200,200);
$fondo=imagecolorallocate ($im, 0, 0, 200);
$blanco=imagecolorallocate ($im, 255, 255, 255);
$negro=imagecolorallocate ($im, 0, 0, 0);
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagefill ($im, 0, 0, $fondo);
imagerectangle ($im, 10, 10, 190, 190, $blanco);
imagefilledrectangle ($im, 20, 20, 180, 180, $amarillo);
imagepolygon ($im, $esquinas, 4, $negro);
```



```
imagejpeg($im);  
imagedestroy($im);  
?>
```

Elipses, circunferencias y arcos

Una misma función nos permite dibujar elipses, circunferencias y arcos. Es esta:

```
imagearc($nom, Xc, Yc , a, b, Gi, Gf, $col)
```

Los parámetros de esta función son los siguientes:

- \$nom es el nombre de la imagen.
- Xc e Yc son las coordenadas del centro de la elipse.
- a es la longitud del eje horizontal de la elipse.
- b es la longitud del eje vertical de la elipse.
- Gi es el punto inicial del arco expresado en grados sexagesimales.
- Gf es el punto final del arco expresado en grados sexagesimales.
- \$col es el color con el que se dibujará la línea.

A los efectos de los ángulos, CERO GRADOS coincide con el cero trigonométrico pero el recorrido de la circunferencia se hace en sentido contrario al trigonométrico es decir, siguiendo el sentido de las agujas del reloj.

Obviamente, para dibujar una circunferencia basta con hacer iguales los valores de a y de b.

Ejemplo:

```
<?php  
$esquinas=array(20,100,100,180,180,100,100,20);  
$im = imagecreate(200,200);  
$fondo=imagecolorallocate ($im, 0, 0, 200);  
$blanco=imagecolorallocate ($im, 255, 255, 255);  
$amarillo=imagecolorallocate ($im, 255, 255,0);  
imagefill ($im, 0, 0, $fondo);  
imagerectangle ($im, 10, 10, 190, 190, $blanco);  
imagefilledrectangle ($im, 20, 20, 180, 180, $amarillo);  
imagepolygon ($im, $esquinas, 4, $blanco);  
imagearc ($im, 100, 100, 160, 160, 0, 360, $fondo);  
imagearc ($im, 100, 100, 160, 100, 0, 360, $rojo);  
imagejpeg($im);  
imagedestroy($im);
```

?>

Dibujar sobre una imagen de fondo

PHP permite crear imágenes utilizando como fondo una preexistente.

Para ello basta con crear una variable indicando el path y el nombre de la imagen.

Ejemplo:

```
$b="./images/cruz.png"
```

ó

```
$b="./images/cruz.jpg"
```

El formato de esta imagen debe coincidir con el de la que pretendemos construir.

Después de definir esta variable bastaría con sustituir la instrucción de creación de imagen.

```
$nombre = imagecreate(x,y)
```

por

```
$nombre= imagecreatefrompng ($b)
```

en el caso de imágenes PNG

o por

```
$nombre= imagecreatefromjpeg ($b)
```

si se tratara de imágenes en formato JPG.

Ejemplo:

```
<?php
header("Content-type: image/jpg");
$esquinas=array(20,100,100,180,180,100,100,20);
$fondo_img="pink.jpg";
$im = imagecreatefromjpeg($fondo_img);
$fondo=imagecolorallocate ($im, 0, 0, 200);
$blanco=imagecolorallocate ($im, 255, 255, 255);
$amarillo=imagecolorallocate ($im, 255, 255,0);
```

```
$negro=imagecolorallocate($im, 0, 0,0);  
imagefill ($im, 0, 0, $fondo);  
imagerectangle ($im, 10, 10, 190, 190, $blanco);  
imagefilledrectangle ($im, 20, 20, 180, 180, $amarillo);  
imagepolygon ($im, $esquinas, 4, $blanco);  
imagearc ($im, 100, 100, 160, 160, 0, 360, $fondo);  
imagearc ($im, 100, 100, 160, 100, 0, 360, $rojo);  
imagejpeg($im);  
imagedestroy($im);  
?>
```

Guardar imágenes

Las imágenes creadas con la sintaxis anterior no se almacenan como tales en servidor.

Si pretendemos que se almacenen deberemos modificar la sintaxis de la etiqueta:

```
Imagepng($nombre) o Imagejpeg($nombre)
```

añadiéndoles un nuevo parámetro con el nombre y la extensión del fichero gráfico que pretendemos almacenar.

Así por ejemplo:

```
Imagepng($nombre, "mi_imagen.png")
```

o

```
Imagejpeg($nombre, "mi_imagen.jpg")
```

se guardarían en el servidor las imágenes creadas con los nombres `mi_imagen.png` o `mi_imagen.jpg`

Trazar segmentos

La función PHP que permite dibujar segmentos rectilíneos es la siguiente:

```
imageline($nom,x0,y0,x1,y1,$col)
```

donde:

- \$nom es el nombre de la variable definida por imagecreate,
- x0 e y0 son las coordenadas de uno de los extremos;
- x1e y1 son las coordenadas del otro extremo y
- \$col es la variable de color con el que será dibujada la línea.

Ejemplo:

```
<?php
header("content-type: image/png");
$im = imagecreate(200,200);
$fondo=imagecolorallocate ($im,0,0,255);
$linea=imagecolorallocate ($im,255,255,255);
imageline($im,0,0,200,200,$linea);
imagepng($im);
imagedestroy($im);
?>
```

Colores de fondo

PHP utiliza como fondo de la imagen y de forma automática el primer color definido por la función ImageColorAllocate.

Esta opción de PHP nos obliga a definir dos colores distintos para conseguir la visibilidad de las líneas.

Crear transparencias

Si pretendemos que un determinado color se vuelva transparente debemos utilizar la función:

`imagecolortransparent ($nom , $col).`

- \$nom es el nombre de la variable definida por imagecreate
- \$color es el color que pretendemos hacer transparente.

No hay que olvidar nunca dos pequeños detalles:

- Si pretendemos un fondo transparente debemos hacer transparente el primero de los colores definidos.
- Esta función solo tiene sentido para imágenes en formato PNG que admiten transparencias. JPG no las permite.

Ejemplo:

```
<?php  
header("content-type: image/png");  
$im = imagecreate(200,200);  
$fondo=imagecolorallocate ($im,0,0,255);  
$linea=imagecolorallocate ($im,255,0,0);  
imagecolortransparent ($im ,$fondo);  
imageline($im,0,0,200,200,$linea); imagepng($im);  
imagedestroy($im);  
?>
```

Textos

Estas son las funciones más sencillas que nos permiten insertar textos dentro de una imagen.

`imagechar ($im, n, x, y, $txt, $col)`

Requiere que la variable `$txt` contenga una cadena y que esté definida previamente y la utilidad de la función es que inserta el primer carácter de la cadena con orientación horizontal.

Los parámetros de la función son los siguientes:

- `$nom` el nombre de la variable definida por `imagecreate`
- `n` es un número comprendido entre UNO y CINCO que asigna el tamaño de la letra de menor a mayor.
- `x` e `y` son las coordenadas del punto donde se colocará la esquina superior izquierda del carácter a representar.
- `$txt` es la cadena de texto de la que se extraerá el primer carácter para ser representado.
- `$col` es el color del carácter a representar

```
<?php
header("content-type: image/png");
$im = imagecreate(150,150);
$t1="tamaño 1";
$t2="tamaño 2";
$t3="tamaño 3";
$t4="tamaño 4";
$t5="tamaño 5";
$fondo=imagecolorallocate ($im, 0, 0, 200);
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagechar ($im, 1, 0, 0, $t1, $amarillo);
imagechar ($im, 2, 20, 20, $t2, $amarillo);
imagechar ($im, 3, 40, 40, $t2, $amarillo);
imagechar ($im, 4, 60, 60, $t2, $amarillo);
imagechar ($im, 5, 80, 80, $t2, $amarillo); imagepng($im);
imagedestroy($im);
?>
```

Tenemos además otras funciones para el ingreso de caracteres, por ejemplo:

`imagecharup ($im, n, x, y, $txt, $col)`

Su funcionamiento es similar a la función anterior con la única diferencia que inserta el carácter con orientación vertical.

Las coordenadas de inserción también se corresponden con la esquina superior izquierda del carácter pero recordemos que ahora estará girado y que por lo tanto ese punto coincidirá con parte inferior izquierda de la imagen del carácter.

Ejemplo:

```
<?php
header("content-type: image/png");
$im = imagecreate(150,150);
$t1="tamaño 1";
$t2="tamaño 2";
$t3="tamaño 3";
$t4="tamaño 4";
$t5="tamaño 5";
$fondo=imagecolorallocate ($im, 0, 0, 200);
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagecharup ($im, 1, 10, 10, $t1, $amarillo);
imagecharup ($im, 2, 20, 20, $t2, $amarillo);
imagecharup ($im, 3, 40, 40, $t2, $amarillo);
imagecharup ($im, 4, 60, 60, $t2, $amarillo);
imagecharup ($im, 5, 80, 80, $t2, $amarillo); imagepng($im);
imagedestroy($im);
?>
```

También podemos utilizar la siguiente función:

`imagestring ($im, n, x, y, $txt, $col)`

Esta función se comporta de forma muy similar a `imagechar` siendo la única diferencia entre ambas que mientras `imagechar` inserta solo el primer carácter, en el caso de `imagestring` se inserta la cadena completa.

Los parámetros de ambas funciones son los mismos. Si la cadena desborda los límites de la imagen solo se visualizará la parte de la misma contenida dentro de estos.

```
<?php
```



```
header("content-type: image/png");
$im = imagecreate(150,150);
$t1="tamaño 1";
$t2="tamaño 2";
$t3="tamaño 3";
$t4="tamaño 4";
$t5="tamaño 5";
$fondo=imagecolorallocate ($im, 0, 0, 200);
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagestring ($im, 1, 10, 20, $t1, $amarillo);
imagestring ($im, 2, 10, 40, $t2, $amarillo);
imagestring ($im, 3, 10, 60, $t3, $amarillo);
imagestring ($im, 4, 10, 80, $t4, $amarillo);
imagestring ($im, 5, 10, 100, $t5, $amarillo); imagepng($im);
imagedestroy($im);
?>
```

La variante vertical sería:

```
imagestringup ($im, n, x, y, $txt, $col)
```

No requiere demasiadas explicaciones esta función. Inserta una cadena completa con orientación vertical y sus parámetros son idénticos a los comentados al hablar de imagecharup.

```
<?php
header("content-type: image/png");
$im = imagecreate(150,150);
$t1="tamaño 1";
$t2="tamaño 2";
$t3="tamaño 3";
$t4="tamaño 4";
$t5="tamaño 5";
$fondo=imagecolorallocate ($im, 0, 0, 200);
$amarillo=imagecolorallocate ($im, 255, 255,0);
imagestringup ($im, 1, 10, 100, $t1, $amarillo);
imagestringup ($im, 2, 20, 100, $t2, $amarillo);
imagestringup ($im, 3, 40, 100, $t3, $amarillo);
imagestringup ($im, 4, 60, 100, $t4, $amarillo);
imagestringup ($im, 5, 80, 100, $t5, $amarillo); imagepng($im);
imagedestroy($im);
?>
```

Tipos de letra

Las funciones anteriores utilizan siempre la fuente predefinida por PHP y solo permiten los cinco tamaños que hemos podido ver en los ejemplos.

Para utilizar tipos de letra y tamaños distintos PHP requiere tener instalada la librería Freetype.

Desde la página oficial de PHP nos remiten a aquí para conseguirla:

<http://www.freetype.org/>