

Creación de imágenes dinámicas

Una imagen dinámica es tan sólo un **script** que contiene las instrucciones para la creación de esa imagen.

Para visualizar una imagen dinámica desde una página web basta con invocar el fichero que contiene el script desde la etiqueta clásica de inserción de imágenes de HTML

```

```

donde **imgxx.php** será el nombre del script que genera la imagen.

Primera etiqueta

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

Trabajaremos con dos de ellos: **JPG** y **PNG**.

La primera instrucción que ha de contener cualquier script que deba generar imágenes ha de ser la siguiente:

```
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**.

¡Cuidado!

Cualquier etiqueta **header** (cabecera) ha de incluirse obligatoriamente **al comienzo del script** antes que ninguna otra instrucción y sin ninguna línea en blanco que la preceda.

Pondremos siempre estas instrucciones inmediatamente debajo de **<? sin que las separe ninguna línea en blanco**.

Creación de imágenes

Definida la etiqueta anterior tenemos que: **crear** la imagen, **dibujarla** y luego **enviarla** al navegador para que pueda ser visualizada y, por último, (no es imprescindible pero si muy conveniente) **borrarla**, con el fin de *liberar la memoria del servidor*

Una primera imagen

```
<?
    Header("Content-type: image/jpeg");

    $im = imagecreate(200,200);

    Imagejpeg($im);

    Imagedestroy($im);
?>
```

[Ver img1.php](#)

¡Cuidado!

No dejes NUNCA líneas en blanco entre la etiqueta **<?** de comienzo del script y la línea que contiene **Header**

Si escribiéramos el script anterior sustituyendo **image/jpeg** por **image/png** e **Imagejpeg(\$im)** por **Imagepng(\$im)** *no visualizaríamos nada*.

El formato **jpg** –a falta de especificaciones– considera la imagen con negro como color de fondo, pero **png** requiere que ese color sea especificado.

Añadir un color de fondo

```
<?
    Header("Content-type: image/jpeg");

    $im = imagecreate(200,200);

        $fondo=imagecolorallocate ($im, 0, 0, 200);

        Imagefill ($im, 0, 0, $fondo);

    Imagejpeg($im);

    Imagedestroy($im);
?>
```

[Ver img2.php](#)

```
<?
    Header("Content-type: image/png");

    $im = imagecreate(200,200);

        $fondo=imagecolorallocate ($im, 0, 0, 200);

        Imagefill ($im, 0, 0, $fondo);
    Imagepng($im);
    Imagedestroy($im);
?>
```

[Ver img3.php](#)

ocupada durante el proceso de generación de la misma.

Estas son las funciones PHP para esos procesos:

\$nom = imagecreate(anc,a)

Con esta función se **crea** una imagen del tamaño indicado en los parámetros **anc** y **al** (en **pixels**) que será **recogida** en la variable **nom**.

Esta función es idéntica para cualquier formato de imagen.

Envío de imágenes al navegador

Para enviar imágenes al navegador (visualización) se usan **funciones diferentes** según el tipo de imagen definida en **Header**.

Si pretendemos que la imagen tenga formato **JPG** habremos puesto en **Header** la indicación **jpeg** (*¡cuidado! observa la sintaxis... jpeg*). En este caso la función de visualización será:

Imagejpeg(\$nom)

Si se tratara de una imagen en formato **PNG** (recuerda que debe estar definido en Header) la sintaxis sería:

Imagepng(\$nom)

Eliminando imágenes de la memoria

Para borrar imágenes de la memoria del servidor (que no del navegador) se utiliza la siguiente sintaxis:

Imagedestroy(\$nom)

Creando colores

PHP permite crear una *paleta* de colores. Para ello se pueden crear **variables de color** (con independencia del formato utilizado) mediante la siguiente función:

\$color=imagecolorallocate(\$nom,R,G,B)

donde la variable recoge el color resultante de mezclar 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 tan sólo con utilizar nombres de variables distintos para cada uno de ellos.

Antes colores de fondo

Dibujar un rectángulo sin relleno

```
<?
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);
?>
```

[Ver img4.php](#)

```
<?
Header("Content-type: image/png");

$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);

Imagepng($im);

Imagedestroy($im);
?>
```

[Ver img5.php](#)

Dibujando un rectángulo con relleno

```
<?
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);
?>
```

[Ver img6.php](#)

Para aplicar un color de fondo a una imagen (no importa el tipo del formato) se utiliza la siguiente función:

Imagefill(\$nom,x,y,\$col)

Aquí **\$nom** es la *variable* que contiene 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) que se pretende aplicar a la imagen.

Mediante esta función todos los puntos **colindantes** con el de coordenadas **x,y** que tengan su mismo color serán rellenados con el color especificado en la variable **\$col**.

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** son las coordenadas del **vértice superior izquierdo** y **x1, y1** las coordenadas del **vértice inferior derecho** y **\$col** el color que pretendemos asignar a las líneas del rectángulo.

El punto (0,0) siempre es la esquina superior izquierda de la imagen y recuerda que si no usas –en las líneas– un color distinto al del fondo no se visualizará el rectángulo.

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.

Polígonos con relleno

Para colorear el fondo de un polígono son necesarias dos operaciones:

Crear un **array** con las coordenadas de cada uno de sus vértices.

Aplicar la función que dibuja polígonos de este tipo.

La creación del array podría hacerse así:

\$v=(x0, y0, x1, y1,... xn, yn)

donde se irían introduciendo las coordenadas de los **sucesivos vértices del polígono** (x e y de cada

```
<?
Header("Content-type: image/png");

$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);

Imagepng($im);

Imagedestroy($im);

?>
```

[Ver img7.php](#)

Dibujando un polígono relleno

```
<?
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);

?>
```

[Ver img8.php](#)

```
<?
Header("Content-type: image/png");

$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);

Imagepng($im);

Imagedestroy($im);
```

vértice).

La creación de un polígono de este tipo requiere la siguiente función:

imagefilledpolygon(\$nom, \$vert, nº vert, \$col)

donde **\$nom** es el nombre de la imagen, **\$vert** es el **array** que contiene las coordenadas de los vértices, **nº vert** es el número de vértices y **\$col** es el color de relleno.

Polígonos sin relleno

Su funcionamiento es idéntico al anterior en tanto 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. Sólo se modifica el nombre de la función que en este caso es:

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

Elipses, circunferencias y arcos

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

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** 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* y se expresa en *grados sexagesimales*.

Gf es el *punto final del arco* también en *grados sexagesimales*.

\$col es el *color* con el que se dibujará la línea.

Respecto a los ángulos, CERO GRADOS coincide con el *cerro trigonométrico* pero el sentido es *contrario*, es decir, el de las *agujas del reloj*.

Obviamente, para dibujar una circunferencia *basta* con hacer iguales los valores de **a** y de **b** y fijar los puntos inicial y final en 0° y 360° respectivamente.

Dibujando 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*, por ejemplo:

?>

[Ver img9.php](#)

Dibujando un polígono sin relleno

```
<?
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);

imagepolygon ($im, $esquinas, 4, $blanco);

Imagejpeg ($im);

Imagedestroy($im);
?>
```

[Ver img10.php](#)

```
<?
Header("Content-type: image/png");

$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);

Imagepng ($im);

Imagedestroy($im);
?>
```

[Ver img11.php](#)

Dibujando circunferencias, elipses y arcos

```
<?
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);
```

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

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

Una vez definida esta variable bastará sustituir la instrucción de *creación de imagen*

\$nom = imagecreate(x,y)

por:

\$nom= imagecreatefrompng (\$b)

en el caso de imágenes **PNG** o por:

\$nom= imagecreatefromjpeg (\$b)

si se tratara de imágenes en formato **JPG**.

El resultado del cambio en el primero de los casos es [este](#) y aquí tienes también la imagen obtenida utilizando [un fondo en formato JPG](#).

Una advertencia al respecto. Como puedes comprobar en los ejemplos, *el tamaño de la imagen es el mismo de la utilizada como fondo*.

Guardando imágenes

Las imágenes que son creadas mediante la sintaxis anterior *no se guardan en servidor*.

Si se pretende guardarlas hay que modificar la sintaxis de las etiquetas:

Imagepng(\$nombre)

o

Imagejpeg(\$nombre)

añadiendo otro parámetro con el nombre y la extensión del fichero que vamos de guardar. Así por ejemplo:

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

o

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

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

```
$amarillo=imagecolorallocate ($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);

?>
```

[Ver img12.php](#)

```
<?
Header("Content-type: image/png");
$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);

Imagepng($im);

Imagedestroy($im);

?>
```

[Ver img13.php](#)

Utilizando imágenes dinámicas

En todos los ejemplos anteriores hemos podido visualizar las imágenes con sólo llamarlas desde el navegador, de la misma forma que podríamos visualizar cualquier otra imagen.

Pero las imágenes dinámicas pueden ser insertadas en una página web de la misma forma que cualquier otra imagen.

Aquí tienes un ejemplo donde se recogen en una página web todas las imágenes dinámicas creadas anteriormente.

[ejemplo103.php](#)

Si observas el *código fuente* verás que es exactamente el mismo que se utiliza para insertar una imagen *normal*, con la única diferencia de que aquí **el nombre de la imagen** será el mismo que el del **fichero PHP** que la genera.

[Anterior](#)

[Índice](#)

[Siguiete](#)