

[Ver índice](#)

Imágenes con líneas y textos



Dibujando segmentos

Trazando 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 mediante *imagecreate*, **x0** e **y0** son las coordenadas de uno de los extremos; **x1** e **y1** son las coordenadas del otro extremo y **\$col** es la variable de color con el que será dibujada la línea.

Colores de fondo

Tal como puedes observar en los primeros ejemplos de esta página, PHP utiliza *como fondo de la imagen* 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 deseamos que un color determinado se comporte como si fuera transparente debemos utilizar la función:

imagecolortransparent (\$nom, \$col).

donde: **\$nom** es el nombre de la variable definida mediante *imagecreate*, y **\$color** es el color que pretendemos hacer transparente.

No olvides estos dos pequeños detalles:

- Si pretendes lograr un fondo transparente debes hacer transparente el primero de los colores definidos.
- Esta función *sólo tiene sentido* en imágenes **PNG** que son las únicas que *permiten zonas transparentes*. Recuerda que **JPG** no las permite.

Insertando textos

Para insertar textos dentro de una imagen hemos de recurrir a una de estas funciones:

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

Aunque presentaremos únicamente el código fuente de una de las imágenes, para hacer la comprobación de las funciones y las diferencias de visualización insertaremos dos ejemplos, uno en formato PGN y otro en JPG.

Recuerda que las únicas diferencias entre ambos radican en utilizar: *Header("Content-type: image/png")* o *Header("Content-type: image/jpeg")* y en las funciones *Imagepng* ó *Imagejpeg*.

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

[Ver img_a1.php](#)
Formato PNG[Ver img_a2.php](#)
Formato JPG

Fondos transparentes

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

[Ver img_a3.php](#)
Formato PNG[Ver img_a4.php](#)
Formato JPG

Aquí tienes una página -con un color de fondo- en la que puedes visualizar las diferencias entre los dos formatos.

[ejemplo104.php](#)

Insertando caracteres

```
<?
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);
```

Requiere que la variable **\$txt** contenga una **cadena** definida con anterioridad. Mediante esta función se inserta el *primer carácter de la cadena con orientación horizontal*.

Los parámetros de la función son los siguientes:

\$nom el nombre de la variable con la que fue 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*, el único que se verá en la imagen.

\$col es el color del carácter a representar.

imagecharup (*\$im, n, x, y, \$txt, \$col*)

Su funcionamiento es similar al de la función anterior, con la única diferencia de que *inserta el carácter con orientación vertical*.

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

imagestring (*\$im, n, x, y, \$txt, \$col*)

Esta función se comporta de forma similar a **imagechar**. La única diferencia entre ambas es que mientras **imagechar** inserta *sólo 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 sólo se visualizará la parte de la misma contenida dentro de éstos.

imagestringup (*\$im, n, x, y, \$txt, \$col*)

Inserta *una cadena completa* con *orientación vertical* y sus parámetros son idénticos a los comentarios cuando nos hemos referido a **imagecharup**.

Tipos de letra

Todas las funciones anteriores **utilizan siempre** la **fente predefinida** por PHP y sólo permiten los **cinco tamaños** que hemos

```
imagechar ($im, 4, 60, 60, $t2, $amarillo);
imagechar ($im, 5, 80, 80, $t2, $amarillo);
    Imagepng($im);
    imagedestroy($im);
?>
```

Ver img17.php
Formato PNG

Ver img18.php
Formato JPG

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

Ver img19.php
Formato PNG

Ver img20.php
Formato JPG

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

Ver img21.php
Formato PNG

Ver img22.php
Formato JPG

```
<?
    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);
```

podido ver en los ejemplos.

Afortunadamente –lo veremos en la página siguiente– PHP también permite usar **fuentes TrueType** y aplicarlas en la creación de imágenes.

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

Ver img23.php
Formato PNG

Ver img24.php
Formato JPG

Anterior
◀

Índice
▲

Siguiente
▶