



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Como generar imagenes dinamicamente con PHP (86935 lectures)

Per **Gabriel**, [Gigi](http://www.degabriel.net) (<http://www.degabriel.net>)

Creado el 13/06/2001 23:00 modificado el 13/06/2001 23:00

Muchas veces nos preguntamos como podriamos generar imagenes dinamicamente sin necesidad de disponer de estas imagenes. PHP dispone de librerias que facilitan la creacion dinamica de graficos de manera muy sencilla, veamoslo como hacerlo:

Es una buena idea este tipo de soluciones, ya que podremos crearnos botones dinamicamente simplemente cambiando la imagen, ya que el texto lo podemos añadir facilmente como veremos mas adelante.

Para empezar tendremos que tener actualizado PHP a la ultima versión, no es que sea necesario, pero es una buena excusa. Y usar la libreria GD para la creacion de imagenes dinamicas.

Para actualizarnos el PHP, y si disponemos de Red Hat 6.2, por ejemplo, podemos ir a una web fantastica y bajarnos lo archivos necesarios en: <http://rpms.arvin.dk/>⁽¹⁾

Con Debian no hay problema, solo que se actualizaria con el APT.

Estos son los ficheros que he utilizado en mi caso:

```
expat-1.95.1-2.arvin.i586.rpm
mod_php-4.0.5-2.arvin.i586.rpm
php-4.0.5-2.arvin.i586.rpm
php-devel-4.0.5-2.arvin.i586.rpm
php-imap-4.0.5-2.arvin.i586.rpm
php-ldap-4.0.5-2.arvin.i586.rpm
php-mysql-4.0.5-2.arvin.i586.rpm
php-pgsql-4.0.5-2.arvin.i586.rpm
php-standalone-4.0.5-2.arvin.i586.rpm
```

Y estos son los que necesitaremos para emplear a la hora de generar las imagenes, una es la propia libreria, las otras dos, son los modulos de PHP para emplearlas en la web:

```
gd-1.8.3-7.i586.rpm
php-gd-4.0.5-2.arvin.i586.rpm (*)
php-gd-with_gif-4.0.5-2.arvin.i586.rpm (**)
```

Si lo que queremos es emplear jpg y png usaremos (*)

Si lo que queremos es usar gif, usaremos(**)

Recuerda que son excluyentes, no podras emplearlos a la vez, o eliges gif, o, jpg y png

Lo que hacemos ahora es instalar los paquetes necesarios

```
[root@p90]# rpm -ivh gd-1.8.3-7.i586.rpm
[root@p90]# rpm -ivh php-gd-4.0.5-2.arvin.i586.rpm
```

Recuerda que si tienes una version anterior, de estos paquetes, tendras que desinstalarlas.

Empecemos con los ejemplos, en este caso vamos a usar imagenes PNG, pero basta con sustituir donde ponga PNG por JPEG y funcionaria igual:

1.- Cargar una imagen que ya tenemos



```
<?php
// En la cabecera HTTP indicamos que lo que devolvemos es una imagen
// de tipo PNG. Para ello utilizamos el tipo MIME: image/png
Header("Content-type: image/png");
// Creamos la imagen a partir de un fichero existente
$im = imagecreatefromPng("./boton.png");
// Mostramos la imagen
ImagePng($im);
// Liberamos la memoria que ocupaba la imagen
ImageDestroy($im);
?>
```

2.- Crar una imagen nueva

```
<?php
Header("Content-type: image/png");
//Creamos una imagen de 50x50
//Creamos el color
$image = ImageCreate(50, 50);
$red = ImageColorAllocate($image, 255, 0,0);
$blue = ImageColorAllocate($image, 0, 0, 255);
//Dibujamos un rectangulo en el interior de la imagen
ImageRectangle($image, 10, 10, 30, 30, $blue);
ImagePng($image);
ImageDestroy($image);
?>
```

3.- Rellenar de un color la imagen

```
<?php
Header("Content-type: image/png");
$im = ImageCreate(200, 200);
$red = ImageColorAllocate($im, 255, 0, 0);
//rellenamos de rojo el rectangulo
ImageFill($im, 100, 100, $red);
ImagePng($im);
?>
```

4.- Añadir texto a la imagen

```
<?php
Header("Content-type: image/png");
$im = ImageCreate(300, 300);
$red = ImageColorAllocate($im, 255, 0, 0);
$white = ImageColorAllocate($im, 255, 255, 255);
$blue = ImageColorAllocate($im, 0, 0, 255);
$black = ImageColorAllocate($im, 0, 0, 0);
ImageFill($im, 0, 0, $red);
// Escribimos el string en (210,30) en negro
//El 5 viene a ser el tamaño de la letra 1-5
ImageString($im, 5, 210, 30, "Hello John", $black);
// para calcular el grosor de la fuente
$font_width = ImageFontWidth(5);
// y calculamos la lingitud del strig
$string_width = $font_width * (strlen("Hello John"));
// y añadimos la linia de subrallado en (210,50) en negro
ImageLine($im, 210, 50, (210+$string_width), 50, $black);
ImagePng($im);
?>
```

5.- Crear un poligono asignando vertices

```
<?php
Header("Content-type: image/png");
$im = ImageCreate(300, 300);
$green = ImageColorAllocate($im, 0, 255, 0);
$white = ImageColorAllocate($im, 255, 255, 255);
$blue = ImageColorAllocate($im, 0, 0, 255);
```



```
// ponemos los vertices por pares, en este caso son 5 vertices
$vertices = Array(150, 10, 250, 100, 250, 200, 50, 200, 50, 100);
ImageFill($im, 0, 0, $white);
// dibujamos un pentagono
ImagePolygon($im, $vertices, 5, $blue);
// relenamos el pentagono de verde y lo bordeamos por azul
ImageFillToBorder($im, 150, 150, $blue, $green);
ImagePng($im);
?>
```

6.- Como crear una bateria de Botones

Esto puede venir bien a la hora de hacer botones para una web. Quizas en Bulma lo podamos emplear, pero eso ya veremos.

Para empezar creamos la pagina **prueba.php**, en el que generamos los botones con el siguiente codigo:

-----prueba.php-----

```
<?
// Creamos un array con los nombre de los botones
$menu_items = Array();
$menu_items[0] = "Lunes";
$menu_items[1] = "Martes";
$menu_items[2] = "Miercoles";
$menu_items[3] = "Jueves";
$menu_items[4] = "Viernes";
$menu_items[5] = "Sabado";
?>

<html>
<head>
<basefont face=arial>
</head>
<body>
<center><h2>Pulsa sobre una dia</h2></center>
<table border=0>

<?
// repite por cada Item
foreach($menu_items as $channel)
{
// y imprime el boton
?>

<tr>
<td>
<a href="index.php?channel=?=$channel?>">
</a>
</td>
</tr>

<?
}
?>

</table>
</body>
</html>
```

----- fin prueba.php-----

A continuacion creamos **boton.php**, que genera el texto que va en cada boton, y lo centramos segun el tamaño del boton y del texto.

----- boton.php -----

```
<?php
```



```
Header("Content-type: image/png");
$im = ImageCreate(120, 40);
$white = ImageColorAllocate($im, 255, 255, 255);
$blue = ImageColorAllocate($im, 0, 0, 255);
ImageFill($im, 0, 0, $blue);
// obtenemos las dimensiones de las fuentes
$font_height = ImageFontHeight(3);
$font_width = ImageFontWidth(3);
// obtenemos las dimensiones de la imagen
$image_height = ImageSY($im);
$image_width = ImageSX($im);
// obtenemos el tamaño del string
$length = $font_width * strlen($channel);
// calculaamos las coordenadas del texto del boton que este centrado
$image_center_x = ($image_width/2)-($length/2);
$image_center_y = ($image_height/2)-($font_height/2);
// escribo el texto del string que le pasamos
ImageString($im, 3, $image_center_x, $image_center_y, $channel, $white);
ImagePng($im);
?>
```

----- fin boton.php -----

Biografia:

- http://www.programacion.net/cursos/php4/php4_ap1.htm⁽²⁾
- http://www.devshed.com/Server_Side/PHP/ImageGeneration/⁽³⁾
- <http://www.boutell.com/gd/>⁽⁴⁾
- <http://rpms.arvin.dk/>⁽¹⁾

Lista de enlaces de este artículo:

1. <http://rpms.arvin.dk/>
2. http://www.programacion.net/cursos/php4/php4_ap1.htm
3. http://www.devshed.com/Server_Side/PHP/ImageGeneration/
4. <http://www.boutell.com/gd/>

E-mail del autor: gginard_ARROBA_gmail.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=671>