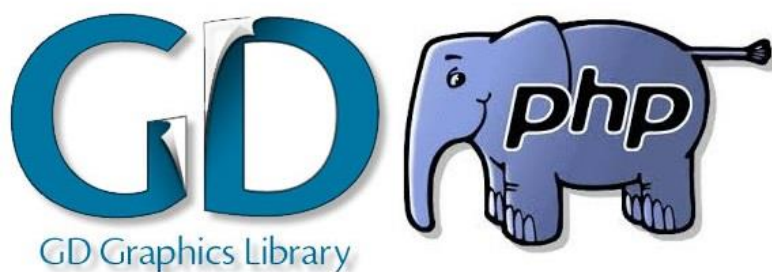


Convertir un texto en una imagen utilizando PHP



Hay ocasiones en las que podemos tener la necesidad de convertir algún texto en imagen de forma dinámica. Por ejemplo, una dirección de **correo electrónico** que aparezca en nuestra web, para evitar de esta forma que nuestra dirección sea rastreada por las arañas, reduciendo de esta forma el envío de correo basura. Convertir un texto en imagen es realmente sencillo utilizando el lenguaje de programación PHP, uno de los más utilizados en la actualidad para el desarrollo web. A lo largo de nuestro **White Paper** explicaremos el proceso para generar una imagen a partir de un texto enviado desde un formulario.

Librería GD de PHP

Para poder trabajar con imágenes, PHP dispone de la librería GD donde se dan cita todas las funciones necesarias para ello. Es por esto, que para poder convertir texto en imagen, es necesario asegurarnos de que está habilitada en nuestro **servidor**. De forma habitual, esto siempre ocurre, pero podemos verificarlo ejecutando la función “**phpinfo()**” de PHP. Esta función nos mostrará por pantalla la configuración que tiene la versión de PHP.

Directive	Local Value	Master Value
filter.default	unsafe_raw	unsafe_raw
filter.default_flags	no value	no value

ftp

FTP support	enabled
FTPS support	enabled

gd

GD Support	enabled
GD Version	bundled (2.1.0 compatible)
FreeType Support	enabled
FreeType Linkage	with freetype
FreeType Version	2.8.0
GIF Read Support	enabled
GIF Create Support	enabled
JPEG Support	enabled
libJPEG Version	9 compatible
PNG Support	enabled
libPNG Version	1.6.34
WBMP Support	enabled
XPM Support	enabled
libXpm Version	30512
XBM Support	enabled
WebP Support	enabled

Directive	Local Value	Master Value
gd.jpeg_ignore_warning	1	1

.. .

Como se puede ver en la imagen superior, nuestro servidor tiene instalada y habilitada la librería GD, por lo que podremos utilizar sus funciones sin problema. Si vuestro **alojamiento web** no cuenta con esta librería, hablad con vuestra empresa de hosting para ver la posibilidad de que la habiliten.

Formulario para el envío del texto a convertir

Como hemos comentado al principio, vamos a desarrollar un ejemplo donde tengamos un **formulario** que nos permite insertar el texto que queremos convertir a imagen. Será un formulario muy sencillo y su apariencia será la siguiente.

Convertor de texto a imagen

Texto a Convertir:

El código que nos permitirá crear este formulario sería el siguiente.

Código HTML

```
<h2>Convertor de texto a imagen</h2>
<form name="form" id="form" method="post" action="index2.php"
enctype="multipart/form-data" onsubmit="return validateForm();">
  <div class="form-row">
    <div>
      <label>Texto a Convertir:</label> <input type="text"
        class="input-field" name="txt_input" maxlength="50">
    </div>
  </div>

  <div class="button-row" style="margin-top:20px;">
    <input type="submit" id="submit" name="submit" value="Convertir">
  </div>
</form>
<div id="validation-response"></div>
```

No hay mucho que explicar del código anterior, ya que se trata de un formulario sencillo. Lo más destacable es última línea del código:

```
<div id="validation-response"></div>
```

Esta línea la utilizaremos para mostrar un mensaje en el caso de que se pulse el botón **“Convertir”** y no se haya escrito ningún texto. Esta validación la haremos mediante JavaScript cada vez que se pulse el botón. En ese momento, se hará la llamada al método **“validateForm()”** que se encargará de revisar si hay o no contenido en la caja de texto. El código que hará todo esto es el siguiente:

Código JavaScript

```
function validateForm() {  
    var valido = true;  
    document.getElementById("validation-response").style.display = "none";  
    document.getElementById("validation-response").value = "";  
  
    if(document.form.txt_input.value == "") {  
        document.getElementById("validation-response").style.display = "block";  
        document.getElementById("validation-response").innerHTML = "El campo  
texto no debe estar vacío";  
        valido = false;  
    }  
  
    return valido;  
}
```

En esta función lo primero que hacemos es declararnos la variable “**valido**” con el valor “**true**”, variable que se devuelve al final del método. Si su valor es “**true**”, entonces significa que el campo texto no está vacío y que se puede convertir a imagen. Si por el contrario, su valor es “**false**”, entonces no se ha introducido ninguna cadena de texto y el programa mostrará un mensaje de error.

El código que se encarga de comprobar si el campo de texto está vacío es el bloque siguiente:

Código JavaScript

```
if(document.form.txt_input.value == "") {  
    document.getElementById("validation-response").style.display = "block";  
    document.getElementById("validation-response").innerHTML = "El campo  
texto no debe estar vacío";  
    valido = false;  
}
```

Si el campo está vacío se recupera la referencia hacia el “**div**” donde se mostrará el mensaje de error y se rellenará con el texto del error, tal y como se puede ver en la imagen siguiente:

Convertor de texto a imagen

Texto a Convertir:

El campo texto no debe estar vacío

Función PHP para convertir

Visto lo anterior, ahora es el momento de crear el código PHP que se encargará de convertir el texto en imagen. Para ello, lo primero que haremos será crearnos el lienzo donde se pintará el texto. Para ello hacemos uso de la función “**imagecreate**” a la que le pasaremos como variables el ancho y el alto de la imagen.

```
$img = imagecreate(500, 100);
```

El siguiente paso será indicar el color de fondo y el color del texto de la imagen. Esto lo haremos con el método “**imagecolorallocate**” que proporciona PHP. A este método se le pasa como primer argumento el lienzo creado y a continuación el color que queremos poner en formato “**RGB**”.

```
$textbgcolor = imagecolorallocate($img, 185, 181, 181);  
$textcolor = imagecolorallocate($img, 0, 0, 0);
```

En nuestro ejemplo hemos elegido un color gris (185,181,181) para el fondo y el negro (0,0,0) para el texto. Siguiendo con el código, lo siguiente que haremos es recuperar el texto que se ha escrito en la caja de texto:

```
$txt = $_POST['txt_input'];
```

Una vez que tenemos el texto almacenado en la variable “**\$txt**” es el momento de pintarla dentro de nuestro lienzo. Esto lo haremos haciendo uso del método “**imagestring**” y que tiene la siguiente estructura:

```
imagestring ( resource $image , int $font , int $x , int $y , string $string , int $color )
```

El significado de cada uno de los parámetros es el siguiente:

- **\$image**: Imagen donde queremos escribir el texto
- **\$font**: Puede tomar valores del 1 al 5 y sirve para indicar el tamaño de la fuente
- **\$x**: Coordenada x de la esquina superior izquierda
- **\$y**: Coordenada y de la esquina superior izquierda
- **\$string**: Cadena de texto a escribir
- **\$color**: Color del texto

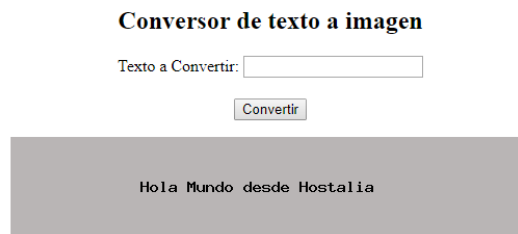
En nuestro ejemplo, el código sería el siguiente:

```
imagestring($img, 5, 15, 15, $txt, $textcolor);
```

Por último activamos el almacenamiento en búfer de la salida, imprimimos la imagen que hemos creado en formato PNG y por último, pintamos la imagen en el **navegador**:

```
ob_start();
imagepng($img);
printf('<center></center>', base64_encode(ob_get_clean()));
```

El resultado final sería algo como lo que se puede ver en la siguiente imagen.



El código completo de PHP sería el siguiente:

Código PHP

```
if (!empty($_POST['txt_input'])) {
    $img = imagecreate(500, 100);

    $textbgcolor = imagecolorallocate($img, 185, 181, 181);
    $textcolor = imagecolorallocate($img, 0, 0, 0);

    if ($_POST['txt_input'] != '') {
        $txt = $_POST['txt_input'];
        imagestring($img, 5, 125, 40, $txt, $textcolor);
        ob_start();
        imagepng($img);
        printf('<center></center>',
base64_encode(ob_get_clean()));
    }
}
```