

Fijación de márgenes

Algunas funciones de inserción de textos permiten colocar los mismos dentro de lo podría llamarse *caja de texto* que no es otra cosa que la superficie de la página delimitada por los márgenes de la misma.

Por defecto, estos márgenes miden **un centímetro**, por la izquierda, la parte superior y la derecha del documento. El tratamiento del margen inferior requiere una función específica que veremos en otro de los epígrafes de esta página.

\$obj->SetMargins(iz,su,de)

Los parámetros **iz**, **su** y **de** son los valores numéricos que permiten establecer, -en la unidad de medida que se indique en el *constructor*- los márgenes izquierdo, superior y derecho, por este orden, del documento.

Si se omite el tercer parámetro (margen derecho) se le asignará el valor que se haya indicado para el izquierdo.

Margen inferior y salto de página

La función:

\$obj->SetAutoPageBreak(a,b)

permite activar o desactivar el salto de página automático y establecer el margen inferior (mínimo) de las páginas. El parámetro **a** puede tomar dos valores: **1** ó **0** (cierto ó falso), que se corresponden con la condición **activo/inactivo**.

Cuando está **activo** se van añadiendo nuevas páginas, de forma automática, hasta que se incluya la totalidad del texto especificado. En el caso de estar **inactivo**, los textos que excedieran los límites de la página no se visualizarían ya que no se añadiría una nueva página.

Por defecto esta opción está **activada**.

El segundo parámetro, el valor **b**, permite indicar el margen inferior mínimo de la página. Su valor -numérico- deberá indicarse en las unidades establecidas por el *constructor*. Si no se indica asumirá, por defecto, un valor igual a **dos centímetros**.

Color del texto

Inserción de textos

La forma más simple de inserción de textos -poco utilizada por el número de inconvenientes que conlleva- es la que utiliza la función:

\$obj->Text(x,y,texto)

donde **x** y **y** son los valores numéricos que representan las coordenadas del punto de la página en el que pretende realizarse la inserción y **texto** es una cadena (o una variable conteniendo una cadena) con el texto que se pretende insertar.

Esta función no considera ni los eventuales *saltos de línea* que pudiera haber en la cadena (inserta todo el texto en una sola línea) ni tampoco los márgenes del documento. Por ello puede ocurrir, tal como puedes comprobar en el ejemplo, que, por desbordar los márgenes del documento, no aparezcan los textos completos.

Una función mucho más útil es la siguiente:

\$obj->Write(interlinea,texto)

en la que **interlinea** es el interlineado (expresado en la misma unidad utilizada por el constructor) y **texto** la cadena o variable que contiene el texto a insertar.

Se comporta de la siguiente forma:

Comienza a escribir el texto en la **posición actual** de *punto de inserción*.

La posición actual del punto de inserción. puede ser una de las siguientes:

La esquina superior izquierda del área de impresión de la página (en el caso en que se ha producido un salto de página y no se hubieran insertado elementos anteriormente).

La posición resultante del final de la inserción del elemento anterior.

La posición establecida como posición actual mediante las funciones **SetX**, **SetY** ó **SetXY**

La inserción de textos mediante la función **Text()** no modifica la localización del punto de inserción ya que utiliza sus propias coordenadas de posicionamiento.

Una vez que la línea alcanza el margen derecho -establecido de forma explícita o por defecto- de la caja de texto produce un salto de línea automático.

Incluye los saltos de línea que encuentra en el propio texto (caso, por ejemplo, de inclusión de ficheros de texto) y también aquellos incluidos en la propia cadena mediante la secuencia de escape **\n**.

Si la opción **AutoPageBreak** (comentada al margen) está activada produce un salto de página automático al alcanzar el margen inferior de la página.

Alinea los textos a la izquierda sin permitir modificar la configuración de esta opción

```
<?
#incluimos el fichero con la clase y definimos la variable FPDF_FONTPATH
# con el mismo criterio comentado en el ejemplo anterior
include("fpdf.php");
define('FPDF_FONTPATH',$ _SERVER['DOCUMENT_ROOT'].'/cursophp/fontspdf/');
/* establecemos las dimensiones del documento
   creamos un nuevo objeto y añadimos una página*/
$dimensiones=array (140,200);
$miPDF=new FPDF('P','mm',$dimensiones);
# ajustamos al 100% la visualización
$miPDF->SetDisplayMode('real');

#insertamos una página en blanco
$miPDF->Addpage();
# establecemos el color de la letra
$miPDF->SetTextColor(255,0,0);
```

El color del texto se establece utilizando la siguiente función:

\$obj->SetTextColor(R,G,B)

donde R, G, B son valores numéricos comprendidos entre **0** y **255** y que representan las respectivas componentes de los colores primarios: **rojo**, **verde** y **azul**. Si se utiliza con un sólo parámetro:

\$obj->SetTextColor(K)

el valor numérico, también comprendido entre **0** y **255**, establece colores de la *escala de grises* desde el negro (cero) hasta el blanco (255).

Si no se especifica color los textos aparecerán en negro.

Una vez establecido un color su valor se mantiene hasta que no sea invocada nuevamente la función **SetTextColor** para su modificación.

Tipo de letra, estilo y tamaño

Podemos asignar estos tres valores mediante la función:

\$obj->SetFont(nom,est,tam)

donde **nom** es una cadena que debe indicar el nombre del tipo de letra elegido (**Helvetica**, **Arial**, **Times**, **Zapfdingbats**, **Courier** ó **Symbol** son las fuentes por defecto).

El estilo (**est**) puede indicarse mediante la *cadena vacía* (""), para el tipo **normal** o mediante **B**, **I**, **U** (**negrita**, **cursiva**, **subrayada**) o sus combinaciones, por ejemplo **BU** para **negrita subrayada**.

Las fuentes de símbolos sólo permiten el estilo **normal**. El parámetro **tam** especifica el tamaño de la letra expresado en **puntos**. Si no se especifica, el valor por defecto es 12 puntos.

Más adelante, dedicamos un apartado a la creación y utilización de nuevas fuentes tipográficas.

Posición actual

La clase FPDF dispone de funciones que determinan el valor actual de las coordenadas del punto de inserción (lugar donde se va a posicionar el texto, imagen o elemento gráfico que se incluya) y otras que permiten modificar esos puntos de inserción.

Son las siguientes:

\$obj->GetX()

y

\$obj->GetY()

que devuelven los valores la

```
# establecemos la fuente a utilizar, estilo
# y tamaño en puntos. Al no incluir estilo
# considerará el estilo normal
$miPDF->SetFont('Arial','',11);
# creamos una variable en la que incluimos un texto
$texto="Este es el texto que tendremos que escribir aquí dentro.\n";
$texto.="Esto habría de ir detrás de un salto de línea. Veremos si funciona";
$texto.="Estamos alargando la línea para comprobar el funcionamiento de ";
$texto.="las diferentes funciones";
# usamos el método Text para presentar el contenido de la variable
# situamos las coordenadas de inserción en el punto (30,40) mm.
# podremos observar que la función Text inserta todo el texto en
# una línea sin tener en cuenta si sobrepasa o no el borde del papel
# ni tampoco los caracteres de salto de línea
$miPDF->Text(30,40,$texto);
# cambiamos el color de la letra. Al pasar un solo parámetro
# será interpretado como escala de grises 0 -255 (0 negro, 255 blanco)
$miPDF->SetTextColor(200);
# cambiamos el tipo de letra (Helvetica y Arial son alias de la misma fuente)
$miPDF->SetFont('Helvetica','B',11);
# utilizaremos la función Write para escribir el texto. Lo hará con
# alineación a la izquierda, interpretando los saltos de línea
# y efectuando uno automático cuando alcanza el margen.
# El valor de la interlínea se expresa en milímetros.

# Dado que la única inserción ha sido mediante Text() que no modifica el
# punto de inserción el resultado de este Write se incluirá al comienzo
# de la página, respetando los márgenes por defecto.
$miPDF->Write(4,$texto);

# Comprobaremos los tipos de letra existentes en el directorio fonts.
$miPDF->SetFont('Symbol','',11);
/* insertamos un salto de línea de 8 unidades que producirá
un desplazamiento del punto de inserción hasta el margen izquierdo
de una línea situada 8 unidades por debajo de la posición del final
de la cadena insertada anteriormente */
$miPDF->Ln(8);
# escribimos el texto anterior con la letra Symbol
$miPDF->Write(4,$texto);
# cambiamos nuevamente el tipo de letra
$miPDF->SetFont('zapfdingbats','',11);
# insertamos un nuevo salto de párrafo de 8 unidades
# y repetimos la impresión del texto con la nueva tipografía
$miPDF->Ln(8);
$miPDF->Write(4,$texto);
# establecemos márgenes (izquierda, superior, derecha)
# expresados en la unidad usada por el constructor (mm).
# Serán utilizados en las páginas siguientes
$miPDF->SetMargins(30,40,25);
# establecemos el modo de inserción automática de página
# indicando el margen inferior mínimo. El salto automático
# no sería necesario ya que, si no se especifica, se configura
# por defecto como tal
$miPDF->SetAutoPageBreak(1,8);

# Insertamos una nueva página. En adelante
# los contenidos aparecerán en esa nueva página
$miPDF->AddPage();
# cambiamos el color y tipo de letra
$miPDF->SetTextColor(0,0,200);
$miPDF->SetFont('Times','I',12);
# insertamos texto mediante Write y observaremos
# que ahora respeta los nuevos márgenes
$miPDF->Write(4,$texto);
# posicionamos el punto de inserción en las coordenadas
# (2,2) y escribimos de nuevo el texto.

$miPDF->SetXY(2,2);
# al visualizar el documento podremos observar que solo
# respeta estas coordenadas la primera línea, ya que las siguientes
# se ajustan a los márgenes establecidos para la página
$miPDF->Write(4,$texto);
# redimensionamos nuevamente los márgenes
$miPDF->SetMargins(20,5,15);

# cambiamos a una nueva página
$miPDF->AddPage();
```

posiciones actuales, horizontal (X) y vertical (Y). El resultado estará expresado en las unidades establecidas por el *constructor* considerando como origen de coordenadas la esquina superior izquierda del documento y valores positivos -horizontal y vertical- los colocados a la derecha (X) y debajo (Y) del origen.

Para definir una nuevo punto de inserción se pueden usar cualquiera de estas funciones:

```
$obj->SetX(valor X)  
$obj->SetY(valor Y)  
$obj->SetXY(valor X, valor Y)
```

que permiten establecer el *punto de inserción* mediante sus coordenadas: horizontal, vertical o ambas.

Cuando el valor X (en el caso SetX) es negativo se considera con referencia al margen derecho de la página. De igual modo, cuando se asigna un valor negativo en SetY se considera respecto a la parte inferior de la página.

Saltos de línea

Mediante *\$obj->Ln (valor)* se ejecuta un salto de línea. La posición actual regresa al margen izquierdo y la vertical se desplaza en el número de unidades indicado en el parámetro *valor*.

```
# cambiamos el estilo de fuente a "normal"  
$MiPDF->SetFont('Times','',12);  
# leemos el fichero de texto y lo recogemos en una variables  
    $f1=fopen('regenta.txt','r');  
    $regental=fread($f1,filesize('regenta.txt'));  
    fclose($f1);  
# insertamos el fichero mediante write  
# podremos observar como se realiza el salto de página  
# de forma automática respetando los márgenes inferiores  
$MiPDF->Write(4,$regental);  
  
# inseramos una nueva página  
$MiPDF->Addpage();  
# desactivamos el salto de página automatico  
# con lo cual no será respetado el margen inferior  
# y el texto rebasará ahora los límites del papal  
$MiPDF->SetAutoPageBreak(0);  
#ponemos el texto en negro y cursiva para poder  
#diferenciar los resultados al visualizar el documento  
$MiPDF->SetTextColor(0);  
# reescribimos el texto anterior  
$MiPDF->Write(4,$regental);  
# visualizamos el documento  
$MiPDF->Output();  
?>
```

ejemplo137.php

Anterior



Índice



Siguiente

