

Textos recuadrados

La función **Cell** ofrece un montón de posibilidades a la hora de insertar *celdas* conteniendo textos. Su sintaxis es la siguiente:

\$Obj->Cell(a,h,text,b,q,p,r,e)

dónde los parámetros son los siguientes: **a** es el ancho de la celda; **h** su altura; **text** la variable o cadena que contiene el texto a insertar; **b** un parámetro que establece los bordes de la celda y que puede tomar los siguientes valores:

Valor	Opción
0	Sin bordes
1	Con bordes
L	Borde por la izquierda
T	Borde superior
R	Borde por la derecha
B	Borde inferior

El valor por defecto es cero. Las especificaciones de bordes laterales pueden agruparse (LR, TL, etcétera) sin que importe el orden en que se haga la agrupación).

El parámetro **q** permite establecer en qué posición se habrá de insertar el elemento posterior a la celda actual (otra celda, un texto mediante **Write**, un gráfico, etcétera). Permite los siguientes valores:

Valor	Opción
0	A la derecha de la celda actual
1	En el margen izquierdo de la línea siguiente
2	Debajo de la celda actual alineado a su borde izquierdo

Por medio del parámetro **p** se puede establecer la alineación del texto (contenido en la celda) respecto a sus bordes. Permite usar **L**, **C** y **R** como indicadores de alineaciones del texto a la izquierda, centro ó derecha.

El parámetro **r** es un valor booleano (**1** ó **0**) que especifica si la celda ha de rellenarse con un **color de fondo** (**1**) o ha de ser **transparente** (**0**).

El parámetro **e** -es opcional- permite establecer la celda como un

Enlaces internos

En el ejemplo que tienes a continuación hemos insertado algunas de las opciones de las funciones comentadas al margen. En los párrafos siguientes incluimos algunos comentarios sobre la manera de utilizar las funciones de inclusión *enlaces internos* en el documento. Esta opción requiere tres pasos: Crear una referencia interna y recogerla en una variable identificadora.

La sintaxis sería la siguiente: **\$enlace=\$Obj->AddLink()**

Definir la zona del documento (área, imagen o texto) en la que, al pulsar sobre ella, se activará el redireccionamiento. Podría hacerse de varias formas, entre otras:

\$Obj->Cell(a,h,text,b,q,p,r,\$enlace)

Mediante esta opción se añade como último parámetro de la función **Cell** la variable creada mediante **AddLink()**. Si se tratara de un enlace a una URL no sería necesario **AddLink** y bastaría con incluir -entre comillas- la dirección de la URL. Convertiría la cadena de texto en un *hiperenlace*.

\$Obj->Link(X,Y,ancho,alto,\$enlace)

Permite definir como *hiperenlace* un área de la página actual definida por un rectángulo cuya esquina superior izquierda está definida por los parámetros **X** y **Y** y cuyo ancho y alto se establecen a través de los parámetros del mismo nombre. La variable **\$enlace** se comporta de forma idéntica a la indicada en el caso anterior.

\$Obj->Write(interlinea,'texto',\$enlace)

En este caso, añadimos un tercer parámetro a la función **Write** (es opcional) que contiene la variable indicadora del enlace. El comportamiento sería idéntico a los supuestos anteriores.

Establecer la posición a la que redigiría el enlace. Será necesario indicar **a dónde** habrá de redirigirse *el cliente* en el momento que se pulse sobre uno de los enlaces anteriores. Eso se indica mediante:

\$Obj->SetLink(\$enlace, pos_vertical, pagina)

La variable **\$enlace** será el indicador mencionado en los párrafos anteriores, **pos_vertical** la distancia del margen superior de la página de destino en la que se inicia la visualización y **pagina** el **número de la página** a la que se redirige mediante el enlace. Si se omiten estos parámetros se entenderá que la posición vertical es 0 y que la página es la página actual.

```
<?
#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 (210,297);
$miPDF=new FPDF('P','mm',$dimensiones);
# ajustamos al 100% la visualización
$miPDF->SetDisplayMode('real');

#insertamos una página en blanco
$miPDF->Addpage();
# estableceremos los colores para bordes, fondos y textos
# color de borde
$miPDF->SetDrawColor(255,0,0);
# color del relleno (gris)
$miPDF->SetFillColor(200);
# color del texto
$miPDF->SetTextColor(0,0,255);
# establecemos espesores de líneas y tipo y tamaño de letra
# espesor de línea 1 milímetro
$miPDF->SetLineWidth(1);
# fuente y tamaño: Arial, negrita de 12 puntos
$miPDF->SetFont("Arial","B",12);
# establecemos el texto para la primera celda
$celda1="Esto irá en la celda 1";
```

Si se especifica una URL se establecerá un enlace a la misma y, además, es posible establecer **enlaces internos** de la forma que puedes ver comentada en el ejemplo.

Tamaño de una cadena de texto

La función:

\$obj->GetStringWidth('cad')

devuelve un número decimal que indica la longitud de la cadena indicada en **cad** con la fuente y tamaño de letra actuales. Es una función muy útil para dimensionar las celdas de modo que los textos que contienen no sobrepasen sus bordes.

Espesor de líneas

Puede especificarse el grosor de las líneas -en los elementos gráficos que las utilizan(rectángulos, celdas, etcétera)- mediante la función:

\$obj->SetLineWidth('grosor')

dónde **grosor** es un valor numérico que especifica el espesor de las líneas en las unidades asignadas en el constructor. Por defecto -si no se asigna espesor mediante esta función- el ancho de línea sería de 0,2 milímetros.

Números de página

La función:

\$obj->PageNo()

devuelve el número de la página actual.

Color de las líneas

También es posible establecer el color de las líneas mediante la función:

\$obj->SetDrawColor('R,G,B')

dónde **R,G,B** son valores numéricos comprendidos entre **0** y **255** que asignan los valores de las componentes de los colores primarios rojo, verde y azul respectivamente. Si se incluye un único parámetro será interpretado como escala de grises. El valor 0 sería el negro y 255 representaría el blanco.

Color de relleno

Algunos elementos gráficos (celdas de texto, rectángulos) permiten que se les aplique un color de fondo. Para establecer tal color basta con ejecutar la función:

\$obj->SetFillColor('R,G,B')

El criterio de asignación de colores

```
# determinamos el tamaño de esta cadena y lo recogemos
# en la variable ancho
$ancho=$MiPDF->GetStringWidth($celda1);
/* definimos la celda estableciendo:
    ancho--- igual al de la cadena que va a contener + 6 mm.
    alto --- 6 milímetros
    texto--- el contenido de la variable $celda1
    borde--- 1 (para que ponga los cuatro bordes
    celda siguiente--- 0 (para que la incluya a continuación de la actual)
    alineación --- C para que centre el texto en la celda horizontalmente
    relleno --- 1 para que aplique el fondo a la celda
    enlace--- pondremos un enlace al buscador google */

# como aun no hemos insertado ningún elemento en la página
# la celda aparecerá en la parte superior de la página y apoyada
# sobre su borde izquierdo

$MiPDF->Cell($ancho+6,6,$celda1,1,0,C,1,"http://www.google.com");

/* pero modificando insertamos una nueva celda con el mismo contenido
    modificando:
    bordes--- ahora los pondremos solo por la parte superior T e inferior E
    posicion de la siguiente celda: 2 (inmediatamente debajo)
    relleno --0 */
# la nueva inserción se realizará a la derecha de la anterior
# ya que así lo especificamos al definir allí la celda siguiente
$MiPDF->Cell($ancho+6,6,$celda1,TB,2,C,0,"http://www.google.com");

# cambiamos el color del borde y el del relleno
# color de borde
$MiPDF->SetDrawColor(255,255,0);
# color del relleno (gris)
$MiPDF->SetFillColor(0,255,255);

/* una nueva inserción de una celda similar modificando
    identica a la primera celda salvo el parámetro de posición
    de la celda siguiente que cambiamos a 1 para que la celda
    siguiente aparezca en el margen izquierdo del documento.
    La actual, aparecerá inmediatamente debajo de la anterior
    ya que se así se especificó en su Cell correspondiente */

$MiPDF->Cell($ancho+6,6,$celda1,1,1,C,1,"http://www.google.com");

# la misma función anterior que ahora debería aparecer en el margen izquierdo
# modificando la especificación para que inserte la potencial celda siguiente
# inmediatamente a la derecha

$MiPDF->Cell($ancho+6,6,$celda1,1,0,C,1,"http://www.google.com");

# insertamos un salto de línea de 10 mm.
$MiPDF->Ln(10);
# una nueva celda que por efecto del salto de línea
# perdería la referencia de situarse a la derecha de la anterior
# se desplazaría 10 unidades hacia abajo y se insertaría en el margen izquierdo
$MiPDF->Cell($ancho+6,6,$celda1,1,0,C,1,"http://www.google.com");

# modificaremos ahora el punto de inserción mediante SetXY a las coordenadas
# -100, -120 mm. valores negativos
# con lo cual las referencias serán al margen derecho y al inferior

$MiPDF->SetXY(-100,-120);

# vamos a crear un enlace interno mediante la función AddLink()
# y vamos a recoger el resultado en la variable $salta
$salta=$MiPDF->AddLink();
# estableceremos la referencia del enlace
$MiPDF->SetLink($salta,0,2);
# crearemos ahora una zona (transparente) de enlace
# será el rectángulo definido como parámetros en la la función Link
# es decir un rectángulo de 297mm. de ancho, 30 de alto
# que tendrá su esquina superior izquierda en el punto (0,40)
# este enlace nos llevará al lugar del documento
# que se especifique mediante SetLink($salta);
# que indiquemos
$MiPDF->Link(0,40,297,30,$salta);
# comprobemos que allí se inserta el texto mediante el cell correspondiente
$MiPDF->Cell($ancho+6,6,$celda1,1,0,C,1,$salta);

# insertemos un salto de página y comprobemos la función nº de página
```

es idéntico al del párrafo anterior.

Una vez asignado un color de relleno (lo mismo ocurre con los colores y espesores de línea) se mantiene a lo largo del documento en tanto no sea modificado por la ejecución del método con unos parámetros distintos.

Número de páginas del documento

La clase FPDF incluye la posibilidad de determinar el número total de páginas del documento. Para ello es preciso ejecutar (después de crear el objeto) la función:

\$obj -> **AliasNbPages**("idn")

donde **idn** es una palabra cualquiera. Si se omite se considerará su valor por defecto que es: **{nb}**.

Para imprimir este valor en el documento será suficiente incluir el identificador utilizado en una cadena de texto. Puedes verlo comentado en el ejemplo.

```
$MiPDF->Addpage();
#añadimos a la cadena de texto el valor del número de pagina actual
$MiPDF->Cell(50,6,"pagina n° ".$MiPDF->PageNo(),1,0,C,1);
# activamos el valor por defecto del señalados de número total de páginas
# al no incluir nada como parámetro de la funcion AliasNbPages();
# deberemos recurrir al señalador por defecto {nb}
$MiPDF->AliasNbPages();
# insertamos una nueva celda y añadimos a la cadena de texto
# anterior "de {nb}". Como puedes ver
# el señalador {nb} forma parte de la cadena como un texto más
$MiPDF->Cell(50,6,"pagina n° ".$MiPDF->PageNo()." de {nb}",1,0,C,1);

$MiPDF->Output();
?>
```

ejemplo138.php

Ejemplo de creación de tablas

A continuación incluimos un ejemplo de utilización de estas funciones para la creación de tablas a partir de un fichero de texto.

En este ejemplo, utilizaremos saltos de página manuales y -dada la dimensión del fichero de texto- tendremos la oportunidad de comprobar el tiempo de generación de un documento de más de 200 páginas.

En páginas posteriores veremos como confeccionar la misma tabla utilizando encabezados, pies de página e inserción de páginas automáticas.

Ver código fuente

Ver fichero de texto

Crear el PDF

Anterior



Índice



Siguiente

