

Encabezados, pies de página y saltos automáticos

Celdas con saltos de línea

La función **Cell**, estudiada en la página anterior, no interpreta los saltos de línea y, en el caso de que la cadena de texto sea más larga que el ancho de la celda, escribirá fuera de sus márgenes.

La clase FPDF dispone de la función:

Sobj->MultiCell(a,h,text,b,aln,r)

donde **a** es el ancho de la celda (si se indica cero ocupará hasta el margen derecho de la página), **h** es el alto de cada una de las celdas aunque, en la práctica, se comporta de forma idéntica al interlineado de un procesador de textos), **text** es la cadena de texto (o variable que lo contiene) a insertar.

b es un parámetro que puede valer: **0** (sin bordes); **1** (con bordes), y también: **L** (borde por el lateral izquierdo de la celda); **T** (borde por la parte superior), **R** (línea de borde en el margen derecho), **B** (línea de borde en la parte inferior) ó agrupaciones de estos últimos valores, en cualquier orden, tales como: **LR** ó **TB**, etcétera.

El parámetro **aln** indica la alineación horizontal que han de tener los textos y permite los valores: **L** (izquierda); **C** (centro); **R** (derecha) ó **J** (justificado).

Por último, el parámetro **r** (relleno) especifica si a las celdas se les aplicará (valor 1) un color de fondo o si se va a tratar (valor 0) de un fondo transparente.

El comportamiento de esta función tiene las siguientes particularidades:

- Inserta los saltos de línea contenidos en el fichero origen y los incluidos en la cadena de texto mediante la secuencia de escape `\n`.
- Inserta saltos de línea automáticos en el momento en que el texto alcanza el borde derecho de la celda.
- Si `AutoPageBreak` está activado inserta saltos de página automáticos en el momento en que el texto alcanza el margen inferior de la página.

Encabezados

```
<?
# include y define idénticos a los anteriores
include("fpdf.php");
define('FPDF_FONTPATH',$ _SERVER['DOCUMENT_ROOT'].'/cursophp/fontspdf/');
# creamos una clase extendida de la clase FPDF
class GranPDF extends FPDF {
    # incluimos la funcion Header (debe ser definida con este nombre)
    # que se ejecuta cada vez que se añade una página, sea en forma
    # manual o en forma automática.
    # Nos permite insertar los encabezados de todas las páginas del
    # documento
    function Header() {
        #insertamos un salto de línea de 2 mm. a partir
        #del margen superior
        $this->Ln(2);
        # establecemos color y estilo de letra del encabezado
        $this->SetTextColor(0,0,0);
        $this->SetFont("Times","I", 11);
        # establecemos una celda cuyo ancho es cero
        # de esta forma se extenderá hasta el margen derecho
        # ocupando toda la caja de texto de la página.
        # le ponemos unicamente brode inferior, texto centrado
        # sin relleno y le adjudicamos como parámetro de posición
        # de la celda siguiente 1 de forma que comience en el margen
        # izquierdo de una línea nueva
        $this->Cell(0,8,"La Regenta",'B',1,'C',0);
        # colocamos un salto de líneas de 3 milímetros para separar el
        # encabezado de los textos de la página
        $this->Ln(3);
    }
    # incluimos la funcion Footer (debe ser definida con este nombre)
    # para insertar pies de página cada vez que cree una página nueva
    function Footer() {
        $this->Ln(2);
        $this->SetTextColor(0,0,0);
        $this->SetFont("Arial","I", 9);
        # en este caso incluimos el número de página con un borde superior
        # de la celda, texto centrado y tambien activando la celda siguiente
        # de modo que se produzca un saldo de línea
        $this->Cell(0,5,"Página ".$this->PageNo(),'T',1,'C',0);
    }
}
# acabada la inserción de la clase extendida continuamos con el código
# establecemos las dimensiones del documento
$dimensiones=array (140,200);
# creamos un nuevo objeto pero ;cuidado! utilizaremos
# la clase extendida GranPDF
$miPDF=new GranPDF('P','mm',$dimensiones);
# ajustamos al 100% la visualización
$miPDF->SetDisplayMode('fullpage');
# añadimos la primera página del documento. La ejecución de esta función
# disparará la ejecución de las funciones Header() y Footer() de la
# clase extendida y, por tanto, incluirá en el documento los encabezados
# y pies de página allí establecidos
$miPDF->Addpage();
# cambiamos el estilo de fuente a "normal"
$miPDF->SetFont('Times','',12);
# leemos un 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 Multicell
# el ancho 0 establece que la celda ocupará desde el margen
# izquierdo hasta el derecho. La interlinea será de 4mm.
# el texto que se incluirá (con salto de línea automático
# e inserción automática de nuevas páginas) será el recogido
```

y pies de página

La clase FPDF contiene dos métodos (funciones) llamados: **Header()** y **Footer()**.

Si editamos el fichero fpdf.php podemos ver que ambas están vacías (no contienen ninguna instrucción) presentando una sintaxis como esta:

```
function Header(){  
    // comentario  
}  
y  
function Footer(){  
    // comentario  
}
```

Ambas son invocadas de forma automática cada vez que se ejecuta la función **AddPage()** y, eso permite crear otras funciones *a medida*, con igual nombre, e incluirlas en una **clase extendida**.

Recuerda que los métodos de la clase extendida prevalecen sobre los que pudieran existir en la clase padre con su mismo nombre. Por esta razón, las nuevas funciones de la clase extendida nos permitirían incluir encabezados y pies de página en cada una de las nuevas páginas añadidas mediante **AddPage()**.

El procedimiento sería este:

- Creamos una clase extendida de la clase original **FPDF**.

- Incluimos en esta nueva clase funciones con nombres **Header()** y **Footer()** (no podemos modificar el nombre ya que han de coincidir con los nombres de las funciones vacías de la clase padre) en las que indicamos los contenidos que han de insertarse (y su posición) mediante cada una de ellas. • Al crear el nuevo objeto mediante:

\$obj= new nombre_clase()

utilizaremos el nombre el de la **clase extendida** que hemos creado para incluir estas funciones.

De esta forma, cada vez que se ejecuta **AddPage()** (añadir una nueva página) se ejecutarán también **Header()** y **Footer()** que, al no ser ya funciones vacías, realizarán una inclusión de contenidos en cada una de las páginas del documento.

Gestión de la inserción automática de páginas

La clase FPDF contiene esta función:

```
function AcceptPageBreak(){  
    // comentario  
    return $this->AutoPageBreak;  
}
```

que igual que ocurría con **Header()** y

```
# del fichero de texto. La celda no tendrá bordes  
# el texto estará justificado y no tendrá relleno de fondo.  
$miPDF->Multicell(0,4,$regenta,0,'J',0);  
# establecemos la visualización del documento  
$miPDF->Output();?>
```

ejemplo141.php

El ejemplo siguiente es una adaptación del **ejemplo139** al uso de encabezados y pies de página.

Ver código fuente

ejemplo142.php

```
<?  
# incluimos la clase fpdf.php y la constante FPDF_FONTPATH  
include("fpdf.php");  
define('FPDF_FONTPATH',$ _SERVER['DOCUMENT_ROOT'].'/cursophp/fontspdf/');  
# creamos la clase GranPDF extendida de fpdf.php  
class GranPDF extends FPDF {  
    # definimos dos nuevas variables internas  
    # el identificador de la columna actual  
    # y el valor de la ordenada de inicio de las columnas  
    var $columna_actual=0;  
    var $ordenada_inicio=21;  
    # la función Header se comporta de forma idéntica al ejemplo anterior  
    function Header(){  
        $this->SetTextColor(0,0,0);  
        $this->SetFont("Times","I", 11);  
        $this->Cell(0,8,"La Regenta",'B',1,'C',0);  
        $this->Ln(3);  
    }  
    # Footer es similar al caso anterior con la única diferencia que  
    function Footer(){  
        $this->SetY(-20);  
        $this->Ln(2);  
        $this->SetTextColor(0,0,0);  
        $this->SetFont("Arial","I", 9);  
        $this->Cell(0,5,"Página ".$this->PageNo(),'T',1,'C',0);  
    }  
    function AcceptPageBreak(){  
        # al alcanzar el margen inferior se activa esta función  
        # que comprueba si el punto de inserción está en la columna  
        # primera (0) ó en la segunda  
        if($this->columna_actual==0){  
            # cambia el valor del número de columna a 1 (pasa a la segunda)  
            $this->columna_actual=1;  
            # cambia el punto de inserción a la coordenada de inicio  
            # que es la parte superior de la página  
            $this->SetY($this->ordenada_inicio);  
            # cambia el margen izquierdo de modo que el texto  
            # aparezca en la columna derecha (a partir de los 72.5 mm).  
            # más abajo está comentada la razón de estos valores  
            $this->SetLeftMargin(72.5);  
            # cambia el punto de inserción al nuevo margen izquierdo  
            $this->SetX(72.5);  
            # establecemos que la función devuelva false  
            # con ello no se produce un salto de página  
            # y el punto de inserción se traslada a las  
            # coordenadas indicadas en las instrucciones anteriores  
            return false;  
        }else{  
            # si la columna es la segunda (no es la cero)  
            # cambia a la primera, asigna el valor cero  
            $this->columna_actual=0;  
            # cambia el punto de inserción a la coordenada de inicio  
            # que es la parte superior de la página  
            $this->SetY($this->ordenada_inicio);  
            # cambia el margen izquierdo de modo que el texto  
            # aparezca en la columna izquierda (a partir de los 10 mm).  
            $this->SetLeftMargin(10);  
            # cambia el punto de inserción al nuevo margen izquierdo  
            $this->SetX(10);  
            # establecemos que la función devuelva true  
            # con ello genera un salto de página  
            # y la inserción de texto continua en la página siguiente
```

