# Crear PDF en PHP con FPDF

### 1.- [Instalación y primeros pasos para generar PDF en PHP con FPDF](https://desarrolloweb.com/articulos/introduccion-fpdf.html)

FPDF es una clase para la generación dinámica de documentos PDF en PHP. Vamos a instalar y crear un primer ejemplo de uso de las librerías FPDF.

### 2.- [Funciones en FPDF](https://desarrolloweb.com/articulos/funciones-fpdf.html)

Vemos algunas de las funciones más importantes que nos podemos encontrar en FPDF.

### 3.- [Edición del pie y la cabecera de tu documento PDF con FPDF](https://desarrolloweb.com/articulos/pie-cabecera-fpdf.html)

Vamos a darle estilo a la cabecera y el pie de un archivos .pdf creado con FPDF

### 4.- [Función Image() en FPDF](https://desarrolloweb.com/articulos/funcion-image-fpdf.html)

Vemos la función Image(), para insertar imágenes en archivos PDF generados desde PHP por FPDF.

### 5.- [Enlaces en FPDF](https://desarrolloweb.com/articulos/enlaces-fpdf.html)

Vemos las distintas formas de realizar enlaces en FPDF.

### 6.- [Insertar archivos de texto con FPDF](https://desarrolloweb.com/articulos/insertar-archivo-txt-fpdf.html)

Mostramos el texto de un archivo txt en nuestro documento PDF con las funciones de FPDF.

### 7.- [Tablas en FPDF](https://desarrolloweb.com/articulos/tablas-fpdf.html)

Vemos cónstruir tablas para nuestros documentos en pdf con FPDF.

### 8.- [Columnas en FPDF](https://desarrolloweb.com/articulos/columnas-fpdf.html)

Vemos como maquetar nuestro archivo pdf a dos columnas.

# Instalación y primeros pasos para generar PDF en PHP con FPDF

**FPDF es una clase para la generación dinámica de documentos PDF en PHP. Vamos a instalar y crear un primer ejemplo de uso de las librerías FPDF.**

FPDF es una clase desarrollada en PHP para poder realizar documentos en PDF, dinámicamente a partir de nuestros scripts PHP. Esta clase trabaja de manera totalmente autónoma, por lo que no requiere utilizar la librería PDFlib ni cualquier otro producto similar.

Se trata de una clase que da bastante juego, ya que podemos modificar la unidad de medida, el formato de la página, los margenes, las cabeceras y los pies de página, los saltos de linea, las imágenes, colores, enlaces, etc.

Un detalle importante es que FPDF es una librería gratuita para cualquier uso, tanto comercial como personal (Su primera "F" inicial significa justamente "Free", osea, libre y gratis), por lo que resultará muy interesante para cualquier uso. En este artículo de DesarrolloWeb.com vamos a centrarnos en una explicación sobre su instalación y luego pasaremos a ver un primer ejemplo de uso.

## Instalación de FPDF

Esta clase requiere al menos PHP 4 o superior. Además tiene algunas extensiones que nos pueden resultar útiles.

Después de esta pequeña introducción vamos a pasar a descargar e instalar las librerías necesarias para su utilización. Para ello nos descargamos la última versión de FPDF en el siguiente [enlace](http://www.fpdf.org/" \t "/home/klvst3r/Documentos\\x/_blank), en la sección de descargas.

Una vez descargado lo subimos a nuestro servidor y lo colocamos en una carpeta llamada fpdf en la raíz del dominio, o cualquier otro directorio que deseemos, con tal que nos acordemos dónde la hemos puesto y la incluyamos correctamente en los scripts donde pensemos utilizarla.

La instalación no requiere ningún otro requisito, por lo que comprobaremos que resulta bastante sencilla.

## Trabajando con FPDF para generar un PDF desde PHP

A partir de aquí ya podemos empezar a ver como se programa con FPDF. Realmente se utiliza como cualquier clase de programación orientada a objetos con PHP, por lo que lo más importante será conocer y dominar sus diferentes métodos y propiedades. Vamos a empezar por un ejemplo muy sencillo que no requiere mucha programación PHP.

El formato general para escribir una pagina en PDF sería el siguiente:

<?php  
require('/fpdf/fpdf.php');  
  
$pdf=new FPDF();  
$pdf->AddPage();  
$pdf->SetFont('Arial','B',16);  
$pdf->Cell(40,10,'¡Mi primera página pdf con FPDF!');  
$pdf->Output();  
?>

Si ejecutamos esto directamente nos aparecerá un archivo PDF con el titulo que hemos puesto.

Analicemos todas las lineas escritas:

* Lo primero que hacemos es incluir la librería fpdf.php
* En la linea $pdf=new FPDF(); lo que estamos haciendo es crear el objeto FPDF. Si no ponemos nada entre los paréntesis, el objeto se creará con los valores por defecto, en este caso serian los siguientes: el tamaño de pagina es A4, el formato alargado y la unidad de medida el milímetro. Si queremos modificar estos parámetros seria en el siguiente orden $pdf=new FPDF('formato','unidad de medida','tamaño');
* En la linea $pfd->AddPage(); añadimos una página.
* Con SetFont(); le damos formato al texto diciendo el tipo de letra, si es en negrita o no, y el tamaño de la letra.
* Ya en la linea $pdf->Cell(); empezamos a escribir el contenido de la página. Empezamos diciendo el ancho de la celda donde vamos a escribir, el alto de la celda, y el contenido de la celda. Tiene algunos parámetros más que iremos explicando detenidamente en los siguientes artículos.
* La ultima linea $pdf->Output(); lo que hace es cerrar el archivo y enviarlo al navegador. Es importante no poner esta linea antes de terminar de escribir el archivo ya que nos dará error. Además si no lo pones justo al final y escribes algunas lineas más de código no relacionado con el PDF puede aparecerte el documento en blanco.

Bueno como vemos este primer ejemplo es sencillo, aunque no hemos profundizado en todas las opciones de las funciones que hemos visto para la generación de archivos PFD desde PHP..

En el siguiente artículo veremos cada una de las funciones nombradas anteriormente y empezaremos a darle un poco de estilo a la cabecera.

# Funciones en FPDF

**Vemos algunas de las funciones más importantes que nos podemos encontrar en FPDF.**

En el [primer articulo sobre FPDF](http://www.desarrolloweb.com/articulos/introduccion-fpdf.html) colocábamos un ejemplo muy sencillo de cómo crear un archivo pdf a partir de PHP. Bien pues ahora vamos a ir viendo una a una las funciones que aparecían en dicho ejemplo.

<?php  
require('fpdf.php');  
  
$pdf=new FPDF();  
$pdf->AddPage();  
$pdf->SetFont('Arial','B',16);  
$pdf->Cell(40,10,'¡Hola, Mundo!');  
$pdf->Output();  
?>

## FPDF()

Vamos a empezar con la clase constructor FPDF() que nos permite crear el documento pdf y darle un formato. Tenemos que tener en cuenta que los parámetros que le pasemos se usarán en todos los métodos. Su sintaxis es la siguiente:

FPDF([string orientación [, string unidad [, mixed formato]]);

* orientación es la forma de colocación de la página, es decir, debemos indicar si es normal o apaisada. El valor por defecto P es normal. El valor para apaisada es L
* unidad es la medida de usuario y sus posibles valores son: pt punto, mm milímetro, cm centímetro e in pulgada. El valor por defecto es el mm
* formato de la página. Puede tener los siguientes valores: A3, A4, A5, Letter y Legal. El valor por defecto es A4

Un ejemplo sería el siguiente:

$pdf=new FPDF('L','pt','Legal');

## AddPage()

Esta función nos añade una página nueva al documento pdf. Como parámetros tan solo tiene la orientación y el formato, el resto de características las coge por defecto del constructor.

Su sintaxis es la siguiente:

$pdf->AddPage([string orientacion[,mixed formato]]);

Los parámetros orientación y formato son iguales que en FPDF(). Si no le pasas parámetros cogerá automáticamente los del constructor.

## SetFont

Es la función que nos permite establecer el formato de la fuente utilizada en el archivo pdf. Es obligatorio llamar a esta función al inicio de la creación del archivo ya que sino el documento no sería valido.

Si queremos añadir un tipo de fuente que no está en el standar debemos utilizar la función AddFont(); que veremos más adelante.

La sintaxis de SetFont es la siguiente:

SetFont(string familia[, string estilo [, float size]]);

* familia: familia de fuente que pueden ser las standars(Courier,Helvetica o Arial, Times, Symbol,ZapfDingbats) o añadir una mediante AddFont();
* estilo: estilo de la fuente que puede ser regular , negrita B, italica I y subindice U.
* size: tamaño de la fuente en puntos. Su valor por defecto es 12.

Un ejemplo seria el siguiente:

SetFont('Helvetica','I',13);

## Cell

Esta función nos imprime una celda donde vamos a imprimir nuestro texto. Tiene bordes opcionales y color de fondo. En esta celda podemos colocar el texto alineado o centrado.

Su sintaxis es la siguiente:

Cell(float w [, float h [, string texto [, mixed borde [, int ln [, string align [, boolean fill [, mixed link]]]]]]])

* w: ancho de la celda. Si ponemos 0 la celda se extiende hasta el margen derecho.
* H: alto de la celda.
* Texto: el texto que le vamos a añadir.
* Borde: nos dice si van a ser visibles o no. si es 0 no serán visibles, si es 1 se verán los bordes.
* Ln: nos dice donde se empezara a escribir después de llamar a esta función. Siendo 0 a la derecha, 1 al comienzo de la siguiente línea, 2 debajo.
* Align: para alinear el texto. L alineado a la izquierda, C centrado y R alineado a la derecha.
* Fill: nos dice si el fondo de la celda va a ir con color o no. los valores son True o False

Un ejemplo sería el siguiente:

$pdf->Cell(10,10,'Estamos viendo',1,1,'C');

## Output()

Nos envía el documento al navegador, a un fichero local o a una cadena. Podemos abrirlo en un cuadro de diálogo o prepararlo para una descarga.

Su sintaxis es la siguiente:

string Output([string nombre, string destino])

* nombre: damos nombre al fichero, si no se indica lo llama por defecto doc.pdf
* destino: destino de envío en el documento. I envía el fichero al navegador con la opción de guardar como..., D envía el documento al navegador preparado para la descarga, F guarda el fichero en un archivo local, S devuelve el documento como una cadena.

Un ejemplo seria el siguiente:

$fpdf->Output('prueba','I');

# Edición del pie y la cabecera de tu documento PDF con FPDF

**Vamos a darle estilo a la cabecera y el pie de un archivos .pdf creado con FPDF**

En este artículo vamos a ver cómo se puede editar la cabecera y el pie del archivo pdf, de tal forma que podamos darle un estilo personal.

Para editar la cabecera necesitamos heredar la clase FPDF y sobrescribir los métodos Header() y Footer().

## Cabecera

Vamos a comenzar con la cabecera de nuestro archivo pdf. Lo que vamos a hacer es colocarle un logo y una serie de estilos.

Pongo el código a continuación y vamos explicándolo paso a paso.

<?  
require('fpdf.php');  
class PDF extends FPDF  
{  
   //Cabecera de página  
   function Header()  
   {  
  
       $this->Image('logo.png',10,8,33);  
  
      $this->SetFont('Arial','B',12);  
  
      $this->Cell(30,10,'Title',1,0,'C');  
  
   }  
}  
  
//Creación del objeto de la clase heredada  
$pdf=new PDF();  
$pdf->AddPage();  
$pdf->SetFont('Times','',12);  
//Aquí escribimos lo que deseamos mostrar...  
$pdf->Output();  
?>

Las dos primeras líneas ya las hemos visto en el artículo  [Instalación y primeros pasos para generar PDF en PHP con FPDF](http://www.desarrolloweb.com/articulos/introduccion-fpdf.html) .

La línea $this->Image ('logo.png',12,9,30); nos coloca la imagen que le pasamos por parámetro. En esta función pasamos como parámetro el archivo donde se encuentra la imagen, la abscisa de la esquina superior izquierda, ordenada de la esquina superior izquierda y la anchura de la imagen. Podemos pasar más parámetros pero ya lo comentaremos más adelante.

En la siguiente línea ($this->SetFont('Arial','B',12);) damos formato al texto de la cabecera. (esta función la hemos visto en el articulo [Funciones de FPDF](http://www.desarrolloweb.com/articulos/funciones-fpdf.html)

La línea $this->Cell (30,10,'Title',1,0,'C'); nos imprime el Título del documento. También lo hemos visto en el artículo [Funciones de FPDF](http://www.desarrolloweb.com/articulos/funciones-fpdf.html).

Una vez que hemos terminado de sobrescribir el método Header tan sólo nos queda crear el objeto de la clase heredada y realizar nuestro documento pdf.

## Pie

Para dar formato al pie tenemos que realizar la misma operación que con la cabecera, es decir, sobrescribir el método Footer. Para ello vamos ha realizar la misma operación que antes. Os dejo el ejemplo y lo vamos explicando.

<?  
require('fpdf.php');  
  
class PDF extends FPDF  
{  
  
//Pie de página  
function Footer()  
{  
  
$this->SetY(-10);  
  
$this->SetFont('Arial','I',8);  
  
$this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');  
   }  
}  
  
//Creación del objeto de la clase heredada  
$pdf=new PDF();  
$pdf->AddPage();  
$pdf->SetFont('Times','',12);  
//Aquí escribimos lo que deseamos mostrar  
$pdf->Output();  
?>

Como en el anterior ejemplo ya hemos explicado lo que hacen las primeras líneas vamos a pasar directamente al método Footer.

La línea $this->SetY (-15); nos posiciona el pie a 1,0 cm del final.   
La siguiente línea nos da el formato del texto del pie.   
La última línea del método nos imprime el número de página actual. Con un margen que se extiende hasta el margen de la derecha, un alto de celda de 10 (el formato de medida depende del dado al inicio), un texto similar a esto Page 2, sin borde, con salto de línea a la derecha y el texto centrado.

Con esto tendríamos maquetados nuestra cabecera y pie de pagina en pdf. Aunque cabe destacar que podemos realizar más operaciones en la cabecera y pie, como por ejemplo darle fondo a las celdas, rellenarlas de colores, etc. Las funciones utilizadas para estos casos las veremos en el próximo artículo sobre fpdf.

# Función Image() en FPDF

**Vemos la función Image(), para insertar imágenes en archivos PDF generados desde PHP por FPDF.**

A lo largo del [Manual de FPDF](http://www.desarrolloweb.com/manuales/manual-fpdf.html), hemos visto cómo generar PDFs básicos desde PHP. En el artículo anterior aprendíamos a [modificar el pie y la cabecera de un archivo pdf con FPDF](http://www.desarrolloweb.com/articulos/pie-cabecera-fpdf.html) y ahora vamos a ver de forma detenida la función Image().

## Image()

Esta función la utilizamos para añadir imágenes a nuestros archivos PDF.

Nos admite los formatos JPEG, PNG y GIF (para este formato necesitamos la extensión GD).   
El formato de la imagen se puede especificar explícitamente o simplemente ser deducido a partir de la extensión del fichero.

Tenemos tres opciones en cuanto a la especificación de tamaño de la imagen:

1. Podemos especificar el ancho y el largo con unidades de medida definidas por nosotros mismos
2. Podemos especificar solo el ancho y el sistema calculará el alto automáticamente
3. No especificar nada, lo que hará que se imprima la imagen a 72 puntos por pulgada

Esta función además nos permite asociar un enlace a la imagen.

Por otro lado si repetimos las imágenes, FPDF solo guardará una copia para así bajar el peso del archivo.

Su sintaxis es la siguiente:

Image(string file [, float x [, float y [, float w [, float h [, string type [, mixed link]]]]]])

Donde:

* file: nombre del archivo de la imagen.
* x: Abscisa de la esquina superior izquierda. Si no se especifica se utilizará la abscisa actual.
* y: Ordenada de la esquina superior izquierda. Si no se especifica se utilizará la ordenada actual.
* w: Ancho de la imagen en la página.
* h: Alto de la imagen en la página.
* type:Formato de la imagen.
* link: identificador devuelto por el método AddLink() o la url del enlace.

Un ejemplo sencillo sería el siguiente: $this->Image('logo.jpg',10,8,22);   
En este ejemplo nos calcularía el alto de la imagen de forma automática.

Un ejemplo completo lo haríamos asi:

<?  
require('/fpdf/fpdf.php');  
  
$pdf=new FPDF();  
//Primera página  
$pdf->AddPage();  
$pdf->SetFont('Arial','',15);  
$pdf->Cell(40,20);  
$pdf->Write(5,'A continuación mostramos una imagen ');  
$pdf->Image('leon.jpg' , 80 ,22, 35 , 38,'JPG', 'http://www.desarrolloweb.com');  
  
$pdf->Output();  
?>

Este ejemplo es bastante sencillo y lo único que hace es mostrar un texto que nos presenta una imagen que aparecerá debajo centrada.

Podemos [ver el ejemplo en una página aparte](https://desarrolloweb.com/articulos/ejemplos/fpdf/image-pdf.php" \t "/home/klvst3r/Documentos\\x/_blank).

En el siguiente artículo vermos como crear enlaces tanto internos como externos.

# Enlaces en FPDF

**Vemos las distintas formas de realizar enlaces en FPDF.**

En este capitulo del [manual de Crear PDF en PHP con FPDF](http://www.desarrolloweb.com/manuales/manual-fpdf.html) vamos a ver las tres funciones que nos permiten crear enlaces internos y externos en nuestros documentos pdf. En el [artículo anterior](http://www.desarrolloweb.com/articulos/funcion-image-fpdf.html) haciamos una pequeña introducción ya que las utilizábamos en alguno de los ejemplos, bueno pues ahora los veremos detenidamente.

## AddLink

Esta función nos permite crear un identificador que más tarde utilizaremos para crear un enlace interno a la página.

Su sintaxis es bastante sencilla ya que no requiere que se le pase ningún parámetro.

int AddLink()

El identificador resultante puede ser pasado a las [funciones Cell()](http://www.desarrolloweb.com/articulos/funciones-fpdf.html), Write(), [Image()](http://www.desarrolloweb.com/articulos/funcion-image-fpdf.html) o Link(). El destino se define con SetLink().

## SetLink

Nos define la posición y la pagina a la que un enlace hace referencia.   
Su sintaxis es la siguiente:

SetLink(int link [, float y [, int page]])

Donde:

* link: identificador obtenido mediante AddLink()
* y: posición donde se encuentra la sección a la que hace referencia el enlace. El valor por defecto es 0 y nos sitúa en la parte superior de la página.
* page: el número de la página a la que nos lleva el enlace.

Tenemos que tener en cuenta que esto es para enlaces internos dentro del documento.

Para que entendáis mejor estas funciones vamos a poner un ejemplo que crea dos paginas con un logo(con enlace externo) en la cabecera y un enlace interno.

<?  
require('../librerias/fpdf/fpdf.php');  
  
  
class PDF extends FPDF  
{  
//Cabecera de página  
function Header()  
{  
//Logo  
$this->Image("leon.jpg" , 10 ,8, 35 , 38 , "JPG" ,"http://www.desarrolloweb.com");  
//Arial bold 15  
$this->SetFont('Arial','B',15);  
//Movernos a la derecha  
$this->Cell(80);  
//Título  
$this->Cell(60,10,'Titulo del archivo',1,0,'C');  
//Salto de línea  
$this->Ln(20);  
     
}  
  
//Pie de página  
function Footer()  
{  
//Posición: a 1,5 cm del final  
$this->SetY(-15);  
//Arial italic 8  
$this->SetFont('Arial','I',8);  
//Número de página  
$this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');  
}  
}  
  
//Creación del objeto de la clase heredada  
$pdf=new PDF();  
$pdf->AliasNbPages();  
//Primera página  
$pdf->AddPage();  
$pdf->SetFont('Arial','',15);  
$pdf->Cell(40,20);  
$pdf->Write(5,'Para ir a la página 2, pulse ');  
$pdf->SetFont('','U');  
$link=$pdf->AddLink();  
$pdf->Write(5,'aquí',$link);  
$pdf->SetFont('');  
//Segunda página  
$pdf->AddPage();  
$pdf->SetLink($link);  
  
$pdf->Output();  
?>

## Link

La función link() se utiliza generalmente para colocar un enlace en una parte concreta de una imagen, ya que puedes seleccionar el área concreta de acción de dicho link. Hay que tener en cuenta que siempre sera un rectángulo.

Su sintaxis es la siguiente:

Link(float x, float y, float w, float h, mixed link)

Donde:

* x: Abscisa de la esquina superior izquierda.
* y: Ordenada de la esquina superior izquierda.
* w: Ancho del rectángulo
* h: alto del rectángulo
* link: Url o identificador devuelto por AddLink().

Podríamos poner un enlace dentro de una imagen de la siguiente forma:

<?  
require('../librerias/fpdf/fpdf.php');  
  
  
class PDF extends FPDF  
{  
//Cabecera de página  
function Header()  
{  
//Logo  
$this->Image("leon.jpg" , 10 ,8, 35 , 38 , "JPG" );  
//Arial bold 15  
$this->SetFont('Arial','B',15);  
//Movernos a la derecha  
$this->Cell(80);  
//Título  
$this->Cell(60,10,'Titulo del archivo',1,0,'C');  
//Salto de línea  
$this->Ln(20);  
     
}  
  
//Pie de página  
function Footer()  
{  
//Posición: a 1,5 cm del final  
$this->SetY(-15);  
//Arial italic 8  
$this->SetFont('Arial','I',8);  
//Número de página  
$this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');  
}  
}  
  
//Creación del objeto de la clase heredada  
$pdf=new PDF();  
$pdf->AliasNbPages();  
//Primera página  
$pdf->AddPage();  
$pdf->SetFont('Arial','',15);  
$pdf->Link(10,8,10,10,"http://www.recetasparatodos.com.es");  
$pdf->Output();  
?>

Puedes [ver el ejemplo funcionando en una página aparte](https://desarrolloweb.com/articulos/ejemplos/fpdf/imagen-enlace-pdf.php" \t "/home/klvst3r/Documentos\\x/_blank).

Como podéis comprobar hemos partido de la [cabecera y el pie del articulo anterior](http://www.desarrolloweb.com/articulos/funcion-image-fpdf.html) y hemos conseguido que el enlace en la imagen solo sea un recuadro en la parte superior izquierda.

# Insertar archivos de texto con FPDF

**Mostramos el texto de un archivo txt en nuestro documento PDF con las funciones de FPDF.**

Estamos realizando una serie de artículos prácticos para aprender a crear PDF directamente desde scripts PHP con las librerías FPDF. En el articulo anterior veíamos [cómo crear los enlaces en el texto de los archivos PDF](http://www.desarrolloweb.com/articulos/enlaces-fpdf.html) pero todavía nos queda bastante por delante.

Bien, pues en este articulo del [manual sobre FPDF](http://www.desarrolloweb.com/manuales/manual-fpdf.html) vamos a ver varias funciones para introducir el texto de un archivo .txt, es decir, vamos a mostrar el contenido de dicho archivo en nuestro PDF.

Vamos a empezar esta práctica colocando un ejemplo, que luego iremos explicándolo paso a paso para que se pueda entender todo lo que hacemos .

<?  
require('fpdf/fpdf.php');  
  
class PDF extends FPDF  
{  
//Cabecera de página  
function Header()  
{  
//Logo  
$this->Image("leon.jpg" , 10 ,8, 35 , 38 , "JPG" );  
//Arial bold 15  
$this->SetFont('Arial','B',15);  
//Movernos a la derecha  
$this->Cell(80);  
//Título  
$this->Cell(60,10,'Titulo del archivo',1,0,'C');  
//Salto de línea  
$this->Ln(20);  
     
}  
  
//Pie de página  
function Footer()  
{  
//Posición: a 1,5 cm del final  
$this->SetY(-15);  
//Arial italic 8  
$this->SetFont('Arial','I',8);  
//Número de página  
$this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');  
}  
  
function TituloArchivo($num,$label)  
{  
   $this->SetY(55);  
//Arial 12  
$this->SetFont('Arial','',12);  
//Color de fondo  
$this->SetFillColor(200,220,255);  
//Título  
$this->Cell(0,6,"Archivo $num : $label",0,1,'L',true);  
//Salto de línea  
$this->Ln(4);  
}  
  
function CuerpoArchivo($file)  
{  
//Leemos el fichero  
$f=fopen($file,'r');  
$txt=fread($f,filesize($file));  
fclose($f);  
//Times 12  
$this->SetFont('Times','',12);  
//Imprimimos el texto justificado  
$this->MultiCell(0,5,$txt);  
//Salto de línea  
$this->Ln();  
  
}  
  
function ImprimirArchivo($num,$title,$file)  
{  
$this->AddPage();  
$this->TituloArchivo($num,$title);  
$this->CuerpoArchivo($file);  
}  
}  
  
$pdf=new PDF();  
$title='Mostramos un archivo txt';  
$pdf->SetTitle($title);  
$pdf->SetY(65);  
$pdf->ImprimirArchivo(1,'Archivo de prueba ','prueba1.txt');  
$pdf->ImprimirArchivo(2,'Otro archivo','prueba2.txt');  
$pdf->Output();  
?>

Las primeras lineas no las vamos a explicar, ya que están detenidamente analizadas en los artículos anteriores de [este manual](http://www.desarrolloweb.com/manuales/manual-fpdf.html). Así que pasamos directamente a la función TituloArchivo(), que nos permite especificar el diseño de los títulos de los archivos que vamos a mostrar. Le estamos dando un tipo de letra Arial con un tamaño 12, un color de fondo azul utilizando la función SetFillColor y colocando el titulo con su correspondiente formato, utilizando la [función Cell](http://www.desarrolloweb.com/articulos/funciones-fpdf.html).

A continuación creamos la función CuerpoArchivo() que es la que va a leer el archivo y imprimirlo en nuestro PDF.   
Le pasamos como parámetro la ruta del archivo. Con fopen abrimos el archivo en modo lectura, después lo leemos con fread() y guardamos en una variable, cerramos el archivo y damos formato al texto que vamos a mostrar en nuestro PDF.

Para finalizar creamos la función ImprimirArchivo() que engloba las dos anteriores y nos hace más sencillo nuestro código.

Ya lo único que nos queda es lo de siempre, abrir nuestro archivo pdf, ponerle titulo y utilizar la función de ImprimirArchivo() las veces que queramos. Por ultimo cerrar el archivo y listo.

A continuación vamos a mostrar las dos funciones nuevas que hemos utilizado para la realización de este código:

## Función SetFillColor()

Esta función nos define el color de relleno para celdas y rectángulos rellenos. Podemos expresar dicho color en RGB o escala de grises   
Su sintaxis es la siguiente:

SetFillColor(int r [, int g, int b]);

siendo:

r: si g y b los ponemos, nos muestra el rojo sino la escala de gris   
g: el verde   
b: azul

## Función MultiCell()

Este método nos permite imprimir texto con saltos de línea. Estos pueden ser automáticos o explícito (con el carácter n).   
Su sintaxis es la siguiente:

MultiCell(float w, float h, string txt [, mixed border [, string align [, boolean fill]]])

Siendo:

* w: Ancho de celdas. Si 0, estos se extienden hasta el margen derecho de la página.
* H : Alto de las celdas.
* Txt:Cadena para imprimir.
* Border : Indica si los bordes deben ser dibujados al rededor del bloque de la celda. El valor puede ser un número:   
  0: no borde   
  1: marco   
  o algún o todos los siguientes caracteres:   
  L: izquierda   
  T: superior   
  R: derecha   
  B: inferior   
  Su valor por defecto es 0.
* align : Establece la alineación de texto.   
  L: a la izquierda   
  C: centrado   
  R: a la derecha   
  J: justificación (valor por defecto)
* fill : Indica si el fondo de la celda debe ser dibujado (true) o transparente (false). Valor por defecto: false.

Podemos [ver el ejemplo funcionando en una página aparte.](https://desarrolloweb.com/articulos/ejemplos/fpdf/archivo-fpdf.php" \t "/home/klvst3r/Documentos\\x/_blank)

# Tablas en FPDF

**Vemos cónstruir tablas para nuestros documentos en pdf con FPDF.**

Siguiendo con nuestro [manual sobre FPDF](http://www.desarrolloweb.com/manuales/manual-fpdf.html) vamos a ver como construir una tabla para el documento pdf.

Vamos a ver dos ejemplos de construcción de tablas, uno más sencillo que el otro, pero ambos igual de interesantes.

La primera tabla es muy sencillita, tan solo 2 lineas y la cabecera. Para ello vamos a pasarle un array con los nombres de cada celda, y después vamos a ir rellenando las columnas y las filas con la [función CELL](http://www.desarrolloweb.com/articulos/funciones-fpdf.html).

La función seria la siguiente:

functionTablaBasica($header)  
   {  
    //Cabecera  
    foreach($header as $col)  
    $this->Cell(40,7,$col,1);  
    $this->Ln();  
      
      $this->Cell(40,5,"hola",1);  
      $this->Cell(40,5,"hola2",1);  
      $this->Cell(40,5,"hola3",1);  
      $this->Cell(40,5,"hola4",1);  
      $this->Ln();  
      $this->Cell(40,5,"linea ",1);  
      $this->Cell(40,5,"linea 2",1);  
      $this->Cell(40,5,"linea 3",1);  
      $this->Cell(40,5,"linea 4",1);  
   }

y la llamada a la función seria la siguiente:

//Creación del objeto de la clase heredada  
$pdf=new PDF();  
//Títulos de las columnas  
$header=array('Columna 1','Columna 2','Columna 3','Columna 4');  
$pdf->AliasNbPages();  
//Primera página  
$pdf->AddPage();  
$pdf->SetY(65);  
//$pdf->AddPage();  
$pdf->Tabla Basica($header);

El siguiente ejemplo de tabla es algo más complicado ya que le vamos a dar colores a las filas y vamos a quitar los bordes inferiores de cala linea.

La función sería la siguiente:

function TablaColores($header)  
{  
//Colores, ancho de línea y fuente en negrita  
$this->SetFillColor(255,0,0);  
$this->SetTextColor(255);  
$this->SetDrawColor(128,0,0);  
$this->SetLineWidth(.3);  
$this->SetFont('','B');  
//Cabecera  
  
for($i=0;$i<count($header);$i++)  
$this->Cell(40,7,$header[$i],1,0,'C',1);  
$this->Ln();  
//Restauración de colores y fuentes  
$this->SetFillColor(224,235,255);  
$this->SetTextColor(0);  
$this->SetFont('');  
//Datos  
$fill=false;  
  
$this->Cell(40,6,"hola",'LR',0,'L',$fill);  
$this->Cell(40,6,"hola2",'LR',0,'L',$fill);  
$this->Cell(40,6,"hola3",'LR',0,'R',$fill);  
$this->Cell(40,6,"hola4",'LR',0,'R',$fill);  
$this->Ln();  
$fill=true;  
$this->Cell(40,6,"col",'LR',0,'L',$fill);  
$this->Cell(40,6,"col2",'LR',0,'L',$fill);  
$this->Cell(40,6,"col3",'LR',0,'R',$fill);  
$this->Cell(40,6,"col4",'LR',0,'R',$fill);  
$fill=!$fill;  
$this->Ln();  
$this->Cell(160,0,'','T');  
}

Todas las funciones que vemos ya están explicadas en otros artículos del [manual de FPDF](http://www.desarrolloweb.com/manuales/manual-fpdf.html) por lo que no entraremos en detalles.   
Hemos establecido el color de relleno de las celdas, el color del texto, el grosor de los bordes y el tipo de letra. A continuación hemos creado la primera linea en rojo y la letra en negrita. Después asignamos otros colores para el relleno y la letra y vamos rellenando el resto de filas. Vamos colocando los bordes donde necesitamos y diciendo si la celda va rellena o no de color con la variable $fill.   
Al final tenemos que añadir el borde de abajo del todo, para ello utilizamos la linea:

$this->Cell(160,0,'','T');

Que nos dice que la linea tiene que ser 160 de ancho, 0 de alto y mostrando solo el borde superior.

Os pongo el ejemplo completo con las dos tablas:

<?  
require('fpdf.php');  
  
class PDF extends FPDF  
{  
//Cabecera de página  
   function Header()  
   {  
    //Logo  
    $this->Image("leon.jpg" , 10 ,8, 35 , 38 , "JPG" ,"http://www.mipagina.com");  
    //Arial bold 15  
    $this->SetFont('Arial','B',15);  
    //Movernos a la derecha  
    $this->Cell(80);  
    //Título  
    $this->Cell(60,10,'Titulo del archivo',1,0,'C');  
    //Salto de línea  
    $this->Ln(20);  
        
   }  
     
   //Pie de página  
   function Footer()  
   {  
    //Posición: a 1,5 cm del final  
    $this->SetY(-15);  
    //Arial italic 8  
    $this->SetFont('Arial','I',8);  
    //Número de página  
    $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');  
   }  
   //Tabla simple  
   function TablaSimple($header)  
   {  
    //Cabecera  
    foreach($header as $col)  
    $this->Cell(40,7,$col,1);  
    $this->Ln();  
      
      $this->Cell(40,5,"hola",1);  
      $this->Cell(40,5,"hola2",1);  
      $this->Cell(40,5,"hola3",1);  
      $this->Cell(40,5,"hola4",1);  
      $this->Ln();  
      $this->Cell(40,5,"linea ",1);  
      $this->Cell(40,5,"linea 2",1);  
      $this->Cell(40,5,"linea 3",1);  
      $this->Cell(40,5,"linea 4",1);  
   }  
     
   //Tabla coloreada  
function TablaColores($header)  
{  
//Colores, ancho de línea y fuente en negrita  
$this->SetFillColor(255,0,0);  
$this->SetTextColor(255);  
$this->SetDrawColor(128,0,0);  
$this->SetLineWidth(.3);  
$this->SetFont('','B');  
//Cabecera  
  
for($i=0;$i<count($header);$i++)  
$this->Cell(40,7,$header[$i],1,0,'C',1);  
$this->Ln();  
//Restauración de colores y fuentes  
$this->SetFillColor(224,235,255);  
$this->SetTextColor(0);  
$this->SetFont('');  
//Datos  
   $fill=false;  
$this->Cell(40,6,"hola",'LR',0,'L',$fill);  
$this->Cell(40,6,"hola2",'LR',0,'L',$fill);  
$this->Cell(40,6,"hola3",'LR',0,'R',$fill);  
$this->Cell(40,6,"hola4",'LR',0,'R',$fill);  
$this->Ln();  
      $fill=!$fill;  
      $this->Cell(40,6,"col",'LR',0,'L',$fill);  
$this->Cell(40,6,"col2",'LR',0,'L',$fill);  
$this->Cell(40,6,"col3",'LR',0,'R',$fill);  
$this->Cell(40,6,"col4",'LR',0,'R',$fill);  
$fill=true;  
   $this->Ln();  
   $this->Cell(160,0,'','T');  
}  
  
     
     
}  
  
$pdf=new PDF();  
//Títulos de las columnas  
$header=array('Columna 1','Columna 2','Columna 3','Columna 4');  
$pdf->AliasNbPages();  
//Primera página  
$pdf->AddPage();  
$pdf->SetY(65);  
//$pdf->AddPage();  
$pdf->TablaSimple($header);  
//Segunda página  
$pdf->AddPage();  
$pdf->SetY(65);  
$pdf->TablaColores($header);  
$pdf->Output();  
?>

Podéis ver el [ejemplo funcionando en el siguiente enlace.](https://desarrolloweb.com/articulos/ejemplos/fpdf/tablas-fpdf.php" \t "/home/klvst3r/Documentos\\x/_blank)

# Columnas en FPDF

**Vemos como maquetar nuestro archivo pdf a dos columnas.**

En este artículo vamos a seguir dándole forma a nuestro archivo PDF. Para ello en esta ocasión vamos a ver como colocar texto en dos columnas, además vamos a trabajar con la estructura de control if.

Vamos a seguir utilizando el ejemplo del [articulo anterior](http://www.desarrolloweb.com/articulos/tablas-fpdf.html) siguiendo la línea de nuestro [manual sobre FPDF](http://www.desarrolloweb.com/manuales/manual-fpdf.html).

Lo primero que vamos a hacer es crear dos variables e inicializarlas:

<?  
require('fpdf/fpdf.php');  
  
class PDF extends FPDF  
{     
   //Columna actual  
   var $col=0;  
   //Ordenada de comienzo de la columna  
   var $y=0;

A continuación añadimos la cabecera y el pie del documento

//Cabecera de página  
   function Header()  
   {  
      //Logo  
      $this->Image("leon.jpg" , 10 ,8, 35 , 38 , "JPG" );  
      //Arial bold 15  
      $this->SetFont('Arial','B',15);  
      //Movernos a la derecha  
      $this->Cell(80);  
      //Título  
      $this->Cell(60,10,'Titulo del archivo',1,0,'C');  
      //Salto de línea  
      $this->Ln(20);  
   }  
  
   //Pie de página  
   function Footer()  
   {  
      //Posición: a 1,5 cm del final  
      $this->SetY(-15);  
      //Arial italic 8  
      $this->SetFont('Arial','I',8);  
      //Número de página  
      $this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');  
   }

Ahora tenemos que crear una función que nos establezca la posición de una columna.

function SetCol($col)  
{  
//Establecer la posición de una columna dada  
$this->col=$col;  
$x=10+$col\*75;  
$this->SetLeftMargin($x);  
$this->SetX($x);  
}

En este caso lo que hacemos es asignar a $this->col el valor de la variable col especificada al inicio. A partir de hay creamos una variable $x que nos dará el margen izquierdo de la posición de dicha columna.

Y a continuación la función que realmente nos interesa ya que es la que realiza el salto de columna o página según la posición en la que estemos.

function AcceptPageBreak()  
{  
//Método que acepta o no el salto automático de página  
if($this->col<2)  
{  
//Ir a la siguiente columna  
$this->SetCol($this->col+1);  
//Establecer la ordenada al principio  
$this->SetY($this->y0);  
//Seguir en esta página  
return false;  
}  
else  
{  
//Volver a la primera columna  
$this->SetCol(0);  
//Salto de página  
return true;  
}  
}

Esta función lo que nos hace es preguntar si estamos en cualquiera de las columnas menor que 2, si estamos aquí cambiamos la variable para en el siguiente paso ir a la columna siguiente, y seguimos en la misma pagina. Si ya estamos en la columna 2 lo único que hacemos es pasar de página y situarnos nuevamente en la columna 0.

Ahora pasamos el resto de funciones que necesitamos para imprimir los textos contenidos en los archivos externos.

function TituloArchivo($num,$label)  
{  
   $this->SetY(55);  
   //Arial 12  
   $this->SetFont('Arial','',12);  
   //Color de fondo  
   $this->SetFillColor(200,220,255);  
   //Título  
   $this->Cell(0,6,"Archivo $num : $label",0,1,'L',true);  
   //Salto de línea  
   $this->Ln(4);  
   //Guardar ordenada  
$this->y0=$this->GetY();  
  
}  
  
function CuerpoArchivo($file)  
{  
   //Leemos el fichero  
   $f=fopen($file,'r');  
   $txt=fread($f,filesize($file));  
   fclose($f);  
   //Times 12  
   $this->SetFont('Times','',12);  
   //Imprimimos el texto justificado  
   $this->MultiCell(60,5,$txt);  
   //Salto de línea  
   $this->Ln();  
   //Volver a la primera columna  
   $this->SetCol(0);  
}  
  
function ImprimirArchivo($num,$title,$file)  
{  
   $this->AddPage();  
   $this->TituloArchivo($num,$title);  
   $this->CuerpoArchivo($file);  
}

Después de esto tenemos que empezar a crear el archivo con el siguiente código:

$pdf=new PDF();  
$title='Mostramos un archivo txt';  
$pdf->SetTitle($title);  
$pdf->SetY(65);  
$pdf->ImprimirArchivo(1,'Archivo de prueba ','prueba1.txt');  
$pdf->ImprimirArchivo(2,'Otro archivo','prueba2.txt');  
$pdf->Output();

Ahora para terminar colocamos el código completo y el enlace a una página aparte donde podemos [verlo funcionando](https://desarrolloweb.com/articulos/ejemplos/fpdf/columnas-fpdf.php" \t "/home/klvst3r/Documentos\\x/_blank).

<?  
require('fpdf/fpdf.php');  
  
class PDF extends FPDF  
{  
   //Columna actual  
   var $col=0;  
   //Ordenada de comienzo de la columna  
   var $y=0;  
   //Cabecera de página  
function Header()  
{  
//Logo  
$this->Image("leon.jpg" , 10 ,8, 35 , 38 , "JPG" );  
//Arial bold 15  
$this->SetFont('Arial','B',15);  
//Movernos a la derecha  
$this->Cell(80);  
//Título  
$this->Cell(60,10,'Titulo del archivo',1,0,'C');  
//Salto de línea  
$this->Ln(20);  
     
}  
  
//Pie de página  
function Footer()  
{  
//Posición: a 1,5 cm del final  
$this->SetY(-15);  
//Arial italic 8  
$this->SetFont('Arial','I',8);  
//Número de página  
$this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');  
}  
function SetCol($col)  
   {  
      //Establecer la posición de una columna dada  
      $this->col=$col;  
      $x=10+$col\*75;  
      $this->SetLeftMargin($x);  
      $this->SetX($x);  
   }  
   function AcceptPageBreak()  
   {  
      //Método que acepta o no el salto automático de página  
      if($this->col<2)  
      {  
      //Ir a la siguiente columna  
       $this->SetCol($this->col+1);  
      //Establecer la ordenada al principio  
      $this->SetY($this->y0);  
      //Seguir en esta página  
      return false;  
      }  
      else  
      {  
      //Volver a la primera columna  
      $this->SetCol(0);  
      //Salto de página  
      return true;  
      }  
   }  
function TituloArchivo($num,$label)  
{  
   $this->SetY(55);  
$this->SetFont('Arial','',12);  
$this->SetFillColor(200,220,255);  
$this->Cell(0,6,"Archivo $num : $label",0,1,'L',true);  
$this->Ln(4);  
//Guardar ordenada  
$this->y0=$this->GetY();  
  
}  
  
function CuerpoArchivo($file)  
{  
//Leemos el fichero  
$f=fopen($file,'r');  
$txt=fread($f,filesize($file));  
fclose($f);  
//Times 12  
$this->SetFont('Times','',12);  
//Imprimimos el texto justificado  
$this->MultiCell(60,5,$txt);  
      //Salto de línea  
      $this->Ln();  
      //Volver a la primera columna  
      $this->SetCol(0);  
  
}  
  
function ImprimirArchivo($num,$title,$file)  
{  
$this->AddPage();  
$this->TituloArchivo($num,$title);  
$this->CuerpoArchivo($file);  
}  
}  
  
$pdf=new PDF();  
$title='Mostramos un archivo txt';  
$pdf->SetTitle($title);  
$pdf->SetY(65);  
$pdf->ImprimirArchivo(1,'Archivo de prueba ','prueba1.txt');  
$pdf->ImprimirArchivo(2,'Otro archivo','prueba2.txt');  
$pdf->Output();