

UD-07a: Generación de PDFs desde PHP

Desarrollo Web en Entorno Servidor

Curso 2020/2021

Creación de ficheros PDF

Podemos crear ficheros PDF con funciones PHP mediante aplicaciones de pago, como PDFlib, o clases gratuitas como FPDF o TCPDF, que usaremos en esta unidad.

La clase TCPDF

Descargaremos un zip desde www.tcpdf.org, desde github. Descomprimiremos el zip, COMPLETO de la última versión, en el directorio del proyecto que vayamos a usar. De todos los archivos, `tcpdf.php` es el que contiene la clase que nos interesa.

Para crear PDFs con TCPDF debemos incluir en los scripts los siguientes ficheros: `tcpdf.php` y `tcpdf_config.php`, que son la propia clase y su archivo de configuración.

```
<?php
    include('tcpdf/config/tcpdf_config.php');
    include('tcpdf/tcpdf.php');
?>
```

Creación de ficheros PDF

El constructor de la clase TCPDF

El uso de la clase TCPDF requiere la creación de un objeto de este tipo:

```
$documento = new TCPDF();
```

El constructor permite seis argumentos:

[EN VERDE, EL VALOR POR DEFECTO]

```
$documento = new TCPDF(orientacion, unidad_de_medida, formato, unicode, codificacion, diskcache);
```

Admite dos valores: **'P'** (normal ó Portrait) y **'L'** (apaisado o Landscape).

Unidad de Medida de las dimensiones del documento: **'in'** (pulgadas), **'pt'** (puntos), **'mm'** (milímetros) o **'cm'** (centímetros).

Formato de cada página del documento (**A4**, A3, A2...)

Indica si los textos tendrán formato UNICODE o no (**TRUE** o FALSE)

Formato de codificación, como **'UTF-8'**

Usar menos memoria RAM almacenando los datos temporales en caché (TRUE o **FALSE**)

Creación de ficheros PDF

La clase TCPDF también permite especificar formatos personalizados, mediante la definición de las dimensiones deseadas en un array (*especificadas en la unidad establecida para el documento*). Posteriormente, veremos un ejemplo en el que incluiremos el array como parámetro en la llamada al constructor de la clase.

Una vez creado el objeto, debemos añadirle páginas mediante el siguiente método:

```
$documento->AddPage(); // añade una nueva página.
```

El método Output() permite visualizar o guardar el objeto creado:

```
$documento->Output(nombre, destino);
```

Si no especificamos ninguno, se le asigna por defecto el valor **doc.pdf**

Puede tener uno o varios de los siguientes valores:

- **'I'** → Permite **visualizar** el fichero **directamente en el navegador**, si el plugin está disponible.
- **'D'** → Envía el fichero al navegador **mostrando la ventana de opción que permite elegir entre Abrir o Descargar**. Si se elige esta última, guardará el fichero con el nombre dado en el parámetro nombre.
- **'F'** → **Guarda el fichero en el directorio actual del servidor** con el nombre asignado en la opción nombre.

Creación de ficheros PDF

El método SetDisplayMode()

Este método sólo afecta a la forma en la se visualiza el documento en la pantalla del cliente. Permite configurar dos parámetros: tamaño y forma de visualización.

```
$documento->SetDisplayMode(tamaño, visualización);
```



Recuerda: no se refiere al documento impreso, sino a la visualización en pantalla. Permite una de estas opciones:

Fullpage → el zoom del visor se adapta de modo que encaje la página completa en la pantalla.

Fullwidth → ajusta el zoom de forma que se visualice el documento ajustando el ancho al de la pantalla.

Real → ajusta el zoom del visor al 100%.

Default → usa el modo por defecto del visor.

Permite, entre otras, estas opciones de visualización de páginas:

Single → muestras las páginas separadas de una en una.

Continuous → va mostrándolas una tras otra de forma continua.

Two → muestra dos columnas, con dos páginas a la vez.

Default → usa el modo por defecto del visor.

<?php

```
// incluimos todos los ficheros requeridos
include('tcpdf/config/tcpdf_config.php');
include('tcpdf/tcpdf.php');

// creamos un nuevo objeto (MiPDF) utilizando la clase TCPDF
$MiPDF = new TCPDF();

// creamos una página en blanco
$MiPDF->Addpage();

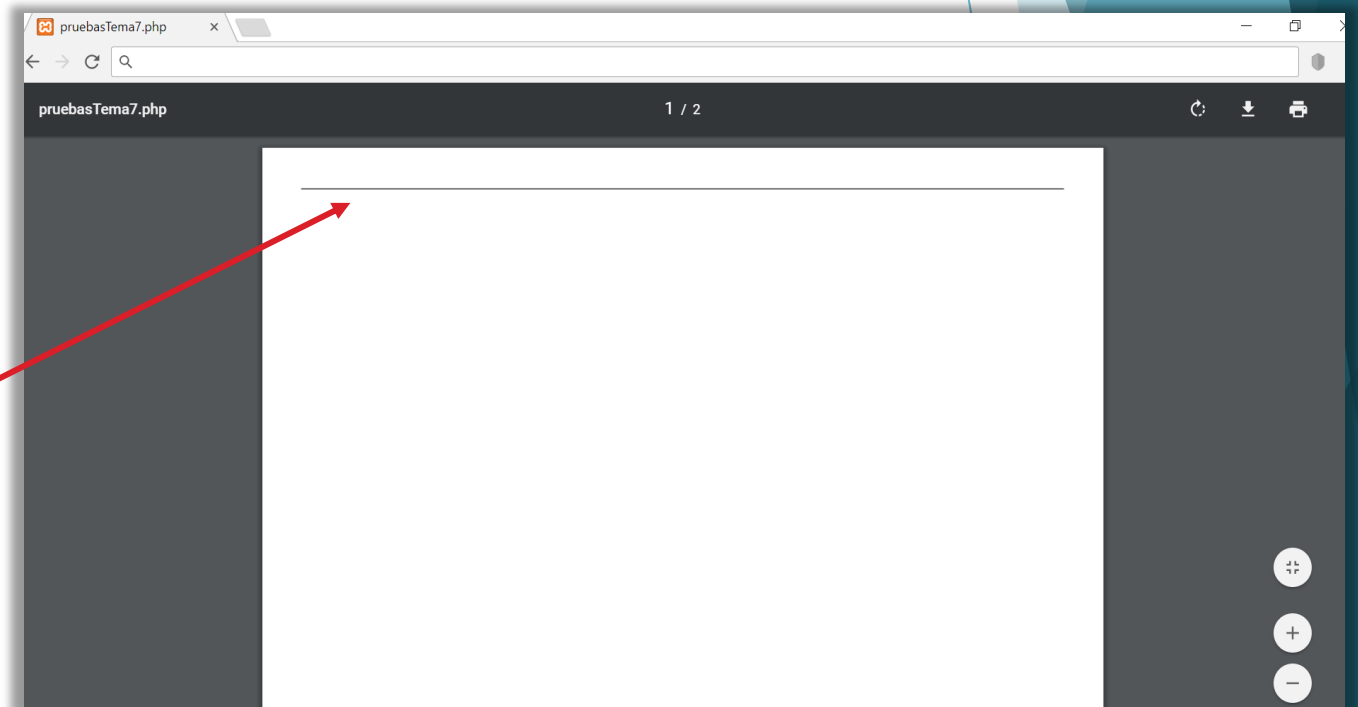
// creamos una segunda página en blanco
$MiPDF->Addpage();

// visualizamos el documento
$MiPDF->Output();
```

?>

Por defecto, esta clase incluye en cada documento un encabezado y un pie de página. Para quitarlo, tras crear el TCPDF:

```
$MiPDF->setPrintHeader(false);
$MiPDF->setPrintFooter(false);
```



```
<?php
```

```
// incluimos todos los archivos requeridos  
include('tcpdf/config/tcpdf_config.php');  
include('tcpdf/tcpdf.php');
```

```
// creamos un array con las dimensiones (ancho y alto);  
$dimensiones = array(1,3);
```

```
// creamos un nuevo objeto de la clase TCPDF incluyendo los valores a usar  
$MiPDF = new TCPDF('P', 'in', $dimensiones);
```

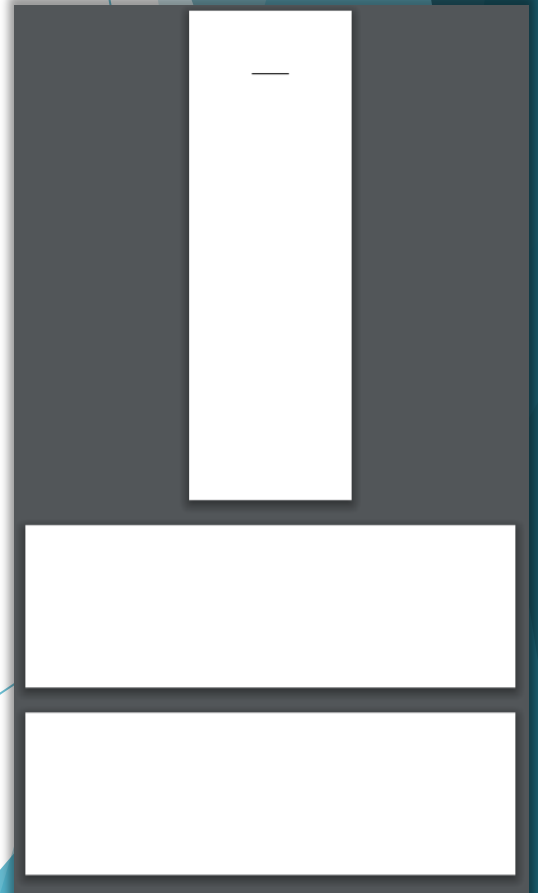
```
// creamos una página en blanco con las indicaciones del constructor  
$MiPDF->Addpage();
```

```
// incluimos una segunda página cambiando la orientación  
$MiPDF->Addpage('L');
```

```
// creamos una segunda página. Al no incluir orientación, se usa la anterior  
$MiPDF->Addpage();
```

```
// visualizamos el documento  
$MiPDF->Output();
```

```
?>
```



Creación de ficheros PDF

Otros métodos de la clase TCPDF

Existen otros métodos, de carácter básicamente informativo:

- `SetAuthor("nombre del autor")` → Permite incluir una cadena con el nombre del autor.
- `SetCreator("nombre de la aplicación")` → Permite incluir una cadena con el nombre del programa usado para crear el documento.
- `SetTitle("título del documento")` → Permite incluir el título del documento.
- `SetKeywords("palabras clave")` → Permite incluir una lista de palabras clave, separadas por espacios.
- `SetSubject("explicación de temas tratados")` → Permite incluir una cadena de texto explicativa de los temas tratados en el contenido del documento

Los datos incluidos en el documento con estos métodos sólo se pueden ver cuando se exploran los metadatos del documento.

Textos en PDF

Fijación de márgenes

Por defecto, los márgenes de la página miden 15 milímetros, por la izquierda y por la derecha del documento y 27 y 25 milímetros por la parte superior e inferior del mismo, respectivamente. También por defecto, se establecen márgenes para encabezado y pie de página de 5 y 10 milímetros respectivamente.

Los valores de esta configuración por defecto están en [tcpdf_config.php](#). Podemos editar este fichero, modificando los valores asignados con la función **define** a las constantes cuyo nombre comienza por [PDF_MARGIN_](#) seguido de [LEFT](#), [TOP](#), etc.

También podemos hacerlo mediante código, para un documento concreto:

```
$documento->SetMargins(izquierdo, superior, derecho)
// Si se omite el 3er parámetro se le asignará el valor indicado para el izquierdo.
// El margen inferior lo controla otra función $doc->SetAutoPageBreak(TRUE, 0);
```

Textos en PDF

Margen inferior y saltos de página automáticos

Cuando introducimos un texto muy largo que abarque más de una página, por defecto, la clase creará saltos de página cada vez que se llegue al final de la misma.

Podemos cambiar esta característica por medio del siguiente método:

```
$documento->SetAutoPageBreak(automatico, margen_inferior)
```

Valor booleano. Si es **true** (por defecto) se van añadiendo nuevas páginas de forma automática, hasta que se incluya todo el texto. Si es false, los textos que excedan los límites de la página no se verán, no se añadirá una nueva página.

Especifica el margen inferior mínimo de la página. Su valor deberá indicarse en las unidades establecidas por el constructor. Si no se indica asumirá cero, como margen por defecto.

Textos en PDF

Tipo de letra, estilo y tamaño

La elección del tipo, estilo y tamaño de letra puede hacerse con la función:

```
$documento->SetFont(nombre, estilo, tamaño)
```

Indica el nombre del tipo de letra (helvetica, times, zapfdingbats, courier y symbol son las fuentes incluidas en la distribución).

Tamaño de la letra expresado en puntos.

Puede especificarse con la cadena vacía ("") para el tipo normal o mediante 'B', 'I', 'U' (negrita, cursiva, subrayada) o sus combinaciones. Por ejemplo, 'BU' indicará negrita subrayada.

Por defecto: **helvetica, normal, 12 puntos.**

Textos en PDF

Colores de texto y fondo

Para cambiar el color del texto podemos usar el método:

```
$documento->SetTextColor(rojo, verde, azul) // (valores de 0 a 255)  
$documento->SetTextColor(grises) // (valor de 0 a 255, escala de grises)
```

Si no se especifica color, los textos aparecerán en negro. Si se establece un color, se mantiene hasta que se establezca otro.

Para cambiar un color de fondo en textos, celdas o dibujos:

```
$documento->SetFillColor(rojo, verde, azul)  
$documento->SetFillColor(grises)
```

Textos en PDF

Posición actual

Hay dos métodos que permiten determinar 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 también modificar el valor actual de esos puntos:

```
$documento->GetX()  
$documento->GetY()
```



El resultado se expresa en las unidades establecidas por el constructor. El origen de coordenadas se encuentra en la esquina superior izquierda del documento y valores positivos los colocados a la derecha (X) y debajo (Y) del origen.

```
$documento->SetX(valorX)  
$documento->SetY(valorY)  
$documento->SetXY(valorX, valorY)
```



Los valores positivos se entienden respecto al borde superior izquierdo, como con GetX() y GetY(). En caso de que el valor de X (en el caso SetX) sea negativo, se considera que la coordenada es relativa 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.

Textos en PDF

Salto de línea y numeración de páginas

Los saltos de línea y la numeración de páginas son posibles mediante estos métodos:

- Introduce un salto de línea. La posición actual regresa al margen izquierdo y la vertical se desplaza 'n' unidades indicado en el parámetro medida.

```
$documento->Ln(n)
```

- Devuelve una cadena que contiene el número de la página actual.

```
$documento->getAliasNumPage()
```

- Devuelve el número total de páginas del documento.

```
$documento->getAliasNbPage()
```

Textos en PDF

Inclusión de texto en el documento

Incluiremos texto con el método:

```
$documento->Write({parámetros de abajo})
```

```
$interlinea, $texto, $enlace = "", $fondo = false, $alineacion = "", $nueva_linea = false
```

Permite establecer el **interlineado** del texto.

Cadena o variable conteniendo **el texto que debe incluirse**. *Se debe comprobar que el texto usa la misma codificación establecida por el constructor del objeto.*

Permite incluir una cadena con una **dirección URI**. El texto se comportaría como un hipervínculo

Si es TRUE el texto aparece sobre un **color de fondo**. Puede establecerse con SetFillColor.

L, C, R o J (**alineación** izquierda, centrada, derecha o justificada)

Lo ponemos en TRUE si queremos **situar el texto en la línea siguiente** del último texto introducido. Si no, sería en la misma.

Textos en PDF

Estilos de línea

Para establecer el color de las líneas, como bordes de celdas y otros elementos gráficos, podemos recurrir a estos métodos:

```
$documento->SetDrawColor(rojo, verde, azul)  
$documento->SetDrawColor(negro)
```

Para el grosor de las líneas (misma unidad de medida que la página, en decimal):

```
$documento->SetLineWidth(grosor)
```

Para el estilo de línea:

```
$documento->SetLineStyle($array)
```

Se explica en la página siguiente.

\$array es un array asociativo donde:

- **\$array['width']** = **numero** → Para el espesor de la línea. Es una alternativa a *SetLineWidth*
- **\$array['join']** = ['miter' | 'round' | 'bevel'] → Admite esos valores: *miter* (inglete), *round* (redondeado) o *bevel* (bisel) y establece la forma en la que se perfilarán los extremos de los trazos de las líneas discontinuas.
- **\$array['cap']** = ['butt' | 'round' | 'square'] → Admite estos valores: *but* (inglete), *round* (redondeado) o *square* (cuadrado) y establece la forma en la que se dibujarán las uniones de las líneas en las esquinas del rectángulo de la celda.
- **\$array['dash']** = "cadena" → Para establecer el tipo de trazos que formarán la línea. Si se indica 0 dibujará una línea continua. Si se especificara algo como "20,10" lo que representaría sería una línea discontinua formada por trazos de color de 20 unidades de longitud seguidos de espacio en blanco de 10 unidades. Si este parámetro fuera "20,10,8,6,5,8" lo que representaría sería: un trazo de 20 unidades seguido de un espacio de 10, después iría un trazo de 8 unidades seguido de un espacio de 6 repitiéndose la secuencia completa hasta efectuar el trazado completo.
- **\$array['phase']** = **numero** → Para especificar, con un numero entero, por cuál de los valores indicados en *dash* comienza el trazado de la línea.
- **\$array['color']** = **array(red, green, blue)** → Para incluir un array escalar que contenga los valores que componen el color de la línea. Es una alternativa a *SetDrawColor*.

```
array('width' => 0.5, 'cap' => 'butt', 'join' => 'miter', 'dash' => 0, 'color' => array(0, 0, 255));
```

Textos en PDF

Creación de tablas

Para crear una tabla en PDF, hay que introducir cada celda que la compone.

Este método es la forma más simple de insertar textos individuales, por ejemplo títulos de una sola línea, o tablas formadas por celdas de una sola línea de texto.

```
$documento->Cell(  
    $ancho,  
    $alto = 0,  
    $texto = "",  
    $borde = 0,  
    $salto_de_linea = 0,  
    $alineacion = "",  
    $fondo = false,  
    $enlace = "",  
    $ajuste_horizontal = 0,  
    $ignore_min_height = false,  
    $alin_vertical_texto = 'T',  
    $alineacion_vertical = 'M');
```

Ver siguiente página

Textos en PDF

Cell imprime un área rectangular con fondo y bordes opcionales y que puede contener una cadena de texto alineable respecto a los bordes de la celda.

La esquina superior izquierda de la celda se sitúa en la posición actual del punto de inserción. Después de ser invocado este método, el punto de inserción puede desplazarse a la derecha o a la línea siguiente.

El texto puede ser utilizado con un hiperenlace y, además, puede expandirse en tamaño o en espaciado para ajustarse a las dimensiones de la celda.

Veamos los parámetros de esta función:

- **\$ancho** → Ancho de la celda expresado en la unidad usada en el constructor. Si es cero se extiende desde el punto de inserción hasta el margen derecho de la página.
- **\$alto** → Alto mínimo de la celda expresado en la unidad de medida establecida en la llamada al constructor. Si se establece altura cero, la celda se adaptará a las dimensiones de su contenido.
- **\$texto** → Cadena o variable conteniendo el texto que debe incluirse. Es aconsejable comprobar que utiliza la misma codificación establecida por el constructor del objeto.

Textos en PDF

- **\$borde** → Si es 0, la celda se visualizará sin bordes. Si es 1, visualiza la celda con un borde del color y ancho especificados por los métodos `SetDrawColor` y `SetLineWidth` o por las opciones por defecto de estos (negro y espesor 1). Si es 'LTRB', se verán los bordes que se indiquen, es decir, L izquierda, T superior, R derecha y B inferior. Por ejemplo: 'TB' dibujaría bordes por la parte superior e inferior de la celda y 'LRB' pondría bordes por todas partes excepto por la parte inferior. Es importante indicar los bordes en el orden correcto.
- **\$salto_de_línea** → Si es 0 (por defecto), se establece que el punto de inserción, después de la llamada a este método, se sitúa, justamente, en el borde derecho de la celda. Si es 1, la celda siguiente estará en el margen izquierdo de la página y debajo de la fila actual. Si es 2, la próxima celda aparecerá debajo de la actual y alineada al borde izquierdo de esta.
- **\$alineación** → Establece la alineación (L izquierda, C centrada, R derecha o J justificada) del texto respecto a la celda contenedora.
- **\$fondo** → Si vale **true** la caja contenedora tendrá color de fondo, que puede establecerse mediante `SetFillColor` y por defecto es negro.
- **\$enlace** → Permite incluir una dirección URI. El texto se comportaría como un enlace.

Textos en PDF

Transformaciones en celdas

La clase TCPDF tiene métodos para hacer transformaciones gráficas de algunos elementos entre los que se incluyen las celdas.

Toda transformación empieza con `$documento->StartTransform()` y acaba con `$documento->StopTransform()`.

La estructura del código podría ser similar a esta:

```
$documento->StartTransform()  
... // funciones de transformación (ver página siguiente)  
$documento->Cell($ancho) // incluir la celda objeto de transformación  
$documento->StopTransform();
```

Textos en PDF

Funciones de transformación

```
$documento->Rotate(angulo, x, y)
// x e y son las coordenadas del punto de giro (por defecto, esquina superior izquierda).

$documento->Translate(x, y)
// si x e y son positivos se mueve derecha y abajo (si son negativos, izquierda y arriba).

$documento->SkewY(angulo, x, y)
$documento->SkewX(angulo, x, y)
// “inclina” el rectángulo de la celda

$documento->ScaleX(porcentaje, x, y)
$documento->ScaleY(porcentaje, x, y)
$documento->ScaleXY(porcentaje, x, y)
// x e y son las coordenadas del centro de transformación (si se omiten, se tomará el
punto de inserción actual).

$documento->MirrorV(angulo, y)
$documento->MirrorH(angulo, x)
$documento->MirrorP(angulo, x, y)
// realiza simetrías cuyo eje varía en cada uno de los tres casos.
```

Textos en PDF

Celdas multilínea

La creación de celdas sin restricción de línea se crea con *Multicell*, que sigue los mismos criterios que *Cell* en lo relativo al posicionamiento del punto de inserción.

```
$documento->Multicell(  
    $ancho, $alto,  
    $texto,  
    $borde = 0,  
    $alineación = 'J',  
    $fondo = false,  
    $salto_de_linea = 1,  
    $x = '', $y = '',  
    $reseth = true,  
    $ajuste_horizontal = 0,  
    $ishtml = false,  
    $autopadding = true,  
    $maxh = 0,  
    $alineacion_vertical = 'T',  
    $fitcell = false));
```

Las propiedades `$ancho`, `$alto`, `$texto`, `$borde`, `$alineacion`, `$fondo`, `$salto_de_linea`, `$ajuste_horizontal` y `$alineacion_vertical` coinciden con los parámetros de *Cell*. El resto, se detallan en las páginas siguientes.

[LEFT] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[RIGHT] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[CENTER] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[JUSTIFY] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[DEFAULT] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[VERTICAL ALIGNMENT - TOP] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[VERTICAL ALIGNMENT - MIDDLE] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

[VERTICAL ALIGNMENT - BOTTOM] Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Textos en PDF

- **\$x** → Abscisa del punto de inserción expresada en la unidad de medida del constructor.
- **\$y** → Ordenada del punto de inserción expresada en la unidad de medida del constructor.
- **\$reseth** → Si es **false** la celda actual mantendrá como interlínea la altura que tuviera la celda que le precede. Caso de ser **true** (valor por defecto), la interlínea se establecerá automáticamente ajustándose al tamaño de la letra en uso.
- **\$ishtml** → Si se establece **true** permite incluir textos en formato HTML. Esta opción en la versión actual de la clase presenta bastantes restricciones.
- **\$autopadding** → Si se configura como **true** el margen interior de la celda se ajusta de manera automática a lo establecido mediante *SetCellPaddings* a los cuatro bordes de la celda. Si su valor es **false** ese ajuste de espacios se realiza sólo respecto a los bordes izquierdo y derecho.
- **\$maxh** → Establece la altura máxima que puede tener la celda. Cuando se especifica cero se desactiva la opción. Requiere que el parámetro **\$ishtml** esté configurado como **false**.
- **\$fitcell** → Si se configura como **true** el tamaño de la fuente puede ser reducido hasta adaptarse a las dimensiones de la celda contenedora. Requiere que esté definido **\$maxh** con un valor distinto de cero.

Textos en PDF

Convertir HTML en PDF

Para pasar de HTML a PDF puede usarse el método siguiente:

```
$documento->WriteHTML(  
    $html,  
    $ln = true,  
    $fondo = false,  
    $reseth = false,  
    $cell = false,  
    $alineacion = '' );
```

Para que los resultados sean correctos es muy importante que el código HTML esté limpio y bien formado. De lo contrario los resultados pueden resultar muy diferentes a los deseados. Hay que tener en cuenta que este método tiene algunas limitaciones.

- `$fondo`, `$reseth` y `$alineacion` coinciden con lo descrito en métodos anteriores.
- `$html` → Es el texto a incluir con sus etiquetas HTML.
- `$ln` → Si se establece como `true` agrega un salto de línea al final del texto.
- `$cell` → Si es `true` agrega el contenido en una celda a la derecha de lo anterior.

Textos en PDF

Textos en diferentes columnas

El siguiente método permite incluir, con [Write](#), textos en varias columnas, vistas como divisiones verticales. Previamente, es preciso establecer el número y el ancho de las columnas:

```
$documento->setEqualColumns(numero, ancho)
```

Después, antes de incluir textos, hay que situar el punto de inserción en la columna adecuada. Para ello, la clase dispone del método siguiente:

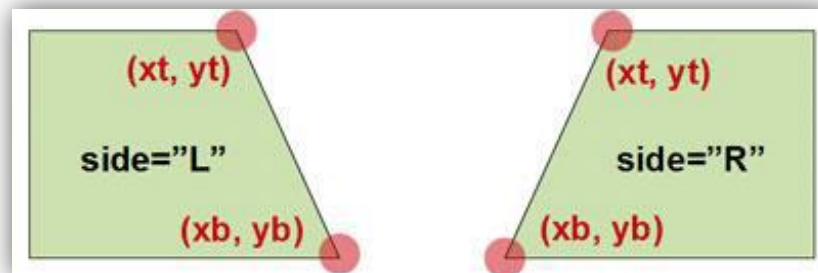
```
$documento->selectColumn(columna)
```

Finalmente, sólo faltaría insertar el texto mediante el método [Write](#).

Textos en PDF

Reservar espacios en páginas

- Nos permite situar regiones de no escritura en la página.
- Puede ser una parte con forma rectangular o trapezoidal que no se cubrirá al escribir texto o código html.
- Puede ser necesario reservar áreas de las páginas para incluir en ellas imágenes u otros elementos gráficos.
- Una región siempre se alinea en el lado izquierdo o derecho del borde de la página y se define mediante un segmento.
- Podemos configurar varias regiones para la misma página.



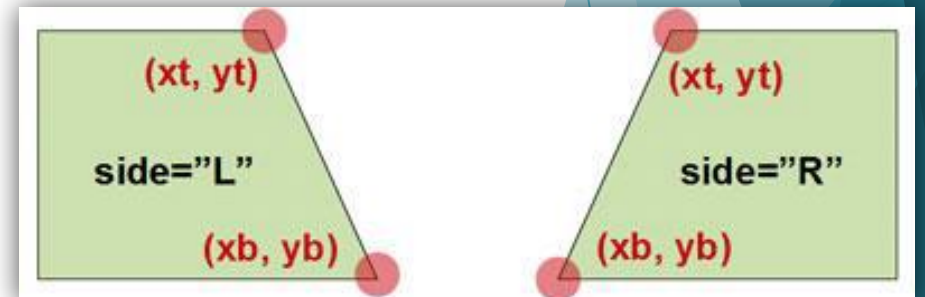
Textos en PDF

Este método debe incluirse antes de empezar a escribir nada en las páginas y requiere haber definido los espacios en blanco en `array_de_regiones` (array de 2D):

```
$array_de_regiones[] = array('page' => "pagina", 'xt' => num,  
'yt' => num, 'xb' => num, 'yb' => num, 'side' => "L");
```

```
$documento->setPageRegions(array_de_regiones)
```

- `page` → recoge en una cadena el número de la página en la reservamos el espacio
- `side` → "L" (izquierda) o "R" (derecha), para especificar si el espacio se guardará en el margen izquierdo o derecho de la página.
- `xt, yt, xb, yb` → coordenadas de los puntos señalados en la imagen. Si establecemos `xt` y `xb` con el mismo valor, el área reservada se transformará en un rectángulo.



EJERCICIO

Crea una nueva página llamada **ud07ejer01** que te permita mostrar un PDF.

El documento pdf te deberá permitir practicar todas (o casi todas) las funciones vistas hasta aquí en esta unidad.

Intenta ser original con tus creaciones y no escribas código sin sentido.

