

Main command TCPDF

`AddPage($orientation = '', $format = '', $keepmargins = false, $tocpage = false)`

Adds a new page to the document. If a page is already present, the Footer() method is called first to output the footer (if enabled). Then the page is added, the current position set to the top-left corner according to the left and top margins (or top-right if in RTL mode), and Header() is called to display the header (if enabled). The origin of the coordinate system is at the top-left corner (or top-right for RTL) and increasing ordinates go downwards.

`endPage($tocpage = false)`

Terminate the current page

`Ln($h = '', $cell = false)`

Performs a line break. The current abscissa goes back to the left margin and the ordinate increases by the amount passed in parameter.

`GetX()`

Returns the relative X value of current position. The value is relative to the left border for LTR languages and to the right border for RTL languages.

`GetAbsX()`

Returns the absolute X value of current position.

`GetY()`

Returns the ordinate of the current position.

`SetX($x, $rtloff = false)`

Defines the abscissa of the current position. If the passed value is negative, it is relative to the right of the page (or left if language is RTL).

`SetY($y, $resetx = true, $rtloff = false)`

Moves the current abscissa back to the left margin and sets the ordinate. If the passed value is negative, it is relative to the bottom of the page.

`SetXY($x, $y, $rtloff = false)`

Defines the abscissa and ordinate of the current position. If the passed values are negative, they are relative respectively to the right and bottom of the page.

`SetAbsX($x)`

Set the absolute X coordinate of the current pointer.

`SetAbsY($y)`

Set the absolute Y coordinate of the current pointer.

`SetAbsXY($x, $y)`

Set the absolute X and Y coordinates of the current pointer.

`SetMargins($left, $top, $right = -1, $keepmargins = false)`

Defines the left, top and right margins.

`SetLeftMargin($margin)`

Defines the left margin. The method can be called before creating the first page. If the current abscissa gets out of page, it is brought back to the margin.

`SetTopMargin($margin)`

Defines the top margin. The method can be called before creating the first page.

`SetRightMargin($margin)`

Defines the right margin. The method can be called before creating the first page.

`SetCellPadding($pad)`

Set the same internal Cell padding for top, right, bottom, left-

`setCellPaddings($left = '', $top = '', $right = '', $bottom = '')`

Set the internal Cell paddings.

`getCellPaddings()`

Get the internal Cell padding array.

```
Text( $x, $y, $txt, $fstroke = false, $fclip = false, $ffill = true, $border = 0, $ln = 0, $align = '', $fill = false, $link = ''
, $stretch = 0, $ignore_min_height = false, $calign = 'T', $valign = 'M', $rtloff = false )
```

Prints a text cell at the specified position. This method allows to place a string precisely on the page.

```
AcceptPageBreak( )
```

Whenever a page break condition is met, the method is called, and the break is issued or not depending on the returned value. The default implementation returns a value according to the mode selected by SetAutoPageBreak().

This method is called automatically and should not be called directly by the application.

```
checkPageBreak( $h = 0, $y = '', $addpage = true )
```

Add page if needed.

```
Cell( $w, $h = 0, $txt = '', $border = 0, $ln = 0, $align = '', $fill = false, $link = '', $stretch = 0,
$ignore_min_height = false, $calign = 'T', $valign = 'M' )
```

Prints a cell (rectangular area) with optional borders, background color and character string. The upper-left corner of the cell corresponds to the current position. The text can be aligned or centered. After the call, the current position moves to the right or to the next line. It is possible to put a link on the text.

If automatic page breaking is enabled and the cell goes beyond the limit, a page break is done before outputting.

```
getCellCode( $w, $h = 0, $txt = '', $border = 0, $ln = 0, $align = '', $fill = false, $link = '', $stretch = 0,
$ignore_min_height = false, $calign = 'T', $valign = 'M' )
```

Returns the PDF string code to print a cell (rectangular area) with optional borders, background color and character string. The upper-left corner of the cell corresponds to the current position. The text can be aligned or centered. After the call, the current position moves to the right or to the next line. It is possible to put a link on the text.

If automatic page breaking is enabled and the cell goes beyond the limit, a page break is done before outputting.

```
MultiCell( $w, $h, $txt, $border = 0, $align = 'J', $fill = false, $ln = 1, $x = '', $y = '', $reseth = true, $stretch = 0,
$ishtml = false, $autopadding = true, $maxh = 0, $valign = 'T', $fitcell = false )
```

This method allows printing text with line breaks. They can be automatic (as soon as the text reaches the right border of the cell) or explicit (via the \n character). As many cells as necessary are output, one below the other.

Text can be aligned, centered or justified. The cell block can be framed and the background painted.

```
getNumLines( $txt, $w = 0, $reseth = false, $autopadding = true, $cellpadding = '', $border = 0 )
```

This method return the estimated number of lines for print a simple text string using Multicell() method.

```
getStringHeight( $w, $txt, $reseth = false, $autopadding = true, $cellpadding = '', $border = 0 )
```

This method return the estimated height needed for printing a simple text string using the Multicell() method. Generally, if you want to know the exact height for a block of content you can use the following alternative technique:

```
Write( $h, $txt, $link = '', $fill = false, $align = '', $ln = false, $stretch = 0, $firstline = false, $firstblock = false,
$maxh = 0, $wadj = 0, $margin = '' )
```

This method prints text from the current position.

```
getRemainingWidth( )
```

Returns the remaining width between the current position and margins.

```
fitBlock( $w, $h, $x, $y, $fitonpage = false )
```

Set the block dimensions accounting for page breaks and page/column fitting

```
Image( $file, $x = '', $y = '', $w = 0, $h = 0, $type = '', $link = '', $align = '', $resize = false, $dpi = 300, $palign = '',
$ismask = false, $imgmask = false, $border = 0, $fitbox = false, $hidden = false, $fitonpage = false, $alt = false,
$altimgs = array() )
```

Puts an image in the page. The upper-left corner must be given. The dimensions can be specified in different ways:

- explicit width and height (expressed in user unit)
- one explicit dimension, the other being calculated automatically in order to keep the original proportions
- no explicit dimension, in which case the image is put at 72 dpi

Supported formats are JPEG and PNG images whitout GD library and all images supported by GD: GD, GD2, GD2PART, GIF, JPEG, PNG, BMP, XBM, XPM; The format can be specified explicitly or inferred from the file extension.

It is possible to put a link on the image.

```
Output( $name = 'doc.pdf', $dest = 'I' )
```

Send the document to a given destination: string, local file or browser. In the last case, the plug-in may be used (if present) or a download ("Save as" dialog box) may be forced.

The method first calls Close() if necessary to terminate the document.