

# Direct writing to document

## Other Methods

mPDF is optimised to accept HTML code and CSS styles. Apart from WriteHTML() there are other methods that can be used to write to the PDF document, but these do not always have full functionality. These are methods available in the original FPDF and successors, and if these are all you are using you may find that you do not need to use mPDF with its extra functions that slow the program down - see FPDF and UFPDF for further details.

The methods Cell() and Text() from FPDF are still present, but should not be used directly as they will not cope with UTF-8 encoded text. Use the WriteCell() and WriteText() methods instead.

All the methods described below handle UTF-8 encoded text, and all but AutosizeText() and watermark() will reverse RTL (right-to-left) text when appropriate.

### Direct writing methods and OTL (updated: mPDF >= 6.0)

- WriteText() WriteCell() Watermark() AutosizeText() and ShadedBox() DO support complex scripts and right-to-left text (RTL).
- Write() does NOT support complex scripts or RTL (NB this is a change - Write() used to support RTL).
- CircularText() does NOT support complex scripts or RTL.
- MultiCell() DOES support complex scripts and RTL, but complex-script line-breaking MAY NOT be accurate.
- MultiCell() does not support kerning and justification. NB This includes <textarea> in forms which uses MultiCell() internally.
- <select> form objects also do NOT support kerning.

Text containing HTML entities, as well as decimal and hex e.g. & apos; & #8812; or & #x21a4; can be used in all of these methods, by setting:

```
$mpdf->text_input_as_HTML = true; (default = false)
```

This will convert all the above to their appropriate characters, otherwise the text will be output as it is.

For most of the methods, you are referred to the originals at [FPDF >> Manual](#) for more information.

**WriteCell(float w, float h, string text[, mixed border[, integer ln[, string align[, integer fill[, mixed link[, float returnx]]]]]])**

Writes a single line of text directly to the PDF document at the current position.

See the details for Cell() at FPDF. An additional parameter returnx has been added; if ln is set, the current position moves not to the left margin, but to the value set as returnx.

**WriteText(float w, float h, string text)**

Writes a single line of text directly to the PDF document at a specified position.

See the details for Text() at FPDF.

**MultiCell(float w, float h, string text[, mixed border[, string align[, integer fill[, mixed link[, string directionality[, boolean encoded]]]]]])**

Writes a block of text directly to the PDF document at the current position. Lines are wrapped at the margins.

See the details for MultiCell() at FPDF. Two additional parameters have been added:

- directionality  
Set to 'rtl' if using RTL language (right-to-left)  
Default = 'ltr'
- encoded  
When set to false (default), UTF-8 encoded text will be appropriately converted to the chosen output file format. It should only be set to true, when the input text has already been converted internally within the program.  
Default = false

**SetX(float x)**  
**SetY(float y)**  
**SetXY(float x, float y)**

Sets the co-ordinates for the current position to write. Note only millimeters can be used as units. X and Y are measured from the top-left corner of the page.

See the details these methods at FPDF

**AutosizeText(string text, float width, string font, string style[, float fontsize])**  
 Writes a single line of text directly to the PDF document at the current position.  
 Font size will be automatically reduced to fit width (but is not increased).  
 NB Does not reverse RTL text

- text  
UTF-8 encoded text to write. Single line only.
- width  
Width of text in millimeters. The font size will be reduced if required to fit this size.
- font  
Font family to use
- style  
Font style used [blank for normal]|i|b|bi
- fontsize  
Maximm font size in points (pt)  
Default = 72

**watermark(string text[, float angle[, float fontsize[, float alpha]])**  
 Writes a single line of text centred on the page, which can be rotated and transparent i.e. a watermark.  
 NB Does not reverse RTL text.

- text  
UTF-8 encoded text
- angle  
Angle of rotation for text (anticlockwise rotation from horizontal)  
Default = 45
- fontsize  
Font size in points (pt)  
Default = 96
- alpha  
Transparency  
Default = 0.2

**RoundedRect(float x, float y, float w, float h, float radius[, string style])**  
 Draws a rectangle with rounded corners directly to the PDF document at the specified position.

- x  
Abscissa of left edge of box - value in millimeters
- y  
Ordinate of top edge of box - value in millimeters
- w  
Width of the box - in millimeters
- h  
Width of the box - in millimeters
- radius  
Radius of the rounded corners
- style  
Box style: D or empty string - draw border (default); F - fill; DF or FD - draw and fill  
Default = "" i.e. border, no fill

**shaded\_box(string title[, string font[, float fontstyle[, float fontsize[, float width[, string style[, float radius[, string backgroundcolor[, string color[, float padding]]]]]]]])**

Writes a single line of text surrounded by a box directly to the PDF document at the current position. The box can have rounded corners, and be filled with background-colour.

- title  
UTF-8 encoded text (single line)
- font  
Font family  
Default = '' i.e. default document font
- fontstyle  
Font style as one of B (bold), I (italic), BI (bold-italic) or blank for normal  
Default = 'B' i.e. bold
- fontsize  
Font size in points (pt)  
Default = '' i.e. default document font size
- width  
Width of the box - any units acceptable in CSS can be used e.g. pt, px, mm, % (of page width)  
Default = '70%'
- style  
Box style: D or empty string - draw border (default); F - fill; DF or FD - draw and fill  
Default = 'DF' i.e. border and fill
- radius  
Radius of the rounded corners  
Default = 2.5
- backgroundcolor  
Fill colour for the box - as #rrggbb  
Default = '#FFFFFF'
- color  
Text colour - as #rrggbb  
Default = '#000000'
- padding  
Padding between text and box border, in millimeters  
Default = 2

### See Also

- preparePreText() - Prepares text to be output ignoring the HTML markup

Printed on Wed 05 Aug 2015 12:12:19 GMT +0100 (DST)