

# FPDF Library

PDF generator

## Manuel de référence de FPDF 1.52



# FPDF 1.52 Reference Manual

[AcceptPageBreak](#) - accept or not automatic page break

[AddFont](#) - add a new font

[AddLink](#) - create an internal link

[AddPage](#) - add a new page

[AliasNbPages](#) - define an alias for number of pages

[Cell](#) - print a cell

[Close](#) - terminate the document

[Error](#) - fatal error

[Footer](#) - page footer

[FPDF](#) - constructor

[GetStringWidth](#) - compute string length

[GetX](#) - get current x position

[GetY](#) - get current y position

[Header](#) - page header

[Image](#) - output an image

[Line](#) - draw a line

[Link](#) - put a link

[Ln](#) - line break

[MultiCell](#) - print text with line breaks

[Open](#) - create a document

[Output](#) - save or send the document

[PageNo](#) - page number

[Rect](#) - draw a rectangle

[SetAuthor](#) - set the document author

[SetAutoPageBreak](#) - set the automatic page breaking mode

[SetCompression](#) - turn compression on or off

[SetCreator](#) - set document creator

[SetDisplayMode](#) - set display mode

[SetDrawColor](#) - set drawing color

[SetFillColor](#) - set filling color

[SetFont](#) - set font

[SetFontSize](#) - set font size

[SetKeywords](#) - associate keywords with document

[SetLeftMargin](#) - set left margin

[SetLineWidth](#) - set line width

[SetLink](#) - set internal link destination

[SetMargins](#) - set margins

[SetRightMargin](#) - set right margin

[SetSubject](#) - set document subject

[SetTextColor](#) - set text color

[SetTitle](#) - set document title

[SetTopMargin](#) - set top margin

[SetX](#) - set current x position

[SetXY](#) - set current x and y positions

[SetY](#) - set current y position

[Text](#) - print a string

[Write](#) - print flowing text

# AcceptPageBreak

**boolean** AcceptPageBreak( )

## Version

1.4

## Description

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.

## Example

The method is overridden in an inherited class in order to obtain a 3 column layout:

```
class PDF extends FPDF
{
    var $col=0;

    function SetCol($col)
    {
        //Move position to a column
        $this->col=$col;
        $x=10+$col*65;
        $this->SetLeftMargin($x);
        $this->SetX($x);
    }

    function AcceptPageBreak()
    {
        if($this->col<2)
        {
            //Go to next column
            $this->SetCol($this->col+1);
            $this->SetY(10);
            return false;
        }
        else
        {
            //Go back to first column and issue page break
            $this->SetCol(0);
        }
    }
}
```

```
        return true;
    }
}
}

$pdf=new PDF();
$pdf->Open();
$pdf->AddPage();
$pdf->SetFont('Arial','',12);
for($i=1;$i<=300;$i++)
    $pdf->Cell(0,5,"Line $i",0,1);
$pdf->Output();
```

## See also

[SetAutoPageBreak\(\)](#).

# AddFont

AddFont(**string** family [, **string** style [, **string** file]])

## Version

1.5

## Description

Imports a TrueType or Type1 font and makes it available. It is necessary to generate a font definition file first with the makefont.php utility.

The definition file (and the font file itself when embedding) must be present either in the current directory or in the one indicated by FPDF\_FONTPATH if the constant is defined. If it could not be found, the error "Could not include font definition file" is generated.

## Parameters

family

Font family. The name can be chosen arbitrarily. If it is a standard family name, it will override the corresponding font.

style

Font style. Possible values are (case insensitive):

- empty string: regular
- B: bold
- I: italic
- BI or IB: bold italic

The default value is regular.

file

The font definition file.

By default, the name is built from the family and style, in lower case with no space.

## Example

```
$pdf->AddFont( 'Comic' , 'I' );
```

is equivalent to:

```
$pdf->AddFont( 'Comic' , 'I' , 'comici.php' );
```

## See also

[SetFont\(\)](#).

# AddLink

```
int AddLink( )
```

## Version

1.5

## Description

Creates a new internal link and returns its identifier. An internal link is a clickable area which directs to another place within the document.

The identifier can then be passed to `Cell()`, `Write()`, `Image()` or `Link()`. The destination is defined with `SetLink()`.

## See also

[Cell\(\)](#), [Write\(\)](#), [Image\(\)](#), [Link\(\)](#), [SetLink\(\)](#).



# AddPage

```
AddPage([string orientation])
```

## Version

1.0

## Description

Adds a new page to the document. If a page is already present, the Footer() method is called first to output the footer. Then the page is added, the current position set to the top-left corner according to the left and top margins, and Header() is called to display the header.

The font which was set before calling is automatically restored. There is no need to call SetFont() again if you want to continue with the same font. The same is true for colors and line width.

The origin of the coordinate system is at the top-left corner and increasing ordinates go downwards.

## Parameters

orientation

Page orientation. Possible values are (case insensitive):

- P or Portrait
- L or Landscape

The default value is the one passed to the constructor.

## See also

[FPDF\(\)](#), [Header\(\)](#), [Footer\(\)](#), [SetMargins\(\)](#).

# AliasNbPages

AliasNbPages([**string** alias])

## Version

1.4

## Description

Defines an alias for the total number of pages. It will be substituted as the document is closed.

## Parameters

alias

The alias. Default value: {nb}.

## Example

```
class PDF extends FPDF
{
function Footer()
{
    //Go to 1.5 cm from bottom
    $this->SetY(-15);
    //Select Arial italic 8
    $this->SetFont('Arial','I',8);
    //Print current and total page numbers
    $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');
}
}

$pdf=new PDF();
$pdf->AliasNbPages();
```

## See also

[PageNo\(\)](#), [Footer\(\)](#).

# Cell

```
Cell(float w [, float h [, string txt [, mixed border [, int ln [,  
string align [, int fill [, mixed link]]]]]])
```

## Version

1.0

## Description

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.

## Parameters

w

Cell width. If 0, the cell extends up to the right margin.

h

Cell height. Default value: 0.

txt

String to print. Default value: empty string.

border

Indicates if borders must be drawn around the cell. The value can be either a number:

- 0: no border
- 1: frame

or a string containing some or all of the following characters (in any order):

- L: left
- T: top

- R: right
- B: bottom

Default value: 0.

## ln

Indicates where the current position should go after the call. Possible values are:

- 0: to the right
- 1: to the beginning of the next line
- 2: below

Putting 1 is equivalent to putting 0 and calling Ln() just after. Default value: 0.

## align

Allows to center or align the text. Possible values are:

- L or empty string: left align (default value)
- C: center
- R: right align

## fill

Indicates if the cell background must be painted (1) or transparent (0). Default value: 0.

## link

URL or identifier returned by AddLink().

## **Example**

```
//Set font
$pdf->SetFont('Arial','B',16);
//Move to 8 cm to the right
$pdf->Cell(80);
//Centered text in a framed 20*10 mm cell and line break
$pdf->Cell(20,10,'Title',1,1,'C');
```

## **See also**

[SetFont\(\)](#), [SetDrawColor\(\)](#), [SetFillColor\(\)](#), [SetTextColor\(\)](#), [SetLineWidth\(\)](#), [AddLink\(\)](#), [Ln\(\)](#),

[MultiCell\(\)](#), [Write\(\)](#), [SetAutoPageBreak\(\)](#).

# Close

`Close()`

## Version

1.0

## Description

Terminates the PDF document. It is not necessary to call this method explicitly because `Output()` does it automatically.

If the document contains no page, `AddPage()` is called to prevent from getting an invalid document.

## See also

[Open\(\)](#), [Output\(\)](#).

# Error

Error(**string** msg)

## Version

1.0

## Description

This method is automatically called in case of fatal error; it simply outputs the message and halts the execution. An inherited class may override it to customize the error handling but should always halt the script, or the resulting document would probably be invalid.

## Parameters

msg

The error message.

# Footer

Footer( )

## Version

1.0

## Description

This method is used to render the page footer. It is automatically called by AddPage() and Close() and should not be called directly by the application. The implementation in FPDF is empty, so you have to subclass it and override the method if you want a specific processing.

## Example

```
class PDF extends FPDF
{
function Footer()
{
    //Go to 1.5 cm from bottom
    $this->SetY(-15);
    //Select Arial italic 8
    $this->SetFont('Arial','I',8);
    //Print centered page number
    $this->Cell(0,10,'Page '.$this->PageNo(),0,0,'C');
}
}
```

## See also

[Header\(\)](#).



# FPDF

```
FPDF([string orientation [, string unit [, mixed format]])
```

## Version

1.0

## Description

This is the class constructor. It allows to set up the page format, the orientation and the measure unit used in all the methods (except for the font sizes).

## Parameters

### orientation

Default page orientation. Possible values are (case insensitive):

- P or Portrait
- L or Landscape

Default value is P.

### unit

User measure unit. Possible values are:

- pt: point
- mm: millimeter
- cm: centimeter
- in: inch

A point equals 1/72 of inch, that is to say about 0.35 mm (an inch being 2.54 cm). This is a very common unit in typography; font sizes are expressed in that unit.

Default value is mm.

### format

The format used for pages. It can be either one of the following values (case insensitive):

- A3
- A4
- A5
- Letter
- Legal

or a custom format in the form of a two-element array containing the width and the height (expressed in the unit given by `unit`).

# GetStringWidth

```
float GetStringWidth(string s)
```

## Version

1.2

## Description

Returns the length of a string in user unit. A font must be selected.

## Parameters

s

The string whose length is to be computed.

# GetX

```
float GetX( )
```

## Version

1.2

## Description

Returns the abscissa of the current position.

## See also

[SetX\(\)](#), [GetY\(\)](#), [SetY\(\)](#).

# GetY

**float** GetY( )

## Version

1.0

## Description

Returns the ordinate of the current position.

## See also

[SetY\(\)](#), [GetX\(\)](#), [SetX\(\)](#).

# Header

Header( )

## Version

1.0

## Description

This method is used to render the page header. It is automatically called by AddPage() and should not be called directly by the application. The implementation in FPDF is empty, so you have to subclass it and override the method if you want a specific processing.

## Example

```
class PDF extends FPDF
{
function Header()
{
    //Select Arial bold 15
    $this->SetFont('Arial','B',15);
    //Move to the right
    $this->Cell(80);
    //Framed title
    $this->Cell(30,10,'Title',1,0,'C');
    //Line break
    $this->Ln(20);
}
}
```

## See also

[Footer\(\)](#).

# Image

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

## Version

1.1

## Description

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.

For JPEG, all flavors are allowed:

- gray scales
- true colors (24 bits)
- CMYK (32 bits)

For PNG, are allowed:

- gray scales on at most 8 bits (256 levels)
- indexed colors
- true colors (24 bits)

but are not supported:

- Interlacing
- Alpha channel

If a transparent color is defined, it will be taken into account (but will be only interpreted by Acrobat 4 and above).

The format can be specified explicitly or inferred from the file extension.

It is possible to put a link on the image.

Remark: if an image is used several times, only one copy will be embedded in the file.

## Parameters

file

Name of the file containing the image.

x

Abscissa of the upper-left corner.

y

Ordinate of the upper-left corner.

w

Width of the image in the page. If not specified or equal to zero, it is automatically calculated.

h

Height of the image in the page. If not specified or equal to zero, it is automatically calculated.

type

Image format. Possible values are (case insensitive): JPG, JPEG, PNG. If not specified, the type is inferred from the file extension.

link

URL or identifier returned by `AddLink()`.

## See also

[AddLink\(\)](#).



# Line

```
Line(float x1, float y1, float x2, float y2)
```

## Version

1.0

## Description

Draws a line between two points.

## Parameters

x1

Abscissa of first point.

y1

Ordinate of first point.

x2

Abscissa of second point.

y2

Ordinate of second point.

## See also

[SetLineWidth\(\)](#), [SetDrawColor\(\)](#).

# Link

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

## Version

1.5

## Description

Puts a link on a rectangular area of the page. Text or image links are generally put via `Cell()`, `Write()` or `Image()`, but this method can be useful for instance to define a clickable area inside an image.

## Parameters

x

Abscissa of the upper-left corner of the rectangle.

y

Ordinate of the upper-left corner of the rectangle.

w

Width of the rectangle.

h

Height of the rectangle.

link

URL or identifier returned by `AddLink()`.

## See also

[AddLink\(\)](#), [Cell\(\)](#), [Write\(\)](#), [Image\(\)](#).

# Ln

`Ln([float h])`

## Version

1.0

## Description

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

## Parameters

h

The height of the break.

By default, the value equals the height of the last printed cell.

## See also

[Cell\(\)](#).

# MultiCell

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

## Version

1.3

## Description

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.

## Parameters

w

Width of cells. If 0, they extend up to the right margin of the page.

h

Height of cells.

txt

String to print.

border

Indicates if borders must be drawn around the cell block. The value can be either a number:

- 0: no border
- 1: frame

or a string containing some or all of the following characters (in any order):

- L: left
- T: top
- R: right

- B: bottom

Default value: 0.

## align

Sets the text alignment. Possible values are:

- L: left alignment
- C: center
- R: right alignment
- J: justification (default value)

## fill

Indicates if the cell background must be painted (1) or transparent (0). Default value: 0.

## **See also**

[SetFont\(\)](#), [SetDrawColor\(\)](#), [SetFillColor\(\)](#), [SetTextColor\(\)](#), [SetLineWidth\(\)](#), [Cell\(\)](#), [Write\(\)](#), [SetAutoPageBreak\(\)](#).

# Open

Open( )

## Version

1.0

## Description

This method begins the generation of the PDF document. It is not necessary to call it explicitly because `AddPage()` does it automatically.

Note: no page is created by this method.

## See also

[AddPage\(\)](#), [Close\(\)](#).

# Output

```
string Output([string name [, string dest]])
```

## Version

1.0

## Description

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.

## Parameters

name

The name of the file. If not given, the document will be sent to the browser (destination I) with the name `doc.pdf`.

dest

Destination where to send the document. It can take one of the following values:

- I: send the file inline to the browser. The plug-in is used if available. The name given by name is used when one selects the "Save as" option on the link generating the PDF.
- D: send to the browser and force a file download with the name given by name.
- F: save to a local file with the name given by name.
- S: return the document as a string. name is ignored.

If the parameter is not specified but a name is given, destination is F. If no parameter is specified at all, destination is I.

Note: for compatibility with previous versions, a boolean value is also accepted (`false` for F and `true` for D).

## See also

[Close\(\)](#).

# PageNo

```
int PageNo( )
```

## Version

1.0

## Description

Returns the current page number.

## See also

[AliasNbPages\(\)](#).



# Rect

```
Rect(float x, float y, float w, float h [, string style])
```

## Version

1.0

## Description

Outputs a rectangle. It can be drawn (border only), filled (with no border) or both.

## Parameters

x

Abscissa of upper-left corner.

y

Ordinate of upper-left corner.

w

Width.

h

Height.

style

Style of rendering. Possible values are:

- D or empty string: draw. This is the default value.
- F: fill
- DF or FD: draw and fill

## See also

[SetLineWidth\(\)](#), [SetDrawColor\(\)](#), [SetFillColor\(\)](#).

# SetAuthor

SetAuthor(**string** author)

## Version

1.2

## Description

Defines the author of the document.

## Parameters

author

The name of the author.

## See also

[SetCreator\(\)](#), [SetKeywords\(\)](#), [SetSubject\(\)](#), [SetTitle\(\)](#).

# SetAutoPageBreak

SetAutoPageBreak(**boolean** auto [, **float** margin])

## Version

1.0

## Description

Enables or disables the automatic page breaking mode. When enabling, the second parameter is the distance from the bottom of the page that defines the triggering limit. By default, the mode is on and the margin is 2 cm.

## Parameters

auto

Boolean indicating if mode should be on or off.

margin

Distance from the bottom of the page.

## See also

[Cell\(\)](#), [MultiCell\(\)](#), [AcceptPageBreak\(\)](#).

# SetCompression

SetCompression(**boolean** compress)

## Version

1.4

## Description

Activates or deactivates page compression. When activated, the internal representation of each page is compressed, which leads to a compression ratio of about 2 for the resulting document. Compression is on by default.

**Note:** the Zlib extension is required for this feature. If not present, compression will be turned off.

## Parameters

compress

Boolean indicating if compression must be enabled.

# SetCreator

SetCreator(**string** creator)

## Version

1.2

## Description

Defines the creator of the document. This is typically the name of the application that generates the PDF.

## Parameters

creator

The name of the creator.

## See also

[SetAuthor\(\)](#), [SetKeywords\(\)](#), [SetSubject\(\)](#), [SetTitle\(\)](#).

# SetDisplayMode

SetDisplayMode(**mixed** zoom [, **string** layout])

## Version

1.2

## Description

Defines the way the document is to be displayed by the viewer. The zoom level can be set: pages can be displayed entirely on screen, occupy the full width of the window, use real size, be scaled by a specific zooming factor or use viewer default (configured in the Preferences menu of Acrobat). The page layout can be specified too: single at once, continuous display, two columns or viewer default. By default, documents use the full width mode with continuous display.

## Parameters

### zoom

The zoom to use. It can be one of the following string values:

- `fullpage`: displays the entire page on screen
- `fullwidth`: uses maximum width of window
- `real`: uses real size (equivalent to 100% zoom)
- `default`: uses viewer default mode

or a number indicating the zooming factor to use.

### layout

The page layout. Possible values are:

- `single`: displays one page at once
- `continuous`: displays pages continuously
- `two`: displays two pages on two columns
- `default`: uses viewer default mode

Default value is `continuous`.

# SetDrawColor

```
SetDrawColor(int r [, int g, int b])
```

## Version

1.3

## Description

Defines the color used for all drawing operations (lines, rectangles and cell borders). It can be expressed in RGB components or gray scale. The method can be called before the first page is created and the value is retained from page to page.

## Parameters

r

If g et b are given, red component; if not, indicates the gray level. Value between 0 and 255.

g

Green component (between 0 and 255).

b

Blue component (between 0 and 255).

## See also

[SetFillColor\(\)](#), [SetTextColor\(\)](#), [Line\(\)](#), [Rect\(\)](#), [Cell\(\)](#), [MultiCell\(\)](#).

# SetFillColor

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

## Version

1.3

## Description

Defines the color used for all filling operations (filled rectangles and cell backgrounds). It can be expressed in RGB components or gray scale. The method can be called before the first page is created and the value is retained from page to page.

## Parameters

r

If g and b are given, red component; if not, indicates the gray level. Value between 0 and 255.

g

Green component (between 0 and 255).

b

Blue component (between 0 and 255).

## See also

[SetDrawColor\(\)](#), [SetTextColor\(\)](#), [Rect\(\)](#), [Cell\(\)](#), [MultiCell\(\)](#).



# SetFont

SetFont(**string** family [, **string** style [, **float** size]])

## Version

1.0

## Description

Sets the font used to print character strings. It is mandatory to call this method at least once before printing text or the resulting document would not be valid.

The font can be either a standard one or a font added via the AddFont() method. Standard fonts use Windows encoding cp1252 (Western Europe).

The method can be called before the first page is created and the font is retained from page to page. If you just wish to change the current font size, it is simpler to call SetFontSize().

**Note:** for the standard fonts, the font metric files must be accessible. There are three possibilities for this:

- They are in the current directory (the one where the running script lies)
- They are in one of the directories defined by the `include_path` parameter
- They are in the directory defined by the `FPDF_FONTPATH` constant

Example for the last case (note the trailing slash):

```
define( 'FPDF_FONTPATH' , ' /home/www/font/' );
require( 'fpdf.php' );
```

If the file corresponding to the requested font is not found, the error "Could not include font metric file" is generated.

## Parameters

family

Family font. It can be either a name defined by AddFont() or one of the standard families (case insensitive):

- Courier (fixed-width)
- Helvetica or Arial (synonymous; sans serif)
- Times (serif)
- Symbol (symbolic)

- ZapfDingbats (symbolic)

It is also possible to pass an empty string. In that case, the current family is retained.

## style

Font style. Possible values are (case insensitive):

- empty string: regular
- B: bold
- I: italic
- U: underline

or any combination. The default value is regular. Bold and italic styles do not apply to Symbol and ZapfDingbats.

## size

Font size in points.

The default value is the current size. If no size has been specified since the beginning of the document, the value taken is 12.

## Example

```
//Times regular 12
$pdf->SetFont('Times');
//Arial bold 14
$pdf->SetFont('Arial','B',14);
//Removes bold
$pdf->SetFont('');
//Times bold, italic and underlined 14
$pdf->SetFont('Times','BIU');
```

## See also

[AddFont\(\)](#), [SetFontSize\(\)](#), [Cell\(\)](#), [MultiCell\(\)](#), [Write\(\)](#).

# SetFontSize

SetFontSize(**float** size)

## Version

1.0

## Description

Defines the size of the current font.

## Parameters

size

The size (in points).

## See also

[SetFont\(\)](#).

# SetKeywords

SetKeywords(**string** keywords)

## Version

1.2

## Description

Associates keywords with the document, generally in the form 'keyword1 keyword2 ...'.

## Parameters

keywords

The list of keywords.

## See also

[SetAuthor\(\)](#), [SetCreator\(\)](#), [SetSubject\(\)](#), [SetTitle\(\)](#).

# SetLeftMargin

SetLeftMargin(**float** margin)

## Version

1.4

## Description

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.

## Parameters

margin

The margin.

## See also

[SetTopMargin\(\)](#), [SetRightMargin\(\)](#), [SetAutoPageBreak\(\)](#), [SetMargins\(\)](#).

# SetLineWidth

```
SetLineWidth(float width)
```

## Version

1.0

## Description

Defines the line width. By default, the value equals 0.2 mm. The method can be called before the first page is created and the value is retained from page to page.

## Parameters

width

The width.

## See also

[Line\(\)](#), [Rect\(\)](#), [Cell\(\)](#), [MultiCell\(\)](#).

# SetLink

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

## Version

1.5

## Description

Defines the page and position a link points to.

## Parameters

link

The link identifier returned by AddLink().

y

Ordinate of target position; -1 indicates the current position. The default value is 0 (top of page).

page

Number of target page; -1 indicates the current page. This is the default value.

## See also

[AddLink\(\)](#).

# SetMargins

```
SetMargins(float left, float top [, float right])
```

## Version

1.0

## Description

Defines the left, top and right margins. By default, they equal 1 cm. Call this method to change them.

## Parameters

left

Left margin.

top

Top margin.

right

Right margin. Default value is the left one.

## See also

[SetLeftMargin\(\)](#), [SetTopMargin\(\)](#), [SetRightMargin\(\)](#), [SetAutoPageBreak\(\)](#).



# SetRightMargin

SetRightMargin(**float** margin)

## Version

1.5

## Description

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

## Parameters

margin

The margin.

## See also

[SetLeftMargin\(\)](#), [SetTopMargin\(\)](#), [SetAutoPageBreak\(\)](#), [SetMargins\(\)](#).

# SetSubject

```
SetSubject(string subject)
```

## Version

1.2

## Description

Defines the subject of the document.

## Parameters

subject

The subject.

## See also

[SetAuthor\(\)](#), [SetCreator\(\)](#), [SetKeywords\(\)](#), [SetTitle\(\)](#).

# SetTextColor

```
SetTextColor(int r [, int g, int b])
```

## Version

1.3

## Description

Defines the color used for text. It can be expressed in RGB components or gray scale. The method can be called before the first page is created and the value is retained from page to page.

## Parameters

r

If g et b are given, red component; if not, indicates the gray level. Value between 0 and 255.

g

Green component (between 0 and 255).

b

Blue component (between 0 and 255).

## See also

[SetDrawColor\(\)](#), [SetFillColor\(\)](#), [Text\(\)](#), [Cell\(\)](#), [MultiCell\(\)](#).

# SetTitle

```
SetTitle(string title)
```

## Version

1.2

## Description

Defines the title of the document.

## Parameters

title

The title.

## See also

[SetAuthor\(\)](#), [SetCreator\(\)](#), [SetKeywords\(\)](#), [SetSubject\(\)](#).

# SetTopMargin

SetTopMargin(**float** margin)

## Version

1.5

## Description

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

## Parameters

margin

The margin.

## See also

[SetLeftMargin\(\)](#), [SetRightMargin\(\)](#), [SetAutoPageBreak\(\)](#), [SetMargins\(\)](#).

# SetX

SetX(**float** x)

## Version

1.2

## Description

Defines the abscissa of the current position. If the passed value is negative, it is relative to the right of the page.

## Parameters

x

The value of the abscissa.

## See also

[GetX\(\)](#), [GetY\(\)](#), [SetY\(\)](#), [SetXY\(\)](#).

# SetXY

```
SetXY(float x, float y)
```

## Version

1.2

## Description

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.

## Parameters

x

The value of the abscissa.

y

The value of the ordinate.

## See also

[SetX\(\)](#), [SetY\(\)](#).

# SetY

SetY(**float** y)

## Version

1.0

## Description

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.

## Parameters

y

The value of the ordinate.

## See also

[GetX\(\)](#), [GetY\(\)](#), [SetY\(\)](#), [SetXY\(\)](#).



# Text

`Text(float x, float y, string txt)`

## Version

1.0

## Description

Prints a character string. The origin is on the left of the first character, on the baseline. This method allows to place a string precisely on the page, but it is usually easier to use `Cell()`, `MultiCell()` or `Write()` which are the standard methods to print text.

## Parameters

x

Abscissa of the origin.

y

Ordinate of the origin.

txt

String to print.

## See also

[SetFont\(\)](#), [SetTextColor\(\)](#), [Cell\(\)](#), [MultiCell\(\)](#), [Write\(\)](#).

# Write

Write(**float** h, **string** txt [, **mixed** link])

## Version

1.5

## Description

This method prints text from the current position. When the right margin is reached (or the \n character is met) a line break occurs and text continues from the left margin. Upon method exit, the current position is left just at the end of the text.  
It is possible to put a link on the text.

## Parameters

h

Line height.

txt

String to print.

link

URL or identifier returned by AddLink().

## Example

```
//Begin with regular font
$pdf->SetFont('Arial','',14);
$pdf->Write(5,'Visit ');
//Then put a blue underlined link
$pdf->SetTextColor(0,0,255);
$pdf->SetFont('','U');
$pdf->Write(5,'www.fpdf.org','http://www.fpdf.org');
```

## See also

[SetFont\(\)](#), [SetTextColor\(\)](#), [AddLink\(\)](#), [MultiCell\(\)](#), [SetAutoPageBreak\(\)](#).