

# Index des méthodes de FPDF 1.3

[AddPage](#) - ajoute une nouvelle page  
[Cell](#) - imprime une cellule  
[Close](#) - termine le document  
[Error](#) - erreur fatale  
[Footer](#) - pied de page  
[FPDF](#) - constructeur  
[GetStringWidth](#) - calcule la longueur d'une chaîne  
[GetX](#) - renvoie la position x courante  
[GetY](#) - renvoie la position y courante  
[Header](#) - en-tête  
[Image](#) - imprime une image  
[Line](#) - trace une ligne  
[Ln](#) - saut de ligne  
[MultiCell](#) - imprime du texte avec saut de ligne  
[Open](#) - crée un document  
[Output](#) - sauve ou envoie le document  
[PageNo](#) - numéro de page  
[Rect](#) - trace un rectangle  
[SetAuthor](#) - définit l'auteur du document  
[SetAutoPageBreak](#) - fixe le mode saut de page automatique  
[SetDisplayMode](#) - fixe le mode d'affichage  
[SetCreator](#) - définit le créateur du document  
[SetDrawColor](#) - définit la couleur de tracé  
[SetFillColor](#) - définit la couleur de remplissage  
[SetFont](#) - fixe la police  
[SetFontSize](#) - fixe la taille de la police  
[SetKeywords](#) - définit les mots-clés associés au document  
[SetLineWidth](#) - fixe l'épaisseur des traits  
[SetMargins](#) - fixe les marges  
[SetSubject](#) - définit le sujet du document  
[SetTextColor](#) - définit la couleur du texte  
[SetTitle](#) - définit le titre du document  
[SetX](#) - fixe la position x courante  
[SetXY](#) - fixe les positions x et y courantes  
[SetY](#) - fixe la position y courante  
[Text](#) - imprime une chaîne

# AddPage

AddPage ( )

## Description

Ajoute une nouvelle page au document. Si une page était en cours, la méthode Footer() est appelée pour traiter le pied de page. Puis la page est ajoutée, la position courante mise en haut à gauche en fonction des marges gauche et haute, et Header() est appelée pour afficher l'en-tête.

La police qui était en cours au moment de l'appel est automatiquement restaurée. Il n'est donc pas nécessaire d'appeler à nouveau SetFont() si vous souhaitez continuer avec la même police.

L'origine est en haut à gauche et les ordonnées croissantes vont vers le bas.

## Voir

[Header\(\)](#), [Footer\(\)](#), [SetFont\(\)](#).

# Cell

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

## Description

Imprime une cellule (zone rectangulaire) avec éventuellement un cadre et une chaîne de caractères. Le coin supérieur gauche de la cellule correspond à la position courante. Le texte peut être aligné ou centré. Après l'appel, la position courante se déplace à droite ou un retour à la ligne est effectué. La chaîne doit utiliser l'encodage de caractères de Windows. Si le saut de page automatique est activé et que la cellule dépasse le seuil de déclenchement, un saut de page est effectué avant de l'imprimer.

## Paramètres

w

Largeur de la cellule. Si elle vaut 0, la cellule s'étend jusqu'à la marge droite de la page (la marge droite étant égale à la marge gauche).

h

Hauteur de la cellule. Valeur par défaut : 0.

txt

Chaîne à imprimer. Valeur par défaut : chaîne vide.

border

Indique si un cadre doit être tracé autour de la cellule (0 : non, 1 : oui). Valeur par défaut : 0.

ln

Indique si la position courante se décale à droite (0) ou passe à la ligne (1). Mettre 1 est équivalent à appeler la méthode Ln() juste après Cell(). Valeur par défaut : 0.

align

Permet de centrer ou d'aligner le texte. Les valeurs possibles sont :

- L ou chaîne vide : alignement à gauche (valeur par défaut)
- C : centrage
- R : alignement à droite

fill

Indique si le fond de la cellule doit être coloré (1) ou transparent (0). Valeur par défaut : 0.

## Exemple

```
//Sélection de la police  
$pdf->SetFont('Arial','B',16);  
//Décalage de 8 cm à droite  
$pdf->Cell(80);  
//Texte centré dans une cellule 20*10 mm encadrée et retour à la ligne  
$pdf->Cell(20,10,'Titre',1,1,'C');
```

## Voir

[SetFont\(\)](#), [SetDrawColor\(\)](#), [SetFillColor\(\)](#), [SetTextColor\(\)](#), [SetLineWidth\(\)](#), [Ln\(\)](#), [MultiCell\(\)](#), [SetAutoPageBreak\(\)](#).

# Close

Close( )

## Description

Cette méthode termine la génération du fichier PDF. Il n'est pas nécessaire de l'appeler explicitement car Output() le fait automatiquement.

## Voir

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

# Error

Error(**string** msg)

## Description

Cette méthode est appelée automatiquement en cas d'erreur fatale; elle se contente d'afficher le message et de mettre fin à l'exécution du script. Une classe dérivée peut la redéfinir mais elle doit impérativement arrêter l'exécution du script, sinon le fichier PDF résultant serait probablement invalide.

## Paramètres

msg

Le message d'erreur.

# Footer

Footer()

## Description

Cette méthode permet de définir le pied de page. Elle est appelée automatiquement par AddPage() et Close() et ne devrait donc pas être appelée explicitement par l'application. L'implémentation de Footer() dans FPDF est vide, donc vous devez dériver la classe et redéfinir la méthode si vous voulez un traitement particulier pour vos pieds de page.

## Exemple

```
class PDF extends FPDF
{

//Constructeur (obligatoire pour PHP3)
function PDF()
{
    $this->FPDF();
}

//Pied de page
function Footer()
{
    //Police Arial italique 8
    $this->SetFont('Arial','I',8);
    //Positionnement à 1,5 cm du bas
    $this->SetY(-15);
    //Décalage
    $this->Cell(80);
    //Numéro de page
    $this->Cell(30,10,'Page '.$this->PageNo());
}

}
```

## Voir

[Header\(\)](#).

# FPDF

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

## Description

Il s'agit du constructeur de la classe. Il permet de fixer l'orientation des pages ainsi que l'unité de mesure utilisée dans toutes les méthodes (sauf pour les tailles de police).

## Paramètres

### orientation

Orientation des pages. Les valeurs possibles sont (indépendamment de la casse) :

- P ou Portrait
- L ou Landscape

La valeur par défaut est P.

### unit

Unité de mesure utilisateur. Les valeurs possibles sont :

- pt : point
- mm : millimètre
- cm : centimètre
- in : pouce

Un point est égal à 1/72ème de pouce, c'est-à-dire environ 0,35 millimètre. C'est une unité très courante en typographie; les tailles de police sont exprimées dans cette unité. Un pouce vaut 2,54 centimètres.

La valeur par défaut est mm .



# GetStringWidth

```
double GetStringWidth(string s)
```

## Version

1.2

## Description

Renvoie la longueur de la chaîne en unité utilisateur. Une police doit être sélectionnée.

## Paramètres

s

La chaîne dont on veut déterminer la longueur.

# GetX

`double GetX( )`

## Version

1.2

## Description

Renvoie l'abscisse de la position courante.

## Voir

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

# GetY

**double** GetY( )

## Description

Renvoie l'ordonnée de la position courante. Elle peut être utile par exemple pour déterminer manuellement une condition de saut de page.

## Voir

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

# Header

Header ( )

## Description

Cette méthode permet de définir l'en-tête de page. Elle est appelée automatiquement par AddPage() et ne devrait donc pas être appelée explicitement par l'application. L'implémentation de Header() dans FPDF est vide, donc vous devez dériver la classe et redéfinir la méthode si vous voulez un traitement particulier pour vos en-têtes.

## Exemple

```
class PDF extends FPDF
{

//Constructeur (obligatoire pour PHP3)
function PDF()
{
    $this->FPDF();
}

//En-tête
function Header()
{
    //Police Arial gras 15
    $this->SetFont('Arial','B',15);
    //Décalage
    $this->Cell(80);
    //Titre
    $this->Cell(30,10,'Titre');
    //Saut de ligne
    $this->Ln(20);
}

}
```

## Voir

[Footer\(\)](#).

# Image

```
Image(string file, double x, double y, double w [, double h [,  
string type]])
```

## Version

1.1

## Description

Imprime une image dans la page. Le coin supérieur gauche ainsi que la largeur doivent être spécifiés; la hauteur peut être précisée ou calculée automatiquement afin de respecter les proportions de l'image.

Les formats supportés sont le JPEG et le PNG.

Pour le JPEG, toutes les variantes sont autorisées :

- niveaux de gris
- couleurs vraies (24 bits)
- CMYK (32 bits)

Pour le PNG, sont autorisées :

- les images en niveaux de gris sur 8 bits au plus (256 teintes)
- les images en couleurs indexées
- les images en couleurs vraies (24 bits)

mais ne sont pas supportés :

- L'interlacement
- L'alpha channel

Si une couleur transparente est définie, elle sera prise en compte (mais ne sera interprétée qu'à partir d'Acrobat 4).

**Note :** dans la version 1.1, l'utilisation du PNG requiert PHP 4 et l'extension Zlib. Ce n'est plus le cas à partir de la version 1.11.

Le format peut être spécifié explicitement ou déduit de l'extension du fichier.

Si une image est utilisée plusieurs fois, une seule copie sera intégrée au document.

## Paramètres

file

Nom du fichier contenant l'image.

x

Abscisse du coin supérieur gauche.

y

Ordonnée du coin supérieur gauche.

w

Largeur de l'image dans la page.

h

Hauteur de l'image dans la page. Si elle n'est pas indiquée ou vaut zéro, elle est automatiquement calculée afin de respecter les proportions d'origine.

type

Format de l'image. Les valeurs possibles sont (indépendamment de la casse) : JPG, JPEG, PNG. S'il n'est pas précisé, le type est déduit de l'extension du fichier.

type est déduit de l'extension du fichier.

# Line

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

## Description

Trace une ligne entre deux points.

## Paramètres

x1

Abscisse du premier point.

y1

Ordonnée du premier point.

x2

Abscisse du second point.

y2

Ordonnée du second point.

## Voir

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

# Ln

`Ln([double h])`

## Description

Effectue un saut de ligne. L'abscisse courante est ramenée à la valeur de la marge gauche et l'ordonnée augmente de la valeur indiquée en paramètre.

## Paramètres

h

L'amplitude du saut de ligne.

Par défaut, la valeur est égale à la hauteur de la dernière cellule imprimée.

## Voir

[Cell\(\)](#).



# MultiCell

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

## Description

Cette méthode permet d'imprimer du texte avec des retours à la ligne. Ceux-ci peuvent être automatiques (dès que le texte atteint le bord droit de la cellule) ou explicites (via le caractère \n). Autant de cellules que nécessaire sont imprimées, les unes en dessous des autres. Le texte peut être aligné, centré ou justifié. Le bloc de cellules peut être encadré et le fond coloré.

## Paramètres

w

Largeur des cellules. Si elle vaut 0, elles s'étendent jusqu'à la marge droite de la page (la marge droite étant égale à la marge gauche).

h

Hauteur des cellules.

txt

Chaîne à imprimer.

border

Indique si un cadre doit être tracé autour du bloc de cellules (0 : non, 1 : oui). Valeur par défaut : 0.

align

Contrôle l'alignement du texte. Les valeurs possibles sont :

- L : alignement à gauche
- C : centrage
- R : alignement à droite
- J : justification (valeur par défaut)

fill

Indique si le fond des cellules doit être coloré (1) ou transparent (0). Valeur par défaut : 0.

## Voir

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

# Open

Open( )

## Description

Cette méthode commence la génération du fichier PDF; elle doit être appelée avant toute méthode d'écriture. Aucune page n'est créée par cette méthode, aussi est-il nécessaire d'appeler AddPage().

## Voir

[Close\(\)](#).

# Output

```
Output([string file [, boolean download])
```

## Description

Sauve le document PDF dans un fichier local ou l'envoie au navigateur. Dans ce dernier cas, on peut utiliser le plug-in (s'il est présent) ou forcer la boîte de téléchargement de fichier.

La méthode commence par appeler `Close()` si nécessaire pour terminer le document.

## Paramètres

### file

Le nom du fichier. S'il est vide ou non renseigné, le document est envoyé au navigateur qui utilisera le plug-in s'il est installé.

### download

Dans le cas où `file` est renseigné, indique si le fichier doit être sauvé en local (`false`) où provoquer la boîte de téléchargement de fichier (`true`). Vaut `false` par défaut.

## Voir

[Close\(\)](#).

# PageNo

**int** PageNo( )

## Description

Renvoie le numéro de page courant.

# Rect

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

## Description

Dessine un rectangle à partir de son coin supérieur gauche. Il peut être tracé, rempli ou les deux à la fois.

## Paramètres

x

Abscisse du coin supérieur gauche.

y

Ordonnée du coin supérieur gauche.

w

Largeur.

h

Hauteur.

style

Manière de tracer le rectangle. Les valeurs possibles sont :

- D ou chaîne vide : contour (draw). C'est la valeur par défaut.
- F : remplissage (fill)
- DF ou FD : contour et remplissage

## Voir

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

# SetAuthor

SetAuthor(**string** author)

## Version

1.2

## Description

Définit l'auteur du document.

## Paramètres

author

Le nom de l'auteur.

## Voir

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

# SetAutoPageBreak

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

## Description

Active ou désactive le mode saut de page automatique. En cas d'activation, le second paramètre représente la distance par rapport au bas de la page qui déclenche le saut. Par défaut, le mode est activé et la marge vaut 2 cm.

## Paramètres

auto

Booléen indiquant si le mode doit être activé.

margin

Distance par rapport au bas de la page.

## Voir

[Cell\(\)](#).

# SetDisplayMode

SetDisplayMode(**string** mode [, **int** z])

## Version

1.2

## Description

Contrôle la manière dont le document sera affiché par Acrobat. On peut afficher la page en entier, occuper toute la largeur de la fenêtre, utiliser la taille réelle, choisir un facteur de zoom ou encore utiliser la valeur par défaut de l'utilisateur (celle paramétrée dans le menu Préférences).

Par défaut, les documents sont en mode pleine largeur.

## Paramètres

mode

Le mode à utiliser. Il peut prendre les valeurs suivantes :

- `fullpage` : affiche entièrement les pages à l'écran
- `fullwidth` : maximise la largeur des pages
- `real` : affiche la taille réelle (équivalent à un zoom de 100%)
- `zoom` : applique le facteur de zoom indiqué par `z`
- `default` : utilise le mode d'affichage par défaut d'Acrobat

z

Dans le cas où `mode` vaut `zoom`, indique le facteur à appliquer.



# SetCreator

SetCreator(**string** creator)

## Version

1.2

## Description

Définit le créateur du document. Il s'agit typiquement du nom de l'application qui génère le PDF.

## Paramètres

creator

Le nom du créateur.

## Voir

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

# SetDrawColor

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

## Version

1.3

## Description

Fixe la couleur pour toutes les opérations de tracé (lignes, rectangles et contours de cellules). Elle peut être indiquée en composantes RGB ou en niveau de gris. La méthode peut être appelée avant que la première page ne soit créée et la valeur est conservée de page en page.

## Paramètres

r

Si g et b sont renseignés, composante de rouge; sinon, indique le niveau de gris. Valeur comprise entre 0 et 255.

g

Composante de vert (entre 0 et 255).

b

Composante de bleu (entre 0 et 255).

## Voir

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

# SetFillColor

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

## Version

1.3

## Description

Fixe la couleur pour toutes les opérations de remplissage (rectangles pleins et fonds de cellules). Elle peut être indiquée en composantes RGB ou en niveau de gris. La méthode peut être appelée avant que la première page ne soit créée et la valeur est conservée de page en page.

## Paramètres

r

Si g et b sont renseignés, composante de rouge; sinon, indique le niveau de gris. Valeur comprise entre 0 et 255.

g

Composante de vert (entre 0 et 255).

b

Composante de bleu (entre 0 et 255).

## Voir

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

# SetFont

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

## Description

Fixe la police utilisée pour imprimer les chaînes de caractères. Il est obligatoire d'appeler cette fonction au moins une fois avant d'imprimer du texte, sinon le document résultant ne sera pas valide. La méthode peut être appelée avant que la première page ne soit créée et la valeur est conservée de page en page.

Si vous souhaitez simplement changer la taille courante, il est plus simple d'appeler SetFontSize().

**Note :** à partir de la version 1.2, les fichiers de métrique des polices doivent être accessibles. Il y a trois possibilités pour cela :

- Ils sont dans le répertoire courant (celui du script en cours d'exécution)
- Ils sont dans un des répertoires définis par le paramètre `include_path`
- Ils sont dans le répertoire défini par la constante `FPDF_FONTPATH`

Exemple pour ce dernier cas (notez le slash final) :

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

Si le fichier correspondant à la police demandée n'est pas trouvé, l'erreur "Could not include font metric file" est générée.

## Paramètres

### family

Famille de la police. Les valeurs possibles sont (indépendamment de la casse) :

- Courier (caractères de largeur fixe)
- Helvetica ou Arial (synonymes; sans serif)
- Times (avec serif)
- Symbol (symboles)
- ZapfDingbats (symboles)

Il est également possible de passer une chaîne vide, auquel cas la famille courante est conservée.

### style

Style de la police. Les valeurs possibles sont (indépendamment de la casse) :

- (chaîne vide) : normal
- B : gras
- I : italique

- BI ou IB : gras italique

La valeur par défaut est le style normal. Le style ne s'applique pas aux familles Symbol et ZapfDingbats.

### size

Taille de la police en points.

La valeur par défaut est la taille courante. Si aucune taille n'a encore été spécifiée depuis le début du document, la valeur prise est 12.

## Exemple

```
//Times normal 12
$pdf->SetFont('Times');
//Arial gras 14
$pdf->SetFont('Arial','B',14);
//Enlève le gras
$pdf->SetFont('');
//Times normal 14
$pdf->SetFont('Times');
```

## Voir

[SetFontSize\(\)](#), [Cell\(\)](#).

# SetFontSize

SetFontSize(**double** size)

## Description

Fixe la taille de la police courante.

## Paramètres

size

La taille (en points).

## Voir

[SetFont\(\)](#).

# SetKeywords

SetKeywords(**string** keywords)

## Version

1.2

## Description

Associe des mot-clés au document, généralement sous la forme 'mot-clé1 mot-clé2 ...'.

## Paramètres

keywords

La liste de mots-clés.

## Voir

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

# SetLineWidth

SetLineWidth(**double** width)

## Description

Fixe l'épaisseur des traits. La valeur est par défaut égale à 0,2 mm. La méthode peut être appelée avant que la première page ne soit créée et la valeur est conservée de page en page.

## Paramètres

width

L'épaisseur (en unité utilisateur).

## Voir

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



# SetMargins

SetMargins(**double** left, **double** top)

## Description

Fixe les marges gauche et haute (en unité utilisateur). Par défaut, elles valent 1 cm. Appelez cette méthode si vous désirez les changer.

## Paramètres

left  
Marge gauche.

top  
Marge haute.

# SetSubject

SetSubject(**string** subject)

## Version

1.2

## Description

Définit le sujet du document.

## Paramètres

subject

L'intitulé du sujet.

## Voir

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

# SetTextColor

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

## Version

1.3

## Description

Fixe la couleur pour le texte. Elle peut être indiquée en composantes RGB ou en niveau de gris. La méthode peut être appelée avant que la première page ne soit créée et la valeur est conservée de page en page.

## Paramètres

r

Si g et b sont renseignés, composante de rouge; sinon, indique le niveau de gris. Valeur comprise entre 0 et 255.

g

Composante de vert (entre 0 et 255).

b

Composante de bleu (entre 0 et 255).

## Voir

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

# SetTitle

SetTitle(**string** title)

## Version

1.2

## Description

Définit le titre du document.

## Paramètres

title

Le titre.

## Voir

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

# SetX

SetX(**double** x)

## Version

1.2

## Description

Fixe l'abscisse de la position courante. Si la valeur transmise est négative, elle est relative à l'extrémité droite de la page.

## Paramètres

x

La valeur de l'abscisse.

## Voir

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

# SetXY

SetXY(**double** x, **double** y)

## Version

1.2

## Description

Fixe l'abscisse et l'ordonnée de la position courante. Si les valeurs transmises sont négatives, elles sont relatives respectivement aux extrémités droite et basse de la page.

## Paramètres

x

La valeur de l'abscisse.

y

La valeur de l'ordonnée.

## Voir

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

# SetY

SetY(**double** y)

## Description

Ramène l'abscisse courante à la marge gauche et fixe l'ordonnée. Si la valeur transmise est négative, elle est relative au bas de la page.

## Paramètres

y  
La valeur de l'ordonnée.

## Voir

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

# Text

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

## Description

Imprime une chaîne de caractères. L'origine est à gauche du premier caractère, sur la ligne de base. Cette méthode permet de positionner précisément une chaîne dans la page, mais il est plus simple d'utiliser `Cell()` qui est la méthode standard pour imprimer du texte.

## Paramètres

x

Abscisse de l'origine.

y

Ordonnée de l'origine.

txt

Chaîne à imprimer.

## Voir

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