**Manual Provisório FPDF  
  
Editado por Simone Frigerio e Lorenzo Abbati  
  
Traduzido por Alexandre Silva**

**AddLink**   
  
**Descrição**  
Cria um novo link interno e devolve seu identificador. Um link interno é uma área clicável que redireciona para outra posição dentro do documento. O identificador pode ser passado a Cell(), Write(), Image() ou Link(). O destino é definido com SetLink().  
  
  
**AddPage**  
AddPage([string orientation])   
  
**Descrição**  
Adiciona uma página ao documento. Se uma página já está presente antes dela é chamado o método Footer() para inserir o pé da pagina. Então a página é somada, a posição atual é colocada no ângulo superior esquerdo, enquanto se lembrando da margem superior e esquerda, depois vem chamando Header() para visualizar o título.  
A fonte colocada antes da chamada é restabelecida automaticamente. Não há nenhuma necessidade de chamar SetFont() novamente se quiser continuar com a mesma fonte. O mesmo acontece para cores e largura de linha.  
A origem do sistema de coordenadas está no ângulo superior esquerdo e o aumento do ordenadas que faz mover para baixo.  
  
**Parâmetros**  
orientation   
Orientação da página. Possíveis valores são (case insensitive):   
• P: Retrato   
• L: Paisagem   
O valor padrão é repassado pelo construtor.  
  
**AliasNbPages**  
AliasNbPages([string alias])   
  
**Descrição**  
Define um alias para o número total de páginas. Será substituído no fim do documento.  
  
**Parâmetros**  
alias   
Aliás. Valor padrão: {nb}.   
Exemplo  
function Footer()  
  
//Vai para 1,5 cm do fundo da página  
this.SetY(-15);  
//Seleciona Arial cursiva 8  
this.SetFont('Arial','I',8);  
//Imprime o número da página atual e total  
this.Cell(0,10,'Page 'pdf.PageNo().'/{nb}',0,0,'C');  
  
end function  
  
  
pdf=new PDF()  
pdf.AliasNbPages()  
  
  
**Cell**  
Cell(float w [, float h [, string txt [, mixed border [, int ln [, string align [, int fill [, mixed link]]]]]]])   
  
**Descrição**  
Imprime uma célula (área retangular) com bordas, cor de fundo e string de caractere opcional. O ângulo superior esquerdo da célula corresponde à posição atual. O texto pode ser alinhado ou centralizado. Depois da chamada, a posição atual o move à direita ou na linha seguinte. É possível inserir um link no texto.   
Se a interrupção de página automática estiver habilitada e a célula for além dos limites, a interrupção de página é feita antes da saída.  
  
**Parâmetros**  
w   
Largura da célula. Se 0, a célula o estende até a margem direita.   
  
h   
Altura da célula. Valor padrão: 0.   
  
txt   
String para imprimir. Valor padrão: string vazia.   
  
border   
Indica se precisa localizar as bordas ao redor da célula. O valor pode ser ambos um número:  
• 0: qualquer borda   
• 1: moldura   
ou uma string contém alguns ou todos os caracteres seguintes (em qualquer ordem):   
• L: esquerdo   
• T: superior  
• R: direito   
• B: inferior  
Valor padrão: 0.   
  
ln   
Indica a posição atual depois da chamada. Possíveis valores são:  
• 0: à direita   
• 1: no início da linha seguinte   
• 2: em baixo   
Inserir 1 equivale a inserir 0 e chamar Ln() imediatamente depois. Valor padrão: 0.   
  
align   
Permite centralizar ou se alinhar o texto. Possíveis valores são:  
• L o string vazia: alinhamento à esquerda (valor padrão)   
• C: centralizado  
• R: alinhamento à direita  
  
fill   
Indica se o fundo da célula dever ser desenhado(1) ou dever ser transparente(0).  
Valor padrão: 0.   
  
link   
O url ou o identificador devolvidos por AddLink ().  
  
**Exemplo**  
//Configura a fonte  
pdf.SetFont('Arial','B',16);  
//Move a 8 cm da direita  
pdf.Cell(80);  
//Texto centralizado em uma célula de 20\*10 mm e interrupção da linha  
pdf.Cell(20,10,'Title',1,1,'C');   
  
  
  
**Close**  
Close()   
  
**Descrição**  
Fecha o documento PDF. Não é necessário chamar este método explicitamente, porque Output() o faz automaticamente. Se não existir páginas no documento, o método AddPage() é chamado para evitar que se obtenha documento inválido.  
  
  
  
**CreatePDF**  
CreatePDF([string orientation [, string unit [, mixed format]]])   
  
**Descrição**  
Trata-se do construtor da classe. Permite colocar o formato da página, a orientação e a unidade de medida usada em todos os métodos (exceto para a dimensão das fontes).  
  
**Parâmetros**  
orientation   
Orientação da pagina padrão. Possíveis valores são (case insensitive):   
• P : Retrato   
• L : Paisagem   
O valor padrão é P.   
  
unit   
Unidade de medida. Possíveis valores são:   
• pt: pontos   
• mm: milímetros  
• cm: centímetros   
• in: polegadas   
Um ponto é equivalente a 1/72 da polegada, ou seja, cerca de 0,35 mm (uma polegada corresponde a 2.54 cm). Trata-se de uma da medida muito comum na tipografia; a dimensão da fonte é expressa nesta unidade.  
  
O valor padrão é mm.   
  
format   
O formato utilizado pela página. Pode ser um dos seguinte valores (case insensitive):   
• A3   
• A4   
• A5   
• Letter   
• Legal   
Ou um formato personalizado na forma de array bidimensional contendo a largura e a altura (expressa na unidade de medida dada na unidade).  
  
**Error**  
Error(string msg)   
  
**Descrição**  
Este método é chamado automaticamente no caso de erro fatal; retorna simplesmente a mensagem e pára a execução. Uma classe derivada pode sobrescrever para customizar a administração dos erros, mas deve sempre parar o script, ou o documento resultante provavelmente poderá ser inválido.  
  
**Parâmetros**  
msg   
A mensagem de erro.  
  
**ExtendsCode**  
  
**Footer**  
  
**GetMargin**  
  
  
  
**GetStringWidth**  
float GetStringWidth(string s)   
  
**Descrição**  
Devolve o comprimento de uma string na unidade de medida utilizada. Deve ser selecionada uma fonte.  
  
**Parâmetros**  
s   
A string de qual tem que calcular o comprimento.  
  
  
  
**GetX**  
float GetX()   
  
**Descrição**  
Devolve a abscissa da posição atual.  
  
  
  
**GetY**  
float GetY()   
  
**Descrição**  
Devolve a ordenada da posição atual.   
  
  
  
**Header**  
Header()   
  
**Descrição**  
Este método é usado para designar o título da página. É chamado automaticamente de AddPage() e não deveria ser chamado diretamente da aplicação. A implementação em FPDF é deixado vazio, então necessita de uma subclasse e sobrescrever o método se uma tentativa particular é desejada.  
  
**Exemplo**  
class PDF extends FPDF  
  
function Header()  
  
//Seleciona Arial negrito 15  
this.SetFont('Arial','B',15);  
//Move-se para a direita  
this.Cell(80);  
//Título sem painel  
this.Cell(30,10,'Title',1,0,'C');  
//Interrupção de linha  
this.Ln(20);  
end function  
  
  
  
**Image**  
Image(string file, float x, float y, float w [, float h [, string type [, mixed link]]])   
  
**Descrição**  
Imprime uma imagem na página. Necessita especificar o ângulo à esquerda superior e pelo menos uma das dimensões. A altura ou a largura podem ser calculadas automaticamente para manter as proporções da imagem.  
  
Os formatos suportados são JPEG e PNG.  
  
Para JPEG são suportados todos os tipos:   
• Escala de cinza  
• True colors (24 bits)   
• CMYK (32 bits)   
Se uma cor transparente for definida, será se lembrado de lá (mas só será interpretado do Acrobat 4 em diante). O formato pode ser especificado explicitamente ou pode ser chamado pela extensão do arquivo. É possível inserir um link na imagem.  
  
Nota: se uma mesma imagem vem sendo usada diversas vezes, no arquivo só será memorizado uma cópia.  
  
**Parâmetros**  
file   
Nome do arquivo contendo a imagem.  
x   
Abscissa do ângulo no alto à esquerda.  
y   
Ordenada do ângulo no alto à esquerda.  
w   
Largura da imagem na página. Se igual à zero, será calculado automaticamente para manter as proporções originais.  
h   
Altura da imagem na página. Se não especificar ou igual à zero, será calculado automaticamente para manter as proporções originais.  
type   
Formato da imagem. Possíveis valores são (case insensitive): JPG, JPEG, PNG. Se não especificou, o tipo será puxado pela extensão do arquivo.  
link   
O url ou o identificador devolvidos de AddLink().   
  
  
**Line**  
Line(float x1, float y1, float x2, float y2)   
  
**Descrição**  
Localiza uma linha entre dois pontos.  
  
**Parâmetros**  
x1   
Abscissa do primeiro ponto.  
y1   
Ordenada do primeiro ponto.   
x2   
Abscissa do segundo ponto.   
y2   
Ordenada do segundo ponto.   
  
  
  
**Link**  
Link(float x, float y, float w, float h, mixed link)   
  
**Descrição**  
Insere um link em uma área retangular da página. Os links em imagens ou texto geralmente vem inserido por Cell(), Write() o Image(), mas este método pode ser útil no caso de precisar que seja definida uma área clicável dentro de uma imagem.  
  
**Parâmetros**  
x   
Abscissa do ângulo no alto à esquerda do retângulo.  
y   
Ordenada do ângulo no alto à esquerda do retângulo.  
w   
Largura do retângulo.  
h   
Altura do retângulo.  
link   
O url ou o identificador devolvidos por AddLink().   
  
  
  
**Ln**  
Ln([float h])   
  
**Descrição**  
Efetua uma quebra de linha. A abscissa atual retorna à margem esquerda e a ordenada aumenta de valor passado por parâmetro.  
  
**Parâmetros**  
h   
A altura da quebra. Por valor padrão equivale à altura da última célula impressa.  
  
  
  
**LoadExtension**  
  
  
  
**LoadModels**  
  
  
  
**MultiCell**  
MultiCell(float w, float h, string txt [, mixed border [, string align [, int fill]]])   
  
**Descrição**  
Este método permite imprimir texto com quebra de linha. Pode ser automático (assim que o texto alcançar a borda direita da célula) ou explícito (através do caracteres \n). São inseridas tantas células quanto forem necessário, uma sob a outra. O texto pode ser alinhado, centralizado ou justificado. É possível inserir bordas e fundo ao bloco de células.  
  
**Parâmetros**  
w   
Largura das células. Se 0, o estende até a borda direita da página.  
  
h   
Altura das células.  
  
txt   
String da impresão.  
  
border   
Indica se precisa desenhar as bordas ao redor do bloco das células. O valor pode ser ambos um número  
• 0: qualquer borda   
• 1: moldura   
Ou uma string contendo alguns ou todos os caracteres seguintes (em qualquer ordem):  
• L: esquerda   
• T: superior   
• R: direita   
• B: inferior   
Valor padrão: 0.   
  
align   
Impõe o alinhamento do texto. Possíveis valores são:  
• L: alinha à esquerda  
• C: centraliza   
• R: alinha à direita   
• J: justificado (valore padrão)   
fill   
Indica se o fundo da célula deve ser exibido (1) ou deve ser transparente (0).  
Valor padrão: 0.   
  
  
  
**Open**  
Open()   
  
**Descrição**  
Este método começa a geração do documento PDF; deve ser chamado antes de qualquer comando de saída. Este método não cria nenhuma página, conseqüentemente é necessário chamar AddPage().  
  
  
  
**Output**  
Output([string file [, boolean download]])   
  
**Descrição**  
Salva o documento PDF em um arquivo local ou o envia ao browser. No segundo caso, é possível usar o plug-in (se existir) ou forçar um download (caixa de diálogo “Salvar como”). O método chama Close() se necessário fechar o documento.  
  
**Parâmetros**  
file   
O nome do arquivo. Se nulo ou não colocado, o documento será enviado ao browser que então usará o plug-in (se presente).  
download   
Se o arquivo for especificado, indica se deve ser salvo no local (falso) ou abrir uma caixa de diálogo "Salvar como" no browser. Valor padrão: false.  
  
  
  
**PageNo**  
int PageNo()   
  
**Descrição**  
Devolve o número da página atual.  
  
  
  
**Rect**  
Rect(float x, float y, float w, float h [, string style])   
  
**Descrição**  
Desenha um retângulo. Pode ser desenhado (só bordas), cheio (sem bordas) ou ambos.  
  
**Parâmetros**  
x   
Abcissa do ângulo no alto à esquerda.   
  
y   
Ordenada do ângulo no alto à esquerda.   
  
w   
Largura.   
  
h   
Altura.   
  
style   
Estilo de esboço. Possíveis valores são:   
• D o uma string vazia: desenha. É o valor padrão.   
• F: preenche   
• DF o FD: desenha e preenche  
  
  
  
**SetAuthor**  
SetAuthor(string author)   
  
**Descrição**  
Define o autor do documento  
  
**Parâmetros**  
author   
Nome do autor.  
  
  
  
**SetAutoPageBreak**  
SetAutoPageBreak(boolean auto [, float margin])   
  
**Descrição**  
Habilita ou desabilita a forma de interrupção de página automática. Se habilitado, o segundo parâmetro representa a distância do fundo da página que define o limite de ativação. Por padrão, a modalidade é on e a borda é 2 cm.  
  
**Parâmetros**  
auto   
Booleano indica se a modalidade deve ser on ou off.  
margin   
Distancia do fundo da página.   
  
  
  
**SetCreator**  
SetCreator(string creator)   
  
**Descrição**  
Define o criador do documento. Tipicamente é o nome da aplicação que gerou o PDF.  
  
**Parâmetros**  
creator   
O nome do criador.  
  
  
  
**SetDisplayMode**  
SetDisplayMode(mixed zoom [, string layout])   
  
**Descrição**  
Define a forma na qual o documento será mostrado no visualizador. É possível ajustar o nível do zoom: as páginas podem ser mostradas completamente na tela, para toda a página, usar as reais dimensões, representado em escala com um fator do zoom específico ou usar os valores padrões do visualizador (como configurado no menu Preferências do Acrobat). É possível também especificar o layout de página: uma por vez, visualização contínua, duas colunas ou padrão do visualizador.  
Por padrão, o documento utiliza a modalidade em todas as páginas e visualização contínua.  
  
**Parâmetros**  
zoom   
O zoom a ser usado. Pode ser um dos seguintes valores de string:   
• fullpage: mostra a tela inteira na pagina   
• fullwidth: para toda a pagina   
• real: usa as reais dimensões (equivalente a zoom=100%)  
• default: utiliza a modalidade padrão do visualizador   
ou um número indica o nível do zoom ser usado.   
  
**layout**   
O layout da página. Possíveis valores são:  
• single: mostra uma página por vez  
• continuous: mostra a página em modo contínuo   
• two: mostre duas páginas em duas colunas  
• default: utiliza a modalidade padrão do visualizador   
O valor padrão é continuous.  
  
  
  
**SetDrawColor**  
SetDrawColor(int r [, int g, int b])   
  
**Descrição**  
Define a cor usada para todas as operações de esboço (linhas, retângulos e bordas da célula). Pode ser expresso em componente RGB ou escala de cinza. O método pode ser chamado antes da primeira página ser criada para manter o valor de página em página.  
  
**Parâmetros**  
r   
Se g e b são dados, indica o componente vermelho; caso contrário indica o nível de cinza. Valores entre 0 e 255.  
g   
Componente verde (entre 0 e 255).   
b   
Componente azul (entre 0 e 255).   
  
  
  
**SetFillColor**  
SetFillColor(int r [, int g, int b])   
  
**Descrição**  
Define a cor usada para todos os operações de preenchimento (retângulos cheios e fundo das células). Pode ser expresso em componente RGB ou escala de cinza. O método pode ser chamado antes da primeira página ser criada para manter o valor de página em página.  
  
**Parâmetros**  
r   
Se g e b são dados, indica o componente vermelho; caso contrário indica o nível de cinza. Valores entre 0 e 255.   
g   
Componente verde (entre 0 e 255).   
b   
Componente azul (entre 0 e 255).   
  
  
  
**SetFont**  
SetFont(string family [, string style [, float size]])   
  
**Descrição**  
Indica a fonte usada para imprimir string de caracteres. E' obrigatório chamar este método ao menos uma vez antes de imprimir algum texto ou o documento resultante podendo não ser válido.   
A fonte pode ser ambos um padrão ou um assistente médio do método AddFont (). As fontes Standard usam a codificação Windows cp1252 (Western Europe).  
O método deve ser chamado antes da criação da primeira página para manter a fonte de página em página.  
Se for desejado que mude a dimensão da fonte, é mais simples chamar SetFontSize ().  
  
Nota: para o fontes standard, o arquivo com o métrica das fontes deve ser acessível. Para este propósito há três possibilidades:  
•São encontrados no diretório atual (no qual reside o script em execução)  
•São encontrados em um dos diretórios definido pelo parâmetro include\_path  
•São encontrados no diretório definido pela constante FPDF\_FONTPATH.  
Exemplo para o último caso (notar a barra de fechamento):   
define('FPDF\_FONTPATH','/home/www/font/');  
require('fpdf.asp');   
  
Se não for encontrado o arquivo correspondente ao que foi requisitado, será gerado um erro “Could not include font metric file”.  
  
**Parâmetros**  
family   
Família da fonte. Pode ser um nome definido por AddFont() ou um das famílias standards (case insensitive):   
• Courier (fixed-width)   
• Helvetica o Arial (sinônimo; sans serif)   
• Times (serif)   
• Symbol (symbolic)   
• ZapfDingbats (symbolic)   
E' também possível passar uma string vazia. Neste caso, é preservada a família atual.  
  
style   
Estilo da fonte. Valores possíveis são (case insensitive):   
• String vazia: normal   
• B: negrito   
• I: itálico   
• U: sublinhado   
Ou qualquer combinação. O valor padrão é normal. Não é possível aplicar os estilos negrito e itálico a Symbol e ZapfDingbats.   
  
size   
Tamanho da fonte em pontos.  
O valor padrão é do tamanho atual. Se nenhuma dimensão não for especificada no início do documento, o valor será 12.  
  
**Exemplo**  
//Times normal 12  
pdf.SetFont('Times')  
//Arial negrito 14  
pdf.SetFont('Arial','B',14)  
//Remove o negrito  
pdf.SetFont('')  
//Times negrito, itálico e sublinhado 14  
pdf.SetFont('Times','BIU')  
  
  
  
**SetFontSize**  
SetFontSize(float size)   
  
**Descrição**  
Define o tamanho da fonte atual.  
  
**Parâmetros**  
size   
O tamanho (em pontos).   
  
  
  
**SetKeywords**  
SetKeywords(string keywords)   
  
**Descrição**  
Associa palavras chaves ao documento, geralmente na forma 'keyword2 de keyword1...'.  
  
**Parâmetros**  
keywords   
A lista do keywords.   
  
  
  
**SetLeftMargin**  
SetLeftMargin(float margin)   
  
**Descrição**  
Imposta il margine sinistro. Il metodo può essere chiamato prima della creazione della prima pagina.   
Se l'ascissa corrente va fuori del margine, viene riportata nel margine.   
Impõe a borda esquerda. O método deve ser chamado antes da criação da primeira página. Se a abscissa atual sai da margem, é trazida de bolta nesta margem.  
  
**Parâmetros**  
margin   
A margem.   
  
  
  
**SetLineStyle**  
  
  
  
**SetLineWidth**  
SetLineWidth(float width)   
  
**Descrição**  
Define a espessura da linha. Por padrão, o valor é de 0.2 mm. O método pode ser chamado antes da criação da primeira página do documento para manter o valor em todas as páginas.  
  
**Parâmetros**  
width   
A espessura.   
  
  
  
[b]SetLink[/b]  
SetLink(int link [, float y [, int page]])   
  
**Descrição**  
Define a página e a posição a qual um link aponta.  
  
**Parâmetros**  
link   
O identificador do link retornado de AddLink().   
y   
Ordenada da posição do alvo; -1 indica a posição atual. O valor padrão é 0 (inicio da pagina).   
page   
Numero da página do alvo; -1 indica a pagina atual. É o valor padrão.   
  
  
  
**SetMargins**  
SetMargins(float left, float top [, float right])   
  
**Descrição**  
Define as margens esquerda, superior e direito. Por padrão são 1 cm. Chamar este método para mudar.  
  
**Parâmetros**  
left   
Margem esquerda.   
top   
Margem superior.  
right   
Margem direita. O valor padrão é igual ao valor da esquerda.   
  
  
  
**SetPath**  
SetPath([text path])   
  
**Descrição**  
Define o caminho do diretório FPDF, o subdiretório, extensões, includes, fontes e modelos.  
  
**Parâmetros**  
path   
Caminho relativo ou virtual do diretório FPDF. Exemplo: pdf.SetPath("../fpdf/").  
  
  
  
**SetRightMargin**  
SetRightMargin(float margin)   
  
**Descrição**  
Define a margem direita. O método pode ser chamado antes da criação da primeira página.  
  
**Parâmetros**  
margin   
A margem.   
  
  
  
**SetSubject**  
SetSubject(string subject)   
  
**Descrição**  
Define o assunto do documento.  
  
**Parâmetros**  
subject   
O assunto.   
  
  
  
**SetTextColor**  
SetTextColor(int r [, int g, int b])   
  
**Descrição**  
Define a cor usado para o texto. Pode ser expresso em componente RGB ou em escala de cinza. O método pode ser chamado antes da criação da primeira página para manter o valor de página em página.  
  
**Parâmetros**  
r   
Se g e b são dados, indica o componente vermelho; caso contrário indica o nível de cinza. Valores entre 0 e 255.  
g   
Componente verde (entre 0 e 255).   
b   
Componente azul (entre 0 e 255).   
  
  
  
**SetTitle**  
SetTitle(string title)   
  
**Descrição**  
Define o título do documento.  
  
**Parâmetros**  
title   
o título.   
  
  
  
**SetTopMargin**  
SetTopMargin(float margin)   
  
**Descrição**  
Define a margem superior. O método pode ser chamado antes da criação da primeira página.  
  
**Parâmetros**  
margin   
Il margine.   
  
  
  
**SetX**  
SetX(float x)   
  
**Descrição**  
Define a abscissa da posição atual. Se o valor passado for negativo, está à direita em relação a esta página.  
  
**Parâmetros**  
x   
O valor da abscissa.   
  
  
  
**SetXY**  
SetXY(float x, float y)   
  
**Descrição**  
Define a abscissa e a ordenada da posição atual. Se os valores passados forem negativos, eles são relativos respectivamente à direita e ao fundo da página.  
  
**Parâmetros**  
x   
O valor da abcissa.   
y   
O valor da ordenada.   
  
  
  
**SetY**  
SetY(float y)   
  
**Descrição**  
Move a abscissa atual de volta para a margem esquerda e impõe a ordenada. Se o valor passado for negativo, é relativo ao fundo da página.  
  
**Parâmetros**  
y   
O valor da ordenada.   
  
  
  
**Text**  
Text(float x, float y, string txt)   
  
**Descrição**  
Imprime um caracteres da string. O início é da esquerda do primeiro caracteres. O verso está na esquerda do primeiro caractere, na linha base. Este método permite posicionar de modo preciso uma string na página, mas é mais comum usar Cell(), MultiCell() or Write() que são os métodos padrões para imprimir algum texto.  
  
**Parâmetros**  
x   
Abcissa do documento.   
y   
Ordenada do documento.   
txt   
string da impressão  
  
  
  
**Write**  
Write(float h, string txt [, mixed link])   
  
**Descrição**  
Este método imprime o texto na posição atual. Quando a borda direita é alcançada (ou o caractere \n for chamado) automaticamente que vai encabeçar e o texto continua da borda esquerda. Na saída de método, a posição atual é deixada à extremidade do texto. É possível pôr uma conexão no texto.  
  
**Parâmetros**  
h   
altura da linha  
txt   
string para imprimir.   
link   
URL ou identificador retornado por AddLink().   
  
**Exemplo**  
pdf.write(5,'www.aspxnet.it', 'http://www.aspxnet.it');