Class

PDF::Wrapper

In: pdfwrapper.rb
Parent: Object

Create PDF files by using the cairo and pango libraries.

Rendering to a file:

```
require 'pdfwrapper'
pdf = PDF::Wrapper.new(:paper => :A4)
pdf.text "Hello World"
pdf.render_to_file("wrapper.pdf")
```

Rendering to a string:

```
require 'pdfwrapper'
pdf = PDF::Wrapper.new(:paper => :A4)
pdf.text "Hello World", :size => 16
puts pdf.render
```

Changing the default font:

```
require 'pdfwrapper'
pdf = PDF::Wrapper.new(:paper => :A4)
pdf.default_font("Monospace")
pdf.text "A Heading", :font => "Sans Serif"
pdf.text "Pretend this is a code sample"
puts pdf.render
```

Methods

absolute_margin_bottom   absolute_margin_left
absolute_margin_right   absolute_margin_top
absolute_x_middle   absolute_y_middle   body_height
body_width   circle   current_point   default_color   default_color=
default_font   default_font=   default_font_size
default_font_size=   font_size   image   line   margin_x_middle
margin_y_middle   move_to   new   rectangle   render
render_to_file   rounded_rectangle   select_font   start_new_page
stroke_color   text

Constants

PAGE_SIZES	= { # :value {...}: #:4A0 => [4767.87, 6740.79], :2A0 => [3370.39, 4767.87], :A0 => [2383.94, 3370.39], :A1 => [1683.78, 2383.94], :A2 => [1190.55, 1683.78], :A3 => [841.89, 1190.55], :A4 => [595.28, 841.89], :A5 => [419.53, 595.28], :A6 => [297.64, 419.53], :A7 => [209.76, 297.64], :A8 => [147.40, 209.76], :A9 => [104.88, 147.40], :A10 => [73.70, 104.88], :B0 => [2834.65, 4008.19], :B1 => [2004.09, 2834.65], :B2 => [1417.32, 2004.09], :B3 => [1000.63, 1417.32], :B4 => [708.66, 1000.63], :B5 => [498.90, 708.66], :B6 => [354.33, 498.90], :B7 => [249.45, 354.33], :B8 => [175.75, 249.45], :B9 => [124.72, 175.75], :B10 => [87.87, 124.72], :C0 => [2599.37, 3676.54], :C1 => [1836.85, 2599.37], :C2 => [1298.27, 1836.85], :C3 => [918.43, 1298.27], :C4 => [649.13, 918.43], :C5 => [459.21, 649.13], :C6 => [323.15, 459.21], :C7 => [229.61, 323.15], :C8 => [161.57, 229.61], :C9 => [113.39, 161.57], :C10 => [79.37, 113.39], :RA0 => [2437.80, 3458.27], :RA1 => [1729.13, 2437.80], :RA2 => [1218.90, 1729.13], :RA3 => [864.57, 1218.90], :RA4 => [609.45, 864.57], :SRA0 => [2551.18, 3628.35], :SRA1 => [1814.17, 2551.18], :SRA2 => [1275.59, 1814.17], :SRA3 => [907.09, 1275.59], :SRA4 => [637.80, 907.09], :LETTER => [612.00, 792.00], :LEGAL => [612.00, 1008.00], :FOLIO => [612.00, 936.00], :EXECUTIVE => [521.86, 756.00]	borrowed from PDF::Writer

Attributes

margin_bottom	[R]
margin_left	[R]
margin_right	[R]
margin_top	[R]
page_height	[R]
page_width	[R]

Public Class methods

new(*opts={}*)

create a new PDF::Wrapper class to compose a PDF document
Options:

`:paper`:	The paper size to use (default :A4)

`:orientation`: :portrait (default) or :landscape
`:background_colour`:The background colour to use (default :white)

Public Instance methods

absolute_margin_bottom*()*

Returns the y value of the bottom margin The top left corner of the page is (0,0)

absolute_margin_left*()*

Returns the x value of the left margin The top left corner of the page is (0,0)

absolute_margin_right*()*

Returns the x value of the right margin The top left corner of the page is (0,0)

absolute_margin_top*()*

Returns the y value of the top margin The top left corner of the page is (0,0)

absolute_x_middle*()*

Returns the x at the middle of the page

absolute_y_middle*()*

Returns the y at the middle of the page

body_height*()*

Returns the height of the useable part of the page (between the top and bottom margins)

body_width*()*

Returns the width of the useable part of the page (between the side margins)

circle_(x, y, r, opts = {})_

draw a circle with radius r and a centre point at (x,y). Parameters:

`:x`:The x co-ordinate of the circle centre.
`:y`:The y co-ordinate of the circle centre.
`:r`:The radius of the circle

Options:

`:color`:	The colour of the circle outline
`:fill_color`:	The colour to fill the circle with. Defaults to nil (no fill)

current_point_()_

return the current position of the cursor returns 2 values - x,y

default_color_(c)_

change the default colour used to draw on the canvas

default_color=_(c)_

Alias for default_color

default_font_(fontname, style = nil, weight = nil)_

change the default font to write with

default_font=_(fontname, style = nil, weight = nil)_

Alias for default_font

default_font_size_(size)_

change the default font size

default_font_size=_(size)_

Alias for default_font_size

font_size(*size*)

Alias for default_font_size

image(*filename, opts = {}*)

add an image to the page at this stage the file must be a PNG or SVG, and no options are supported

line(*x0, y0, x1, y1, opts = {}*)

draw a line from x1,y1 to x2,y2

Options:

:`color`:The colour of the rectangle outline

margin_x_middle(*)*

Returns the x coordinate of the middle part of the useable space between the margins

margin_y_middle(*)*

Returns the y coordinate of the middle part of the useable space between the margins

move_to(*x,y*)

move the cursor to an arbitary position on the current page

rectangle(*x, y, w, h, opts = {}*)

draw a rectangle starting at x,y with w,h dimensions. Parameters:

:`x`:The x co-ordinate of the top left of the rectangle.
:`y`:The y co-ordinate of the top left of the rectangle.
:`w`:The width of the rectangle
:`h`:The height of the rectangle

Options:

`:color:`	The colour of the rectangle outline
`:fill_color:`	The colour to fill the rectangle with. Defaults to nil (no fill)

render*()*

render the PDF and return it as a string

render_to_file*(filename)*

save the rendered PDF to a file

rounded_rectangle*(x, y, w, h, r, opts = {})*

draw a rounded rectangle starting at x,y with w,h dimensions.
Parameters:

`:x`:The x co-ordinate of the top left of the rectangle.
`:y`:The y co-ordinate of the top left of the rectangle.
`:w`:The width of the rectangle
`:h`:The height of the rectangle
`:r`:The size of the rounded corners

Options:

`:color:`	The colour of the rectangle outline
`:fill_color:`	The colour to fill the rectangle with. Defaults to nil (no fill)

select_font*(fontname, style = nil, weight = nil)*

Alias for default_font

start_new_page*()*

move to the next page

stroke_color*(c)*

Alias for default_color

text_(str, opts={})_

write text to the page By default the text will be rendered using all the space within the margins and using the default font sylings set by default_font(), default_font_size, etc To override all these defaults, use the options hash

Options:

`:left`:	The x co-ordinate of the left-hand side of the text.
`:top`:	The y co-ordinate of the top of the text.
`:width`:	The width of the text to wrap at
`:font`:	The font family to use as a string
`:font_size`:	The size of the font in points
`:alignment`:	Align the text along the left, right or centre. Use :left, :right, :center
`:justify`:	Justify the text so it exapnds to fill the entire width of each line. Note that this only works in pango >= 1.17
`:spacing`:	Line spacing

[Validate]