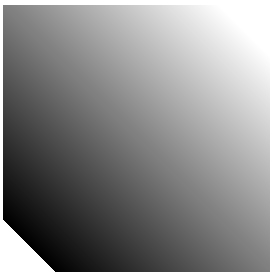
Shading

pslib supports a PostScript level 3 feature called shading. A shading starts at a certain point on the page with a given color and ends at a second point with different color. The type of shading can be ’axial’ or ’radial’. Shadings are used for many purposes like shadows, three dimensional appearance or simply to make a nice background for a diagramm.

Shading 1

The bar’s gradient fill to the right of this page starts at (500, 0) and ends at (590, 0). It extends over the whole page in y direction which is normal. I you want to restrict the gradient fill to a given area, one will have to apply PS_clip() on a path before calling PS_shfill() (see Shading 1).


Shading 2

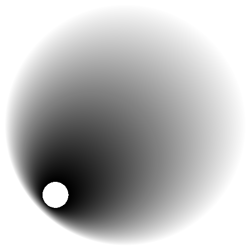
A grandient fill can easily be drawn in an angle if the start and end position have different x and y coordinates like in Shading 2.

Shading 3

As you can see at the bar, the shading extends orthogonal to the drawing direction but does not extend in the drawing direction. This behaviour can also be altered by the two parameters ’extend0’ and ’extend1’. Both are passed as part of the option list. Setting them to ’true’ will extend the gradient fill towards the drawing direction with the color where it left off or starts (see Shading 3).

Shading 4

The second class of gradient fills are of type ’radial’. They start at one circle and grow/shrink towards a second circle. The coordinates mentioned above now specify the middle points of the circles. The radi of both circles are passed again within the option list, named ’r0’ and ’r1’ (see Shading 4).


Shading 5

The previous example starts with a circle whose radius is 0. Making this a value greater 0 will punch a whole into the gradient fill.

This file has been created with pslib 0.2.3

The world of color shading

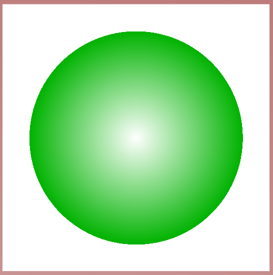
I suppose it does not really supprise you that gradient fills can be colorful. The first example is the background of this page which is a radial gradient fill from white to red (RGB [0,0,0.7]). Its outer circle has a very large radius of 795 pixels. The inner circle’s radius is just 40 pixels. If there was something behind the gradient fill it would shine through the inner circle, because the gradient fill does not extend into that direction. There is a continuation of the red color beyond the outer circle, but that does not make a difference in this case, because its outside of the page and therefore not visible.

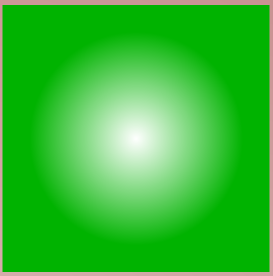
Shading 1 and Shading 2 illustrate the difference of the extend1 parameter being set to false in Shading 1 or true in Shading 2. Using extend is always a bit dangerous because it easily fills up the whole page, unless you clip the drawing area, which was done in all the examples on this page.

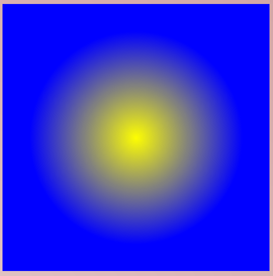
Using white as the start or end color is quite common but not nessecary. Shading 3 shows that any other color can be used as well.

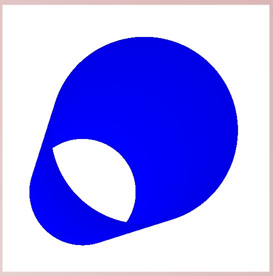
The tube in Shading 4 is a bit of a misuse of shadings. It used alsmost identical start and end colors for the shading without any extend parameter set. This only works if the two colors are not identical. If they were identical you would see just the start and circle overlapping, because the gradient function has no domain it can run over. Well, this is anyway not the way to draw tubes.

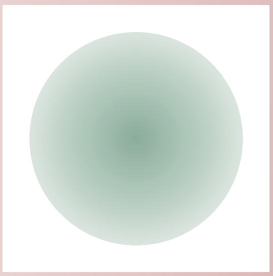
You wonder if this all works with CMYK colors? Yes, of course it does! And what about spot colors? No problem either, with one little restriction. The start and end spot color must be same one, but with different tint value. That means that the gradient just changes the tint value but not the color itself. I am not sure if this is an unbearable restriction or not. Shading 5 shows an example of Pantone 5565 C starting at a tint value of 0.2 (the outer circle) and ending with 0.8 (the middle point).


Shading 1


Shading 2

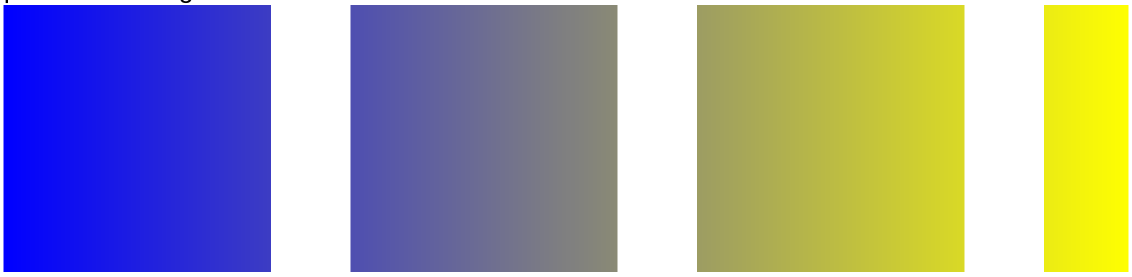

Shading 3


Shading 4

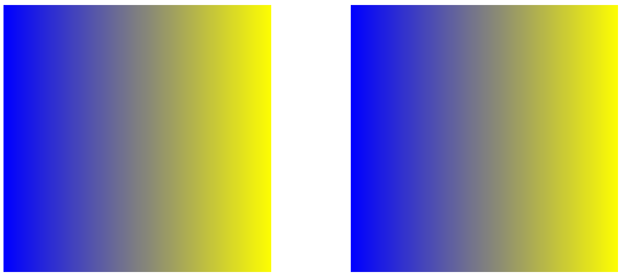

Shading 5

Shading pattern

The examples so far were using the PS_shading() and PS_shfill() in combination wiht PS_clip(). They are more than sufficient in most cases, but pslib has a second approach to create areas with a gradient fill. Beside regular patterns — which are used like regular colors — used for filling an area with a drawing, one can use shading pattern in the same way. The results are similar, they way of doing it is different. Each filled rectangle below this text uses the same shading pattern for filling.



The line of rectangles illustrates one important aspect of shading. The shading is defined relativ to the current coordinate system. In this case it starts at x-position 50 and ends at x-position 470 in an unmodified coordinate system. The right most rectangle is not filled completly because of the shadings end. Filling areas is like punching wholes in a white coat on top of the page and peeking through. This is mostly not what you want. If you would like to fill the rectangle with the full range of the shading, you will have to create a shading starting at 0 and ending at the width of the rectangle. Before filling the rectangle you will have to translate the coordinate system to the lower left corner of the rectangle, create the pattern in the modified coordinate system and draw a filled rectangle at (0, 0). The pattern must be created after the translation, because it always uses the active coordinate system.



Using shading patterns for drawing

A pattern is like a color and be used like one. The examples on the previous pages used the pattern for filling rectangles. Why not use it for something more fancy like filling the outline of a text or drawing with a pattern.

Some text.