

Colour

Five colour spaces



gray 0.35



rgb 0.85 0.2 0.15



cmymk 0 .85 .85 .05



separation Varnish



icc chart (sRGB)

The second and third are the same red said two ways. On a screen they look alike; on a press they are different inks. A CMYK value under an sRGB output intent is a PDF/A risk - the intent claims a space the numbers are not in. The fifth is the same red a third way: the numbers of the second, anchored to the sRGB2014 profile - device independent, so it passes under any PDF/A output intent.

Grey ramp, eleven steps



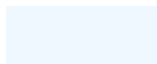
The same hue through both spaces

Converting RGB to CMYK naively is loss-free on the way back - measured: error 0.0 across 4096 colours, 0.78 microseconds per conversion. That is arithmetic, not colour management: it says nothing about how the ink will look.



upper: rgb lower: the same colour as cmyk

Named colours - 148 of them, without Tk



aliceblue



coral



crimson



darkolivegreen



dodgerblue



firebrick



gold



indigo



khaki



lavender



limegreen



maroon



navy



olive



orchid



peru



plum



salmon



seagreen



sienna



slateblue



steelblue



teal



thistle



tomato



turquoise



violet



wheat

A hexadecimal value works too: #f0a is short for #ff00aa, not for #f00a00.



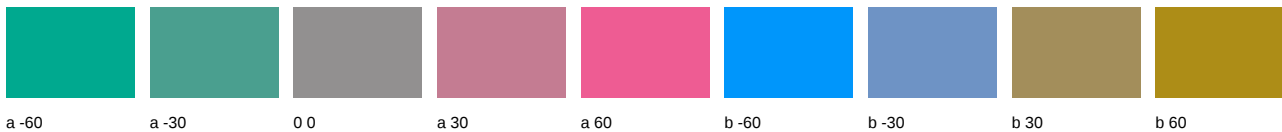
Lab

L* runs from 0 to 100, a* from green to red and b* from blue to yellow; both of those run over the range the space names, -100 to 100 unless -range says otherwise (ISO 32000-2, 8.6.5.4). The numbers are not 0..1 like every other space on the page before - a Lab colour clamped to that range would come out black.

L*, at a* 0 and b* 0

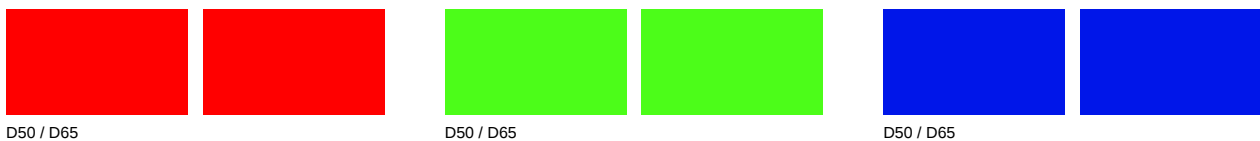


a* and b* at L* 60



The same three numbers under two white points

Left the default D50, right the D65 point of the example in 8.6.5.4. Nothing about the colour changed - only the light it is measured under, and that is a property of the SPACE, not of the value.



Outside the range, and the way past it

The third pair above is sRGB blue, whose b* is -112.05 - outside the default range, so it is pulled back to -100. The standard says to do that silently: "Component values falling outside the specified range shall be adjusted to the nearest valid value without error indication". The wider range of the example in 8.6.5.4 lets it through.



default range / -range {-128 127 -128 127}

A spot colour with a Lab alternate

One plate, eleven coverages. The name is what the pressroom will call it; the three numbers are what it measured. No colour catalogue is shipped with this package - the data is licensed - and this is the way that needs none.



separation "Spot Red" {lab 48.3 68.5 47.3}, tint 0 to 1

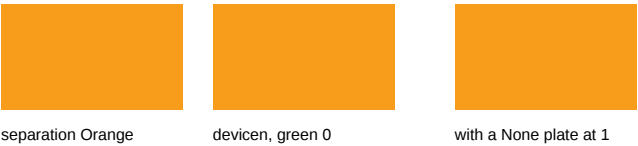
Lab is device independent like an ICC based colour, so PDF/A admits it under every output intent - measured with veraPDF: the page above passes 3B under an sRGB, a grey and a CMYK intent alike, where the same colours as {rgb ...} are refused under the CMYK one. A gradient cannot take a Lab colour: a shading names its space by family in its dictionary, and a Lab space is an array.

DeviceN

Each colourant is written the way a separation is: the name of the plate and the colour it paints at full tint. The tints follow in the same order, one per name. Below, an orange and a green plate crossed - the top row and the left column are each plate on its own, and every other square is the two of them overprinting.



One plate, two ways



None is the third colourant of the square on the right, at full tint - and it changes nothing: a component named None is never painted (ISO 32000-2, 8.6.6.5) and this writer keeps it out of the fallback colour as well. All, which a separation may be, is refused here: it means every plate at once and cannot be one component among several. Both are the standard's rules, and they differ from the ones for a separation.

The tint transform is a type 4 function - PostScript calculator code - because a type 2 or type 3 function takes ONE input by definition and this one needs n. Ink amounts add and are clipped by the function's own /Range; in an RGB or grey alternate, where the numbers count light rather than ink, they multiply instead, so no ink at all leaves paper. The space also carries a /Colorants attribute holding each plate on its own - the one thing a combined transform cannot say - which is why a DeviceN colour is written as PDF 1.6 here. Under PDF/A the alternate space is what the output intent judges: these plates fall back to CMYK, so the document needs a CMYK intent, exactly as a separation with a CMYK alternate does.

Blend modes

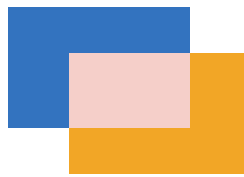
The blue square is drawn first, the orange one over it with the mode named underneath. Only the second call carries -blend; the mode is graphics state like the alpha, so it is wrapped in the shape's own save/restore and does not reach the next one.



Normal



Multiply



Screen



Overlay



Darken



Lighten



ColorDodge



ColorBurn



HardLight



SoftLight



Difference



Exclusion



Hue



Saturation



Color



Luminosity

Compatible is refused: it has been deprecated since PDF 1.4 and means Normal, so naming it says nothing a reader could act on. PDF/A parts 2 and 3 permit every mode above.

As state: opacity, with its second word



opacity 0.35 fill



opacity 0.35 stroke



opacity 0.35 both

The blend mode and the whole drawing state have a page of their own - the next one.

Blend modes as state

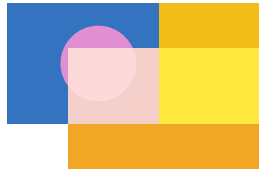
The same two colours as on the page before, drawn over blue, over gold and over the white of the page. This time the mode is set once per cell with [blend], and the rectangle and the circle after it carry nothing - the state does the work.



blend Normal



blend Multiply



blend Screen



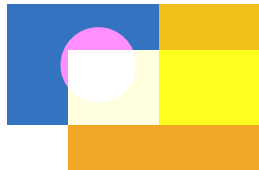
blend Overlay



blend Darken



blend Lighten



blend ColorDodge



blend ColorBurn



blend HardLight



blend SoftLight



blend Difference



blend Exclusion



blend Hue



blend Saturation



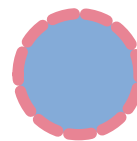
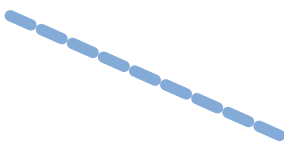
blend Color



blend Luminosity

blend Compatible - tclpdf: the blend mode Compatible is deprecated and means Normal - name Normal instead

style: the drawing state in one call



style -fill -stroke -width 1.5 -dash {3 1.5} -cap round -join bevel -miter 4 -opacity 0.6 -blend Multiply, then a line and a rect naming nothing, a polygon naming its stroke, a circle naming both

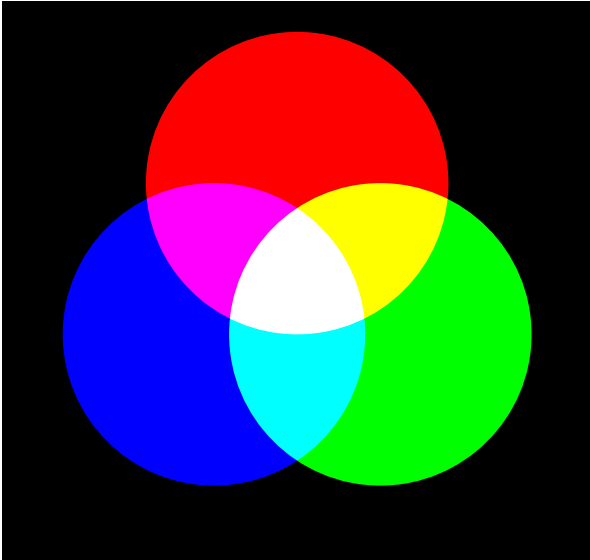
after the reset

Why there are two colour models

Both diagrams are three circles and nothing else. What differs is the ground they stand on and the blend mode: light adds up from black, ink takes away from white.

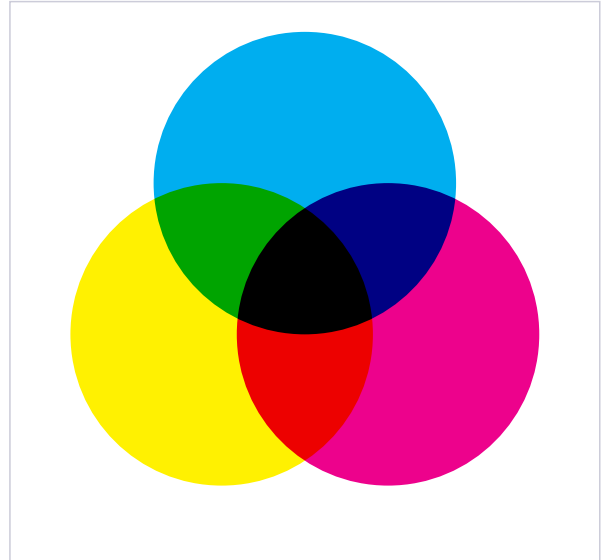
RGB - additive, light

Screen on a black panel. Where two lights overlap the result is brighter; all three make white.



CMYK - subtractive, ink

Multiply on bare paper. Where two inks overlap less light comes back; all three make near black.



And why there is a fourth plate

The middle above is where C, M and Y meet at full strength. On paper that is not black but a dark muddy brown, at 300 per cent ink coverage - which no press will lay down on ordinary stock. The K plate gives a neutral black with one ink instead of three.



cmyk 1 1 1 0 - three inks



cmyk 0 0 0 1 - one ink

On screen these two look alike, and that is the honest answer: a reader has no inks. It converts both to RGB and lands on much the same grey. The difference is on paper, and it is the reason a control strip is printed rather than previewed - see example 01.16.