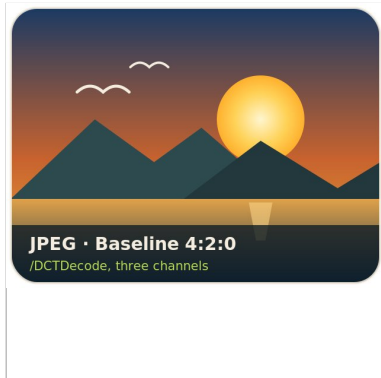


Fitting a picture into a box

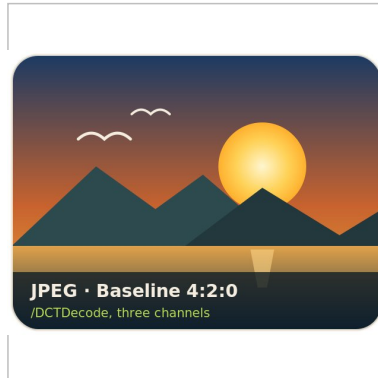
The photograph is 640 by 480 pixels and states no resolution, so its natural size is 225.8 by 169.3 mm - four by three. Every box below is 50 by 50, a square, so contain and cover always differ. The thin rule is the box; the picture is what is inside it.

contain - the whole picture inside the box

The picture is scaled until BOTH edges fit, which leaves a band on the axis that had room to spare. -valign says where the band goes.



-valign top (the default)



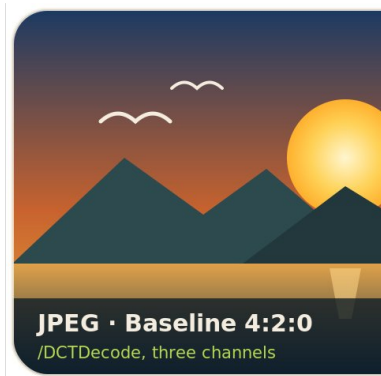
-valign middle



-valign bottom

cover - the whole box covered, the overhang cut away

The picture is scaled until NEITHER edge falls short, so it hangs over the box on one axis and is clipped back to it. -align then decides which part of the picture survives - which is the whole point: a portrait is cropped at the sides, not through the face.



-align left (the default)



-align center

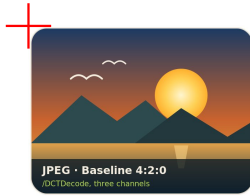


-align right

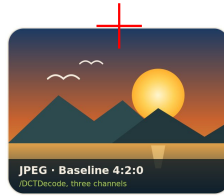
A picture that covers its box declares the BOX as its bounding box in a tagged document, not the placement: ISO 32000-2, 14.8.5.4.3 asks for the rectangle that encloses the VISIBLE content, and three quarters of a covering picture is not visible.

The same two options without a box

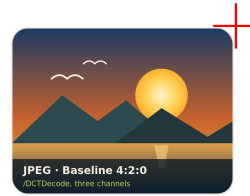
Without -fit the box has no size, and the same arithmetic is the plain anchor: the corner moves by the fraction times the picture's own extent. The cross marks -at in each case.



-align left -valign top



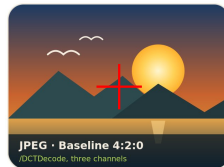
-align center -valign top



-align right -valign top



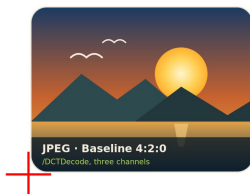
-align left -valign middle



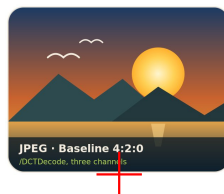
-align center -valign middle



-align right -valign middle



-align left -valign bottom



-align center -valign bottom



-align right -valign bottom

The right column is what this is for: a logo whose right edge is to sit on the right margin of the type area needs -align right and no arithmetic at all, and it stays right where the logo is replaced by one of another shape.

```
$doc image place photo -at {20 20} -fit {40 40} -rotate 90
```

tcldpf: -rotate of image place turns the placement about the point -at names, and -fit places it in an upright box - a turned placement leaves that box, and a covered box would be cut by a rectangle standing at an angle to it; ask for the fitted size ("image size" takes -fit and answers it) and give it with -size, which -rotate turns about -at as before

The fitted size is what 40.0 30.0 gives, and -size takes it - so a turned picture scaled to a box is one line, and it is the caller who decides where the turn puts it.

How the document opens

initialView writes the three catalogue entries of ISO 32000-2, Table 28. -pageMode says which panel a reader shows beside the page (UseNone, UseOutlines, UseThumbs, FullScreen, UseOC, UseAttachments), -pageLayout how it arranges the pages (SinglePage, OneColumn, TwoColumnLeft, TwoColumnRight, TwoPageLeft, TwoPageRight), and -page with -to and -zoom where it puts the reader first. They are a command of their own rather than options of viewerPreferences: a viewer preference is a wish a reader may ignore, and these are what it does when the file opens.

```
$doc initialView -pageMode UseOutlines \\  
  -pageLayout TwoColumnLeft -page 1 -to {20 20}
```

That is what this file carries, so a reader opens it on page two, in two columns, with the bookmark pane out. An /OpenAction may also be an ACTION rather than a destination, and this command does not write one: measured with veraPDF 1.30, a /Launch and a /JavaScript open action each fail PDF/A clause 6.5.1 - so an -action option would be a promise half of whose values break the moment pdfa is declared. Where one is really wanted, catalogEntry OpenAction takes any dictionary.