IT'S IN THE DETAILS

HANS HAGEN
PRAGMA ADE
HASSELT NL

On the ConTEXt mailing list, occasionally a user asks if we can post a complete document with the associated style. One reason for not honouring this request is that we want users to cook up their own styles. Besides that, there are a couple of styles in the regular ConTEXt distribution.

When browsing through this document, a ConTEXt user may wonder what style was used to achieve its look and feel. We hope that while reading the text and playing with the examples, the reader will accomplish the skills to define more than just simple layouts.

This document is not easy reading. Occasionally we spend some time explaining features not described in other manuals. The design of this document is to a large extent determined by its purpose, and as a result not always functional. For instance, we typeset on a grid which doesn't look too good. Also the order of presenting features, tips and tricks is kind of random and unstructured. The idea is that the visual effects will draw you to the right trick. Also, if you really want to benefit from these features, there is no way but to read the whole story.

In spite of all its shortcomings, I hope that you enjoy reading this (yet unfinished) manual. Keep in mind that this manual is far from finished.

Hans Hagen
Hasselt NL

2002^+ MkII
2015^+ MkIV

The grid snapper in MkIV is quite different from the one in MkII. For not too complex layouts the old grid snapper was quite ok, but the new one should be a bit more robust. In the old situation the running text was assumed to fit on the grid and the normal baseline skip should do the job in combination with the grid aware spacing features and placement mechanisms like tables and figures.

Snapping on a fixed grid is sort of counter intuitive in TEX because it has an a advanced spacing model with glue. Publishers however love grids so we do need to support it. Of course when complex layouts are involved in a later stage of document preparation the grid is often abandoned. This manual uses the grid but I personally never use the grid. There are better ways to make your document look good and often a grid snapped document doesn't look that great anyway, because all elements should somehow fit in multiples of the line height.

The MkIV snapper does more analysis and therefore can compensate for the more nasty cases. Of course it can still fail but we hope to fix those cases when we run into them. Grids are controlled by keywords or a combination of them.

none	don't enlarge
halfline	enlarge by halfline/halfline
line	enlarge by line/line
strut	enlarge by ht/dp (default)
first	align to top line
last	align to bottom line
mindepth	round depth down
maxdepth	round depth up
minheight	round height down
maxheight	round height up
local	use local interline space
offset:-3tp	vertical shift within box
bottom:lines	
top:lines	
box	centers a box rounded upwards (box:.5 -> tolerance)
min	centers a box rounded downwards
max	centers a box rounded upwards

We combine these directives in so called grid options:

```
\definegridsnapping [normal]   [maxheight,maxdepth,strut]
\definegridsnapping [standard] [maxheight,maxdepth,strut]
\definegridsnapping [yes]      [maxheight,maxdepth,strut]

\definegridsnapping [strict]   [maxdepth:0.8,maxheight:0.8,strut]
\definegridsnapping [tolerant] [maxdepth:1.2,maxheight:1.2,strut]
\definegridsnapping [math]     [maxdepth:1.05,maxheight:1.05,strut]

\definegridsnapping [top]      [minheight,maxdepth,strut]
\definegridsnapping [bottom]   [maxheight,mindepth,strut]
```

```
\definegridsnapping [both]      [minheight,mindepth,strut]

\definegridsnapping [broad]     [maxheight,maxdepth,strut,0.8]
\definegridsnapping [fit]       [maxheight,maxdepth,strut,1.2]

\definegridsnapping [first]     [first]
\definegridsnapping [last]      [last]
\definegridsnapping [high]      [minheight,maxdepth,none]
\definegridsnapping [one]       [minheight,mindepth]
\definegridsnapping [low]       [maxheight,mindepth,none]
\definegridsnapping [none]      [none]
\definegridsnapping [line]      [line]
\definegridsnapping [strut]     [strut]
\definegridsnapping [box]       [box]
\definegridsnapping [min]       [min]
\definegridsnapping [max]       [max]

\definegridsnapping [middle]    [maxheight,maxdepth]
```

As you can see, keys like `maxdepth` can take a criterium as extra detail, separated by a colon. These options look obscure and often you need to trial and error a bit to get what you want. This is no real problem because most cases are handles automatically. Only headings and structuring elements that exceed a line height might need some treatment. For instance you might want to be more tolerant for (fractions of) a point overflow or when you know that always a blank follows you can decide to limit the height of some element to a line. Some of the options, like `math` and `middle` are used internally.

On the next pages we show how the `maxheight` and `maxdepth` fractions work on a sample line.

maxdepth:0.0,maxheight:0.0

maxdepth:0.0,maxheight:0.1

maxdepth:0.0,maxheight:0.2

maxdepth:0.0,maxheight:0.3

maxdepth:0.0,maxheight:0.4

maxdepth:0.0,maxheight:0.5

maxdepth:0.0,maxheight:0.6

maxdepth:0.0,maxheight:0.7

maxdepth:0.0,maxheight:0.8

maxdepth:0.0,maxheight:0.9

maxdepth:0.0,maxheight:10

maxdepth:0.1,maxheight:0.0

maxdepth:0.1,maxheight:0.1

maxdepth:0.1,maxheight:0.2

maxdepth:0.1,maxheight:0.3

maxdepth:0.1,maxheight:0.4

maxdepth:0.1,maxheight:0.5

maxdepth:0.1,maxheight:0.6

maxdepth:0.1,maxheight:0.7

maxdepth:0.1,maxheight:0.8

maxdepth:0.1,maxheight:0.9

maxdepth:0.1,maxheight:10

maxdepth:0.2,maxheight:0.0

maxdepth:0.2,maxheight:0.1

maxdepth:0.2,maxheight:0.2

maxdepth:0.2,maxheight:0.3

maxdepth:0.2,maxheight:0.4

maxdepth:0.2,maxheight:0.5

maxdepth:0.2,maxheight:0.6

maxdepth:0.2,maxheight:0.7

maxdepth:0.2,maxheight:0.8

maxdepth:0.2,maxheight:0.9

maxdepth:0.2,maxheight:10

maxdepth:0.3,maxheight:0.0

maxdepth:0.3,maxheight:0.1

maxdepth:0.3,maxheight:0.2

maxdepth:0.3,maxheight:0.3

maxdepth:0.3,maxheight:0.4

maxdepth:0.3,maxheight:0.5

maxdepth:0.3,maxheight:0.6

maxdepth:0.3,maxheight:0.7

maxdepth:0.3,maxheight:0.8

maxdepth:0.3,maxheight:0.9

maxdepth:0.3,maxheight:10

maxdepth:0.4,maxheight:0.0

maxdepth:0.4,maxheight:0.1

maxdepth:0.4,maxheight:0.2

maxdepth:0.4,maxheight:0.3

maxdepth:0.4,maxheight:0.4

maxdepth:0.4,maxheight:0.5

maxdepth:0.4,maxheight:0.6

maxdepth:0.4,maxheight:0.7

maxdepth:0.4,maxheight:0.8

maxdepth:0.4,maxheight:0.9

maxdepth:0.4,maxheight:10

maxdepth:0.5,maxheight:0.0

maxdepth:0.5,maxheight:0.1

maxdepth:0.5,maxheight:0.2

maxdepth:0.5,maxheight:0.3

maxdepth:0.5,maxheight:0.4

maxdepth:0.5,maxheight:0.5

maxdepth:0.5,maxheight:0.6

maxdepth:0.5,maxheight:0.7

maxdepth:0.5,maxheight:0.8

maxdepth:0.5,maxheight:0.9

maxdepth:0.5,maxheight:10

maxdepth:0.6,maxheight:0.0

maxdepth:0.6,maxheight:0.1

maxdepth:0.6,maxheight:0.2

maxdepth:0.6,maxheight:0.3

maxdepth:0.6,maxheight:0.4

maxdepth:0.6,maxheight:0.5

maxdepth:0.6,maxheight:0.6

maxdepth:0.6,maxheight:0.7

maxdepth:0.6,maxheight:0.8

maxdepth:0.6,maxheight:0.9

maxdepth:0.6,maxheight:10

maxdepth:0.7,maxheight:0.0

maxdepth:0.7,maxheight:0.1

maxdepth:0.7,maxheight:0.2

maxdepth:0.7,maxheight:0.3

maxdepth:0.7,maxheight:0.4

maxdepth:0.7,maxheight:0.5

maxdepth:0.7,maxheight:0.6

maxdepth:0.7,maxheight:0.7

maxdepth:0.7,maxheight:0.8

maxdepth:0.7,maxheight:0.9

maxdepth:0.7,maxheight:10

maxdepth:0.8,maxheight:0.0

maxdepth:0.8,maxheight:0.1

maxdepth:0.8,maxheight:0.2

maxdepth:0.8,maxheight:0.3

maxdepth:0.8,maxheight:0.4

maxdepth:0.8,maxheight:0.5

maxdepth:0.8,maxheight:0.6

maxdepth:0.8,maxheight:0.7

maxdepth:0.8,maxheight:0.8

maxdepth:0.8,maxheight:0.9

maxdepth:0.8,maxheight:10

maxdepth:0.9,maxheight:0.0

maxdepth:0.9,maxheight:0.1

maxdepth:0.9,maxheight:0.2

maxdepth:0.9,maxheight:0.3

maxdepth:0.9,maxheight:0.4

maxdepth:0.9,maxheight:0.5

maxdepth:0.9,maxheight:0.6

maxdepth:0.9,maxheight:0.7

maxdepth:0.9,maxheight:0.8

maxdepth:0.9,maxheight:0.9

maxdepth:0.9,maxheight:10

maxdepth:10,maxheight:0.0

maxdepth:10,maxheight:0.1

maxdepth:10,maxheight:0.2

maxdepth:10,maxheight:0.3

maxdepth:10,maxheight:0.4

maxdepth:10,maxheight:0.5

maxdepth:10,maxheight:0.6

maxdepth:10,maxheight:0.7

maxdepth:10,maxheight:0.8

maxdepth:10,maxheight:0.9

maxdepth:10,maxheight:10

Next we show some of the options in action. For practical reasons we start a new page each time. The sample is input as:

```
\bf                                none \par
\bfb \hskip2cm                     none \par
\bfd \hskip6cm                     none \par
\bf                                test \par
\bfb \hskip2cm                     test \par
\bfd \hskip6cm                     test \par
\bf                                grid \par
\bfb \hskip2cm                     grid \par
\bfd \hskip6cm                     grid \par
\bf                         \strut strut \par
\bfb \hskip2cm              \strut strut \par
\bfd \hskip6cm              \strut strut \par
\bfb \hskip2cm \setstrut \strut setstrut \par
\bfd \hskip6cm \setstrut \strut setstrut \par
```

1.1 Grid snapping method “normal”

This is just a line to start with but next we show what method `normal` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.2 Grid snapping method “strict”

This is just a line to start with but next we show what method `strict` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.3 Grid snapping method “tolerant”

This is just a line to start with but next we show what method `tolerant` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.4 Grid snapping method “top”

This is just a line to start with but next we show what method `top` does.

none
none
none
test
test
test
grid
grid
grid
strut
strut
strut
setstrut
setstrut

And here we end the demo.

1.5 Grid snapping method “bottom”

This is just a line to start with but next we show what method `bottom` does.

none
none
none
test
test
test
grid
grid
grid
strut
strut
strut
setstrut
setstrut

And here we end the demo.

1.6 Grid snapping method “both”

This is just a line to start with but next we show what method `both` does.

none

test

grid

strut

none

test

grid

strut

setstrut

none

test

grid

strut

setstrut

And here we end the demo.

1.7 Grid snapping method “broad”

This is just a line to start with but next we show what method `broad` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.8 Grid snapping method “fit”

This is just a line to start with but next we show what method `fit` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.9 Grid snapping method “first”

This is just a line to start with but next we show what method `first` does.

none

test

grid

strut

none

test

grid

strut

setstrut

none

test

grid

strut

setstrut

And here we end the demo.

1.10 Grid snapping method “last”

This is just a line to start with but next we show what method `last` does.

none

test

grid

strut

none

test

grid

strut

setstrut

none

test

grid

strut

setstrut

And here we end the demo.

1.11 Grid snapping method “high”

Here is just a line to start with. Next we show what method `high` does.

test grid strut

test grid strut setstrut

test grid strut setstrut

And here we end the demo.

1.12 Grid snapping method “one”

This is just a line to start with but next we show what method `one` does.

none

test

grid

strut

none

test

grid

strut

setstrut

none

test

grid

strut

setstrut

And here we end the demo.

1.13 Grid snapping method “low”

This is just a line to start with but next we show what method `low` does.

none
test
grid
strut

none
test
grid
strut
setstrut

none
test
grid
strut
setstrut

And here we end the demo.

1.14 Grid snapping method “none”

[illegible] just a line to start with [illegible] we show what method none does.
And here we end the demo.

1.15 Grid snapping method “line”

This is just a line to start with but next we show what method `line` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.16 Grid snapping method “strut”

This is just a line to start with but next we show what method `strut` does.

none

test

grid

strut

none

test

grid

strut

setstrut

none

test

grid

strut

setstrut

And here we end the demo.

1.17 Grid snapping method “box”

This is just a line to start with but next we show what method `box` does.

none
test
grid
strut

none
test
grid
strut
setstrut

none
test
grid
strut
setstrut

And here we end the demo.

1.18 Grid snapping method “min”

This is just a line to start with but next we show what method `min` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.19 Grid snapping method “max”

This is just a line to start with but next we show what method `max` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

1.20 Grid snapping method “middle”

This is just a line to start with but next we show what method `middle` does.

none

none

none

test

test

test

grid

grid

grid

strut

strut

strut

setstrut

setstrut

And here we end the demo.

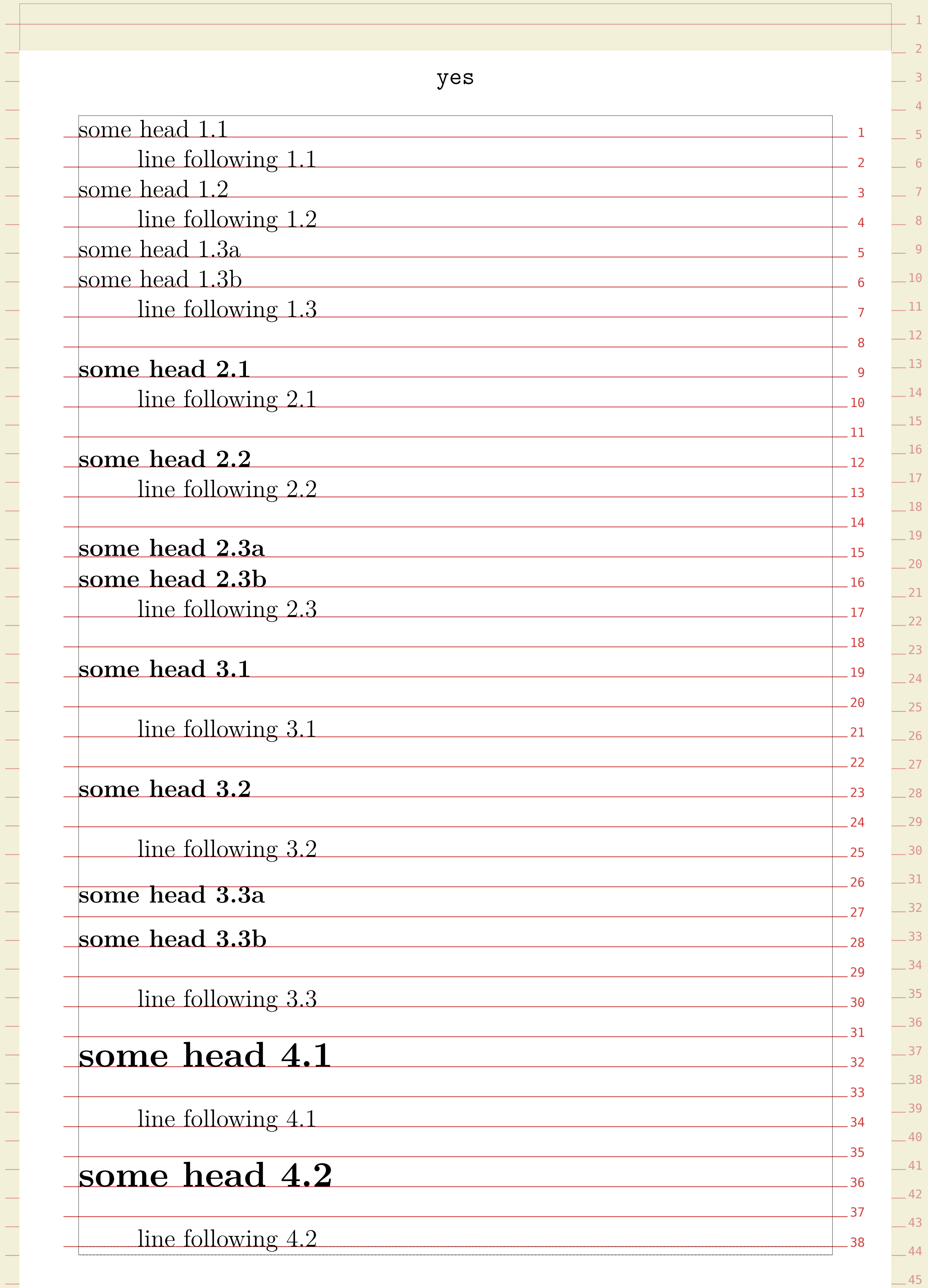
We now come to the topic of this chapter: snapping heads. The problem with section heads is that they often exceed the line height. Even worse, they can be more than one line high.

The next pages show some ways to control snapping around heads. The result can be confusing, even when we use a font that we assume behaves like a regular style. For instance in Latin Modern the bold style has larger heights and depths than the regular style and even 0.1pt can force the snapper to add a line. The examples use that font.

The grid option of `setuphead` normally takes one keyword that refers to the local snapper. However, the result gets then snapped again. This is because the local snapper can use a different line height. Historically the local snapper is the default but you can force global snapping by prefixing with the `global` keyword. The next table summarizes the ways you can control snapping:

`(nothing)`	local snapping plus global snapping
`local`	local snapping plus global snapping
`foo`	local `foo` snapping cf. font style plus global snapping
`local:foo`	local `foo` snapping cf. font style plus global snapping
`global`	global snapping
`global:foo`	global `foo` snapping

```
   \bf                             none \par
   \bfb \hskip2cm                  none \par
   \bfd \hskip6cm                  none \par
   \bf                             test \par
   \bfb \hskip2cm                  test \par
   \bfd \hskip6cm                  test \par
   \bf                             grid \par
   \bfb \hskip2cm                  grid \par
   \bfd \hskip6cm                  grid \par
   \bf                      \strut strut \par
   \bfb \hskip2cm           \strut strut \par
   \bfd \hskip6cm           \strut strut \par
   \bfb \hskip2cm \setstrut \strut setstrut \par
   \bfd \hskip6cm \setstrut \strut setstrut \par
```



yes

some head 1.1

line following 1.1

some head 1.2

line following 1.2

some head 1.3a

some head 1.3b

line following 1.3

some head 2.1

line following 2.1

some head 2.2

line following 2.2

some head 2.3a

some head 2.3b

line following 2.3

some head 3.1

line following 3.1

some head 3.2

line following 3.2

some head 3.3a

some head 3.3b

line following 3.3

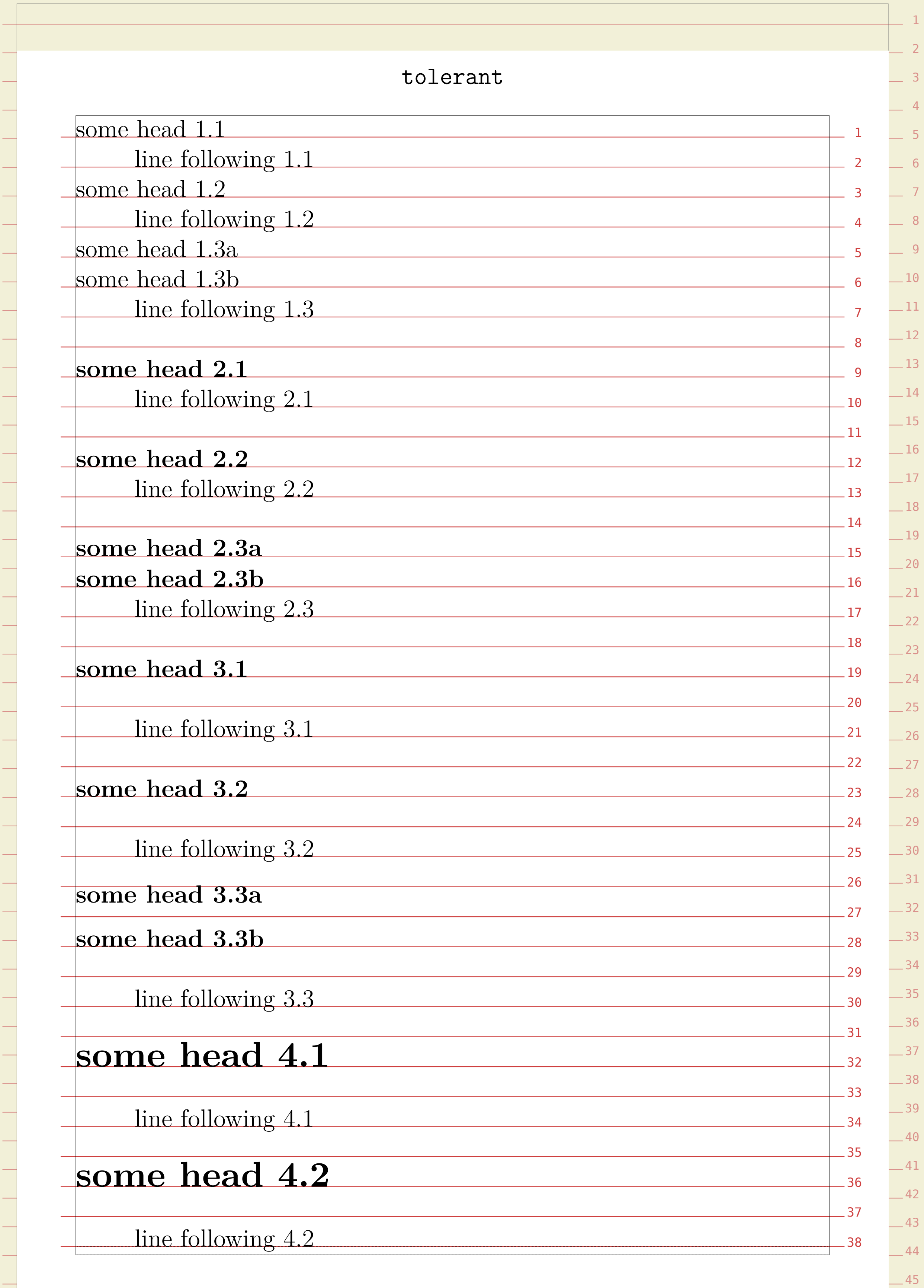
some head 4.1

line following 4.1

some head 4.2

line following 4.2

Figure 1.1



`tolerant`

some head 1.1

line following 1.1

some head 1.2

line following 1.2

some head 1.3a

some head 1.3b

line following 1.3

some head 2.1

line following 2.1

some head 2.2

line following 2.2

some head 2.3a

some head 2.3b

line following 2.3

some head 3.1

line following 3.1

some head 3.2

line following 3.2

some head 3.3a

some head 3.3b

line following 3.3

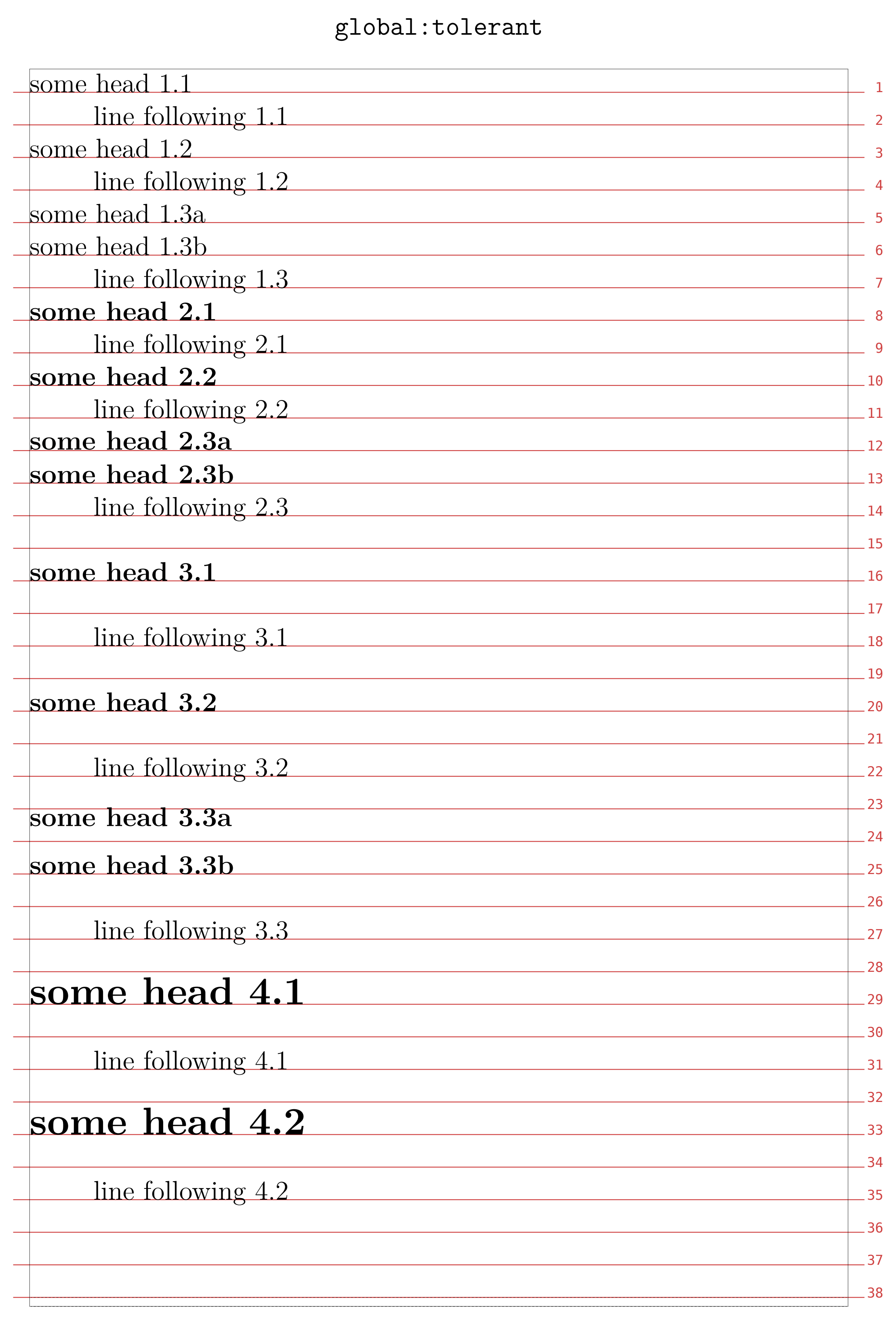
some head 4.1

line following 4.1

some head 4.2

line following 4.2

Figure 1.2



`global:tolerant`

some head 1.1

line following 1.1

some head 1.2

line following 1.2

some head 1.3a

some head 1.3b

line following 1.3

some head 2.1

line following 2.1

some head 2.2

line following 2.2

some head 2.3a

some head 2.3b

line following 2.3

some head 3.1

line following 3.1

some head 3.2

line following 3.2

some head 3.3a

some head 3.3b

line following 3.3

some head 4.1

line following 4.1

some head 4.2

line following 4.2

Figure 1.3

In desk top publishing applications the grid is pretty dominant in defining layouts. On the other hand, TEX is pretty good defining layouts in terms of relative dimensions. This means that mapping a desk top publishing layout into its TEX (or ConTEXt) counterpart takes some effort. For what it's worth, personally I don't like grids that much, specially not in complex documents, unless one makes sure that all elements are suitable sized for the grid used.

We not only have to deal with vertical grids, but also with horizontal ones. Here we focus on the second category. When implementing designs, it is best first to look into the normal page layout areas. For most documents these are sufficient, but occasionally we need a more detailed approach.

When playing with grids, you need to make sure that grid snapping is turned on. It helps if you turn on the grid so that you can see where things end up. When a horizontal grid is defined, gray vertical rules show their boundaries.

```
\setuplayout[grid=yes] \showgrid
```

The `\setuplayout` command has a few settings that have to do with so called pseudo columns. These are in no sense related to multi-column typesetting and only play a role in placing text on specific locations.

```
\setuplayout
  [columndistance=12pt,
   columns=3]
```

You can use `\layoutcolumnoffset` for positioning relative to the left boundary of the running text:

```
\hskip\layoutcolumnoffset{2}{\red Text positioned in column 2!}
```

Text positioned in column 2!

This mechanism is actually meant to ease the definition of complicated (title) pages where many text and graphic elements need to be anchored at well defined places. The layer mechanism is the most natural candidate for this.

```
\definelayer [text] \setupbackgrounds [text] [background=text]
```

When anchoring elements on a layer, you can specify absolute positions using the `x` and `y` keys but grid based positioning is possible with the `column` and `line` keys. We need to pass `grid` as location specifier.

```
\setlayer[text][column=1,line=48,location=grid]{these are not}
\setlayer[text][column=2,line=47,location=grid]{real columns}
\setlayer[text][column=3,line=48,location=grid]{but fake ones}
```

real columns

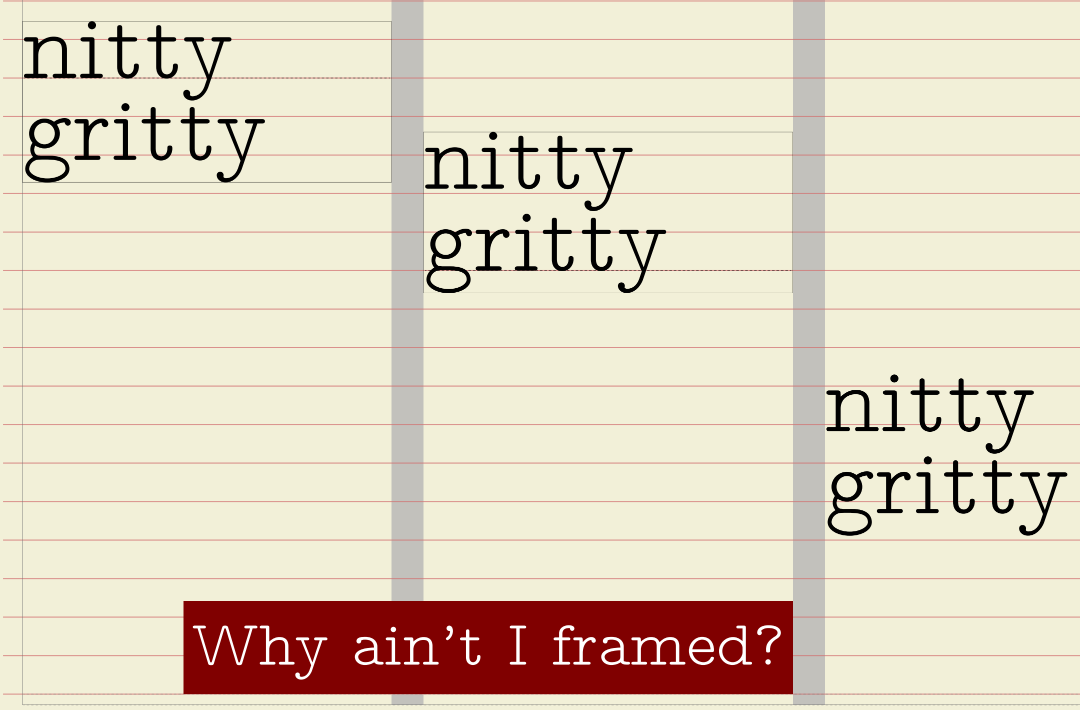
these are not but fake ones

```
\setlayer [text] [column=1,line=32,location=grid]
  {\ruledvtop {\hsize\layoutcolumnwidth
     \style[regular:3]{nitty\par gritty}}}

\setlayer [text] [column=2,line=37,location=grid]
  {\ruledvbox {\hsize\layoutcolumnwidth
     \style[regular:3]{nitty\par gritty}}}

\setlayer [text] [column=3,line=42,location=grid]
  {\ruledvcenter {\hsize\layoutcolumnwidth
     \style[regular:3]{nitty\par gritty}}}
```

The data that goes into the layer is collected and flushed as soon as TEX builds the page. The buffer associated to the layer is then ready for new data (for the next page).

In this example, you can see that the baselines of the boxes (here visualized by dashed rules) are put at the specified lines. You can use the TEX box commands `\vbox`, `\vtop` and `\vcenter` to specify where the main baseline of the box content is positioned (at the top or bottom line, or centered).

```
\setlayer
  [text]
  [column=2,line=48,x=\layoutcolumnwidth,location=left]
  {\framed
     [background=color,backgroundcolor=red,
      foregroundstyle=regular:2,foregroundcolor=white,
      frame=off]
     {Why ain't I framed?}}
```



On the previous page we demonstrated a more complicated call to `\setlayer` and more features will be introduced in later chapters. We position the framed text in column 2 and at line 48. In addition we shift the text over the pseudo column width, i.e. we position the text at the right of the column. The location specifier aligns the text left from the point of positioning.

When we have set up the pseudo columns, we have access to a couple of variables:

`\layoutcolumns`	counter	number of columns
`\layoutlines`	counter	number of gridlines
`\layoutcolumnwidth`	dimension	width of one column
`\layoutcolumnoffset{n}`	macro	position of column n

This is typically a feature that has been there for quite a while but that I forget about. It’s probably because I never have to use grids myself.

In the examples before we used some predefined (font) styles:

```
\definefont[regular:1][Regular*default sa 1]
\definefont[regular:2][Regular*default sa 2]
\definefont[regular:3][Regular*default sa 3]
\definefont[regular:4][Regular*default sa 4]
```

A rather common way to draw attention to a passage, is to add a background. In this chapter we will therefore discuss how to enhance your document with those colorful areas that either or not follow the shape of your paragraph. Be warned: this chapter has so many backgrounds that you might start to dislike them.

In the previous paragraph we demonstrated two important features of the background handler: you can nest backgrounds and backgrounds can be tight or wide. Features like this will often be used in combination with others, like special section headers. The raw coding of the previous paragraph is therefore not representative.

```
\starttextbackground[intro]
A rather common way to draw attention to a passage, is to add a
background. In this chapter we will therefore discuss how to enhance
your
document with \starttextbackground [subintro] those colorful areas
that either
or not follow the shape of your paragraph. \stoptextbackground\ Be
warned: this chapter has so many backgrounds that you might start
to
dislike them.
\stoptextbackground
```

The outer background commands is defined as follows:

```
\definetextbackground
  [intro]
  [backgroundcolor=infogray,
   backgroundoffset=.25cm,
   frame=off,
   location=paragraph,
   color=red]
```

Here, the `paragraph` option ensures that the background covers the width of the body text. The inner background is defined in a similar way, but this time we choose `text` location.

```
\definetextbackground
  [subintro]
  [backgroundcolor=textgray,
   backgroundoffset=0pt,
   frame=off,
   location=text,
   color=blue]
```

In this document we use protruding characters (hanging punctuation) so we’ve chosen a rather large offset, one that also matches the rest of the page design. Those who are familiar with the way TEX works will probably see what problems can occur with backgrounds like this. What happens for instance when we cross page boundaries, and how will more complicated paragraph shapes be handled? The current implementation tries to handle page breaks and paragraph shapes as good as possible. This works well in normal one–column mode as well as in columns.

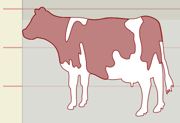


Figure 3.1

In this example, the paragraph shape is determined by the graphic placed left of the text. This feature is implemented using the `\hangindent` and `\hangafter` primitives, which means that we need to keep track of their state. In addition, we need to handle the indentation directives `\leftskip`, `\rightskip` and `\parindent`. Because backgrounds end up in a different background overlay, nesting them is no problem, and it is even possible to move them to the front and back, as we will demonstrate later on. While the mechanism discussed here will always be improved when we find border cases, the fundaments it is built upon are quite stable.

```
\placefigure[left]{}{\externalfigure[detcow][width=2cm]}

\starttextbackground [A]
  In this example, the paragraph shape is determined by the graphic
placed
  left of the text.
    \starttextbackground [B]
      This feature is implemented using the \type {\hangindent} and
\type
      {\hangafter} primitives, which means that we need to keep
track of
      their state. In addition, we need to handle the indentation
directives
      \type {\leftskip}, \type {\rightskip} and \type {\parindent}.
    \stoptextbackground\
  Because backgrounds end up in a different background overlay,
nesting
  them is no problem, and it is even possible to move them to the
front
  and back, as we will demonstrate later on. While the mechanism
discussed
  here will always be improved when we find border cases, the
fundaments
  it is built upon are quite stable.
\stoptextbackground
```

The backgrounds were defined as:

```
\definetextbackground [A] [backgroundcolor=infogray]
\definetextbackground [B] [backgroundcolor=textgray]

\setuptextbackground
```

```
      [backgroundoffset=0pt,
       offset=0pt,
       frame=off,
       location=text]
```

In this example, the paragraph shape is determined by the graphic placed left of the text. [redacted] Because backgrounds end up in a different background overlay, nesting them is no problem, and it is even possible to move them to the front and back, as we will demonstrate later on. While the mechanism discussed here will always be improved when we find border cases, the fundaments it is built upon are quite stable.

Figure 3.2

This time we moved the inner background a few levels up. By default they reside at `level=-1`. This way, by using a non transparent color, we can hide information.

```
\setuptextbackground [B] [backgroundcolor=darkgray,level=+2]
```

Unless you mess around too much with boxes, backgrounds work as expected in most situations. According to the Merriam–Webster on the authors laptop:

background	the part of a painting representing what lies behind objects is the foreground	one
foreground	the part of a scene or representation that is nearest to and in front of the spectator	two
spectator	one who looks on or watches	three

This is coded similar to normal running text. A table like this is in a way still part of the text flow. As floating body (see table 3.1) it can virtually end up everywhere. We add a frame to make clear where the boundaries are.

background	the part of a painting representing what lies behind objects is the foreground	one
foreground	the part of a scene or representation that is nearest to and in front of the spectator	two
spectator	one who looks on or watches	three

Table 3.1

background	the part of a painting representing what lies behind objects is the foreground	one
foreground	the part of a scene or representation that is nearest to and in front of the spectator	two
spectator	one who looks on or watches	three

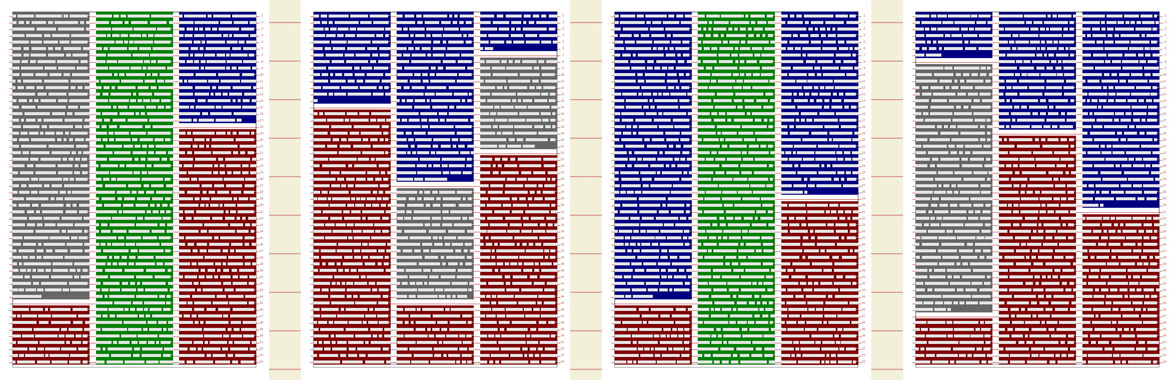
Table 3.2

Keeping track of the state of a paragraph in a table in combination with background is not entirely trivial. The current implementation evolved from less clever ones and, unless you start doing complicated box manipulations with the float content, works quite well. One reason why we made backgrounds work in tables (and especially floating tables) is that is was needed for typesetting books for primary and secundary education. In there, we want to be able to hide the answers that students are supposed to fill in.

In figure 3.3 you can see an advanced example of backgrounds running over columns. If you look carefully, you will notice that the background depends on the kind of background at hand:

1. the text starts and flows on
2. the text flows on (or stands alone)
3. the text flows on and ends

This information is available when you want to draw your own backgrounds. Here the graphic was defined as follows:



Page 1 Page 2 Page 3 Page 4

Figure 3.3

```
\startuseMPgraphic{mpos:par:color}
  for i=1 upto nofmultipars :
    fill multipars[i] withcolor
      if     multikind[i]="single" : "darkgray" ;
      elseif multikind[i]="first"  : "red" ;
      elseif multikind[i]="middle" : "green" ;
      elseif multikind[i]="last"   : "blue" ;
      else                         : "black" ;
      fi ;
  endfor ;
\stopuseMPgraphic
```

This graphic is hooked into the background setup by setting the `mp` variable.

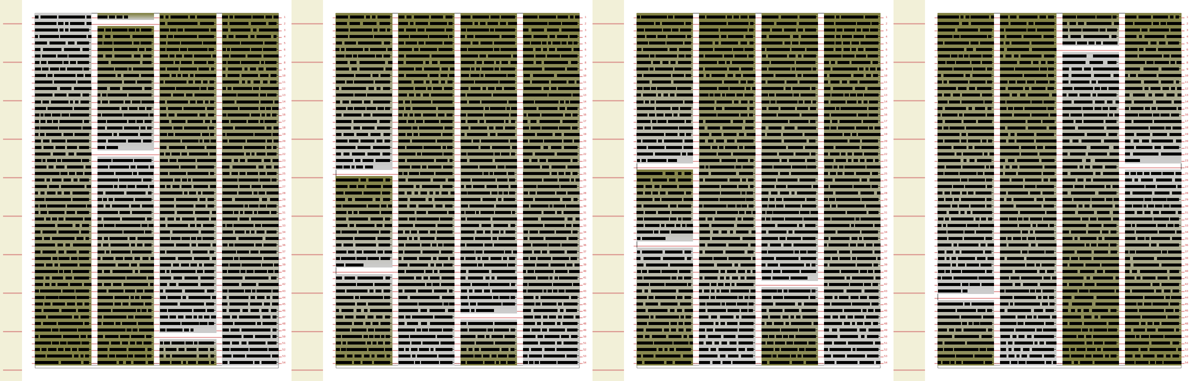
```
\definetextbackground
  [shade]
  [location=paragraph,
   mp=mpos:par:color,
   before=\blank,
   after=\blank]
```

A variant is the following. This time we use a shade:

```
\startuseMPgraphic{mpos:par:columnset:shade}
  numeric h ;
  for i=1 upto nofmultipars :
    h := bbheight(p) ;
    if multikind[i] = "single" :
      fill multipars[i] topenlarged -.5h
        withshademethod "linear"
        withshadedirection shadedup
        withcolor boxfillcolor shadedinto .8white ;
      fill multipars[i] bottomenlarged -.5h
        withshademethod "linear"
        withshadedirection shadedup
        withcolor .8white shadedinto boxfillcolor ;
    elseif multikind[i] = "first" :
      fill multipars[i]
        withshademethod "linear"
        withshadedirection shadedup
        withcolor boxfillcolor shadedinto .8white ;
    elseif multikind[i] = "middle" :
      fill multipars[i] topenlarged -.5h
        withshademethod "linear"
        withshadedirection shadedup
        withcolor boxfillcolor shadedinto .8white ;
      fill multipars[i] bottomenlarged -.5h
        withshademethod "linear"
        withshadedirection shadedup
        withcolor .8white shadedinto boxfillcolor ;
    elseif multikind[i] = "last" :
      fill multipars[i]
        withshademethod "linear"
        withshadedirection shadedup
        withcolor .8white shadedinto boxfillcolor ;
    fi ;
  endfor ;
\stopuseMPgraphic
```

When we hook it into the background we get figure 3.4 as result:

```
\definetextbackground
  [shade]
  [location=paragraph,
   backgroundcolor=shadecolor,
   mp=mpos:par:columnset:shade,
   before=\blank,
   after=\blank]
```



Page 1 Page 2 Page 3 Page 4

Figure 3.4

The complexity of the backgrounds mechanism is partly due to the fact that we want to use arbitrary MetaPost code to render the background. For instance, we want to have a proper shape so that not only the filled shape but also the drawn shape comes out right. You can compare this to a glyph in a font: when rendered filled the outline can be anything as it will not be drawn but when we use the outline we can run into overlaps and such. Where glyphs can use the odd-even filling methods, background can only use that for simple cases.

When a background is rectangular it's all quite easy but as soon as some holes occur we need to do more work. Holes can be the result of a image placed next to the running text, or an image flushed at a page break or in the middle of a background. Paragraph shapes are another example. Backgrounds can cross page boundaries too. Yet another property is nesting and in such cases the shape is a bit more complex as we cross lines partially.

In MkII the background mechanism already was quite useable but it had some limitations. Calculating the background was mostly delegated to MetaPost which is reasonable. In MkIV some work is delegated to Lua instead but that doesn't mean that the code is cleaner or easier to understand. So, to summarize, there are several cases that we need to take into account, like:

- A background can run behind a paragraph in which case the start is leftmost and end rightmost. In this case inserts (like floats) have to be dealt with after the shape has been calculated.

- A background can be in-line (the `text` location variant) in which case we need to follow the paragraph shape, if set. In that case we have a mix of calculating the background shape and afterwards compensating for inserts.

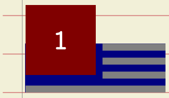
- A third case is tabulation and tables where we have dedicated regions to deal with. When these float we need to make sure that the backgrounds are adapted to the where they end up.

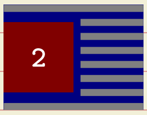
- Yet another case is in columns, where we hape multiple regions to deal with.

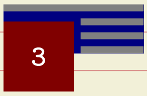
- As mentioned, floats need special treatment and they can be part of the page flow but also end up left or right of the text (either or not shifted) but also in the margins, edges, back- or cutspace. Their placement influences the way backgrounds are calculated so additional information needs to travel with them.

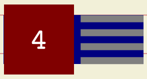
We distinguish between a paragraph background, which runs between the left and right skip areas and a text background which follows a shape. In figure 3.5 we see a test case with several such shapes.

In the case of side floats the following cases occur. Of course multiple such cases can follow each order so in practice we have to deal with an accumulation.

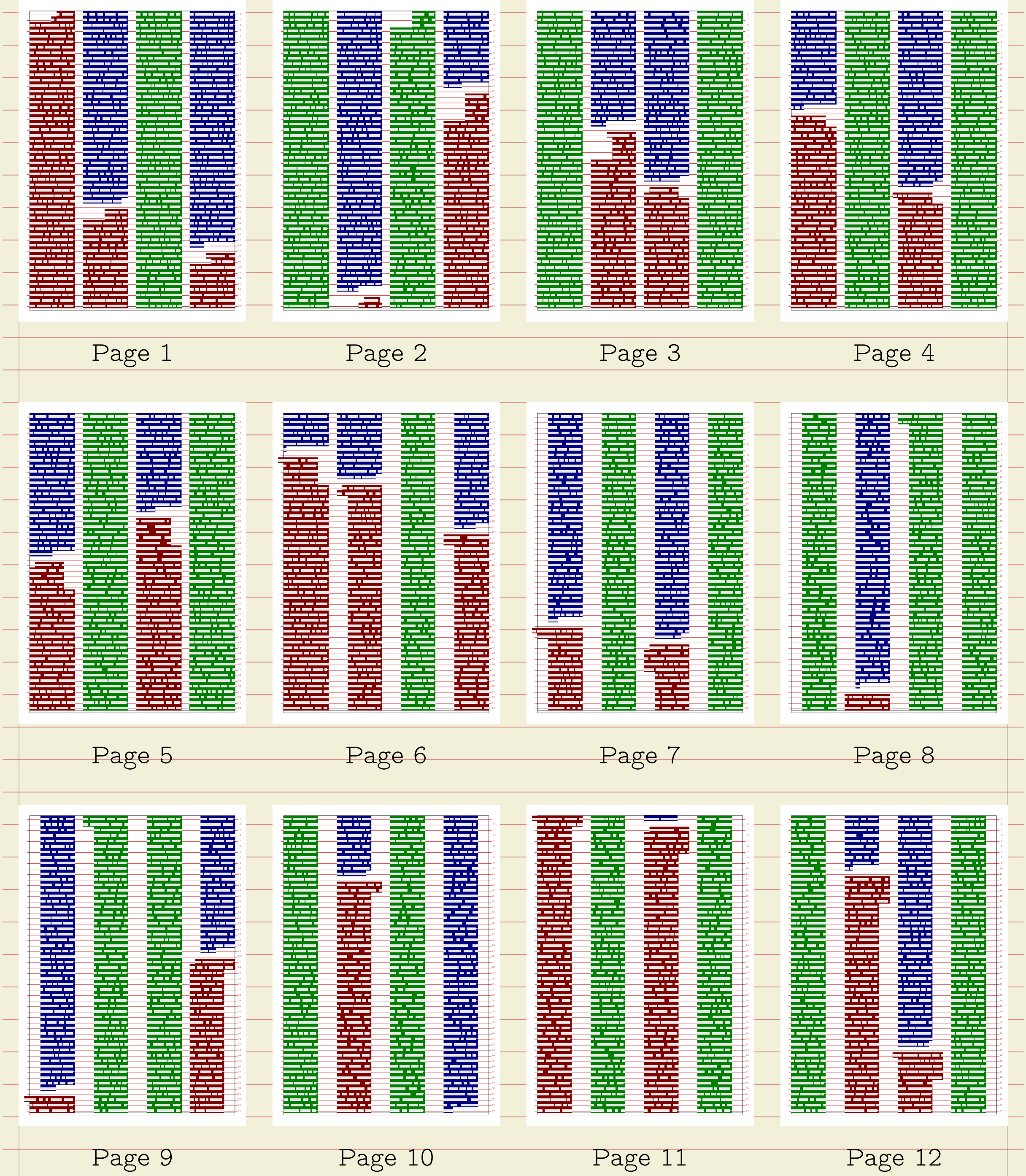








As often in TeX coming up with a solution is not a the problem but interference is. You can cook up a solution for one case that fails in another. Backgrounds fall into this category, as do side floats. In the next pages we will demonstrate a few cases. In practice you can probably always come up with something that works out well, but in an automated workflow (like unattended xml to pdf conversion) you can best play safe. We show some examples on the next pages.


Page 1 Page 2 Page 3 Page 4

Page 5 Page 6 Page 7 Page 8

Page 9 Page 10 Page 11 Page 12

Figure 3.5

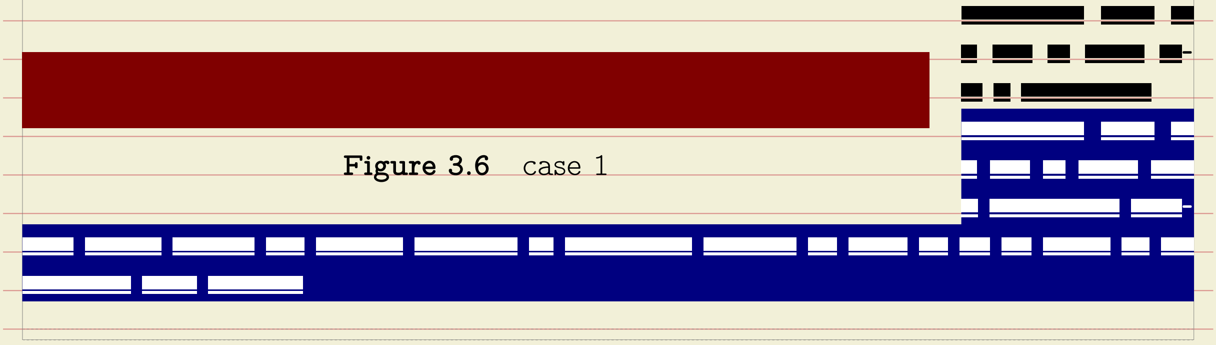


Figure 3.6 case 1

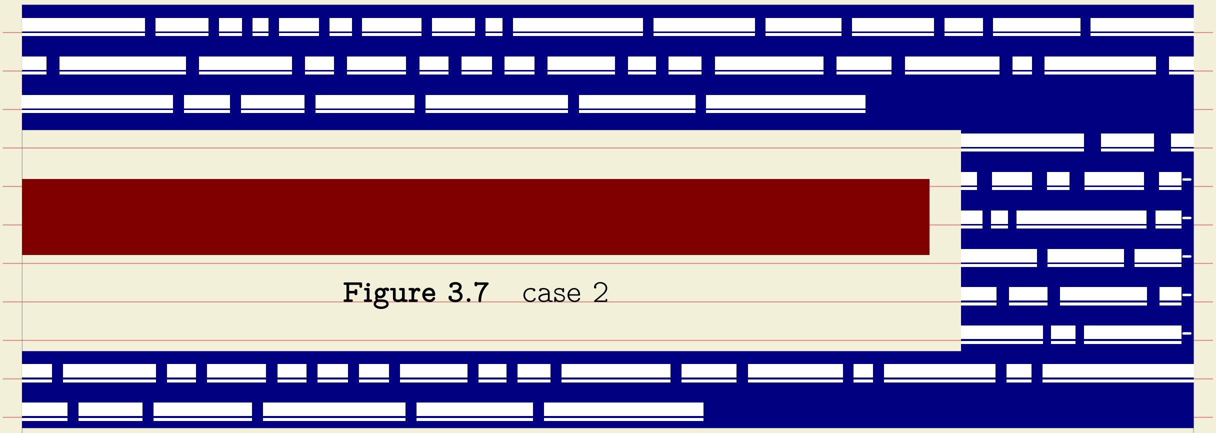


Figure 3.7 case 2

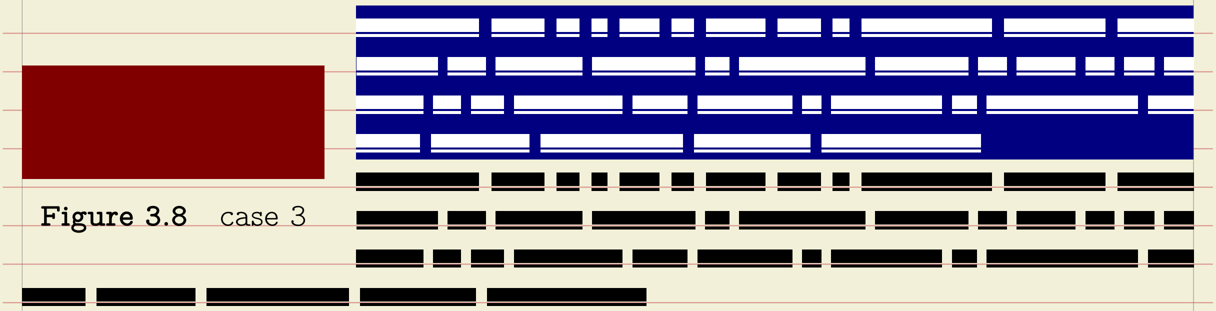


Figure 3.8 case 3

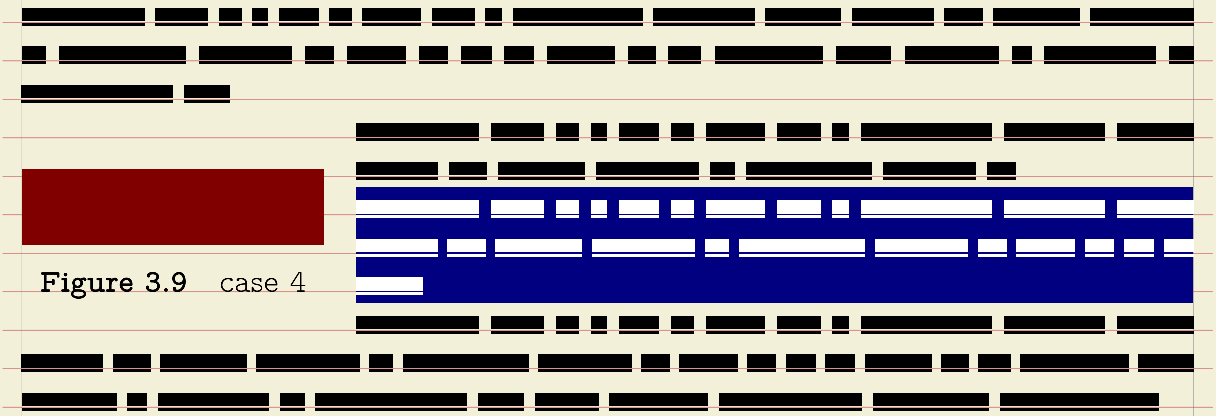


Figure 3.9 case 4

The previous examples were typeset with:

```
\placefigure
  [left]
  {case 1}
  {\blackrule[width=12cm,height=1cm,color=red]}
\simulatewords[demo][n=10]
\starttextbackground[demobg]
    \simulatewords[demo][n=30]
\stoptextbackground
\flushsidefloats

\blank

\starttextbackground[demobg]
```

```
    \simulatewords[demo][n=40]
    \placefigure
      [left]
      {case 2}
      {\blackrule[width=12cm,height=1cm,color=red]}
    \simulatewords[demo][n=40]
\stoptextbackground
\flushsidefloats

\blank

\placefigure
  [left]
  {case 3}
  {\blackrule[width=4cm,height=15mm,color=red]}
\starttextbackground[demobg]
    \simulatewords[demo][n=40]
\stoptextbackground
\simulatewords[demo][n=40]
\flushsidefloats

\blank

\simulatewords[demo][n=35]
\placefigure
  [left]
  {case 4}
  {\blackrule[width=4cm,height=1cm,color=red]}
\simulatewords[demo][n=20]
\starttextbackground[demobg]
    \simulatewords[demo][n=25]
\stoptextbackground
\simulatewords[demo][n=40]
\flushsidefloats

\blank
```

Regular (page flow) floats are a different story. Here we have the problem that a float might be postpones because there is no room on the current page and they are moved forward (which is why they're called float). Again we show some examples.

One problem introduced by the internet is that one can view music online. Well it's actually not really a problem as it is fun to do, but it does interfere with development of code: one can enter distraction mode quite easily.

Figure 3.10

One problem introduced by the internet is that one can view music online. Well it's actually not really a problem as it is fun to do, but it does interfere with development of code: one can enter distraction mode quite easily.

Figure 3.11

One problem introduced by the internet is that one can view music online. Well it's actually not really a problem as it is fun to do, but it does interfere with development of code: one can enter distraction mode quite easily.

Figure 3.12

One problem introduced by the internet is that one can view music online. Well it's actually not really a problem as it is fun to do, but it does interfere with development of code: one can enter distraction mode quite easily.

The input is:

```
\starttextbackground[demobg]
    \par \getbuffer[sample] \par
    \placefigure{}{\blackrule[width=4cm,height=1cm,color=red]}
    \par \getbuffer[sample] \par
    \placefigure{}{\blackrule[width=4cm,height=3cm,color=red]}
    \par \getbuffer[sample] \par
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \par \getbuffer[sample] \par
\stoptextbackground
```

A combination of both background avoiding mechanisms is shown on the next page (we flush a few more grapohics so that we cross a page boundary):

Figure 3.13

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day — and we humans are the cigarettes.

Figure 3.14

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day — and we humans are the cigarettes.

Figure 3.15

Figure 3.16

Figure 3.17

Figure 3.18

Figure 3.19

Figure 3.20

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day — and we humans are the cigarettes.

Figure 3.21

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several in fact. It would be happening whether humans had ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day — and we humans are the cigarettes.

This is the result from:

```
\starttextbackground[demobg]
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \par \input ward \par
    \placefigure[left]{}{\blackrule[width=4cm,height=2cm,color=red]}
    \par \input ward \par
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \par \input ward \par
    \placefigure{}{\blackrule[width=4cm,height=2cm,color=red]}
    \par \input ward \par
\stoptextbackground
```

You can control the interaction between backgrounds and floars with the `freeregion` parameter.

```
\starttextbackground[demobg]
    \simulatewords[demo][n=40]
    \startplacefigure
      [location=left,
       title={free}]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \startplacefigure
      [location=left,
       title={non|-|free},
       freeregion=no,
       color=textcolor]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \startplacefigure
      [location=here,
       title={free}]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \startplacefigure
      [location=here,
       title={non|-|free},
       freeregion=no,
       color=textcolor]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
\stoptextbackground
```

The next pages show the result, first with some tracing enabled sop that you can see what gets freed. This visual effect is enabled with:

```
\enabletrackers[floats.freeregion]
```

We now move to the next page.

Figure 3.22 free

Figure 3.23 non-free

Figure 3.24 free

Figure 3.25 non-free

Figure 3.26 free

Figure 3.27 non-free

Figure 3.28 free

Figure 3.29 non-free

We have some control over side float placement and of course that will interfere with backgrounds. Say that we have this:

```
\definefloat
  [demofigureleft]
  [figure]
  [default=left,
  margin=1cm,
   leftmargindistance=2cm,
   rightmargindistance=2cm]

\definefloat
  [demofigureright]
  [demofigureleft]
  [default=right]
```

Combined with the following we get the result on the next pages.

```
\starttextbackground[demobg]
    \startplacefloat[figure][location=left]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \blank
    \startplacefloat[figure][location=right]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \blank
    \startplacefloat[demofigureleft]
        \blackrule[width=10cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \blank
    \startplacefloat[demofigureright]
        \blackrule[width=10cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
    \startplacefloat[figure] % [freeregion=no]
        \blackrule[width=12cm,height=1cm,color=red]
    \stopplacefigure
    \simulatewords[demo][n=40]
\stoptextbackground
```

Figure 3.30

Figure 3.31

Figure 3.32

Figure 3.33

Figure 3.34

Figure 3.35

Figure 3.36

Figure 3.37

Figure 3.38

Figure 3.39

Because of its look and feel, a math formula can look too widely spaced when put on a grid. There are a few ways to control this. First of all, the default grid option bound to math is already more tolerant. But you can control it locally too. Take the following formula:

$$a = b^c$$

This has been entered as:

```
\startformula
  a = b^c
\stopformula
```

and because it is just a line of math it comes out as expected. The next code

```
\startformula
  a = \frac {a} {b}
\stopformula
```

produces a higher line:

$$a = \frac{a}{b}$$

as does:

```
\startformula
  a = \frac {\frac {b} {c}} {\frac {d} {e}}
\stopformula
```

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}$$

We will now demonstrate three ways to compensate fo rexcessive spacing. The first variant just sets a grid parameter:

```
\startformula[grid=math:-halfline]
    a = \frac {\frac {b} {c}} {\frac {d} {e}}
\stopformula
```

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}$$

You can also pass this as an option. Only a few such grid related options are defined: `halfline`, `line`, `-halfline` and `-grid`.

```
\startformula[-halfline]
    a = \frac {\frac {b} {c}} {\frac {d} {e}}
\stopformula
```

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}$$

If you need to compensate frequently you can consider defining an instance:

```
\defineformula[tight][grid=math:-halfline]

\starttightformula
    a = \frac {\frac {b} {c}} {\frac {d} {e}}
\stoptightformula
```

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}$$

The result can be somewhat unexpected at the top and bottom of a page. When we subtract half a line from the height we can end up above the text area. This is where the `split` directive comes in. So, the compensations are actually defined as

`math`	`maxdepth:1.05,maxheight:1.05,strut`
`math:line`	`maxdepth:1.05,maxheight:1.05,strut,line,split`
`math:halfline`	`maxdepth:1.05,maxheight:1.05,strut,halfline,split`
`math:-line`	`maxdepth:1.05,maxheight:1.05,strut,-line,split`
`math:-halfline`	`maxdepth:1.05,maxheight:1.05,strut,-halfline,split`

You can define your own variants building on top of an existing one:

```
\definegridsnapping[math:my][math,....]
```

We demonstrate the effect of the `split` directive here. It triggers a check at the page boundaries but you need to keep in mind that this is not always robust as such boundaries themselves can be triggered by and inject anything.

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 1 default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 2 default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 3 default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 4 default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 5 default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 6 default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 7 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 8 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 9 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 10 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 11 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 12 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 13 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 14 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 15 default})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 1 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 2 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 3 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 4 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 5 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 6 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 7 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 8 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(top 9 compensated)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 10 compensated})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 11 compensated})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 12 compensated})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 13 compensated})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 14 compensated})$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}(\text{top 15 compensated})$$

As said, the compensation is achieved with the `page` directive. The previous pages were rendered using:

```
\dorecurse {15} {
    \startformula[grid={math,-halfline}]
        a = \frac {\frac {b} {c}} {\frac {d} {e}}
        (\hbox{top #1 default})
    \stopformula
    \blank[samepage]
    \fakeline
}
```

and

```
\dorecurse {15} {
    \startformula[grid={math,-halfline,split}]
```

```
          a = \frac {\frac {b} {c}} {\frac {d} {e}}
          (\hbox{top #1 compensated})
      \stopformula
      \blank[samepage]
      \fakeline
   }
```

In order to get a consistent result we keep the depth of the formula the same but effectively shift it down a bit, still honouring the grid. So what about the bottom.

We can decide that the snapped formula doesn't fit and force a new page but we can also accept that it sticks out to the bottom, which is less worse than the top-of-the-page case.

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(bottom default)}$$

$$a = \frac{\frac{b}{c}}{\frac{d}{e}}\text{(bottom compensated)}$$

These mechanisms might be improved over time but as we don't use it frequently that might take a while.

The following formula was posted at the ConTEXt mailing list in a grid snapping thread and we will use it to demonstrate how you can mess a bit with the snapping.

```
g(x_{*}) = \lim_{n\to\infty} g(a_{n}) \leq 0 \leq \lim_{n\to\infty}
g(b_{n}) = g(x_{*})
```

We show the given grid parameter as well as its expansion into the low level grid directives.

`grid=math` expanded: `maxdepth:1.05,maxheight:1.05,strut`

$$g(x_*) = \lim_{n\to\infty} g(a_n) \leq 0 \leq \lim_{n\to\infty} g(b_n) = g(x_*)$$

`grid=low,halfline` expanded: `maxheight,mindepth,none,halfline`

$$g(x_*) = \lim_{n\to\infty} g(a_n) \leq 0 \leq \lim_{n\to\infty} g(b_n) = g(x_*)$$

`grid=math,nodepth` expanded: `maxdepth:1.05,maxheight:1.05,strut,nodepth`

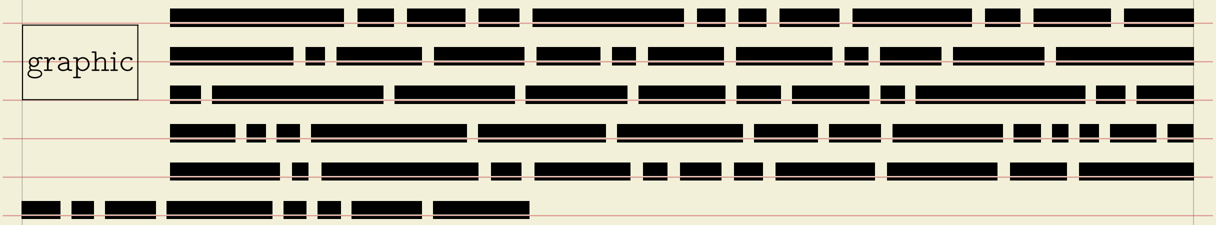
$$g(x_*) = \lim_{n\to\infty} g(a_n) \leq 0 \leq \lim_{n\to\infty} g(b_n) = g(x_*)$$

Graphics, tables and alike are often treated as floating bodies. This means that when such a body does not fit on the current page, it will be moved to the next one. In the examples we will use figures, but much of what we demonstrate here applies to all floats.

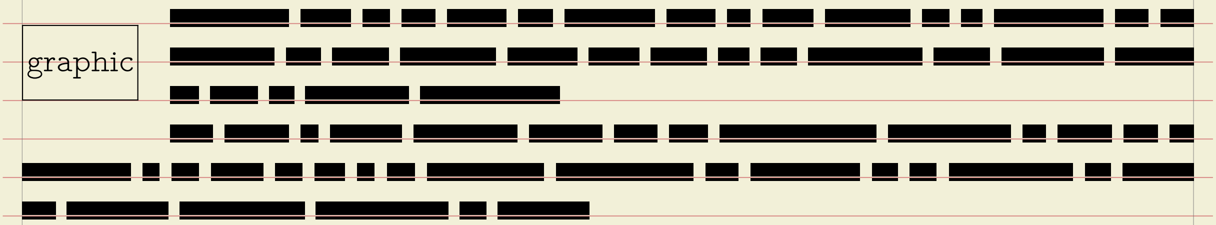
A side float is a float which placement one way or another depends on the text that follows it. In its simplest form, the text flows around it, for instance in:

```
\placefigure[left,none]{caption}{\framed[height=1cm]{graphic}}
```

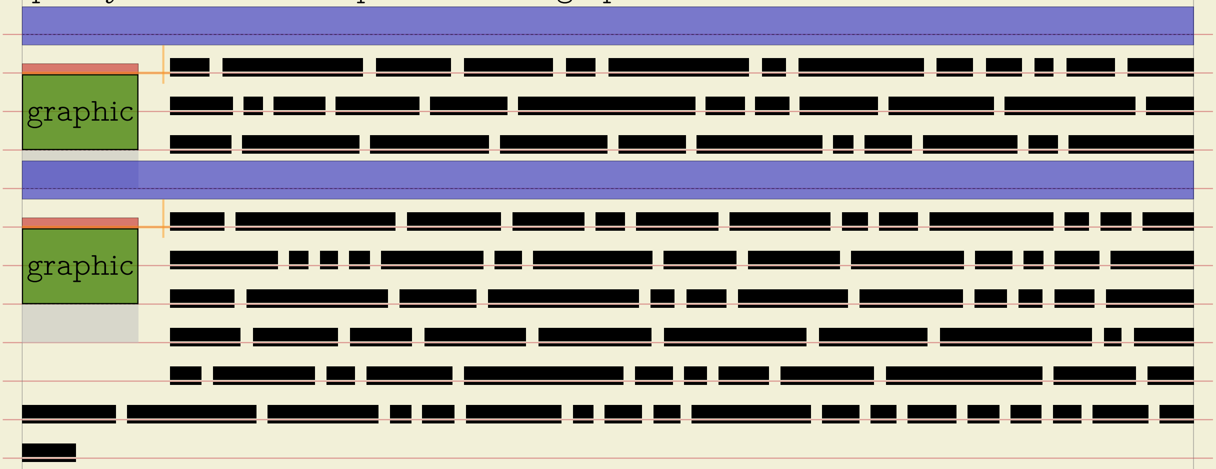
The first keyword of such a call is treated as a placement directive, so this figure will be placed left. The `none` directive nils the caption.



When the figure does not fit on the page, a page break is issued. A figure can span multiple paragraphs. When a next graphic is placed the previous figure will be padded if needed. First an example of multiple paragraphs.



Multiple floats in a row will lead to padding. The amount of padding is a combination of empty lines and the normal white space following the float. The visual quality of the result depends on the graphic itself.



Here we show the baseline of the first paragraph after the float as well as the filler. The whitespace around a graphic also depends on the inter-paragraph whitespace. As with many automated mechanisms, compromises are made. A one point smaller figure may result in an extra empty line.

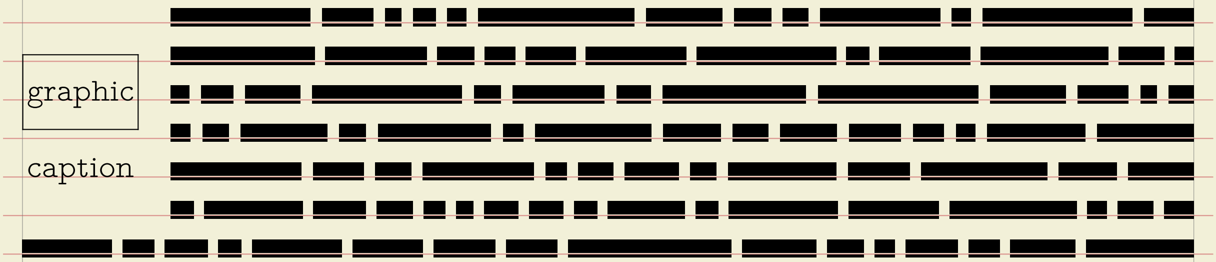
Later we will demonstrate a lot of tuning options, but first we give a few more examples. Most of the tuning options can be driven by keywords as well as (global) settings.

```
\placefigure
  [left,nonumber]
  {caption} {\framed[height=1cm]{graphic}}
```

The `nonumber` keyword suppresses the label and figure number. You can do this for all figures with

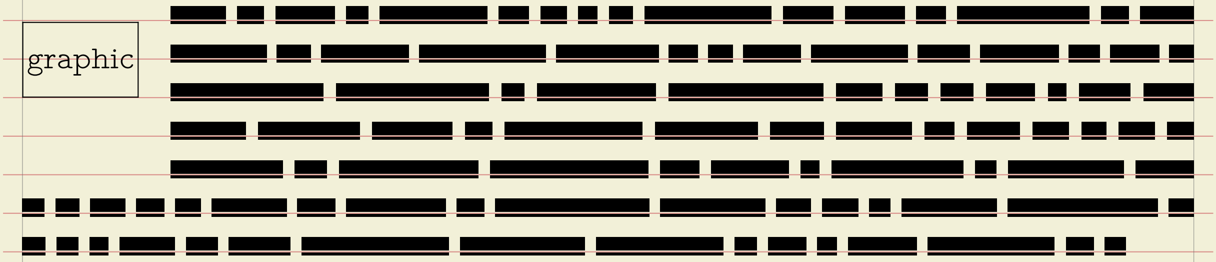
```
\setupcaption[figure][number=no]
```

The previous placement command results in the following side float.



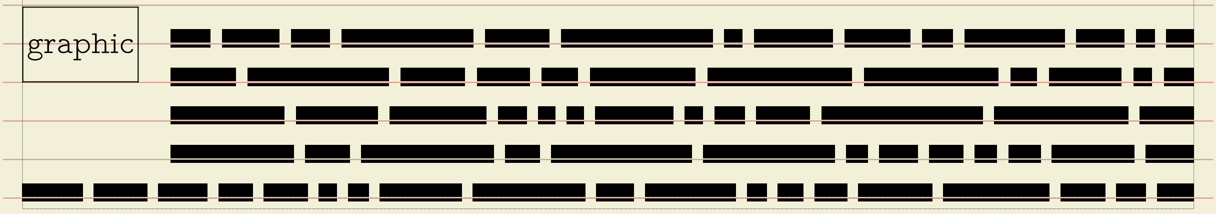
Another handy keyword is `none`.

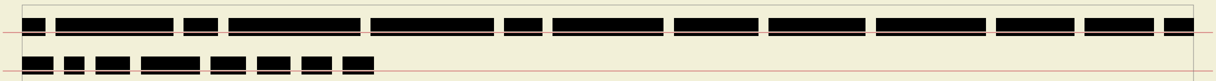
```
\placefigure
  [left,none]{quoting knuth}
  {\framed[height=1cm]{graphic}}
```



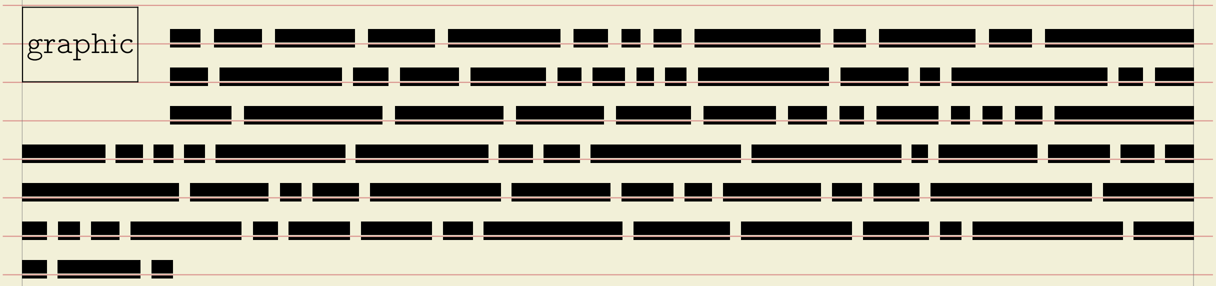
Control over spacing is exercised by means of the keywords `high`, `low` and `fit`.

```
\placefigure
  [left,none,high]{}
  {\framed[height=1cm]{graphic}}
```

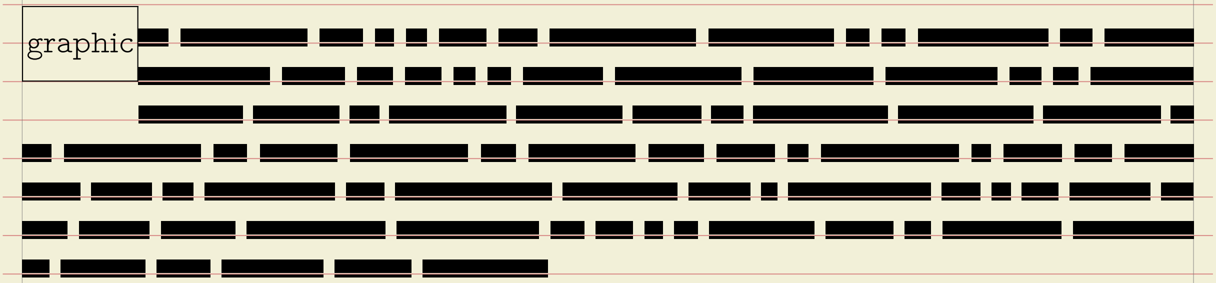




```
\placefigure
  [left,none,high,low]{}
  {\framed[height=1cm]{graphic}}
```



```
\placefigure
  [left,none,fit]{}
  {\framed[height=1cm]{graphic}}
```

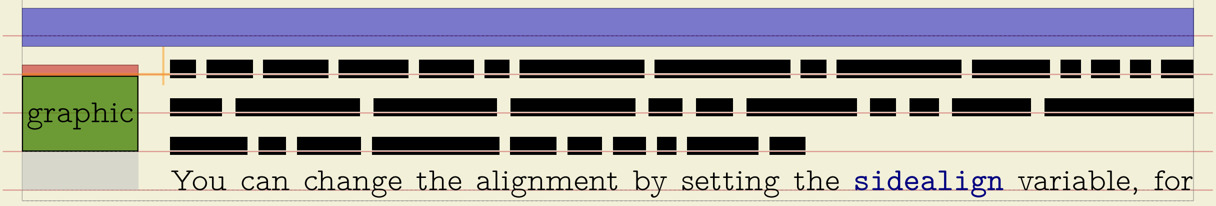


In the examples so far, we saw additional spacing around the graphic. We will now (for a while) disable the surrounding whitespace.

```
\setupfloat
  [figure]
  [sidespacebefore=none,
   sidespaceafter=none]
```

With these settings a simple left placement looks as follows. The top of the side float aligns with the maximum height of a line.

```
\placefigure
  [left,none]
  {} {\framed[height=1cm]{graphic}}
```



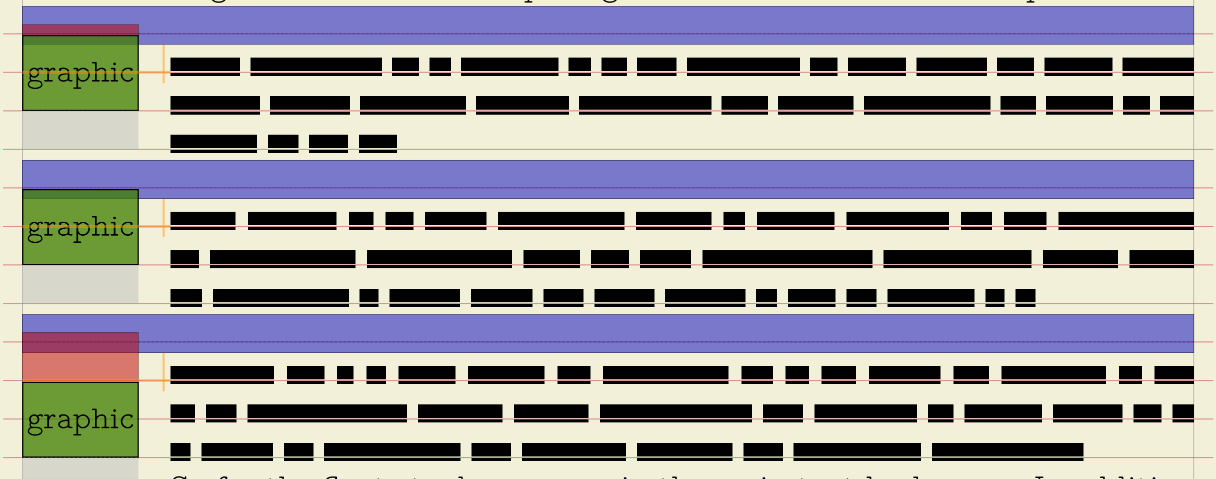
You can change the alignment by setting the `sidealign` variable, for instance:

```
\setupfloat
  [figure]
  [sidealign=line]
```

The three keywords `height`, `line` and `depth` can also be passed directly:

```
\placefigure
  [left,none,height]{}
  {\framed[height=1cm]{graphic}}
```

The three alignments disable the spacing before the float and show up as follows.

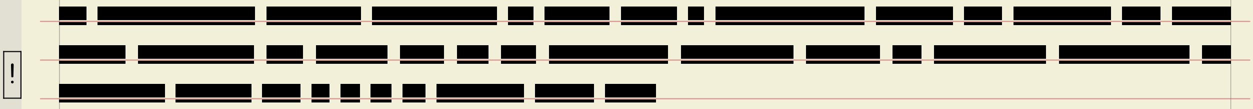


So far the floats took up space in the main text body area. In addition to the `left` (or `right`) directive we can use `inner` or `outer` to force left or right placement depending in the spread.

Instead of spoiling paper in the text areas, we can use the margin and edges: `leftmargin` and `leftedge`, `rightmargin` and `rightedge`, but also `innermargin` and `outermargin`, `inneredge` and `outeredge`.

The next couple of pages we will highlight the margins and edges so that we can see what happens.

```
\placefigure
  [leftmargin,none]
  {} {\framed{!}}
```



```
\placefigure
  [leftmargin,none]
  {} {\framed[width=1cm]{!}}
```

!

```
\placefigure
  [leftmargin,none]
  {} {\framed[width=1.5cm]{!}}
```

!

The placement directives can be combined with setting distance and width parameters, thereby not only opening a world of possibilities, but also creating confusion. Therefore, we will illustrate these features by cloning floats.

```
\definefloat
  [marginfigure]
  [figure]

\setupfloat
  [marginfigure]
  [leftmargindistance=-\leftmargintotal,
   default={left,none,low}]
```

The definition command clones figure into a new class of figures. There are two ways to use such a float :

```
\placefloat
  [marginfigure]
  {} {\framed[width=1.5cm]{!}}
```

or directly:

```
\placemarginfigure
  {} {\framed[width=1.5cm]{!}}
```

Both placement calls will result in a figure sticking into the margin.

!

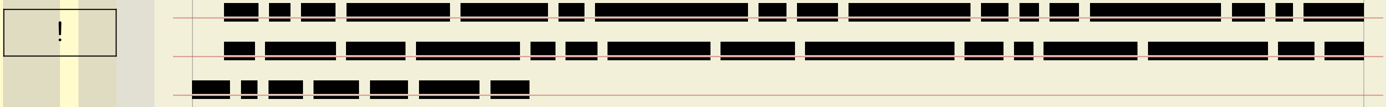
By manipulating the margin distance, you can align graphics to vertical grid lines, like the edge:

```
\definefloat
  [edgefigure]
  [figure]

\setupfloat
```

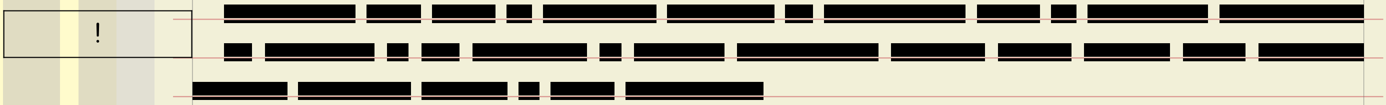
```
    [edgefigure]
    [leftmargindistance=-\innercombitotal,
     default={left,none,low,high}]
```

The `\innercombitotal` is one of the many available dimensions. This measure is the combined width of the margin and edge.

```
\placeedgefigure
  {} {\framed[width=1.5cm]{!}}
```



```
\placeedgefigure
  {} {\framed[width=\innercombitotal]{!}}
```



You need to be aware of the fact that the margins and edges are not related to the backspace and cut space settings. When you set up a layout, you need to think of the right page as starting point. In a double sided layout, the margins are swapped in the page composition stage. Unless you explicitly go to a left or right page, you don't know if your left margin will be swapped or not.
For this reason ConTEXt provides the inner and outer margin/edge dimensions. These are automatically synchronized when the float is constructed. So, if you want to automatically adapt the float placement and width to the current left margin in a double sided document, you can use the inner dimensions.

dimension	left page	right page
`\outermarginwidth`	`\leftmarginwidth`	`\rightmarginwidth`
`\innermarginwidth`	`\rightmarginwidth`	`\leftmarginwidth`
`\outermargindistance`	`\leftmargindistance`	`\rightmargindistance`
`\innermargindistance`	`\rightmargindistance`	`\leftmargindistance`

Similar dimensions are available for the edges. You can save yourself some calculations by using the following dimensions:

`\leftmargintotal`	left margin width	+	left margin distance
`\rightmargintotal`	right margin width	+	right margin distance
`\innermargintotal`	inner margin width	+	inner margin distance
`\outermargintotal`	outer margin width	+	outer margin distance

As you may expect, the edge totals are available as well, which leave a few more totals, namely the combinations of margin and edge.

`\leftsidetotal`	left margin width +left edge total
`\rightsidetotal`	right margin width +right edge total
`\innersidetotal`	inner margin width+inner edge total
`\outersidetotal`	outer margin width+outer edge total
`\leftcombitotal`	left margin total +left edge total
`\rightcombitotal`	right margin total +right edge total
`\innercombitotal`	inner margin total +inner edge total
`\outercombitotal`	outer margin total +outer edge total

Adaptive back- and cutspace dimensions are also available:

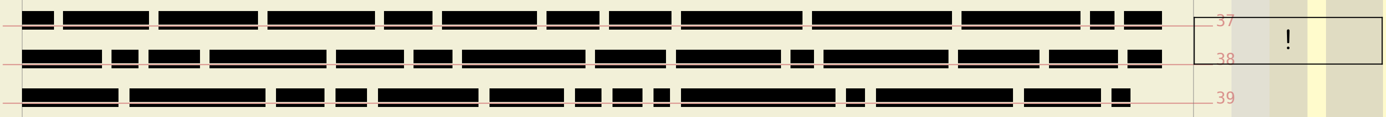
`\innerspacewidth`	adaptive backspace
`\outerspacewidth`	adaptive cutspace

There is one drawback in using the inner and outer dimensions: if you also change the height of the float dynamically, you may end up in a kind of loop because a page break may occur at a non–expected place.
While negative values move float into the margin, positive values will move the float into the text. It will be of no surprise that you can also set the right margin distance. Keep in mind that this distance is not related to the text margin, but to the float margin.

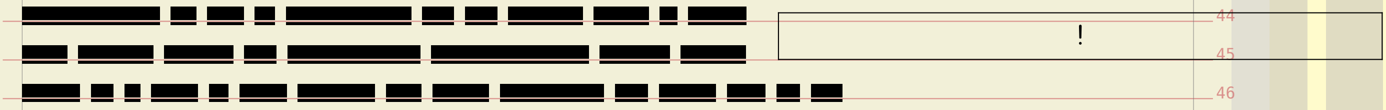
```
\setupfloat
  [edgefigure]
  [leftmargindistance=-\outercombitotal,
   rightmargindistance=-\outercombitotal,
   default={outer,none,low,high}]
```

The locations `inner` and `outer` change with the left or right page.

```
\placeedgefigure
  {} {\framed[width=\outercombitotal]{!}}
```

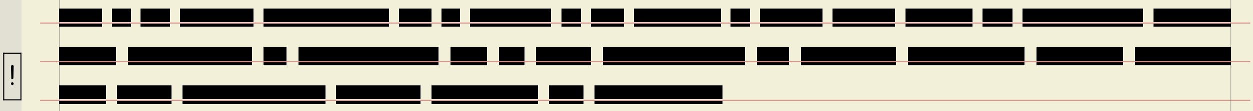


```
\placeedgefigure
  {} {\framed[width=8cm]{!}}
```

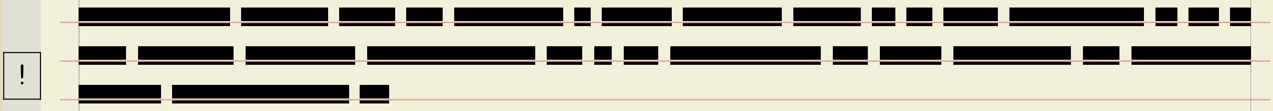


As a result of manipulating the floats margin settings, the side floats can start in the margin (or edge). You should not confuse this with margin floats, i.e. side floats that are explicitly placed in the margins.

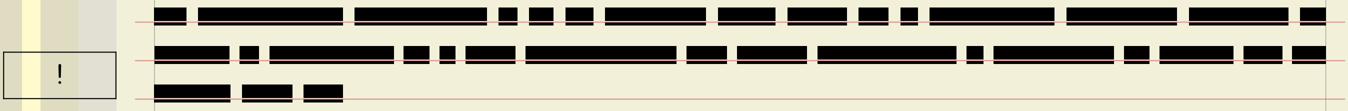
```
\placefigure[leftmargin,none]
  {} {\framed{!}}
```



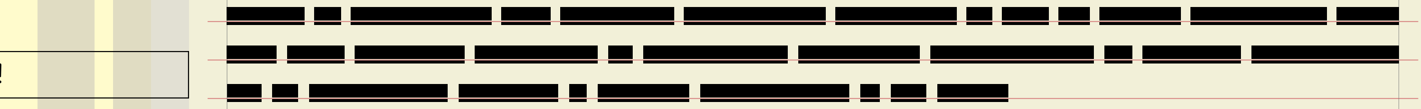
```
\placefigure[leftmargin,none]
  {} {\framed[width=.5cm]{!}}
```



```
\placefigure[leftmargin,none]
  {} {\framed[width=1.5cm]{!}}
```

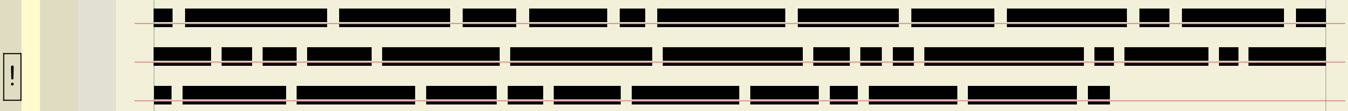


```
\placefigure[leftmargin,none]
  {} {\framed[width=5cm]{!}}
```



The margin side floats align to the margin and the edge floats to the edge. This way you can create bleeding figures.

```
\placefigure[leftedge,none]
  {} {\framed{!}}
```



There are situations where you don't know the dimensions in advance. In order to prevent unwanted side effects, for instance part of a graphic disappearing outside the page boundary, ConTEXt provides a few options. The most crude one is setting the `criterium`, as in:

```
\setupfloat
  [figure]
  [criterium=.25\textwidth]
```

This will automatically turn figures that are wider than 25% of the text width into normal floats instead of side floats. But let's not fall back on that feature now.

You can use maxwidth and minwidth variables to control the placement in more detail. The exact result depends on the settings of location. By default we center, but you can set the location to left or right to achieve a different alignment.

```
\definefloat
  [midmarginfigure]
  [figure]

\setupfloat
  [midmarginfigure]
  [minwidth=\leftmarginwidth,
   default={leftmargin,none}]
```

You can use maxwidth and minwidth variables to control the placement in more detail. The exact result depends on the settings of location. By default we center, but you can set the location to left or right to achieve a different alignment.

```
\placemidmarginfigure
  {} {\framed[width=1.5cm]{!}}
```

!

The meaning of maxwidth depends on the kind of float. First we place a left float with a width smaller than maxwidth.

```
\setupfloat[figure][maxwidth=2cm]

\placefigure[left,none]{}{\framed[width=1cm]{!}}
```

!

When the width exceeds the maxwidth, the float will be centered. This is because we have no reference alignment point.

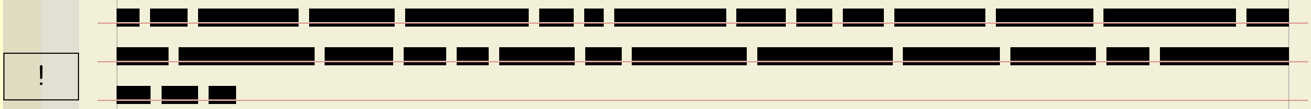
```
\placefigure[left,none]{}{\framed[width=5cm]{!}}
```

!

In margin floats, the maxwidth settings have a different result. First we place a small graphic.

```
\setupfloat[figure][maxwidth=\leftmarginwidth]
```

```
\placefigure[leftmargin,none]{}{\framed[width=1cm]{!}}
```

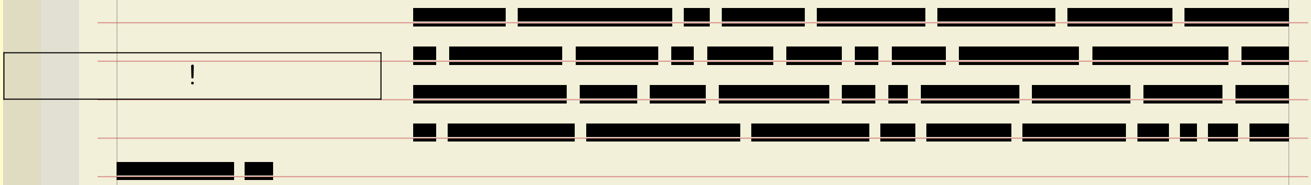


Because the left and right margin of this document are the same —the edges differ— we don't need to use inner and outer dimensions.

```
\setupfloat[figure][maxwidth=\leftmarginwidth]
```

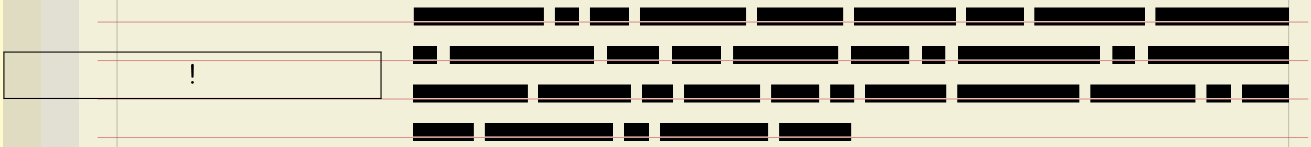
A wider than `maxwidth` graphic will behave like a mixture of a margin and text side float. Watch how we align the float to the margin.

```
\placefigure[leftmargin,none]{}{\framed[width=5cm]{!}}
```

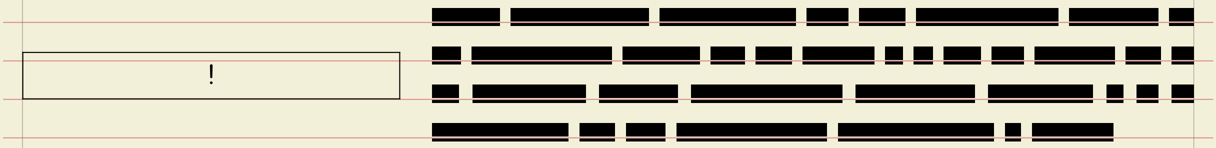


Instead of setting the width you can give `hanging` a try. The next examples demonstrate this.

```
\placefigure[leftmargin,hanging,none]{}{\framed[width=5cm]{!}}
```

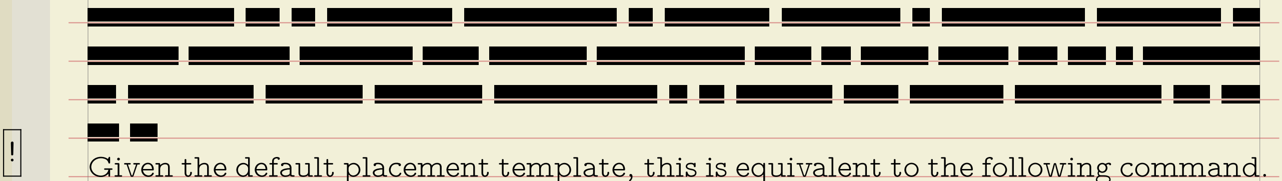


```
\placefigure[left,hanging,none]{}{\framed[width=5cm]{!}}
```



You can move down/up margin floats with the `\movesidefloat` macro. Such shifts come in handy when you have multiple side floats near to each other.

```
\movesidefloat [+2*line]
\placemidmarginfigure {} {\framed{!}}
```



Given the default placement template, this is equivalent to the following command. Watch out, a simple `line` has a different effect (alignment).

```
\placemidmarginfigure
  [leftmargin,none,+2*line]
  {} {\framed{!}}
```

Another nice keyword is `long`:

```
\placefigure
  [leftmargin,none,long]
  {} {\framed[height=2cm,width=2cm]{!}}

Watch how we move down. The effect is that we skip over the margin
figure.

\placefigure
  [leftmargin,none]
  {} {\framed[height=1cm,width=2cm]{!}}
```

Watch how we move down. The effect is that we skip over the margin figure.

!

!

```
\placefigure
  [leftmargin,none]
  {} {\framed[height=2cm,width=2cm]{!}}

Do we clash or not?

\placefigure
  [leftmargin,none]
  {} {\framed[height=2cm,width=2cm]{!}}

Did we clash or not?
```

Do we clash or not?

!

!

Did we clash or not?

There are a few macros that can be of help with solving clashes in side floats:

`\flushsidefloats`	This macro moves down as much as is needed to separate the side floats of each other.
`\forgetsidefloats`	this macro kind of forgets that a side float is in progress.

Use these macros with care. If you change the dimensions of the graphic and/or content involved, reconsider the use of these directives.

The next couple of spreads we will demonstrate some example definitions. These placements are taken from one of the styles we made for typesetting a series of school math books which illustrations and tables all over the pages.

First we fine tune the spacing around side floats and verbatim text.

```
\setupfloats
  [sidespacebefore=none,
   sidespaceafter=depth]

\setuptyping
  [margin=]
```

The placements have rather verbose names. In this case the word ‘edge’ is used to identify bleeding floats (with an cut–off margin of 3mm). The ‘text’ floats are side floats positioned in the main text flow.

```
\setupfloats
  [sidespacebefore=none,
   sidespaceafter=depth]

\setuptyping
  [margin=]
```

Watch how we define fall backs for too wide content (`criterium` as well as use `maxwidth` to manipulate the placement of content that falls off the margins.

The black rules are set up with:

```
\setupblackrules[color=tred,depth=0pt,height=1.5cm]
```

```
\setupfloat
  [marginfigure]
  [criterium=.5\textwidth,
   maxwidth=\rightmarginwidth,
   default={outermargin,none}]
```

```
\placemarginfigure{}{\blackrule[width=.25cm]}
```

```
\placemarginfigure{}{\blackrule[width=.5cm]}
```

```
\placemarginfigure{}{\blackrule[width=1cm]}
```

```
\placemarginfigure{}{\blackrule[width=2cm]}
```

```
\placemarginfigure{}{\blackrule[width=4cm]}
```

```
\placemarginfigure{}{\blackrule[width=8cm]}
```

```
\placemarginfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [marginfigure]
  [criterium=.5\textwidth,
   maxwidth=\rightmarginwidth,
   default={outermargin,none}]


\placemarginfigure{}{\blackrule[width=.25cm]}


\placemarginfigure{}{\blackrule[width=.5cm]}


\placemarginfigure{}{\blackrule[width=1cm]}


\placemarginfigure{}{\blackrule[width=2cm]}


\placemarginfigure{}{\blackrule[width=4cm]}


\placemarginfigure{}{\blackrule[width=8cm]}


\placemarginfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [middlemarginfigure]
  [minwidth=\rightmarginwidth,
   criterium=\backspace,
   location=middle,
   default={outermargin,none}]
```

```
\placemiddlemarginfigure{}{\blackrule[width=.25cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=.5cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=1cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=2cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=4cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=8cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [middlemarginfigure]
  [minwidth=\rightmarginwidth,
   criterium=\backspace,
   location=middle,
   default={outermargin,none}]
```

```
\placemiddlemarginfigure{}{\blackrule[width=.25cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=.5cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=1cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=2cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=4cm]}
```

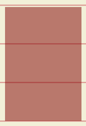
```
\placemiddlemarginfigure{}{\blackrule[width=8cm]}
```

```
\placemiddlemarginfigure{}{\blackrule[width=16cm]}
```

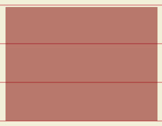
```
\setupfloat
  [middlefigure]
  [default={here,none}]
```

```
\placemiddlefigure{}{\blackrule[width=.25cm]}
```

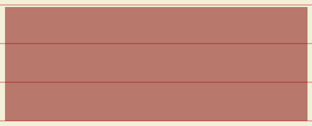
```
\placemiddlefigure{}{\blackrule[width=.5cm]}
```



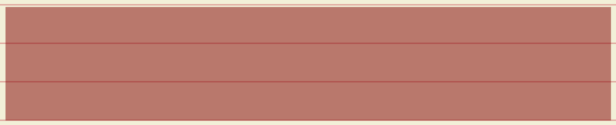
```
\placemiddlefigure{}{\blackrule[width=1cm]}
```



```
\placemiddlefigure{}{\blackrule[width=2cm]}
```



```
\placemiddlefigure{}{\blackrule[width=4cm]}
```

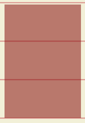


```
\placemiddlefigure{}{\blackrule[width=8cm]}
```

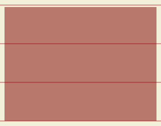
```
\setupfloat
  [middlefigure]
  [default={here,none}]
```

```
\placemiddlefigure{}{\blackrule[width=.25cm]}
```

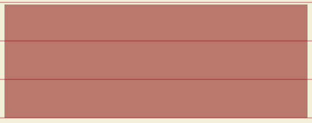
```
\placemiddlefigure{}{\blackrule[width=.5cm]}
```



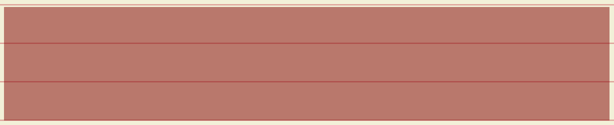
```
\placemiddlefigure{}{\blackrule[width=1cm]}
```



```
\placemiddlefigure{}{\blackrule[width=2cm]}
```



```
\placemiddlefigure{}{\blackrule[width=4cm]}
```



```
\placemiddlefigure{}{\blackrule[width=8cm]}
```

```
\setupfloat
   [textfigure]
   [criterium=.5\textwidth,
    default={outer,none}]
```

```
\placetextfigure{}{\blackrule[width=.25cm]}
```

```
\placetextfigure{}{\blackrule[width=.5cm]}
```

```
\placetextfigure{}{\blackrule[width=1cm]}
```

```
\placetextfigure{}{\blackrule[width=2cm]}
```

```
\placetextfigure{}{\blackrule[width=4cm]}
```

```
\placetextfigure{}{\blackrule[width=8cm]}
```

```
\placetextfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [textfigure]
  [criterium=.5\textwidth,
   default={outer,none}]
```

```
\placetextfigure{}{\blackrule[width=.25cm]}
```

```
\placetextfigure{}{\blackrule[width=.5cm]}
```

```
\placetextfigure{}{\blackrule[width=1cm]}
```

```
\placetextfigure{}{\blackrule[width=2cm]}
```

```
\placetextfigure{}{\blackrule[width=4cm]}
```

```
\placetextfigure{}{\blackrule[width=8cm]}
```

```
\placetextfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [leftfigure]
  [criterium=.5\textwidth,
   default={left,none}]
```

```
\placeleftfigure{}{\blackrule[width=.25cm]}
```

```
\placeleftfigure{}{\blackrule[width=.5cm]}
```

```
\placeleftfigure{}{\blackrule[width=1cm]}
```

```
\placeleftfigure{}{\blackrule[width=2cm]}
```

```
\placeleftfigure{}{\blackrule[width=4cm]}
```

```
\placeleftfigure{}{\blackrule[width=8cm]}
```

```
\placeleftfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [leftfigure]
  [criterium=.5\textwidth,
   default={left,none}]
```

```
\placeleftfigure{}{\blackrule[width=.25cm]}
```

```
\placeleftfigure{}{\blackrule[width=.5cm]}
```

```
\placeleftfigure{}{\blackrule[width=1cm]}
```

```
\placeleftfigure{}{\blackrule[width=2cm]}
```

```
\placeleftfigure{}{\blackrule[width=4cm]}
```

```
\placeleftfigure{}{\blackrule[width=8cm]}
```

```
\placeleftfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [rightfigure]
  [criterium=.5\textwidth,
   default={right,none}]

\placerightfigure{}{\blackrule[width=.25cm]}

\placerightfigure{}{\blackrule[width=.5cm]}

\placerightfigure{}{\blackrule[width=1cm]}

\placerightfigure{}{\blackrule[width=2cm]}

\placerightfigure{}{\blackrule[width=4cm]}

\placerightfigure{}{\blackrule[width=8cm]}

\placerightfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [rightfigure]
  [criterium=.5\textwidth,
   default={right,none}]
```

```
\placerightfigure{}{\blackrule[width=.25cm]}
```

```
\placerightfigure{}{\blackrule[width=.5cm]}
```

```
\placerightfigure{}{\blackrule[width=1cm]}
```

```
\placerightfigure{}{\blackrule[width=2cm]}
```

```
\placerightfigure{}{\blackrule[width=4cm]}
```

```
\placerightfigure{}{\blackrule[width=8cm]}
```

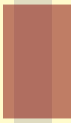
```
\placerightfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [bleedfigure]
  [criterium=.5\textwidth,
   leftmargindistance=-1mm,
   rightmargindistance=-1mm,
   default={backspace,none}]
```

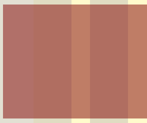
```
\placebleedfigure{}{\blackrule[width=.25cm]}
```

```
\placebleedfigure{}{\blackrule[width=.5cm]}
```

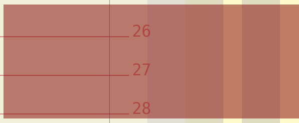
```
\placebleedfigure{}{\blackrule[width=1cm]}
```

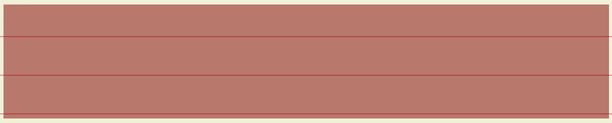


```
\placebleedfigure{}{\blackrule[width=2cm]}
```

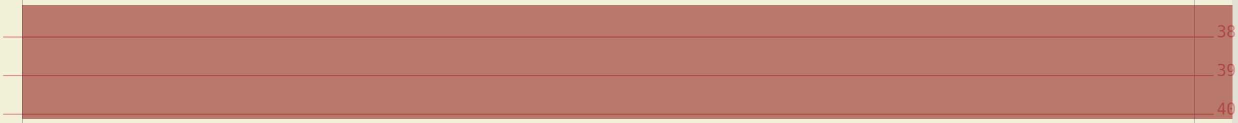


```
\placebleedfigure{}{\blackrule[width=4cm]}
```





```
\placebleedfigure{}{\blackrule[width=8cm]}
```



```
\placebleedfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [bleedfigure]
  [criterium=.5\textwidth,
   leftmargindistance=-1mm,
   rightmargindistance=-1mm,
   default={backspace,none}]
```

```
\placebleedfigure{}{\blackrule[width=.25cm]}
```

```
\placebleedfigure{}{\blackrule[width=.5cm]}
```

```
\placebleedfigure{}{\blackrule[width=1cm]}
```

```
\placebleedfigure{}{\blackrule[width=2cm]}
```

```
\placebleedfigure{}{\blackrule[width=4cm]}
```

```
\placebleedfigure{}{\blackrule[width=8cm]}
```

```
\placebleedfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [bleedfigure]
  [criterium=.5\textwidth,
   leftmargindistance=-1mm,
   rightmargindistance=-1mm,
   default={cutspace,none}]
```

```
\placebleedfigure{}{\blackrule[width=.25cm]}
```

```
\placebleedfigure{}{\blackrule[width=.5cm]}
```

```
\placebleedfigure{}{\blackrule[width=1cm]}
```

```
\placebleedfigure{}{\blackrule[width=2cm]}
```

```
\placebleedfigure{}{\blackrule[width=4cm]}
```

```
\placebleedfigure{}{\blackrule[width=8cm]}
```

```
\placebleedfigure{}{\blackrule[width=16cm]}
```

```
\setupfloat
  [bleedfigure]
  [criterium=.5\textwidth,
   leftmargindistance=-1mm,
   rightmargindistance=-1mm,
   default={cutspace,none}]


\placebleedfigure{}{\blackrule[width=.25cm]}



\placebleedfigure{}{\blackrule[width=.5cm]}



\placebleedfigure{}{\blackrule[width=1cm]}



\placebleedfigure{}{\blackrule[width=2cm]}



\placebleedfigure{}{\blackrule[width=4cm]}
```

```
\placebleedfigure{}{\blackrule[width=8cm]}
```

```
\placebleedfigure{}{\blackrule[width=16cm]}
```

At ConTEXt and BachoTEXmeetings it is now a tradition that Harald König and I spend some time on figuring out what happens with border cases and interfences with user intervention. As it's hard to nail down I decided to add some more tracing and control. So, the remainder of this chapter is dedicated to Harald.

We will now demonstrate some features in a way that makes it possible to compare to the simple default case. Options can be passed as keywords:

```
\placefigure
  [left,...]
  [fig:whatever]
  {caption}
  {content}
```

or as settings:

```
\startplacefigure
  [default={left,...},
   title=caption,
   reference=fig:whatever]

    content

\stopplacefigure
```

It is important to realize that all that spacing can interfere with additional hard coded corrections at the users end. We don't show the effects of `sidespacebefore` and `sidespaceafter`, the two general vertical spacing hooks. These are currently set to `big` and `big` respectively. The `sidealign` parameter is always winning from a keyword doing the same.

The last few examples demonstrate that you can define an instance. Often that's the best way to deal with special cases in a consistent way. For instance:

```
\definefloat
  [LeftTwo]
  [figure]

\setupfloat
  [LeftTwo]
  [default=left,
   sidealign=line]
```

First we show some keyword variant, next some parameter driven versions.

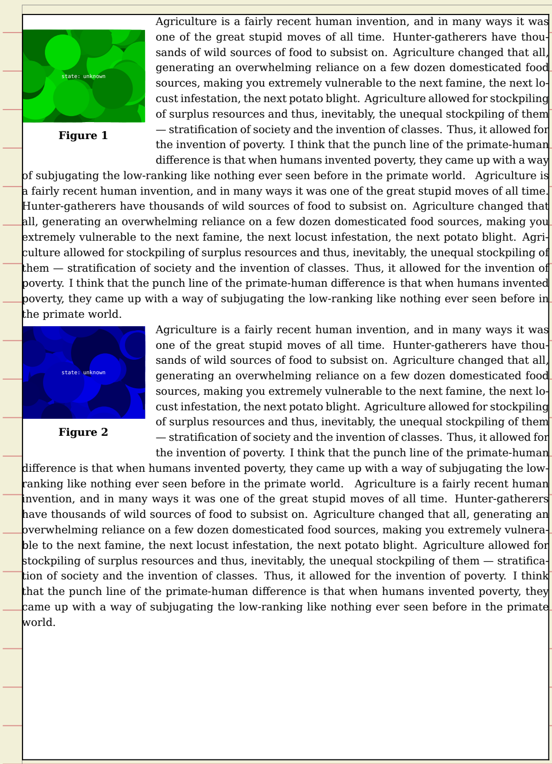


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

left,high

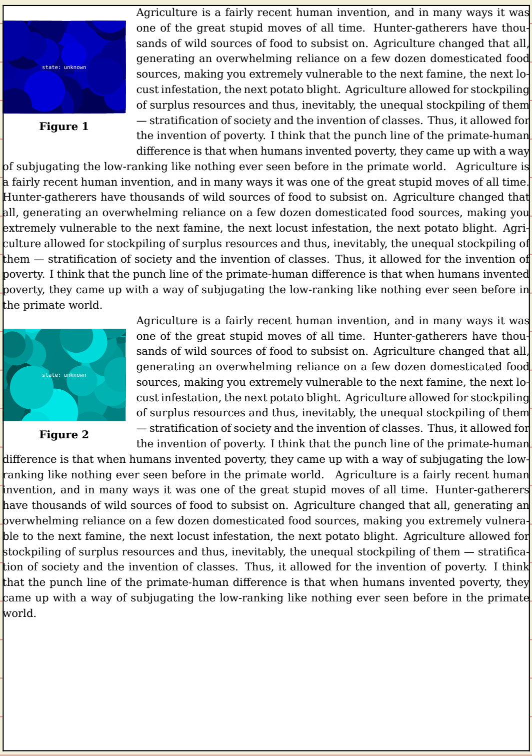


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

left,low

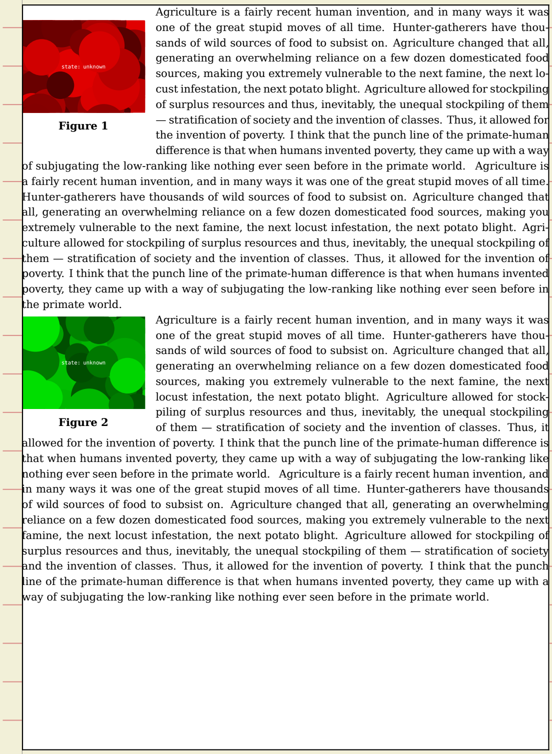


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

left,high,low

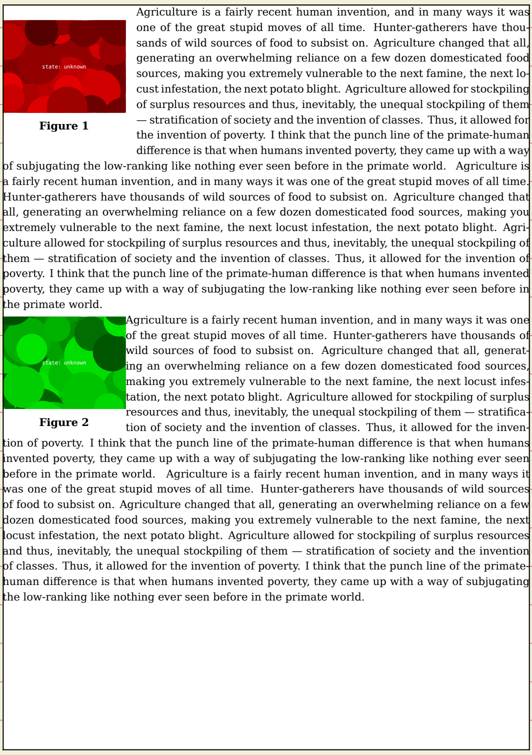


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

left,fit

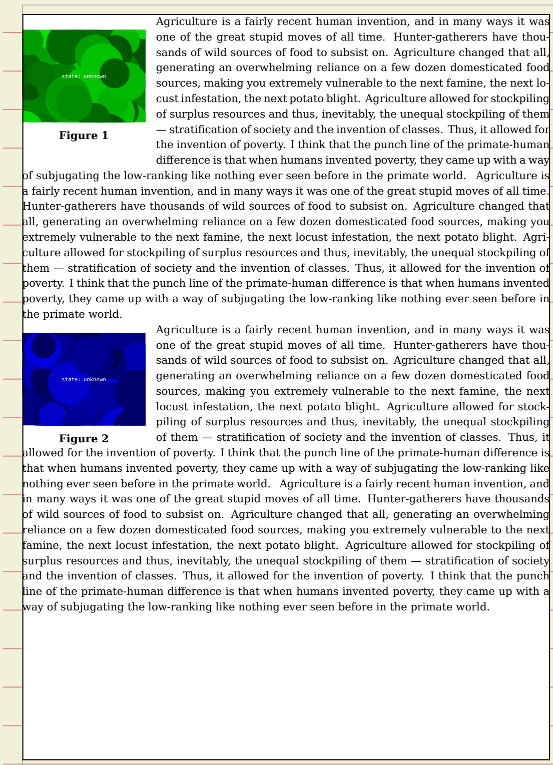


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`left,halfline`

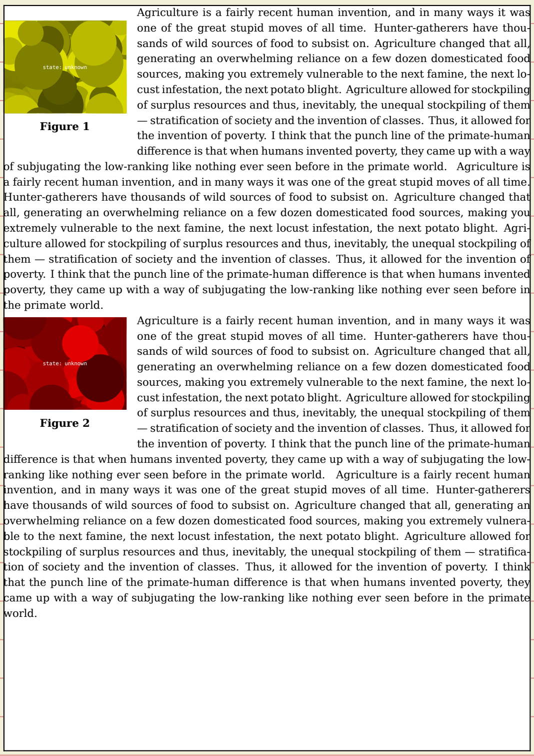


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`left,height`

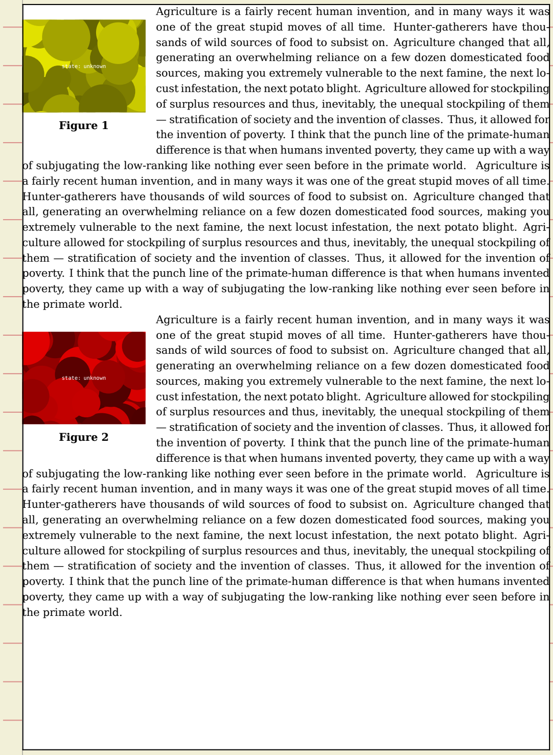


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`left,depth`

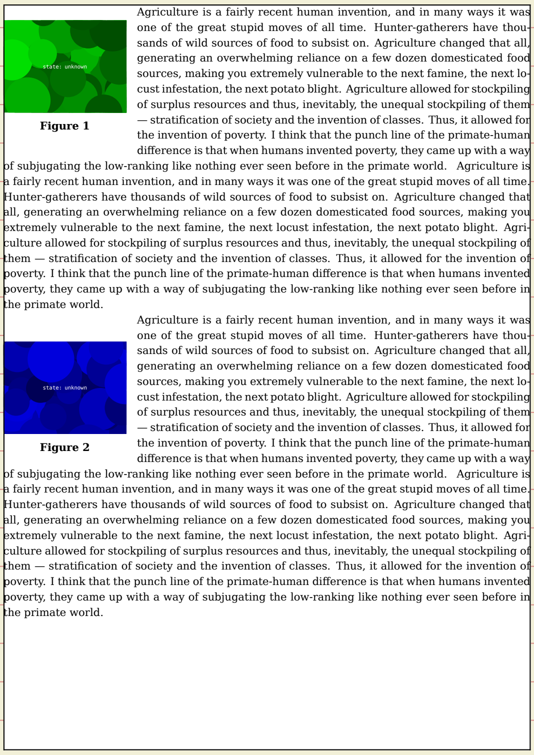


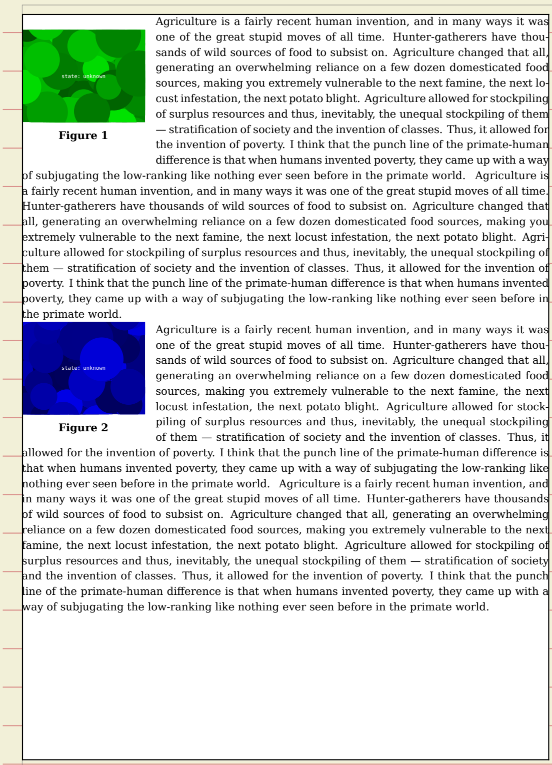
Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`left,grid`



state: unknown

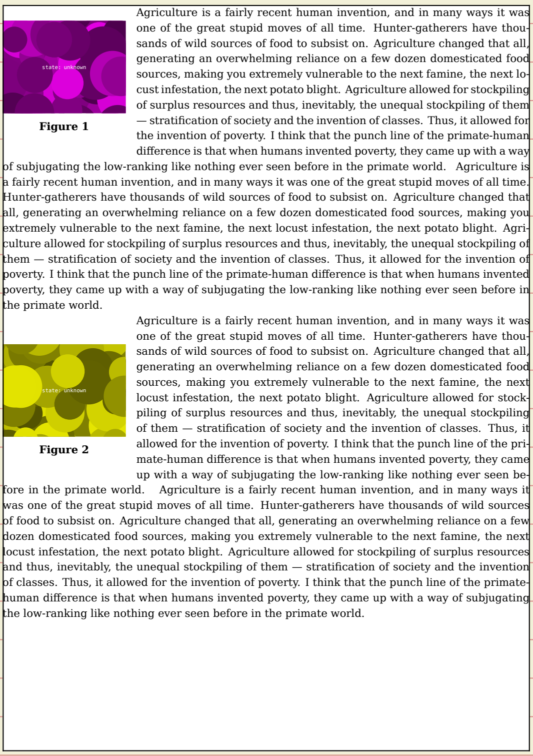
Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

state: unknown

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.



state: unknown

Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

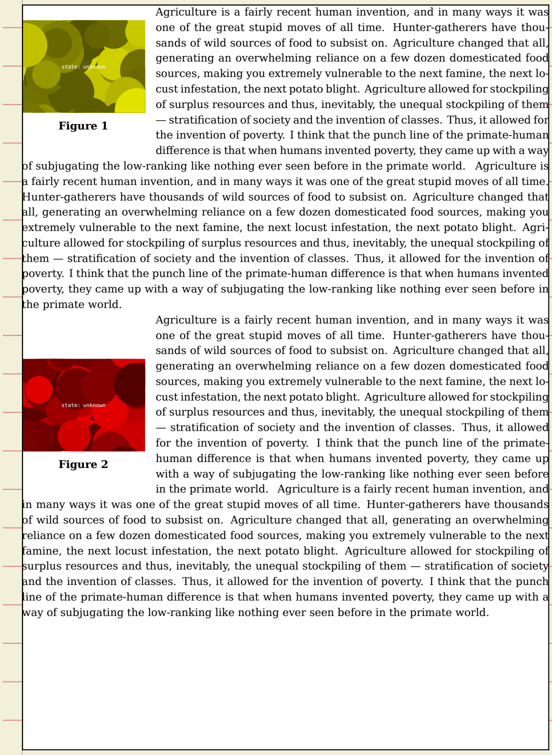
state: unknown

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

left,line

left,1*line



state: unknown

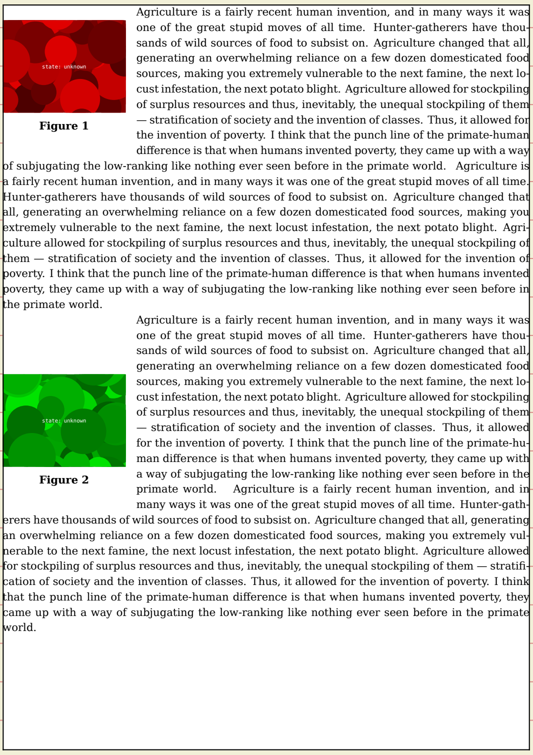
Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

state: unknown

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.



state: unknown

Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

state: unknown

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

left,2*line

left,3*line

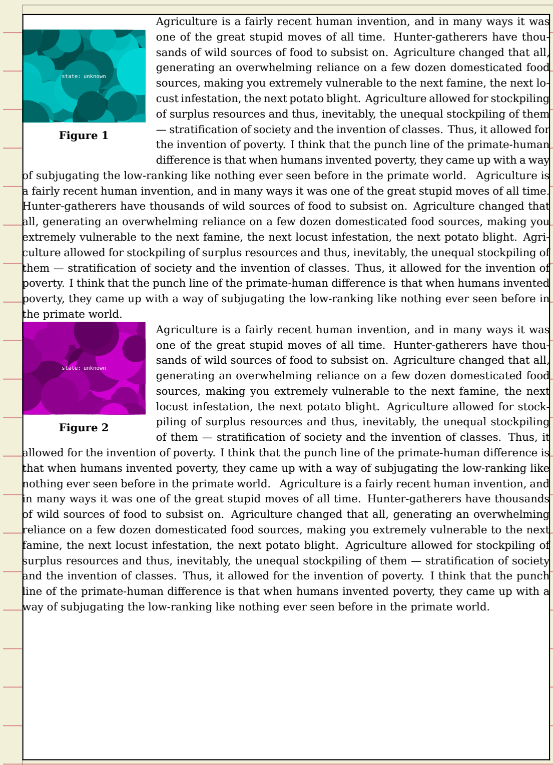


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`default={left,line}`

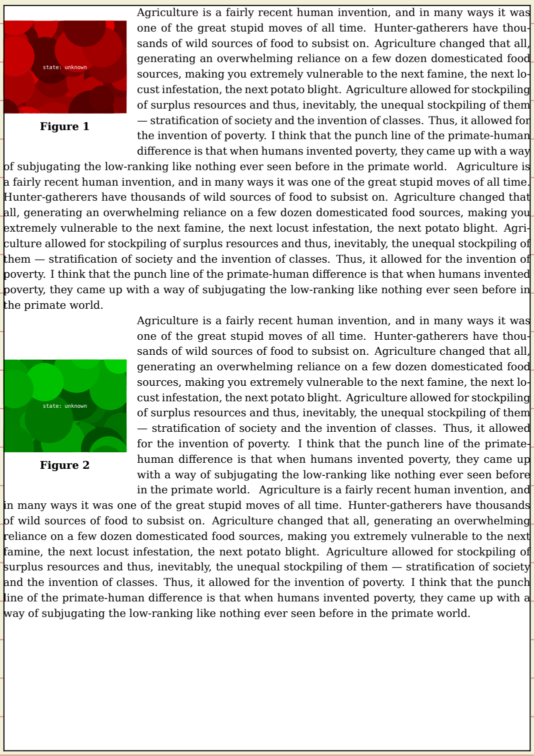


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`default={left,2*line}`

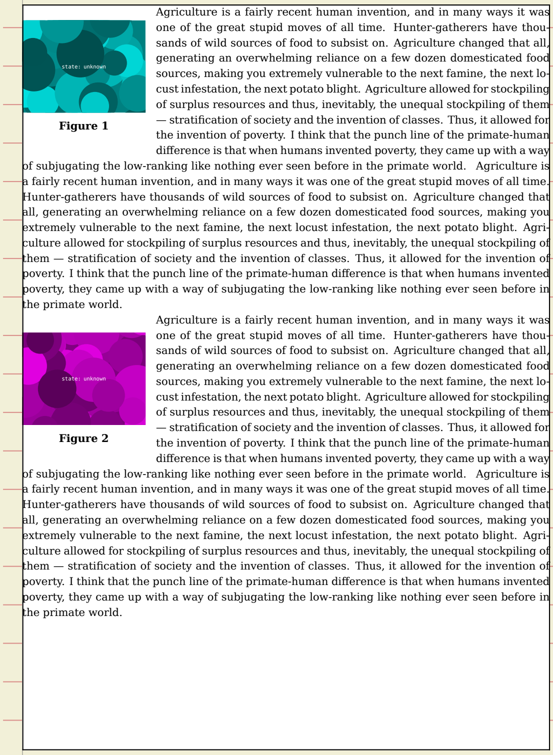


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`default=left, topoffset=5pt`

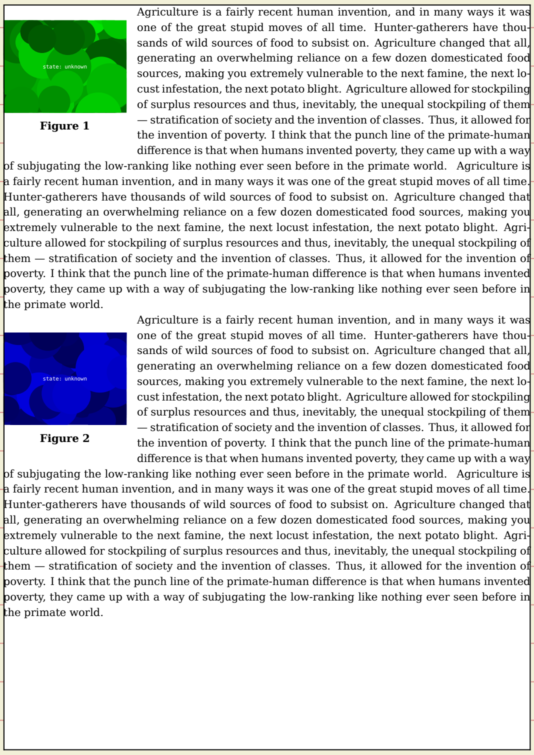


Figure 1

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

Figure 2

Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world. Agriculture is a fairly recent human invention, and in many ways it was one of the great stupid moves of all time. Hunter-gatherers have thousands of wild sources of food to subsist on. Agriculture changed that all, generating an overwhelming reliance on a few dozen domesticated food sources, making you extremely vulnerable to the next famine, the next locust infestation, the next potato blight. Agriculture allowed for stockpiling of surplus resources and thus, inevitably, the unequal stockpiling of them — stratification of society and the invention of classes. Thus, it allowed for the invention of poverty. I think that the punch line of the primate-human difference is that when humans invented poverty, they came up with a way of subjugating the low-ranking like nothing ever seen before in the primate world.

`default=left, topoffset=5pt, bottomoffset=5pt`

There is some tracing built in but as this mechanism is rather complex it only gives an idea about what is going on. Here is an example:

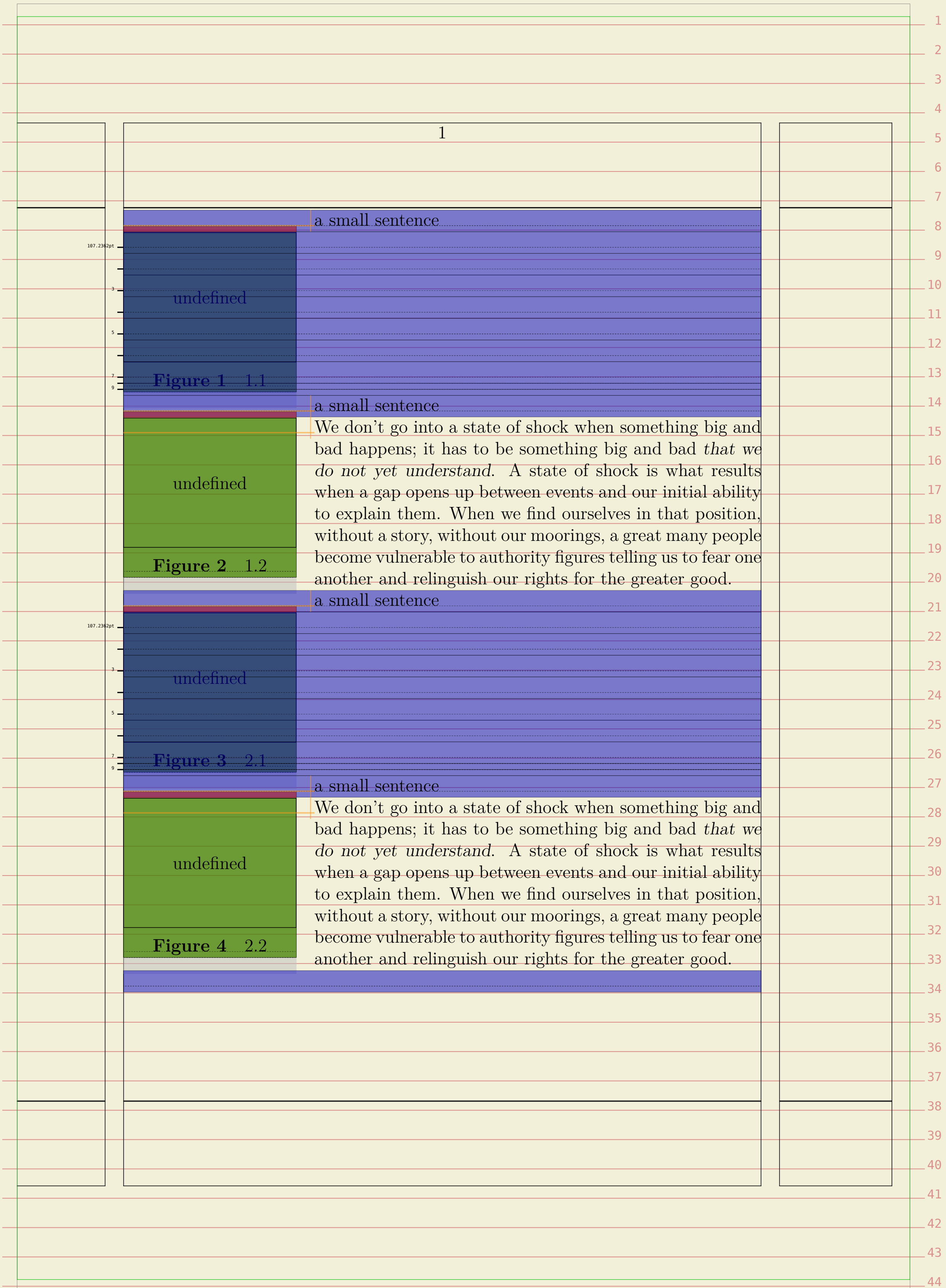
```
\enabletrackers[floats.anchoring]
```

```
\showframe

\setupfloat
  [sidespacebefore=big,
   sidespaceafter=big]

\starttext
    \dorecurse{10}{
        \placefigure[left]{#1.1}{}
        a small sentence \par
        \placefigure[left]{#1.2}{}
        a small sentence \par
        \input klein \par
    }
\stoptext
```

In figure 5.1 and figure 5.2 you see the first two pages of the typeset result. The anchor to the text is showed in orange and an optional shift in red. The content is in green and a depth compensation in magenta. Dummy lines added for proper spacing as well as progressing beyond a previous float are in blue.
A second example that uses different settings is shown in figure 5.3 and figure 5.4.

```
\enabletrackers[floats.anchoring]

\setupfloat
  [sidespacebefore=,
   sidespaceafter=big,
   step=small]

\showframe

\starttext
    \dorecurse{10}{
        \placefigure[left]{#1.1}{}
        a small sentence \par
        \placefigure[left]{#1.2}{}
        a small sentence \par
        \input klein \par
    }
\stoptext
```

Progressing next to a side float and determining how many lines to indent is a somewhat complex mechamism because many factors play a role and spacing can interfere badly. The decision about the number of lines to hang is to some extend controllable but there are cases when you need to steer it (for instance by scaling an image). In the next overviews we see the result of the following somewhat complex setup:

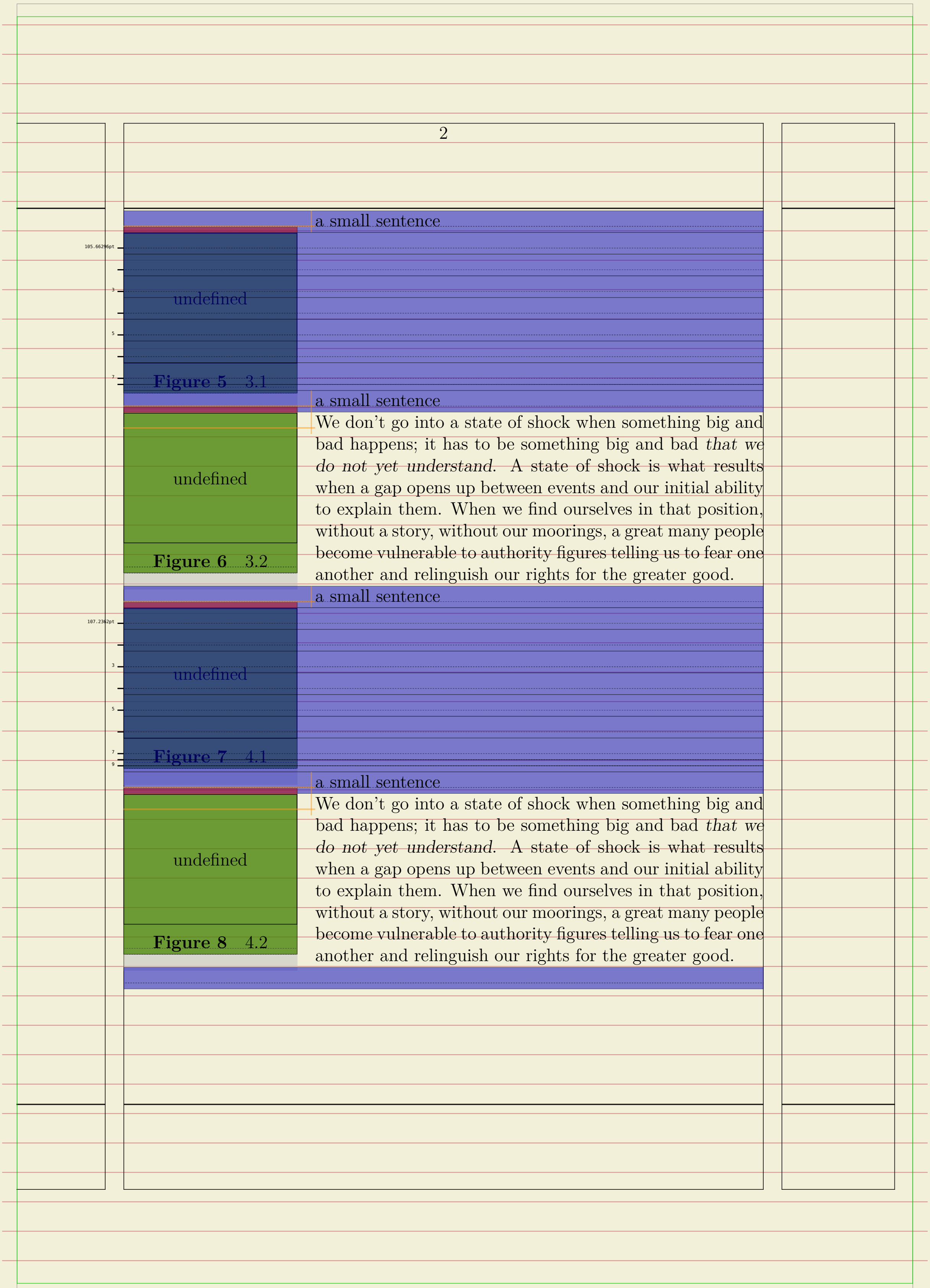


We don’t go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinquish our rights for the greater good.

We don’t go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinquish our rights for the greater good.

Figure 5.1 Side float tracing example 1, page 1.

```
\usemodule[simulate]
```



2

a small sentence

a small sentence

We don’t go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

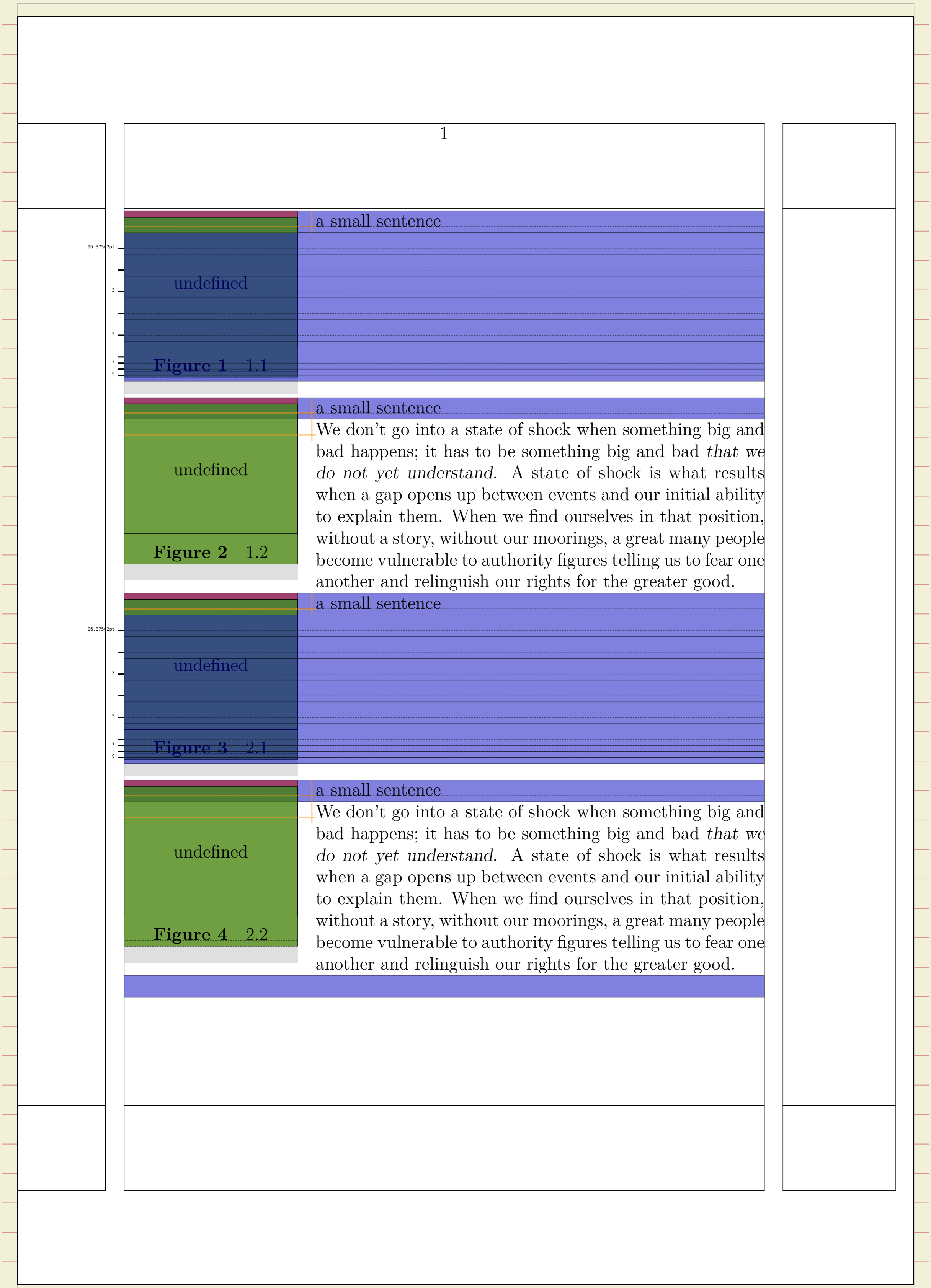
a small sentence

a small sentence

We don’t go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

Figure 5.2 Side float tracing example 1, page 2.

```
\setuplayout
  [tight]
```



1

undefined

Figure 1 1.1

a small sentence

undefined

Figure 2 1.2

a small sentence

We don’t go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand.* A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

undefined

Figure 3 2.1

a small sentence

undefined

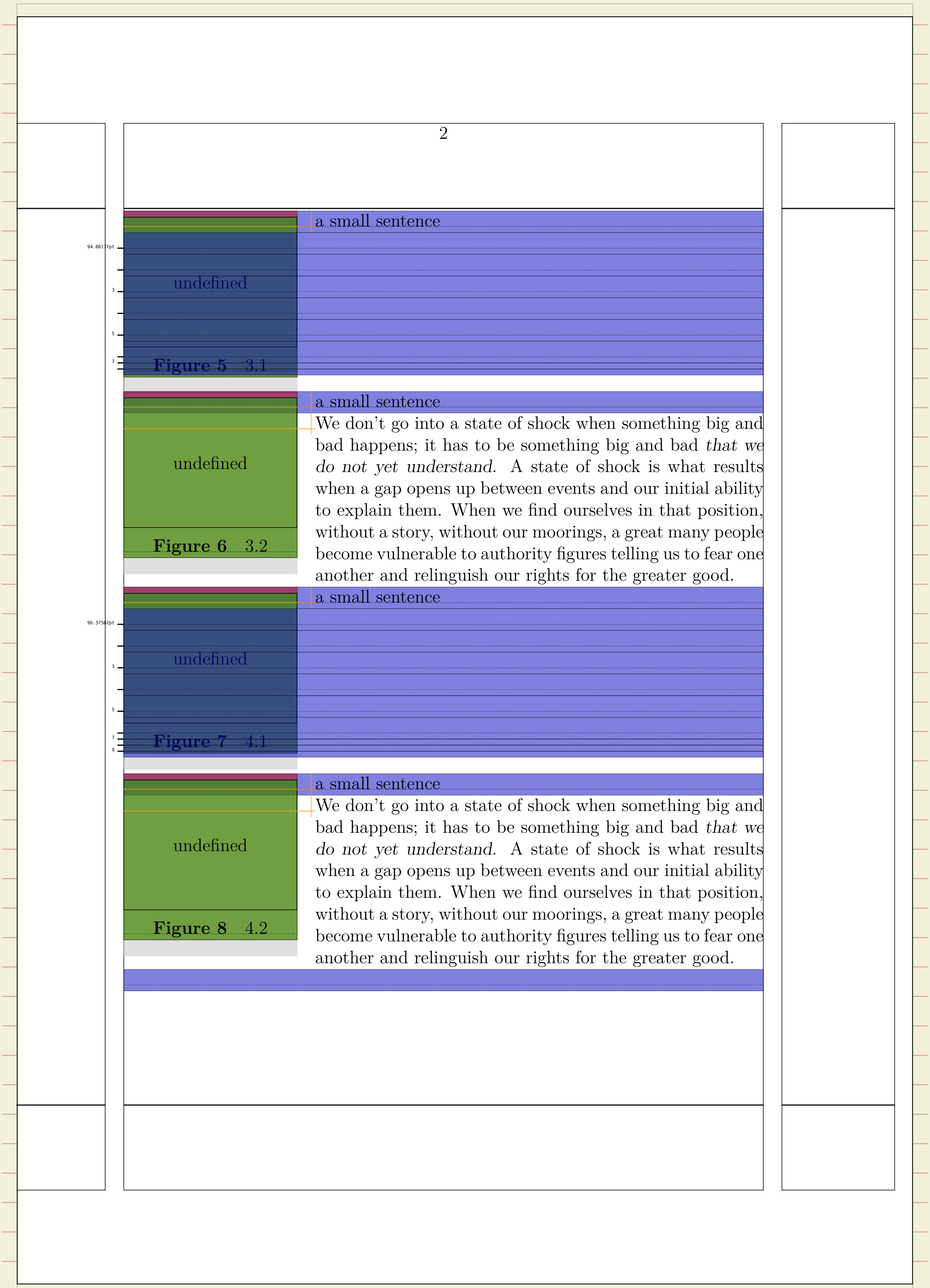
Figure 4 2.2

a small sentence

We don’t go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand.* A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

Figure 5.3 Side float tracing example 2, page 1.

```
\setupbodyfont
  [dejavu]
```



2

a small sentence

undefined

Figure 5 3.1

a small sentence

undefined

Figure 6 3.2

We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

a small sentence

undefined

Figure 7 4.1

a small sentence

undefined

Figure 8 4.2

We don't go into a state of shock when something big and bad happens; it has to be something big and bad *that we do not yet understand*. A state of shock is what results when a gap opens up between events and our initial ability to explain them. When we find ourselves in that position, without a story, without our moorings, a great many people become vulnerable to authority figures telling us to fear one another and relinguish our rights for the greater good.

Figure 5.4 Side float tracing example 2, page 2.

```
\enabletrackers
  [floats.anchoring]
```

```
\setupfloats
  [sidethreshold=.5\strutdp, % default, use "old" for previous
implementation
   step=small]

\definemeasure[MyHeight][3cm]
\definemeasure[MyWidth] [3cm]

% \setupheadertexts
%   [width=\measure{MyWidth}\quad height=\measure{MyHeight}]

\unexpanded\def\FakeWords#1%
  {\simulatewords
     [n=#1,m=#1,min=1,max=5,hyphen=no,color=text,line=yes,random=1234]}

\starttext

\startbuffer
    \FakeWords{100}\par
    \placefigure
        [left] {oeps}
        {\framed[width=\measure{MyWidth},height=\measure{MyHeight}]{}}
    \FakeWords  {2}\par
    \FakeWords  {3}\par
    \FakeWords  {5}\par
    \FakeWords  {4}\par
    \FakeWords{200}\par
    \placefigure
        [left] {oeps}
        {\framed[width=\measure{MyWidth},height=\measure{MyHeight}]{}}
    \FakeWords{200}\par
\stopbuffer

\dostepwiserecurse {\number\dimexpr3cm} {\number\dimexpr4cm} {\number\dimexpr0.
{
    \definemeasure[MyWidth][#1sp]
    \dostepwiserecurse {\number\dimexpr3cm} {\number\dimexpr4cm}
{\number\dimexpr0.25cm} {
        \definemeasure[MyHeight][##1sp]
        \start
            \setupwhitespace[none]
            \getbuffer \page
        \stop
        \start
            \setupwhitespace[big]
            \getbuffer \page
        \stop
    }
    }
    \stoptext
```

The `step` parameter controls how we fill up the space when we need to progress beyond it for instance because another float shows up or because we issue a `\flushsidefloats`. Its value can be `big`, `medium` or `small` and defaults to `small` which gives of enough precision. The `sidethreshold` parameter controls the number of lines that we hang around the float. Here we only show the consequence of the the threshold. A larger threshold result in mode whitespace below the side float. You can zoom in to see what happens at the bottom of the float (or run the examples yourself).

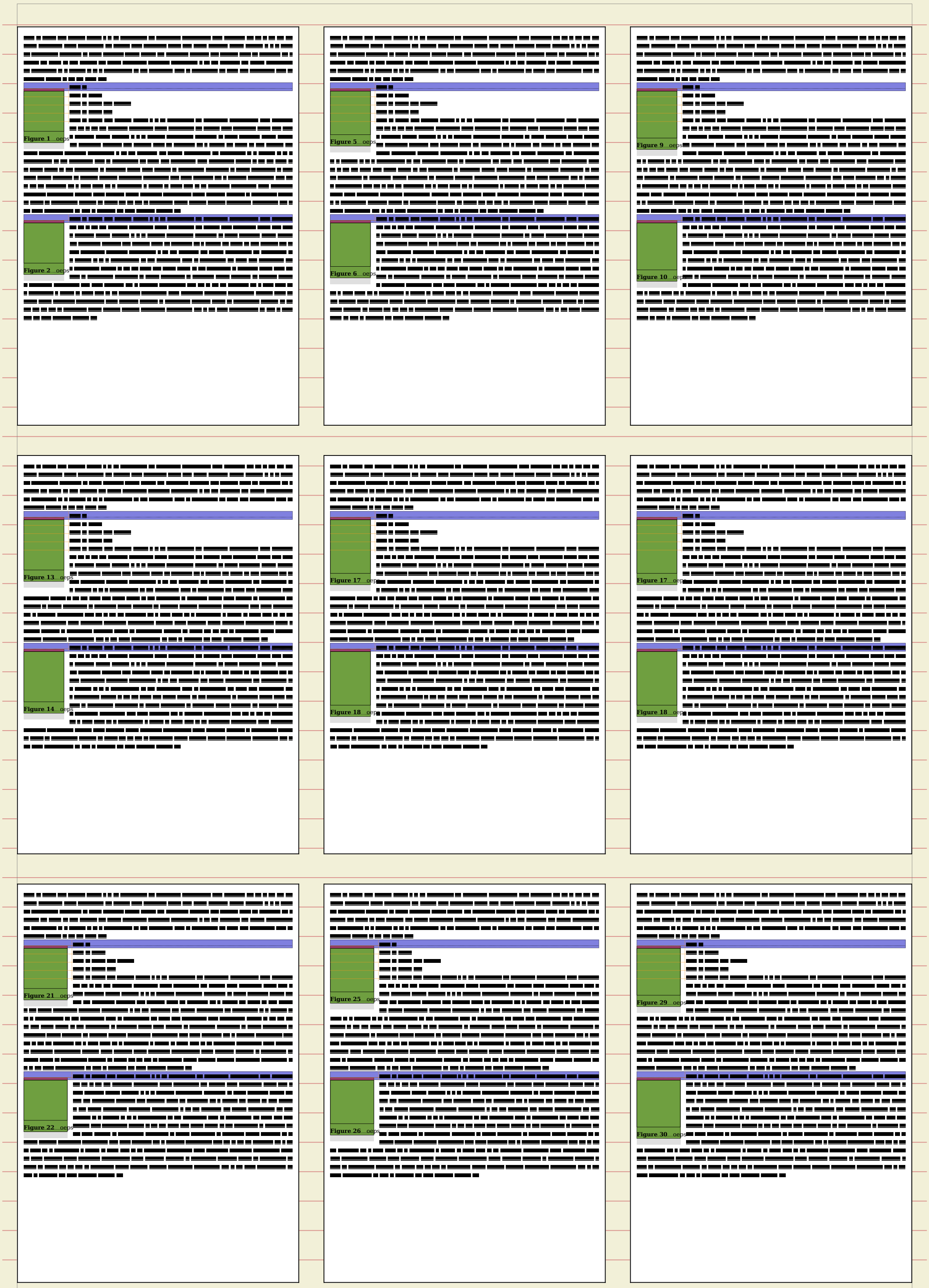


Figure 5.5 The working of default step and side threshold (no whitespace.

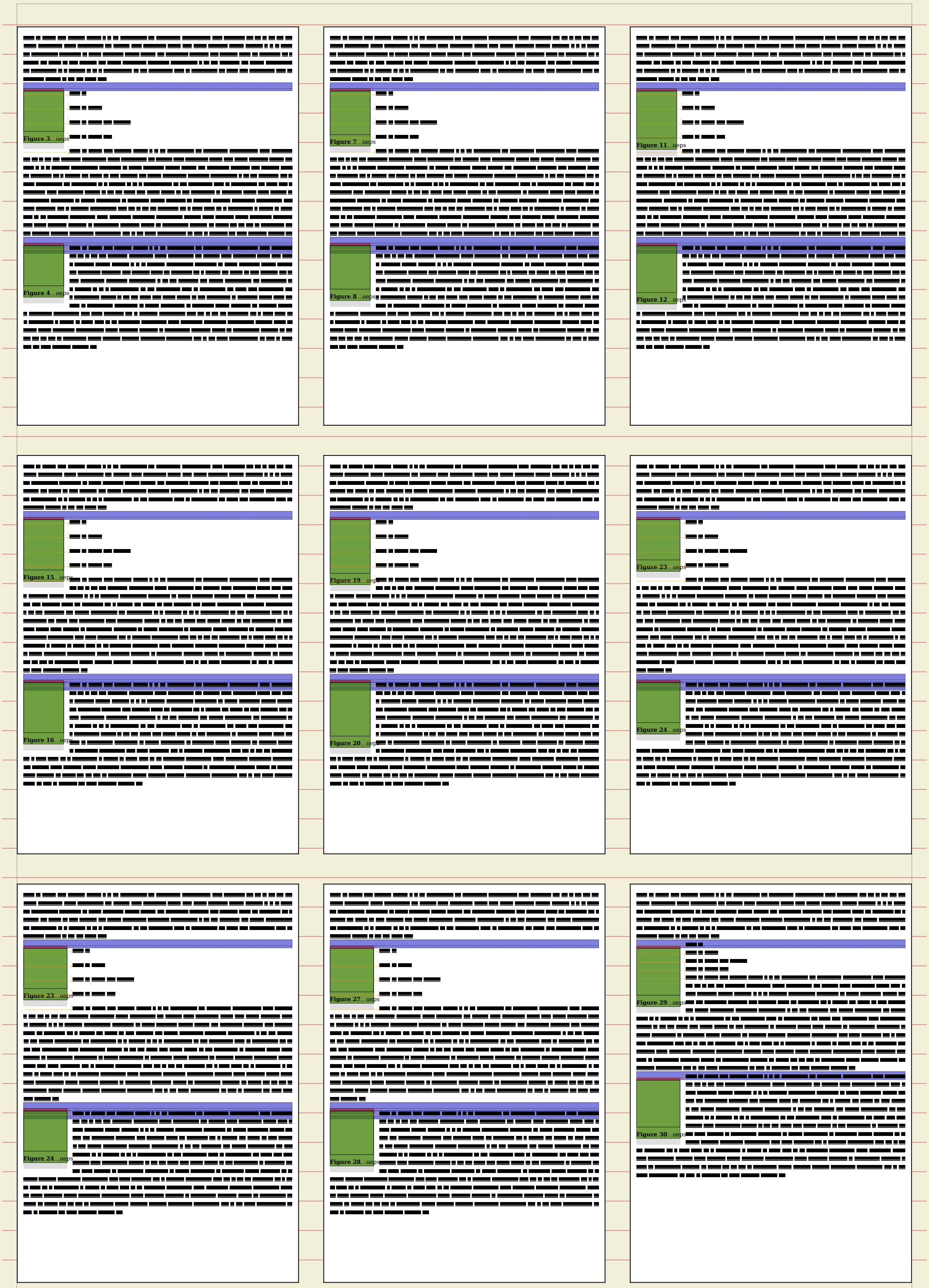


Figure 5.6 The working of default step and side threshold (whitespace).

In this chapter we will discuss a few more tricks to control float placement. This control is needed if you want to typeset documents in a semi desk top publishing way.

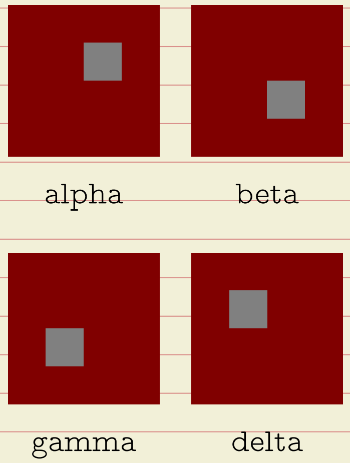
When you combine technical graphics, you may wish to align the content optically. This can be done with the `offset` command. We will demonstrate this with a couple of MetaPost graphics:

```
\startreusableMPgraphic{alpha}
  fill fullsquare xyscaled(  2cm,  2cm) withcolor \MPcolor{red} ;
  fill unitsquare xyscaled(+.5cm,+.5cm) withcolor \MPcolor{gray} ;
\stopreusableMPgraphic

\startreusableMPgraphic{beta}
  fill fullsquare xyscaled(  2cm,  2cm) withcolor \MPcolor{red} ;
  fill unitsquare xyscaled(+.5cm,-.5cm) withcolor \MPcolor{gray} ;
\stopreusableMPgraphic

\startreusableMPgraphic{gamma}
  fill fullsquare xyscaled(  2cm,  2cm) withcolor \MPcolor{red} ;
  fill unitsquare xyscaled(-.5cm,-.5cm) withcolor \MPcolor{gray} ;
\stopreusableMPgraphic

\startuseMPgraphic{delta}
  fill fullsquare xyscaled(  2cm,  2cm) withcolor \MPcolor{red} ;
  fill unitsquare xyscaled(-.5cm,+.5cm) withcolor \MPcolor{gray} ;
\stopuseMPgraphic

\startcombination[2*2]
  {\reuseMPgraphic{alpha}} {alpha}
  {\reuseMPgraphic {beta}}  {beta}
  {\reuseMPgraphic{gamma}} {gamma}
  {\reuseMPgraphic{delta}} {delta}
\stopcombination
```

In figure 6.1 we place these graphics in a `2*2` grid. As you can see, the centers don't align well.

In figure 6.2 the centers of the graphic align well. This is accomplished by adding some space around the graphics.

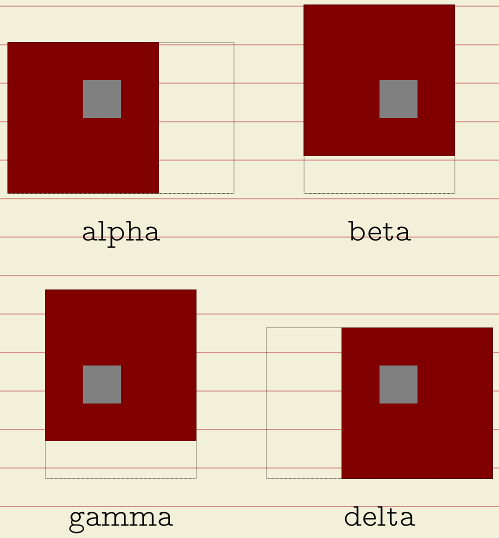
```
\startcombination[2*2]
  {\offset[rightoffset=1cm]   {\reuseMPgraphic{alpha}}} {alpha}
  {\offset[bottomoffset=.5cm]{\reuseMPgraphic {beta}}}  {beta}
  {\offset[bottomoffset=.5cm]{\reuseMPgraphic{gamma}}} {gamma}
  {\offset[leftoffset=1cm]    {\reuseMPgraphic{delta}}} {delta}
\stopcombination
```



alpha beta

gamma delta

Figure 6.1



alpha beta

gamma delta

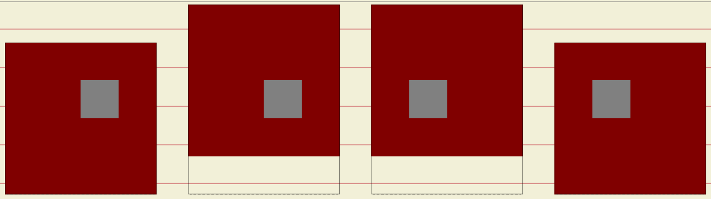
Figure 6.2

If we align the centers vertically, as demonstrated in figure 6.2 we can stick to a few bottom offsets.

```
\startcombination[4*1]
                             {\reuseMPgraphic{alpha}}  {alpha}
  {\offset[bottomoffset=.5cm]{\reuseMPgraphic {beta}}}  {beta}
  {\offset[bottomoffset=.5cm]{\reuseMPgraphic{gamma}}} {gamma}
                             {\reuseMPgraphic{delta}}  {delta}
\stopcombination
```

These examples demonstrate that the dimensions change with the offset. You can retain the dimensions but still align them by using the `x` and `y` parameter. This kind of manipulations will often result in a ugly spacing because the placement macros handle on the original dimensions. Figure 6.4 demonstrates this.

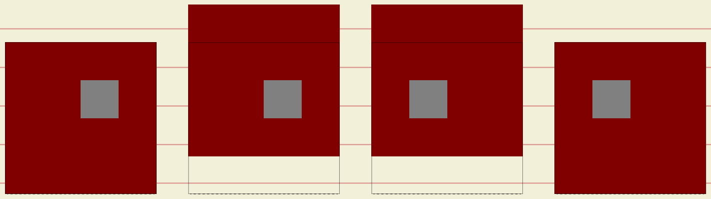
```
\startcombination[4*1]
```



alpha beta gamma delta

Figure 6.3

```
                   {\reuseMPgraphic{alpha}}  {alpha}
  {\offset[y=-.5cm]{\reuseMPgraphic {beta}}}  {beta}
  {\offset[y=-.5cm]{\reuseMPgraphic{gamma}}} {gamma}
                   {\reuseMPgraphic{delta}}  {delta}
\stopcombination
```



alpha beta gamma delta

Figure 6.4

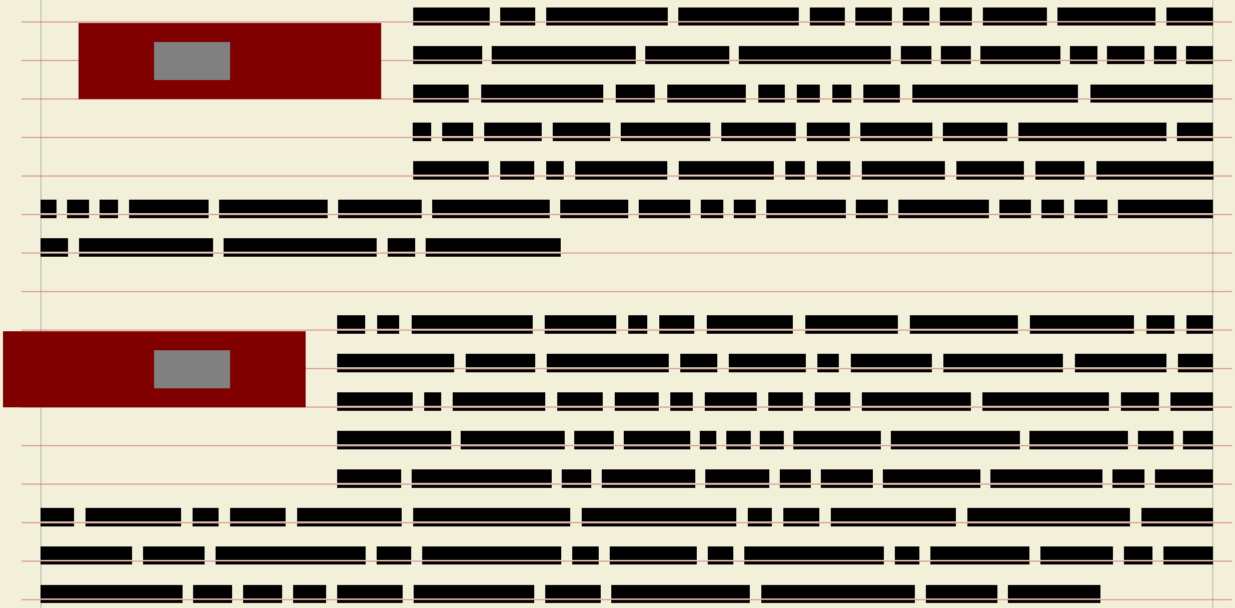
In the previous chapter we demonstrated how a side float can be moved up or down by providing a placement directive or by preceding the placement with `\movesidefloat`. Such a move can be used to align a graphic with particular line of text. This command can also be used for alignment purposes similar to the `\offset` command. We will demonstrate this with the following graphics.

```
\startreusableMPgraphic{gnu}
  fill fullsquare xyscaled( 4cm, 1cm) withcolor \MPcolor{red} ;
  fill unitsquare xyscaled(-1cm,.5cm)
                   shifted (0,-.25cm) withcolor \MPcolor{gray} ;
\stopreusableMPgraphic

\startreusableMPgraphic{gnat}
  fill fullsquare xyscaled( 4cm, 1cm) withcolor \MPcolor{red} ;
  fill unitsquare xyscaled(+1cm,.5cm)
                   shifted (0,-.25cm) withcolor \MPcolor{gray} ;
\stopreusableMPgraphic
```

In the next two examples we shift the `gnu` and `gnat` graphics horizontally in order to get them aligned. The move does not change the dimensions of the float, but they do influence the paragraph shape.

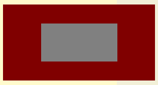
```
\movesidefloat [x=.5cm]
\placefigure   [left,none] {} {\reuseMPgraphic{gnu}}
```

```
\movesidefloat [x=-.5cm]
\placefigure   [left,none] {} {\reuseMPgraphic{gnat}}
```

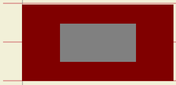


It is possible to shift vertically by setting `y`, but this is often a bad idea and definitely may spoil alignment of graphics to the grid. If you have to revert to this trick, you are probably working in document screw–up mode. This is why in grid mode, we automatically round to an equal number of lines.
If you know what text you're dealing with and also can be sure about the height of a graphic, you can trick ConTEXt to ignore the dimensions of a graphic. Here we use the graphic:

```
\startreusableMPgraphic{gnome}
  fill fullsquare xyscaled(2cm, 1cm) withcolor \MPcolor{red} ;
  fill fullsquare xyscaled(1cm,.5cm) withcolor \MPcolor{gray} ;
\stopreusableMPgraphic

\placefigure[leftmargin,none,reset]{}{\reuseMPgraphic{gnome}}
```



The graphic is moved into the margin (`leftmargin`), has no caption (`none`), and all kind of tricky housekeeping is reset (`reset`).



Now the next graphic is not influenced by the previous one, so we can place them close to each other. Use these tricks with care, especially if your document source is reused and the typeset products are not carefully checked.

```
\placefigure[left,none,high,low]{}{\reuseMPgraphic{gnome}}
```

When ConTEXt tries to determine if a float fits, it makes a couple of assumptions, for instance that the available room equals the text height minus the height of the text so far. You can slightly influence the way these values are interpreted by setting the calculation method. You can set the methods as follows:

```
\setupfloats[textmethod=0,sidemethod=1]
```

Method 0 just looks at the raw dimensions, while method 1 lessens the maximum text height by one percent, thereby playing safe. Method 2 takes a window of 1 point. This may lead to better decisions since we may run into rounding errors of several scaled points (which is small but troublesome). Method 2 is well suited when typesetting on a grid, because there everything has to fit in a rounded number of lines, which leaves no room for rounding errors.

grid mode	**yes**	**no**
`sidemethod`	2	1
`textmethod`	2	0

As you may know by now, we can use the directives `high`, `low`, `height`, `depth` and `line` to influence the spacing around a side float. A real tight spacing can be achieved with `fit`.

```
\placefigure[left,fit,none]{}{some graphic}
```

This kind of placements only make sense in special situations, because normally you don’t want the graphic to touch the text. If you think that this is all a user may want, you’re wrong. It is not imaginary that graphics have small pieces sticking out and/or lots of white space as part of their design. In that case, the bounding box can be set to a smaller size.

Now, when handling a side float, ConTEXt first places the float, and then starts with typesetting the paragraph, cleverly avoiding the graphic. However, when the graphic is virtually larger than its known size, it may cover part of the preceding paragraph.
How come that the graphic starting this paragraph does not do that? It is because we explicitly moved it to the background. This involves some preparation. At the document level, we define a layer called `graphic`.

```
\definelayer[graphics][position=yes]
```

The position directive tells ConTEXt that it should honour the position of the graphic. Next we must make sure that this layer is placed.

```
\setupbackgounds[page][background=graphics]
```

Now we’re ready to move graphics to this layer:

```
\placefigure
```

```
  [left,fit,none]
  {}{\setlayer[graphics]{graphic}}
```

It’s now a small step to more advanced movements. Say that you want to move the graphic a little bit to the left. In that case you can tell the layer placement to do so.

```
\placefigure
  [left,fit,none]{}{\setlayer[graphics][hoffset=-12pt]{graphic}}
```

From this you can deduce that there is also a movement in the vertical direction using `voffset`. In addition you can anchor the graphic using the `location` parameter and provide offsets.

As soon as you run into situations where float placement is to be consistently enforced, you will feel the need for dedicate placement macros. For example:

```
\definefloat
  [somefloat]
  [figure]

\setupfloat
  [somefloat]
  [sidespaceafter=,
   sidespacebefore=,
   default={left,none}]
```

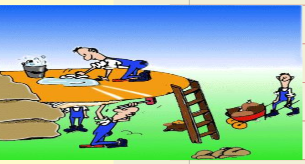
Instead of resetting the side spacing, we could have default to `high,low`, but this way we can overload the default placement and still get zero spacing.
Throughout this manual we discuss features related to overlays and layers. These permit you to move content around in ways that either or not depend on the text flow. We have now come to another trick based on these mechanisms: bleeding.
When printing a document, you need to take into account that when graphics go beyond the page boundary, you need to compensate for inaccuracies in cutting the pages. Such graphics are called bleeding graphics and the amount of bleed is often a few millimeters.
The best way to handle such graphics is to use the correct dimensions and play with the edge widths and distances in combination with backspace and cut space. In a properly set up layout and by using a well designed set of predefined graphic placements, you can handle this quite well. A bleeding figure can be defined as follows:

```
\definefloat
  [edgefigure]
  [figure]

\setupfloat
```

```
  [edgefigure]
  [default={inner,height,high,low,none},
   maxwidth=4cm]

\defineexternalfigure
  [edgefigure]
  [width=\dimexpr\backspace+4cm-1mm\relax,
   lines=4]
```

The default placement is pre-configured to have no additional vertical space and align on the height of a line (this is default behaviour so the `height` key is redundant here. The 1mm in the previous definition simulates what happens when a page is cut off slightly wrong: we get an annoying gap.

```
\placeedgefigure
  {}
  {\externalfigure[hacker][edgefigure]}
```



One of the nice things about TEX is that you can fine tune dimensions pretty well. So, instead of the previous placement, which turns out rather ugly, we can come up with a better one:

```
\setupfloat
  [edgefigure]
  [default={inner,height,high,low,none},
   maxwidth=4cm,
   margin=\strutdepth]

\defineexternalfigure
  [edgefigure]
  [width=\dimexpr\backspace+4cm+2mm\relax,
   height=\dimexpr3\lineheight+\strutheight\relax]
```

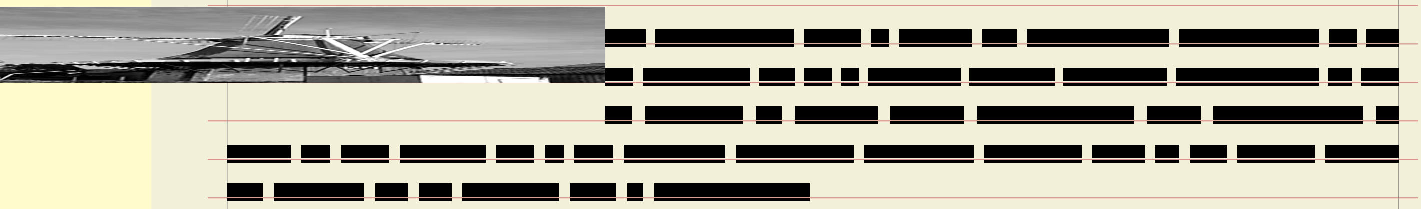
This time we take no risk and add 2mm to the dimensions so that we can be sure that the edge of the graphic falls outside the page boundary.

The ConTEXt resourse library modules provide means to report back the dimensions of graphics used in a document, so that you can develop (tune) them with the proper dimensions. In practice a slightly wider than normal graphic (scaling it horizontally a few millimeters more) does not harm the visual appearance that much, so adapting a graphic to this kind of bleeding is not really needed.

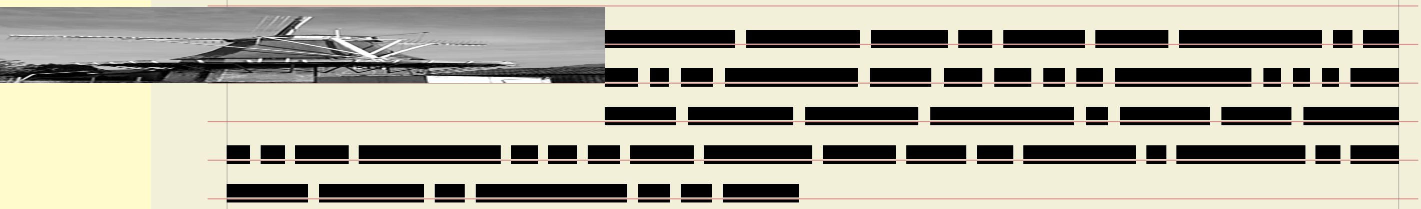
In addition to this (rather natural) way of adding bleed to a graphic, you can apply the `\bleed` macro. In the previously discussed method the figure placement mechanisms work with the real dimensions. The `bleed` macro is using scaling in a different way: from the perspective of ConTeXt the graphic remains its original dimensions and the figure placement mechanisms will act accordingly. We will give a couple of examples of using this macro.

Permitted bleeding locations are `l`, `r`, `t`, `b`, `lr`, `bl`, `br`, `tl` and `tr`.

```
\placesomefloat
  [left,none,fit]
  {}
  {\setupbleeding[offset=5mm]%
   \bleed[width=5cm,height=1cm,location=l]
     {\externalfigure[mill][bleed]}}
```



```
\placesomefloat
  [left,none,fit]
  {}
  {\setupbleeding[offset=2mm]%
   \bleed[width=5cm,height=1cm,location=l]
     {\externalfigure[mill][bleed]}}
```



The amount of bleeding depends on the postprocessing. In the previous paragraph we used a bleed offset of 5mm, and here we used 2mm. Because the graphic is scaled in order to match the bleed, it will be slightly distorted. With small values this will go unnoticed. You can set the offset with:

```
\setupbleeding[offset=5mm]
```

Bleeding itself is accomplished by the `\bleed` macro as in:

```
\bleed
```

```
    [width=5cm,height=1cm,location=l]
    {\externalfigure[mill][width=\bleedwidth,height=\bleedheight]}
```

It is kind of awkward to pass those two dimensions so here is a shorter way of doing the same:

```
\bleed
  [width=5cm,height=1cm,location=l]
  {\externalfigure[mill][bleed]}
```

In fact, this uses the following definition:

```
\defineexternalfigure[bleed][width=\bleedwidth,height=\bleedheight]
```

You can influence the scaling of a graphic by setting the `stretch` parameters. The location parameter determines the direction of the stretch: `l` (left), `r` (right), `t` (top), `b` (bottom) or a combination of these. We will now combine the previous example code with this knowledge.

```
\placefigure
  [left]
  {}
  {\bleed
     [stretch=no,voffset=0pt,hoffset=1cm]
     {\externalfigure[detcow][bleed]}}
```

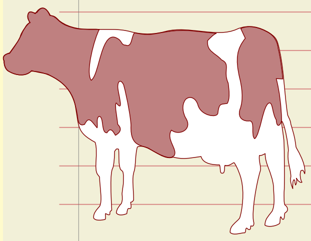


Figure 6.5

```
\placefigure
  [left]
  {}
  {\bleed
     [width=5cm,height=3cm,location=l]
     {\externalfigure[detcow][bleed]}}
```

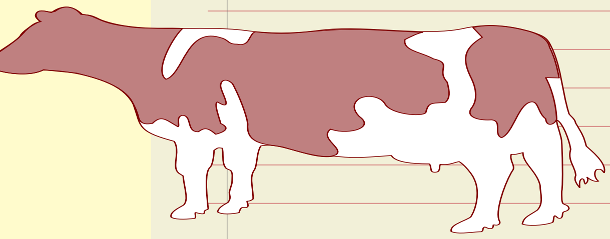


Figure 6.6

```
\placefigure
  [right]
  {}
  {\bleed
     [width=5cm,height=3cm,location=r]
     {\externalfigure[detcow][bleed]}}
```

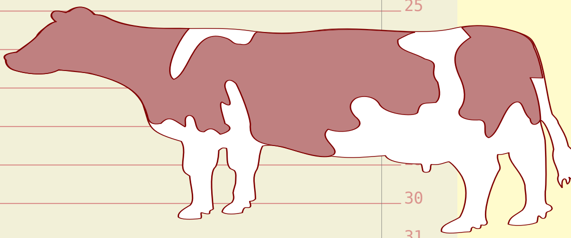


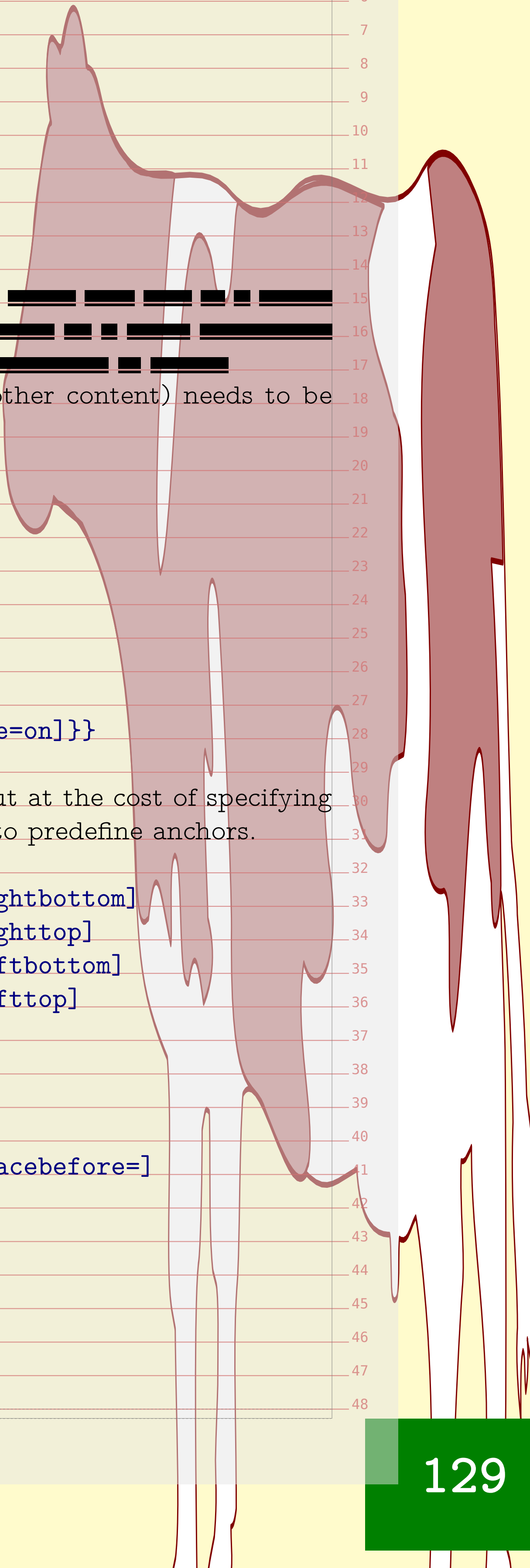
Figure 6.7

You can combine this feature with layers. We will now show a few applications which may look like magic at first glance, but will become natural to your repertoire once you have played with them.

The next example moves the graphic to a layer associated with the (current) page.

```
\placefigure
  [right,none]
  {}
  {\setlayer
     [graphics]
```

```
        {\bleed
           [width=5cm,height=3cm,location=rb]
           {\externalfigure[detcow][bleed]}}}
```



You can also predefine locations where graphics (or other content) needs to be anchored. A direct call to anchor looks as follows:

```
\placefigure
  [left,none]
  {}
  {\anchor
     [text-1]
     [location=lt,hoffset=max,voffset=max]
     [width=3cm,height=3cm,frame=on]%
     {\externalfigure[detcow][width=5cm,frame=on]}}
```

This will anchor a graphic in one of the text layers, but at the cost of specifying this in the document source. One way around this is to predefine anchors.

```
\defineanchor[rightbottom][text-1][preset=rightbottom]
\defineanchor[righttop]    [text-1][preset=righttop]
\defineanchor[leftbottom]  [text-1][preset=leftbottom]
\defineanchor[lefttop]     [text-1][preset=lefttop]
```

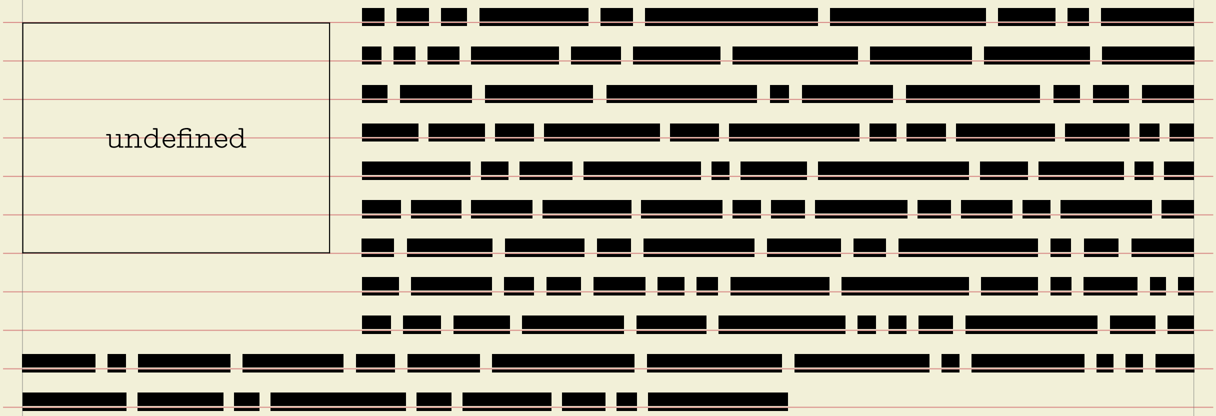
We will apply this to a predefined float type.

```
\definefloat[myfigure][figure]
\setupfloat[myfigure][sidespaceafter=,sidespacebefore=]
```

Our previous example can now be reduced to:

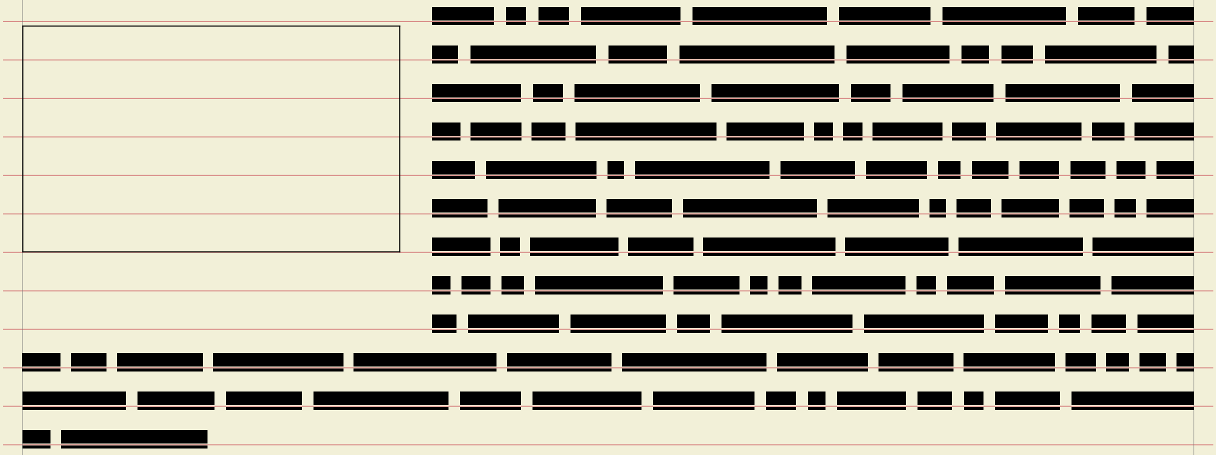
```
\placemyfigure
  [left,none]
  {}
  {\anchor[rightbottom]
```

```
        {\externalfigure[detcow][width=5cm,frame=on]}}
```

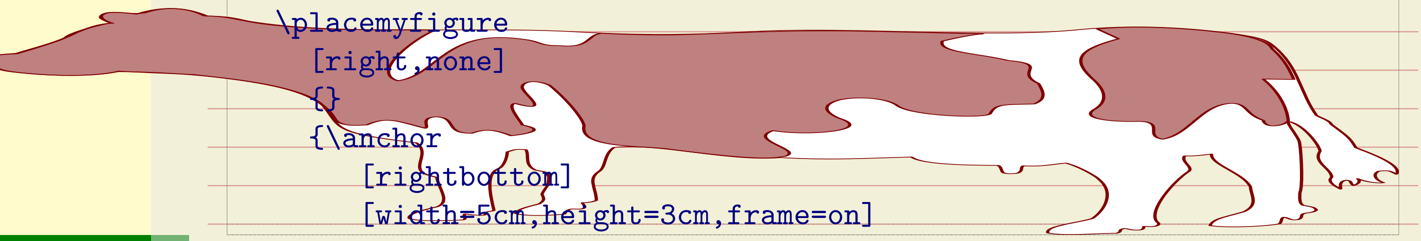


You can still specify dimensions and anchors can be combined with bleeding. Of course this kind of mixed usage means that you need to have some feeling for what these mechanisms do.

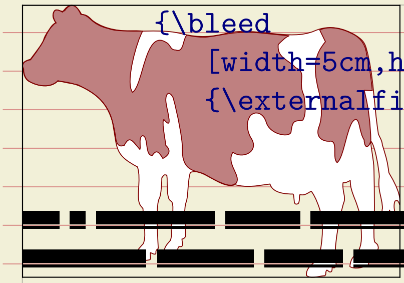
```
\placemyfigure
  [left,none]
  {}
  {\anchor
     [rightbottom]
     [width=5cm,height=3cm,frame=on]
     {\bleed
        [width=5cm,height=3cm,location=l]
        {\externalfigure[detcow][bleed]}}}
```



```
\placemyfigure
  [right,none]
  {}
  {\anchor
     [rightbottom]
     [width=5cm,height=3cm,frame=on]
```

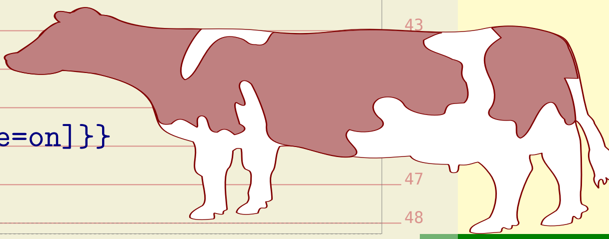


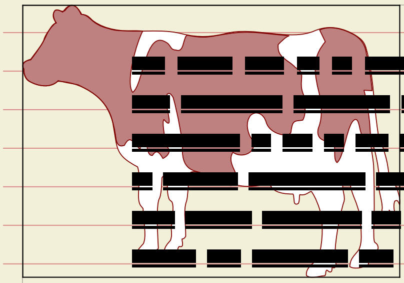
```
{\bleed
    [width=5cm,height=3cm,location=r]
    {\externalfigure[detcow][bleed]}}
```



```
\placemyfigure
  [left,none]
  {}
  {\anchor
      [lefttop]
      [width=3cm,height=3cm,frame=on]
      {\externalfigure[detcow][width=5cm,frame=on]}}
```

```
\placemyfigure
  [left,none]
  {}
  {\anchor
      [lefttop]
      [width=3cm,height=3cm,frame=on]
      [offset=.5cm]
      {\externalfigure[detcow][width=5cm,frame=on]}}
```





Todo: parameter specifications of all those macros.

The background mechanisms present in ConTeXt have evolved over time and with computers becoming faster, you can expect new functionality to show up and existing functionality to start using this technology. A simple background consist of a colored area. Many commands accept settings like:

```
...[background=color,backgroundcolor=red,backgroundoffset=3pt]
```

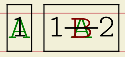
Instead of such an area you can define one or more so called overlays:

```
\defineoverlay[one][...]
\defineoverlay[two][...]

...[background={one,two}]
```

The name overlay comes from the fact that you stack them on top of each other. A special overlay is `foreground`, and deep down in ConTeXt there are more predefined overlays.
In the MetaFun manual you will find example of usage, so here we stick to a simple code snippet for testing this functionality:

```
\defineoverlay[one][\green A]
\defineoverlay[two][\red   B]

\framed[background=one]        {1}
\framed[background={one,two}] {1---2}
```

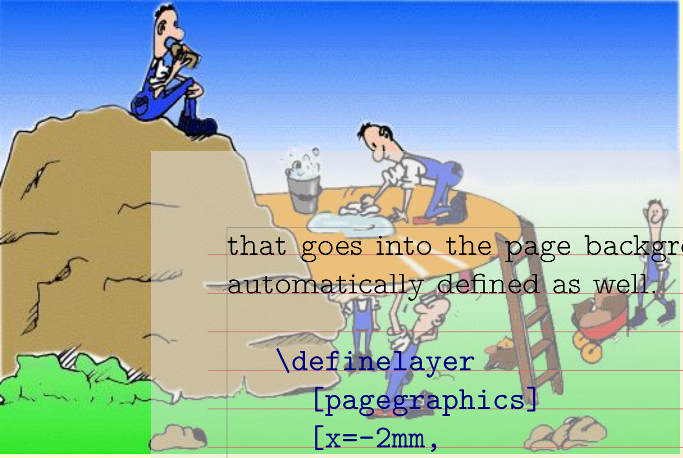
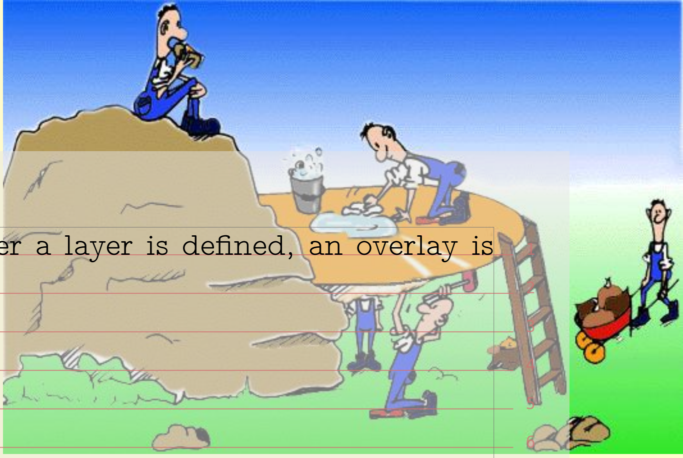
The rather ugly result is:



You can construct overlays by using TeX boxing primitives or commands like `\framed`. Alternatively you can use another mechanism: layers. Layers collect content and flush that when asked, for instance when an overlay is constructed. Layers can be independent of a page, or bound to a specific page number, left or right hand pages. Here we look at independent layers.
All these mechanisms are fine tuned for cooperating with the output routine (the part of TeX that deals with composing pages) and are well interact quite well with MetaPost graphics. Details of usage and tricks are revealed in this manual as well as in styles that come with ConTeXt. In this chapter we will apply layers to graphics. For this we need a few setups, like:

```
\setupbackgrounds
  [page]
  [background=pagegraphics]
```

Here we have set up the page background to use an overlay called `pagegraphics`. However, instead of an overlay, we will use a layer. This layer will collect content

that goes into the page background. Whenever a layer is defined, an overlay is automatically defined as well.

```
\definelayer
  [pagegraphics]
  [x=-2mm,
   y=-2mm,
   width=\paperwidth,
   height=\paperheight]
```

When you fill a layer with content, you can influence the placement with the `x` and `y` parameters as well as `hoffset` and `voffset`, whichever you prefer. The reference point and alignment are set with `corner` and `location`.
Live can be made easier by using presets, especially for our intended usage. The following presets are predefined.

```
\definelayerpreset
  [lefttop]     [corner={left,top},     location={right,bottom}]
\definelayerpreset
  [righttop]    [corner={right,top},    location={left,bottom}]
\definelayerpreset
  [leftbottom]  [corner={left,bottom}, location={right,top}]
\definelayerpreset
  [rightbottom] [corner={right,bottom},location={left,top}]
```

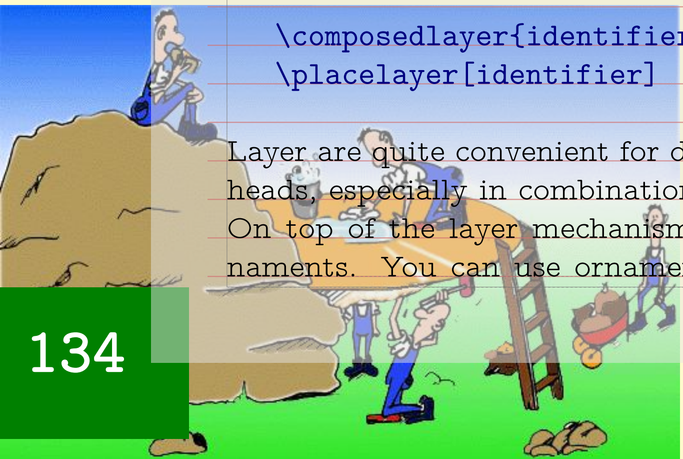
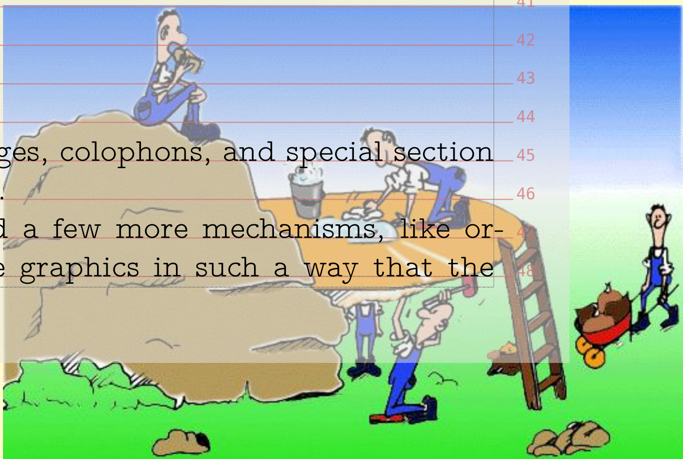
Because for this layer we have also preset the `x` and `y`, those corners are laying a few millimeters outside the page area. We have preset the size as well, otherwise all corners would end up in the top left corner.
We will now fill this layer. Because the layer is hooked into the page, it will be flushed when the page is constructed. After the page is written to the output file, the layer is emptied, unless its `state` is set to `repeat`.

```
\setlayer [extras] [preset=lefttop]     {\externalfigure[hacker]}
\setlayer [extras] [preset=righttop]    {\externalfigure[hacker]}
\setlayer [extras] [preset=leftbottom]  {\externalfigure[hacker]}
\setlayer [extras] [preset=rightbottom] {\externalfigure[hacker]}
```

Once you got the picture of layering, you will start using this mechanism for all kind of tasks. Instead of putting layers in a background, you can also directly place them, by using one of the two (equivalent) commands:

```
\composedlayer{identifier}
\placelayer[identifier]
```

Layer are quite convenient for defining title pages, colophons, and special section heads, especially in combination with `\framed`.
On top of the layer mechanism we have build a few more mechanisms, like ornaments. You can use ornaments to annotate graphics in such a way that the

dimensions stay unchanged.

```
\defineornament
  [affiliation]
  [rotation=90,corner={right,bottom},location={right,top},
   hoffset=-.25ex]
  [frame=on,background=color,backgroundcolor=red,offset=0pt]
```

The negative offset will overlay the text outside the graphic. The meaning of the sign of coordinates and offsets depends on the corner. Figure 7.1 shows the result. We have put the reference point in the right bottom corner. The ornament is anchored at the right top corner of the dot you can picture at the reference point. The ornament is shifted .25ex outwards.

```
\placefigure
  {}
  {\affiliation{graphic}{\externalfigure[hacker][width=3cm]}}
```

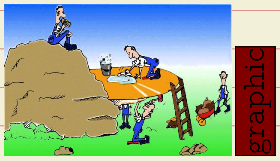


Figure 7.1 Number 1

There are two ways to handle the placement. Alternative `a` will change the dimensions of the graphic according to the size of the ornament, while alternative `b` acts as a pure overlay. In figure 7.2 the ornament is not taken into account when calculating the dimensions of the graphic. This is often the preferred placement, because this way the (often small) ornament will not it will not spoil visual alignment of similar graphics.

```
\defineornament
  [affiliation]
  [rotation=90,corner={right,bottom},location={right,top},
   hoffset=-.25ex,alternative=b]
  [frame=on,background=color,backgroundcolor=red,offset=0pt]
```

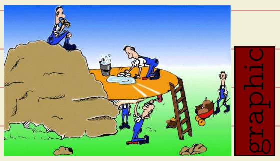


Figure 7.2 Number 2

A positive offset will place the ornament on top of the graphic (see figure 7.3).

```
\defineornament
```

```
  [affiliation]
  [rotation=90,corner={right,bottom},location={left,top},
   hoffset=.25ex,voffset=.25ex,alternative=a]
  [background=color,style=\ss\tfxx,backgroundcolor=white,offset=0pt]
```

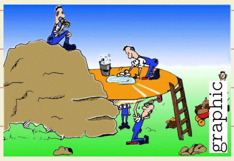


Figure 7.3 Number 3

You need to play a bit with this mechanism in order to get a feeling for what the parameters do.

```
\defineornament
  [affiliation]
  [rotation=90,corner={right,bottom},location={left,top},
   hoffset=.25ex,voffset=.25ex,alternative=b]
  [background=color,style=\ss\tfxx,backgroundcolor=white,offset=0pt]
```

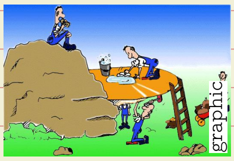


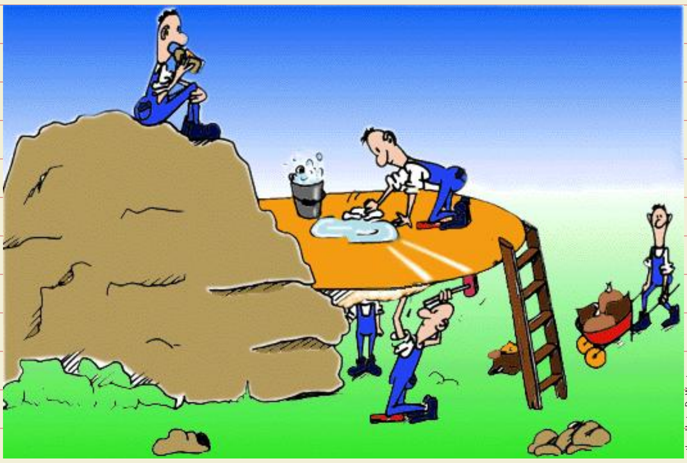
Figure 7.4 Number 4

Because the text is normally typeset quite small, you'd better use a font that can be scaled down a lot.

```
\definefont[AffiliationFont][Sans sa .25]

\defineornament
  [SomeAffiliation]
  [rotation=90,corner={right,bottom},location={right,top},
   hoffset=-.125ex,alternative=b]
  [style=AffiliationFont,offset=0pt]
```

This affiliation is used as:

```
\placefigure
  {Affiliations normally are typeset pretty small.}
  {\SomeAffiliation
     {author: Hester De Weert}
     {\externalfigure[hacker]}}
```

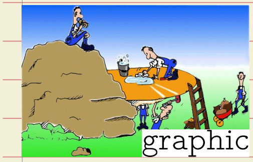


author: Hester De Weert

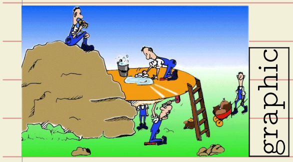
Figure 7.5 Affiliations normally are typeset pretty small.

Ornaments are implemented in terms of layers and collectors. A few examples demonstrate how these can be used.

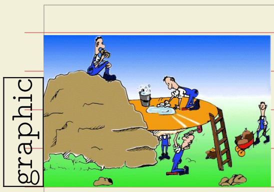
```
\layeredtext
   [corner={right,bottom},location={left,top}]
   [background=color,backgroundcolor=white,offset=0pt]
   {graphic}
   {\externalfigure[hacker][width=3cm]}
```



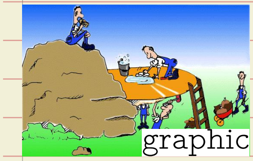
```
\layeredtext
   [rotation=90,corner={right,bottom},location={right,top}]
   [frame=on,offset=0pt]
   {graphic}
   {\externalfigure[hacker][width=3cm]}
```



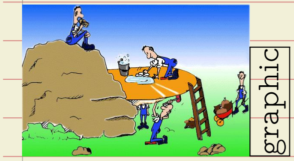
```
\layeredtext
   [rotation=90,corner={left,bottom},location={left,top}]
   [frame=on,offset=0pt]
   {graphic}
   {\externalfigure[hacker][width=3cm]}
```



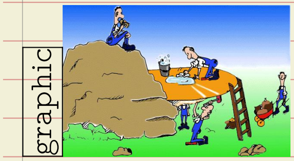
```
\collectedtext
  [corner={right,bottom},location={left,top}]
  [background=color,backgroundcolor=white,offset=0pt]
  {graphic}
  {\externalfigure[hacker][width=3cm]}
```



```
\collectedtext
  [rotation=90,corner={right,bottom},location={right,top}]
  [frame=on,offset=0pt]
  {graphic}
  {\externalfigure[hacker][width=3cm]}
```



```
\collectedtext
  [rotation=90,corner={left,bottom},location={left,top}]
  [frame=on,offset=0pt]
  {graphic}
  {\externalfigure[hacker][width=3cm]}
```



There are several methods to construct title pages, headers, and other compositions. Of course there are the low level box constructors like `\hbox`, `\vbox` and positioning primitives like `\hskip`, `\hfill` and alike.

Another option is to fall back on the low level box macros in the ConTEXt support file `supp-box` or the higher level `\framed` macro. You can use `\framed` nested and by cleverly using the offsets and dimensions you can do a lot.

Layers are another means. You can or instance construct a title page in the following way:

```
\definelayer
  [titlepage]
  [width=\textwidth,
   height=\textheight]

\setlayer
  [titlepage]
  [preset=righttop,location={left,bottom},y=1cm,x=1cm]
  {\definedfont[Regular at 60pt]Welcome}

\setlayer
  [titlepage]
  [preset=rightbottom,location={right,top},y=2cm,x=2cm]
  {\definedfont[Regular at 30pt]By Me}
```

This just fills the layer. Placement is done with:

```
\startstandardmakeup
  \flushlayer[titlepage]
\stopstandardmakeup
```

or alternatively:

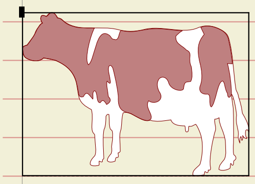
```
\setupbackgrounds[text][background=titlepage]
\startstandardmakeup \stopstandardmakeup
\setupbackgrounds[text][background=]
```

Another way to collect content is to use a collector. A collector starts out empty with:

```
\definecollector[test][state=repeat]
```

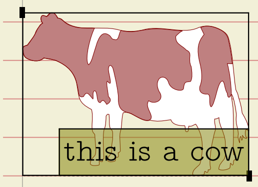
We can now stepwise fill this collector. For educational purposes we've turn of tracing so that you can see what the anchor points.

```
\setcollector[test]
  [location={right,bottom}]
  {\externalfigure[detcow][frame=on,width=3cm]}
```

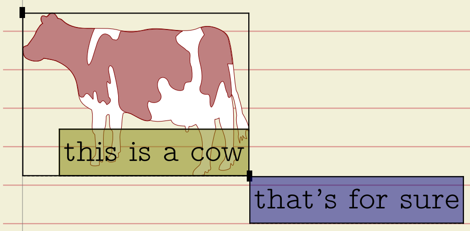


```
\setcollector[test]
```

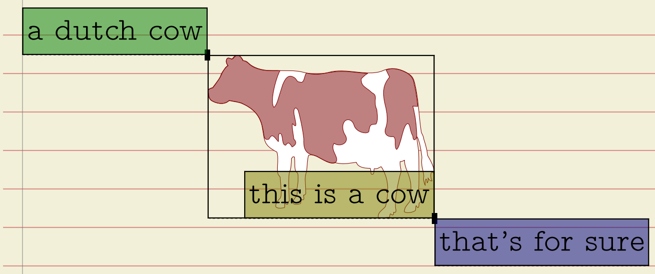
```
  [corner={right,bottom},location={left,top}]
  {\framed[background=color,backgroundcolor=tyellow]{this is a
cow}}
```



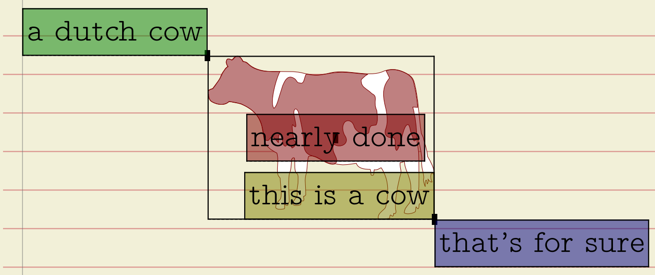
```
\setcollector[test]
  [corner={right,bottom},location={right,bottom}]
  {\framed[background=color,backgroundcolor=tblue]{that's for
sure}}
```



```
\setcollector[test]
  [corner={left,top},location={left,top}]
  {\framed[background=color,backgroundcolor=tgreen]{a dutch cow}}
```



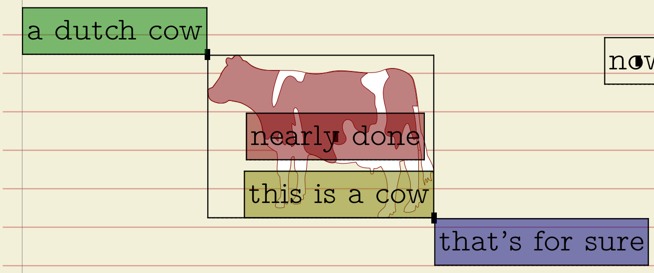
```
\setcollector[test]
  [corner=middle,
   location=middle]
  {\framed[background=color,backgroundcolor=tred]{nearly done}}
```



In addition to the parameters shown here, you can also provide additional ones:

`x`, `y`, `offset`, `hoffset` and `voffset` for positioning and `rotation` for (as expected) rotating the content in steps of 90 degrees. As with layers, the coordinates and offsets can be used intermixed.

```
\setcollector[test]
  [hoffset=4cm,
   voffset=-1cm,
   corner=middle,
   location=middle]
  {\framed{now}}
```



We can show the intermediate results because we have set the state of this collector to repeat. In this case you need to erase the content manually, using:

```
\resetcollector[test]
```

The chapter titles of this document are (as usual in a ConTEXt document) typeset by the `\chapter` macro. When thinking about implementing a non standard head, those familiar with ConTEXt’s head macros will probably first think of using one of the hooks, like:

```
\setuphead[chapter][command=\MyChapterHead]
```

Here we have followed a different approach. First we set up the chapter head. The `empty` directive instructs ConTEXt not to place the head itself, but still to include the associated data in the text stream. This means that we will not see a chapter title, but that there will be an entry in the table of contents, that references will be set up, that so called marks will be available, etc.

```
\setuphead
  [chapter]
  [placehead=empty,
   header=chapter,
   style=\BigText,
   numberstyle=\BigNumber]
```

The `header` parameters instructs the head handler to mark this page as special with regards to header texts. This text is set up as follows:

```
\definetext
```

```
  [chapter]
  [header]
  [\setups{chapter}]
  []
```

The setups are just series of typesetting instructions. For the sake of readability, we have split them up.

```
\startsetups chapter
  \setups[chapter:title]
  \setups[chapter:number]
  \setups[chapter:finish]
\stopsetups
```

The setups will use a dedicated layer for the chapter title:

```
\definelayer
  [chapter]
  [width=\dimexpr\makeupwidth+\cutspace\relax,
   height=\headerheight]
```

The following code uses a macro `\setlayerframed`. This is a combination between `\setlayer` and `\framed`. We use two placement macros to typeset the title and number. When doing so, we need to take care of both numbered chapters and unnumbered titles.

```
\startsetups chapter:title

  \setlayerframed
    [chapter]
    [x=\dimexpr\makeupwidth+\cutspace\relax,location={left,bottom}]
    [height=\headerheight,
     foregroundcolor=white,
     background=color,
     backgroundcolor=blue,
     frame=off,
     offset=none,
     align={right,lohi}]
    {\hbox spread .5\cutspace
       {\hss
        \doiftextelse{\placeheadtext[chapter]}%
          {\placeheadtext[chapter]}%
          {\placeheadtext[title]}%
        \hss}\space
     \vskip.5cm}

\stopsetups
```

Definitions like these may look complicated but in practice you will construct them piece-wise.

```
\startsetups chapter:number

  \setlayerframed
    [chapter]
    [x=\dimexpr\makeupwidth+\cutspace\relax,
     y=\vsize,
     location={left,bottom}]
    [width=\dimexpr\cutspace-\rightmargindistance\relax,
     height=\dimexpr\cutspace-\rightmargindistance\relax,
     foregroundcolor=white,
     background=color,
     backgroundcolor=red,
     frame=off,
     offset=none,
     align={middle,lohi}]
    {\hbox to \hsize
       {\hskip.5cm\hss
        \doifmode{*bodypart}{\placeheadnumber[chapter]}%
        \hss}}

\stopsetups
```

The finishing touch is just a dummy frame with the chapter background. We could have used the header text background instead.

```
\startsetups chapter:finish

  \framed
    [width=\makeupwidth,
     height=\headerheight,
     background=chapter,
     frame=off]
    {}

\stopsetups
```

As the title of this manual suggests: it's in the details. Most of our time is spent in optimizing spacing issues. If you're designing the layout yourself, for a large part you can fall back on the consistent spacing provided by TEX, i.e. think in terms of `em`'s, `ex`'s and fractions or multiples of `\bodyfontsize`, as well as base you're dimensions on those provided by the layout. When dealing with translating a dtp layout into something TEX, definitions like the above will often look more messy.

In this manual we pay quite some words on ways to snap your content on a grid. When dealing with grids, we often run into conflicting situations where we have to make the best of it. Let's again deal with an aspect of graphics.

One of the strong points of TEX is that it can deal with graphics automatically, which means that you seldom have to tweak dimensions or placements unless . . . you're dealing with grids. In that case you need to make sure that the height of graphics consistently match the height of lines (or multiples of lines). It is for this purpose that the graphic inclusion macro has a `grid` entry.

We will illustrate its usage using a dedicated figure class where we have set the space between figure and caption to zero.

```
\definefloat[tightfigure][tightfigures][figure]
\setupcaption[tightfigure][inbetween=]
```

The `grid` parameter controls rounding of the height of a graphic in the following way:

`yes`	safe rounding to an equal number of lines
`fit`	tight rounding to an equal number of lines
`height`	same as `yes` but incremented by linedepth

On the next pages we demonstrate the effects of these settings. At the bottom of a page we show the placement commands. On the last pages we've hidden the captions with:

```
\setupfloat[tightfigure][default={here,none}]
```

As you will notice, the `height` option is handy when the caption is positioned directly under the graphic.

state: unknown

Figure 8.1

state: unknown

Figure 8.2

state: unknown

Figure 8.3

state: unknown

Figure 8.4

state: unknown

Figure 8.5

```
\placetightfigure{}{\externalfigure[dummy][lines=1.3,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.4,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.5,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.6,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.7,grid=yes]}
```

state: unknown

Figure 8.6

state: unknown

Figure 8.7

state: unknown

Figure 8.8

state: unknown

Figure 8.9

state: unknown

Figure 8.10

```
\placetightfigure{}{\externalfigure[dummy][lines=1.3,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.4,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.5,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.6,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.7,grid=fit]}
```

state: unknown

Figure 8.11

state: unknown

Figure 8.12

state: unknown

Figure 8.13

state: unknown

Figure 8.14

state: unknown

Figure 8.15

```
\placetightfigure{}{\externalfigure[dummy][lines=1.3,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.4,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.5,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.6,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.7,grid=height]}
```

state: unknown

state: unknown

state: unknown

state: unknown

state: unknown

```
\placetightfigure{}{\externalfigure[dummy][lines=1.3,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.4,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.5,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.6,grid=yes]}
\placetightfigure{}{\externalfigure[dummy][lines=1.7,grid=yes]}
```

state: unknown

state: unknown

state: unknown

state: unknown

state: unknown

```
\placetightfigure{}{\externalfigure[dummy][lines=1.3,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.4,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.5,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.6,grid=fit]}
\placetightfigure{}{\externalfigure[dummy][lines=1.7,grid=fit]}
```

state: unknown

state: unknown

state: unknown

state: unknown

state: unknown

```
\placetightfigure{}{\externalfigure[dummy][lines=1.3,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.4,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.5,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.6,grid=height]}
\placetightfigure{}{\externalfigure[dummy][lines=1.7,grid=height]}
```

It's hard to predict what kind of caption placements users want. The amount of variation if large and thereby any system of specifying them will look complex. So, examples are the best way to show them.

```
\setupcaption
  [figure]
  [location=bottom]
\placefigure
  [left]
  {}{\externalfigure[dummy][lines=2,width=4cm]}
\fakewords{60}{80} \par
```

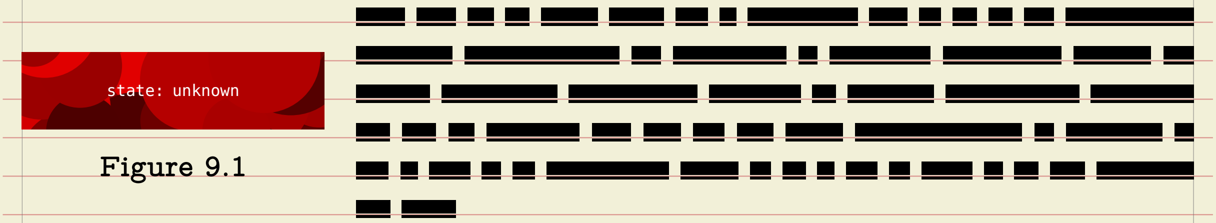
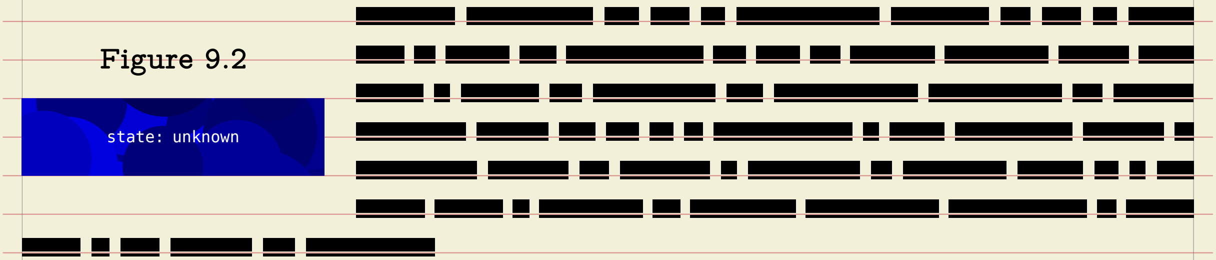


Figure 9.1

```
% \setupfloats[grid=yes]
% \setupfloats[grid=tolerant]
\setupcaption
  [figure]
  [location=top]
\placefigure
  [left]
  {}{\externalfigure[dummy][lines=2,width=4cm]}
\fakewords{60}{80} \par
```

Figure 9.2



In this document we typeset on a grid. For more complex cases and when a document is processed without any user intervention, this is often a bad idea because the snapper can decide to make sure that there is enough space above and below an element. You can however influence the snapper explicitly:

```
\setupcaption
  [figure]
  [location=top]
\placefigure
```

```
    [left,line]
    {}{\externalfigure[dummy][lines=2,width=4cm]}
  \fakewords{60}{80} \par
```

Figure 9.3

state: unknown

Normally a side float plus caption has a normalized (strut) depth while also top skip gets applied. When one of the grid related options `height`, `line`, `depth`, `grid` or `halfline` is given the top skip correction is removed. The `grid` option removes the depth too. The `grid` option removes the depth while the `height` and `depth` options adds an extra amount of strut depth. The `depth` option also adds a line and `halfline` removes a line but adds strut height. Indeed this sounds complicated so best play with it a bit.

Keep in mind that the snapper plays safe and therefore tends to add more space when needed. You can set a grid parameter that controls it:

```
  \setupfloats[grid=tolerant]
```

Currently this only applies to side floats but in the future we might support it for regular floats too.

```
  \setupcaption
    [figure]
    [width=4cm,align=flushright,location={left,high}]
  \placefigure
    [left]
    {}{\externalfigure[dummy][lines=2,width=4cm]}
  \fakewords{60}{80} \par
```

Figure 9.4

state: unknown

```
  \setupcaption
    [figure]
    [width=4cm,align=flushright,location={high,lefthanging}]
```

```
\placefigure
  [left]
  {}{\externalfigure[dummy][lines=2,width=4cm]}
\fakewords{60}{80} \par
```

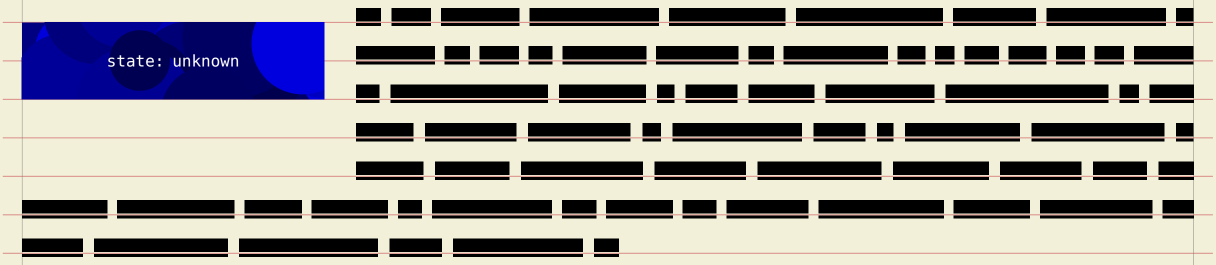


Figure 9.5

```
\setupcaption
  [figure]
  [width=4cm,align=flushleft,location={high,righthanging}]
\placefigure
  [right]
  {}{\externalfigure[dummy][lines=2,width=4cm]}
\fakewords{60}{80} \par
```

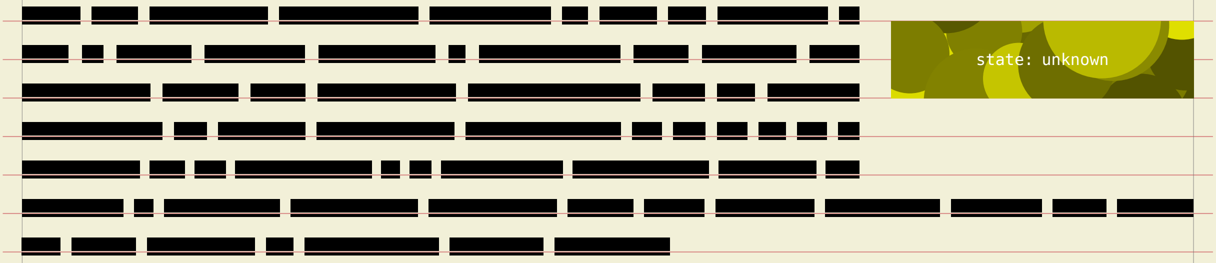


Figure 9.6

```
\setupcaption
  [figure]
  [width=4cm,align=flushleft,location={high,rightmargin}]
\placefigure
  [right]
  {}{\externalfigure[dummy][lines=2,width=4cm]}
\fakewords{60}{80} \par
```

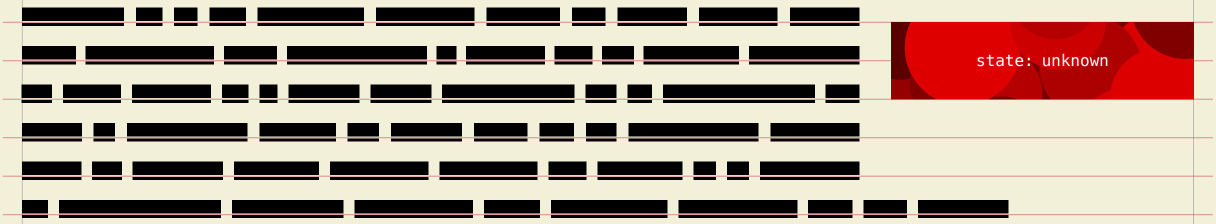


Figure 9.7

The location of a caption is determined by the keywords `top`, `bottom` and for the side captions `high`, `middle`, `low`, either or not in combination with `left`, `right`,

`leftmargin`, `rightmargin`, lefthanging or `righthanging`.
The next series of examples shows the regular (non–side) floats.

```
\setupcaption
  [figure]
  [location={high,left}]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.8

state: unknown

```
\setupcaption
  [figure]
  [width=4cm,align=flushright,location={high,left}]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.9

state: unknown

```
\setupcaption
  [figure]
  [width=4cm,align=flushright,location={middle,left}]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.10

state: unknown

```
\setupfloat
  [figure]
  [location=right]
\setupcaption
  [figure]
  [width=4cm,align=flushright,location=high]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.11

state: unknown

```
\setupfloat
  [figure]
```

```
    [location=right]
  \setupcaption
    [figure]
    [width=4cm,align=flushright,location={high,left}]
  \placefigure
    {}{\externalfigure[dummy][lines=2,width=4cm]}

  \setupfloat
    [figure]
    [location=left]
  \setupcaption
    [figure]
    [width=4cm,align=flushleft, location={high,left}]
  \placefigure
    {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.12 state: unknown

Figure 9.13 state: unknown

```
  \setupfloat
    [figure]
    [location=middle]
  \setupcaption
    [figure]
    [width=4cm,align=flushright, location={high,lefthanging}]
  \placefigure
    {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.14 state: unknown

```
  \setupfloat
    [figure]
    [location=middle]
  \setupcaption
    [figure]
    [width=4cm,align=flushleft,  location={high,righthanging}]
  \placefigure
    {}{\externalfigure[dummy][lines=2,width=4cm]}
```

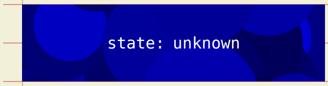
Figure 9.15

```
\setupfloat
  [figure]
  [location=right]
\setupcaption
  [figure]
  [width=4cm,align=flushleft, location={high,rightmargin}]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.1

```
\setupfloat
  [figure]
  [location=left]
\setupcaption
  [figure]
  [width=4cm,align=flushright,location={high,leftmargin}]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.17



```
\setupfloat
  [figure]
  [location=middle]
\setupcaption
  [figure]
  [width=4cm,align=flushright,location={high,outermargin}]
\placefigure
  {}{\externalfigure[dummy][lines=2,width=4cm]}
```

Figure 9.18

```
\setupfloat
  [figure]
  [location=middle]
\setupcaption
  [figure]
```

```
    [width=4cm,align=flushleft, location={high,innermargin}]
  \placefigure
    {}{\externalfigure[dummy][lines=2,width=4cm]}
```

state: unknown

The `innermargin` and `outermargin` are special cases. They adapt to the kind of page.

This document is typeset in ConTEXt using LuaTEX with MetaPost. We use only one font: the Computer Modern Typewriter. The verbatim portions of the text are typeset in its mono spaced variant. One of the reasons that I chose this font is that we need a mono spaced font to typeset the example code, and the Computer Modern Typewriter is one the best there is. This font combines well with many other typefaces, but the sometimes excessive use of different fonts (and sizes) in the styles that I have to implement made me long for simplicity. And so I decided to stick to one font. A careful reader will notice that this document has character protruding enabled (resulting in hanging punctuation).

We use a couple of colors. Again, I went for simplicity and use rather primary colors, although I do use them in transparent variants as well.

There is not much more to say, apart from that I want to thank our customers as well as ConTEXt users for asking me to implement dtp competing styles and features. Their demands drive ConTEXt in directions we could not have foreseen when we started its development.

We use a (transparent) gray background behind the text so that we have an indication where the text area is positioned relative to the page. It also enables us to comfortably turn on the grid.

Some features shown here are relatively new and therefore they occasionally are improved. As a result some aspects of their functionality may change.

CONTEXT

October 15, 2017