

Indian T_EX Users Group

URL: <http://www.river-valley.com/tug>



On-line Tutorial on L^AT_EX

The Tutorial Team

Indian T_EX Users Group, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
2000

Prof. (Dr.) K. S. S. Nambooripad, *Director, Center for Mathematical Sciences, Trivandrum, (Editor);*
Dr. E. Krishnan, *Reader in Mathematics, University College, Trivandrum; T. Rishi, Focal Image (India) Pvt. Ltd.,*
Trivandrum; L. A. Ajith, Focal Image (India) Pvt. Ltd., Trivandrum, A. M. Shan, Focal Image (India) Pvt. Ltd.,
Trivandrum; C. V. Radhakrishnan, River Valley Technologies, Software Technology Park, Trivandrum constitute
the Tutorial team

*This document is generated from L^AT_EX sources compiled with pdfL^AT_EX v. 14e in
an INTEL Pentium III 700 MHz system running Linux kernel version 2.2.14-12.
The packages used are hyperref.sty and pdfscreen.sty*

©2000, Indian T_EX Users Group. This document may be distributed under the terms of
the L^AT_EX Project Public License, as described in lppl.txt in the base L^AT_EX distribution,
either version 1.0 or, at your option, any later version



5 Several Kinds of Boxes

The theory of composing pages out of boxes lies at the very heart of \TeX and many \LaTeX constructs are available to take advantage of this method of composition.

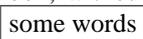
A *box* is an object that is treated by \TeX as a single character. A box cannot be split and broken across lines or pages. Boxes can be moved up, down, left and right. \LaTeX has three types of boxes.

LR (left-right)—The content of this box are typeset from left to right.

Par (paragraphs)—This kind of box can contain several lines, which will be typeset in paragraph mode just like normal text. Paragraphs are put one on top of the other. Their widths are controlled by a user specified value.

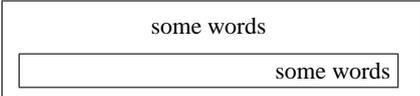
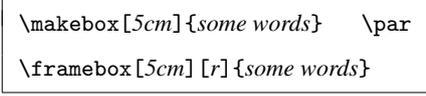
Rule A thin or thick line that is often used to separate various logical elements on the output page, such as between table rows and columns and between running titles and the main text.

5.1 LR Boxes

The usage information of four types of LR boxes are given below. The first line considers the *text* inside the curly braces as a box, without or with a frame drawn around it. For instance, `\fbox{some words}` gives  whereas `\mbox` will do the same thing, but without the ruled frame around the text.

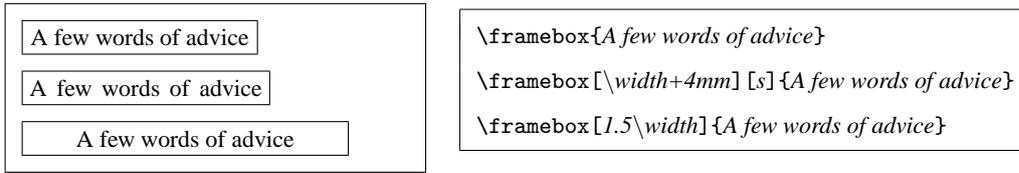
```
\mbox{text}
\makebox[width][pos]{text}
\fbox{text}
\framebox[width][pos]{text}
```

The commands in the third and fourth lines are a generalization of the other commands. They allow the user to specify the width of the box and the positioning of text inside.

| | |
|---|--|
|  |  |
|---|--|

In addition to the centering the text with positional argument `[c]` (the default), you can position the text flush left (`[l]`). \LaTeX also offers you an `[s]` specifier that will stretch your text from the left margin to the right margin of the box provided it contains some stretchable space. The inter-word space is also stretchable and shrinkable to a certain extent.

With \LaTeX , the above box commands with arguments for specifying the dimensions of the box allow you to make use of four special length parameters: `\width`, `\height`, `\depth` and `\totalheight`. They specify the natural size of the text, where `\totalheight` is the sum of the `\height` and `\depth`.



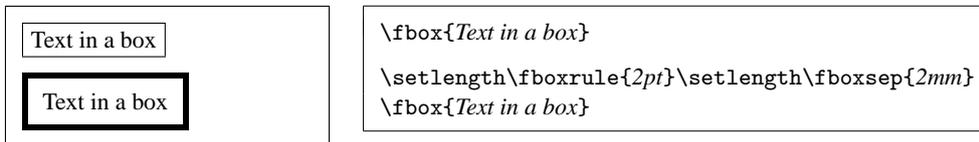
⇔ As seen in the margin of the current line, boxes with zero width can be used to make text stick out in the margin. This effect was produced by beginning the paragraph as follows:

```
\makebox[0mm][r]{\color{red}$\Leftrightarrow$}
As seen in the margin of the . . .
```

The appearance of frameboxes can be controlled by two style parameters.

\fboxrule The width of the lines comprising the box produced with the command `\fbox` or `\framebox`. The default value in all standard classes is 0.4pt.

\fboxsep The space left between the edge of the box and its contents by `\fbox` or `\framebox`. The default value in all standard classes is 3pt.



Another interesting possibility is to raise or lower boxes. This can be achieved by the very powerful `\raisebox` command, which has two obligatory and two optional parameters, defined as follows:

```
\raisebox{lift}[depth][height]{contents}
```

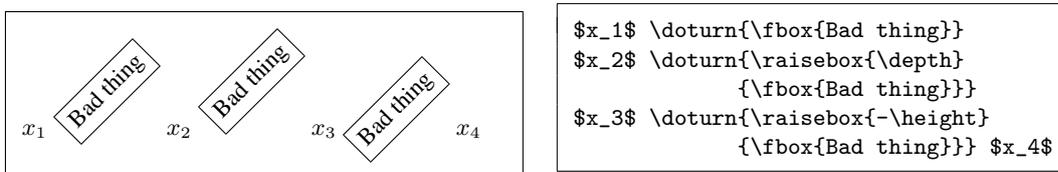
Given below is an example of lowered and elevated text boxes.



As with `\makebox` and `\framebox` the \LaTeX implementation of `\raisebox` offers you the use of the lengths `\height`, `\depth`, `\totalheight` and `\width` in the first three arguments. Thus, to pretend that a box extends only 90% of its actual height above the baseline you could write:

```
\raisebox{0pt}[0.9\height]{text}
```

or to rotate a box around its lower left corner (instead of its reference point lying on the baseline), you could raise it by its `\depth` first, e.g.:



5.2 Paragraph Boxes

Paragraph boxes are constructed using the `\parbox` command or `minipage` environment. The text material is typeset in paragraph mode inside a box of width *width*. The vertical positioning of the box with respect to the text baseline is controlled by the one-letter optional parameter *pos* (`[c]`, `[t]`, and `[b]`).

```
\parbox[pos]{width}{text}
```

is the usage for `\parbox` command, whereas that of the `minipage` environment will look like:

```
\begin{minipage}[pos]{width}
... here goes the text matter ...
\end{minipage}
```

The center position is the default as shown by the next example. You can also observe that \LaTeX might produce wide inter-word spaces if the measure is incredibly small.

This is the contents of the left-most parbox.

CURRENT LINE

This is the right-most parbox. Note that the typeset text looks sloppy because \LaTeX cannot nicely balance the material in these narrow columns.

The code for generating these three `\parbox`'s in a row is given below:

```
\parbox{.3\linewidth}
{This is the contents of the left-most parbox.}
\hfill CURRENT LINE \hfill
\parbox{.3\linewidth}
{This is the right-most parbox. Note that the typeset
text looks sloppy because \LaTeX{} cannot nicely balance
the material in these narrow columns.}
```

The `minipage` environment is very useful for the placement of material on the page. In effect, it is a complete mini-version of a page and can contain its own footnotes, paragraphs, and `array`, `tabular` and `multicols` (we will learn about these later) environments. A simple example of `minipage` environment at work is given below. The baseline is indicated with a small red line.

```

\begin{minipage}[b]{.3\linewidth}
  The minipage environment creates a vertical box
  like the parbox command. The bottom line of this
  minipage is aligned with the
\end{minipage}\hrulefill
\begin{minipage}[c]{.3\linewidth}
  middle of this narrow parbox, which in turn is
  aligned with
\end{minipage}\hrulefill
\begin{minipage}[t]{.3\linewidth}
  the top line of the right hand minipage. It is recommended
  that the user experiment with the positioning arguments to
  get used to their effects.
\end{minipage}

```

The minipage environment creates a vertical box like the `parbox` command. The bottom line of this minipage is aligned with the middle of this narrow `parbox`, which in turn is aligned with the top line of the right hand minipage. It is recommended that the user experiment with the positioning arguments to get used to their effects.

5.3 Paragraph boxes with specific height

In \LaTeX , the syntax of the `\parbox` and `minipage` has been extended to include two more optional arguments.

```
\parbox[pos][height][inner pos]{width}{text}
```

is the usage for `\parbox` command, whereas that of the `minipage` environment will look like:

```

\begin{minipage}[pos][height][inner pos]{width}
  ... here goes the text matter ...
\end{minipage}

```

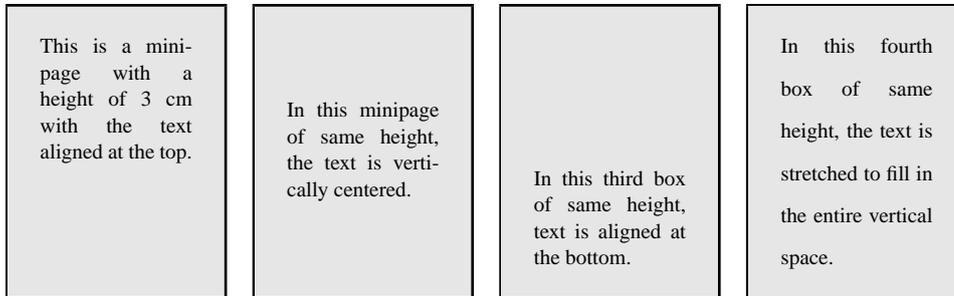
In both cases, *height* is a length specifying the height of the box; the parameters `\height`, `\width`, `\depth`, and `\totalheight` may be employed within the *emph* argument in the same way as in the *width* argument of `\makebox` and `\framebox`.

The optional argument *inner pos* states how the text is to be positioned internally, something that is only meaningful if *height* has been given. Its possible values are:

- t** to push the text to the top of the box
- b** to shove it to the bottom
- c** to center it vertically
- s** to stretch it to fill up the whole box

In the last case, rubber lengths (glue element) should be present where the vertical stretching is to take place.

Note the difference between the external positioning argument *pos* and the internal one *inner pos*: the former states how the box is to be aligned with the surrounding text, while the latter determines how the contents are placed within the box itself. See an example below. We frame the minipages to make it more comprehensive.



See the code that generated above boxed material:

```
\fbox{%
  \begin{minipage}[b][3cm][t]{2cm}
    This is a minipage with a height of 3~cm with the text aligned
    at the top.
  \end{minipage}}\hfill
\fbox{%
  \begin{minipage}[b][3cm][c]{2cm}
    In this minipage of same height, the text is vertically centered.
  \end{minipage}}\hfill
\fbox{%
  \begin{minipage}[b][3cm][b]{2cm}
    In this third box of same height, text is aligned at the bottom.
  \end{minipage}}\hfill
\fbox{%
  \begin{minipage}[b][3cm][s]{2cm}
    \baselineskip 10pt plus 2pt minus 2pt
    In this fourth box of same height, the text is stretched
    to fill in the entire vertical space.
  \end{minipage}}
```

5.4 Nested boxes

The box commands described above may be nested to any desired level. Including an LR box within a parbox or a minipage causes no obvious conceptual difficulties. The opposite, a parbox within an LR box, is also possible, and is easy to visualize if one keeps in mind that every box is a unit, treated by T_EX as a single character of the corresponding size.

A parbox inside an `\fbox` command has the effect that the entire parbox is framed. The present structure was made with

```
\fbox{\fbox{\parbox{.75\linewidth}{A parbox ...}}}
```

This is a parbox of width `.75\linewidth` inside a `fbox` inside a second `fbox`, which thus produces the double framing effect.

5.5 Rule boxes

A rule box is a basically a filled-in black rectangle. The syntax for the general command is:

```
\rule[lift]{width}{height}
```

which produces a solid rectangle of width *width* and height *height*, raised above the baseline by an amount *lift*. Thus `\rule{8mm}{3mm}` generates  and `\rule{3in}{.2pt}` generates .

Without an optional argument *lift*, the rectangle is set on the baseline of the current line of the text. The parameters *lift*, *width* and *height* are all lengths. If *lift* has a negative value, the rectangle is set below the baseline.

It is also possible to have a rule box of zero width. This creates an invisible line with the given *height*. Such a construction is called a *strut* and is used to force a horizontal box to have a desired height or depth that is different from that of its contents.