

LaTeX/Boxes

LaTeX builds up its pages by pushing around boxes. At first, each letter is a little box, which is then glued to other letters to form words. These are again glued to other words, but with special glue, which is elastic so that a series of words can be squeezed or stretched as to exactly fill a line on the page.

Admittedly, this is a very simplistic description of what really happens, but the point is that TeX operates with glue and boxes. Letters are not the only things that can be boxes. One can put virtually everything into a box, including other boxes. Each box will then be handled by LaTeX as if it were a single letter.

The past chapters have already dealt with some boxes, although they weren't described as such. The `tabular` environment and the `\includegraphics`, for example, both produce a box. This means that one can easily arrange two tables or images side by side. You just have to make sure that their combined width is not larger than the `\textwidth`.

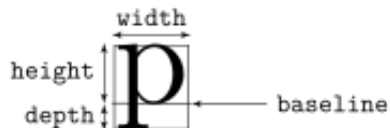
Contents

- 1 TeX character boxes
- 2 `makebox` and `mbox`
- 3 `framebox`
- 4 `framed`
- 5 `raisebox`
- 6 `minipage` and `parbox`
 - 6.1 Paragraphs in all boxes
- 7 `savebox`
- 8 `rotatebox`
- 9 `colorbox` and `fcolorbox`
- 10 `resizebox` and `scalebox`
- 11 `fancybox`

TeX character boxes

TeX characters are stored in boxes like every printed element. Boxes have three dimensional properties:

- The *height* is the length between the baseline and the top of the box.
- The *depth* is the length between the baseline and the bottom of the box.
- The *width* is the width of the box.



makebox and mbox

While `\parbox` packs up a whole paragraph doing line breaking and everything, there is also a class of boxing commands that operates only on horizontally aligned material. We already know one of them; it's called `\mbox`. It simply packs up a series of boxes into another one, and can be used to prevent LaTeX from breaking two words. (See Hyphenation.) As you can put boxes inside boxes, these horizontal box packers give you ultimate flexibility.

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

width defines the width of the resulting box as seen from the outside. This means it can be smaller than the material inside the box. You can even set the width to `0pt` so that the text inside the box will be typeset without influencing the surrounding boxes. Besides the length expressions, you can also use `\width`, `\height`, `\depth` and `\totalheight` in the *width* parameter. They are set from values obtained by measuring the typeset text. The *pos* parameter takes a one letter value: `center`, `flushleft`, `flushright`, or `spread` the text to fill the box.

```
\makebox[0pt]{Some text} over this text

\makebox[15ex][s]{Censored text}\hspace{-15ex}\makebox[15ex][s]{X X X X X}

Text \makebox[2\width][r]{running away}
```

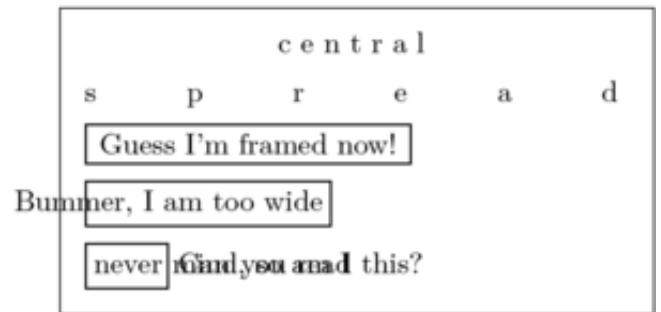
framebox

The command `\framebox` works exactly the same as `\makebox`, but it draws a box around the text.

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

The following example shows you some things you could do with the `\makebox` and `\framebox` commands:

```
\makebox[\textwidth]{c e n t r a  
l} \par  
\makebox[\textwidth][s]{s p r e a  
d} \par  
\framebox[1.1\width]{Guess I'm  
framed now!} \par  
\framebox[0.8\width][r]{Bummer, I  
am too wide} \par  
\framebox[1cm][l]{never mind, so  
am I}  
Can you read this?
```



You can tweak the following frame lengths.

- `\fboxsep`: the distance between the frame and the content.
- `\fboxrule`: the thickness of the rule.

This prints a thick and more distant frame:

```
\setlength{\fboxsep}{10pt}  
\setlength{\fboxrule}{5pt}  
\fbox{A frame.}
```

This shows the box frame of a letter.

```
\setlength{\fboxsep}{0pt}  
\fbox{A}
```

framed

An alternative to these approaches is the usage of the `framed` environment (you will need to include the `framed` package to use it). This provides an easy way to box a paragraph within a document:

```

\usepackage{framed}
% ...

\begin{framed}
This is an easy way to box text within a document!
\end{framed}

```

You can do it manually with a parbox.

raisebox

Now that we control the horizontal, the obvious next step is to go for the vertical. No problem for LaTeX. The

```

\raisebox{lift}[height][depth]{text}

```

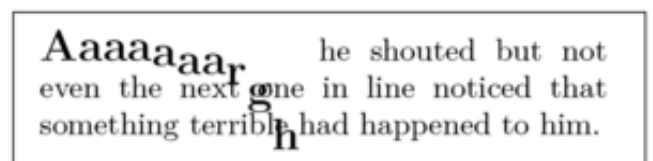
command lets you define the vertical properties of a box. You can use `\width`, `\height`, `\depth` and `\totalheight` in the first three parameters, in order to act upon the size of the box inside the text argument. The two optional parameters set for the height and depth of the raisebox. For instance you can observe the difference when embedded in a framebox.

```

\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}
{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}
}
}

```

he shouted but not even the next
one in line noticed that
something
terrible had happened to him.



minipage and parbox

Most standard LaTeX boxes are not *long* commands, *i.e.* they do not support breaks nor paragraphs. However you can pack a paragraph of your choice into a box with either the `\parbox[pos][height][contentpos]{width}{text}` command or the `\begin{minipage}`

`[pos][height][contentpos]{width} text \end{minipage}` environment.

The `pos` parameter can take one of the letters `center`, `top` or `bottom` to control the vertical alignment of the box, relative to the baseline of the surrounding text. The `height` parameter is the height of the parbox or minipage. The `contentpos` parameter is the position of the content and can be one of `center`, `top`, `bottom` or `spread`. `width` takes a length argument specifying the width of the box. The main difference between a `minipage` and a `\parbox` is that you cannot use all commands and environments inside a parbox, while almost anything is possible in a minipage.

```
\noindent
\fbbox{\parbox[b][4em][t]{0.33\textwidth}{Some \text{ } text} }
\fbbox{\parbox[c][4em][s]{0.33\textwidth}{Some \vfill text} }
\fbbox{\parbox[t][4em][c]{0.33\textwidth}{Some \text{ } text} }
```

This should print 3 boxes on the same line. Do not put another linebreak between the `\fbbox`, otherwise you will put the following `\fbbox` in another paragraph on another line.

Paragraphs in all boxes

You can make use of the *long* capabilities of minipage and parbox to embed paragraphs in non-long boxes. For instance:

```
\fbbox{
  \parbox{\textwidth}{
    Some very long text...
  }
}
```

This prevents the overfull badness.

You can also use

```
\pbox{\textwidth}{my text}
```

from the `pbox` package which will create a box of minimal size around the text. Note that the `\pbox` command takes an optional argument that specifies the vertical position of the text:

```
\pbox[b]{\textwidth}{my text}
```

The valid values are b (bottom), t (top), and c (center). If you specify a length in the first (required) argument, the text will be wrapped:

```
\pbox[b]{5cm}{This is long text that will be wrapped once it reaches five
centimeters.}
```

savebox

A `\savebox` is a reference to a box filled with contents. You can use it as a way to print or manipulate something repeatedly.

```
\newsavebox{\boxname}
\savebox{\boxname}{some content}
\usebox{\boxname}
```

The command `\newsavebox` creates a placeholder for storing a text; the command `\savebox` stores the specified text in this placeholder, and does not display anything in the document; and `\usebox` recalls the content of the placeholder into the document.

rotatebox

See Rotations.

colorbox and fcolorbox

See Colors. `\fcolorbox` can also be tweaked with `\fboxsep` and `\fboxrule`.

resizebox and scalebox

The `graphicx` package feature additional boxes.

```
\resizebox{10ex}{2\baselineskip}{Dunhill style}
\scalebox{10}{Giant}
```

fancybox

the `fancybox` package provides additional boxes.

- `\doublebox`
- `\ovalbox`
- `\shadowbox`

Retrieved from "<http://en.wikibooks.org/w/index.php?title=LaTeX/Boxes&oldid=2739393>"

-
- This page was last modified on 1 December 2014, at 17:03.
 - Text is available under the Creative Commons Attribution-ShareAlike License.; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.