LaTeX Course 2011
Part 3: Typographical commands

Arho Virkki

VTT Technical Research Centre of Finland

Elements

The non-textual elements, including

- boxes (`\makebox, \framebox ...`)
- lines (`\rule `)
- distances (`\hspace, \hspace*, \vspace, ...`)
- …

are described with commands inside the document.

Their arguments are given in millimeters, centimeters, pixels, or in proportion of the font 'M' or 'x' characters,

10mm, 1.5cm, 12px, 1em, 0.8ex, …

Examples

Let us inspect some of the most common commands:

- `\rule[voffset]{width}{height}`
  e.g.: `Baseline text\rule[-5pt]{2cm}{1ex}`:
  Baseline text.

- `\framebox[width][pos]{text}`
  e.g.: `\framebox[9cm][s]{foo bar baz}`:

  foo bar baz

  where s=stretch.The other options are l=left and r=right

Examples. . .

- `\makebox[width][pos]{text}`
  is the same as `framebox`, but with no borders
- `\fbox{text}`
  is a short form: `\fbox{Inspect this!}`
  $\boxed{\text{Inspect this!}}$
- `\mbox{text}`
  The invisible box `\mbox{the contents}`
  $\mapsto$ the contents is treated as a single entity. This property is useful, and will be needed many times in the future. . .

Examples...

- `\vspace*{height} ja \hspace*{width}` `\hspace*{0.3\textwidth}` adds certain amount of space to the documents[1][2]. Vertical space can be added with `\vspace`. (Negative distances move the following contents to the left or up. The command with star is enforcing, otherwise adding the space is optional, and will be omitted at certain places, e.g. at the end of paragraph)
- `\parbox[pos][height][ipos]{width}{text}` produces a box. For example,

```
The \fbox{\parbox[b][1cm][b]{4cm}{Schrödinger's cat}
```

The $\boxed{\text{Schrödinger's cat}}$

[1] Observe the relative distance measure `\textwidth`
[2] Footnotes can be produces with `\footnote...`

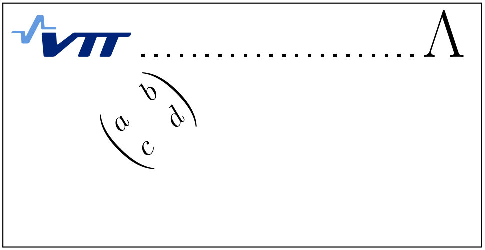
Examples...

- `pos` can be t=top (line) or b=bottom (line)
- `ipos` can be t=top, b=bottom, c=centered or s=stretched
- `parbox` behaves like a small embedded page[3], end it is treated as a single element.

[3] `\minipage` will behave exactly like an embedded page...

Examples. . .

```
\fbox{\parbox[t][3cm][t]{6cm}{%
    \includegraphics[width=1.6cm]
    {img/vttplain} \dotfill {\Huge $\Lambda$} \\
    \hspace*{1cm}\rotatebox{45}{%
      $\scriptstyle\begin{pmatrix}
        a & b \\ c & d \end{pmatrix}$}}
}
```



Examples. . .

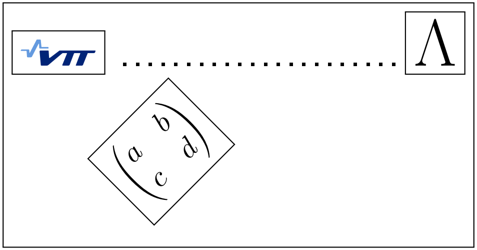
What just happened?

- `\fbox` included borders to the box (in contrast to mbox)
- `\includegraphics` included the image (as pdf or png)
- `\dotfill` fills the area with dots
- `\hspace*` moved the «pencil« horizontally

The following shows the same example with all elements having a box around them.

Examples. . .

Now we see what happens



The only difference is that the borders take some space, which makes a small difference in the image.

Conclusions

We saw that LaTeX code becomes complicated, if we construct detailed elements.
Because of this reason, the logical content of the document and fiddling with the details should be kept **separated.**
In summary,

- everything can be tuned,
- it is not wise to tune everything,

but from experience I suppose that you would like to change the default font sooner or later. . .

Fonts

The easiest way to get started with different document styles is to load a pre-defined style package (from a `.sty` file) in the manuscript preamble:

```
\usepackage{mystyle}
```

where the style is possibly one from the list `helvet`, `palatino`, `avant`, `charter`, `bookman`, `newcent` or `times` (that are probably already installed in your LaTeX-distribution package).

The size of the font can be changed with the descriptive commands `{\large }` and `{\small }`.

Fonts. . .

`{\tiny Rabbit}`	Rabbit
`{\scriptsize Rabbit}`	Rabbit
`{\footnotesize Rabbit}`	Rabbit
`{\small Rabbit}`	Rabbit
`{\normalsize Rabbit}`	Rabbit
`{\large Rabbit}`	Rabbit
`{\Large Rabbit}`	Rabbit
`{\LARGE Rabbit}`	Rabbit
`{\huge Rabbit}`	Rabbit
`{\Huge Rabbit}`	Rabbit

Fonts. . .

Observations

- These commands are declarations, not functions: `\large` = change to large font and keep in until the end of this (scope or) environment.
- Anything between the `\begin{}` . . . `\end{}` -block is an environment. A new environment can also be defined with the curly braces { . . . }.
- Use the curly braces and think of the environments. This is better style (from the programmer's point of view) than scattering the document with commands like `\normalsize` without a clear reference to the beginning of the "non-normal" behaviour.

Fonts. . .

I admit that even the last one `\Huge` was not *that* large:

Q: How to do a HUGE Rabbit?

A: `\scalebox{4}{Rabbit}`

Rabbit

Fonts. . .

In addition, there are the ordinary functions to decorate the text

`\texttt{Rabbit}`	`Rabbit`
`\textit{Rabbit}`	*Rabbit*
`\textbf{Rabbit}`	**Rabbit**
`\textsl{Rabbit}`	*Rabbit*
`\textsc{Rabbit}`	RABBIT
`\underline{Rabbit}`	$\underline{\text{Rabbit}}$

Fonts. . .

These commands suffice well for a sophisticated writer.

More fonts can be found from the LaTeX-distributions by searching the `.fd` (font definition) files. For example, `t1pbk.fd` (T1 encoded postscript Bookman font definition) can be selected with the definition

```
\fontfamily{pbk}\selectfont
This is Bookman Font
```

This is bookman Font

Fonts…

If you really need to change the font all the time (which is, in general, a bad idea) , it may be advantageous to define a custom command in the beginning of the document :

```
\newcommand{\avantgar}[1]{
{\fontfamily{pag}\selectfont #1}}
```

etc…

```
\newcommand{\bookman}[1]{{\fontfamily{pbk}\selectfont #1}}
\newcommand{\courier}[1]{{\fontfamily{pcr}\selectfont #1}}
\newcommand{\cmodern}[1]{{\fontfamily{cmr}\selectfont #1}}
\newcommand{\helvetic}[1]{{\fontfamily{phv}\selectfont #1}}
\newcommand{\newcent}[1]{{\fontfamily{pnc}\selectfont #1}}
\newcommand{\tmroman}[1]{{\fontfamily{ptm}\selectfont #1}}
\newcommand{\script}[1]{{\fontfamily{pzc}\selectfont #1}}
```

Fonts. . .

Now we have defined a couple of commands that fit the LaTeX philosophy.

`\avantgar{Rabbit}`	Rabbit
`\bookman{Rabbit}`	Rabbit
`\courier{Rabbit}`	Rabbit
`\cmodern{Rabbit}`	Rabbit
`\helvetic{Rabbit}`	Rabbit
`\tmroman{Rabbit}`	Rabbit
`\script{Rabbit}`	*Rabbit*

These commands are not included by default, because they will most likely ruin your (technical or scientific) document. For the purposes you would be using them, some of the office packages might be a better option.

Conclusions. . .

A good strategy for writing a paper with LaTeX is to

- Make the title page by hand (e.g. by using the command just learned)
- Write the text with clean LaTeX, using perhaps some of the extension packages from AMS (American Mathematical Society) to stick with portability.
- Make your own style sheet, if necessary
- Include minor things in the document preamble before the `\begin{document}` clause (even though it is sometimes legal to define command later in the document — but what is the advantage?)

Conclusions. . .

- LaTeX and plain TeX spaghetti mixed with the text is a nightmare to maintain. Recall the experiences from HTML 3.2 before properly working CSS.
- LaTeX is a large software with a very long history (starting from 1978). Knowing the basics suffices well.