# OpTEX

Format Based on Plain \TeX{} and OPmac\(^1\)

Petr Olšák, 2020

[http://petr.olsak.net/optex](http://petr.olsak.net/optex)

## Contents

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Starting with OpTEX</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Page layout</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Setting the margins</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Concept of default page</td>
<td>3</td>
</tr>
<tr>
<td>2.3</td>
<td>Footnotes and marginal notes</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Fonts</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>Font families</td>
<td>4</td>
</tr>
<tr>
<td>3.2</td>
<td>Font sizes</td>
<td>5</td>
</tr>
<tr>
<td>3.3</td>
<td>Typesetting math</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Typical elements of document</td>
<td>6</td>
</tr>
<tr>
<td>4.1</td>
<td>Chapters and sections</td>
<td>6</td>
</tr>
<tr>
<td>4.2</td>
<td>Another numbered objects</td>
<td>6</td>
</tr>
<tr>
<td>4.3</td>
<td>References</td>
<td>7</td>
</tr>
<tr>
<td>4.4</td>
<td>Hyperlinks, outlines</td>
<td>8</td>
</tr>
<tr>
<td>4.5</td>
<td>Lists</td>
<td>8</td>
</tr>
<tr>
<td>4.6</td>
<td>Tables</td>
<td>9</td>
</tr>
<tr>
<td>4.7</td>
<td>Verbatim</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Autogenerated lists</td>
<td>12</td>
</tr>
<tr>
<td>5.1</td>
<td>Table of contents</td>
<td>12</td>
</tr>
<tr>
<td>5.2</td>
<td>Making the index</td>
<td>12</td>
</tr>
<tr>
<td>5.3</td>
<td>Bib\TeX{}ing</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>Graphics</td>
<td>15</td>
</tr>
<tr>
<td>6.1</td>
<td>Colors</td>
<td>15</td>
</tr>
<tr>
<td>6.2</td>
<td>Images</td>
<td>15</td>
</tr>
<tr>
<td>6.3</td>
<td>PDF transformations</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Others</td>
<td>16</td>
</tr>
<tr>
<td>7.1</td>
<td>Using more languages</td>
<td>16</td>
</tr>
<tr>
<td>7.2</td>
<td>Pre-defined styles</td>
<td>17</td>
</tr>
<tr>
<td>7.3</td>
<td>Lorem ipsum dolor sit</td>
<td>17</td>
</tr>
<tr>
<td>7.4</td>
<td>The last page</td>
<td>17</td>
</tr>
<tr>
<td>7.5</td>
<td>Use OpTEX</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>Summary</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>Compatibility with Plain \TeX{}</td>
<td>19</td>
</tr>
</tbody>
</table>

\(^1\) The OPmac package is a set of simple additional macros to plain \TeX{}. It enables users to take advantage of basic \H\TeX{} functionality but keeps plain \TeX{} simplicity. See [http://petr.olsak.net/opmac-e.html](http://petr.olsak.net/opmac-e.html) for more information about it. For OPmac users: the red triangle ◀ in the right margin means that there is a difference from standard OPmac features.
Introduction

Opt\TeX{} is Lua\TeX{} format with plain \TeX{} and OPmac. Only Lua\TeX{} engine is supported. The main goal of Opt\TeX{} is:

- Opt\TeX{} keeps the simplicity (like in plain \TeX{} and OPmac macros).
- There is no old obscurities concerning with various 8-bit encodings and various engines.
- Opt\TeX{} provides a powerful font selection system (for Unicode font families, of course).
- Opt\TeX{} supports hyphenations of all languages installed in your \TeX{} system.
- All features from OPmac macros are copied.
- Macros are documented in the same place where code is (macros for printing this documentation will come in the future).
- User name space of control sequences is separated from internal name space of Opt\TeX{} and primitives (\texttt{\_foo} versus \texttt{\_foo}).

Opt\TeX{} should be a modern plain \TeX{} with power from OPmac (fonts selection system, colors, external graphics, references, hyperlinks, indexing, bibliography, ...) with preferred Unicode fonts.

If you need to customize your document or you need to use something very specific, then you can copy relevant parts of Opt\TeX{} macros into your macro file and do changes of these macros here. This is significant difference from \LaTeX{} or Con\TeXt{}, which are an attempt to create a new user level with a plenty of non-primitive parameters and syntax hiding \TeX{} internals. The macros from Opt\TeX{} are simple and straightforward because they solves only what is explicitely needed, they does not create a new user level over \TeX{}. And you can use them, understand them an modify them.

Opt\TeX{} offers a markup language for authors of texts (like \LaTeX{}), i.e. the fixed set of tags to define the structure of the document. This markup is different from the \LaTeX{} markup. It may offer to write the source text of the document somewhat clearer and more attractive.

Disclaimer: This software is under construction. It is possible that some features documented here will be changed in future. There exists a large part of original OPmac macros included in Opt\TeX{} which waits to its re-implementation at current state of development.

1 Starting with Opt\TeX{}

Opt\TeX{} is compiled as a format for Lua\TeX{}. Maybe there is a command \texttt{optex} in your \TeX{} distribution. Then you can write into command line

\begin{verbatim}
  optex document
\end{verbatim}

You can try to process \texttt{optex op-demo} or \texttt{optex optex-doc}.

If there is no \texttt{optex} command, see more information about installation Opt\TeX{} at \url{http://petr.olsak.net/optex}.

A minimal document should be

\begin{verbatim}
  \fontfam[LMfonts]
  Hello World! \bye
\end{verbatim}

The first line \texttt{\fontfam[LMfonts]} tells that Unicode Latin Modern fonts (derived from Computer Modern) are used. If you omit this line then preloaded Latin Modern fonts are used but preloaded fonts cannot be in Unicode\(^2\). So the sentence \texttt{Hello World} will be OK without the first line, but you cannot print such sentence in another languages (for example \texttt{Ahoj světe!}) where Unicode fonts are needed because of the characters like \texttt{è} are not mapped correctly in preloaded fonts.

A somewhat larger example with common settings should be:

\begin{verbatim}
  \fontfam[Termes]  % selecting Unicode font family Termes
  \typosize[11/13]  % setting the basic font size and the baselineskip
  \margins/1 a4 (1,1,1,1)in  % setting lin margins for A4 paper
  \cslang           % Czech hyphenation patterns

  Tady je text.
  \bye
\end{verbatim}

You can look at \texttt{op-demo.tex} file for more examples.

\(^2\) This is technical limitations of Lua\TeX{} for fonts downloaded in formats: only 8bit fonts can be preloaded.
\section*{2 Page layout}

\subsection*{2.1 Setting the margins}

\LaTeX{} declares paper formats a4, a4l (landscape a4), a5, a5l, b5, letter and user can declare another own format by \texttt{\textbackslash def}:

\begin{verbatim}
\sdef{pgs:b5l}{(250,176)mm}
\sdef{pgs:letterl}{(11,8.5)in}
\end{verbatim}

The \texttt{\margins} command declares margins of the document. This command have the following parameters:

\begin{verbatim}
\margins/\langle\texttt{pg}\rangle \langle\texttt{fmt}\rangle (\langle\texttt{left}\rangle,\langle\texttt{right}\rangle,\langle\texttt{top}\rangle,\langle\texttt{bot}\rangle)\langle\texttt{unit}\rangle
\end{verbatim}

\textbf{Example:}

\begin{verbatim}
\margins/1 a4 (2.5,2.5,2,2)cm
\end{verbatim}

\textbf{Parameters are:}

- \texttt{\langle pg\rangle} ... 1 or 2 specifies one-page or two-pages design.
- \texttt{\langle fmt\rangle} ... paper format (a4, a4l, a5, letter, etc. or user defined).
- \texttt{\langle left\rangle, \langle right\rangle, \langle top\rangle, \langle bot\rangle} ... gives the amount of left, right, top and bottom margins.
- \texttt{\langle unit\rangle} ... unit used for values \texttt{\langle left\rangle}, \texttt{\langle right\rangle}, \texttt{\langle top\rangle}, \texttt{\langle bot\rangle}.

Each of the parameters \texttt{\langle left\rangle, \langle right\rangle, \langle top\rangle, \langle bot\rangle} can be empty. If both \texttt{\langle left\rangle} and \texttt{\langle right\rangle} are nonempty then \texttt{\hsize} is set. Else \texttt{\hsize} is unchanged. If both \texttt{\langle left\rangle} and \texttt{\langle right\rangle} are empty then typesetting area is centered in the paper format. The analogical rule works when \texttt{\langle top\rangle} or \texttt{\langle bot\rangle} parameter is empty (\texttt{\vsize} instead \texttt{\hsize} is used). Examples:

\begin{verbatim}
\margins/1 a4 (,,,)mm % \hsize, \vsize untouched, typesetting area centered
\margins/1 a4 (,2,,)cm % right margin set to 2cm % \hsize, \vsize untouched, vertically centered
\end{verbatim}

If \texttt{\langle pg\rangle}=1 then all pages have the same margins. If \texttt{\langle pg\rangle}=2 then the declared margins are true for odd pages. The margins at the even pages are mirrored in such case, it means that \texttt{\langle left\rangle} is replaced by \texttt{\langle right\rangle} and vice versa.

The \texttt{\langle fmt\rangle} can be in the form (\langle\texttt{width}\rangle,\langle\texttt{height}\rangle)\langle\texttt{unit}\rangle where \texttt{\langle unit\rangle} is optional. If it is missing then \texttt{\langle unit\rangle} after margins specification is used. For example:

\begin{verbatim}
\margins/1 (100,200) (7,7,7,7)mm
\end{verbatim}

declares the paper 100×200 mm with all four margins 7 mm. The spaces before and after \texttt{\langle fmt\rangle} parameter are necessary.

The command \texttt{\magscale[\langle factor\rangle]} scales the whole typesetting area. The fixed point of such scaling is the upper left corner of the paper sheet. Typesetting (breakpoints etc.) is unchanged. All units are relative after such scaling. Only paper formats dimensions stays unscaled. Example:

\begin{verbatim}
\margins/2 a5 (22,17,19,21)mm
\magscale[1414] \margins/1 a4 (,,,)mm
\end{verbatim}

The first line sets the \texttt{\hsize} and \texttt{\vsize} and margins for final printing at a5 format. The setting on the second line centers the scaled typesetting area to the true a4 paper while breaking points for paragraphs and pages are unchanged. It may be usable for review printing. After review is done, the second line can be commented out.

\subsection*{2.2 Concept of default page}

\LaTeX{} uses for page design very similar “output routine” like plain \TeX{}. There is \texttt{\headline} followed by “page body” followed by \texttt{\footline}. The \texttt{\headline} is empty by default and it can be used for running headers repeated on each page. The \texttt{\footline} prints page number in the center by default.

You can set the \texttt{\footline} to empty using \texttt{\nopagenumbers} macro.

The margins declared by \texttt{\margins} macro is concerned to the page body, i.e. the \texttt{\headline} and \texttt{\footline} are placed to the top and bottom margins.
The distance between \headline and the top of page body is given by \headlinedist register. The distance between bottom of page body and \footline is given by \footlinedist. The default values are:

\headline = {}
\footline = \{\_hss\_rmfixed \_folio \_hss\} \% \folio expands to page number
\headlinedist = 14pt \% from baseline of \headline to top of page body
\footlinedist = 24pt \% from last line in pagebody to baseline of footline

The page body should be divided to top insertions (floating tables and figures), real text and footnotes. Typically only real text is here.

The \pgbackground tokens list is empty by default but it can be used for creating background of each page (colors, picture, watermark for example). The macro \draft uses this register and puts big text DRAFT as watermark to each page. You can try it.

More about the page layout is documented in files parameters.opm and output.opm.

2.3 Footnotes and marginal notes

The plain \TeX’s macro \footnote can be used as usual. But a new macro \fnote{⟨text⟩} is defined. The footnote mark is added automatically and it is numbered on each chapter from one\(^3\). The \(⟨text⟩\) is scaled by \typoscale[800]. User can redefine footnote mark or scaling, as shown in the file fnotes.opm.

The \fnote macro is fully applicable only in “normal outer” paragraph. It doesn’t work inside boxes (tables for example). If you are solving such case you can use \fnote\{number\} inside the box (only the footnote mark is generated). When the box is finished you can use \fnote\{⟨text⟩\}. This macro puts the \(⟨text⟩\) to the footnote. The \(number\) after \fnote have to be ”1” if only one such command is in the box. Second \fnote\{number\} inside the same box have to have the parameter 2 etc. The same number of \fnote\{number\} have to be written after the box as the number of \fnote\{number\} inserted inside the box.

The marginal note can be printed by the \mnote\{⟨text⟩\} macro. The \(⟨text⟩\) is placed to the right margin on the odd pages and it is placed to the left margin on the even pages. This is done after second \TeX run because the relevant information is stored in an external file. If you need to place the notes only to the fixed margin write \fixmnotes\{right\} or \fixmnotes\{left\}.

The \(⟨text⟩\) is formatted as a little paragraph with the maximal width \mnotesize ragged left on the left margins or ragged right on the right margins. The first line of this little paragraph is at the same height as the invisible mark created by \mnote in the current paragraph. The exceptions are possible by \mnoteskip register. You can implement such exceptions to each \mnote manually in final printing in order to margin notes do not overlap. The positive value of \mnoteskip shifts the note up and negative value shifts it down. For example \mnoteskip=2\baselineskip \mnote\{⟨text⟩\} shifts this (and only this) note two lines up.

3 Fonts

3.1 Font families

You can select the font family by \fontfam[⟨Family-name⟩]. The argument ⟨Family-name⟩ is case insensitive and spaces are ignored. So, \fontfam[LM Fonts] is equal to \fontfam[LMfonts] and it is equal to \fontfam[Lmfonts]. Several aliases are prepared, thus \fontfam[Latin Modern] can be used for loading Latin Modern family too.

If you write \fontfam[?] then all font families registered in \Op\TeX are listed on the terminal and in the log file.

If you write \fontfam[catalog] then a catalog of all fonts registered in \Op\TeX and available in your \TeX system is printed. And the instructions how to register your own font family is appended in such catalog.

If the family is loaded then font modifiers applicable in such font family are listed on the terminal: (\caps, \cond for example). And there are four basic variant selectors (\rm, \bf, \it, \bi). The

\(^3\) You can declare \fnotenumglobal if you want footnotes numbered in whole document from one or \fnotenumpages if you want footnotes numbered at each page from one. Default setting is \fnotenumchapters.
font modifiers (\caps, \cond for example) can be used immediately before a basic variant selector and they can be (independently) combined: \caps\it or \cond\caps\bf. The modifiers keeps their internal setting until group ends or until another modifier which negates the previous feature is used. So \caps \rm text \it text uses normal and italics in Caps and SmallCaps.

There is one special variant selector \currvar which does not change the selected variant but reloads the font in respect of the (maybe newly specified) font modifiers(s). The context between variants \rm–\it and \bf–\bi is kept by the \em macro (emphasize text). It switches from current \rm to \it, from current \it to \rm, from current \bf to \bi and from current \bi to \bf. The needed italics correction \v is inserted automatically. Example:

This is {\em important} text. % = This is {\it important}/\v text.
\it This is {\em important} text. % = This is/\v {\rm important} text.
\bf This is {\em important} text. % = This is {\bi important}/\v text.
\bi This is {\em important} text. % = This is/\v {\bf important} text.

More about the OpTeX font selection system is written the file fonts-select.opm. You can mix more font families in your document, you can declare your variant selectors or modifiers etc.

3.2 Font sizes

The command \typosize\[\langle\text{fontsize}\rangle/\langle\text{baselineskip}\rangle\] sets the font size of text and math fonts and baselineskip. If one of these two parameters is empty, the corresponding feature stays unchanged. Don’t write the unit of these parameters. The unit is internally set to \ptunit which is 1pt by default. You can change the unit by the command \ptunit=\langle\text{something-else}\rangle, for instance \ptunit=1mm enlarges all font sizes declared by \typosize. Examples:

\typosize[10/12] % default of plainTeX
\typosize[11/12.5] % font 11pt, baseline 12.5pt
\typosize[8/] % font 8pt, baseline unchanged

The commands for font size setting described in this section have local validity. If you put them into a group, the settings are lost when the group is finished. If you set something relevant with paragraph shape (baselineskip given by \typosize for example) then you must first finalize the paragraph and second to close the group: \{\typosize[12/14] \ldots\langle\text{text of paragraph}\rangle\ldots \par\}.

The command \typoscale\[\langle\text{font-factor}\rangle/\langle\text{baselineskip-factor}\rangle\] sets the text and math fonts size and baselineskip as a multiple of the current fonts size and baselineskip. The factor is written in “scaled”-like way, it means that 1000 means factor one. The empty parameter is equal to the parameter 1000, i.e. the value stays unchanged. Examples:

\typoscale[800/800] % fonts and baselineskip re-size to 80 %
\typoscale[magstep2/] % fonts bigger 1,44times

First usage of \typosize or \typoscale macro in your document sets so called main values, i.e. main font size and main baselineskip. They are internally saved in registers \mainfosize and \mainbaselineskip.

The \typoscale command does scaling in respect to current values by default. If you want to do it in respect to main values, type \scalemain immediately before \typoscale command.

\typosize[12/14.4] % first usage in document, sets main values internally
\typosize[15/18] % bigger font
\scalemain \typoscale[800/800] % reduces from main values, no from current.

The size of the current font can be changed by the command \thefontsize\[\langle\text{font-size}\rangle\] or can be rescaled by \thefontscale\[\langle\text{factor}\rangle\]. These macros don’t change math fonts sizes nor baselineskip.

There is \setfontsize\[\langle\text{size-spec}\rangle\] command which behaves like font modifiers and sets given font size o fonts loaded by next variant selectors. For example \setfontsize{at15pt}\currvar.

More information about resizing of fonts is documented in fonts-resize.opm file.

3.3 Typesetting math

OpTeX preloads a collection of 7bit Computer Modern and AMS fonts. You can use them in any size and in the \boldmath variant. Most declared font families are configured with recommended
Unicode math font. This font is automatically loaded unless you specify \noloadmath before first \fontfam command. See log file for more information about loading text font family and Unicode math fonts. If you prefer another Unicode math font, specify it by \loadmath\{⟨font-file⟩\} or \loadmath\{font-name\} before first \loadfam command.

Hundreds math symbols and operators like in AMSTeX are accessible. For example \alpha α, \geq ≥, \sum ∑, \sphericalangle ∢, \bumpeq ≍. See AMSTeX manual for complete list of symbols.

The following math alphabets are available:

\mit % mathematical variables abc–xyz, ABC–XYZ
\it % text italics abc–xyz, ABC–XYZ
\rm % text roman abc–xyz, ABC–XYZ
\cal % normal calligraphics ABC–XYZ
\script % script A BC – X Y Z
\frak % fracture abc–xyz, ABC–XYZ
\bbchar % double stroked letters ABC–XYZ
\bf % sans serif bold abc–xyz, ABC–XYZ
\bi % sans serif bold slanted abc–xyz, ABC–XYZ

The last two selectors \bf and \bi select the sans serif fonts regardless current text font family.

The math fonts can be scaled by \typosize and \typoscale macros. Two math fonts collections are prepared: \normalmath for normal weight and \boldmath for bold. The first one is set by default.

You can use \mathbox\{⟨text⟩\} inside math mode. It behaves as \{\hbox{⟨text⟩}\} (i.e. the ⟨text⟩ is printed in horizontal non-math mode) but the size of the ⟨text⟩ is adapted to the context of math size (text or script or scriptscript).

4 Typical elements of document
4.1 Chapters and sections

The document can be divided into chapters, sections and subsections and titled by \tit command. The parameters are separated by the end of current line (no braces are used):

\tit Document title ⟨end of line⟩
\chap Chapter title ⟨end of line⟩
\sec Section title ⟨end of line⟩
\secc Subsection title ⟨end of line⟩

The chapters are numbered by one number, sections by two numbers (chapter.section) and subsections by three numbers. If there are no chapters then section have only one number and subsection two.

The implicit design of the titles of chapter etc. are implemented in the macros \printchap, \printsec and \printsecc. User can simply change these macros if he/she needs another behavior.

The first paragraph after the title of chapter, section and subsection is not indented but you can type \let\firstnoindent=\relax if you need all paragraphs indented.

If a title is so long then it breaks to more lines. It is better to hint the breakpoints because TeX does not interpret the meaning of the title. User can put the \nl (it means newline) macro to the breakpoints.

The chapter, section or subsection isn’t numbered if the \nonum precedes. And the chapter, section or subsection isn’t delivered to the table of contents if \notoc precedes.

4.2 Another numbered objects

Apart from chapters, sections and subsections, there are another automatically numbered objects: equations and captions for tables and figures.

If user write the \eqmark as the last element of the display mode then this equation is numbered. The format is one number in brackets. This number is reset in each section.

If the \eqalignno is used, then user can put \eqmark to the last column before \cr. For example:

\eqalignno{a^2+b^2 &= c^2 \cr c &= \sqrt{a^2+b^2} & \eqmark \cr}
The next numbered object is caption which is tagged by \caption/t for tables and \caption/f for figures. The caption text follows. The \cskip can be used between \caption text and the real object (table or figure). It is irrelevant, if caption or object is first. The \cskip creates appropriate vertical space between them. Example:

\caption/t The dependency of the computer-dependency on the age.
\cskip

\hfil\table{rl}{
  age & value \
  0--1 & unmeasured \\cr
  1--6 & observable \\cr
  6--12 & significant \\cr
  12--20 & extremal \\cr
  20--40 & normal \\cr
  40--60 & various \\cr
  60--$\infty$ & moderate}

This example produces:

\caption/t The dependency of the computer-dependency on the age.
\hspace{1cm}

<table>
<thead>
<tr>
<th>age</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–1</td>
<td>unmeasured</td>
</tr>
<tr>
<td>1–6</td>
<td>observable</td>
</tr>
<tr>
<td>6–12</td>
<td>significant</td>
</tr>
<tr>
<td>12–20</td>
<td>extremal</td>
</tr>
<tr>
<td>20–40</td>
<td>normal</td>
</tr>
<tr>
<td>40–60</td>
<td>various</td>
</tr>
<tr>
<td>60–$\infty$</td>
<td>moderate</td>
</tr>
</tbody>
</table>

You can see that the word “Table” followed by a number is added by the macro \caption/t. The caption text is centered. If it occupies more lines then the last line is centered.

The macro \caption/f behaves like \caption/t but it is intended for figure captions. The word (Table, Figure) depends on the actual selected language (see section 7.1 about languages).

If you wish to make the table or figure as floating object, you need to use plain \TeX macros \midinsert or \topinsert terminated by \endinsert.

4.3 References

Each automatically numbered object documented in sections 4.1 and 4.2 can be referenced if optional parameter [⟨label⟩] is appended to \chap, \sec, \secc, \caption/t, \caption/f or \eqmark. The alternative syntax is to use \label[⟨label⟩] before mentioned commands (not necessarily directly before). The reference is realized by \ref[⟨label⟩] or \pgref[⟨label⟩]. Example:

\sec[beatle] About Beatles

\hfil\table{rl}{...} % the table
\cskip
\caption/t [comp-depend] The dependency of the computer-dependency on the age.

\label[pythagoras]
$$ a^2 + b^2 = c^2 \ \eqmark $$

Now we can point to the section-\ref[beatle] on the page-\pgref[beatle] or write about the equation-\ref[pythagoras]. Finally there is an interesting Table-\ref[comp-depend].

If there are forward referenced objects then user have to run \TeX twice. During each pass, the working *.ref file (with references data) is created and this file is used (if it exists) at the beginning of the document.
You can create a reference to whatever else by commands \label{⟨label⟩} \wlabel{⟨text⟩}. The connection between ⟨label⟩ and ⟨text⟩ is established. The \ref{⟨label⟩} will print ⟨text⟩.

### 4.4 Hyperlinks, outlines

If the command \hyperlinks{(color-in)}{(color-out)} is used at the beginning of the file, then the following objects are hyperlinked when PDF output is used:

- numbers generated by \ref or \pgref,
- numbers of chapters, sections and subsections in the table of contents,
- numbers or marks generated by \cite command (bibliography references),
- texts printed by \url command.

The last object is an external link and it is colored by ⟨color-out⟩. Others links are internal and they are colored by ⟨color-in⟩. Example:

\hyperlinks \Blue \Green % internal links blue, URLs green.

You can use another marking of active links: by frames which are visible in the PDF viewer but invisible when the document is printed. The way to do it is to define the macros \pgborder, \tocborder, \citeborder, \refborder and \urlborder as the triple of RGB components of the used color. Example:

\def\pgborder {1 0 0} % links to pages: red frame
\def\tocborder {0 1 0} % links in table of contents: green frame
\def\citeborder {0 0 1} % links to references: blue frame

By default these macros are not defined. It means that no frames are created.

There are “low level” commands to create the links. You can specify the destination of the internal link by \dest{⟨type⟩}{⟨label⟩}. The active text linked to the \dest can be created by \ilink{⟨type⟩}{⟨label⟩}{⟨text⟩}. The ⟨type⟩ parameter is one of the toc, pg, cite, ref or another special for your purpose. These commands create internal links only when \hyperlinks is declared.

The \url macro prints its parameter in \tt font and creates a potential breakpoints in it (after slash or dot, for example). If \hyperlinks declaration is used then the parameter of \url is treated as an external URL link. An example: \url{http://www.olsak.net} creates http://www.olsak.net.

The characters %, \, #, { and } have to be protected by backslash in the \url argument, the other special characters ~, ^, & can be written as single character. You can insert the \| command in the \url argument as a potential breakpoint.

If the linked text have to be different than the URL, you can use \ulink{⟨url⟩}{⟨text⟩} macro. For example: \ulink[http://petr.olsak.net/optex]{OpTEX page} creates OpTEX page.

The PDF format provides outlines which are notes placed in the special frame of the PDF viewer. These notes can be managed as structured and hyperlinked table of contents of the document. The command \outlines{⟨level⟩} creates such outlines from data used for table of contents in the document. The ⟨level⟩ parameter gives the level of opened sub-outlines in the default view. The deeper levels can be open by mouse click on the triangle symbol after that.

If you are using a special macro in section titles then \outlines macro may crash. You must declare variant of the macro for outlines case which is expandable using \regmacro. See section 5.1 for more information about \regmacro.

The command \insertoutline{⟨text⟩} inserts next entry into PDF outlines at the main level 0. This entry can be placed before table of contents (created by \outlines) or after it.

### 4.5 Lists

The list of items is surrounded by \begitems and \enditems commands. The asterisk (•) is active within this environment and it starts one item. The item style can be chosen by \style parameter written after \begitems:

---

4 More exactly, there is the same rules as for \code command, see section 4.7.
For example:

\begitems
  * First idea
  * Second idea in subitems:
    \begitems \style i
      * First sub-idea
      * Second sub-idea
      * Last sub-idea
    \enditems
  * Finito
\enditems

produces:

- First idea
- Second idea in subitems:
  (i) First sub-idea
  (ii) Second sub-idea
  (iii) Last sub-idea
- Finito

Another style can be defined by the command \sdef \item {⟨style⟩}{⟨text⟩}. Default item can be redefined by \def \normalitem {⟨text⟩}. The list environments can be nested. Each new level of item is indented by next multiple of \iindent which is set to \parindent by default. The vertical space at begin and end of the environment is inserted by the macro \iiskip.

4.6 Tables

The macro \table {⟨declaration⟩}{⟨data⟩} provides similar ⟨declaration⟩ as in \LaTeX: you can use letters l, r, c, each letter declares one column (aligned to left, right, center respectively). These letters can be combined by the | character (vertical line). Example

\table{||lc|r||}{
  \crl
  Month & commodity & price  \\
  January & notebook & \$ 700 \cr
 February & skateboard & \$ 100 \cr
  July     & yacht      & k\$ 170 \crl}

generates the following result:

<table>
<thead>
<tr>
<th>Month</th>
<th>commodity</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>January</td>
<td>notebook</td>
<td>$ 700</td>
</tr>
<tr>
<td>February</td>
<td>skateboard</td>
<td>$ 100</td>
</tr>
<tr>
<td>July</td>
<td>yacht</td>
<td>k$ 170</td>
</tr>
</tbody>
</table>
Apart from l, r, c declarators, you can use the \texttt{p\{\textit{size}\}} declarator which declares the column of given width. More precisely, a long text in the table cell is printed as a paragraph with given width. To avoid problems with narrow left-right aligned paragraphs you can write \texttt{p\{\textit{size}\}\texttt{\textbackslash{raggedright}}} then the paragraph will be only left aligned.

You can use \texttt{\langle\langle\textit{text}\rangle\rangle} in the \texttt{\langle\textit{declaration}\rangle}. Then this text is applied in each line of the table. For example \texttt{r\{\kern10pt\}} adds more 10 pt space between \texttt{r} and \texttt{l} rows.

An arbitrary part of the \texttt{\langle\textit{declaration}\rangle} can be repeated by a \texttt{\langle\textit{number}\rangle} prefixed. For example \texttt{3c} means \texttt{ccc} or \texttt{c|c|c|c}. Note that spaces in the \texttt{\langle\textit{declaration}\rangle} are ignored and you can use them in order to more legibility.

\texttt{\textbackslash{cr}} used in the \texttt{\langle\textit{data}\rangle} part of the table (the end row separator) is generally known. It marks end row of the table. Moreover \texttt{Op\TeX} defines following similar commands:

- \texttt{\textbackslash{crl}} ... the end of the row with a horizontal line after it.
- \texttt{\textbackslash{crl1}} ... like \texttt{\textbackslash{crl}} but the horizontal line doesn’t intersect the vertical double lines.
- \texttt{\textbackslash{crlp\{\langle\textit{list}\rangle\}}} ... like \texttt{\textbackslash{crl1}} but the lines are drawn only in the columns mentioned in comma separated \texttt{\langle\textit{list}\rangle} of their numbers. The \texttt{\langle\textit{list}\rangle} can include \langle\textit{from}\rangle–\langle\textit{to}\rangle declarators, for example \texttt{\textbackslash{crlp\{1-3,5\}}} is equal to \texttt{\textbackslash{crlp\{1,2,3,5\}}}.

The \texttt{\textbackslash{ta\textbackslash{skip}}\{\textit{dimen}\}} command works like the \texttt{\textbackslash{noalign}}\texttt{\textbackslash{vskip}}\texttt{\{\textit{dimen}\}} after \texttt{\textbackslash{cr\*}} commands but it doesn’t interrupt the vertical lines.

You can use following parameters for the \texttt{\textbackslash{table}} macro. Default values are listed too.

- \texttt{everytable=} % code used after settings in \texttt{\vbox} before table processing
- \texttt{thistable=} % code used when \texttt{\vbox} starts, is is removed after using it
- \texttt{tabiteml=\{\textit{enspace}\}} % left material in each column
- \texttt{tabitemr=\{\textit{enspace}\}} % right material in each column
- \texttt{tabstrut=\{\textit{strut}\}} % strut which declares lines distance in the table
- \texttt{tablinespace=2pt} % additional vertical space before/after horizontal lines
- \texttt{vvkern=1pt} % space between double vertical line
- \texttt{hhkern=1pt} % space between double horizontal line

If you do \texttt{\textbackslash{tabiteml}={$\textit{enspace}$}\textbackslash{tabitemr}={$\textit{enspace}$}} then the \texttt{\textbackslash{table}} acts like \LaTeX’s \texttt{array} environment.

If there is an item which spans to more than one column in the table then \texttt{\textbackslash{multispan\{\langle\textit{number}\rangle\}}} macro from plain \TeX\ can help you or, you can use \texttt{\textbackslash{mspan\{\langle\textit{number}\rangle\}}\{\langle\textit{declaration}\rangle\}\{\langle\textit{text}\rangle\}} which spans \langle\textit{number}\rangle columns and formats the \langle\textit{text}\rangle by the \langle\textit{declaration}\rangle. The \langle\textit{declaration}\rangle must include a declaration of right one column with the same syntax as common \texttt{\textbackslash{table}} \langle\textit{declaration}\rangle. If your table includes vertical rules and you want to create continuous vertical rules by \texttt{\textbackslash{mspan}}, then use rules only after \texttt{c}, \texttt{l} or \texttt{r} letter in \texttt{\textbackslash{mspan}} \langle\textit{declaration}\rangle. The exception is only in the case when \texttt{\textbackslash{mspan}} includes first column and the table have rules on the left side. The example of \texttt{\textbackslash{mspan}} usage is below.

The \texttt{\textbackslash{frame\{\langle\textit{text}\rangle\}}} makes a frame around \langle\textit{text}\rangle. You can put the whole \texttt{\textbackslash{table}} into \texttt{\textbackslash{frame}} if you need double-ruled border of the table. Example:

```
\textbackslash{frame}\{\textbackslash{table}\{c|l|l|l\}\}\{\textbackslash{crl}\}
\textbackslash{mspan3\{c\}\\{bf Title\}\}\{\textbackslash{crl1}\}\noalign{\kern\textbackslash{hhkern}}\noalign{\kern\textbackslash{khkern}}\noalign{\kern\textbackslash{vskern}}\textbackslash{crl1}
first & second & third & fourth & fifth & sixth & seventh & eighth & ninth & tenth
\textbackslash{crl1}\}
```

creates the following result:

<table>
<thead>
<tr>
<th></th>
<th>first</th>
<th>second</th>
<th>third</th>
</tr>
</thead>
<tbody>
<tr>
<td>seven</td>
<td>eight</td>
<td>nine</td>
<td></td>
</tr>
</tbody>
</table>

The \texttt{c}, \texttt{l}, \texttt{r} and \texttt{p} are default \langle\textit{declaration}\rangle letters but you can define more such letters by \texttt{\textbackslash{def\{\textbackslash{tabdeclare}}\{\langle\textit{letter}\rangle\}\{\langle\textit{left}\rangle#\langle\textit{right}\rangle\}}. More about it is in technical documentation in the file table.opm.
The rule width of tables (and implicit width of all \vrule and \hrule) can be set by the command \rulewidth={\dimen}. The default value given by \TeX{} is 0.4 pt.

Many tips about tables can be seen on \url{http://petr.olsak.net/opmac-tricks-e.html}.

4.7 Verbatim

The display verbatim text have to be surrounded by the \begtt and \endtt couple. The in-line verbatim have to be tagged (before and after) by a character which is declared by \activettchar{\char}. For example \activettchar{`} declares the character ` for in-line verbatim markup. And you can use \texttt{`relax} for verbatim \texttt{relax} (for example). Another alternative of printing in-line verbatim text is \code{⟨text⟩} (see below).

If the numerical register \ttline is set to the non-negative value then display verbatim will number the lines. The first line has the number \ttline+1 and when the verbatim ends then the \ttline value is equal to the number of last line printed. Next \begtt...\endtt environment will follow the line numbering. \OpTeX{} sets \ttline=-1 by default.

The indentation of each line in display verbatim is controlled by \ttindent register. This register is set to the \parindent by default. User can change values of the \parindent and \ttindent independently.

The \begtt command starts internal group in which the catcodes are changed. Then the \everytt string reister is run. It is empty by default and user can control fine behavior by it. For example the catcodes can be reset here. If you need to define active character in the \everytt, use \adef as in the following example:

\everytt={\adef!{?}\adef?{!}}
\begtt
Each occurrence of the exclamation mark will be changed to the question mark and vice versa. Really? You can try it!
\endtt

The \adef command sets its parameter as active after the body of \everytt is read. So you can’t worry about active categories.

The \everytt is applied to all \begtt...\endtt environments (if it is not declared in a group). There are tips for such global \everytt definitions here:

\everytt={\typosize[9/11]} % setting font size for verbatim
\everytt={\ttline=0} % each listing will be numbered from one
\everytt={\adef {}:\char9251 {}} % visualization of spaces (Unicode fonts)

If you want to apply a \everytt material only for one \begtt...\endtt environment then don’t set any \everytt but put desired material at the same line where \begtt is. For example:

\begtt
\adef!{?}\adef?{!}
Each occurrence of ? will be changed to ! and vice versa.
\endtt

The in-line verbatim surrounded by an \activettchar doesn’t work in parameter of macros and macro definitions, especially in titles declared by \chap, \sec etc. You ca use more robust command \code{⟨text⟩} in such situations, but you must escape following characters in the ⟨text⟩: \, #, %, braces (if the braces are unmatched in the ⟨text⟩), and space or \(^\text{^}^\text{M}\) (if there are more than one subsequent spaces or \(^\text{^}^\text{M}\) in the ⟨text⟩). Examples:

\code{\text, \%\#} ... prints \text, \%
\code{@{\ldots}*\$ $} ... prints @{\ldots}*\$ without escaping, but you can escape these characters too, if you want.
\code{a \ b} ... two spaces between a b, the second one must be escaped
\code{xy\{z} ... xy{z ... unbalanced brace must be escaped
\code{\~\~\~}\code{M} ... prints \~\~\~\M, the second \~ must be escaped

There is an alternative to \everytt named \everyintt which is used for in-line verbatim surrounded by an \activettchar or processed by the \code command.

You can print verbatim listing from external files by \verbinput command. Examples:
The \texttt{line} influences the line numbering by the same way as in \texttt{...} environment. If \texttt{line}=-1 then real line numbers are printed (this is default). If \texttt{line}<-1 then no line numbers are printed.

The \verbinput can be controlled by \everytt, \ttindent just like in \begtt...\endtt.

5 Autoencoded lists

5.1 Table of contents

The \maketoc command prints the table of contents of all \chap, \sec and \secc used in the document. These data are read from external *.ref file, so you have to run \TeX more than once (typically three times if the table of contents is at the beginning of the document).

The name of the section with table of contents is not printed. The direct usage of \chap or \sec isn’t recommended here because the table of contents is typically not referenced to itself. You can print the unnumbered and unreferenced title of the section by the code:

\nonum\notoc\sec Table of Contents

If you are using a special macro in section titles or chapter titles and you need different behavior of such macro in other cases then use \regmacro{\langle case-toc \rangle}{\langle case-mark \rangle}{\langle case-outline \rangle}. The parameters are applied locally in given cases. The \regmacro can be used repeatedly: the parameters are accumulated (for more macros). If a parameter is empty then original definition is used in given case. For example:

% default value of \mylogo macro used in text and in the titles:
\def\mylogo{\leavevmode\hbox{\Red{\it My}\Black{\setfontsize{mag1.5}{\rm Lo}Go}}}
% another variants:
\regmacro \{\def\mylogo{\hbox{\Red My\Black LoGo}}} % used in TOC
\{\def\mylogo{\hbox{\it My\Black LoGo}}} % used in running heads
\{\def\mylogo{MyLoGo}} % used in outlines

5.2 Making the index

The index can be included into document by \makeindex macro. No external program is needed, the alphabetical sorting are done inside \TeX at macro level.

The \ii command (insert to index) declares the word separated by the space as the index item. This declaration is represented as invisible atom on the page connected to the next visible word. The page number of the page where this atom occurs is listed in the index entry. So you can type:

The \ii resistor resistor is a passive electrical component ...

You cannot double the word if you use the \iid instead \ii:

The \iid resistor is a passive electrical component ...

or:

Now we'll deal with the \iid resistor .

Note that the dot or comma have to be separated by space when \iid is used. This space (before dot or comma) is removed by the macro in the current text.

The multiple-words entries are commonly organized in the index by the format (for example):

linear dependency 11, 40–50
— independency 12, 42–53
To do this you have to declare the parts of the words by the / separator. Example:

\bf Definition.
\ii linear\space, vector\space
\em Linear space (or \em vector space) is a nonempty set of...

The number of the parts of one index entry is unlimited. Note, that you can spare your typing by the comma in the \ii parameter. The previous example is equivalent to \ii linear\space \ii vector\space.

Maybe you need to propagate to the index the similar entry to the linear\space in the form space\linear. You can do this by the shorthand ,@ at the end of the \ii parameter. Example:

\ii linear\space, vector\space,@
is equivalent to:
\ii linear\space, vector\space \ii space\linear, space\vector

If you really need to insert the space into the index entry, write ~.

If the \ii or \iid command is preceded by \iitype ⟨letter⟩, then such reference (or more references generated by one \ii) has specified type. They should have different format in the index. OpTEX implements \iitype b and \iitype i. This prints the page number in bold or in italics in the index. Default is empty index type, which prints page numbers in normal font. See \TeX{}book index as an example.

The \texttt{\textbackslash makeindex} creates the list of alphabetically sorted index entries without the title of the section and without creating more columns. OpTEX provides another macros for more columns:

\texttt{\textbackslash begmulti ⟨number of columns⟩
\langle text⟩
\textbackslash endmulti}

The columns will be balanced. The Index can be printed by the following code:

\texttt{\sec Index
\begmulti 3 \makeindex \endmulti}

Only “pure words” can be propagated to the index by the \ii command. It means that there cannot be any macro, \TeX{} primitive, math selector etc. But there is another possibility to create such complex index entry. Use “pure equivalent” in the \ii parameter and map this equivalent to the real word which is printed in the index by \iis command. Example:

The \ii chiquadrat $\chi$-quadrat method is
...
If the \ii relax \texttt{\\relax} command is used then \TeX/ is relaxing.
...
\iis chiquadrat {$\chi$-quadrat}
\iis relax \{code{\relax}}
...

The \iis \texttt{(equivalent) \{⟨text⟩\}} creates one entry in the dictionary of the exceptions. The sorting is done by the \texttt{(equivalent)} but the \texttt{(text)} is printed in the index entry list.

The sorting rules when \texttt{\makeindex} runs depends on the current language. See section 7.1 about languages selection.

5.3 Bib\TeX{}ing

The command \texttt{\cite{⟨label⟩}} or its variants of the type \texttt{\cite{⟨label-1⟩, ⟨label-2⟩,..., ⟨label-n⟩}} create the citations in the form [42] or [15, 19, 26]. If \texttt{\shortcitations} is declared at the beginning of the document then continuous sequences of numbers are re-printed like this: [3–5, 7, 9–11]. If \texttt{\sortcitations} is declared then numbers generated by one \cite command are sorted upward.

If \texttt{\nonumcitations} is used then the marks instead numbers are generated depending on the used bib-style. For example the citations look like [Now08] or [Nowak, 2008].
The \rcite\[\langle labels \rangle\] creates the same list as \cite\[\langle labels \rangle\] but without the outer brackets. Example: \[\rcite[\{tbn\}, pg.-13]\] creates \[4, pg.13\].

The \ecite\[\langle label \rangle\{\langle text \rangle\}\] prints the \langle text \rangle only, but the entry labeled \langle label \rangle is decided as to be cited. If \hyperlinks is used then \langle text \rangle is linked to the references list.

You can define alternative formatting of \cite command. Example:

\begin{verbatim}
\def\cite[#1]{\{\rcite[#1]\}} % \cite\[\langle label \rangle\] creates (27)
\def\cite[#1]{^\{\rcite[#1]\}\}} % \cite\[\langle label \rangle\] creates^27
\end{verbatim}

The numbers printed by \cite correspond to the same numbers generated in the list of references.

There are two possibilities to generate this references list:

- Manually using \bib\[\langle label \rangle\] commands.
- By \usebib/\langle type ⟩\(\langle style \rangle\)\(\langle bbl-base \rangle\) command which reads *.bib databases directly.

Note that another two possibilities documented in OPmac (using external BibT EX program) isn’t supported because BibT EX is old program which does not supports Unicode. And Biber seems to be not compliant with Plain T EX.

References created manually using \bib\[\langle label \rangle\] command.

\bib\[\{tbn\}\] P. Olšák. \it\TeX{}book naruby. 468~s. Brno: Konvoj, 1997.
\bib\[\{tst\}\] P. Olšák. \it Typografický systém \TeX{}\}.

If you are using \nonumcitations then you need to declare the \langle marks \rangle used by \cite command. To do it you must use long form of the \bib command in the format \bib\[\langle label \rangle\] = \{\langle mark \rangle\}. The spaces around equal sign are mandatory. Example:

\bib\[\{tbn\}\] = \{Olšák, 2001\}

Direct reading of .bib files is possible by \usebib macro. This macro reads macro package opmac-bib.tex (on demand) which uses the external package librarian.tex by Paul Isambert. The usage is:

\begin{verbatim}
\usebib/c ((style) \langle bib-base \rangle) % sorted by \cite-order (c=cite),
\usebib/s ((style) \langle bib-base \rangle) % sorted by style (s=style).
% example:
\usebib/s (simple) op-example
\end{verbatim}

The \langle bib-base \rangle is one or more *.bib database source files (separated by spaces and without extension) and the \langle style \rangle is the part of the filename bib-⟨style⟩.opm where the formatting of the references list is defined. Possible styles are simple or iso690. The behavior of opmac-bib.tex and opmac-bib-iso690.tex is full documented in these files (after \endinput command).

The command \usebib select from database files only such bib-records which were used in \cite or \nocite commands in your document. It means, not all records are printed. The \nocite behaves as \cite but prints nothing. It only tells that mentioned bib-record should be printed in the reference list. If \notcite\[*\] is used then all records from \langle bib-base \rangle are printed.

Formatting of the references list is controlled by the \printb macro. It is called at the begin of each entry. The default \printb macro prints the number of the entry in square brackets. If the \nonumcitations is set then no numbers are printed, only all lines (but no first one) are indented. The \printb macro can use the following values: \thebibnum (the number of the entry) and \thebibmark (the mark of the entry used when \nonumcitations is set). Examples:

\begin{verbatim}
% The numbers are without square brackets:
\def\printbib{\hangindent=\parindent \indent \llap{\thebibnum. \}}
% Printing of \langle marks \rangle when \nonumcitations is set:
\def\printbib{\hangindent=\parindent \noindent \thebibmark\quad}
\end{verbatim}

Another examples can be found on the OPmac tricks WWW page.
6 Graphics

6.1 Colors

\textup{\LaTeX} provides a small number of color selectors: \texttt{\Blue}, \texttt{\Red}, \texttt{\Brown}, \texttt{\Green}, \texttt{\Yellow}, \texttt{\Cyan}, \texttt{\Magenta}, \texttt{\White}, \texttt{\Grey}, \texttt{\LightGrey} and \texttt{\Black}. User can define more such selectors by setting the CMYK components. For example

\begin{verbatim}
def \Orange {\setcmykcolor{0 0.5 1 0}}
\end{verbatim}

The command \texttt{\morecolors} reads more definitions of color selectors from \LaTeX\ file \texttt{x11nam.def}. There is about 300 color names like \texttt{\DeepPink}, \texttt{\Chocolate} etc. If there are numbered variants of the same name, then you can append letters B, C, etc. to the name in \LaTeX. For example \texttt{\Chocolate} is Chocolat1, \texttt{\ChocolateB} is Cocolate2 etc.

The color selectors work locally in groups by default but with limitations. See the file \texttt{colors.omd} for more information.

Default colors are defined by four CMYK components using \texttt{\setcmykcolor} like in the example above. But you can define a color with three RGB components too by \texttt{\setrgbcolor}, for example \texttt{\def\Orange{\setrgbcolor{1 0.5 0}}}. All colors defined by \texttt{\morecolors} are in RGB.

You can define your color by a linear combination of previously defined colors using \texttt{\colordef}. For example:

\begin{verbatim}
colordef \myCyan {0.3\Green + 0.5\Blue} % 30 \% green, 50 \% blue, rest is white
colordef \DarkBlue {\Blue + 0.4\Black} % Blue mixed with 40 \% of black
ncolordef \myGreen{\Cyan+\Yellow} % exact the same as \Green
ncolordef \MyColor {0.3\Orange+.5\Green+.2\Yellow}
\end{verbatim}

If a convex linear combination (as in the last example above) is used then it emulates color behavior on a painter’s palette.

Only \texttt{\setcmykcolor} is used in default colors in \LaTeX\ and \texttt{\colordef} creates also a colors defined by \texttt{\setcmykcolor}. If you define your own colors by \texttt{\setrgbcolor} or you use \texttt{\morecolors} then a mix of color spaces should be in the PDF output. The \texttt{\onlyrgb} or \texttt{\onlycmyk} commands solves this problem: only specified color space is used in the PDF output and if a color is specified in another color space then it is converted. The \texttt{\onlyrgb} creates colors more bright (usable for computer presentations). On the other hand CMYK makes colors more true when they are printing.

More about colors, about CMYK versus RGB and about the \texttt{\colordef} command is written in the \texttt{colors.omd} file.

The following example defines a macro which creates the colored text on the colored background.

\textbf{Usage:} \texttt{\coloron\{background\}\{foreground\}\{text\}}

The \texttt{\coloron} can be defined as follows:

\begin{verbatim}
def\coloron#1#2#3{%
  \setbox0=\hbox{#3}\leavevmode
  \rlap{#1\strut \vrule width \wd0#2\box0}%
}
\end{verbatim}

\begin{verbatim}
\coloron\Yellow\Brown{The brown text on the yellow backround}
\end{verbatim}

The watermark is grey text on the backround of the page. \LaTeX\ offers an example: the macro \texttt{\draft} which creates grey scaled and rotated text DRAFT on the backround of every page.

6.2 Images

The \texttt{\inspic} \texttt{(filename).\{extension\}\{space\}} or \texttt{\inspic \{filename\}.\{extension\}} inserts the picture stored in the graphics file with the name \texttt{\{filename\}.\{extension\}}. You can set the picture width by \texttt{\picw=\{dimen\}} before \texttt{\inspic} command which declares the width of the picture. The image files can be in the PNG, JPG, JBIG2 or PDF format.

The \texttt{\picwidth} is an equivalent the register to \texttt{\picw}. Moreover there is an \texttt{\picheight} register which denotes the height of the picture. If both registers are set then the picture will be (probably) deformed.

The image files are searched in \texttt{\picdir}. This token string is empty by default, this means that the image files are searched in the current directory.
If you want to create a vector graphics (diagrams, schema, geometry skicing) then you can do it in Wysiwyg graphics editor (Inkscape for example), export the result to PDF and include it by \inspic. If you want to “proramm” such pictures then Tikz package is recommended. It works in plain \TeX.

## 6.3 PDF transformations

All typesetting elements are transformed by linear transformation given by the current transformation matrix. The \pdfsetmatrix \{\langle a \rangle \langle b \rangle \langle c \rangle \langle d \rangle\} command makes the internal multiplication with the current matrix so linear transformations can be composed. One linear transformation given by the \pdfsetmatrix above transforms the vector \{0, 1\} to \{(a, b)\} and \{1, 0\} to \{(c, d)\}. The stack-oriented commands \pdfsave and \pdfrestore gives a possibility of storing and restoring the current transformation matrix and current point. The position of current point have to be the same from \TeX’s point of view as from transformation point of view when \pdfrestore is processed. Due to this fact the \pdfsave\rlap{(transformed text)}\pdfrestore or something similar is recommended.

\TeX provides two special transformation macros:

\begin{align*}
\pdfscale{\langle horizontal-factor \rangle}{\langle vertical-factor \rangle} \\
\pdfrotate{\langle angle-in-degrees \rangle}
\end{align*}

These macros simply calls the properly \pdfsetmatrix primitive command.

It is known that the composition of transformations is not commutative. It means that the order is important. You have to read the transformation matrices from right to left. Example:

\begin{align*}
\text{First:} & \\\\\pdfsave \pdfrotate{30}\pdfscale{-2}{2}\rlap{text1}\pdfrestore & \% text1 is scaled two times and it is reflected about vertical axis  \\
& \% and next it is rotated by 30 degrees left.  \\
\text{second:} & \\\\\pdfsave \pdfscale{-2}{2}\pdfrotate{30}\rlap{text2}\pdfrestore & \% text2 is rotated by 30 degrees left then it is scaled two times  \\
& \% and reflected about vertical axis.  \\
\text{third:} & \\\\\pdfsave \pdfrotate{-15.3}\pdfsetmatrix{2 0 1.5 2}\rlap{text3}\pdfrestore & \% first slanted, then rotated by 15.3 degrees right
\end{align*}

This gives the following result. First: \text{text1} second: \text{text2} third: \text{text3}

## 7 Others

### 7.1 Using more languages

\TeX prepares hyphenation patterns for all languages if such patterns are available in your \TeX system. Only USenglish patterns (original from Plain \TeX) are preloaded. Hyphenation patterns of all another languages are loaded on demand when you first use the \iso-code\lang command in your document. For example \delang for German, \cslang for Czech, \pllang for Polish. The \iso-code\lang is a shortcut of the language (mostly from ISO 639-1). You can list all available languages by \langlist macro. This macro prints now:

en(USenglish) enus(USenglishmax) engb(UKenglish) it(italian) id(idonesian) cs(czech) sk(slovak) de(nGerman) fr(French) pl(Polish) cy(Welsh) da(Danish) es(Spanish) sl(Slovenian) fi(Finnish) hu(Hungarian) tr(Turkish) et(Estonian) eu(Euscan) gb(Irish) nb(Bokmal) nn(Nyorsk) nl(Dutch) pt(PORTUGUESE) ro(Romanian) hr(Croatian) zh(pinyin) is(Icelandic) bs(UpperSorbian) af(Afrikaans) gl(Galician) kmr(Kurmanji) tk(Turkmen) la(Latin) lac clas-Latin) eml(monoGreek) ep(Greek) grc(ancientGreek) ca(Catalan) cop(Coptic) mn(Mongolian) sa(Sanskrit) ru(Russian) uk(Ukrainian) by(Armenian) as(assamese) hi(Hindi) kn(Kannada) lv(Latvian) it(Lithuanian) ml(Malayalam) mr(Marathi) or(Oriya) pa(Panjabi) ta(Tamil) te(Telugu)

For compatibility with e-plain macros, there is the command \uselanguage{\langle language \rangle}. The parameter \langle language \rangle is long form of language name, i.e. \uselanguage{Czech} does the same work as \cslang. The \uselanguage parameter is case insensitive.

For compatibility with \迄今plain there are macros \ehyph, \chyph, \shyph which are equivalent to \enlang, \cslang and \sklang.
You can switch between language patterns by \(\text{\textbackslash iso-code\textbackslash lang}\) commands mentioned above. Default is \texttt{\textbackslash enlang}.

\LaTeX{} generates three words used for captions and titles in technical articles or books: “Chapter”, “Table” and “Figure”. These words need to be known in used language and it depends on the previously used language selectors \texttt{\textbackslash iso-code\textbackslash lang}. \LaTeX{} declares these words only for few languages: Czech, German, Spanish, French, Greek, Italian, Polish, Russian, Slovak and English. If you need to use these words in another languages or you want to auto-generate more words in your macros, then you can declare it by \texttt{\textbackslash def} commands as shown in the file \texttt{languages.omp}.

The \texttt{makeindex} command needs to know the sorting rules used in your language. \LaTeX{} defines only few language rules for sorting: Czech, Slovak and English. How to declare sorting rules for more languages are described in the file \texttt{makeindex.omp}.

### 7.2 Pre-defined styles

\LaTeX{} defines two style-declaration macros \texttt{\textbackslash report} and \texttt{\textbackslash letter} You can use them at the beginning of your document if you are preparing these types of document and you don’t need to create your own macros.

The \texttt{\textbackslash report} declaration is intended to create reports. It sets default font size to 11 pt and \texttt{\textbackslash parindent} (paragraph indentation) to 1.2 em. The \texttt{\textbackslash tit} macro uses smaller font because we assume that “chapter” level will be not used in reports. The first page has no page number, but next pages are numbered (from number 2). Footnotes are numbered from one in whole document. The macro \texttt{\textbackslash author \{authors\}(end-line)} can be used when \texttt{\textbackslash report} is declared. It prints \texttt{\{authors\}} in italics at center of the line. You can separate authors by \texttt{\textbackslash n} to more lines.

The \texttt{\textbackslash letter} declaration is intended to create letters. It sets default font size to 11 pt and \texttt{\textbackslash parindent} to 0 pt. It sets half-line space between paragraphs. The page numbers are not printed. The \texttt{\textbackslash subject} macro can be used, it prints the word “Subject:” or “Věc” (or something else depending on current language) in bold. Moreover, the \texttt{\textbackslash address \{address\}} macro can be used when \texttt{\textbackslash letter} is declared. The usage of the \texttt{\textbackslash address} macro looks like:

\begin{verbatim}
\address
\{first line of address\}
\{second line of address\}
\{etc.\}
\{empty line\}
\end{verbatim}

It means that you need not to use any special mark at the end of lines: end of lines in the source file are the same as in printed output. The \texttt{\textbackslash address} macro creates \texttt{\textbackslash vtop} with address lines. The width of such \texttt{\textbackslash vtop} is equal to the most wide line used in it. So, you can use \texttt{\textbackslash hfill\textbackslash address...} in order to put the address box to the right side of the document. Or you can use \texttt{\{prefixed text\}\textbackslash address...} to put \texttt{\{prefixed text\}} before first line of the address.

Analogical declaration macros \texttt{\textbackslash book} or \texttt{\textbackslash slides} are not prepared. Each book needs an individual typographical care so you need to create specific macros for design. And you can find an inspiration of slides in \texttt{OPmac tricks 0017 and 0022}.

### 7.3 Lorem ipsum dolor sit

A designer needs to concentrate to the design of the output and maybe he/she needs a material for testing macros. There is the possibility to generate a neutral text for such experiments. Use \texttt{\textbackslash lorem[\{number\}]} or \texttt{\textbackslash lorem[\{from\}--\{to\}]}.

It prints a paragraph (or paragraphs) with neutral text. The numbers \texttt{\{number\}} or \texttt{\{from\}, \{to\}} must be in the range 1 to 150 because there are 150 paragraphs with neutral text prepared for you. The \texttt{\textbackslash lipsum} macro is equivalent to \texttt{\textbackslash lorem}. Example \texttt{\textbackslash lipsum[1-150]} prints all prepared paragraphs.

### 7.4 The last page

The number of the last page (it may be different from number of pages) is expanded by \texttt{\textbackslash lastpage} macro. It expands to \? in first \TeX{} run and to the last page in next \TeX{} runs.

There is an example for footlines in the format “current page / last page”:

\begin{verbatim}
\footline={\hss \texttt{\textbackslash fixedrm} \texttt{\textbackslash folio}/\texttt{\textbackslash lastpage} \hss}
\end{verbatim}
The \lastpage expands to the last \folio which is a decimal number or roman numeral (when \pageno is negative). If you need to know total pages used in the document, use \totalpages macro. It expands to zero (in first \TeX run) or to the number of all pages in the document (in next \TeX runs).

7.5 Use Op\TeX

The command \useOp\TeX (or \useoptex) does nothing in Op\TeX but it causes an error (undefined control sequence) when another format is used. You can use it as the first command in your document:

\useOp\TeX % we are using Op\TeX format, no \LaTeX

8 Summary

\tit Title (terminated by end of line)
\chap Chapter Title (terminated by end of line)
\sec Section Title (terminated by end of line)
\secc Subsection Title (terminated by end of line)

\maketoc \% table of contents generation
\ii item1,item2 \% insertion the items to the index
\makeindex \% the index is generated

\label [labname] \% link target location
\ref [labname] \% link to the chapter, section, subsection, equation
\pgref [labname] \% link to the page of the chapter, section, ...

\caption/t \% a numbered table caption
\caption/f \% a numbered caption for the picture
\eqmark \% a numbered equation

\begitems \% start list of the items
\enditems \% end of list of the items
\begtt \% start verbatim text
\endtt \% end verbatim text
\activettchar X \% initialization character X for in-text verbatim
\code \% another alternative for in-text verbatim
\verbinput \% verbatim extract from the external file
\begmulti num \% start multicolumn text (num columns)
\endmulti \% end multicolumn text

\cite [labnames] \% refers to the item in the lists of references
\rcite [labnames] \% similar to \cite but [] are not printed.
\sortcitations \shortcitations \nonumcitations \% cite format
\bib [labname] \% an item in the list of references
\usebib/? \% (style) bib-base \% direct using of .bib file, ? in \{s,c\}

\fontfam [FamilyName] \% selection of font family
\typosize [font-size/baselineskip] \% size setting of typesetting
\typoscale [factor-font/factor-baselineskip] \% size scaling
\thefontsize [size] \% current font size
\inspic file.ext \% insert a picture, extensions: jpg, png, pdf
\table \{rule\}{data} \% simple macro for the tables like in \LaTeX

\fnote \% footnote (local numbering on each page)
\mnote \% note in the margin (left or right by page number)

\hyperlinks \{color-in\}{color-out} \% PDF links activate as clickable
\outlines \{level\} \% PDF will have a table of contents in the left tab
\magscale[factor] \% resize typesetting, line/page breaking unchanged
\margins/pg format (left, right, top, bottom)unit \% margins setting

\report \letter \% style declaration macros

9 Compatibility with Plain \TeX

All macros of plain \TeX are re-written in Op\TeX. Common macros should be work in the same sense as in original plain \TeX. Internal control sequences like \p@ or \f@t are removed and mostly replaced by control sequences prefixed by \_ (like \_this). All primitives and common macros have two control sequences in prefixed and unprefixed form with the same meaning. For example \hbox is equal to \_hbox. Internal macros of Op\TeX have and use only prefixed form. User should use unprefixed forms, but prefixed forms are accessible too, because the \_ is set as a letter category code globally (in macro files and in users document too). User should re-define unprefixed forms of control sequences with no worries that something internal will be broken (only the sequence \par cannot be re-defined without internal change of \TeX behavior because it is hard-coded in \TeXs tokenization processor).

The Latin Modern 8bit fonts instead Computer Modern 7bit fonts are preloaded in the format, but only few ones. The full family set is ready to use after the command \fontfam[LMfonts] which reads the fonts in OTF format.

The accents macros like \', \v are undefined in Op\TeX. Use real letters like á, ř, ž in your source document instead these old accents macros. If you really want to use them, you can initialize them by \oldaccents command. But we don’t recommend it.

The default paper size is not set as letter with 1 in margins but as a4 with 2.5 cm margins. You can change it, for example by \margins/1 letter \(1,1,1,1)\) in. This example sets the classical plain \TeX page layout.

The origin for typographical area is not at top left 1 in 1 in coordinates but at top left paper corner exactly. For example, \hoffset includes directly left margin.