TUTORIAL III

BIBLIOGRAPHY

III.1. INTRODUCTION

Bibliography is the environment which helps the author to cross-reference one publication from the list of sources at the end of the document. LaTeX helps authors to write a well structured bibliography, because this is how LaTeX works—by specifying structure.

It is easy to convert the style of bibliography to that of a publisher's requirement, without touching the code inside the bibliography. We can maintain a bibliographic data base using the program BibTeX. While preparing the articles, we can extract the needed references in the required style from this data base. harvard and natbib are widely used packages for generating bibliography.

To produce bibliography, we have the environment **thebibliography**[1], which is similar to the **enumerate** environment. Here we use the command `\bibitem` to separate the entries in the bibliography and use `\cite` to refer to a specific entry from this list in the document. This means that at the place of citation, it will produce number or author-year code connected with the list of references at the end.

```
\begin{thebibliography}{widest-label}
      \bibitem{key1}
      \bibitem{key2}
    \end{thebibliography}
```

The `\begin{thebibliography}` command requires an argument that indicates the width of the widest label in the bibliography. If you know you would have between 10 and 99 citations, you should start with

```
\begin{thebibliography}{99}
```

You can use any two digit number in the argument, since all numerals are of the same width. If you are using customized labels, put the longest label in argument, for example `\begin{thebibliography}{Long-name}`. Each entry in the environment should start with

```
\bibitem{key1}
```

If the author name is Alex and year 1991, the key can be coded as ale91 or some such mnemonic string[2]. This *key* is used to cite the publication within the document text. To cite a publication from the bibliography in the text, use the `\cite` command, which takes with the corresponding key as the argument. However, the argument to `\cite` can also be two or more keys, separated by commas.

[1] Bibiliography environment need two compilations. In the first compilation it will generate file with `aux` extension, where `citation` and `bibcite` will be marked and in the second compilation `\cite` will be replaced by numeral or author-year code.

[2] Key can be any sequence of letters, digits and punctuation characters, except that it may not contain a comma (maximum 256 characters).

`\cite{`*key1*`}` `\cite{`*key1,key2*`}`

In bibliography, numbering of the entries is generated automatically. You may also add a note to your citation, such as page number, chapter number etc. by using an optional argument to the `\cite` command. Whatever text appears in this argument will be placed within square brackets, after the label.

`\cite[page~25]{`*key1*`}`

See below an example of bibliography and citation. The following code

```
It is  hard to write unstructured and disorganised documents using
\LaTeX~\cite{les85}.It is interesting to typeset one
equation~\cite[Sec 3.3]{les85} rather than setting ten pages of
running matter~\cite{don89,rondon89}.

\begin{thebibliography}{9}
\bibitem{les85}Leslie Lamport, 1985. \emph{\LaTeX---A Document
Preparation System---User's Guide and Reference Manual},
Addision-Wesley, Reading.

\bibitem{don89}Donald E. Knuth, 1989. \emph{Typesetting Concrete
Mathematics}, TUGBoat, 10(1):31-36.

\bibitem{rondon89}Ronald L. Graham, Donald E. Knuth, and Ore
Patashnik, 1989. \emph{Concrete Mathematics: A Foundation for
Computer Science}, Addison-Wesley, Reading.
\end{thebibliography}
```

produces the following output:

It is hard to write unstructured and disorganised documents using LaTeX [1]. It is interesting to typeset one equation [1, Sec 3.3] rather than setting ten pages of running matter [2,3].

Bibliography

[1] Leslie Lamport, 1985. *LaTeX—A Document Preparation System—User's Guide and Reference Manual*, Addision-Wesley, Reading.

[2] Donald E. Knuth, 1989. *Typesetting Concrete Mathematics*, TUGBoat, 10(1):31-36.

[3] Ronald L. Graham, Donald E. Knuth, and Ore Patashnik, 1989. *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley, Reading.

III.2. NATBIB

The natbib package is widely used for generating bibliography, because of its flexible interface for most of the available bibliographic styles. The natbib package is a re-implementation of the LaTeX `\cite` command, to work with both author–year and numerical citations. It is compatible with the standard bibliographic style files, such as plain.bst, as well as with those for harvard, apalike, chicago, astron, authordate, and of course natbib. To load the package; use the command.

`\usepackage[options]{natbib}`

III.2.1. Options for natbib

`round`	(default) for round parentheses
`square`	for square brackets
`curly`	for curly braces
`angle`	for angle brackets
`colon`	(default) to separate multiple citations with colons
`comma`	to use commas as separators
`authoryear`	(default) for author–year citations
`numbers`	for numerical citations
`super`	for superscripted numerical citations, as in *Nature*
`sort`	orders multiple citations into the sequence in which they appear in the list of references
`sort&compress`	as `sort` but in addition multiple numerical citations are compressed if possible (as 3–6, 15)
`longnamesfirst`	makes the first citation of any reference the equivalent of the starred variant (full author list) and subsequent citations normal (abbreviated list)
`sectionbib`	redefines `\thebibliography` to issue `\section*` instead of `\chapter*`; valid only for classes with a `\chapter` command; to be used with the `chapterbib` package
`nonamebreak`	keeps all the authors' names in a citation on one line; causes overfull hboxes but helps with some `hyperref` problems.

You can set references in the *Nature style* of citations (superscripts) as follows

```
\documentclass{article}
  \usepackage{natbib}
  \citestyle{nature}
\begin{document}
  . . . . . .
  . . . . . .
\end{document}
```

III.2.2. Basic commands

The natbib package has two basic citation commands, `\citet` and `\citep` for *textual* and *parenthetical* citations, respectively. There also exist the starred versions `\citet*` and `\citep*` that print the full author list, and not just the abbreviated one. All of these may take one or two optional arguments to add some text before and after the citation.

Normally we use author name and year for labeling the bibliography.

```
\begin{thebibliography}{widest-label}
  \bibitem{Leslie(1985)}{les85}Leslie Lamport, 1985.
    \emph{\LaTeX---A Document Preparation}...
  \bibitem{Donale(00)}{don89}Donald E. Knuth, 1989.
    \emph{Typesetting Concrete Mathematics},...
  \bibitem{Ronald, Donald and Ore(1989)}{rondon89}Ronald L. Graham, ...
\end{thebibliography}
```

Year in parentheses is mandatory in optional argument for bibitem. If year is missing in any of the bibitem, the whole author–year citation will be changed to numerical citation. To avoid this, give '(0000)' for year in optional argument and use partial citations (`\citeauthor`) in text.

Don't put 'space character' before opening bracket of year in optional argument.

```
\citet{ale91}                 ⇒  Alex et al. (1991)
\citet[chap.~4]{ale91}        ⇒  Alex et al. (1991, chap. 4)
\citep{ale91}                 ⇒  (Alex et al., 1991)
\citep[chap.~4]{ale91}        ⇒  (Alex et al., 1991, chap. 4)
\citep[see][]{ale91}          ⇒  (see Alex et al., 1991)
\citep[see][chap.~4]{jon91}   ⇒  (see Alex et al., 1991, chap. 4)
\citet*{ale91}                ⇒  Alex, Mathew, and Ravi (1991)
\citep*{ale91}                ⇒  (Alex, Mathew, and Ravi, 1991)
```

III.2.3. Multiple citations

Multiple citations may be made as usual, by including more than one citation key in the `\cite` command argument.

```
\citet{ale91,rav92}     ⇒  Alex et al. (1991); Ravi et al. (1992)
\citep{ale91,rav92}     ⇒  (Alex et al., 1991; Ravi et al. 1992)
\citep{ale91,ale92}     ⇒  (Alex et al., 1991, 1992)
\citep{ale91a,ale91b}   ⇒  (Alex et al., 1991a,b)
```

III.2.4. Numerical mode

These examples are for author–year citation mode. In numerical mode, the results are different.

```
\citet{ale91}                 ⇒  Alex et al. [5]
\citet[chap.~4]{ale91}        ⇒  Alex et al. [5, chap. 4]
\citep{ale91}                 ⇒  [5]
\citep[chap.~4]{ale91}        ⇒  [5, chap. 4]
\citep[see][]{ale91}          ⇒  [see 5]
\citep[see][chap.~4]{ale91}   ⇒  [see 5, chap. 4]
\citep{ale91a,ale91b}         ⇒  [5, 12]
```

III.2.5. Suppressed parentheses

As an alternative form of citation, `\citealt` is the same as `\citet` but *without any parentheses*. Similarly, `\citealp` is `\citep` with the parentheses turned off. Multiple references, notes, and the starred variants also exist.

```
\citealt{ale91}           ⇒  Alex et al. 1991
\citealt*{ale91}          ⇒  Alex, Mathew, and Ravi 1991
\citealp{ale91}           ⇒  Alex., 1991
\citealp*{ale91}          ⇒  Alex, Mathew, and Ravi, 1991
\citealp{ale91,ale92}     ⇒  Alex et al., 1991; Alex et al., 1992
\citealp[pg.~7]{ale91}    ⇒  Alex., 1991, pg. 7
\citetext{short comm.}    ⇒  (short comm.)
```

The `\citetext` command allows arbitrary text to be placed in the current citation parentheses. This may be used in combination with `\citealp`.

III.2.6. Partial citations

In author–year schemes, it is sometimes desirable to be able to refer to the authors without the year, or vice versa. This is provided with the extra commands

```
\citeauthor{ale91}     ⇒  Alex et al.
\citeauthor*{ale91}    ⇒  Alex, Mathew, and Ravi
\citeyear{ale91}       ⇒  1991
\citeyearpar{ale91}    ⇒  (1991)
```

III.2.7. Citations aliasing

Sometimes one wants to refer to a reference with a special designation, rather than by the authors, i.e. as Paper I, Paper II. Such aliases can be defined and used, textually and/or parenthetically with:

```
\defcitealias{jon90}{Paper~I}
```

```
\citetalias{ale91}   ⇒  Paper I
\citepalias{ale91}   ⇒  (Paper I)
```

These citation commands function much like `\citet` and `\citep`: they may take multiple keys in the argument, may contain notes, and are marked as hyperlinks.

III.2.8. Selecting citation style and punctuation

Use the command `\bibpunct` with one optional and six mandatory arguments:

1. The opening bracket symbol, `default = (`
2. The closing bracket symbol, `default = )`
3. The punctuation between multiple citations, `default = ;`
4. The letter ‘n’ for numerical style, or ‘s’ for numerical superscript style, any other letter for author–year, `default = author--year;`
5. The punctuation that comes between the author names and the year
6. The punctuation that comes between years or numbers when common author lists are suppressed (`default = ,`);

The optional argument is the character preceding a post-note, default is a comma plus space. In redefining this character, one must include a space if that is what one wants.

Example 1

```
\bibpunct{[}{]}{,}{a}{}{;}
```

changes the output of

```
\citep{jon90,jon91,jam92}
```

into

[Jones et al. 1990; 1991, James et al. 1992].

Example 2

```
\bibpunct[;]{(}{)}{,}{a}{}{;}
```

changes the output of

```
\citep[and references therein]{jon90}
```

into

(Jones et al. 1990; and references therein).

TUTORIAL IV

BIBLIOGRAPHIC DATABASES

Bibliographic database is a database in which all the useful bibliographic entries can be stored. The information about the various publications is stored in one or more files with the extension `.bib`. For each publication, there is a *key* that identifies it and which may be used in the text document to refer to it. And this is available for all documents with a list of reference in the field. This database is useful for the authors/researchers who are constantly referring to the same publications in most of their works. This database system is possible with the BibTeX program supplied with the LaTeX package.

IV.1. The BibTeX program

BibTeX is an auxiliary program to LaTeX that automatically constructs a bibliography for a LaTeX document from one or more databases. To use BibTeX, you must include in your LaTeX input file a `\bibliography` command whose argument specifies one or more files that contain the database. For example

```
\bibliography{database1,database2}
```

The above command specifies that the bibliographic entries are obtained from *database1.bib* and *database2.bib*. To use BibTeX, your LaTeX input file must contain a `\bibliographystyle` command. This command specifies the *bibliography style*, which determines the format of the source list. For example, the command

```
\bibliographystyle{plain}
```

specifies that entries should be formatted as specified by the `plain` bibliography style (`plain.bst`). We can put `\bibliographystyle` command anywhere in the document after the `\begin{document}` command.

IV.2. BibTeX style files

plain Standard BibTeX style. Entries sorted alphabetically with numeric labels.

unsrt Standard BibTeX style. Similar to plain, but entries are printed in order of citation, rather than sorted. Numeric labels are used.

alpha Standard BibTeX style. Similar to plain, but the labels of the entries are formed from the author's name and the year of publication.

abbrv Standard BibTeX style. Similar to plain, but entries are more compact, since first names, month, and journal names are abbreviated.

acm Alternative BibTeX style, used for the journals of the Association for Computing Machinery. It has the author name (surname and first name) in small caps, and numbers as labels.

apalike Alternative BIBTEX style, used by the journals of the American Psychology Association. It should be used together with the LATEX apalike package. The bibliography entries are formatted alphabetically, last name first, each entry having a hanging indentation and no label.

Examples of some other style files are:

abbrv.bst, abstract.bst, acm.bst, agsm.bst, alpha.bst, amsalpha.bst, authordate*i*.bst, authordate1-4.sty, bbs.bst, cbe.bst, cell.bst, dcu.bst, harvard.sty, ieeetr.bst, jtb.bst, kluwer.bst, named.bst, named.sty, natbib.sty, natbib.bst, nature.sty, nature.bst, phcpc.bst, phiaea.bst, phjcp.bst, phrmp.bst plainyr.bst, siam.bst

Various organisations or individuals have developed style files that correspond to the house style of particular journals or editing houses. We can also customise a bibliography style, by making small changes to any of the `.bst` file, or else generate our own using the `makebst` program.

IV.2.1. Steps for running BIBTEX with LATEX

1. Run LATEX, which generates a list of `\cite` references in its auxiliary file, `.aux`.
2. Run BIBTEX, which reads the auxiliary file, looks up the references in a database (one or more `.bib` files, and then writes a file (the `.bbl` file) containing the formatted references according to the format specified in the style file (the `.bst` file). Warning and error messages are written to the log file (the `.blg` file). It should be noted that BIBTEX never reads the original LATEX source file.
3. Run LATEX again, which now reads the `.bbl` reference file.
4. Run LATEX a third time, resolving all references.

Occasionally the bibliography is to include publications that were *not* referenced in the text. These may be added with the command

```
\nocite{key}
```

given anywhere within the main document. It produces no text at all but simply informs BIBTEX that this reference is also to be put into the bibliography. With `\nocite{*}`, *every* entry in all the databases will be included, something that is useful when producing a list of all entries and their keys.

After running BIBTEX to make up the `.bbl` file, it is necessary to process LATEX *at least twice* to establish both the bibliography and the in-text reference labels. The bibliography will be printed where the `\bibliography` command is issued; it infact inputs the `.bbl` file.

IV.3. CREATING A BIBLIOGRAPHIC DATABASE

Though bibliographic database creation demands more work than typing up a list of references with the `thebibliography` environment; it has a great advantage that, the entries need to be included in the database only once and are then available for all future publications even if a different bibliography style is demanded in later works, all the information is already on hand in the database for BIBTEX to write a new `thebibliography` environment in another format. Given below is a specimen of an entry in bibliographic database:

```
@BOOK{knuth:86a,
 AUTHOR           ="Donald E. Knuth",
```

```
  TITLE           ={The \TeX{}book},
  EDITION         ="third"
  PUBLISHER       ="Addison-Wesley",
  ADDRESS         ={Reading, MA},
  YEAR            =1986 }
```

The first word, prefixed `@`, determines the *entry_type*. The *entry_type* is followed by the reference information for that entry enclosed in curly braces { }. The very first entry is the *key* for the whole reference by which it is referred to in the `\cite` command. In the above example it is `knuth:86a`. The actual reference information is then entered in various *fields*, separated from one another by commas. Each *field* consists of a *field_name*, an = sign, with optional spaces on either side, and the *field text*. The *field_names* shows above are `AUTHOR`, `TITLE`, `PUBLISHER`, `ADDRESS`, and `YEAR`. The *field text* must be enclosed either in curly braces or in double quotation marks. However, if the text consists solely of a number, as for `YEAR` above, the braces or quotation marks may be left off.

For each entry type, certain fields are *required*, others are *optional*, and the rest are *ignored*. These are listed with the description of the various entry types below. If a required field is omitted, an error message will appear during the BIBTEX run. Optional fields will have their information included in the bibliography if they are present, but they need not be there. Ignored fields are useful for including extra information in the database that will not be output, such as a comment or an abstract of a paper. Ignored fields might also be ones that are used by other database programs.

The general syntax for entries in the bibliographic database reads

```
@entry_type{key,
  field_name = {field text},
  ....
  field_name = {field text} }
```

The names of the *entry_types* as well as the *field_names* may be written in capitals or lower case letters, or in a combination of both. Thus `@BOOK`, `@book`, and `@bOOk` are all acceptable variations.

The outermost pair of braces for the entire entry may be either curly braces { }, as illustrated, or parentheses ( ). In the latter case, the general syntax reads

```
  @entry_type(key, ... ..)
```

However, the *field text* may only be enclosed within curly braces {...} or double quotation marks ... as shown in the example above.

The following is a list of the standard entry types in alphabetical order, with a brief description of the types of works for which they are applicable, together with the required and optional fields that they take.

`@article`:	Entry for an article from a journal or magazine.
required fields:	author, title, journal, year.
optional fields:	volume, number, pages, month, note.
`@book`:	Entry for a book with a definite publisher.
required fields:	author or editor, title, publisher, year.
optional fields:	volume or number, series, address, edition, month, note.
`@booklet`:	Entry for a printed and bound work without the name of a publisher or sponsoring organisation.
required fields:	title.
optional fields:	author, howpublished, address, month, year, note.

`@conference`:	Entry for an article in conference proceedings.
required fields:	author, title, booktitle, year.
optional fields:	editor, volume or number, series, pages, address, month, organisation, publisher, note.
`@inbook`:	Entry for a part (chapter, section, certain pages) of a book.
required fields:	author or editor, title, chapter and/or pages, publisher, year.
optional fields:	volume or number, series, type, address, edition, month, note.
`@incollection`:	Entry for part of a book that has its own title.
required fields:	author, title, booktitle, publisher, year.
optional fields:	editor, volume or number, series, type, chapter, pages, address, edition, month, note.
`@inproceedings`:	Entry for an article in conference proceedings.
required fields:	author, title, booktitle, year.
optional fields:	editor, volume or number, series, pages, address, month, organisation, publisher, note.
`@manual`:	Entry for technical documentation.
required fields:	title.
optional fields:	author, organisation, address, edition, month, year, note.
`@masterthesis`:	Entry for a Master's thesis.
required fields:	author, title, school, year.
optional fields:	type, address, month, note.
`@misc`:	Entry for a work that does not fit under any of the others.
required fields:	none.
optional fields:	author, title, howpublished, month, year, note.
`@phdthesis`:	Entry for a PhD thesis.
required fields:	author, title, school, year.
optional fields:	type, address, month, note.
`@proceedings`:	Entry for conference proceedings.
required fields:	title, year.
optional fields:	editor, volume or number, series, address, month, organisation, publisher, note.
`@unpublished`:	Entry for an unpublished work with an author and title.
required fields:	author, title, note.
optional fields:	month, year.

IV.3.1. Example of a LaTeX file (sample.tex) using bibliographical database (bsample.bib)

```
\documentclass{article}
\pagestyle{empty}
\begin{document}

\section*{Example of Citations of Kind \texttt{plain}}
Citation of a normal book~\cite{Eijkhout:1991} and an edited
book~\cite{Roth:postscript}. Now we cite an article written by a
single~\cite{Felici:1991} and by multiple
authors~\cite{Mittlebatch/Schoepf:1990}. A reference to an
article inside proceedings~\cite{Yannis:1991}.
We refer to a manual~\cite{Dynatext} and a technical
report~\cite{Knuth:WEB}. A citation of an unpublished
work~\cite{EVH:Office}. A reference to a chapter in a
book~\cite{Wood:color} and to a PhD thesis~\cite{Liang:1983}.
```

```
An example of multiple
citations~\cite{Eijkhout:1991,Roth:postscript}.

\bibliographystyle{plain} %% plain.bst
\bibliography{bsample}    %% bsample.bib
\end{document}
```

IV.3.2. Procedure for producing references for the above file sample.tex which uses bibliographic data base bsample.bib

```
$ latex sample     % 1st run of LaTeX

$ bibtex sample    % BibTeX run
                   % Then sample.bbl file will
                   % be produced

$ latex sample     % 2nd run of LaTeX
```

If still unresolved citation references

```
$ latex sample     % 3rd run of LaTeX
```

TUTORIAL V

TABLE OF CONTENTS, INDEX AND GLOSSARY

V.1. TABLE OF CONTENTS

A *table of contents* is a special list which contains the section numbers and corresponding headings as given in the standard form of the sectioning commands, together with the page numbers on which they begin. Similar lists exist containing reference information about the floating elements in a document, namely, the *list of tables* and *list of figures*. The structure of these lists is simpler, since their contents, the captions of the floating elements, all are on the same level.

Standard LaTeX can automatically create these three contents lists. By default, LaTeX enters text generated by one of the arguments of the sectioning commands into the `.toc` file. Similarly, LaTeX maintains two more files, one for the list of figures (`.lof`) and one for the list of tables (`.lot`), which contain the text specified as the argument of the `\caption` command for figures and tables.

`\tableofcontents` produces a table of contents. `\listoffigures` and `\listoftables` produce a list of figures and list of tables respectively. These lists are printed at the point where these commands are issued. Occasionally, you may find that you do not like the way LaTeX prints a table of contents or a list of figures or tables. You can fine-tune an individual entry by using the optional arguments to the sectioning command or `\caption` command that generates it. Formatting commands can also be introduced with the `\addtocontents`. If all else fails, you can edit the `.toc`, `lof`, `lot` files yourself. Edit these files only when preparing the final version of your document, and use a `\nofiles` command to suppress the writing of new versions of the files.

V.1.1. Additional entries

The *-form sectioning commands are not entered automatically in the table of contents. LaTeX offers two commands to insert such information directly into a contents file:

`\addtocontents{`*file*`}{`*text*`}` `\addcontentsline{`*file*`}{`*type*`}{`*text*`}`

file	The extension of the contents file, usually `toc`, `lof` or `lot`.
type	The type of the entry. For the `toc` file the *type* is normally the same as the heading according to the format of which an entry must be typeset. For the `lof` or `lot` files, `figure` or `table` is specified.
text	The actual information to be written to the *file* mentioned. LaTeX commands should be protected by `\protect` to delay expansion

The `\addtocontents` command does not contain a *type* parameter and is intended to enter *user-specific* formatting information. For example, if you want to generate additional spacing in the middle of a table of contents, the following command can be issued:

```
\addtocontents{toc}{\protect\vspace{2ex}}
```

The \addcontentsline instruction is usually invoked *automatically* by the document sectioning commands, or by the \caption commands. If the entry contains numbered text, then \numberline must be used to separate the section number (*number*) from the rest of the text for the entry (*heading*) in the *text* parameter:

\protect\numberline{*number*}{*heading*}

For example, a \caption command inside a figure environment saves the text annotating the figure as follows:

\addcontentsline{lof}{figure}{\protect\numberline{\thefigure}*captioned text*}

Sometimes \addcontentsline is used in the source to complement the actions of standard LaTeX. For instance, in the case of the starred form of the section commands, no information is written to the .toc file. So if you do not want a heading number (starred form) but an entry in the .toc file you can write something like:

```
\chapter*{Forward}
\addcontentsline{toc}{chapter}{\numberline{}Forward}
```

This produces an indented "chapter" entry in the table of contents, leaving the space where the chapter number would go free. Omitting the \numberline command would typeset the word "Forward" flush left instead.

V.1.2. Typesetting a contents list

As discussed above, contents lists consist of entries of different types, corresponding to the structural units that they represent. Apart from these standard entries, these lists may contain any commands. A standard entry is specified by the command:

\contentsline{*type*}{*text*}{*page*}

type Type of the entry, e.g. section, or figure.
text Actual text as specified in the argument of the sectioning or \caption commands.
page Pagenumber.

Note that section numbers are entered as a parameter of the \numberline command to allow formatting with the proper indentation. It is also possible for the user to create a table of contents by hand with the help of the command \contentsline. For example:

```
\contentsline {section}
  {\numberline {2.4}Structure of the Table of Contents}{31}
```

To format an entry in the table of contents files, standard LaTeX makes use of the following command:

\@dottedtocline{*level*}{*indent*}{*numwidth*}{*text*}{*page*}

The last two parameters coincide with those of \contentsline, since the latter usually invokes \@dottedtocline command. The other parameters are the following:

level The nesting level of an entry. This parameter allows the user to control how many nesting levels will be displayed. Levels greater than the value of counter tocdepth will not appear in the table of contents.
indent This is total indentation from the left margin.
numwidth The width of the box that contains the number if *text* has a \numberline command. This is also the amount of extra indentation added to the second and later lines of a multiple line entry.

Additionally, the command \@dottedtocline uses the following formatting parameters, which specify the visual appearance of all entries:

\@pnumwidth	The width of the box in which the page number is set.
\@tocmarg	The indentation of the right margin for all but the last line of multiple line entries. Dimension, but changed with \renewcommand.
\@dotsep	The separation between dots, in mu (math units). It is a pure number (like 1.7 or 2). By making this number large enough you can get rid of the dots altogether. Changed with \renewcommand as well.

V.1.3. Multiple tables of contents

The minitoc package, initially written by Nigel Ward and Dan Jurafsky and completely redesigned by Jean-Pierre Drucbert, creates a mini-table of contents (a "minitoc") at the beginning of each chapter when you use the book or report classes.

The mini-table of contents will appear at the beginning of a chapter, after the \chapter command. The parameters that govern the use of this package are discussed below:

Table V.1: Summary of the minitoc parameters

\dominitoc	Must be put just in front of \tableofcontents, to initialize the minitoc system (Mandatory).
\faketableofcontents	This command replaces \tableofcontents when you want minitocs but not table of contents.
\minitoc	This command must be put right after each \chapter command where a minitoc is desired.
\minitocdepth	A LaTeX counter that indicates how many levels of headings will be displayed in the minitoc (default value is 2).
\mtcindent	The length of the left/right indentation of the minitoc (default value is 24pt).
\mtcfont	Command defining the font that is used for the minitoc entries (The default definition is a small roman font).

For each mini-table, an auxiliary file with extension .mtc<N> where <N> is the chapter number, will be created.

By default, these mini-tables contain only references to sections and subsections. The minitocdepth counter, similar to tocdepth, allows the user to modify this behaviour.

As the minitoc takes up room on the first page(s) of a chapter, it will alter the page numbering. Therefore, three runs normally are needed to get correct information in the mini-table of contents.

To turn off the \minitoc commands, merely replace the package minitoc with minitocoff on your \usepackage command. This assures that all \minitoc commands will be ignored.

V.2. INDEX

To find a topic of interest in a large document, book, or reference work, you usually turn to the table of contents or, more often, to the index. Therefore, an index is a very important part of a document, and most users' entry point to a source of information is precisely through a pointer in the index. The most generally used index preparation program is *MakeIndex*.

```
Page vi:  \index{animal}
Page 5:   \index{animal}
Page 6:   \index{animal}
Page 7:   \index{animal}
Page 11: \index{animalism|see{animal}}
Page 17: \index{animal@\emph{animal}}
         \index{mammal|textbf}
Page 26: \index{animal!mammal!cat}
Page 32: \index{animal!insect}
```

(a) The input file

```
\indexentry{animal}{vi}
\indexentry{animal}{5}
\indexentry{animal}{6}
\indexentry{animal}{7}
\indexentry{animalism|seeanimal}{11}
\indexentry{animal@\emph{animal}}{17}
\indexentry{mammal|textbf}{17}
\indexentry{animal!mammal!cat}{26}
\indexentry{animal!insect}{32}
```

(b) The `.idx` file

```
\begin{theindex}
  \item animal, vi, 5–7
    \subitem insect, 32
    \subitem mammal
      \subsubitem cat, 26
  \item \emph{animal}, 17
  \item animalism, \see{animal}{11}
\indexspace
  \item mammal, \textbf{17}
\end{theindex}
```

(c) The `.ind` file

animal, vi 5–7
insect, 32
mammal
cat, 26
animal, 17
animalism, *see* animal

mammal, **17**

(d) The typeset output

Figure V.1: Stepwise development of index processing

Each `\index` command causes LaTeX to write an entry in the `.idx` file. This command writes the text given as an argument, in the `.idx` file. This `.idx` will be generated only if we give `\makeindex` command in the preamble otherwise it will produce nothing.

`\index{`*index_entry*`}`

To generate index follow the procedure given below:

1. Tag the words inside the document, which needs to come as index, as an argument of `\index` command.
2. Include the makeidx package with an `\usepackage` command and put `\makeindex` command at the preamble.
3. Put a `\printindex` command where the index is to appear, normally before `\end{document}` command.
4. LaTeX *file*. Then a raw index (`file.idx`) will be generated.
5. Then run `makeindex`. (makeindex *file.idx* or makeindex *file*). Then two more files will be generated, *file.ind* which contains the index entries and *file.ilg*, a transcript file.
6. Then run LaTeX again. Now you can see in the dvi that the index has been generated in a new page.

V.2.1. Simple index entries

Each `\index` command causes LaTeX to write an entry in the `.idx` file. For example

`\index{`*index_entry*`}`

```
fonts                                  Page ii:  \index{table|(}
      Computer Modern, 13–25           Page xi:  \index{table|)}
      math, see math, fonts            Page 5:   \index{fonts!PostScript|(}
      PostScript, 5                              \index{fonts!PostScript|)}
table, ii–xi, 14                       Page 13   \index{fonts!Computer Modern |(}
                                       Page 14: \index{table}
                                       Page 17: \index{fonts!math|see{math, fonts}}
                                       Page 21: \index{fonts!Computer Modern}
                                       Page 25: \index{fonts!Computer Modern|)}
```

Figure V.2: Page range and cross-referencing

V.2.2. Sub entries

Up to three levels of index entries (main, sub, and subsub entries) are available with LaTeX-`MakeIndex`. To produce such entries, the argument of the `\index` command should contain both the main and subentries, separated by `!` character.

Page 5: `\index{`*dimensions!rule!width*`}`

This will come out as

dimensions
    rule
        width, 5

V.2.3. Page ranges and cross-references

You can specify a page range by putting the command `\index{...|(}` at the beginning of the range and `\index{...|)}` at the end of the range. Page ranges should span a homogeneous numbering scheme (e.g., Roman and Arabic page numbers cannot fall within the same range).

You can also generate cross-reference index entries without page numbers by using the `see` encapsulator. Since "see" entry does not print any page number, the commands `\index{...|see{...}}` can be placed anywhere in the input file after the `\begin{document}` command. For practical reasons, it is convenient to group all such cross-referencing commands in one place.

V.2.4. Controlling the presentation form

Sometimes you may want to sort an entry according to a key, while using a different visual representation for the typesetting, such as Greek letters, mathematical symbols, or specific typographic forms. This function is available with the syntax: *key@visual*, where *key* determines the alphabetical position and the string *value* produces the typeset text of the entry.

For some indexes certain page numbers should be formatted specially, with an italic page number (for example) indicating a primary reference, and an *n* after a page number denoting that the item appears in a footnote on that page. *MakeIndex* allows you to format an individual page number in any way you want by using the encapsulator syntax specified | character. What follows the | sign will "encapsulate" or enclose the page number associated with the index entry. For instance, the command `\index{keyword|xxx}` will produce a page number of the form `\xxx{n}`, where *n* is the page number in question.

```
delta, 14                          Page ii:  \index{tabular|textbf}
δ, 23                              Page 5:   \index{ninety-five}
delta wing, 16                     Page 7:   \index{tabbing}
flower, 19                         Page 14:  \index{delta}
ninety, 26                         Page 16:  \index{delta wing}
xc, 28                             Page 19:  \index{flower@\textbf{flower}}
ninety-five, 5                     Page 21:  \index{tabular|textit}
tabbing, 7, 34–37                  Page 22:  \index{tabular|nn}
tabular, ii, 21, 22n               Page 23:  \index{delta@δ}
tabular environment, 23                      \index{tabular@\texttt{tabular}
                                               environment}
                                   Page 26:  \index{ninety}
                                   Page 28:  \index{ninety@xc}
                                   Page 34:  \index{tabbing|(textit}
                                   Page 36:  \index{tabbing|)}
```

Figure V.3: Controlling the presentation form

```
@ sign, 2                                  \index{bar@\texttt{"|}|see{vertical bar}}
|, see vertical bar            Page 1:     \index{quote (\verb+""+)}
exclamation (!), 4                         \index{quote@\texttt{""} sign}
    Ah!, 5                     Page 2:     \index{atsign@\texttt{"@} sign}
Mädchen, 3                     Page 3:     \index{maedchen@M\"{a}dchen}
quote ("), 1                   Page 4:     \index{exclamation ("!)}
" sign, 1                      Page 5:     \index{exclamation ("!)!Ah"!}
```

Figure V.4: Printing those special characters

Similarly, the command `\index{keyword|(xxx)}` will generate a page range of the form `\xxx{n-m}`

```
\newcommand{\nn}[1]{#1n}
```

V.2.5. Printing those special characters

To typeset one of the characters having a special meaning to *MakeIndex* (`!`, `"`, `@`, or `|`) in the index, precede it with a `"` character. More precisely, any character is said to be quoted if it follows an unquoted `"` that is not part of a `\"` command. The latter case is for allowing umlaut characters. Quoted `!`, `@`, `"`, or `|` characters are treated like ordinary characters, losing their special meaning. The `"` preceding a quoted character is deleted before the entries are alphabetised.

V.3. GLOSSARY

A 'glossary' is a special index of terms and phrases alphabetically ordered together with their explanations. To help set up a glossary, LaTeX offers the commands

`\makeglossary`	in the preamble and
`\glossary{`*glossary-entry*`}`	in the text part

which function just like the commands for making up an index register. The entries are written to a file with extension `.glo` after the command `\makeglossary` has been given in the preamble. The form of these file entries from each `\glossary` command is

`\glossaryentry\textit{glossary-entry}{`*pagenumber*`}`

The information in the `.glo` file can be used to establish a glossary. However, there is no equivalent to the `theindex` environment for a glossary, but a recommended structure is the `description` environment or a special list environment.

TUTORIAL VI

DISPLAYED TEXT

There are many instances in a document when we want to visually separate a portion of text from its surrounding material. One method of doing this is to typeset the distinguished text with added indentation. It is called *displaying*. LaTeX has various constructs for displaying text depending the nature of the displayed text.

VI.1. BORROWED WORDS

Quotations are often used in a document, either to add weight to our arguments by referring to a higher authority or because we find that we cannot improve on the way an idea has been expressed by someone else. If the quote is a one-liner, we can simply include it within double-quotes and be done with it (remember how to use quotes in TeX?) But if the quotation is several lines long, it is better to display it. Look at the following example:

Some mathematicians elevate the spirit of Mathematics to a kind of intellectual aesthetics. It is best voiced by Bertrand Russell in the following lines.

> The true spirit of delight, the exaltation, the sense of being more than man, which is the touchstone of the highest excellence, is to be found in Mathematics as surely as in poetry.... Real life is, to most men, a long second best, a perpetual compromise between the ideal and the possible; but the world of pure reason knows no compromise, no practical limitations, no barriers to the creative activity embodying in splendid edifices the passionate aspiration after the perfect, from which all great work springs.

Yes, to men like Russell, Mathematics is more of an art than science.

This was type set as shown below

```
Some mathematicians elevate the spirit of Mathematics  to a kind of
intellectual aesthetics. It is best voiced  by Bertrand Russell in the
following lines.
 \begin{quote}
   The true spirit of ..............................from which
   all great work springs.
 \end{quote}
```

Note that here we give instructions to TeX to typeset some material in a separate paragraph with additional indentation on either side and indicate the start and end of material requiring special treatment, by means of the commands

```
\begin{quote} ... \end{quote}
```