3. Displayed Text

There are a variety of ways to display or emphasize the text. Changing font style or font size, centering, indentation, marking and paragraphs and so on. Latex supplies us with commands for the most common forms of display.

3.1 Changing font

In typography, a set of letters, numbers and characters of a certain size and appearance is called a font. The 3 possible basic sizes are 10, 11 and 12 pt depending on the size options 10pt (default), 11pt and 12pt.

The parenthesis characters ( ) extend the full height and depth of the font size.

The differences in the visual appearance of the three standard sizes are,

This is an example of the 10pt font. ( )
And this is the 11 pt font for comparision. ( )
And finally this is a sample of 12 pt font. ( )

3.2 Emphasis

The typesetter will transform a text into italics for the printed version, with the command \emph (or) \em.

The \em declarations functions just as the other font declarations. The change of font remains in effect until negated by another appropriate declaration or until the end of the current environment.

An environment is initiated with the command,

\begin{ } → name of the environment
\end{ } → name of the environment

: name of the environment must be the same in both the commands

This is easiest way to emphasize short pieces of text.

For example,

This is the easiest way to \emph{emphasize} short pieces of text.

The \em declaration is more suitable for longer text that is enclosed in an environment,

Example:

```
\begin{em}
text
\end{em}
```

will display the output as the text in italic.

Both the declaration and the command switch to an emphasizing font.

That means, if the current font is upright it switches to italics, whereas if the text is already in italic, an upright font is selected.

This can be done by using nested emphasis.

Example:

The \emph{first}, second, and \emph{ third font switch}

The {\em first}, {\em second}, and {\em third font switch}

both produce, The *first*, second, and *third* font switch

3.3 Choice of font size

The following declarations are available in Latex for changing the font size:

\tiny – smallest
\scriptsize – very small
\footnotesize – smaller
\small – small
\normalsize – normal
\large – Large
\Large – Larger
\LARGE – even Larger
\huge – Still Larger
\Huge – Largest

The standard size is 10pt, which is then the size selected with \normalsize.

The font size declarations behave as all other declarations; they make an immediate change that remains in effect until loss its effect by another size declaration (or) until the end of the current environment.

Example:

normal{ \large large \Large larger} normal again

normal large larger normal again

Changing the font size with one of the above commands also automatically changes the interline spacing. For every font size there is a corresponding natural linespacing. This may be altered at any time. If the natural line spacing is 12pt, the command \setlength{\baselineskip}{15pt} will increase it in to 15pt.

3.4 Font attributes

The size of a font is only one of several attributes that may be used to describe it. It is possible to select fonts strictly by these attributes, which was introduced as a part of LaTeX2e, with the NFSS (New font selection scheme).

There are some declarations and corresponding commands to simplify this procedure.

(1) Family : for the general overall style. The standard LaTeX installation provides 3 families with declarations,

\rmfamily – to switch to a Roman font
\ttfamily – to switch to a typewriter font
\sffamily – to select a sans serif font

(2) Shape : for the form of the font. The shape declarations are

\upshape – to switch to an upright font.
\itshape – to select an italic shape.
\slshape – to choose a font that is slanted.
\scshape – to switch to CAPS and SMALL CAPS.

(3) Series : for the width and/or weight (boldness) of the font. The declarations are,

\mdseries – to switch to medium weight
\bfseries – to select a bold face font.

Example:

normal and {\bfseries bold and {\slshape slanted} and back} again.

produces: normal and **bold and *slanted* and back** again.

Finally, the declaration \normalfont resets all the attributes back to their defaults: Roman, upright, medium weight.

3.5 Font commands

For each of the font declarations listed above, there is a corresponding font command that sets its argument in a font with the specified attribute.

Family: \textrm{text} \texttt{text} \textsf{text}

Shape: \textup{text} \textit{text} \textsl{text}
\textsc{text}

Series: \textmd{text} \textbf{text}

Default: \textnormal{text}

Emphasis: \emph{text}

Note that the \emph command is included here, corresponding to the declaration \em. The argument of \textnormal is set in the standard font selected with \normalfont.

The use of such commands to change the font for short pieces of text or single words.

The previous example now becomes,

normal and \textbf{bold and \textsl{slanted} and back} again.

produces: normal and **bold and *slanted* and back** again.

The old two-letter Tex declarations such as \bf and \tt, which were part of Latex 2.09, are still available but are now considered obsolete and should be avoided.

3.6 Additional fonts

Tex installation has even more fonts and sizes than those listed above. They may be available for use within a Latex document either by referring to them by name or by their attributes if they have been set up for NFSS.

To load a new font explicitly by name, the command

\newfont{\fnt}{name scaled factor}

(or)

\newfont{\fnt}{name at size}

is given, which assigns the font to the new command named \fnt. In the first case factor is a number, 1000 times scaling factor that is is to be used to magnify or reduce the font from its basic size. In the second case, the font is scaled to be of size specified.

3.7 Character Sets and Symbols

The individual character sets are each stored in their own files. The names of the 75 standard Tex fonts are available. Each symbol within a character set is addressed by means of a number between 0 and 127 (or 255) by the command

\symbol{num}

will produce that symbol with the internal identification number 'num' in the current font.

Example: é - internal number 62

Centering and indenting

(1) Centered text (2) One-sided justification (3) Two-sided indentation (4) verse indentations

(1) Centered text

The environment

```
\begin{center}
line1 \\ line2 \\ ..... linen
\end{center}
```

Centers the sections of text. If the text is too long for one line, it is splitted as many lines by using uniform word spacing.

Within an environment, the command \centering may be used to center the text. The effect of this declaration lasts until the end of that environment.

A single line may be centered by using the command,

```
\centerline{text}
```

(2) One-sided justification

The environments

```
\begin{flushleft}
line1 \\ line2 \\ ... linen
\end{flushleft}
```

```
\begin{flushright}
line1 \\ line2 \\ .... linen
\end{flushright}
```

produce text that is left (or) right justified.

As in the case of center environment, if a section of text does not fit on to a single line, it is spread over several lines using uniform word spacing, again word division does not occur.

The same results may be produced within an environment with the declarations,

\raggedright – replacing the flushleft environment

\raggedleft – replacing the flushright environment

(3) Two-sided indentation

A section of text may be displayed by indenting it by an equal amount on both sides, with the environments,

```
\begin{quote}
text
\end{quote}
```

```
\begin{quotation}
text
\end{quotation}
```

Additional vertical spacing is inserted above and below the text to separate its visual from the normal text. The text to be displayed may be of any length, it may be a sentence (or) a whole paragraph (or) several paragraphs. paragraphs are separated as usual with an empty line, although no empty lines are needed at the beginning and end of the text since additional vertical spacing is inserted here anyway.

In the quotation environment, paragraphs are marked by extra indentation of the first line, whereas in the quote environment, they are indicated with more vertical spacing between them.

(4) Verse indentations

For indenting rhymes, poetry, verses, etc. on both sides, the environment

```
\begin{verse}
poem
\end{verse}
```

is more appropriate.

Example:

```
Contrary to popular belief, limericks are not
always ribald.
\begin{verse}
A mathematician confided \\ A Möbius is one
sided \\ you'll get quite a laugh
\end{verse}
There is an extension of this to Klein's bottle also
```

Produces: (*)

Stanzas are seperated by blank lines while the individual lines of the stanza are divided by the \\ command.

If a line is too long for the reduced text width, it will be left and right justified and continued on the next line.

The above indenting schemes may be nested inside one another. Within a quote environment there may be another quote, quotation (or) verse environment. Each time, additional indentations are created on both sides of the text and vertical spacing is added above and below. Maximum, a number of six nestings are allowed.

(*)
Contrary to popular belief, limericks are not always ribald.
A mathematician confided
A Möbius is one sided
You'll get quite a laugh
There is an ...

Lists

There are three environments available for producing formatted lists:

`\begin{itemize}`	`\begin{enumerate}`	`\begin{description}`
text list	text list	text list
`\end{itemize}`	`\end{enumerate}`	`\end{description}`

In each of these environments, the 'text list' is indented from the left margin and a label or marker is included. What type of label is used depends on the selected list environment. The command to produce the label is \item.

Sample itemize

The itemize environment gives a bullet-list.

Example:

```
One should keep the following in mind when using \TeX
\begin{itemize}
\item \TeX\ is a typesetting language and not a word processor
\item Tex is a program and not an application
\item There is no meaning in compaling \TeX\ to a word processor, since the design purposes are different
\end{itemize}
Being a program, \TeX\ offers a high degree of flexibility.
```

Produces:

One should keep the following in mind when using TEX

- TEX is a typesetting language and not a word processor
- Tex is a program and not an application
- There is no meaning in comparing TEX to a word processor since the design purposes are different

Being a program, TEX offers a high degree of flexibility

Sample enumerate

A numbered list is produced by the enumerate environment in Latex.

Example:

The three basic steps to prepare a document using Latex:

```
\begin{enumerate}
\item prepare a source file with the extension .tex
\item Compile it with \LaTeX\ to produce a PDF file
\item Print the document
\end{enumerate}
```

Produces:

The three basic steps to prepare a document using Latex:

1. Prepare a source file with the extension .tex
2. Compile it with LATEX to produce a PDF file
3. Print the document

Sample description

There is a third type of list available in Latex, which is used in typesetting lists like this.

Example:

Let us take stock of what we have learnt

```
\begin{description}
\item[\TeX] A typesetting program
\item[Emacs] A text editor and also
  \begin{description}
  \item a programming environment
  \item a mailer
  \item and a lot else besides
  \end{description}
```

```
\item [AbiWord] A word processor
\end{description}
```

Produces:

Let us take stock of what we have learnt

TEX A typesetting program

Emacs A text editor and also
a programming environment
a mailer
and a lot else besides

AbiWord A word processor

The \item [option] command contains an optional argument that appears in bold face as the label.

Nested Lists

The above lists may be included within one another, either mixed or of one type, to a depth of four levels.

The type of label used depends on the depth of the nesting. The indentation is always relative to the left margin of the enclosing list.

Example:

```
\documentclass [12pt] {article}
\begin{document}
This is the Latex document
 \begin{itemize}
 \item The first item in the first level
 \item The second item in the first level
  \begin{itemize}
  \item The first item in the second level
```

```
\item The second item in the second level
\begin{itemize}
\item The first item in the third level
\item The second item in the third level
\begin{itemize}
\item The first item in the fourth level
\item The second item in the fourth level
\end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
\end{document}
```

Produces:

This is the Latex document

- The first item in the first level
- The second item in the first level
  - The first item in the second level
  - The second item in the second level
  - The second item in the third level
    - * The first item in the third level
    - * The second item in the fourth level
      - · The first item in the fourth level
      - · The second item in the fourth level

The above specified labels (•, –, *, ·) are default. If you are not satisfied with these labels, we can change this by using the command,

\renewcommand{\labelitemn}{ } → label name

(n → number)

Example:

```
\renewcommand{\labelitemi}{$\triangleright$}
  \begin{itemize}
   \item First item of a new list
   \item Second item
    \end{itemize}
```

Produces:

- First item of a new list
- Second item

Theorem-Like Declarations:

In Mathematical documents we often have special statements such as axioms and theorems [which are the conclusions obtained, sometimes known by other names like propositions or lemmas].

These are often typeset in different font to distinguish them from surrounding text and given a name and a number for subsequent reference.

We use the term theorem-like statements for all such statements. Latex provides the declaration \newtheorem to define the theorem-like statements needed in a document.

\newtheorem{struct_type}{struct title}[in_counter]

Latex will keep track of the numbering automatically.

Here, the argument struct_type is the name assign to the environment and the second argument struct_title is the word is printed in bold face followed by the running number.

However, if the name of an existing counter such as chapter is given for in_counter, the numbering is reset every time that counter is augmented and both are printed together [Ex: Theorem 4.1]

The predefined structures are called with the command,

\begin{struct_type}[extra_title]

text

\end{struct_type}

The optional extra_title also appears in bold face within parentheses ( ).

Example

```
\newtheorem{thm}{Theorem}
\begin{thm}
The sum of the angles of a triangle is $180^\circ$.
\end{thm}
```

Produces

Theorem 1. The sum of the angles of a triangle is 180°.

Occasionally a structure is not numbered on its own but together with another structure. This can be included in the definition with another optional argument

```
\newtheorem{struct_type}[num_like]{struct_name}
```

where num_like is the name of an existing theorem structure that shares the same counter.

The package amsthm affords a high level of customization in formatting theorem-like statements. The predefined styles available in this package.

Tables

Constructing tables

The environments tabular, tabular* and array are the basic tools with which tables and matrices can be constructed. The syntax for these environments is,

(1)
```
\begin{array}[pos]{cols}
    rows
\end{array}
```

(2)
```
\begin{tabular}[pos]{cols}
    rows
\end{tabular}
```

(3)
```
\begin{tabular*}{width}[pos]{cols}
    rows
\end{tabular*}
```

The array enviroment can only be applied in mathematical mode. All three enviroments actually create a minipage.

pos – vertical positioning argument

t – the top line of the table is aligned with the baseline of the external line of text.

b – the bottom line of the table is aligned with the external baseline.

width – This argument applies only to the tabular* environment and determines its overall width.

cols – The column formatting argument. There must be an entry for every column

The possible column formatting symbols are

l – the column contents are left justified

r – the column contents are right justified

c – the column contents are centered.

p{wth} – the text in this column is set into lines of width 'wth'.

*{num}{cols} – the column format contained in cols is reproduced 'num' times.

example *{5}{|c} is same as |c|c|c|c|c|

The available formatting symbols for the left and right borders and for the intercolumn spacing are

| – draws a vertical line

|| – draws two vertical lines next to each other

@{text} – this entry is referred to as an @-expression and inserts text in every line of the table between two columns it appears.

An @-expression removes the inter-column spacing that is automatically put between each pair of columns.

An \extracolsep{with} command will produce extra spacing of amount 'with' between all the columns until to meet another \extracolsep command.

rows - contain the actual entries in the table. Each horizontal row being terminated with a \\ command.

\hline - This command is used to produce horizontal line before the row and after the row

\cline{m-n} - This command draws a horizontal line from the left side of column 'm' to the right side of column 'n'.

\multicolumn{num}{col}{text} - This command combines the following num columns into a single column with their total width including inter-column spacings.

\vline - This command draws a vertical line with the height of the row

In order to center the table, it must be enclosed within,

```
\begin{center}
  table
\end{center}
```

Table examples

(1)
```
\begin{document}
 \begin{tabular}{|r|r|} \hline
  Mango & Mixed \\
  \hline
  Jackfruit & Kolli Hills \\ \hline
  Banana & Green \\ \hline
  \end{tabular}
   \end{document}
```

Produces

Mango	Mixed
Jackfruit	Kolli Hills
Banana	Green

(2)
```
\begin{document}
 \begin{center}
 \begin{tabular}{|l|l|p{2cm}|}
  \hline
  h1 & h2 & h3 \\ \hline
  a & b & this is the size of my column \\ \hline
  \end{tabular}
  \end{center}
  \end{document}
```

produces

h1	h2	h3
a	b	this is the size of my column

3)
```
\begin{document}
\begin{tabular} {||l|c|c|c|r|} \hline
\multicolumn 2 {|c|} {Fruit details} &
\multicolumn 3 {|c|} {Cost calculations} \\ \hline
Fruit & Type & No. of units & cost/unit & cost (Rs.) \\
\hline
Mango & Mixed & 20 & 75 & 1,500 \\ \hline
Jackfruit & Kolli Hills & 10 & 50 & 500 \\ \hline
Banana & Green & 10 & 20 & 200 \\ \hline
\end{tabular}
\end{document}
```

Produces

Fruit details		Cost calculations		
Fruit	Type	No. of units	cost/unit	cost (Rs.)
Mango	Mixed	20	75	1,500
Jackfruit	Kolli Hills	10	50	500
Banana	Green	10	20	200

Printing literal text

Occasionally it is necessary to print text exactly as it is typed with all special characters, blanks and line breaks appearing literally, unformated and in a typewriter font. This is accomplished with the environments,

```
\begin{verbatim}
text
\end{verbatim}
```

```
\begin{verbatim*}
text
\end{verbatim*}
```

A new line is inserted before and after these enviroments. With the *-form, blanks are printed with the symbol ␣ to make them visible.

To display a single command in-line, the command \verb and \verb* can be used.

To display a whole block we can use \begin{verbatim} to open the environment and \end{verbatim} to close it.

Example

(1)
```
\begin{verbatim}
  \addtocounter{footnote}{-1}\footnotetext{Small insects}
   \stepcounter{footnote}\footnotetext{Large mammals}
\end{verbatim}
```

(2) \verb=\emph{words of text}= \emph{words of text}
\verb*=\emph{words of text}= \emph{words␣of␣text}
where the first character after \verb or \verb* (here =) is the delimiter, such that all text after this is to be printed literally. This character may not appear in the literal text.

In contrast to the behaviour in the verbatim environment, the literal text must be all on one line in the input text, otherwise an error message is printed.

Extension packages for literal text

Package: alltt — An alltt environment prints its contents literally in a typewriter font except that the characters \ { } retain their normal meaning. Latex commands can also be included within the literal text.

Example:

```
\begin{alltt}
Underlining \underline{typewriter}
 text is also possible.
Note that dollar ($) and
 percent (%) signs are
 treated \emph{literally}.
\end{alltt}
```

Produces:

Underlining typewriter text is also possible. Note that dollar ($) and percent (%) signs are treated literally

package: shortvrb

This package offers a shorthand for the \verb command. After issuing \makeshortverb{\|}, one can print short literal text. The counter command \DeleteShortVerb{\|} then restores the original meaning to |. Any character may be temporarily turned into a literal meaning this way.

package: verbatim

One problem with the verbatim environment is that the entire literal text is input and stored before processing. Sometime that can lead to memory overflows. The verbatim package offers two other extra features. It provides a comment environment that simply ignores its contents, as though each line started with a % sign. And it adds a command \verbatiminput{filename} to input the specified file as literal text.

Email and Internet addresses

Email and Internet addresses present some special problems that are solved with the url package by Donald Arseneau.

Example:

```
An Internet address may be given as
\url{http://address.edu/home/page/}
or an email address might be given as
\url=fred.smith@general.servies.gov=
```

produces

An Internet address may be given as http://address.edu/home/page/ or an email address might be given as fred.smith@general.servies.gov

In fact, the argument of the \url command may be either be enclosed in curly braces as usual or delimited by some arbitrary character, just as with \verb.

The printed address appearance can be controlled by specifying \urlstyle{style}, where style is one of tt (default), rm, sf or same (typewriter font, roman, sans serif, unchanged font).

Footnotes and marginal notes: [Standard footnotes]

Footnotes are generated with the command

\footnote{ footnote_text }

The text 'footnote_text' appears as a footnote in a smaller typeface at the bottom of the page. The first line of the footnote is indented and is given the same footnote marker as that inserted in the main text. The first footnote on a page is seperated from the rest of the page text by means of a short horizontal line.

The standard footnote marker is a small, raised number, which is sequentially numbered.

The footnote numbering is incremented throughout the whole document for the article class, whereas it is reset to '1' for each new chapter in the report and book classes.

Non-Standard footnotes

If the user wishes the footnote numbering to be reset to 1 for each \section command with the article class, this may be acheived with

\setcounter{footnote}{0}

just before or after a \section command.

The internal footnote counter has the name footnote. The \footnote counter is incremented one by one and prints the new value in Arabic numbering as the footnote marker. A different style of marker can be implemented with the command

\renewcommand{\thefootnote}{\number_style{footnote}}

number_style → \arabic, \roman, \Roman, \alph, \Alph

example

\renewcommand{\thefootnote}{\arabic{footnote}}

The another command \fnsymbol, which prints the counter values 1-9 as one of nine symbols:

* † ‡ § ¶ ‖ ** †† #

example

\renewcommand{\thefootnote}{\fnsymbol{footnote}}

An optional argument may be added to the \footnote command

\footnote [num] {footnote_text}

num - the value of the footnote counter for the marker. It is a fixe integel.

Footnotes in minipages

Footnotes produced with the \footnote command inside a minipage environment use the mpfootnote counter and are typeset at the bottom of the parbox produced by the minipage.

However, if you use the \footnotemark command in a minipage it will produce a footnote mark in the same style and sequence as the main text footnotes.

You place a \footnotemark inside the minipage and the corresponding \footnotetext after it.

Example:

```
\begin{minipage}{5cm}
Footnotes in a minipage are numbered using lowercase letters.
\footnote{Inside minipage} \par This text references a footnote
at the bottom of the page. \footnotemark
\end{minipage}
\footnotetext{at the bottom of the page}
```

Produces

Footnotes in a minipage are numbered using lowercase letters[a].
This text references a footnote at the bottom of the page![1]

[a] Inside minipage

[1] at the bottom of the page

Footnotes in tabular material

Footnote appearing inside tabular material are not typeset by standard Latex. Only tabular and Longtable environments will treat footnotes correctly. We can discuss about this enviroment in a minipage.

Example:

```
\begin{minipage}{5cm}
\renewcommand{\thefootnote}{\thempfootnote}
 \begin{tabular}{ll}
  \caption{ Postscript type font} \\
  \multicolumn{2}{c}{\bf Postscript type font} \\
   Courier \footnote{Donated by IBM} & a, b, c \\
   Nimbus \footnote{Donated by URW} & d, e, f \\
   Antiqua \footnotemark[\value{mpfootnote}] & g \\
   Utopia \footnote{Donated by Adobe} & h, i, j
  \end{tabular}
 \end{minipage}
```

Produces

Postscript type font

Postscript type font

Courier[a]	a, b, c
Nimbus[b]	d, e, f
Antiqua[b]	g
Utopia[c]	h, i, j

[a] Donated by IBM
[b] Donated by URW
[c] Donated by Adobe

Marginal notes

\marginpar{left-text}{right-text}

The \marginpar{note_text} command generates a marginal note. This command typesets the text given as an argument in the margin.

When only the mandatory argument right-text is specified, then the text goes to the right margin for one-sided printing; to the outside margin for two-sided printing; and to the nearest margin for two-column formatting (ie) left for left column and right for right column.

Example — Outside margin - right margin for odd pages and left margin for even pages.

This ... The marginal note \marginpar{This \\ is a \\ margin-al note} appearing here...

This is a margin-al note

The text is normally enclosed in a parbox of width 1.9 cm. such a narrow box causes great difficulties with line breaking that is why the lines are broken manually with the \\ command in the above example.

Another common use for the marginal note is to draw a vertical line in the margin to give attention to certain text passages. This is often done to indicate changes in a text, for comparison with earlier versions after updated single sheets have been redistributed. This is done by the command,

\marginpar{\rule[-17.5mm]{1mm}{20mm}}

In fact, the direction of marginal markings arrow turns depends on the page number or column.

'⇐' for odd pages and '⇒' for even pages.

The standard positioning of the marginal notes can be switched with the command \reversemarginpar. Once this command has been given, marginal notes will appear in the left margin (or) in the 'inner margin' for the twosided option. The command \normalmarginpar restores normal behaviour. These commands have no effect with the twocolumn option.

Style parameters for footnotes and marginal notes:

\footnotesep – The vertical spacing between two footnotes

\footnoterule – The command that draws a horizontal line between the page text and the footnotes.

\marginparwidth – Determine the width of the margin box

\marginparsep – Sets the seperation between the margin box and the main text

\marginparpush – Is the smallest vertical distance between two marginal notes.

These parameters are all lengths and are assigned new values as usual with the \setlength command.