LaTeX (0)

Acknowledgments Bibliography Commands

Acknowledgments (1)

This file is a translation of a VMS help file compiled by:
George D. Greenwade <bed_gdg@SHSU.edu>
from a number of sources.

The port to MS Windows Help format is by:
Michael F. Reid <mfreid@hkuxa.hku.hk>
Department of Physics, University of Hong Kong.
From April 1993:
Michael F. Reid <mfr@phys.canterbury.ac.nz>
Department of Physics and Astronomy, University of Canterbury
Christchurch, New Zealand.

January 13, 1993.

Bibliography (2)

The LaTeX command typesets a file of text using the TeX program and the LaTeX Macro package for TeX. To be more specific, it processes an input file containing the text of a document with interspersed commands that describe how the text should be formatted. It produces two files as output, a Device Independent (DVI) file that contains commands that can be translated into commands for a variety of output devices, and a `transcript' or `log file' that contains summary information and diagnostic messages for any errors discovered in the input file.

For a description of what goes on inside TeX, you should consult The TeXbook by Donald E. Knuth, ISBN 0-201-13448-9, published jointly by the American Mathematical Society and Addison-Wesley Publishing Company.

For a description of LaTeX, you should consult "A Document Preparation System: LaTeX" by Leslie Lamport, ISBN 0-201-15790-X, published jointly by the American Mathematical Society and Addison-Wesley Publishing Company.

Commands (3)

A LaTeX command begins with the command name, which consists of a \ followed by either (a) a string of letters or (b) a single non-letter. Arguments contained in square brackets [] are optional while arguments contained in braces {} are required.

NOTE: LaTeX is case sensitive. Enter all commands in lower case unless explicitly directed to do otherwise.

Counters Cross_References Definitions Document_Styles Environments Footnotes Lengths Letters Line_and_Page_Breaking Making_Paragraphs Math_Formulas Modes Page_Styles Sectioning Spaces_and_Boxes Special_Characters Splitting_the_Input Starting_and_Ending Table_of_Contents Terminal_Input_and_Output Typefaces _{exp} (subscript) ^{exp} (superscript) \\ \- \; \: \, \! \= \> \< \+ \- \' \` \addcontentsline \addtocontents \addtocounter \address \addtolength \addvspace \alph \appendix \arabic array \author \bf \bibitem \bigskip \cdots center \centering \circle \cite \cleardoublepage \clearpage \cline \closing \dashbox \date \ddots description \dotfill \em enumerate eqnarray equation figure \fbox \flushbottom flushleft flushright \fnsymbol \footnote \footnotemark \footnotesize \footnotetext \frac \frame \framebox \hfill \hline \hrulefill \hspace \huge \Huge (capital "h") \hyphenation \include \includeonly \indent \input \it itemize \kill \label \large \Large (capital "l") \LARGE (all caps) \ldots \line \linebreak \linethickness list \location \makebox \maketitle \mark \mbox \medskip minipage \multicolumn \multiput \newcommand \newcounter \newenvironment \newfont \newlength \newline \newpage \newsavebox \newtheorem \nocite \noindent \nolinebreak \normalsize (default) \nopagebreak \onecolumn \opening \oval \overbrace \overline \pagebreak \pagenumbering \pageref \pagestyle \par \parbox picture \put quotation quote \raggedbottom \raggedleft \raggedright \raisebox \ref \rm \roman \rule \savebox \sc \scriptsize \setcounter \setlength \settowidth \sf \shortstack \signature \sl \small \smallskip \sqrt tabbing table tabular \telephone \thanks thebibliography theorem \thispagestyle \tiny \title titlepage \tt \twocolumn \typeout \typein \underbrace \underline \usebox \usecounter \value \vdots \vector \verb verbatim verse \vfill \vline \vspace

Counters (4)

Everything LaTeX numbers for you has a counter associated with it. The name of the counter is the same as the name of the environment or command that produces the number, except with no \. Below is a list of the counters used LaTeX's standard document styles to control numbering.

part chapter section subsection subsubsection paragraph subparagraph
page equation figure table footnote mpfootnote
enumi enumii enumiii enumiv

\addtocounter \alph \arabic \fnsymbol \newcounter \roman \setcounter \usecounter \value

\addtocounter (5)

```
\addtocounter{counter}{value}
```

The \addtocounter command increments the counter by the amount specified by the value argument. The value argument can be negative.

\alph (6)

\alph{counter}

This command causes the value of the counter to be printed in alphabetic characters. The \alph command causes lower case alphabetic alphabetic characters, i.e., a, b, c... while the \Alph command causes upper case alphabetic characters, i.e., A, B, C...

\arabic (7)

\arabic{counter}

The \arabic command causes the value of the counter   to be printed   in arabic numbers, i.e., 3.

\fnsymbol (8)

```
\fnsymbol{counter}
```

The \fnsymbol command causes the value of the counter to be printed in a specific sequence of nine symbols that can be used for numbering footnotes.

\newcounter (9)

```
\newcounter{foo}[counter]
```

The \newcounter command defines a new counter named foo. The optional argument [counter] causes the counter foo to be reset whenever the counter named in the optional argument is incremented.

\roman (10)

```
\roman{counter}
```

This command causes the value of the counter to be printed in roman numerals. The \roman command causes lower case roman numerals, i.e., i, ii, iii..., while the \Roman command causes upper case roman numerals, i.e., I, II, III...

\setcounter (11)

```
\setcounter{counter}{value}
```

The \setcounter command sets the value of the counter to that specified by the value argument.

\usecounter (12)

```
\usecounter{counter}
```

The \usecounter command is used in the second argument of the list environment to allow the counter specified to be used to number the list items.

\value (13)

```
\value{counter}
```

The \value command produces the value of the counter named in the mandatory argument. It can be used where LaTeX expects an integer or number, such as the second argument of a \setcounter or \addtocounter command, or in

```
        \hspace{\value{foo}\parindent}
```

It is useful for doing arithmetic with counters.

Cross_References (14)

One reason for numbering things like figures and equations is to refer the reader to them, as in "See Figure 3 for more details."

\label \pageref \ref

\label (15)

```
\label{key}
```

A \label command appearing in ordinary text assigns to the key the number of the current sectional unit; one appearing inside a numbered environment assigns that number to the key.

A key can consist of any sequence of letters, digits, or punctuation characters. Upper- and lowercase letters are different.

\pageref (16)

```
\pageref{key}
```

The \pageref command produces the page number of the place in the text where the corresponding \label command appears.

\ref (17)

```
\ref{key}
```

The \ref command produces the number of the sectional unit, equation number, ... of the corresponding \label command.

Definitions (18)

\newcommand	\newenvironment	\newtheorem	\newfont

\newcommand (19)

```
\newcommand{cmd}[args]{def}
\renewcommand{cmd}[args]{def}
```

These commands define (or redefine) a command.

- cmd: A command name beginning with a \. For \newcommand it must not be already defined and must not begin with \end; for \renewcommand it must already be defined.

- args: An integer from 1 to 9 denoting the number of arguments of the command being defined. The default is for the command to have no arguments.

- def: The text to be substituted for every occurrence of cmd; a parameter of the form #n in cmd is replaced by the text of the nth argument when this substitution takes place.

\newenvironment (20)

\newenvironment{nam}[args]{begdef}{enddef}
\renewenvironment{nam}[args]{begdef}{enddef}

These commands define or redefine an environment.

- nam: The name of the environment. For \newenvironment there must be no currently defined environment by that name, and the command \nam must be undefined. For \renewenvironment the environment must already be defined.

- args: An integer from 1 to 9 denoting the number of arguments of the newly-defined environment. The default is no arguments.

- begdef: The text substituted for every occurrence of \begin{name}; a parameter of the form #n in cmd is replaced by the text of the nth argument when this substitution takes place.

- enddef: The text substituted for every occurrence of \end{nam}. It may not contain any argument parameters.

\newtheorem (21)

```
\newtheorem{env_name}{caption}[within]
\newtheorem{env_name}[numbered_like]{caption}
```

This command defines a theorem-like environment.

- env_name: The name of the environment -- a string of letters. Must not be the name of an existing environment or counter.

- caption: The text printed at the beginning of the environment, right before the number.

- within: The name of an already defined counter, usually of a sectional unit. Provides a means of resetting the new theorem counter within the sectional unit.

- numbered_like: The name of an already defined theorem-like environment.

The \newtheorem command may have at most one optional argument.

\newfont (22)

```
\newfont{cmd}{font_name}
```

Defines the command name cmd, which must not be currently defined, to be a declaration that selects the font named font_name to be the current font.

Document_Styles (23)

Valid LaTeX document styles include:

- o article
- o report
- o letter
- o book

Other document styles are described under the Help Topic LaTeX_Styles.

They are selected with the following command:

```
\documentstyle [options] {style}
```

The options for the different styles are:

1. article: 11pt, 12pt, twoside, twocolumn, draft, fleqn, leqno, acm
2. report: 11pt, 12pt, twoside, twocolumn, draft, fleqn, leqno, acm
3. letter: 11pt, 12pt, fleqn, leqno, acm
4. book: 11pt, 12pt, twoside,twocolumn, draft, fleqn, leqno

If you specify more than one option, they must be separated by a comma.

\flushbottom \onecolumn \raggedbottom \twocolumn

\flushbottom (24)

The \flushbottom declaration makes all text pages the same height, adding extra vertical space when necessary to fill out the page.

\onecolumn (25)

The \onecolumn declaration starts a new page and produces single-column output.

\raggedbottom (26)

The \raggedbottom declaration makes all pages the height of the text on that page. No extra vertical space is added.

\twocolumn (27)

The \twocolumn declaration starts a new page and produces two-column output.

Environments (28)

LaTeX provides a number of different paragraph-making environments. Each environment begins and ends in the same manner.

```
\begin{environment-name}
.
.
.
\end{environment-name}
```

array center description enumerate eqnarray equation figure
flushleft flushright itemize list minipage picture
quotation quote tabbing table tabular thebibliography
theorem titlepage verbatim verse

array (29)

```
\begin{array}{col1col2...coln}
column 1 entry & column 2 entry ... & column n entry \\
.
.
.
\end{array}
```

Math arrays are produced with the array environment. It has a single mandatory argument describing the number of columns and the alignment within them. Each column, coln, is specified by a single letter that tells how items in that row should be formatted.

- c for centered
- l for flushleft
- r for flushright

Column entries must be separated by an &. Column entries may include other LaTeX commands. Each row of the array must be terminated with the string \\.

center (30)

```
\begin{center}
Text on line 1 \\
Text on line 2 \\
.
.
.
\end{center}
```

The center environment allows you to create a paragraph consisting of lines that are centered within the left and right margins on the current page. Each line must be terminated with the string \\.

\centering

\centering (31)

This declaration corresponds to the center environment. This declaration can be used inside an environment such as quote or in a parbox. The text of a figure or table can be centered on the page by putting a \centering command at the beginning of the figure or table environment.

Unlike the center environment, the \centering command does not start a new paragraph; it simply changes how LaTeX formats paragraph units. To affect a paragraph unit's format, the scope of the declaration must contain the blank line or \end command (of an environment like quote) that ends the paragraph unit.

description (32)

```
\begin{description}
\item [label] First item
\item [label] Second item
.
.
.
\end{description}
```

The description environment is used to make labeled lists. The label is bold face and flushed right.

enumerate (33)

```
\begin{enumerate}
\item First item
\item Second item
.
.
.
\end{enumerate}
```

The enumerate environment produces a numbered list. Enumerations can be nested within one another, up to four levels deep. They can also be nested within other paragraph-making environments.

Each item of an enumerated list begins with an \item command. There must be at least one \item command within the environment.

eqnarray (34)

```
\begin{eqnarray}
math formula 1 \\
math formula 2 \\
.
.
.
\end{eqnarray}
```

The eqnarray environment is used to display a sequence of equations or inequalities. It is very much like a three-column array environment, with consecutive rows separated by \\ and consecutive items within a row separated by an &. An equation number is placed on every line unless that line has a \nonumber command.

equation (35)

```
\begin{equation}
  math formula
\end{equation}
```

The equation environment   centers your equation on the page and places the equation number in the right margin.

figure (36)

```
\begin{figure}[placement]

  body of the figure

\caption{figure title}
\end{figure}
```

Figures are objects that are not part of the normal text, and are usually "floated" to a convenient place, like the top of a page. Figures will not be split between two pages.

The optional argument [placement] determines where LaTeX will try to place your figure. There are four places where LaTeX can possibly put a float:

- h: Here - at the position in the text where the figure environment appears.

- t: Top - at the top of a text page.

- b: Bottom - at the bottom of a text page.

- p: Page of floats - on a separate float page, which is a page containing no text, only floats.

The standard report and article styles use the default placement tbp.

The body of the figure is made up of whatever text, LaTeX commands, etc. you wish. The \caption command allows you to title your figure.

flushleft (37)

```
\begin{flushleft}
Text on line 1 \\
Text on line 2 \\
.
.
.
\end{flushleft}
```

The flushleft environment allows you to create a paragraph consisting of lines that are flushed left to the left-hand margin. Each line must be terminated with the string \\.

\raggedright

\raggedright (38)

This declaration corresponds to the flushleft environment. This declaration can be used inside an environment such as quote or in a parbox.

Unlike the flushleft environment, the \raggedright command does not start a new paragraph; it simply changes how LaTeX formats paragraph units. To affect a paragraph unit's format, the scope of the declaration must contain the blank line or \end command (of an environment like quote) that ends the paragraph unit.

flushright (39)

```
\begin{flushright}
Text on line 1 \\
Text on line 2 \\
.
.
.
\end{flushright}
```

The flushright environment allows you to create a paragraph consisting of lines that are flushed right to the right-hand margin. Each line must be terminated with the string \\.

\raggedleft

\raggedleft (40)

This declaration corresponds to the flushright environment. This declaration can be used inside an environment such as quote or in a parbox.

Unlike the flushright environment, the \raggedleft command does not start a new paragraph; it simply changes how LaTeX formats paragraph units. To affect a paragraph unit's format, the scope of the declaration must contain the blank line or \end command (of an environment like quote) that ends the paragraph unit.

itemize (41)

```
\begin{itemize}
\item First item
\item Second item
.
.
.
\end{itemize}
```

The itemize environment produces a bulleted list. Itemizations can be nested within one another, up to four levels deep. They can also be nested within other paragraph-making environments.

Each item of an itemized list begins with an \item command. There must be at least one \item command within the environment.

list (42)

```
\begin{list}{label}{spacing}
\item First item
\item Second item
.
.
.
\end{list}
```

The {label} argument specifies how items should be labeled. This argument is a piece of text that is inserted in a box to form the label. This argument can and usually does contain other LaTeX commands.

The {spacing} argument contains commands to change the spacing parameters for the list. This argument will most often be null, i.e., {}. This will select all default spacing which should suffice for most cases.

minipage (43)

```
\begin{minipage}[position]{width}
  text
\end{minipage}
```

The minipage environment is similar to a \parbox command. It takes the same optional position argument and mandatory width argument. You may use other paragraph-making environments inside a minipage.

Footnotes in a minipage environment are handled in a way that is particularly useful for putting footnotes in figures or tables. A \footnote or \footnotetext command puts the footnote at the bottom of the minipage instead of at the bottom of the page, and it uses the mpfootnote counter instead of the ordinary footnote counter.

NOTE: Don't put one minipage inside another if you are using footnotes; they may wind up at the bottom of the wrong minipage.

picture (44)

```
\begin{picture}(width,height)(x offset,y offset)
.
  picture commands
.
\end{picture}
```

The picture environment allows you to create just about any kind of picture you want containing text, lines, arrows and circles. You tell LaTeX where to put things in the picture by specifying their coordinates. A coordinate is a number that may have a decimal point and a minus sign - a number like 5, 2.3 or -3.1416. A coordinate specifies a length in multiples of the unit length \unitlength, so if \unitlength has been set to 1cm, then the coordinate 2.54 specifies a length of 2.54 centimeters. You can change the value of \unitlength anywhere you want, using the \setlength command, but strange things will happen if you try changing it inside the picture environment.

A position is a pair of coordinates, such as (2.4,-5), specifying the point with x-coordinate 2.4 and y-coordinate -5. Coordinates are specified in the usual way with respect to an origin, which is normally at the lower-left corner of the picture. Note that when a position appears as an argument, it is not enclosed in braces; the parentheses serve to delimit the argument.

The picture environment has one mandatory argument, which is a position. It specifies the size of the picture. The environment produces a rectangular box with width and height determined by this argument's x- and y-coordinates.

The picture environment also has an optional position argument, following the size argument, that can change the origin. (Unlike ordinary optional arguments, this argument is not contained in square brackets.) The optional argument gives the coordinates of the point at the lower-left corner of the picture (thereby determining the origin). For example, if \unitlength has been set to 1mm, the command

```
\begin{picture}(100,200)(10,20)
```

produces a picture of width 100 millimeters and height 200 millimeters, whose lower-left corner is the point (10,20) and whose upper-right corner is therefore the point (110,220). When you first draw a picture, you will omit the optional argument, leaving the origin at the lower-left corner. If you then want to modify your picture by shifting everything, you just add the appropriate optional argument.

The environment's mandatory argument determines the nominal size of the picture. This need bear no relation to how large the picture really is; LaTeX will happily allow you to put things outside the picture, or even off the page. The picture's nominal size is used by TeX in determining how much room to leave for it.

Everything that appears in a picture is drawn by the \put command. The command

```
\put (11.3,-.3){...}
```

puts the object specified by "..." in the picture, with its reference point at coordinates (11.3,-.3). The reference points for various objects will be described below.

The \put command creates an LR box. You can put anything in the text argument of the \put command that you'd put into the argument of an \mbox and related commands. When you do this, the reference point will be the lower left corner of the box.

\circle \dashbox \frame \framebox \line \linethickness \makebox \multiput \oval \put \shortstack \vector

\circle (45)

```
\circle[*]{diameter}
```

The \circle command produces a circle of the specified diameter. If the *-form of the command is used, LaTeX draws a solid circle.

\dashbox (46)

```
\dashbox{dash length}(width,height){...}
```

The \dashbox has an extra argument which specifies the width of each dash. A dashed box looks best when the width and height are multiples of the dash length.

\frame (47)

```
\frame{...}
```

The \frame command puts a rectangular frame around the object specified in the argument. The reference point is the bottom left corner of the frame. No extra space is put between the frame and the object.

\framebox (48)

```
\framebox(width,height)[position]{...}
```

The \framebox command is analogous to the \makebox command.

\line (49)

```
\line(x slope,y slope){length}
```

The \line command draws a line of the specified length and slope.

\linethickness (50)

```
\linethickness{dimension}
```

Declares the thickness of horizontal and vertical lines in a picture environment to be dimension, which must be a positive length. It does not affect the thickness of slanted lines and circles, or the quarter circles drawn by \oval to form the corners of an oval.

\makebox (51)

\makebox(width,height)[position]{...}

The \makebox command for the picture environment is similar to the normal \makebox command except that you must specify a width and height in multiples of \unitlength.

The optional argument, [position], specifies the quadrant that your text appears in. You may select up to two of the following:

- t: Moves the item to the top of the rectangle
- b: Moves the item to the bottom
- l: Moves the item to the left
- r: Moves the item to the right

\multiput (52)

```
\multiput(x coord,y coord)(delta x,delta y){number of copies}{object}
```

The \multiput command can be used when you are putting the same object in a regular pattern across a picture.

\oval (53)

\oval(width,height)[portion]

The \oval command produces a rectangle with rounded corners. The optional argument, [portion], allows you to select part of the oval.

- t: Selects the top portion
- b: Selects the bottom portion
- r: Selects the right portion
- l: Selects the left portion

\put (54)

```
\put(x coord,y coord){ ...   }
```

The \put command places the item specified by the mandatory argument at the given coordinates.

\shortstack (55)

\shortstack[position]{... \\ ... \\ ...}

The \shortstack command produces a stack of objects. The valid positions are:

- r: Moves the objects to the right of the stack
- l: Moves the objects to the left of the stack
- c: Moves the objects to the center of the stack (default)

\vector (56)

```
\vector(x slope,y slope){length}
```

The \vector command draws a line with an arrow of the specified length and slope. The x and y values must lie between -4 and +4, inclusive.

quotation (57)

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

The margins of the quotation environment are indented on the left and the right. The text is justified at both margins and there is paragraph indentation. Leaving a blank line between text produces a new paragraph.

quote (58)

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

The margins of the quote environment are indented on the left and the right. The text is justified at both margins. Leaving a blank line between text produces a new paragraph.

tabbing (59)

```
\begin{tabbing}
text \= more text \= still more text \= last text \\
second row \>   \> more \\
.
.
.
\end{tabbing}
```

The tabbing environment provides a way to align text in columns. It works by setting tab stops and tabbing to them much the way you do with an ordinary typewriter.

\=   \>   \<   \+   \-   \'   \`   \kill

\= (60)

The \= command sets the tab stops.

\> (61)

The \> command causes LaTeX to advance to the next tab stop.

\< (62)

The \< command   allows you to put something   to the left of the local margin without changing the margin.

\+ (63)

The \+ command moves the left margin of the next and all the following commands one tab stop to the right.

\- (64)

The \- command moves the left margin of the next and all the following commands one tab stop to the left.

\' (65)

The \' command moves everything that you have typed so far in the current column , everything starting from the most recent \>, \<, \', \\, or \kill command, to the right of the previous column, flush against the current column's tab stop.

\` (66)

The \` command allows you to put text flushed right against any tab stop, including tab stop 0. However, it can't move text to the right of the last column because there's no tab stop there. The \` command moves all the text that follows it, up to the \\ or \end{tabbing} command that ends the line, to the right margin of the tabbing environment. There must be no \> or \' command between the \` and the command that ends the line.

\kill (67)

The \kill command allows you to set tab stops without producing text. It works just like the \\ except that it throws away the current line instead of producing output for it. The effect of any \=, \+ or \- commands in that line remain in effect.

table (68)

```
\begin{table}[placement]

  body of the table

\caption{table title}
\end{table}
```

Tables are objects that are not part of the normal text, and are usually "floated" to a convenient place, like the top of a page. Tables will not be split between two pages.

The optional argument [placement] determines where LaTeX will try to place your table. There are four places where LaTeX can possibly put a float:

- h: Here - at the position in the text where the table environment appears.
- t: Top - at the top of a text page.
- b: Bottom - at the bottom of a text page.
- p: Page of floats - on a separate float page, which is a page containing no text, only floats.

The standard report and article styles use the default placement tbp.

The body of the table is made up of whatever text, LaTeX commands, etc., you wish. The \caption command allows you to title your table.

tabular (69)

```
\begin{tabular}[pos]{cols}
column 1 entry & column 2 entry ... & column n entry \\
.
.
.
\end{tabular}

                    or

\begin{tabular*}{width}[pos]{cols}
column 1 entry & column 2 entry ... & column n entry \\
.
.
.
\end{tabular*}
```

These environments produce a box consisting of a sequence of rows of items, aligned vertically in columns. The mandatory and optional arguments consist of:

- o width: Specifies the width of the tabular* environment. There must be rubber space between columns that can stretch to fill out the specified width.

- o pos: Specifies the vertical position; default is alignment on the center of the environment.

  - t - align on top row

  - b - align on bottom row

- o cols: Specifies the column formatting. It consists of a sequence of the following specifiers, corresponding to the sequence of columns and intercolumn material.

  - l - A column of left-aligned items.

  - r - A column of right-aligned items.

  - c - A column of centered items.

  - | - A vertical line the full height and depth of the environment.

  - @{text} - This inserts text in every row. An @-expression suppresses the intercolumn space normally inserted between columns; any desired space between the inserted text and the adjacent items must be included in text. An \extracolsep{wd} command in an @-expression causes an extra space of width wd to appear to the left of all subsequent columns, until countermanded by another \extracolsep command. Unlike ordinary intercolumn space, this extra space is not suppressed by an @-expression. An \extracolsep command can be used only in an @-expression in the cols argument.

- p{wd} - Produces a column with each item typeset in a parbox of width wd, as if it were the argument of a \parbox[t]{wd} command. However, a \\ may not appear in the item, except in the following situations: (i) inside an environment like minipage, array, or tabular, (ii) inside an explicit \parbox, or (iii) in the scope of a \centering, \raggedright, or \raggedleft declaration. The latter declarations must appear inside braces or an environment when used in a p-column element.

- *{num}{cols} - Equivalent to num copies of cols, where num is any positive integer and cols is any list of column-specifiers, which may contain another *-expression.

\cline \hline \multicolumn \vline

\cline (70)

```
\cline{i-j}
```

The \cline command draws horizontal lines across the columns specified, beginning in column i and ending in column j, which are identified in the mandatory argument.

\hline (71)

The \hline command will draw a horizontal line the width of the table. It's most commonly used to draw a line at the top, bottom, and between the rows of the table.

\multicolumn (72)

```
\multicolumn{cols}{pos}{text}
```

The \multicolumn is used to make an entry that spans several columns. The first mandatory argument, cols, specifies the number of columns to span. The second mandatory argument, pos, specifies the formatting of the entry; c for centered, l for flushleft, r for flushright. The third mandatory argument, text, specifies what text is to make up the entry.

\vline (73)

The \vline command will draw a vertical line extending the full height and depth of its row. An \hfill command can be used to move the line to the edge of the column. It can also be used in an @-expression.

thebibliography (74)

```
\begin{thebibliography}{widest-label}
\bibitem[label]{cite_key}
.
.
.
\end{thebibliography}
```

The thebibliography environment produces a bibliography or reference list. In the article style, this reference list is labeled "References"; in the report style, it is labeled "Bibliography".

- o widest-label: Text that, when printed, is approximately as wide as the widest item label produces by the \bibitem commands.

\bibitem \cite \nocite

\bibitem (75)

```
\bibitem[label]{cite_key}
```

The \bibitem command generates an entry labeled by label. If the label argument is missing, a number is generated as the label, using the enumi counter. The cite_key is any sequence of letters, numbers, and punctuation symbols not containing a comma. This command writes an entry on the aux file containing cite_key and the item's label. When this aux file is read by the \begin{document} command, the item's label is associated with cite_key, causing the reference to cite_key by a \cite command to produce the associated label.

\cite (76)

```
\cite[text]{key_list}
```

The key_list argument is a list of citation keys. This command generates an in-text citation to the references associated with the keys in key_list by entries on the aux file read by the \begin{document} command.

\nocite (77)

```
\nocite{key_list}
```

The \nocite command produces no text, but writes key_list, which is a list of one or more citation keys, on the aux file.

theorem (78)

```
\begin{theorem}
  theorem text
\end{theorem}
```

The theorem environment produces "Theorem x" in boldface followed by your theorem text.

titlepage (79)

```
\begin{titlepage}
  text
\end{titlepage}
```

The titlepage environment creates a title page, i.e. a page with no printed page number or heading. It also causes the following page to be numbered page one. Formatting the title page is left to you. The \today command comes in handy for title pages.

verbatim (80)

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

The verbatim environment is a paragraph-making environment that gets LaTeX to print exactly what you type in. It turns LaTeX into a typewriter with carriage returns and blanks having the same effect that they would on a typewriter.

\verb

\verb (81)

\verb char literal_text char \verb*char literal_text char

Typesets literal_text exactly as typed, including special characters and spaces, using a typewriter (\tt) type style. There may be no space between \verb or \verb* and char (space is shown here only for clarity). The *-form differs only in that spaces are printed.

verse (82)

```
\begin{verse}
  text
\end{verse}
```

The verse environment is designed for poetry, though you may find other uses for it.

Footnotes (83)

Footnotes can be produced in one of two ways. They can be produced with one command, the \footnote command. They can also be produced with two commands, the \footnotemark and the \footnotetext commands. See the specific command for information on why you would use one over the other.

\footnote \footnotemark \footnotetext

\footnote (84)

```
\footnote[number]{text}
```

The \footnote command places the numbered footnote text at the bottom of the current page. The optional argument, number, is used to change the default footnote number. This command can only be used in outer paragraph mode.

\footnotemark (85)

The \footnotemark command puts the footnote number in the text. This command can be used in inner paragraph mode. The text of the footnote is supplied by the \footnotetext command.

\footnotetext (86)

```
\footnotetext[number]{text}
```

The \footnotetext command produces the text to be placed at the bottom of the page. This command can come anywhere after the \footnotemark command. The \footnotetext command must appear in outer paragraph mode.

The optional argument, number, is used to change the default footnote number.

Lengths (87)

A length is a measure of distance. Many LaTeX commands take a length as an argument.

\newlength \setlength \addtolength \settowidth

\newlength (88)

```
\newlength{\gnat}
```

The \newlength command defines the mandatory argument, \gnat, as a length command with a value of 0in. An error occurs if a \gnat command already exists.

\setlength (89)

```
\setlength{\gnat}{length}
```

The \setlength command is used to set the value of a length command. The length argument can be expressed in any terms of length LaTeX understands, i.e., inches (in), millimeters (mm), points (pt), etc.

\addtolength (90)

```
\addtolength{\gnat}{length}
```

The \addtolength command increments a length command by the amount specified in the length argument. It can be a negative amount.

\settowidth (91)

```
\settowidth{\gnat}{text}
```

The \settowidth command sets the value of a length command equal to the width of the text argument.

Letters (92)

You can use LaTeX to typeset letters, both personal and business. The letter document style is designed to make a number of letters at once, although you can make just one if you so desire.

Your .TEX source file has the same minimum commands as the other document styles, i.e., you must have the following commands as a minimum:

```
\documentstyle{letter}
\begin{document}
  ... letters ...
\end{document}
```

Each letter is a letter environment, whose argument is the name and address of the recipient. For example, you might have

```
\begin{letter}{Mr. John Doe \\ 2345 Jones St.
       \\ Oakland, CA   91123}
   ...
\end{letter}
```

The letter itself begins with the \opening command. The text of the letter follows. It is typed as ordinary LaTeX input. Commands that make no sense in a letter, like \chapter, don't work. The letter closes with a \closing command.

After the closing, you can have additional material. The \cc command produces the usual "cc: ...". There's also a similar \encl command for a list of enclosures.

Declarations \opening \closing

Declarations (93)

The following commands are declarations which take a single argument.

\address \signature \location \telephone

\address (94)

```
\address{Return address}
```

The return address, as it should appear on the letter and the envelope. Separate lines of the address should be separated by \\ commands. If you do not make an \address declaration, then the letter will be formatted for copying onto your organization's standard letterhead. If you give an \address declaration, then the letter will be formatted as a personal letter.

\signature (95)

```
\signature{Your name}
```

Your name, as it should appear at the end of the letter underneath the space for your signature. Items that should go on separate lines should be separated by \\ commands.

\location (96)

```
\location{address}
```

This modifies your organization's standard address. This only appears if the firstpage pagestyle is selected.

\telephone (97)

```
\telephone{number}
```

This is your telephone number. This only appears if the firstpage pagestyle is selected.

\opening (98)

```
\opening{text}
```

The letter begins with the \opening command. The mandatory argument, text, is what ever text you wish to start your letter, i.e.,

```
\opening{Dear John,}
```

\closing (99)

```
\closing {text}
```

The letter closes with a \closing command, i.e.,

```
\closing{Best Regards,}
```

Line_and_Page_Breaking (100)

The first thing LaTeX does when processing ordinary text is to translate your input file into a string of glyphs and spaces. To produce a printed document, this string must be broken into lines, and these lines must be broken into pages. In some environments, you do the line breaking yourself with the \\ command, but LaTeX usually does it for you.

\\ \- \cleardoublepage \clearpage \hyphenation \linebreak \newline \newpage \nolinebreak \nopagebreak \pagebreak

\\ (101)

```
\\[*][extra-space]
```

The \\ command tells LaTeX to start a new line. It has an optional argument, extra-space, that specifies how much extra vertical space is to be inserted before the next line. This can be a negative amount.

The \\* command is the same as the ordinary \\ command except that it tells LaTeX not to start a new page after the line.

\- (102)

The \- command tells LaTeX that it may hyphenate the word at that point. LaTeX is very good at hyphenating, and it will usually find all correct hyphenation points. The \- command is used for the exceptional cases.

\cleardoublepage (103)

The \cleardoublepage command ends the current page and causes all figures and tables that have so far appeared in the input to be printed. In a two-sided printing style, it also makes the next page a right-hand (odd-numbered) page, producing a blank page if necessary.

\clearpage (104)

The \clearpage command ends the current page and causes all figures and tables that have so far appeared in the input to be printed.

\hyphenation (105)

```
\hyphenation{words}
```

The \hyphenation command declares allowed hyphenation points, where words is a list of words, separated by spaces, in which each hyphenation point is indicated by a - character.

\linebreak (106)

\linebreak[number]

The \linebreak command tells LaTeX to break the current line at the point of the command. With the optional argument, number, you can convert the \linebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

The \linebreak command causes LaTeX to stretch the line so it extends to the right margin.

\newline (107)

The \newline command breaks the line right where it is. The \newline command can be used only in paragraph mode.

\newpage (108)

The \newpage command ends the current page.

\nolinebreak (109)

```
\nolinebreak[number]
```

The \nolinebreak command prevents LaTeX from breaking the current line at the point of the command. With the optional argument, number, you can convert the \nolinebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

\nopagebreak (110)

```
\nopagebreak[number]
```

The \nopagebreak command prevents LaTeX form breaking the current page at the point of the command. With the optional argument, number, you can convert the \nopagebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

\pagebreak (111)

\pagebreak[number]

The \pagebreak command tells LaTeX to break the current page at the point of the command. With the optional argument, number, you can convert the \pagebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

Making_Paragraphs (112)

A paragraph is ended by one or more completely blank lines -- lines not containing even an %. A blank line should not appear where a new paragraph cannot be started, such as in math mode or in the argument of a sectioning command.

\indent \noindent \par

\indent (113)

This produces a horizontal space whose width equals the width of the paragraph indentation. It is used to add paragraph indentation where it would otherwise be suppressed.

\noindent (114)

When used at the beginning of the paragraph, it suppresses the paragraph indentation. It has no effect when used in the middle of a paragraph.

\par (115)

Equivalent to a blank line; often used to make command or environment definitions easier to read.

Math_Formulas (116)

There are three environments that put LaTeX in math mode: math, displaymath, and equation. The math environment is for formulas that appear right in the text. The displaymath environment is for formulas that appear on their own line. The equation environment is the same as the displaymath environment except that it adds an equation number in the right margin.

The math environment can be used in both paragraph and LR mode, but the displaymath and equation environments can be used only in paragraph mode. The math and displaymath environments are used so often that they have the following short forms:

```
\(...\) instead of \begin{math}...\end{math}

\[...\] instead of \begin{displaymath}...\end{displaymath}
```

In fact, the math environment is so common that it has an even shorter form:

```
$ ... $ instead of \(...\)
```

Subscripts_and_Superscripts Math_Symbols Spacing_in_Math_Mode Math_Miscellany

Subscripts_and_Superscripts (117)

To get an expression exp to appear as a subscript, you just type _{exp}. To get exp to appear as a superscript, you type ^{exp}. LaTeX handles superscripted superscripts and all of that stuff in the natural way. It even does the right thing when something has both a subscript and a superscript.

Math_Symbols (118)

TeX provides almost any mathematical symbol you're likely to need. The commands for generating them can be used only in math mode. For example, if you include π in your source, you will get the symbol "pi" in your output.

Spacing_in_Math_Mode (119)

In a math environment, LaTeX ignores the spaces you type and puts in the spacing that it thinks is best. LaTeX formats mathematics the way it's done in mathematics texts. If you want different spacing, LaTeX provides the following four commands for use in math mode:

1. \; - a thick space

2. \: - a medium space

3. \, - a thin space

4. \! - a negative thin space

Math_Miscellany (120)

\cdots \ddots \frac \ldots \overbrace \overline \sqrt \underbrace \underline \vdots

\cdots (121)

The \cdots command produces a horizontal   ellipsis where the dots are raised to the center of the line.

\ddots (122)

The \ddots command produces a diagonal ellipsis.

\frac (123)

```
\frac{num}{den}
```

The \frac command produces the fraction num divided by den.

\ldots (124)

The \ldots command produces an ellipsis. This command works in any mode, not just math mode.

\overbrace (125)

\overbrace{text}

The \overbrace command generates a brace over text.

\overline (126)

```
\overline{text}
```

The \overline command causes the argument text to be overlined.

\sqrt (127)

```
\sqrt[root]{arg}
```

The \sqrt command produces the square root of its argument. The optional argument, root, determines what root to produce, i.e., the cube root of x+y would be typed as `$\sqrt[3]{x+y}$`.

\underbrace (128)

```
\underbrace{text}
```

The \underbrace command generates text with a brace underneath.

\underline (129)

\underline{text}

The \underline   command   causes   the argument   text to be underlined.
This command can also be used in paragraph and LR modes.

\vdots (130)

The \vdots command produces a vertical ellipsis.

Modes (131)

When LaTeX is processing your input text, it is always in one of three modes:

- o Paragraph mode
- o Math mode
- o Left-to-right mode, called LR mode for short

LaTeX changes mode only when it goes up or down a staircase to a different level, though not all level changes produce mode changes. Mode changes occur only when entering or leaving an environment, or when LaTeX is processing the argument of certain text-producing commands.

Paragraph mode is the most common; it's the one LaTeX is in when processing ordinary text. In that mode, LaTeX breaks your text into lines and breaks the lines into pages. LaTeX is in math mode when it's generating a mathematical formula. In LR mode, as in paragraph mode, LaTeX considers the output that it produces to be a string of words with spaces between them. However, unlike paragraph mode, LaTeX keeps going from left to right; it never starts a new line in LR mode. Even if you put a hundred words into an \mbox, LaTeX would keep typesetting them from left to right inside a single box, and then complain because the resulting box was too wide to fit on the line.

LaTeX is in LR mode when it starts making a box with an \mbox command. You can get it to enter a different mode inside the box - for example, you can make it enter math mode to put a formula in the box. There are also several text-producing commands and environments for making a box that put LaTeX in paragraph mode. The box make by one of these commands or environments will be called a parbox. When LaTeX is in paragraph mode while making a box, it is said to be in inner paragraph mode. Its normal paragraph mode, which it starts out in, is called outer paragraph mode.

Page_Styles (132)

The \documentstyle command determines the size and position of the page's head and foot. The page style determines what goes in them.

\maketitle \pagenumbering \pagestyle \thispagestyle

\maketitle (133)

```
\maketitle
```

The \maketitle command generates a title on a separate title page - except in the article style, where the title normally goes at the top of the first page. Information used to produce the title is obtained from the following declarations.

\author \date \thanks \title

\author (134)

```
\author{names}
```

The \author command declares the author(s), where names is a list of authors separated by \and commands. Use \\ to separate lines within a single author's entry -- for example, to give the author's institution or address.

NOTE: The milstd and book-form styles have re-defined the \maketitle command. The \title declaration is the only command of those shown below that has any meaning.

\date (135)

```
\date{text}
```

The \date command declares text to be the document's date. With no \date command, the current date is used.

\thanks (136)

```
\thanks{text}
```

The \thanks command produces a footnote to the title.

\title (137)

```
\title{text}
```

The \title command declares text to be the title. Use \\ to tell LaTeX where to start a new line in a long title.

\pagenumbering (138)

\pagenumbering{num_style}

Specifies the style of page numbers. Possible values of num_style are:

- arabic: Arabic numerals
- roman: Lowercase roman numerals
- Roman: Uppercase roman numerals
- alph: Lowercase letters
- Alph: Uppercase letters

\pagestyle (139)

\pagestyle {option}

The \pagestyle command changes the style from the current page on throughout the remainder of your document.

The valid options are:

- plain: Just a plain page number.

- empty: Produces empty heads and feet - no page numbers.

- headings: Puts running headings on each page. The document style specifies what goes in the headings.

- myheadings: You specify what is to go in the heading with the \markboth or the \markright commands.

\mark

\mark (140)

```
\markboth{left head}{right head} \markright{right head}
```

The \markboth and \markright commands are used in conjunction with the page style myheadings for setting either both or just the right heading. In addition to their use with the myheadings page style, you can use them to override the normal headings in the headings style, since LaTeX uses these same commands to generate those heads. You should note that a left-hand heading is generated by the last \markboth command before the end of the page, while a right-hand heading is generated by the first \markboth or \markright that comes on the page if there is one, otherwise by the last one before the page.

\thispagestyle (141)

```
\thispagestyle{option}
```

The \thispagestyle command works in the same manner as the \pagestyle command except that it changes the style for the current page only.

Sectioning (142)

Sectioning commands provide the means to structure your text into units.

- o \part
- o \chapter (report style only)
- o \section
- o \subsection
- o \subsubsection
- o \paragraph
- o \subparagraph
- o \subsubparagraph (milstd and book-form styles only)
- o \subsubsubparagraph (milstd and book-form styles only)

All sectioning commands take the same general form, i.e.,

```
\chapter[optional]{title}
```

In addition to providing the heading in the text, the mandatory argument of the sectioning command can appear in two other places:

1. the table of contents

2. the running head at the top of the page

You may not want the same thing to appear in these other two places as appears in the text heading. To handle this situation, the sectioning commands have an optional argument that provides the text for these other two purposes.

The sectioning commands have *-forms that print a title, but do not include a number and do not make an entry in the table of contents. For example, the *-form of the \subsection command could look like:

```
\subsection*{Example subsection}
```

\appendix

\appendix (143)

```
\appendix
```

The \appendix command changes the way sectional units are numbered. The \appendix command generates no text and does not affect the numbering or parts.

Spaces_and_Boxes (144)

\addvspace \bigskip \dotfill \fbox \framebox \hfill \hrulefill \
hspace \makebox \mbox \medskip \newsavebox \parbox \
raisebox \rule \savebox \smallskip \usebox \vfill \vspace

\addvspace (145)

```
\addvspace{length}
```

The \addvspace command normally adds a vertical space of height length. However, if vertical space has already been added to the same point in the output by a previous \addvspace command, then this command will not add more space than needed to make the natural length of the total vertical space equal to length.

\bigskip (146)

The \bigskip command is equivalent to \vspace{bigskipamount} where bigskipamount is determined by the document style.

\dotfill (147)

The \dotfill command produces a rubber length that produces dots instead of just spaces.

\fbox (148)

```
\fbox{text}
```

The \fbox command is exactly the same as the \mbox command, except that it puts a frame around the outside of the box that it creates.

\framebox (149)

\framebox[width][position]{text}

The \framebox command is exactly the same as the \makebox command, except that it puts a frame around the outside of the box that it creates.

The framebox command produces a rule of thickness \fboxrule, and leaves a space \fboxsep between the rule and the contents of the box.

\hfill (150)

The \hfill fill command produces a rubber length which can stretch or shrink horizontally. It will be filled with spaces.

\hrulefill (151)

The \hrulefill fill command produces a rubber length which can stretch or shrink horizontally.   It will be filled with a horizontal rule.

\hspace (152)

```
\hspace[*]{length}
```

The \hspace command adds horizontal space. The length of the space can be expressed in any terms that LaTeX understands, i.e., points, inches, etc. You can add negative as well as positive space with an \hspace command. Adding negative space is like backspacing.

LaTeX removes horizontal space that comes at the end of a line. If you don't want LaTeX to remove this space, include the optional * argument. Then the space is never removed.

\makebox (153)

```
\makebox[width][position]{text}
```

The \makebox command creates a box to contain the text specified. The width of the box is specified by the optional width argument. The position of the text within the box is determined by the optional position argument.

- c - centered (default)
- l - flushleft
- r - flushright

\mbox (154)

```
\mbox {text}
```

The \mbox command creates a box just wide enough to hold the text created by its argument.

\medskip (155)

The \medskip command is equivalent to \vspace{medskipamount} where medskipamount is determined by the document style.

\newsavebox (156)

```
\newsavebox{cmd}
```

Declares cmd, which must be a command name that is not already defined, to be a bin for saving boxes.

\parbox (157)

```
\parbox[position]{width}{text}
```

A parbox is a box whose contents are created in paragraph mode. The \parbox has two mandatory arguments:

1. width: specifies the width of the parbox; and

2. text: the text that goes inside the parbox.

LaTeX will position a parbox so its center lines up with the center of the text line. An optional first argument, position, allows you to line up either the top or bottom line in the parbox.

A \parbox command is used for a parbox containing a small piece of text, with nothing fancy inside. In particular, you shouldn't use any of the paragraph-making environments inside a \parbox argument. For larger pieces of text, including ones containing a paragraph-making environment, you should use a minipage environment.

\raisebox (158)

```
\raisebox{distance}[extend-above][extend-below]{text}
```

The \raisebox command is used to raise or lower text. The first mandatory argument specifies how high the text is to be raised (or lowered if it is a negative amount). The text itself is processed in LR mode.

Sometimes it's useful to make LaTeX think something has a different size than it really does - or a different size than LaTeX would normally think it has. The \raisebox command lets you tell LaTeX how tall it is.

The first optional argument, extend-above, makes LaTeX think that the text extends above the line by the amount specified. The second optional argument, extend-below, makes LaTeX think that the text extends below the line by the amount specified.

\rule (159)

\rule[raise-height]{width}{thickness}

The \rule command is used to produce horizontal lines. The arguments are defined as follows.

- o raise-height: specifies how high to raise the rule (optional)
- o width: specifies the length of the rule (mandatory)
- o thickness: specifies the thickness of the rule (mandatory)

\savebox (160)

```
\sbox{cmd}[text]
\savebox{cmd}[width][pos]{text}
```

These commands typeset text in a box just as for \mbox or \makebox. However, instead of printing the resulting box, they save it in bin cmd, which must have been declared with \newsavebox.

\smallskip (161)

```
\smallskip
```

The \smallskip command is equivalent to \vspace{smallskipamount} where smallskipamount is determined by the document style.

\usebox (162)

```
\usebox{cmd}
```

Prints the box most recently saved in bin cmd by a \savebox command.

\vfill (163)

The \vfill fill command produces a rubber length which can stretch or shrink vertically.

\vspace (164)

```
\vspace[*]{length}
```

The \vspace command adds vertical space. The length of the space can be expressed in any terms that LaTeX understands, i.e., points, inches, etc. You can add negative as well as positive space with an \vspace command.

LaTeX removes vertical space that comes at the end of a page. If you don't want LaTeX to remove this space, include the optional * argument. Then the space is never removed.

Special_Characters (165)

The following characters play a special role in LaTeX and are called special printing characters, or simply special characters.

$ % & ~ _ ^ \ { }

Whenever you put one of these special characters into your file, you are doing something special. If you simply want the character to be printed just as any other letter, include a \ in front of the character. For example, \$ will produce $ in your output.

The exception to the rule is the \ itself because \\ has its own special meaning. A \ is produced by typing $\backslash$ in your file.

Splitting_the_Input (166)

A large document requires a lot of input. Rather than putting the whole input in a single large file, it's more efficient to split it into several smaller ones. Regardless of how many separate files you use, there is one that is the root file; it is the one whose name you type when you run LaTeX.

\include \includeonly \input

\include (167)

```
\include{file}
```

The \include command is used in conjunction with the \includeonly command for selective inclusion of files. The file argument is the first name of a file, denoting FILE.TEX. If file is one the file names in the file list of the \includeonly command or if there is no \includeonly command, the \include command is equivalent to

```
\clearpage \input{file} \clearpage
```

except that if the file FILE.TEX does not exist, then a warning message rather than an error is produced. If the file is not in the file list, the \include command is equivalent to \clearpage.

The \include command may not appear in the preamble or in a file read by another \include command.

\includeonly (168)

```
\includeonly{file_list}
```

The \includeonly command controls which files will be read in by an \include command. It can only appear in the preamble.

\input (169)

\input{file}

The \input command causes the indicated file to be read and processed, exactly as if its contents had been inserted in the current file at that point. The file name may be a complete file name with extension or just a first name, in which case the file FILE.TEX is used.

Starting_and_Ending (170)

Your input file must contain the following commands as a minimum.

```
\documentstyle{style}
\begin{document}
   ... your text goes here ...
\end{document}
```

where the style selected is one the valid styles for LaTeX. See Document_Styles within this help file.

You may include other LaTeX commands between the \documentstyle and the \begin{document} commands.

Table_of_Contents (171)

A table of contents is produced with the \tableofcontents command. You put the command right where you want the table of contents to go; LaTeX does the rest for you. It produces a heading, but it does not automatically start a new page. If you want a new page after the table of contents, include a \newpage command after the \tableofcontents command.

There are similar commands \listoffigures and \listoftables for producing a list of figures and a list of tables, respectively. Everything works exactly the same as for the table of contents.

NOTE: If you want a any of these items to be generated, you can not have the \nofiles command in your document.

\addcontentsline \addtocontents

\addcontentsline (172)

\addcontentsline{file}{sec_unit}{entry}

The \addcontentsline command adds an entry to the specified list or table where

- file is the extension of the file on which information is to be written: toc (table of contents), lof (list of figures), or lot (list of tables).

- sec_unit controls the formatting of the entry. It should be one of the following, depending upon the value of the file argument:

  - toc: the name of the sectional unit, such as part or subsection.

  - lof: figure

  - lot: table

- entry is the text of the entry.

\addtocontents (173)

```
\addtocontents{file}{text}
```

The \addtocontents command adds text (or formatting commands) directly to the file that generates the table of contents or list of figures or tables.

- file is the extension of the file on which information is to be written: toc (table of contents), lof (list of figures), or lot (list of tables).

- text is the information to be written.

Terminal_Input_and_Output (174)

\typeout \typein

\typeout (175)

```
\typeout{msg}
```

Prints msg on the terminal and in the log file. Commands in msg that are defined with \newcommand or \renewcommand are replaced by their definitions before being printed.

LaTeX's usual rules for treating multiple spaces as a single space and ignoring spaces after a command name apply to msg. A \space command in msg causes a single space to be printed.

\typein (176)

```
\typein[cmd]{msg}
```

Prints msg on the terminal and causes LaTeX to stop and wait for you to type a line of input, ending with return. If the cmd argument is missing, the typed input is processed as if it had been included in the input file in place of the \typein command. If the cmd argument is present, it must be a command name. This command name is then defined or redefined to be the typed input.

Typefaces (177)

The typeface is specified by giving the size and style. A typeface is also called a font.

Styles	Sizes

Styles (178)

The following type style commands are supported by LaTeX.

- \rm: Roman.
- \it: Italics.
- \em: Emphasis (toggles between \it and \rm).
- \bf: Boldface.
- \sl: Slanted.
- \sf: Sans serif.
- \sc: Small caps.
- \tt: Typewriter.

Sizes (179)

The following type size commands are supported by LaTeX.

- \tiny
- \scriptsize
- \footnotesize
- \small
- \normalsize (default)
- \large
- \Large (capital "l")
- \LARGE (all caps)
- \huge
- \Huge (capital "h")

_{exp} (subscript) (180)

To get an expression exp to appear as a subscript, you just type _{exp}. Use in math mode.

SEE ALSO Math_Formulas Subscripts_and_Superscripts

^{exp} (superscript) (181)

To get an expression exp to appear as a superscript, you just type ^{exp}. Use in math mode.

SEE ALSO Math_Formulas Subscripts_and_Superscripts

\\ (182)

```
\\[*][extra-space]
```

The \\ command tells LaTeX to start a new line. It has an optional argument, extra-space, that specifies how much extra vertical space is to be inserted before the next line. This can be a negative amount.

The \\* command is the same as the ordinary \\ command except that it tells LaTeX not to start a new page after the line.

SEE ALSO Line_and_Page_Breaking

\\- (183)

The \\- command tells LaTeX that it may hyphenate the word at that point. LaTeX is very good at hyphenating, and it will usually find all correct hyphenation points. The \\- command is used for the exceptional cases.

SEE ALSO Line_and_Page_Breaking

\; (184)

Include a thick space in math mode.

SEE ALSO Math_Formulas Spacing_in_Math_Mode

\: (185)

Include a medium space in math mode.

SEE ALSO Math_Formulas Spacing_in_Math_Mode

\, (186)

Include a thin space in math mode.

SEE ALSO Math_Formulas Spacing_in_Math_Mode

\! (187)

Include a negative thin space in math mode.

SEE ALSO Math_Formulas Spacing_in_Math_Mode

\= (188)

The \= command sets the tab stops.

SEE ALSO Environments tabbing

\> (189)

The \> command causes LaTeX to advance to the next tab stop.

SEE ALSO Environments tabbing

\< (190)

The \< command allows you to put something to the left of the local margin without changing the margin.

SEE ALSO Environments tabbing

\+ (191)

The \+ command moves the left margin of the next and all the following commands one tab stop to the right.

SEE ALSO Environments tabbing

\- (192)

The \- command moves the left margin of the next and all the following commands one tab stop to the left.

SEE ALSO Environments tabbing

\' (193)

The \' command moves everything that you have typed so far in the current column , everything starting from the most recent \>, \<, \', \\, or \kill command, to the right of the previous column, flush against the current column's tab stop.

SEE ALSO Environments tabbing

\` (194)

The \` command allows you to put text flushed right against any tab stop, including tab stop 0. However, it can't move text to the right of the last column because there's no tab stop there. The \` command moves all the text that follows it, up to the \\ or \end{tabbing} command that ends the line, to the right margin of the tabbing environment. There must be no \> or \' command between the \` and the command that ends the line.

SEE ALSO Environments tabbing

\addcontentsline (195)

\addcontentsline{file}{sec_unit}{entry}

The \addcontentsline command adds an entry to the specified list or table where

- file is the extension of the file on which information is to be written: toc (table of contents), lof (list of figures), or lot (list of tables).

- sec_unit controls the formatting of the entry. It should be one of the following, depending upon the value of the file argument:

  - toc: the name of the sectional unit, such as part or subsection.

  - lof: figure

  - lot: table

- entry is the text of the entry.

SEE ALSO Table_of_Contents

\addtocontents (196)

```
\addtocontents{file}{text}
```

The \addtocontents command adds text (or formatting commands) directly to the file that generates the table of contents or list of figures or tables.

- file is the extension of the file on which information is to be written: toc (table of contents), lof (list of figures), or lot (list of tables).

- text is the information to be written.

SEE ALSO Table_of_Contents

\addtocounter (197)

```
\addtocounter{counter}{value}
```

The \addtocounter command increments the counter by the amount specified by the value argument. The value argument can be negative.

SEE ALSO Counters

\address (198)

```
\address{Return address}
```

The return address, as it should appear on the letter and the envelope. Separate lines of the address should be separated by \\ commands. If you do not make an \address declaration, then the letter will be formatted for copying onto your organization's standard letterhead. If you give an \address declaration, then the letter will be formatted as a personal letter.

SEE ALSO Letters Declarations

\addtolength (199)

```
\addtolength{\gnat}{length}
```

The \addtolength command increments a length command by the amount specified in the length argument. It can be a negative amount.

SEE ALSO Lengths

\addvspace (200)

```
\addvspace{length}
```

The \addvspace command normally adds a vertical space of height length. However, if vertical space has already been added to the same point in the output by a previous \addvspace command, then this command will not add more space than needed to make the natural length of the total vertical space equal to length.

SEE ALSO Spaces_and_Boxes

\alph (201)

```
\alph{counter}
```

This command causes the value of the counter to be printed in alphabetic characters. The \alph command causes lower case alphabetic characters, i.e., a, b, c... while the \Alph command causes upper case alphabetic characters, i.e., A, B, C...

SEE ALSO Counters

\appendix (202)

```
\appendix
```

The \appendix command changes the way sectional units are numbered. The \appendix command generates no text and does not affect the numbering or parts.

SEE ALSO Sectioning

\arabic (203)

\arabic {counter}

The \arabic command causes the value of the counter to be printed in arabic numbers, i.e., 3.

SEE ALSO Counters

array (204)

```
\begin{array}{col1col2...coln}
column 1 entry & column 2 entry ... & column n entry \\
.
.
.
\end{array}
```

Math arrays are produced with the array environment. It has a single mandatory argument describing the number of columns and the alignment within them. Each column, coln, is specified by a single letter that tells how items in that row should be formatted.

- c for centered
- l for flushleft
- r for flushright

Column entries must be separated by an &. Column entries may include other LaTeX commands. Each row of the array must be terminated with the string \\.

SEE ALSO Environments

\author (205)

```
\author{names}
```

The \author command declares the author(s), where names is a list of authors separated by \and commands. Use \\ to separate lines within a single author's entry -- for example, to give the author's institution or address.

NOTE: The milstd and book-form styles have re-defined the \maketitle command. The \title declaration is the only command of those shown below that has any meaning.

SEE ALSO Page_Styles \maketitle

\bf (206)

Boldface typeface.

SEE ALSO Typefaces Styles

\bibitem (207)

```
\bibitem[label]{cite_key}
```

The \bibitem command generates an entry labeled by label. If the label argument is missing, a number is generated as the label, using the enumi counter. The cite_key is any sequence of letters, numbers, and punctuation symbols not containing a comma. This command writes an entry on the aux file containing cite_key and the item's label. When this aux file is read by the \begin{document} command, the item's label is associated with cite_key, causing the reference to cite_key by a \cite command to produce the associated label.

SEE ALSO Environments thebibliography

\bigskip (208)

The \bigskip command is equivalent to \vspace{bigskipamount} where bigskipamount is determined by the document style.

SEE ALSO Spaces_and_Boxes

\cdots (209)

The \cdots command produces a horizontal ellipsis where the dots are raised to the center of the line.

SEE ALSO Math_Formulas Math_Miscellany

center (210)

```
\begin{center}
Text on line 1 \\
Text on line 2 \\
.
.
.
\end{center}
```

The center environment allows you to create a paragraph consisting of lines that are centered within the left and right margins on the current page. Each line must be terminated with a \\.

SEE ALSO Environments

\centering (211)

This declaration corresponds to the center environment. This declaration can be used inside an environment such as quote or in a parbox. The text of a figure or table can be centered on the page by putting a \centering command at the beginning of the figure or table environment.

Unlike the center environment, the \centering command does not start a new paragraph; it simply changes how LaTeX formats paragraph units. To affect a paragraph unit's format, the scope of the declaration must contain the blank line or \end command (of an environment like quote) that ends the paragraph unit.

SEE ALSO Environments center

\circle (212)

```
\circle[*]{diameter}
```

The \circle command produces a circle of the specified diameter. If the *-form of the command is used, LaTeX draws a solid circle.

SEE ALSO Environments picture

\cite (213)

```
\cite[text]{key_list}
```

The key_list argument is a list of citation keys. This command generates an in-text citation to the references associated with the keys in key_list by entries on the aux file read by the \begin{document} command.

SEE ALSO Environments thebibliography

\cleardoublepage (214)

The \cleardoublepage command ends the current page and causes all figures and tables that have so far appeared in the input to be printed. In a two-sided printing style, it also makes the next page a right-hand (odd-numbered) page, producing a blank page if necessary.

SEE ALSO Line_and_Page_Breaking

\clearpage (215)

The \clearpage command ends the current page and causes all figures and tables that have so far appeared in the input to be printed.

SEE ALSO Line_and_Page_Breaking

\cline (216)

```
\cline{i-j}
```

The \cline command draws horizontal lines across the columns specified, beginning in column i and ending in column j, which are identified in the mandatory argument.

SEE ALSO Environments tabular

\closing (217)

```
\closing{text}
```

The letter closes with a \closing command, i.e.,

```
\closing{Best Regards,}
```

SEE ALSO Letters

\dashbox (218)

```
\dashbox{dash length}(width,height){ ...   }
```

The \dashbox has an extra argument which specifies the width of each dash. A dashed box looks best when the width and height are multiples of the dash length.

SEE ALSO Environments picture

\date (219)

```
\date{text}
```

The \date command declares text to be the document's date. With no \date command, the current date is used.

SEE ALSO Page_Styles \maketitle

\ddots (220)

The \ddots command produces a diagonal ellipsis.

SEE ALSO Math_Formulas Math_Miscellany

description (221)

```
\begin{description}
\item [label] First item
\item [label] Second item
.
.
.
\end{description}
```

The description environment is used to make labeled lists. The label is bold face and flushed right.

SEE ALSO Environments

\dotfill (222)

The \dotfill command produces a rubber length that produces dots instead of just spaces.

SEE ALSO Spaces_and_Boxes

\em (223)

Emphasis (toggles between \it and \rm).

SEE ALSO Typefaces Styles

enumerate (224)

```
\begin{enumerate}
\item First item
\item Second item
.
.
.
\end{enumerate}
```

The enumerate environment produces a numbered list. Enumerations can be nested within one another, up to four levels deep. They can also be nested within other paragraph-making environments.

Each item of an enumerated list begins with an \item command. There must be at least one \item command within the environment.

SEE ALSO Environments

eqnarray (225)

```
\begin{eqnarray}
math formula 1 \\
math formula 2 \\
.
.
.
\end{eqnarray}
```

The eqnarray environment is used to display a sequence of equations or inequalities. It is very much like a three-column array environment, with consecutive rows separated by \\ and consecutive items within a row separated by an &. An equation number is placed on every line unless that line has a \nonumber command.

SEE ALSO Environments

equation (226)

```
\begin{equation}
  math formula
\end{equation}
```

The equation environment centers your equation on the page and places the equation number in the right margin.

SEE ALSO Environments

figure (227)

```
\begin{figure}[placement]

  body of the figure

\caption{figure title}
\end{figure}
```

Figures are objects that are not part of the normal text, and are usually "floated" to a convenient place, like the top of a page. Figures will not be split between two pages.

The optional argument [placement] determines where LaTeX will try to place your figure. There are four places where LaTeX can possibly put a float:

- h: Here - at the position in the text where the figure environment appears.
- t: Top - at the top of a text page.
- b: Bottom - at the bottom of a text page.
- p: Page of floats - on a separate float page, which is a page containing no text, only floats.

The standard report and article styles use the default placement tbp.

The body of the figure is made up of whatever text, LaTeX commands, etc., you wish. The \caption command allows you to title your figure.

SEE ALSO Environments

\fbox (228)

\fbox{text}

The \fbox command is exactly the same as the \mbox command, except that it puts a frame around the outside of the box that it creates.

SEE ALSO Spaces_and_Boxes

\flushbottom (229)

The \flushbottom declaration makes all text pages the same height, adding extra vertical space when necessary to fill out the page.

SEE ALSO Document_Styles

flushleft (230)

```
\begin{flushleft}
Text on line 1 \\
Text on line 2 \\
.
.
.
\end{flushleft}
```

The flushleft environment allows you to create a paragraph consisting of lines that are flushed left to the left-hand margin. Each line must be terminated with a \\.

SEE ALSO Environments

flushright (231)

```
\begin{flushright}
Text on line 1 \\
Text on line 2 \\
.
.
.
\end{flushright}
```

The flushright environment allows you to create a paragraph consisting of lines that are flushed right to the right-hand margin. Each line must be terminated with a \\.

SEE ALSO Environments

\fnsymbol (232)

```
\fnsymbol{counter}
```

The \fnsymbol command causes the value of the counter to be printed in a specific sequence of nine symbols that can be used for numbering footnotes.

SEE ALSO Counters

\footnote (233)

```
\footnote[number]{text}
```

The \footnote command places the numbered footnote text at the bottom of the current page. The optional argument, number, is used to change the default footnote number. This command can only be used in outer paragraph mode.

SEE ALSO Footnotes

\footnotemark (234)

The \footnotemark command puts the footnote number in the text. This command can be used in inner paragraph mode. The text of the footnote is supplied by the \footnotetext command.

SEE ALSO Footnotes

\footnotesize (235)

Third smallest of 10 typefaces available. This is the default size for footnotes.

SEE ALSO Typefaces Sizes

\footnotetext (236)

\footnotetext [number] {text}

The \footnotetext command produces the text to be placed at the bottom of the page. This command can come anywhere after the \footnotemark command. The \footnotetext command must appear in outer paragraph mode.

The optional argument, number, is used to change the default footnote number.

SEE ALSO Footnotes

\frac (237)

```
\frac{num}{den}
```

The \frac command produces the fraction num divided by den.

SEE ALSO Math_Formulas Math_Miscellany

\frame (238)

```
\frame{ ...   }
```

The \frame command puts a rectangular frame around the object specified in the argument. The reference point is the bottom left corner of the frame. No extra space is put between the frame and the object.

SEE ALSO Environments picture

\framebox (239)

```
\framebox[width][position]{text}
```

The \framebox command is exactly the same as the \makebox command, except that it puts a frame around the outside of the box that it creates.

The framebox command produces a rule of thickness \fboxrule, and leaves a space \fboxsep between the rule and the contents of the box.

SEE ALSO Spaces_and_Boxes Environments picture

\hfill (240)

The \hfill fill command produces a rubber length which can stretch or shrink horizontally. It will be filled with spaces.

SEE ALSO Spaces_and_Boxes

\hline (241)

The \hline command will draw a horizontal line the width of the table. It's most commonly used to draw a line at the top, bottom, and between the rows of the table.

SEE ALSO Environments tabular

\hrulefill (242)

The \hrulefill fill command produces a rubber length which can stretch or shrink horizontally.   It will be filled with a horizontal rule.

SEE ALSO   Spaces_and_Boxes

\hspace (243)

```
\hspace[*]{length}
```

The \hspace command adds horizontal space. The length of the space can be expressed in any terms that LaTeX understands, i.e., points, inches, etc. You can add negative as well as positive space with an \hspace command. Adding negative space is like backspacing.

LaTeX removes horizontal space that comes at the end of a line. If you don't want LaTeX to remove this space, include the optional * argument. Then the space is never removed.

SEE ALSO Spaces_and_Boxes

\huge (244)

Second largest of 10 typefaces available.

SEE ALSO Typefaces Sizes

\Huge (capital "h") (245)

Largest of 10 typefaces available. All fonts may not be available in this size.

SEE ALSO Typefaces Sizes

\hyphenation (246)

```
\hyphenation{words}
```

The \hyphenation command declares allowed hyphenation points, where words is a list of words, separated by spaces, in which each hyphenation point is indicated by a - character.

SEE ALSO Line_and_Page_Breaking

\include (247)

```
\include{file}
```

The \include command is used in conjunction with the \includeonly command for selective inclusion of files. The file argument is the first name of a file, denoting FILE.TEX. If file is one the file names in the file list of the \includeonly command or if there is no \includeonly command, the \include command is equivalent to

```
\clearpage \input{file} \clearpage
```

except that if the file FILE.TEX does not exist, then a warning message rather than an error is produced. If the file is not in the file list, the \include command is equivalent to \clearpage.

The \include command may not appear in the preamble or in a file read by another \include command.

SEE ALSO Splitting_the_Input

\includeonly (248)

\includeonly{file_list}

The \includeonly command controls which files will be read in by an \include command. It can only appear in the preamble.

SEE ALSO Splitting_the_Input

\indent (249)

This produces a horizontal space whose width equals the width of the paragraph indentation. It is used to add paragraph indentation where it would otherwise be suppressed.

SEE ALSO Making_Paragraphs

\input (250)

```
\input{file}
```

The \input command causes the indicated file to be read and processed, exactly as if its contents had been inserted in the current file at that point. The file name may be a complete file name with extension or just a first name, in which case the file FILE.TEX is used.

SEE ALSO Splitting_the_Input

\it (251)

Italics typeface.

SEE ALSO Typefaces Styles

itemize (252)

```
\begin{itemize}
\item First item
\item Second item
.
.
.
\end{itemize}
```

The itemize environment produces a bulleted list. Itemizations can be nested within one another, up to four levels deep. They can also be nested within other paragraph-making environments.

Each item of an itemized list begins with an \item command. There must be at least one \item command within the environment.

SEE ALSO Environments

\kill (253)

The \kill command allows you to set tab stops without producing text. It works just like the \\ except that it throws away the current line instead of producing output for it. The effect of any \=, \+ or \- commands in that line remain in effect.

SEE ALSO Environments tabbing

\label (254)

```
\label{key}
```

A \label command appearing in ordinary text assigns to the key the number of the current sectional unit; one appearing inside a numbered environment assigns that number to the key.

A key con consist of any sequence of letters, digits, or punctuation characters. Upper- and lowercase letters are different.

SEE ALSO Cross_References

\large (255)

Slightly larger than default typeface size.

SEE ALSO Typefaces Sizes

\Large (capital "l") (256)

Fourth largest of typefaces available. Is generally the default for titles.

SEE ALSO Typefaces Sizes

\LARGE (all caps) (257)

Third largest of typefaces available.

SEE ALSO Typefaces Sizes

\ldots (258)

The \ldots command produces an ellipsis. This command works in any mode, not just math mode.

SEE ALSO Math_Formulas Math_Miscellany

\line (259)

```
\line(x slope,y slope){length}
```

The \line command draws a line of the specified length and slope.

SEE ALSO Environments picture

\linebreak (260)

```
\linebreak[number]
```

The \linebreak command tells LaTeX to break the current line at the point of the command. With the optional argument, number, you can convert the \linebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

The \linebreak command causes LaTeX to stretch the line so it extends to the right margin.

SEE ALSO Line_and_Page_Breaking

\linethickness (261)

\linethickness{dimension}

Declares the thickness of horizontal and vertical lines in a picture environment to be dimension, which must be a positive length. It does not affect the thickness of slanted lines and circles, or the quarter circles drawn by \oval to form the corners of an oval.

SEE ALSO Environments picture

list (262)

```
\begin{list}{label}{spacing}
\item First item
\item Second item
.
.
.
\end{list}
```

The {label} argument specifies how items should be labeled. This argument is a piece of text that is inserted in a box to form the label. This argument can and usually does contain other LaTeX commands.

The {spacing} argument contains commands to change the spacing parameters for the list. This argument will most often be null, i.e. {}. This will select all default spacing which should suffice for most cases.

SEE ALSO Environments

\location (263)

```
\location {address}
```

This modifies your organization's standard address. This only appears if the firstpage pagestyle is selected.

SEE ALSO Letters Declarations

\makebox (264)

\makebox[width][position]{text}

The \makebox command creates a box to contain the text specified. The width of the box is specified by the optional width argument. The position of the text within the box is determined by the optional position argument.

- c - centered (default)
- l - flushleft
- r - flushright

SEE ALSO Spaces_and_Boxes

* * *

\makebox(width,height)[position]{ ... }

The \makebox command for the picture environment is similar to the normal \makebox command except that you must specify a width and height in multiples of \unitlength.

The optional argument, [position], specifies the quadrant that your text appears in. You may select up to two of the following:

- t: Moves the item to the top of the rectangle
- b: Moves the item to the bottom
- l: Moves the item to the left
- r: Moves the item to the right

SEE ALSO Environments picture

\maketitle (265)

```
\maketitle
```

The \maketitle command generates a title on a separate title page - except in the article style, where the title normally goes at the top of the first page. Information used to produce the title is obtained from the following declarations.

SEE ALSO Page_Styles

\mark (266)

```
\markboth{left head}{right head} \markright{right head}
```

The \markboth and \markright commands are used in conjunction with the page style myheadings for setting either both or just the right heading. In addition to their use with the myheadings page style, you can use them to override the normal headings in the headings style, since LaTeX uses these same commands to generate those heads. You should note that a left-hand heading is generated by the last \markboth command before the end of the page, while a right-hand heading is generated by the first \markboth or \markright that comes on the page if there is one, otherwise by the last one before the page.

SEE ALSO Page_Styles \pagestyle

\mbox (267)

```
\mbox{text}
```

The \mbox command creates a box just wide enough to hold the text created by its argument.

SEE ALSO Spaces_and_Boxes

\medskip (268)

The \medskip command is equivalent to \vspace{medskipamount} where medskipamount is determined by the document style.

SEE ALSO Spaces_and_Boxes

minipage (269)

```
\begin{minipage}[position]{width}
  text
\end{minipage}
```

The minipage environment is similar to a \parbox command. It takes the same optional position argument and mandatory width argument. You may use other paragraph-making environments inside a minipage.

Footnotes in a minipage environment are handled in a way that is particularly useful for putting footnotes in figures or tables. A \footnote or \footnotetext command puts the footnote at the bottom of the minipage instead of at the bottom of the page, and it uses the mpfootnote counter instead of the ordinary footnote counter.

NOTE: Don't put one minipage inside another if you are using footnotes; they may wind up at the bottom of the wrong minipage.

SEE ALSO Environments

\multicolumn (270)

```
\multicolumn{cols}{pos}{text}
```

The \multicolumn is used to make an entry that spans several columns. The first mandatory argument, cols, specifies the number of columns to span. The second mandatory argument, pos, specifies the formatting of the entry; c for centered, l for flushleft, r for flushright. The third mandatory argument, text, specifies what text is to make up the entry.

SEE ALSO Environments tabular

\multiput (271)

```
\multiput(x coord,y coord)(delta x,delta y){number of copies}{object}
```

The \multiput command can be used when you are putting the same object in a regular pattern across a picture.

SEE ALSO Environments picture

\newcommand (272)

```
\newcommand{cmd}[args]{def}
\renewcommand{cmd}[args]{def}
```

These commands define (or redefine) a command.

- cmd: A command name beginning with a \. For \newcommand it must not be already defined and must not begin with \end; for \renewcommand it must already be defined.

- args: An integer from 1 to 9 denoting the number of arguments of the command being defined. The default is for the command to have no arguments.

- def: The text to be substituted for every occurrence of cmd; a parameter of the form #n in cmd is replaced by the text of the nth argument when this substitution takes place.

SEE ALSO Definitions

\newcounter (273)

```
\newcounter{foo}[counter]
```

The \newcounter command defines a new counter named foo. The optional argument [counter] causes the counter foo to be reset whenever the counter named in the optional argument is incremented.

SEE ALSO Counters

\newenvironment (274)

```
\newenvironment{nam}[args]{begdef}{enddef}
\renewenvironment{nam}[args]{begdef}{enddef}
```

These commands define or redefine an environment.

- nam: The name of the environment. For \newenvironment there must be no currently defined environment by that name, and the command \nam must be undefined. For \renewenvironment the environment must already be defined.

- args: An integer from 1 to 9 denoting the number of arguments of the newly-defined environment. The default is no arguments.

- begdef: The text substituted for every occurrence of \begin{name}; a parameter of the form #n in cmd is replaced by the text of the nth argument when this substitution takes place.

- enddef: The text substituted for every occurrence of \end{nam}. It may not contain any argument parameters.

SEE ALSO Definitions

\newfont (275)

```
\newfont{cmd}{font_name}
```

Defines the command name cmd, which must not be currently defined, to be a declaration that selects the font named font_name to be the current font.

SEE ALSO Definitions

\newlength (276)

```
\newlength{\gnat}
```

The \newlength command defines the mandatory argument, \gnat, as a length command with a value of 0in. An error occurs if a \gnat command already exists.

SEE ALSO Lengths

\newline (277)

The \newline command breaks the line right where it is. The \newline command can be used only in paragraph mode.

SEE ALSO Line_and_Page_Breaking

\newpage (278)

The \newpage command ends the current page.

SEE ALSO Line_and_Page_Breaking

\newsavebox (279)

```
\newsavebox{cmd}
```

Declares cmd, which must be a command name that is not already defined, to be a bin for saving boxes.

SEE ALSO Spaces_and_Boxes

\newtheorem (280)

```
\newtheorem{env_name}{caption}[within]
\newtheorem{env_name}[numbered_like]{caption}
```

This command defines a theorem-like environment.

- env_name: The name of the environment -- a string of letters. Must not be the name of an existing environment or counter.

- caption: The text printed at the beginning of the environment, right before the number.

- within: The name of an already defined counter, usually of a sectional unit. Provides a means of resetting the new theorem counter within the sectional unit.

- numbered_like: The name of an already defined theorem-like environment.

The \newtheorem command may have at most one optional argument.

SEE ALSO Definitions

\nocite (281)

```
\nocite{key_list}
```

The \nocite command produces no text, but writes key_list, which is a list of one or more citation keys, on the aux file.

SEE ALSO Environments thebibliography

\noindent (282)

When used at the beginning of the paragraph, it suppresses the paragraph indentation. It has no effect when used in the middle of a paragraph.

SEE ALSO Making_Paragraphs

\nolinebreak (283)

\nolinebreak[number]

The \nolinebreak command prevents LaTeX from breaking the current line at the point of the command. With the optional argument, number, you can convert the \nolinebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

SEE ALSO Line_and_Page_Breaking

\normalsize (default) (284)

The size of \normalsize is defined by as 10pt unless the 11pt or 12pt document style option is used.

SEE ALSO Typefaces Sizes

\nopagebreak (285)

```
\nopagebreak[number]
```

The \nopagebreak command prevents LaTeX form breaking the current page at the point of the command. With the optional argument, number, you can convert the \nopagebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

SEE ALSO Line_and_Page_Breaking

\onecolumn (286)

The \onecolumn declaration starts a new page and produces single-column output.

SEE ALSO Document_Styles

\opening (287)

```
\opening{text}
```

The letter begins with the \opening command. The mandatory argument, text, is what ever text you wish to start your letter, i.e.,

```
\opening{Dear John,}
```

SEE ALSO Letters

\oval (288)

\oval(width,height)[portion]

The \oval command produces a rectangle with rounded corners. The optional argument, [portion], allows you to select part of the oval.

- t: Selects the top portion
- b: Selects the bottom portion
- r: Selects the right portion
- l: Selects the left portion

SEE ALSO Environments picture

\overbrace (289)

```
\overbrace{text}
```

The \overbrace command generates a brace over text.

SEE ALSO Math_Formulas Math_Miscellany

\overline (290)

```
\overline{text}
```

The \overline command causes the argument text to be overlined.

SEE ALSO Math_Formulas Math_Miscellany

\pagebreak (291)

```
\pagebreak[number]
```

The \pagebreak command tells LaTeX to break the current page at the point of the command. With the optional argument, number, you can convert the \pagebreak command from a demand to a request. The number must be a number from 0 to 4. The higher the number, the more insistent the request is.

SEE ALSO <u>Line_and_Page_Breaking</u>

\pagenumbering (292)

\pagenumbering{num_style}

Specifies the style of page numbers. Possible values of num_style are:

- arabic: Arabic numerals
- roman: Lowercase roman numerals
- Roman: Uppercase roman numerals
- alph: Lowercase letters
- Alph: Uppercase letters

SEE ALSO Page_Styles

\pageref (293)

```
\pageref{key}
```

The \pageref command produces the page number of the place in the text where the corresponding \label command appears.

SEE ALSO Cross_References

\pagestyle (294)

\pagestyle{option}

The \pagestyle command changes the style from the current page on throughout the remainder of your document.

The valid options are:

- plain: Just a plain page number.

- empty: Produces empty heads and feet - no page numbers.

- headings: Puts running headings on each page. The document style specifies what goes in the headings.

- myheadings: You specify what is to go in the heading with the \markboth or the \markright commands.

SEE ALSO Page_Styles

\par (295)

Equivalent to a blank line; often used to make command or environment definitions easier to read.

SEE ALSO Making_Paragraphs

\parbox (296)

```
\parbox[position]{width}{text}
```

A parbox is a box whose contents are created in paragraph mode. The \parbox has two mandatory arguments:

1. width: specifies the width of the parbox, and

2. text: the text that goes inside the parbox.

LaTeX will position a parbox so its center lines up with the center of the text line. An optional first argument, position, allows you to line up either the top or bottom line in the parbox.

A \parbox command is used for a parbox containing a small piece of text, with nothing fancy inside. In particular, you shouldn't use any of the paragraph-making environments inside a \parbox argument. For larger pieces of text, including ones containing a paragraph-making environment, you should use a minipage environment.

SEE ALSO <u>Spaces_and_Boxes</u>

picture (297)

```
\begin{picture}(width,height)(x offset,y offset)
.
  picture commands
.
\end{picture}
```

The picture environment allows you to create just about any kind of picture you want containing text, lines, arrows and circles. You tell LaTeX where to put things in the picture by specifying their coordinates. A coordinate is a number that may have a decimal point and a minus sign - a number like 5, 2.3 or -3.1416. A coordinate specifies a length in multiples of the unit length \unitlength, so if \unitlength has been set to 1cm, then the coordinate 2.54 specifies a length of 2.54 centimeters. You can change the value of \unitlength anywhere you want, using the \setlength command, but strange things will happen if you try changing it inside the picture environment.

A position is a pair of coordinates, such as (2.4,-5), specifying the point with x-coordinate 2.4 and y-coordinate -5. Coordinates are specified in the usual way with respect to an origin, which is normally at the lower-left corner of the picture. Note that when a position appears as an argument, it is not enclosed in braces; the parentheses serve to delimit the argument.

The picture environment has one mandatory argument, which is a position. It specifies the size of the picture. The environment produces a rectangular box with width and height determined by this argument's x- and y-coordinates.

The picture environment also has an optional position argument, following the size argument, that can change the origin. (Unlike ordinary optional arguments, this argument is not contained in square brackets.) The optional argument gives the coordinates of the point at the lower-left corner of the picture (thereby determining the origin). For example, if \unitlength has been set to 1mm, the command

```
\begin{picture}(100,200)(10,20)
```

produces a picture of width 100 millimeters and height 200 millimeters, whose lower-left corner is the point (10,20) and whose upper-right corner is therefore the point (110,220). When you first draw a picture, you will omit the optional argument, leaving the origin at the lower-left corner. If you then want to modify your picture by shifting everything, you just add the appropriate optional argument.

The environment's mandatory argument determines the nominal size of the picture. This need bear no relation to how large the picture really is; LaTeX will happily allow you to put things outside the picture, or even off the page. The picture's nominal size is used by TeX in determining how much room to leave for it.

Everything that appears in a picture is drawn by the \put command. The command

```
\put (11.3,-.3){ ... }
```

puts the object specified by "..." in the picture, with its reference point at coordinates (11.3,-.3). The reference points for various objects will be described below.

The \put command creates an LR box. You can put anything in the text argument of the \put command that you'd put into the argument of an \mbox and related commands. When you do this, the reference point will be the lower left corner of the box.

SEE ALSO Environments

\put (298)

```
\put(x coord,y coord){ ...   }
```

The \put command places the item specified by the mandatory argument at the given coordinates.

SEE ALSO Environments picture

quotation (299)

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

The margins of the quotation environment are indented on the left and the right. The text is justified at both margins and there is paragraph indentation. Leaving a blank line between text produces a new paragraph.

SEE ALSO Environments

quote (300)

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

The margins of the quote environment are indented on the left and the right. The text is justified at both margins. Leaving a blank line between text produces a new paragraph.

SEE ALSO Environments

\raggedbottom (301)

The \raggedbottom declaration makes all pages the height of the text on that page. No extra vertical space is added.

SEE ALSO Document_Styles

\raggedleft (302)

This declaration corresponds to the flushright environment. This declaration can be used inside an environment such as quote or in a parbox.

Unlike the flushright environment, the \raggedleft command does not start a new paragraph; it simply changes how LaTeX formats paragraph units. To affect a paragraph unit's format, the scope of the declaration must contain the blank line or \end command (of an environment like quote) that ends the paragraph unit.

SEE ALSO Environments flushright

\raggedright (303)

This declaration corresponds to the flushleft environment. This declaration can be used inside an environment such as quote or in a parbox.

Unlike the flushleft environment, the \raggedright command does not start a new paragraph; it simply changes how LaTeX formats paragraph units. To affect a paragraph unit's format, the scope of the declaration must contain the blank line or \end command (of an environment like quote) that ends the paragraph unit.

SEE ALSO Environments flushleft

\raisebox (304)

```
\raisebox{distance}[extend-above][extend-below]{text}
```

The \raisebox command is used to raise or lower text. The first mandatory argument specifies how high the text is to be raised (or lowered if it is a negative amount). The text itself is processed in LR mode.

Sometimes it's useful to make LaTeX think something has a different size than it really does - or a different size than LaTeX would normally think it has. The \raisebox command lets you tell LaTeX how tall it is.

The first optional argument, extend-above, makes LaTeX think that the text extends above the line by the amount specified. The second optional argument, extend-below, makes LaTeX think that the text extends below the line by the amount specified.

SEE ALSO Spaces_and_Boxes

\ref (305)

```
\ref{key}
```

The \ref command produces the number of the sectional unit, equation number, ... of the corresponding \label command.

SEE ALSO Cross_References

\rm (306)

Roman typeface (default).

SEE ALSO Typefaces Styles

\roman (307)

\roman {counter}

This command causes the value of the counter to be printed in roman numerals. The \roman command causes lower case roman numerals, i.e., i, ii, iii..., while the \Roman command causes upper case roman numerals, i.e., I, II, III...

SEE ALSO Counters

\rule (308)

```
\rule[raise-height]{width}{thickness}
```

The \rule command is used to produce horizontal lines. The arguments are defined as follows.

- o raise-height: specifies how high to raise the rule (optional)
- o width: specifies the length of the rule (mandatory)
- o thickness: specifies the thickness of the rule (mandatory)

SEE ALSO Spaces_and_Boxes

\savebox (309)

```
\sbox{cmd}[text]
\savebox{cmd}[width][pos]{text}
```

These commands typeset text in a box just as for \mbox or \makebox. However, instead of printing the resulting box, they save it in bin cmd, which must have been declared with \newsavebox.

SEE ALSO Spaces_and_Boxes

\sc (310)

Small caps typeface.

SEE ALSO Typefaces Styles

\scriptsize (311)

Second smallest of 10 typefaces available.

SEE ALSO Typefaces Sizes

\setcounter (312)

```
\setcounter{counter}{value}
```

The \setcounter command sets the value of the counter to that specified by the value argument.

SEE ALSO Counters

\setlength (313)

```
\setlength{\gnat}{length}
```

The \setlength command is used to set the value of a length command. The length argument can be expressed in any terms of length LaTeX understands, i.e., inches (in), millimeters (mm), points (pt), etc.

SEE ALSO Lengths

\settowidth (314)

```
\settowidth{\gnat}{text}
```

The \settowidth command sets the value of a length command equal to the width of the text argument.

SEE ALSO Lengths

\sf (315)

Sans serif typeface.

SEE ALSO Typefaces Styles

\shortstack (316)

```
\shortstack[position]{...   \\ ...   \\ ...}
```

The \shortstack command produces a stack of objects. The valid positions are:

- r: Moves the objects to the right of the stack
- l: Moves the objects to the left of the stack
- c: Moves the objects to the center of the stack (default)

SEE ALSO Environments picture

\signature (317)

```
\signature{Your name}
```

Your name, as it should appear at the end of the letter underneath the space for your signature. Items that should go on separate lines should be separated by \\ commands.

SEE ALSO Letters Declarations

\sl (318)

Slanted typeface.

SEE ALSO Typefaces Styles

\small (319)

Slightly smaller than default typeface size.

SEE ALSO Typefaces Sizes

\smallskip (320)

```
\smallskip
```

The \smallskip command is equivalent to \vspace{smallskipamount} where smallskipamount is determined by the document style.

SEE ALSO Spaces_and_Boxes

\sqrt (321)

```
\sqrt[root]{arg}
```

The \sqrt command produces the square root of its argument. The optional argument, root, determines what root to produce, i.e. the cube root of x+y would be typed as $\sqrt[3]{x+y}$.

SEE ALSO Math_Formulas Math_Miscellany

tabbing (322)

```
\begin{tabbing}
text \= more text \= still more text \= last text \\
second row \>   \> more \\
.
.
.
\end{tabbing}
```

The tabbing environment provides a way to align text in columns. It works by setting tab stops and tabbing to them much the way you do with an ordinary typewriter.

SEE ALSO Environments tabbing

table (323)

```
\begin{table}[placement]

  body of the table

\caption{table title}
\end{table}
```

Tables are objects that are not part of the normal text, and are usually "floated" to a convenient place, like the top of a page. Tables will not be split between two pages.

The optional argument [placement] determines where LaTeX will try to place your table. There are four places where LaTeX can possibly put a float:

- h: Here - at the position in the text where the table environment appears.
- t: Top - at the top of a text page.
- b: Bottom - at the bottom of a text page.
- p: Page of floats - on a separate float page, which is a page containing no text, only floats.

The standard report and article styles use the default placement tbp.

The body of the table is made up of whatever text, LaTeX commands, etc., you wish. The \caption command allows you to title your table.

SEE ALSO Environments

tabular (324)

```
\begin{tabular}[pos]{cols}
column 1 entry & column 2 entry ... & column n entry \\
.
.
.
\end{tabular}

                    or

\begin{tabular*}{width}[pos]{cols}
column 1 entry & column 2 entry ... & column n entry \\
.
.
.
\end{tabular*}
```

These environments produce a box consisting of a sequence of rows of items,aligned vertically in columns. The mandatory and optional arguments consist of:

o width: Specifies the width of the tabular* environment. There must be rubber space between columns that can stretch to fill out the specified width.

o pos: Specified the vertical position; default is alignment on the center of the environment.

- t - align on top row

- b - align on bottom row

o cols: Specifies the column formatting. It consists of a sequence of the following specifiers, corresponding to the sequence of columns and intercolumn material.

- l - A column of left-aligned items.

- r - A column of right-aligned items.

- c - A column of centered items.

- | - A vertical line the full height and depth of the environment.

- @{text} - This inserts text in every row. An @-expression suppresses the intercolumn space normally inserted between columns; any desired space between the inserted text and the adjacent items must be included in text. An \extracolsep{wd} command in an @-expression causes an extra space of width wd to appear to the left of all subsequent columns, until countermanded by another \extracolsep command. Unlike ordinary intercolumn space, this extra space is not suppressed by an @-expression. An \extracolsep command can be used only in an @-expression in the cols argument.

- p{wd} - Produces a column with each item typeset in a parbox of width wd, as if it were the argument of a \parbox[t]{wd} command. However, a \\ may not appear in the item, except in the following situations: (i) inside an environment like minipage, array, or tabular, (ii) inside an explicit \parbox, or (iii) in the scope of a \centering, \raggedright, or \raggedleft declaration. The latter declarations must appear inside braces or an environment when used in a p-column element.

- *{num}{cols} - Equivalent to num copies of cols, where num is any positive integer and cols is any list of column-specifiers, which may contain another *-expression.

SEE ALSO Environments

\telephone (325)

```
\telephone{number}
```

This is your telephone number. This only appears if the firstpage pagestyle is selected.

SEE ALSO Letters Declarations

\thanks (326)

```
\thanks{text}
```

The \thanks command produces a footnote to the title.

SEE ALSO Page_Styles \maketitle

thebibliography (327)

```
\begin{thebibliography}{widest-label}
\bibitem[label]{cite_key}
.
.
.
\end{thebibliography}
```

The thebibliography environment produces a bibliography or reference list. In the article style, this reference list is labeled "References"; in the report style, it is labeled "Bibliography".

- widest-label: Text that, when printed, is approximately as wide as the widest item label produces by the \bibitem commands.

SEE ALSO Environments

theorem (328)

```
\begin{theorem}
  theorem text
\end{theorem}
```

The theorem environment produces "Theorem x" in boldface followed by your theorem text.

SEE ALSO Environments

\thispagestyle (329)

```
\thispagestyle{option}
```

The \thispagestyle command works in the same manner as the \pagestyle command except that it changes the style for the current page only.

SEE ALSO Page_Styles

\tiny (330)

Smallest of 10 typefaces available. All fonts may not be available in this size.

SEE ALSO Typefaces Sizes

\title (331)

```
\title{text}
```

The \title command declares text to be the title. Use \\ to tell LaTeX where to start a new line in a long title.

SEE ALSO Page_Styles \maketitle

titlepage (332)

```
\begin{titlepage}
  text
\end{titlepage}
```

The titlepage environment creates a title page, i.e., a page with no printed page number or heading. It also causes the following page to be numbered page one. Formatting the title page is left to you. The \today command comes in handy for title pages.

SEE ALSO Environments

\tt (333)

Typewriter typeface.

SEE ALSO Typefaces Styles

\twocolumn (334)

The \twocolumn declaration starts a new page and produces two-column output.

SEE ALSO Document_Styles

\typeout (335)

\typeout{msg}

Prints msg on the terminal and in the log file. Commands in msg that are defined with \newcommand or \renewcommand are replaced by their definitions before being printed.

LaTeX's usual rules for treating multiple spaces as a single space and ignoring spaces after a command name apply to msg. A \space command in msg causes a single space to be printed.

SEE ALSO Terminal_Input_and_Output

\typein (336)

\typein[cmd]{msg}

Prints msg on the terminal and causes LaTeX to stop and wait for you to type a line of input, ending with return. If the cmd argument is missing, the typed input is processed as if it had been included in the input file in place of the \typein command. If the cmd argument is present, it must be a command name. This command name is then defined or redefined to be the typed input.

SEE ALSO Terminal_Input_and_Output

\underbrace (337)

\underbrace{text}

The \underbrace command generates text with a brace underneath.

SEE ALSO Math_Formulas Math_Miscellany

\underline (338)

```
\underline{text}
```

The \underline command causes the argument text to be underlined. This command can also be used in paragraph and LR modes.

SEE ALSO Math_Formulas Math_Miscellany

\usebox (339)

```
\usebox{cmd}
```

Prints the box most recently saved in bin cmd by a \savebox command.

SEE ALSO Spaces_and_Boxes

\usecounter (340)

\usecounter {counter}

The \usecounter command is used in the second argument of the list environment to allow the counter specified to be used to number the list items.

SEE ALSO <u>Counters</u>

\value (341)

```
\value {counter}
```

The \value command produces the value of the counter named in the mandatory argument. It can be used where LaTeX expects an integer or number, such as the second argument of a \setcounter or \addtocounter command, or in

```
\hspace{\value{foo}\parindent}
```

It is useful for doing arithmetic with counters.

SEE ALSO Counters

\vdots (342)

The \vdots command produces a vertical ellipsis.

SEE ALSO Math_Formulas Math_Miscellany

\vector (343)

```
\vector(x slope,y slope){length}
```

The \vector command draws a line with an arrow of the specified length and slope. The x and y values must lie between -4 and +4, inclusive.

SEE ALSO Environments picture

\verb (344)

\verb char literal_text char \verb*char literal_text char

Typesets literal_text exactly as typed, including special characters and spaces, using a typewriter (\tt) type style. There may be no space between \verb or \verb* and char (space is shown here only for clarity). The *-form differs only in that spaces are printed.

SEE ALSO Environments verbatim

verbatim (345)

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

The verbatim environment is a paragraph-making environment that gets LaTeX to print exactly what you type in. It turns LaTeX into a typewriter with carriage returns and blanks having the same effect that they would on a typewriter.

SEE ALSO Environments

verse (346)

```
\begin{verse}
  text
\end{verse}
```

The verse environment is designed for poetry, though you may find other uses for it.

SEE ALSO Environments

\vfill (347)

The \vfill fill command produces a rubber length which can stretch or shrink vertically.

SEE ALSO Spaces_and_Boxes

\vline (348)

The \vline command will draw a vertical line extending the full height and depth of its row. An \hfill command can be used to move the line to the edge of the column. It can also be used in an @-expression.

SEE ALSO Environments tabular

\vspace (349)

```
\vspace[*]{length}
```

The \vspace command adds vertical space. The length of the space can be expressed in any terms that LaTeX understands, i.e., points, inches, etc. You can add negative as well as positive space with an \vspace command.

LaTeX removes vertical space that comes at the end of a page. If you don't want LaTeX to remove this space, include the optional * argument. Then the space is never removed.

SEE ALSO Spaces_and_Boxes