

Formatting Texts I

Although L^AT_EX has numerous predefined macros for automatic and uniform formatting of a document without any mistake¹, many L^AT_EX users commit mistake by attempting to format a document manually. Manual formatting includes inserting vertical or horizontal spacing with `\bigskip`, `\vskip`, `\vfill`, etc., setting sectional headings via explicit font instructions, such as `\centerline{\bf Section 1: Introduction}` or `{\bf Theorem 4:}`, etc. Another mistake may be committed in keeping track of manually set sectional units. Various options for automatic formatting of a document are discussed here.

3.1 Sectional Units

Various sectional units, like chapters and sections, are generated using the `\chapter{}`, `\section{}`, `\subsection{}`, `\subsubsection{}`, `\paragraph{}` and `\subparagraph{}` commands, whose argument is the heading or title of a sectional unit, e.g., the current section of this book is written as `\section{Sectional Units}`. The sectional unit commands work in order and hence they should be nested properly, i.e., a `\subsection{}` command should follow a `\section{}` command or a `\subparagraph{}` command should follow a `\paragraph{}` command. L^AT_EX assigns three-tier serial numbers to chapters, sections, subsections, and subsubsections (paragraphs and subparagraphs are not numbered). These numbering system is shown in Fig. 3.1.

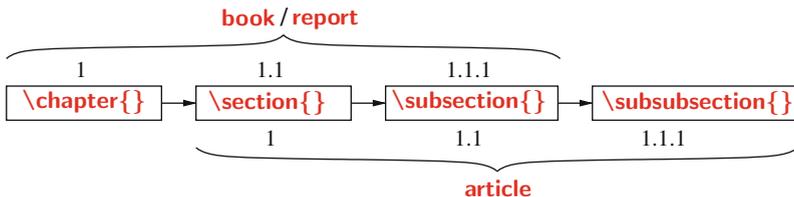


Fig. 3.1 Default three-tier numbering of sectional units

¹L^AT_EX has numerous predefined macros for automatic and uniform formatting without any mistake.

In the case of document-class **report** or **book**, which is composed of chapters, numbering is possible only to the `\chapter{}`, `\section{}`, and `\subsection{}` commands. Even if a `\subsubsection{}` command is used, it will not be numbered in the document-class **report** and **book**. A chapter is numbered by a whole number preceded by the label-word ‘Chapter’ and followed by its heading. A chapter may contain a number of sections, which are designated as 1.2, 3.5, etc., where the second number is the serial number of a section and the first number is the serial number of the chapter in which the section belongs. Similarly, the third tier of serial numbers are assigned to subsections, which are numbered as 2.5.3, 6.2.2, etc. On the other hand, since the document-class **article** is composed of sections and paragraphs only (it does not support the `\chapter{}` command), the three-tier numbering is assigned to the `\section{}`, `\subsection{}` and `\subsubsection{}` commands, whose numbering is similar with the numbering of `\chapter{}`, `\section{}` and `\subsection{}` commands in the document-class **report** and **book** (however, the first numbering is not preceded by any label-word as done by ‘Chapter’ in **report** and **book**). Note that the document-class **letter** does not support any sectional unit.

In some cases, the numbering of a sectional unit may need to be omitted (e.g., the *Preface* of a book is usually written under the `\chapter{}` command, but it is not numbered). L^AT_EX takes care of it through its `\chapter*{}`, `\section*{}`, `\subsection*{}` and `\subsubsection*{}` commands, which are not numbered because of their starred (*) forms.

3.2 Labeling and Referring Numbered Items

Like to sectional units addressed in §3.1, L^AT_EX assigns serial numbers to many environments or elements of an environment (e.g., **table**, **figure**, **equation**, or **item** as discussed in following Hours). This default numbering system eliminates the possibility of committing any mistake as may happen in manual numbering. Moreover, L^AT_EX allows to label a numbered item by a unique *reference key*, which can be used to refer the item in any part within the same document (unnumbered items, say `\paragraph{}`, cannot be referred in this way). The labeling and referring of an item are performed through `\label{rkey}` and `\ref{rkey}` respectively, where *rkey* is the assigned unique reference key of the item².

An example of labeling and referring an item is shown in Table 3.1 on the next page. The `\section{}` commands are followed by the `\label{}` commands containing the reference keys of ‘sec:cg’ and ‘sec-ex’ respectively (two different types of reference keys are considered to show that labeling can be done in any fashion, but without any blank space in between the texts of a reference key). In the body of the second `\section{}`, the first `\section{}` is referred through the `\ref{}` command using

²A numbered item can be labeled using `\label{rkey}` and then it can be referred using `\ref{rkey}` anywhere within the same document, where *rkey* is the assigned unique reference key of the item.

Table 3.1 Labeling and referring numbered items

L ^A T _E X input	Output
<pre>\section{Centre of gravity}\label{sec:cg} This is the point though which the resultant of the gravitational forces of all elemental weights of a body acts. % \section{Centre of mass}\label{sec-ex} The definition of the centre of gravity is given in Section~\vref{sec:cg} ...</pre>	<p>3.2 Centre of gravity</p> <p>This is the point though which the resultant of the gravitational forces of all elemental weights of a body acts.</p> <p>3.3 Centre of mass</p> <p>The definition of the centre of gravity is given in Section 3.2 ...</p>

its reference key as the argument of the command. As a result, the first section is referred in the output automatically by its serial number (not by its reference key), thus eliminates any possibility of referring an item by a wrong number.

Similar to the `\ref{}` command, there exists some commands for printing the page number where a numbered item appears. Such commands include `\pageref{rkey}`, `\vref{rkey}` and `\vpageref{rkey}`³.

- ▷ The `\pageref{}` command prints the page number where its argument is labeled, e.g., ‘CG is defined on page~`\pageref{sec:cg}`’ will print ‘CG is defined on page 17’. The `\pageref{}` command may not be preferred, particularly when labeling and referring of an item appear on the same page, as the command will print the serial number of a page on that page itself as done in the above example. Instead of generating the page number, some other texts, such as ‘on this page’ may suit better in this situation. Such facilities are provided in the `\vref{}` and `\vpageref{}` commands, which are defined in the `varioref` package.
- ▷ The `\vref{}` command usually executes a pair of `\ref{}` and `\pageref{}` commands.
 - It executes only the `\ref{}` command when labeling and referring of an item appear on the same page.
 - In addition to executing the `\ref{}` command, the `\vref{}` command also generates strings like ‘on the facing page’, ‘on the preceding page’ or ‘on the next page’ when the page numbers of labeling and referring of an item differ by one.
 - When the difference in page numbers of labeling and referring of an item is more than one, the `\vref{}` command executes both the `\ref{}` and `\pageref{}` commands. For example, ‘CG is defined in `\S\vref{sec:cg}`’ will print ‘CG is defined in §3.2’, while ‘quoted texts are discussed in `\S\vref{sec:quote}`’ will print ‘quoted texts are discussed in §3.4 on the next page’, or ‘text-mode fonts are discussed in `\S\vref{sec:text-fonts}`’ will print ‘text-mode fonts are discussed in §2.1 on page 9’.

³The `\pageref{rkey}`, `\vref{rkey}` and `\vpageref{rkey}` commands may be used in a similar way to `\ref{}` for referring the page number of an item, whose assigned unique reference key is `rkey`.

- ▷ If the `\ref{}` command is to be ignored, but strings similar to that produced by the `\vref{}` command are required, the `\vpageref{}` command may be used. For example, ‘CG is defined `\vpageref{sec:cg}`’ will print ‘CG is defined on the preceding page’. Two more controls can be obtained through two optional arguments to the `\vpageref{}` command as `\vpageref[asame][adiffer_]{rkey}`, where `asame` specifies the texts to be printed when labeling and referring appear on the same page, while `adiffer` specifies the texts if they fall on different pages. For example, ‘the `\vpageref[above definition][definition_]{sec:cg}` is for the center of gravity’ will produce ‘the definition on the preceding page is for the center of gravity’. Notice that a blank space (`_`) is maintained after `adiffer` in `\vpageref[asame][adiffer_]{rkey}`, i.e., after the string of the second optional argument, for maintaining a gap between the string and the page number (this is not required for the first argument, in which case no page number is generated).

3.3 Texts Alignment

By default L^AT_EX prints texts with both side aligned. Other types of alignment can be obtained through the `flushleft`, `flushright` and `center` environments, which print texts left, right, and center aligned, respectively. Some applications of these environments are shown in Table 3.2. Such alignments can also be produced using the corresponding declarations of these environments, which are `\raggedright`, `\raggedleft` and `\centering`, respectively.

Table 3.2 User-defined alignments of texts

L ^A T _E X input	Output
<code>\begin{flushleft}</code> L ^A T _E X prints texts with both side aligned. Left aligned texts can be produced through the ‘flushleft’ environment. <code>\end{flushleft}</code>	L ^A T _E X prints texts with both side aligned. Left aligned texts can be produced through the ‘flushleft’ environment.
<code>\begin{flushright}</code> L ^A T _E X prints texts with both side aligned. Right aligned texts can ... <code>\end{flushright}</code>	L ^A T _E X prints texts with both side aligned. Right aligned texts can be produced through the ‘flushright’ environment.
<code>\begin{center}</code> L ^A T _E X prints texts with both side aligned. Center aligned texts can ... <code>\end{center}</code>	L ^A T _E X prints texts with both side aligned. Center aligned texts can be produced through the ‘center’ environment.

3.4 Quoted Texts

Often some texts are required to be quoted within single or double quotation marks, which are usually done using (‘) and (”) symbols (generally both appear on the same button of a keyboard). However, L^AT_EX suggests to use (‘) as the left-hand

single quote and (') as the right-hand single quote (´ generally appears in a keyboard on the same button with ~). For double quotation mark, the single quote symbols may be used twice. For example, `'single-quote'` will produce ‘single-quote’, while `''double-quote''` will produce “double-quote”.

Sometime an existing statement may need to be quoted without any change, which is usually done in a narrowed width of a page. L^AT_EX provides the **quote** and **quotation** environments for that purpose. Both the environments are used in the same way, but the **quote** is usually for a short display, while the **quotation** for quoting more than one paragraph. Quoted statements can also be printed in a different line spacing through the **spacing** environment. An application of the **quotation** and **spacing** environments is shown in Table 3.3, where the mandatory argument of the **spacing** environment (1.2 is taken here) specifies the line spacing to be maintained within the

Table 3.3 Quoted texts in a narrowed width and specified line spacing

L ^A T _E X input	Output
<code>\LaTeX</code> prints texts with both side aligned, covering the specified width of a page. <code>\begin{quotation}</code> <code>\begin{spacing}{1.2}</code> Quoted statements are also printed with both side aligned, but in a narrowed width. <code>\begin{flushright}</code> <code>{lit - Anonymous}</code> <code>\end{flushright}</code> <code>\end{spacing}</code> <code>\end{quotation}</code> The 'quotation' environment is used for printing quoted statements in a narrowed width.	L ^A T _E X prints texts with both side aligned, covering the specified width of a page. Quoted statements are also printed with both side aligned, but in a narrowed width. <p style="text-align: right;"><i>- Anonymous</i></p> The 'quotation' environment is used for printing quoted statements in a narrowed width.

environment. If single line spacing is required, instead of the **spacing** environment, the **singlespace** environment may also be used without any argument. Both the **spacing** and **singlespace** environments are supported by the **setspace** package. Notice the nested environments in Table 3.3, where three environments, **quotation**, **spacing** and **flushright**, are nested. It is permitted in L^AT_EX to create such nested environments.

3.5 New Lines and Paragraphs

L^AT_EX does not respond to a new line or paragraph set manually by pressing the enter button of the keyboard. Unless specified commands are used, L^AT_EX considers everything in a single line and single paragraph.

3.5.1 Creating New Lines

The direct command for creating a new line is **\newline**. The texts preceded by a **\newline** command are printed in a new line. A new line can also be created using a line break command (**\linebreak**, **\l**, **\lll**, or one or more blank lines) at the end of

the previous line⁴. Some applications of these commands, using the same sentence, are shown in Table 3.4. To easily distinguish the differences of the commands, the

Table 3.4 Creating new lines

L^AT_EX input	Output
(1) There are certain key issues to attract investors, which need to be addressed. \newline	(1) There are certain key issues to attract investors, which need to be addressed. (2) There are certain key issues to attract investors, which need to be addressed.
(2) There are certain key issues to attract investors, which need to be addressed. \vlinebreak	(3) There are certain key issues to attract investors, which need to be addressed.
(3) There are certain key issues to attract investors, which need to be addressed. \	(4) There are certain key issues to attract investors, which need to be addressed.
(4) There are certain key issues to attract investors, which need to be addressed. \	(5) There are certain key issues to attract investors, which need to be addressed.
(5) There are certain key issues to attract investors, which need to be addressed. \	(6) There are certain key issues to attract investors, which need to be addressed.
(6) There are certain key issues to attract investors, which need to be addressed.	

sentence is numbered in each case. The effects of **\vlinebreak** in (2), **** in (4), and a blank line prior to (6), are most noticeable. The **\vlinebreak** command forces the texts of the last line to cover the entire page width. The **** command not only prints the following texts in a new line, but creates a blank line also. On the other hand, a blank line in the input file creates a new line with indentation (some blank space) at the beginning of the line (same effect would be resulted even if more than one blank line were left).

Some extra vertical space above the next new line can be specified in `[]` after the **** command at the end of the previous line, e.g., **\[2mm]** will create an extra vertical space of 2 mm above the next line. Further, the ***** or ***[]** command may be used for line breaking, but preventing page breaking at that point.

3.5.2 Creating New Paragraphs

Although a new paragraph can be started manually by creating a new line as discussed in §3.5.1, L^AT_EX suggests to use its predefined macros for creating uniform paragraphs. The direct command for creating a new paragraph is **\par**. On the other hand, the **\paragraph{}** and **\subparagraph{}** commands can also be used for creating new paragraphs with the arguments of the commands as the headings of the paragraphs. Table 3.5 on the facing page shows some applications of these commands, along with three more commands **\parindent**, **\noindent** and **\parskip**. The first **\par** command prints the texts in (1) in a new paragraph with default indentation. The indentation size for paragraph (2) is increased to 8 mm through the **\parindent = 8mm** command, while the indentation for paragraph (3) is skipped using the **\noindent**

⁴L^AT_EX does not create a new line set manually by pressing the `enter` button of the keyboard, rather the same can be obtained by inserting preferably **\newline** or **** at the end of the previous line.

Table 3.5 Creating new paragraphs

L ^A T _E X input	Output
<pre> \par (1) There are certain key issues to attract investors, which need to be addressed. % \par \parindent = 8mm (2) There are certain key issues to attract investors, which need to be addressed. % \par \noindent \parskip 2mm (3) There are certain key issues to attract investors, which need to be addressed. \par (4) There are certain key issues to attract investors, which need to be addressed. % \paragraph{(5) Investing policies:} There are certain key issues to attract investors, which need to be addressed. % \subparagraph{(6) Investing policies:} There are certain key issues to attract investors, which need to be addressed. </pre>	<p>(1) There are certain key issues to attract investors, which need to be addressed.</p> <p>(2) There are certain key issues to attract investors, which need to be addressed.</p> <p>(3) There are certain key issues to attract investors, which need to be addressed.</p> <p>(4) There are certain key issues to attract investors, which need to be addressed.</p> <p>(5) Investing policies: There are certain key issues to attract investors, which need to be addressed.</p> <p>(6) Investing policies: There are certain key issues to attract investors, which need to be addressed.</p>

command (indentation can be reactivated using the `\indent` command). Further, a vertical space of 2 mm is created above paragraph (3) using the `'\parskip 2mm'` command. Note that the effects of the `\parindent` and `\parskip` commands are global, which will act on all the paragraphs preceded by them⁵. These can be viewed in paragraphs (2), (3), and (4). Although the `\noindent` command is used before paragraph (3), paragraph (4) is indented by 8 mm following the `\parindent = 8mm` command used before paragraph (2). On the other hand, the `\paragraph{}` command prints its argument as the heading of paragraph (5). Moreover, the `\paragraph{}` command starts paragraph (5) without any indentation and also separates it from paragraph (4) by some predefined vertical blank space. The effect of the `\subparagraph{}` command, shown in paragraph (6), is similar with that of the `\paragraph{}` command, the only difference is that it maintains some indentation at the beginning of the paragraph.

3.6 Creating and Filling Blank Space

Excess blank spaces, created by pressing the `spacebar` or `tab` button of the keyboard, are just ignored in L^AT_EX, i.e., a sequence of blank spaces is treated as a single one only (similar to converting a sequence of blank lines into a single new line as discussed in §3.5.1). L^AT_EX provides its own commands for creating a blank space of a specified size, both in horizontal and vertical directions⁶, which are given in

⁵The `\parindent` and `\parskip` commands are generally used in the preamble to make their effects globally in all the paragraphs of a document.

⁶Excess blank spaces cannot be created by pressing the `spacebar` or `tab` button of the keyboard, rather L^AT_EX provides some commands for creating blank spaces of specified sizes both in horizontal and vertical directions.

Table 3.6 Creating blank spaces

Command	Package	Function	Application	
<code>\quad</code>	—	Predefined horizontal single space	<code>x\quad\y</code>	x y
<code>\qquad</code>	—	Predefined horizontal double space	<code>x\qquad\y</code>	x y y
<code>\,</code> or <code>\thinspace</code>	—	Predefined horizontal thin space	<code>x\,\y</code>	x y
<code>\;</code> or <code>\medspace</code>	amsmath	Predefined horizontal medium space	<code>x\;\y</code>	x y
<code>\;</code> or <code>\thickspace</code>	amsmath	Predefined horizontal thick space	<code>x\;\y</code>	x y
<code>\!</code>	amsmath	Predefined horizontal negative thin space	<code>x!\y</code>	xy
<code>\!\!</code>	amsmath	Predefined horizontal negative medium space	<code>x!\!\y</code>	xy
<code>\!\!\!</code>	amsmath	Predefined horizontal negative thick space	<code>x!\!\!\y</code>	xy
<code>\bigskip</code>	—	Predefined vertical space	refer Table 3.7	
<code>\vskip</code>	—	User defined vertical space	refer Table 3.7	
<code>\vspace{}</code>	—	User defined vertical space	refer Table 3.7	
<code>\hspace{}</code>	—	User defined horizontal space	refer Table 3.7	
<code>\vfill</code>	—	Vertical space to fill up a page	—	
<code>\hfill</code>	—	Horizontal space to fill up a line	refer Table 3.7	

Table 3.6 (commands of the first part are applicable in both text-mode and math-mode). Note in Table 3.6 that there should not be any blank space on any side of `\,`, `\;`, `\!`, `\!\!`, and `\!\!\!` commands; while `\quad` and `\qquad` commands must be followed by a blank space (a command, ended by an alphabet and followed by another alphabet, must be followed by a blank space in order to avoid the formation of a different command which even may not exist). The need of a blank space after a command, ended by an alphabet and followed by another alphabet, can be avoided by writing the following alphabet or word in `{}`, e.g., `'x\quad{y}'` to produce the same output as that by `'x\quad\y'`. The commands ended by a punctuation can also be applied in this way, e.g., `'x\,{y}'` in place of `'x\,\y'`.

Some applications of the commands given in the second part of Table 3.6 are shown in Table 3.7, where the `\bigskip` command followed by the line break

Table 3.7 Applications of some blank space creating commands

L ^A T _E X input	Output
<code>\begin{center}</code> <code>\LaTeX</code> in 24 Hours <code>\bigskip\</code> A Practical Guide for Scientific Writing <code>\end{center}</code>	L ^A T _E X in 24 Hours A Practical Guide for Scientific Writing
<code>\begin{center}</code> <code>\LaTeX</code> in 24 Hours <code>\vskip 8mm</code> A Practical Guide for Scientific Writing <code>\end{center}</code>	L ^A T _E X in 24 Hours A Practical Guide for Scientific Writing
<code>\begin{center}</code> <code>\LaTeX</code> in 24 Hours <code>\vspace{8mm}\</code> A Practical Guide for Scientific Writing <code>\end{center}</code>	L ^A T _E X in 24 Hours A Practical Guide for Scientific Writing
Writing language: <code>\hspace{5mm}</code> English.	Writing language: English.
Marks: 100 <code>\hfill</code> Time: 3 Hours.	Marks: 100 Time: 3 Hours.

command `\` (or any other new line command addressed in §3.5.1) creates a predefined vertical positive space between two lines. In contrary, the `\vskip` or `\vspace{}` command creates a user-defined vertical space between two lines or paragraphs. Similarly, the `\hspace{}` command produces a user-defined horizontal space between two words. Note that the argument of the `\vskip` command is written with a blank space, while those of the `\vspace{}` and `\hspace{}` commands are written in `{}`. Moreover, the `\vspace{}` command is to be followed by the line break command `\`, otherwise some texts from the next line may be shifted to the previous line to fill up its blank space, if any.

The unit of the rigid lengths in the arguments of the `\vskip`, `\vspace{}` and `\hspace{}` commands can be any one of `mm` (millimeter), `cm` (centimeter), `in` (inch), `pt` (point), `em` (width of M) and `ex` (width of x). The unit `em` is usually preferred for horizontal lengths, while `ex` for vertical lengths⁷. Apart from these units, a length can also be taken as a fraction of `\textheight` (height of texts on a page), `\textwidth` (width of texts on a page) or `\linewidth` (width of a column), e.g., `0.2\textheight` for a vertical space of 20% of `\textheight` or `0.3\linewidth` for a horizontal space of 30% of `\linewidth`.

L^AT_EX removes blank spaces from the beginning and end of a line as well as a page, for which the `\hspace{}` and `\vspace{}` commands may not have any effect in such places, i.e., at the beginning and end of a line or a page. If so, the commands may be preceded by a `~`. Alternatively, the starred forms of the commands, i.e., the `\hspace*{}` and `\vspace*{}` commands, may be used for producing blank spaces at the beginning or end of a line and a page, respectively. Similarly, instead of the `\hfill`⁸ and `\vfill` commands (abbreviations of the `\hspace{\fill}` and `\vspace{\fill}` commands respectively), the `\hspace*{\fill}` or `\vspace*{\fill}` command is to be used for filling a line (or a page) by creating necessary horizontal (or vertical) space at the beginning or end of the line (or the page).

Note that a positive valued length argument to the `\vskip`, `\vspace{}` or `\hspace{}` command will create a positive blank space, while a negative valued length argument will reduce space by that amount, e.g., `\hspace{-7mm}` will reduce the space between two words by 7 mm (words may get overlapped also, refer foot note 4 on page 12). The `\vskip` or `\vspace{}` command with a negative value may be used (particularly before and after equations, listed items, tables, figures, etc.) to shrink vertical space enabling to accommodate more amount of materials on a page (§5.1.3 on page 39 describes other techniques for the same purpose).

Further, note that the space created by `\hspace{}` may be limited to the available space in a line even for a larger input value in `\hspace{}`, while the space created by `\hspace*{}` with a larger input value (either between two words or at one side of a line) will always continue beyond the page width or even to the next line.

⁷L^AT_EX accepted units for rigid lengths include `mm` (millimeter), `cm` (centimeter), `in` (inch), `pt` (point), `em` (width of M) and `ex` (width of x), where `em` is usually preferred for horizontal lengths and `ex` for vertical lengths.

⁸The `\dotfill` and `\hrulefill` commands are similar to the `\hfill` command, except they fill the gap by dots and horizontal lines respectively.

3.7 Producing Dashes Within Texts

L^AT_EX provides dashes of three different lengths: -, –, and —, which are produced by -, -- and ---, respectively. Out of these dashes, the shortest one is used between inter-related words, the medium one is used to indicate a range, while the longest one is used to show the extension of an expression. Example of each of these three dashes is shown in Table 3.8.

Table 3.8 Dashes of different lengths

L ^A T _E X input	Output
Inter-related	Inter-related
May--August	May–August
Weather condition --- like clear sky	Weather condition — like clear sky

3.8 Preventing Line Break*

Because of the full alignment in a predefined page width, a long statement may need to be split between two words, or even a single word into two parts, continuing the rest in the next line. However, sometime the splitting of a statement between two particular words may need to be prevented, e.g., it is not desirable to split ‘Dilip Datta’ into ‘Dilip’ and ‘Datta’, and then to place them in two lines. Such phrases can be forced to be printed in the same line by using the `~` or `\` command between the words without any gap, e.g., ‘Dilip~Datta’ or ‘10\inch’. The `\nolinebreak` is another command for preventing line breaking at a particular point, e.g., ‘Dilip\nolinebreak\ Datta’, where `\` means a blank space (the `\nolinebreak` command is to be followed by `\`, refer § 1.5.1 on page 5 for detail).

The `~`, `\`, and `\nolinebreak` commands, however, are always not appropriate, particularly when a word is also to be prevented from splitting. If these commands are set manually between two letters of a word, some undesirable gap may get generated within the word. In that case, the appropriate option is to use the `\mbox{}` command, which prevents its argument from splitting in two lines⁹. For example, ‘NSGA-II-UCTO’ can be forced to be printed in the same line using `\mbox{NSGA-II-UCTO}`. Note that the `~`, `\`, `\nolinebreak` and `\mbox{}` commands may sometime print texts beyond the page width. So, it would be a good practice to check the output after the use of such commands.

3.9 Adjusting Blank Space After a Period Mark*

Generally a sentence is ended by a period (full-stop) mark (.) and the next sentence is started with some prespecified gap after that period mark. Sometime a period mark is to be used within a sentence also, e.g., ‘Dr. Datta’, which L^AT_EX will interpret

⁹The `\mbox{}` command may be used to prevent its argument from splitting in two lines.

as the end of a sentence and hence will print ‘Dr.’ and ‘Datta’ with a big gap between them. Such a big gap can be reduced by commands like `\,` and `\~` without any blank space on any side (reduction of space is minimum under the `\,` command), or `_` (`_` means a blank space). If a period mark is followed by a right parenthesis or a right quote, the `\` command is to be used after the parenthesis or the quote, otherwise the combination of `\` and parenthesis/quote will be treated as a different command. On the other hand, when a sentence is ended by an uppercase letter, the next sentence is started with a smaller gap. For maintaining a proper gap in this case, the `\@` symbol may be used in between the said uppercase letter and the following period mark. Some applications of these commands are shown in Table 3.9.

Table 3.9 Maintaining proper gap after a period (full-stop) mark

L^AT_EX input	Output
Dr. Datta	Dr. Datta
Dr.\,Datta	Dr.Datta
Dr.\~Datta	Dr. Datta
Dr._Datta	Dr. Datta
classes (article, letter, etc.) may be used	classes (article, letter, etc.) may be used
classes (article, letter, etc.)\ may be used	classes (article, letter, etc.) may be used
made in USA. United ...	made in USA. United ...
made in USA\@. United ...	made in USA. United ...

3.10 Hyphenating a Word*

If the last word of a line goes beyond the page width, it is generally full-aligned forcibly by breaking the word into two parts with a hyphen to the first part and then putting the second part in the next line. However, in many cases L^AT_EX may continue such a word beyond the page width, or break it in an unpleasant place. Proper word-breaking in that case may be specified manually by inserting the `\-` command in between a word for forcibly hyphenating at that point, e.g., ‘kind\-ness’ to print ‘kind-’ at the end of the current line and ‘ness’ at the starting of the next line. If L^AT_EX fails to provide proper hyphenation in a paragraph, particularly in the case of a non-dictionary word, it can be put in the **sloppypar** environment also, e.g., `\begin{sloppypar}Maneswarirabha\end{sloppypar}`. However, it would be a good practice to check the output and subsequently to do some manual adjustment, if required.