

Brevity is the soul of wit : How \LaTeX can help.

*

S. Parthasarathy[†]
drpartha@gmail.com

Abstract This essay is about using “lists” in \LaTeX . Lists are very useful, in presenting material in a crisp and compact form. This makes technical documents, less verbose, and easier to follow. The author hopes that this paper will make \LaTeX enjoyable for more people.

This paper was published in Practex Journal Vol. 2007-4, Dec. 2007

1 Preamble

“Brevity is the soul of wit”, said a great author once. Technical documents have to be “to the point”, and well focused. Lists provide an excellent artefact for technical writers. Lists help you avoid long, verbose paragraphs. The points in a list, stand out on their own, prominently. Cross referencing points is easy with lists. They make documents, much more easy to follow. Thus, mastering lists, is an essential activity for anyone who wishes to succeed in technical writing. \LaTeX provides you three possible list environments (presented below, as a description list):

Enumerate environment, using `\begin{enumerate} ... \end{enumerate}` – for producing a “numbered” list

*Ask the author for a copy of the \LaTeX source of this document (brevity0.tex) . You can try out all the examples yourself, by hacking the source.

[†]Algologic Research and Solutions, 78 Sancharpuri Colony, Bowenpally, Secunderabad 500 011, India WWW-URL – <http://algolog.tripod.com/nupartha.htm>

Itemize environment, using `\begin{itemize} ... \end{itemize}` – for producing a “bulleted” list

Description environment, using `\begin{description} ... \end{description}` – for a list of term/description pairs, like the entries you will find in a traditional dictionary. This current list, uses a description environment. You can see, it is so very clean, and easy to follow.

And, of course you can cook up your own list environment too.

A few simple facts (presented below, as a numbered list):

1. Each list must have at least one item.
2. You can embed lists inside lists, creating a hierarchical structure (nested lists, or multi-level lists).
3. You can mix the above three basic types of lists inside hierarchically structured lists (multi-level, multi-mode lists).
4. You can number lists either in ascending order, or in descending order.
5. You can change the “look-and-feel” of lists
6. As a standard practice, you can go four levels deep, in nested lists. This is much more than necessary, for all practical purposes.

L^AT_EX offers a very rich choice of possibilities when you use lists (presented below, as a bulleted list):

- You can use nested lists – bullets within bullets, or numbers within numbers, or bullets within numbers, or numbers within bullets
- You can number in ascending order, or in descending order
- You can change the numbering style
- You can change the starting number for numbered lists
- You can change the bullet style
- You can combine the above options

We will see some examples, to illustrate the above facts. We will also see several nested `\itemize` and `\enumerate` lists now. This document will demonstrate different possibilities with nested `\itemize` and `\enumerate` lists.

2 How to work with nested lists – multi-level lists, multi-mode lists

2.1 Nested itemize – bullets within bullets

- Partha is a great guy
 - But no one knows that.
 - And, even if they know, they don't accept that
- Yet, Partha keeps saying this again and again

2.2 Nested enumerate – numbers within numbers

1. First first-level number
2. Second first level number
 - (a) First second level
 - (b) Second second level
3. Third first level

When you use multi-level lists, L^AT_EX numbers each level using a different numbering style : 2aiA (arabic, alpha, roman, Alpha). You can decide to change these numbering styles, by using the `\renewcommand` facility, like this :

```
\renewcommand{\theenumi}{Roman{enumi}}
```

2.3 Numbers within bullets

- First bullet
- second bullet
 1. First number
 2. Second number
- Bullet again

2.4 Bullets within numbers – why not ?

1. first number
2. second number
 - First bullet
 - Second bullet. Now we will go, one level more, for bullets
 - second level bullet
 - second second level bullet
3. third number – and go one level deeper for numbers
 - (a) second level – first item
 - (b) secondlevel – second item – this will have two levels of bullets
 - one bullet inside a number which was under another number
 - another level bullet
 - (c) and now the next second level number

3 L^AT_EX magic !

You can decide to get the items numbered in the descending order. Why on earth would you need to do that ? For instance, if you are preparing a list of items, arranged in the reverse chronological order, you would like the first item to carry the largest number since it was the last to be produced in the chronological order. Confusing ? Imagine having to introduce your children, youngest first, eldest last. Imagine you have four children. You would say “Meet xxx, my fourth child”, then “Meet yyyy, my third child” and so on. Or, imagine having to present your company. You want to start with the smallest employee, and end with the big boss. And you want to give the big boss – numero uno, although he is presented last of all. And, above all, you do not want to actually count the numbers yourself. You want L^AT_EX to do it for you automatically. The numbers should get adjusted automatically when you add or remove an element from the list. Got the idea ? We achieve this using the `revnum` package.

And, of course you can mix forward counting with reverse counting, like in the following example:

3. One

2. Two
 - (a) Un
 - (b) Deux
 - (c) Trois
1. Three
 - (c) Ein
 - (b) Zwei
 - (a) Drei

But, how does L^AT_EX know from where to start numbering ? Do you have to tell L^AT_EX how many items you have in the list , so it can start counting down from this maximum ? NO, a big NO. L^AT_EX can find out by itself. To do this, you will have to pass your document through L^AT_EX two times. In the first pass, L^AT_EX just makes a count and stores the maximum count, in the aux file. In the second pass, L^AT_EX will start counting down from this maximum limit which it found out during the first pass. That is pretty smart, isn't it ?

3.1 And some more L^AT_EX magic !

Actually, you can start numbering down, from any arbitrary maximum value, like this :

17. we can start from any maximum
16. and count down
15. like this

Actually, reverse numbering of lists can also be achieved using the etarenume package (etarenume is enumerate written backwards). They say, etarenume is smarter than revnum. It uses lesser memory.

```
\begin{etarenume}
\item hhh
```

```

\item jjj
\item kkk
\end{etaremune}

```

And, how about some personalised numbering scheme for your lists ?

1. First first-level item
2. Second first level item
- 2++a** First second level item
- 2++b** Second second level item
3. Third first level item

We used the `\renewcommand` to modify the definition of `\labelenumii`. I decided to have the second level displayed along with the number of the first level, and with some `++` and `**`, just for fun. Caution : You must undo this `\renewcommand` by doing a `\renewcommand` again `\renewcommand{\labelenumii}{\theenumii}`, so that the labels would look normal again, like this :

1. First first-level number
2. Second first level number
 - a First second level . (Try commenting out the `\renewcommand`, used just before the `\begin{enumerate}` above, and see the difference.)
 - b Second second level
3. Third first level

Can you customise the bullets too ? YES, of course. Just redefine the `\labelitem`, and you can get this:

- ◇ how is this ?
 - ◁ or this one ?
 - ◁ or this one ?
- ◇ do you like this ?

You can't ask for more proof about the power of \LaTeX , can you ?

4 Closing remarks

L^AT_EX can be used for producing very elegant looking documents. It gives you tools for making your documents, crisp and compact. The “list” facility of L^AT_EX is an important feature for creating documents with technical content. L^AT_EX also gives you many options for using lists, with flexibility and ease. You can personalise these lists, to reflect your own taste and preferences.

This paper, predictably, was made using L^AT_EX . It used the Kile front-end provided by Suse Linux. It uses the revnum package, downloaded from the CTAN site.

5 About the author

Parthasarathy teaches discrete mathematics, and preaches LaTeX and Linux, to undergraduate students of Computer Science, at Hyderabad, India. He also runs a specialised enterprise which uses Linux and L^AT_EX exclusively. His website (marked in the footnote, in the title page) will give more specific details about him.

* * * * *