Introduction to list structures in LaTeX

Thomas Thurnherr

Abstract

Lists are frequently used structures in documents as well as presentations. LaTeX distinguishes three types of lists: bulleted list, ordered list, and descriptive list. In this article, I introduce these three types of lists, describe basic manipulations, and provide information about packages that expand on standard list structures by adding extra flexibility.

1 Introduction

LaTeX distinguishes between three types of lists: bulleted list, ordered list, and descriptive list. The bulleted list, where the order of elements is not important, is called `itemize`. On the other hand, ordered lists are termed `enumerate`, as their elements are numbered. Lastly, `description` is a descriptive list, which generally describes words or phrases. Usage of the three list types is similar; they are implemented as environments and elements are added via the `\item` command.

```
\begin{list-type}
  \item text
  \item text
\end{list-type}
```

2 Bulleted/unordered list

A bulleted list is a list where the ordering of elements does not matter. The `itemize` environment creates an unordered list and elements are added with `item`. Here is an example:

```
\begin{itemize}
  \item A bulleted item
  \item Another bulleted item
  \item And another bulleted item
\end{itemize}
```

⇓

- A bulleted item
- Another bulleted item
- And another bulleted item

The default label for unordered lists is a bullet (•). I will show later (section 6) how to replace the label with another symbol.

3 Numbered/ordered list

The behavior of an ordered list is similar to the unordered list. The only difference is that the label is a number or letter from the alphabet, which is incremented for every element. By default, the label is an arabic number followed by a dot. Again, I will show later how to make changes to the way the label is typeset.

```
\begin{enumerate}
  \item An ordered item
  \item Another ordered item
  \item And another ordered item
\end{enumerate}
```

⇓

1. An ordered item
2. Another ordered item
3. And another ordered item

LaTeX uses a `counter` (called `enumi`) to keep track of the number of elements in an ordered list. Therefore, list items can be referenced within as well as outside the list. An example is given below to illustrate how to cross-reference a list element.

```
\begin{enumerate}
  \item An ordered item\label{itm:myList}
  \item A reference to item \ref{itm:myList}
\end{enumerate}
```

⇓

1. An ordered item
2. A reference to item 1

4 Descriptive lists

Unlike `itemize` and `enumerate`, a descriptive list, or `description`, does not have a label. Instead, a word or phrase is used, which is passed to `item` as an optional argument. This is shown in the following:

```
\begin{description}
  \item[First] A descriptive item
  \item[Second] Another descriptive item
  \item[Third] And another descriptive item
\end{description}
```

⇓

First A descriptive item
Second Another descriptive item
Third And another descriptive item

5 Nested lists

Lists can be nested by placing a list environment inside another list environment. Different list types may be combined, as illustrated in the example below. When the same list types are nested, LaTeX uses different, predefined labels for each level of nesting. The default maximum level of nesting for each type

of list is four. If more than four lists of the same type are nested, LaTeX throws the error: "Too deeply nested".

```
\begin{enumerate}
  \item An ordered list item
  \begin{enumerate}
    \item A nested ordered list item
    \begin{itemize}
      \item A nested unordered list item
    \end{itemize}
    \item Another nested ordered list item
  \end{enumerate}
  \item Another ordered list item
\end{enumerate}
```

⇓

1. An ordered list item
   (a) A nested ordered list item
       • A nested unordered list item
   (b) Another nested ordered list item
2. Another ordered list item

6 List manipulations

The appearance of a list is alterable. For example, one might want a different label from the default, or to add/remove space between items. There are several packages which implement macros for list manipulations. Some of the most commonly used packages are: `enumerate` [1], `enumitem` [2], and `mdwlist` [4].

Here, I will focus on `enumitem` as it provides the most comprehensive set of macros to manipulate list structures and comes with extensive documentation. The commands here assume that this package is loaded in the preamble:

```
\usepackage{enumitem}
```

6.1 Changing the label

The label can be changed using the optional environment parameter `label`. In the example below I change the label of an unordered list, which is a bullet by default, to a diamond. Some symbols frequently used as labels for unordered lists are given in table 1 (left column).

```
\begin{itemize}[label=$\diamond$]
  \item A diamond-labelled item
  \item Another diamond-labelled item
\end{itemize}
```

⇓

◇ A diamond-labelled item
◇ Another diamond-labelled item

Table 1: Label options for `itemize` and `enumerate`.

`itemize`		`enumerate`	
label	code	label	code
•	`$\bullet$`	1,2,...	`\arabic*`
–	`$-$`	i,ii,...	`\roman*`
·	`$\cdot$`	I,II,...	`\Roman*`
∗, ⋆	`$*$`, `$\star$`	a,b,...	`\alph*`
◇	`$\diamond$`	A,B,...	`\Alph*`

Analogous to an unordered list, the label may be changed in a numbered list. Again, available options are listed in table 1 (right column). In the example below, I change the default label (`arabic`) to `roman` labelling. Moreover, the number may be combined with parentheses or any punctuation symbol.

```
\begin{enumerate}[label=(\roman*)]
  \item A roman-numeralled item
  \item Another roman-numeralled item
\end{enumerate}
```

⇓

(i) A roman-numeralled item
(ii) Another roman-numeralled item

6.2 Resume numbering from previous ordered list

The `enumitem` package provides the option `resume`, which resumes numbering from the preceding ordered list in a new ordered list.

```
Ordered list:
\begin{enumerate}
  \item Ordered list item
\end{enumerate}
Resume previous ordered list:
\begin{enumerate}[resume]
  \item Resumed list item
\end{enumerate}
```

⇓

Ordered list:
1. Ordered list item

Resume previous ordered list:
2. Resumed list item

6.3 Vertical space

`enumitem` implements several parameters to control vertical spacing outside and inside list structures. The parameters are summarized below:

topsep Whitespace above and below list

partopsep Extra whitespace when list starts new paragraph
itemsep Spacing between elements in list
parsep Spacing between paragraphs in list
noitemsep Sets `itemsep` and `parsep` to `0pt`
nosep Removes vertical space completely

Except for the last two options, these parameters are used as key–value pairs, where the value specifies the amount of whitespace.

```
\begin{enumerate}[topsep=10pt]
  \item A numbered item
  \item Another numbered item
\end{enumerate}
```

⇓

1. A numbered item
2. Another numbered item

6.4 Horizontal space

The following parameters control horizontal spacing outside and inside list structures:

leftmargin Limits the list to the left
rightmargin Limits the list to the right
listparindent Paragraph indent within a list
labelsep Separation between label and body
itemindent Item indent (first line)

Both vertical and horizontal spacing may be controlled globally using the `\setlist` command. Changes can be applied to all list types simultaneously or restricted to a specific list type via an optional list type parameter. In the example below, the first command is limited to numbered lists, whereas the second command changes all three list types simultaneously.

```
\setlist[enumerate]{nosep}
\setlist{topsep=0pt, itemsep=0pt}
```

7 Inline list

The `paralist` package [5] provides macros for inline lists, where all elements of a list are displayed within the same paragraph. The package implements three alternative list environments called: `inparaitem`, `inparaenum`, and `inparadesc`. Their usage is similar to the standard list environments.

```
The paralist package implements environments
for inline lists. These are:
\begin{inparaenum}
  \item inparaitem,
  \item inparaenum, and
  \item inparadesc.
\end{inparaenum}
```

⇓

The paralist package implements environments for inline lists. These are: 1. inparaitem, 2. inparaenum, and 3. inparadesc.

8 Reverse numbered list

The `etaremune` package [3] implements an environment with the same name as the package to reverse the numbers of elements in an ordered list. Here is an example:

```
The three most popular movies in IMDB are,
starting from the third:
\begin{etaremune}
  \item The Godfather: Part II (1974)
  \item The Godfather (1972)
  \item The Shawshank Redemption (1994)
\end{etaremune}
```

⇓

The three most popular movies in IMDB are, starting from the third:

3. The Godfather: Part II (1974)
2. The Godfather (1972)
1. The Shawshank Redemption (1994)

References

[1] `enumerate` — enumerate with redefinable labels. `http://ctan.org/pkg/enumerate`. Accessed: 2015-09-14.

[2] `enumitem` — control layout of itemize, enumerate, description. `http://ctan.org/pkg/enumitem`. Accessed: 2015-09-14.

[3] `etaremune` — reverse-counting enumerate environment. `http://ctan.org/pkg/etaremune`. Accessed: 2015-09-14.

[4] `mdwlist` — miscellaneous list-related commands. `http://ctan.org/pkg/mdwlist`. Accessed: 2015-09-18.

[5] `paralist` — enumerate and itemize within paragraphs. `http://ctan.org/pkg/paralist`. Accessed: 2015-09-14.

◇ Thomas Thurnherr
`thomas.thurnherr (at) gmail dot com`
`http://texblog.org`