Overview of the bullcntr package[*]

G. Mezzetti
gustavo.mezzetti@istruzione.it

October 1, 2007

Abstract

We describe a LaTeX package which defines a new command, similar to `\fnsymbol`, for displaying the value of a counter: this command typesets a number of bullets equal to the value of the counter, in a nice arrangement. The value of the counter must lie between 1 and 9, inclusive.

We also present an ancillary package which defines a new environment, similar to `enumerate`, for creating lists of items numbered in this way.

Contents

[*] This document describes version 0.04 (2007/04/02). Copyright © 2007 by G. Mezzetti (see page 2 for details).

Copyright notice

This document has been produced by feeding to a LaTeX typesetting engine the file `bullcntr-man.dtx`. This file is part of a work named "bullcntr package", and may distributed and/or modified only as a part thereof. Subsection 1.3 on page 4 explains the conditions under which the bullcntr package may be distributed and/or modified.

Permission is granted to make printed copies of this document, by any means (*e.g.*, photocopying, direct printing from the electronic version, and so on).

Note, though, that distribution of this document alone is "distribution of only part of the Work" in the sense of the LaTeX Project Public License, and should be avoided: please distribute, in addition, the source file `bullcntr-man.dtx` together with all other files listed in `manifest.txt`, as explained as well in Subsection 1.3 and in the file `manifest.txt` itself.

Introduction

I can't remember exactly where the idea of this package arose from, but I am pretty sure that the first time I saw a list of items numbered with an increasing number of bullets was in certain handwritten notes about sheaves cohomology, may be a decade or so ago. Since then, I had repeatedly thought of writing a LaTeX package to implement this kind of numbering, but I had never resolved to actually do so, until very recently, when I suddenly decided to sit down and write the code! :-)

Regrettably, I had to discover a major bug in the macros I had dashed off in such a haste, which required extensive (and more careful!) redesign of the code. I took this opportunity to thoroughly reorganize the package, making it easier, for the user, to switch from one style of bullets to another, and also adding an ancillary package that simplifies the creation of lists numbered with bullets.

This document describes the bullcntr package version **0.04** (**2007/04/02**).

1 Installing the bullcntr package

As usual for LaTeX distributions, the bullcntr package is distributed in the form of a `.dtx` file, namely `bullcntr.dtx`, and an accompanying `.ins` file, namely `bullcntr.ins`.

1.1 Installation

To install the package, run LaTeX (or Plain TeX) once on the file `bullcntr.ins`. This will generate the following LaTeX input files:

```
bullcntr.sty
bullenum.sty
```

To finish the installation, move them into a LaTeX input directory. The above listing of the files you need to move is also displayed on the terminal at the end of the run of the file `bullcntr.ins`. The documentation of your TeX installation should tell you how to find the LaTeX input directory/ies, and probably also how to create new LaTeX input directories reserved to contain your private classes and packages. If your TeX installation offers you the chance of defining your private LaTeX input directories, I recommend you exploit this possibility and place the generated files into such a directory.

Please note that before using the bullcntr package you must read the license (see Subsection 1.3) to see whether its terms are acceptable for you, especially for what concerns the lack of any warranty; if they are not, don't use it.

1.2 Documentation

To produce the documentation, run LaTeX three times (for the table of contents to be correct) on the file `bullcntr.dtx`. This won't produce the index and the change history, however: for those, you have to run *MakeIndex* on the files `bullcntr.idx` and `bullcntr.glo`, produced during the last of the aforesaid three LaTeX runs, with the styles `gind.ist` and `gglo.ist`, respectively (these styles are part of the standard LaTeX distribution); then pass `bullcntr.dtx` through LaTeX twice more.

1.3 License

The bullcntr package is *not* in the public domain: its author, Gustavo Mezzetti, owns the copyright, and in general retains all the rights therein; but as a special exception, the author grants you the permissions indicated below.

1.3.1 Distribution and/or modification

The bullcntr package may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in

`http://www.latex-project.org/lppl.txt`

and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

The bullcntr package has the LPPL maintenance status "author-maintained".

The file `manifest.txt` included in the bullcntr package specifies what this package consists of; more precisely, it explains how the locutions "Work" and "Compiled Work", used in the LaTeX Project Public License, are to be interpreted in the case of this work.

1.3.2 Use

The use of the bullcntr package is unrestricted, provided that you accept the terms and conditions of the LaTeX Project Public License and of the following subsection for what concerns the absence of any warranty.

1.3.3 No warranty

There is absolutely no warranty for the bullcntr package. The Copyright Holder provides the bullcntr package "as is", without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the bullcntr package is with you. Should the bullcntr package prove defective, you assume the cost of all necessary servicing, repair, or correction.

In no event will The Copyright Holder, or any other party who may distribute and/or modify the bullcntr package as permitted by the LaTeX Project Public License, be liable to you for damages, including any general, special, incidental or consequential damages arising out of any use of the bullcntr package or out of inability to use it (including, but not limited to, loss of data, data being rendered inaccurate, or losses sustained by anyone as a result of any failure of the bullcntr package to operate with any other programs), even if The Copyright Holder or said other party has been advised of the possibility of such damages.

1.4 Your comments

If you have comments, suggestions, etc. about the bullcntr package, please let me know them: I'll be happy to read what you think. My e-mail address is indicated under the title of this document. Please include the *exact* phrase `bullcntr␣package` in the *subject* of your message; otherwise, it could be thrown away by a mail filter.

2 Using the bullcntr package

Version 0.04 of the bullcntr package is a bit more elaborate than its predecessor: in addition to the `\bullcntr` main command and to the hooks for the bullets, it now features options and predefined bullet styles.

2.1 Package invocation and options

You load the bullcntr package by means of the usual `\usepackage` declaration:

```
\usepackage⟨options⟩{bullcntr}
```

You may specify one of the following, mutually exclusive options, which affect the size (or the type) of the bullets used by `\bullcntr` command:

`largectrbull` (Relatively) “large” bullets are used for all values from 1 to 9. This has exactly the same effect as a `\largectrbull` global declaration (see Subsection 2.4).

`smallctrbull` The opposite: “small” bullets are used for all values from 1 to 9. This has exactly the same effect as a `\smallctrbull` global declaration (see Subsection 2.4).

`smartctrbull` (Relatively) “large” bullets are used for values from 1 through 5 (inclusive), while “small” ones are used for those greater than or equal to 6. This has exactly the same effect as a `\smartctrbull` global declaration (see Subsection 2.4).

`heartctrbull` Hearts are used (instead of bullets) for all values from 1 to 9. This has exactly the same effect as a `\heartctrbull` global declaration (see Subsection 2.4).

If you don’t specify any, the default option is `smartctrbull`.

2.2 The `\bullcntr` command

This is a command analogous to the `\fnsymbol` command (see the *LaTeXbook* [1, Subsection C.8.4]): you may use it directly in the text, as in

```
...the counter value displays as~\bullcntr{⟨ctr⟩}...
```

(where ⟨*ctr*⟩ is a counter name, defined with `\newcounter`), or you may redefine the `\the`⟨*ctr*⟩ command, as in

```
\renewcommand*{\the⟨ctr⟩}{\bullcntr{⟨ctr⟩}}
```

By way of example, in the case of a counter named `myctr`, you would write

```
The counter value displays as~\bullcntr{myctr}.
```

directly in the text, or the following to redefine the `\themyctr` command:

```
\renewcommand*{\themyctr}{\bullcntr{myctr}}
```

When `\value{⟨ctr⟩}` $= 0$, the command `\bullcntr{⟨ctr⟩}` silently typesets nothing (not even a space); but when `\value{⟨ctr⟩}` < 0 or `\value{⟨ctr⟩}` > 9, an error is generated.

2.3 Hooks for the bullets

The main task of the bullcntr package is to calculate the position of the "bullets" used to display the value of the given counter; but you are free to decide what characters are actually drawn in these positions: they need not necessarily be "bullets". Indeed, two "hooks" are provided in the package to define the actual characters employed, and the package simply invokes these hooks.

If the value of the counter being displayed is less than 6 (less than 6 bullets need to be drawn), the `\bullcntr` command invokes the `\counterlargebullet` macro to draw each of them. If you want, you should redefine this macro (using `\renewcommand`) so that it generates a symbol suitable for arrangements in which the "bullets" are relatively spaced out. For example,

```
\renewcommand*{\counterlargebullet}{\textbullet}
```

You don't *need* to redefine this command: a convenient default definition is already set by the bullcntr package itself.

When, on the other hand, the value of the counter being displayed is greater than or equal to 6 (6 or more bullets need to be drawn), the `\bullcntr` command calls the `\countersmallbullet` macro. It is intended that this macro typesets a character suitable for situations in which the "bullets" have to be grouped in a compact cluster; typically, this means that a smaller "bullet" is used than the one typeset by `\counterlargebullet`, although it is perfectly in order to use the same character in both situations. If you want (but you don't *need* to do this, since the bullcntr package already defines this macro) you may change the definition with `\renewcommand`, *e.g.*:

```
\renewcommand*{\countersmallbullet}{\textperiodcentered}
```

For both hooks, it is best to `\protect` fragile commands that appear in the code used in their (re)definition.

"Macho TeX programmers" may also use `\def` instead of `\renewcommand`; for example:

```
\def\counterlargebullet{\textbullet}
\def\countersmallbullet{\textperiodcentered}
```

(Actually, in a case like this, even `\let` could be used.)

2.4 Predefined bullet styles

To spare you the nuisance of redefining the hooks mentioned in Subsection 2.3, at least in normal situations, four "bullet styles" are already predefined by the bullcntr package, covering what the author believes are the most common cases. You can switch to any of these predefined styles by means of the following four declarations.

`\largectrbull` This declaration causes (relatively) "large" bullets to be used for all values from 1 to 9.

`\smallctrbull` This declaration causes "small" bullets to be used for all values from 1 to 9.

`\smartctrbull` This declaration causes (relatively) "large" bullets to be used for values from 1 to 5 (inclusive), and "small" bullets for values equal to or above 6.

`\heartctrbull` This declaration causes little hearts (instead of bullets) to be used for all values from 1 to 9.

These are in all respects normal declarations, obeying to ordinary scoping rules. The corresponding options described in Subsection 2.1 simply set the style at the outer (global, or document-wide) level.

2.5 Cross-references

Cross-references now work fine with the `\bullcntr` command. You may use `\label` (preceded by `\refstepcounter` if necessary) in the usual way to label, for example, an item in a list numbered via `\bullcntr` (recall that `\item` calls `\refstepcounter` by itself), and elsewhere cross-reference that item with `\ref`: the reference will be typeset correctly, using the same bullet style that was in effect at the spot where `\label` (and not `\ref`) was used.

Examples of such cross-references are shown later, in Subsection 4.4.

3 Using the bullenum package

The bullcntr package was conceived for the purpose of imitating in print an unusual way of numbering items in a list, that the author saw in certain handwritten notes (see the Introduction). With the `\bullcntr` command at hand, this could be achieved simply by saying something like

```
\begin{enumerate}
    \renewcommand*{\theenumi}{\bullcntr{enumi}}
    \renewcommand*{\labelenumi}{\theenumi}

    ⟨items⟩

\end{enumerate}
```

This, nonetheless, is not perfect, since it breaks if the `enumerate` environment is moved inside another `enumerate`, because of the explicit mention of `\theenumi` and of `\labelenumi`.

For this reason, and also to facilitate the use of the `\bullcntr` command for numbering list items, a new package has been added to the bullcntr package; this helper package, called bullenum, simply defines the `bullenum` environment, a variation of `enumerate` that numbers its items using the `\bullcntr` command. Indeed,

```
\begin{bullenum}

    ⟨items⟩

\end{bullenum}
```

is equivalent to

```
\begin{enumerate}
    \renewcommand*{\theenum⟨lvl⟩}{\bullcntr{enum⟨lvl⟩}}
    \renewcommand*{\labelenum⟨lvl⟩}{\theenum⟨lvl⟩}

    ⟨items⟩

\end{enumerate}
```

where ⟨*lvl*⟩ is `i`, `ii`, `iii`, or `iv`, depending on the current level of the `enumerate` environment.

In other words, `bullenum` can be used in every place where `enumerate` can, and numbers its items with an increasing number of bullets, regardless of the definition of `\theenum`⟨*lvl*⟩ and `\labelenum`⟨*lvl*⟩. Of course, the environment must contain no more than 9 items. Moreover, if you use `\label` to set up a reference, subsequent `\ref`'s will typeset the reference as a "bulletted" number, using, for the "bullets", the character that was current at the time the `\label` was encountered (in brief, they will do the right thing).

Notice that you should *not* load the bullcntr package if you invoke bullenum, because the latter calls the former. As explained below, you may pass to the bullenum package the options that you would require from bullcntr: the outer package will forward them to the inner one. Actually, most users will just use the bullenum package, and never call bullcntr directly.

3.1 Package invocation and options

You invoke the bullenum package with the usual `\usepackage` declaration put in the preamble of your document. In this declaration you may specify exactly the same options that you would pass to the bullcntr package, with the same meaning (see Subsection 2.1). The bullenum package then calls bullcntr for you, just passing on to it the options, if any, you asked for.

3.2 The `bullenum` environment

This environment is syntactically analogous to `enumerate`, but numbers its items with an increasing number of bullets (using the `\bullcntr` command). You must include no more than 9 items in this environment. The style of the bullets is governed by the declarations described in Subsection 2.4, according to ordinary scoping rules. A change local to the environment being employed can be obtained by enclosing the relevant declaration inside the environment itself; for example,

```
\begin{bullenum}
    \smallctrbull

    ⟨items⟩

\end{bullenum}
```

will use "small" bullets regardless of whatever style might have been in force outside the `bullenum` environment.

If you want finer control over the appearance of the "bullets", just redefine the hooks of Subsection 2.3; doing this inside the environment itself will keep changes to the hooks local. For example, you might say:

```
\begin{bullenum}
    \renewcommand*{\counterlargebullet}
        {$\scriptscriptstyle \diamondsuit$}
    \renewcommand*{\countersmallbullet}{\counterlargebullet}

    ⟨items⟩

\end{bullenum}
```

(yes, "macho TeX programmers" may also use `\def`, or `\let` where appropriate).

Of course, this environment respects the definitions of `\theenum`$\langle lvl \rangle$ and of `\labelenum`$\langle lvl \rangle$ in force outside the environment itself, for $\langle lvl \rangle =$ `i`, …, `iv`.

3.3 Nesting `bullenum` environments

Because a `bullenum` environment is nothing more than a modified `enumerate`, it can be nested in the same way, up to four levels deep.

For what concerns the syntax, it is perfectly in order to nest one `bullenum` environment inside another; it is from the viewpoint of the resulting visual appearance that this is not recommendable, unless quite different bullet styles are used for the two enumerations, so as to sharply mark the visual distinction between their respective labels. Of course, this problem does not arise if you nest a `bullenum` environment inside an `enumerate`, or viceversa.

Examples of such nested environments will be presented in Subsection 4.2.

4 Examples

A few examples follow, illustrating the usage of the bullcntr/bullenum packages.

4.1 A sample list

Here we illustrate the use of the `bullenum` environment to produce lists whose items are numbered with the `\bullcntr` command.

The lists in the various examples are produced by the following source code:

```
\begin{bullenum}

    ⟨items⟩

\end{bullenum}
```

We shall now show a few variations of a list of the above kind, changing the definition of the commands `\counterlargebullet` and `\countersmallbullet` among the various examples, so as to illustrate the use of several bullet sizes. In three examples out of five, however, this will be actually done by means of the declarations described in Subsection 2.4, thus falling back on one of the predefined bullet styles.

4.1.1 The standard sizes (`\smartctrbull`)

In our first example, actually, we do not alter the default definitions of the two hooks that typeset the bullets (see Subsection 2.3). The following is the output the bullcntr/bullenum packages will produce by themselves, if you invoke either of them without options and do not alter the default bullet style; in any case, it corresponds to the style selected by `\smartctrbull`.

- • This is the first item of the list, that is, item #1. It is therefore numbered with a single bullet. Poor bullet! How lonely it must feel!

- •• This is the second item of the list, that is, item #2. It is therefore numbered with two bullets, one on the left and one on the right. If you are a romantic nature, you might even think of them as Romeo and Juliet...

- ∴ This is the third item of the list, that is—guess what?—item #3. Not surprisingly, it is numbered with three bullets neatly arranged in the shape of an equilateral triangle.

- ⁘ This is the fourth item, item #4. A square has exactly four vertices, so it seems quite appropriate to lay out the four bullets that must be used to number this item exactly in this form, as you can see in the label.

- ⁙ If you have not lost the track, you will agree that this is the fifth item of the list. A regular pentagon would probably have been the optimal choice for displaying this value of the counter, but for some reason the arrangement shown in the label of this item has been preferred by the author of the bullcntr package.

- •••••• Sixth item! Item #6! A regular hexagon is used here! Note, however, that the bullets have shrunk to a smaller size, because the size used so far would have made the bullet cluster look a bit too crowded.

- ••••••• In the opinion of the author, the shape used to display the value 7 is even more pleasant than the regular hexagon used for the value 6; indeed, this is a regular hexagon too, but with the center marked by an additional bullet, which makes the figure look more symmetric.

- •••••••• If 7 was the perfect number of bullets, here we face the disruption of this perfection. The solution adopted for arranging 8 bullets, which you can see here, is, in the author's opinion, the less inelegant among all possibilities.

- ⠿⠇ Item #9, the last value that the `\bullcntr` command can display. The bullets are just stacked in a 3×3 square pattern—the obvious choice!

4.1.2 Big bullets

You might think that the `\textbullet` character is fitter for labelling list items; this is probably true for item numbers that do not exceede three, but begins to become questionable for higher values. In any case, here is an extreme example: `\textbullets` is used for all values from 1 to 9. To achieve this, the following source code should be used:

```
\begin{bullenum}
    \renewcommand*{\counterlargebullet}{\textbullet}
    \renewcommand*{\countersmallbullet}{\textbullet}

    ⟨items⟩

\end{bullenum}
```

This is what we write below.

• This is the first item of the list, that is, item #1. It is therefore numbered with a single bullet. Poor bullet! How lonely it must feel!

• • This is the second item of the list, that is, item #2. It is therefore numbered with two bullets, one on the left and one on the right. If you are a romantic nature, you might even think of them as Romeo and Juliet…

••• This is the third item of the list, that is—guess what?—item #3. Not surprisingly, it is numbered with three bullets neatly arranged in the shape of an equilateral triangle.

•••• This is the fourth item, item #4. A square has exactly four vertices, so it seems quite appropriate to lay out the four bullets that must be used to number this item exactly in this form, as you can see in the label.

••••• If you have not lost the track, you will agree that this is the fifth item of the list. A regular pentagon would probably have been the optimal choice for displaying this value of the counter, but for some reason the arrangement shown in the label of this item has been preferred by the author of the bullcntr package.

•••••• Sixth item! Item #6! A regular hexagon is used here! Note, this time, that the bullets have *not* shrunk to a smaller size, and that the resulting picture looks very crowded—pretty awful, I would say!

••••••• In the opinion of the author, the shape used to display the value 7 is even more pleasant than the regular hexagon used for the value 6; indeed, this is a regular hexagon too, but with the center marked by an additional bullet, which makes the figure look more symmetric.

•••••••• If 7 was the perfect number of bullets, here we face the disruption of this perfection. The solution adopted for arranging 8 bullets, which you can see here, is, in the author's opinion, the less inelegant among all possibilities.

••••••••• Item #9, the last value that the `\bullcntr` command can display. The bullets are just stacked in a 3 × 3 square pattern—the obvious choice!

A cross-reference to an item run into the main text may look really bad with bullets thus large: for instance, referencing the second item produces • • (quite acceptable), referencing the third one gives ••• (just about acceptable), but a reference to the sixth item is typeset as ••••••, and the ninth one even yields •••••••••! Note, however, that lines are not spaced out any further than usual.

4.1.3 Moderately large bullets (`\largectrbull`)

Our next example illustrates the output produced by the style selected with a `\largectrbull` declaration; this style continues to use, for the values over 5, the same bullet size adopted for the first five values. So, we say

```
\begin{bullenum}
    \largectrbull

    ⟨items⟩

\end{bullenum}
```

to produce the desired list.

• This is the first item of the list, that is, item #1. It is therefore numbered with a single bullet. Poor bullet! How lonely it must feel!

•• This is the second item of the list, that is, item #2. It is therefore numbered with two bullets, one on the left and one on the right. If you are a romantic nature, you might even think of them as Romeo and Juliet...

••• This is the third item of the list, that is—guess what?—item #3. Not surprisingly, it is numbered with three bullets neatly arranged in the shape of an equilateral triangle.

•••• This is the fourth item, item #4. A square has exactly four vertices, so it seems quite appropriate to lay out the four bullets that must be used to number this item exactly in this form, as you can see in the label.

••••• If you have not lost the track, you will agree that this is the fifth item of the list. A regular pentagon would probably have been the optimal choice for displaying this value of the counter, but for some reason the arrangement shown in the label of this item has been preferred by the author of the bullcntr package.

•••••• Sixth item! Item #6! A regular hexagon is used here! Note, once again, that the bullets have *not* shrunk to a smaller size, so that the resulting picture looks pretty crowded, although not intolerably bad-looking. Compare the label of this item with a reference to the corresponding item •••••• under 4.1.1, and with a reference to the corresponding item •••••• under 4.1.2.

••••••• In the opinion of the author, the shape used to display the value 7 is even more pleasant than the regular hexagon used for the value 6; indeed, this is a regular hexagon too, but with the center marked by an additional bullet, which makes the figure look more symmetric.

•••••••• If 7 was the perfect number of bullets, here we face the disruption of this perfection. The solution adopted for arranging 8 bullets, which you can see here, is, in the author's opinion, the less inelegant among all possibilities.

••••••••• Item #9, the last value that the `\bullcntr` command can display. The bullets are just stacked in a 3 × 3 square pattern—the obvious choice!

4.1.4 Small bullets (`\smallctrbull`)

The reciprocal example is to use small bullets throughout, from 1 to 9. Thus, we use the following source code:

```
\begin{bullenum}
    \smallctrbull

    ⟨items⟩

\end{bullenum}
```

And here is the output it yields.

• This is the first item of the list, that is, item #1. It is therefore numbered with a single bullet. Poor bullet! How lonely it must feel!

•• This is the second item of the list, that is, item #2. It is therefore numbered with two bullets, one on the left and one on the right. If you are a romantic nature, you might even think of them as Romeo and Juliet. . .

••• This is the third item of the list, that is—guess what?—item #3. Not surprisingly, it is numbered with three bullets neatly arranged in the shape of an equilateral triangle.

•••• This is the fourth item, item #4. A square has exactly four vertices, so it seems quite appropriate to lay out the four bullets that must be used to number this item exactly in this form, as you can see in the label.

••••• If you have not lost the track, you will agree that this is the fifth item of the list. A regular pentagon would probably have been the optimal choice for displaying this value of the counter, but for some reason the arrangement shown in the label of this item has been preferred by the author of the `bullcntr` package.

•••••• Sixth item! Item #6! A regular hexagon is used here! Note, once more, that the bullets have *not* shrunk to a smaller size, because they were *already* small in the items shown thus far.

••••••• In the opinion of the author, the shape used to display the value 7 is even more pleasant than the regular hexagon used for the value 6; indeed, this is a regular hexagon too, but with the center marked by an additional bullet, which makes the figure look more symmetric.

•••••••• If 7 was the perfect number of bullets, here we face the disruption of this perfection. The solution adopted for arranging 8 bullets, which you can see here, is, in the author's opinion, the less inelegant among all possibilities.

••••••••• Item #9, the last value that the `\bullcntr` command can display. The bullets are just stacked in a 3 × 3 square pattern—the obvious choice!

Again, compare the references to items #3 and #6 from different examples: standard bullets (••• and ••••••), big bullets (••• and ••••••), moderately large bullets (••• and ••••••), small bullets (••• and ••••••).

4.1.5 Tiny bullets

We come now to another extreme example: the use of `\textperiodcentered`. In other words,

```
\begin{bullenum}
    \renewcommand*{\counterlargebullet}{\textperiodcentered}
    \renewcommand*{\countersmallbullet}{\textperiodcentered}

    ⟨items⟩

\end{bullenum}
```

is asserted in order to produce the list.

· This is the first item of the list, that is, item #1. It is therefore numbered with a single bullet. Poor bullet! How lonely it must feel!

· · This is the second item of the list, that is, item #2. It is therefore numbered with two bullets, one on the left and one on the right. If you are a romantic nature, you might even think of them as Romeo and Juliet. . .

∴ This is the third item of the list, that is—guess what?—item #3. Not surprisingly, it is numbered with three bullets neatly arranged in the shape of an equilateral triangle.

⁘ This is the fourth item, item #4. A square has exactly four vertices, so it seems quite appropriate to lay out the four bullets that must be used to number this item exactly in this form, as you can see in the label.

⁙ If you have not lost the track, you will agree that this is the fifth item of the list. A regular pentagon would probably have been the optimal choice for displaying this value of the counter, but for some reason the arrangement shown in the label of this item has been preferred by the author of the bullcntr package.

⠿ Sixth item! Item #6! A regular hexagon is used here! Note that it was really impossible, this time, for the bullets to shrunk to a still smaller size than the one that has been adopted in the above items!

⡿ In the opinion of the author, the shape used to display the value 7 is even more pleasant than the regular hexagon used for the value 6; indeed, this is a regular hexagon too, but with the center marked by an additional bullet, which makes the figure look more symmetric.

⣿ If 7 was the perfect number of bullets, here we face the disruption of this perfection. The solution adopted for arranging 8 bullets, which you can see here, is, in the author's opinion, the less inelegant among all possibilities.

⋮⋮⋮ Item #9, the last value that the `\bullcntr` command can display. The bullets are just stacked in a 3 × 3 square pattern—the obvious choice!

This bullet style might be employed in a `bullenum` environment nested inside another one.

4.1.6 St. Valentine (\heartctrbull)

To end this inventory of variations, let us illustrate the use of a fancier “bullet”: a little heart. We obtain this by saying

```
\begin{bullenum}
    \heartctrbull

    ⟨items⟩

\end{bullenum}
```

This could be used in valentines to enumerate to your beloved the terms of your passion. . . :-)

♡ This is the first item of the list, that is, item #1. It is therefore numbered with a single heart. Poor heart! It is a true lonely heart!

♡♡ This is the second item of the list, that is, item #2. It is therefore numbered with two hearts, one on the left and one on the right. Thinking of Romeo and Juliet is absolutely natural in this context. . .

♡♡♡ This is the third item of the list, that is—guess what?—item #3. Not surprisingly, it is numbered with three hearts neatly arranged in the shape of an equilateral triangle.

♡♡♡♡ This is the fourth item, item #4. A square has exactly four vertices, so it seems quite appropriate to lay out the four hearts that must be used to number this item exactly in this form, as you can see in the label.

♡♡♡♡♡ If you have not lost the track, you will agree that this is the fifth item of the list. A regular pentagon would probably have been the optimal choice for displaying this value of the counter, but for some reason the arrangement shown in the label of this item has been preferred by the author of the bullcntr package.

♡♡♡♡♡♡ Sixth item! Item #6! A regular hexagon is used here, adorned with hearts! Of course, the hearts do not shrink, as doesn’t yours, and as doesn’t your ardent passion!

♡♡♡♡♡♡♡ In the opinion of the author, the shape used to display the value 7 is even more pleasant than the regular hexagon used for the value 6; indeed, this is a regular hexagon too, but with the center marked by an additional heart, which makes the figure look more symmetric.

♡♡♡♡♡♡♡♡ If 7 was the perfect number of hearts, here we face the disruption of this perfection. The solution adopted for arranging 8 hearts, which you can see here, is, in the author’s opinion, the less inelegant among all possibilities.

♡♡♡♡♡♡♡♡♡ Item #9, the last value that the `\bullcntr` command can display. The hearts are just stacked in a 3 × 3 square pattern—the obvious choice!

Cross-references are a bit funny with this style; they look like this: ♡ , ♡♡♡, ♡♡♡♡, ♡♡♡♡♡, ♡♡♡♡♡♡♡, and ♡♡♡♡♡♡♡♡♡.

4.2 Nested enumerations

Try not to overdo this: carrying it to excess could lead to confused output.

4.2.1 Nesting one `bullenum` inside another

We use “large” bullets for the outer environment, and “tiny” bullets for the inner one.

• First item: let this one alone, just to have a term of comparison.

•• Second item. Here we nest another `bullenum` environment:

  · First sub-item.

  ·· Second sub-item.

  ∴ Third sub-item.

  ⁘ Fourth sub-item. OK, this should suffice.

  Don’t forget, we are still inside item •• of the outer enumeration.

∴ Third item (of the outer enumeration). Let us repeat exactly the same nested enumeration as in item ••.

  · First sub-item.

  ·· Second sub-item.

  ∴ Third sub-item.

  ⁘ Fourth sub-item. OK, this should suffice.

  Back to item ∴ of the outer enumeration.

Now the bad news: you might want to refer to, say, item ∴⁘ under item ∴. As you can see, the reference comes out wrong, since the label from the outer enumeration is typeset with the style of the inner one. :-(This is a known bug: I decided not to correct it, since doing so would require introducing the concept of bullet styles distinguished by level, which I deem intolerably burdensome. Be aware of this limitation when you nest `bullenum` environments inside one another (this does not apply, however, to `bullenum`’s nested within `enumerate`’s, as we’ll see in the following examples).

4.2.2 Nesting `bullenum` inside `enumerate`

Let us repeat the same example shown above, but changing the outer environment to `enumerate`, and suppressing style variations.

1. First item: let this one alone, just to have a term of comparison.

2. Second item. Here we nest a `bullenum` environment:

   • First sub-item.

   •• Second sub-item.

   ∴ Third sub-item.

    ⁘ Fourth sub-item. OK, this should suffice.

    Don’t forget, we are still inside item 2 of the outer enumeration.

3. Third item (of the outer enumeration). Let us repeat exactly the same nested enumeration as in item 2.

    • First sub-item.

    • • Second sub-item.

    ∴ Third sub-item.

    ⁘ Fourth sub-item. OK, this should suffice.

    Back to item 3 of the outer enumeration.

See Subsection 4.4 for examples of cross-references to the items of these lists.

4.2.3 Nesting `enumerate` inside `bullenum`

As before, but the other way around: `bullenum` is now the outer environment, and `enumerate` the inner one.

• First item: let this one alone, just to have a term of comparison.

• • Second item. Here we nest an `enumerate` environment:

    (a) First sub-item.

    (b) Second sub-item.

    (c) Third sub-item.

    (d) Fourth sub-item. OK, this should suffice.

    Don’t forget, we are still inside item • • of the outer enumeration.

∴ Third item (of the outer enumeration). Let us repeat exactly the same nested enumeration as in item • •.

    (a) First sub-item.

    (b) Second sub-item.

    (c) Third sub-item.

    (d) Fourth sub-item. OK, this should suffice.

    Back to item ∴ of the outer enumeration.

Again, see Subsection 4.4 for examples of cross-references to the items of the above lists.

4.2.4 Deep nesting

Just to show what the output looks like, here is an example of `enumerate` and `bullenum` environments nested down to the fourth level. We avoid, however, nesting `bullenum` environments one inside the other, because of the problem with cross-references outlined under 4.2.1.

1. Invertebrates are of many kinds.

2. We detail the vertebrates.

    (a) Fishes. They include, for instance:

    - • barracuda;
    - • • carp;
    - ∴ swordfish.

    (b) Amphibians. They include, for instance:

    - • frogs, among which are
        - A. the barking frog,
        - B. the flying frog,
        - C. the green frog,
        - D. the tree frog,

      and may others;
    - • • toads, that comprise
        - A. the cane toad,
        - B. the giant toad,

      just to name a few;
    - ∴ salamanders, *e.g.*,
        - A. the green salamander,
        - B. the slimy salamander,
        - C. the tiger salamander.

    (c) Reptiles. Among these we find:

    - • snakes, for instance
        - A. cobra,
        - B. python,
        - C. anaconda,
        - D. viper,
        - E. rattlesnake;
    - • • turtles, of which the following
        - A. green turtle,
        - B. painted turtle,
        - C. snapping turtle

      are just a few examples.

    (d) Birds. (I'm a bit bored, now.)

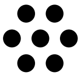
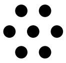
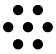
    (e) Mammals.

4.3 An example of use in the middle of text

We have just defined, in the source file of this document, a new counter named `mytest`, and we have set the value of the `mytest` counter to 3. With the default bullet sizes, this value is displayed by the command `\bullcntr{mytest}` as ∴. Other examples:

`\Huge \bullcntr{mytest}`	produces ∴
`\huge \bullcntr{mytest}`	produces ∴
`\LARGE \bullcntr{mytest}`	produces ∴
`\Large \bullcntr{mytest}`	produces ∴
`\large \bullcntr{mytest}`	produces ∴
`\normalsize \bullcntr{mytest}`	produces ∴
`\small \bullcntr{mytest}`	produces ∴
`\footnotesize \bullcntr{mytest}`	produces ∴
`\scriptsize \bullcntr{mytest}`	produces ∴
`\tiny \bullcntr{mytest}`	produces ∴

Let us repeat this example after assigning to the `mytest` counter the value 7, which `\bullcntr{mytest}` displays as :

`\Huge \bullcntr{mytest}`	produces 
`\huge \bullcntr{mytest}`	produces 
`\LARGE \bullcntr{mytest}`	produces 
`\Large \bullcntr{mytest}`	produces
`\large \bullcntr{mytest}`	produces
`\normalsize \bullcntr{mytest}`	produces
`\small \bullcntr{mytest}`	produces
`\footnotesize \bullcntr{mytest}`	produces
`\scriptsize \bullcntr{mytest}`	produces
`\tiny \bullcntr{mytest}`	produces (Ugh!)

But look at the output of

```
\begin{quote}
    \renewcommand*{\countersmallbullet}{\textperiodcentered}
    \tiny
    The value is~\bullcntr{mytest}.
\end{quote}
```

Here it is:

> The value is ⁙.

How cute it looks! :-)

4.4 Examples of cross-references

Cross-references are printed using the bullet style that was in effect when the `\label` command was given. The following examples illustrate this.

In 4.1.1 we had references to ∴ and to ⁙. In 4.1.6, on the other hand, the references were to ♡, to ♡♡♡, to ♡♡♡♡, to ♡♡♡♡♡, to ♡♡♡♡♡♡♡, and to ♡♡♡♡♡♡♡♡.

We turn now to nested lists. We let alone the example of 4.2.1, we already know that references come out wrong in this case. In 4.2.2 we had items 2 • , 2• •, 2∴, and 2•∴•, all listed under item 2, and similarly items 3 • , 3• •, 3∴, and 3•∴•, as part of item 3. In 4.2.3, on the other hand, we had items • •a, • •b, • •c, and • •d, all listed under item • •: as you can see, the bullets now precede the reference to the item of `enumerate`; similarly, items ∴a, ∴b, ∴c, and ∴d are now a part of the "bulletted" item ∴.

Coming to 4.2.4 (the animal kingdom), we see that, for instance, the green frog is 2(b) • C, an example of frog (under 2(b) •), which, in turn, is an example of amphibian (see 2b). On the other hand, the tiger salamander is a kind of salamander (cf. 2(b)∴) listed under 2(b)∴C.

And so on…

References

[1] L. Lamport, *LaTeX—A Document Preparation System—User's Guide and Reference Manual*, 2nd edition, Addison-Wesley, Reading, 1994.