create horizontal columned lists

Clemens Niederberger

https://github.com/cgnieder/tasks/

contact@mychemistry.eu

Table of Contents

1 Motivation 1 6 Custom Labels 13
2 License and Requirements 2 7 New Tasks 13
3 How it works 2 8 Styling TASKS 16
   3.1 The Basics 2 8.1 The tasks Object 6
   3.2 Items Spanning More Than One Column 4 8.1.1 Available Options 16
       8.1.2 Predefined Instances 17
4 Available Options 8 Index 18
5 Available Instances 12

1 Motivation

Originally TASKS has been an integral part of the ExSheets package. However, users told me that it indeed could be useful to have it as a stand-alone package not having to load the whole ExSheets beast just for having the tasks environment available. Since I agree with this the environment has been extracted into a package if its own, TASKS. Since then TASKS has been distributed as a package of its own but as part of the ExSheets bundle. With v0.10 I decided to make it a completely independent package. So the relation to ExSheets only is a historical one.

The reason for the tasks environment is an unwritten agreement in German maths textbooks (especially in (junior) high school textbooks) to organize exercises in columns counting horizontally rather than vertically. That is what tasks primarily is for. If you don’t need this feature you’re better off using traditional LaTeX lists and the enumitem package for customization.
2 License and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the \LaTeX\ Project Public License (LPPL), version 1.3 or later (http://www.latex-project.org/lppl.txt). The software has the status "maintained."

\texttt{tasks} requires the \texttt{l3kernel [L3Pa]} bundle, \texttt{xparse},\texttt{xtemplate} and \texttt{l3keys2e} which are part of the \texttt{l3packages [L3Pb]} bundle, epic [Pod86], cntformats [Nie14], and environ [Rob13].

3 How it works

3.1 The Basics

The \texttt{tasks} environment is similar to a list like \texttt{enumerate} but not the same. Here are some of the differences:

- A first difference: there is no pagebreak possible inside an item but only between items.
- A second difference: the enumeration default is a), b), c) …
- A third difference: the body of the \texttt{tasks} environment is split at \texttt{every} occurrence of the item separator. For this reason the default separator is \texttt{not} \texttt{item} but \texttt{task} so it is unique to this environment only. This directly leads to...
- … a fourth difference: the \texttt{tasks} environment cannot be nested. You can, however, use an \texttt{itemize} environment or another "real" list in it.
- A fifth difference: verbatim material cannot be used in it. You’ll have to use \texttt{string}, \texttt{texttt} or \texttt{detokenize}. If this won’t suffice then don’t use \texttt{tasks}.

\begin{tasks}[(\langle options \rangle)](\langle num of columns \rangle)
List like environment where the single items are introduced with \texttt{task}.

Let’s see an example:

\begin{tabular}{l}
1 \% \Sample is defined to contain some sample text: \% \texttt{\def\Sample{This is some sample text we will use to create a somewhat longer text spanning a few lines.}} \% \texttt{\Sample\ \Sample\par\Sample} \\
5 Some text before the list.
6 \texttt{\begin{tasks}} \\
7 \texttt{\task \Sample} \\
8 \texttt{\task \Sample} \\
9 \texttt{\task \Sample}
\end{tabular}

1. on CTAN as \texttt{xparse}: http://mirrors.ctan.org/macros/latex/contrib/xparse/  
2. on CTAN as \texttt{l3keys2e}: http://mirrors.ctan.org/macros/latex/contrib/l3keys2e/
3 How it works

And also some text after it.

Some text before the list.

a) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

This is some sample text we will use to create a somewhat longer text spanning a few lines.

b) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

This is some sample text we will use to create a somewhat longer text spanning a few lines.

c) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

This is some sample text we will use to create a somewhat longer text spanning a few lines.

And also some text after it.

The environment takes the optional argument (**num of columns**) with which the number of columns used by the environment is specified.

\begin{tasks}(2)
\task Sample
\task Sample
\task Sample
\task Sample
\task Sample
\task Sample
\end{tasks}
### 3 How it works

a) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

b) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

c) This is some sample text we will use to create a somewhat longer text spanning a few lines.

d) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

This is some sample text we will use to create a somewhat longer text spanning a few lines.

e) This is some sample text we will use to create a somewhat longer text spanning a few lines.

This is some sample text we will use to create a somewhat longer text spanning a few lines.

### 3.2 Items Spanning More Than One Column

Sometimes it may come in handy if an item is allowed to span more than one column. `{tasks}` supports items using the remaining space by adding an optional star to `{task}`:

```latex
\begin{tasks}(3)
  \task \sample
  \task* \sample
  \task* \sample
  \task \sample
  \task \sample
  \task \sample
  \end{tasks}
```
3 How it works

a) This is some sample text we will use to create a somewhat longer text spanning a few lines.

b) This is some sample text we will use to create a somewhat longer text spanning a few lines.

c) This is some sample text we will use to create a somewhat longer text spanning a few lines.

d) This is some sample text we will use to create a somewhat longer text spanning a few lines.

e) This is some sample text we will use to create a somewhat longer text spanning a few lines.

Tasks also supports items that span all columns in any case by adding an optional bang to \task.

\begin{tasks}(3)
\task \sample
\task! \sample
\task! \sample
\task \sample
\task \sample
\end{tasks}

a) This is some sample text we will use to create a somewhat longer text spanning a few lines.

b) This is some sample text we will use to create a somewhat longer text spanning a few lines.

c) This is some sample text we will use to create a somewhat longer text spanning a few lines.
d) This is some sample text we will use to create a somewhat longer text spanning a few lines.

e) This is some sample text we will use to create a somewhat longer text spanning a few lines.

The optional star has itself an optional argument with parentheses where you can specify the number of columns the item is supposed to span:

\begin{tasks}(4)
\task the first
\task the second
\task the third
\task the fourth
\task*(3) the fifth item is way too long for this and needs three columns
\task the sixth
\task the seventh
\task*(2) the eighth item is way too long for this and needs two columns
\task the nineth
\task the tenth
\end{tasks}

If there are not enough columns left (say two columns but you said \task*(3)) the argument is ignored and the maximum number of remaining columns is used (two in case of our example).

Both optional star and optional bang can be combined with the optional argument for a custom label:
3 How it works

\begin{tasks}(3)
\task \sample
\task* \sample
\task*{(x)} \sample
\task \sample
\task \sample
\end{tasks}

\begin{tasks}(4)
\task the first
\task the second
\task the third
\task the fourth
\task \rlap{the fifth item is way too long for this so we start a new row}\startnewitemline
\task \rlap{the eighth item also is too long} \startnewitemline
\task the ninth
\task the tenth
\end{tasks}

Forcing a new item line manually is also possible using the following command:

\startnewitemline
Introduce a new line in a tasks environment.

a) This is some sample text we will use to create a somewhat longer text spanning a few lines.

(x) This is some sample text we will use to create a somewhat longer text spanning a few lines.

c) This is some sample text we will use to create a somewhat longer text spanning a few lines.

d) This is some sample text we will use to create a somewhat longer text spanning a few lines.

Introduced in version 0.9
4 Available Options

While this works it also needs a bit of care since the width of the items doesn’t change which means in order to use the full width you’d have to use trickery like `\rlap` which then means the danger of the item text sticking into the margin...

4 Available Options

The \texttt{tasks} package does not have any package options.

The environment \texttt{tasks} has a number of options, though, namely the following ones that can be set using a setup command:

\begin{tasks}{⟨options⟩}
\end{tasks}

\texttt{\settasks{⟨options⟩}}

Setup command for \texttt{tasks}.

\texttt{style = ⟨⟨instance⟩⟩} \hspace{5cm} (initially empty)

Choose the instance to be used. Read more on this in section 8.1.

\texttt{counter-format = ⟨⟨counter specs⟩⟩} \hspace{5cm} Default: tsk[a])

Sets a custom label. The letters tsk are replaced with the task-counter. An optional argument directly following these letters specifies the counter format: 1: \texttt{arabic}, a: \texttt{alph}, A: \texttt{Alph}, r: \texttt{roman} and R: \texttt{Roman}.

\texttt{label-format = ⟨⟨code⟩⟩} \hspace{5cm} (initially empty)

Can be used to apply a formatting like, e.g., \texttt{bfseries} to the labels.

\texttt{label = ⟨⟨code⟩⟩} \hspace{5cm} (initially empty)

Overwrite the automatic label to a custom one.

\texttt{label-width = ⟨⟨dim⟩⟩} \hspace{5cm} Default: 1em

Sets the width of the item labels.

\texttt{label-offset = ⟨⟨dim⟩⟩} \hspace{5cm} Default: .3333em

Sets the offset, i.e., the distance between label and item.
Available Options

\begin{figure}
\centering
\begin{tabular}{c|c|c|c}
label width & item offset & column sep & item indent \\
\hline
\end{tabular}
\caption{A visual representation of the used lengths.}
\end{figure}

\texttt{item-format} = \{\texttt{code}\} \\
\hspace{1cm} (initially empty)

Can be used to apply a formatting like, e.g., \texttt{\bfseries} to the items. This may be a macro accepting the item as mandatory argument.

\texttt{item-indent} = \{\texttt{dim}\} \\
\hspace{1cm} Default: 2.5em

The indent of an item, \textit{i.e.}, the horizontal space available for both label and label-offset. If

\[ \text{indent} = \text{label-width} + \text{label-offset} \]

the label will align with the textblock above (if \texttt{label-align} = \{\texttt{left}\} is set). Please see figure 1 for a sketch of the available lengths and how they are set.

\texttt{column-sep} = \{\texttt{dim}\} \\
\hspace{1cm} Default: 0pt

A horizontal length that is inserted between columns of items.

\texttt{label-align} = \texttt{left}|\texttt{right}|\texttt{center} \\
\hspace{1cm} Default: \texttt{left}

Determines how the labels are aligned within the label-box whose width is set with \texttt{label-width}.

\texttt{before-skip} = \{\texttt{skip}\} \\
\hspace{1cm} Default: 0pt

Sets the skip before the list.

\texttt{after-skip} = \{\texttt{skip}\} \\
\hspace{1cm} Default: 0pt

Sets the skip after the list.

\texttt{after-item-skip} = \{\texttt{skip}\} \\
\hspace{1cm} Default: 1ex plus 1ex minus 1ex

This vertical skip is inserted between rows of items.

\texttt{resume} = \texttt{true}|\texttt{false} \\
\hspace{1cm} Default: \texttt{false}

The enumeration will resume from a previous tasks environment. In order to use this option properly you shouldn’t mix different tasks environments that both count their items.

\texttt{debug} = \texttt{true}|\texttt{false} \\
\hspace{1cm} Default: \texttt{false}

If set to \texttt{true} \texttt{\fboxsep} is set to 0pt inside the tasks environment and \texttt{\fbox} is used to draw a frame around the label boxes and the item boxes.

Now the same list as above but with three columns and a different label:
Let’s use it inside a question, *i.e.*, inside **ExSheets**’ question environment:
Question 1.
I have these two tasks for you. Shall we begin?

1.1 The first task: easy! 1.2 The second task: even more so!

Solution 1.
Now, let’s see… ah, yes:

1.1 This is the first solution. Told you it was easy.

1.2 This is the second solution. And of course you knew that!

Finally let’s see what the \texttt{debug} option does (you could see it already on page 6):

\begin{tasks}(2)
5 Available Instances

There are currently three additional instances for the `tasks` object available:

- **itemize** uses `\labelitemi` as labels.
- **enumerate** enumerates the items with 1., 2., ...
- **multiplechoice** a – well – ‘multiple choice’ list.
6 Custom Labels

If you want to change a single label inside a list, you can use the optional argument of `\task`. This will temporarily overwrite the default label.

\begin{tasks}[style=itemize]
\task a standard item
\task another one
\task[+] a different one
\task and another one
\end{tasks}

7 New Tasks

It is possible to add custom environments that work like the `tasks` environment.

\begin{Verbatim}
\texttt{\def\NewTasks[\langle options\rangle]{\langle name\rangle}[\langle separator\rangle][\langle cols\rangle]}
\texttt{\def\RenewTasks[\langle options\rangle]{\langle name\rangle}[\langle separator\rangle][\langle cols\rangle]}
\end{Verbatim}

Define environment $\langle name\rangle$ that uses $\langle separator\rangle$ to introduce a new item. Default for $\langle separator\rangle$ is `\task`, default for $\langle cols\rangle$ is 1. The $\langle options\rangle$ are the ones described in section 4.

The `tasks` environment is defined as follows:
7 New Tasks

\NewTasks\{tasks\}

The separator does not have to be a control sequence:

\begin\{done\}\(3\)
* First task
* Second task
*! Third task spanning the full width available

Although this might seem handy or even nice I strongly advise against using something different than a command sequence. Remember that the items will be split at every occurrence of the separator. So in order to use the separator (here for example for a starred variant of a command) within an item it has to be hidden in braces. This is avoided of you use a command sequence which even doesn’t have to be defined.

Please also keep in mind that the separator still has an optional star argument (see 4), an optional bang argument and the standard optional argument. Using * will prevent the optional star argument.
Let’s say you want a multiplechoice environment that has three columns in its default state. You could do something like this:

Exercise 2.

☐ First choice  ☐ Second choice  ☐ Third choice
Solution 2.

- First choice
- Second choice
- Third choice

The last example shows you two additional commands:

\choicebox
- Print an empty square.

\checkedchoicebox
- Print a crossed-out square.

8 Styling \texttt{tasks}

Equivalent to the styling of \texttt{ExSheets tasks} uses \texttt{xtemplate} to declare additional instances for the lists.

8.1 The \texttt{tasks} Object

The object that’s defined by \texttt{tasks} is the ‘tasks’ object. This time there are four instances available for the one template (again ‘default’) that was defined.

8.1.1 Available Options

This section only lists the options that can be used when defining an instance of the ‘default’ template. The following subsections will give some examples of their usage.

\begin{verbatim}
\DeclareTemplateInterface{tasks}{default}{3}
{
  % option : type = default
  enumerate : boolean = true ,
  label : tokenlist ,
  indent : length = 2.5em ,
  counter-format : tokenlist = tsk[a]) ,
  label-format : tokenlist ,
  label-width : length = 1em ,
  label-offset : length = .3333em ,
  after-item-skip : skip = 1ex plus 1ex minus 1ex
}
\end{verbatim}
8.1.2 Predefined Instances

This is rather brief this time:

\begin{alltt}
\% ALPHABETIZE: a) b) c)
\DeclareInstance{tasks}{alphabetize}{default}{}
\% available when `load-tasks=true':
\% ITEMIZE:
\DeclareInstance{tasks}{itemize}{default}
{
  enumerate = false ,
  label-width = 1.125em
}
\% ENUMERATE:
\DeclareInstance{tasks}{enumerate}{default}
{ counter-format = tsk. }
\% MULTIPLECHOICE:
\DeclareInstance{tasks}{multiplechoice}{default}
{
  enumerate = false ,
  label = \choicebox ,
}
\end{alltt}

References


Index

A
after-item-skip ........................................ 9
after-skip ........................................ 9

B
before-skip ........................................ 9

C\checkedchoicebox ..................................... 16
\choicebox ........................................ 16f.
cntformats (package) .................................. 2
column-sep ........................................ 9
counter-format ....................................... 8
CTAN .................................................. 2

D
debug ................................................. 9, 11

E enumerate (environment) .......................... 2
enumitem (package) .................................. 1
environ (package) ..................................... 2
epic (package) ....................................... 2

I
item-format ............................................ 9
item-indent .......................................... 9

L\lthreekernel (bundle) ................................ 2
\lthreekeysze (package) ............................... 2
\lthreepackages (bundle) ............................ 2
label ................................................. 8
label-align ......................................... 9
label-format ........................................ 8
label-offset ........................................ 8
label-width ........................................ 8f.
LPPL ............................................... 2

N\NewTasks ............................................ 13f.
Niederberger, Clemens ............................... 2

P Podar, Sunil ....................................... 2

Q question (environment) ........................... 10

R\RenewTasks ......................................... 13
resume ............................................... 9
Robertson, Will ...................................... 2
\lthreeteachines .................................... 9
\settasks ........................................... 6, 8, 11
\startnewitemline ................................... 7
style ............................................... 8

T\lthreeH TIMES ...................................... 13
\task ................................................. 2–7, 10–15
tasks (environment) ............................... 1f., 7ff., 13
The \texttt{\LaTeX3} Project Team ..................... 2

X\xparse (package) ................................... 2
\xtemplate (package) ................................. 2, 16