The gatherenum package
Gathered, display-like `enumerate`*

Julien "_FrnchFrgg_" Rivaud†

Released 2019/09/29

1 Documentation

1.1 Package description

This package (ab)uses the inline enumeration capabilities of enumitem to add a "displayed" enumeration mode, triggered by adding `gathered` to the key-val option list of the `enumerate` environment.

The end result is similar to The end result is similar to a regular enumerate environment wrapped in a multicols environment, with the following advantages:

- *gathered* enumerate can pack items depending on their actual width rather than a fixed, constant number per line;
- it fills items in a line-major order (instead of column-major order), which my students found less confusing.Your mileage may vary.

All settings of the standard enumeration that are relevant still apply to the gathered mode — which would not have been the case with a separate inline enumeration like the `enumerate*` provided by enumitem.

Individual items are separately boxed, to mimic the behavior of `align`. This package is not for you if you want free-flow enumeration in a paragraph, since enumitem already provides a reasonable one, and you probably do not want to share that many settings with regular `enumerate` due to the very different nature and look of inline enumerations.

Say:

```
\setlist{enumerate}{label=\arabic*.}
\begin{enumerate}
  \item Test
  \item $\int_a^b$ with math
  \item And another
  \item $f(x) = 6$
  \item $D(t) = 0$
  \item $d$~and~$d'$ are parallel
\end{enumerate}
```

*This file describes v1.8, last revised 2019/09/29.

†E-mail: frnchfrgg@free.fr

produces:

1. Test
2. $\int_a^b$ with math
3. And another
4. $f(x) = 6$
5. $D(t) = 0$
6. d and d' are parallel

Then:

```
\begin{enumerate}[gathered]
  \item Test
  \item $\int_a^b$ with math
  \item And another
  \item $f(x) = 6$
  \item $D(t) = 0$
  \item $d$~and~$d'$ are parallel
\end{enumerate}
```

will produce:

1. Test 2. $\int_a^b$ with math 3. And another 4. $f(x) = 6$ 5. $D(t) = 0$ 6. d and d' are parallel

1.2 Customizing

Most of the changes are in fact done through the usual enumitem interface. `itemjoin` can be a useful setting to tweak. gatherenum provides several additional options:

- `gatherformat` this is put just before every item content; the last token in that option can take an argument, which will be the unboxed item content. Note that since the item has already been typeset in an horizontal box, you cannot change the font family, shape or size at that point, but can add a frame around for instance.
- `center` display the items in a centered paragraph. This is the most display-like and is the default. Equivalent to `before*=\centering`.
- `alignleft` display the items in a left-aligned paragraph.
  Equivalent to `before*=\raggedright`.

Lastly, an example of something I use a lot:

Compute the following:

$a = 1 + 1$ $b = \sin\left(2\pi + \frac{\pi}{3}\right)$ $c = \displaystyle\int_0^{+\infty} \mathrm{e}^{-t^2}\mathrm{d}t$ $d = \varphi(52330)$

```
Compute the following:
\begin{enumerate}[gathered, label={$\alph* = {}$}, labelsep=0pt]
  \item $1+1$
  \item $\sin\left( 2\pi + \frac{\pi}{3} \right)$
  \item $\displaystyle \int_0^{+\infty} \mathrm{e}^{-t^2} \mathrm{d}t$
  \item $\varphi(52330)$
\end{enumerate}
```

---

2 gatherenum implementation

```
⟨*package⟩
⟨@@=gatherenum⟩
\ProvidesExplPackage
  {\ExplFileName}{\ExplFileDate}{\ExplFileVersion}{\ExplFileDescription}
```

We use enumitem as a base, and xparse to define our environment.

```
\RequirePackage{enumitem}
\RequirePackage{xparse}
```

enumitem has a mechanism to make the `enumerate` environment use settings and counters prefixed by `\enum` instead of `\enumerate` (because that's what LaTeX does). We take advantage of that to define our internal enumeration, because we want to share settings and counter with `enumerate` since we will act as an option of that environment.

```
\tl_set:Nn \enit@shortgatherenum {enum}
\newlist{gatherenum}{enumerate*}{3}
```

Since enumitem inline enumerations are intended for pararaph-like enumerations, the item content is unpacked with `\unhbox` even in *boxed* mode. That's not what we want, so we have to add another layer of boxing. While we're at it, we add a formatting command for the user to customize.

```
\tl_new:N \l__gatherenum_itemformat
\cs_new_protected_nopar:Nn \__gatherenum_boxitem: {
    \nobreak\skip_horizontal:n {\labelsep}
    \hbox_set:Nn \enit@inbox {
        \hbox:n {
            \l__gatherenum_itemformat{\hbox_unpack:N \enit@inbox}
        }
    }
}
\enitkv@key{enumitem}{gatherformat}{\tl_set:Nn\l__gatherenum_itemformat{#1}}
```

Now is the time to change the enumerate environment. We save the start and end of existing environment from enumitem to be able to fall back to it.

```
\let\__gatherenum_save_enumerate:w\enumerate
\let\__gatherenum_save_endenumerate:w\endenumerate
\RenewDocumentEnvironment{enumerate}{ O{} }{
```

The trigger for gathered enumeration is the `gathered` option given by the user in the list of local options. If there is no such option, we directly use the original `\endenumerate`.

```
    \clist_if_in:nnTF { #1 } { gathered } {
```

To act more display-like, we ensure that running heads that are not already typeset are flushed now. This is especially important in the default centered typesetting: we don't want to center the theorem or enumerate heads along with our content.

```
        \if@inlabel
            \leavevmode
        \fi
        \par
        \centering
```

Synchronise the current depth of gatherenum with that of enumerate (oddly `\enit@shortgatherenum` doesn't imply that). That enables us to get the right settings depending on depth (labels, mostly, since indention is irrelevant here).

```
        \int_set_eq:NN \enitdp@gatherenum \enitdp@enumerate
```

Insert a display-like penalty, and start our inline enumeration. Since setting defaults for gatherenum would clobber (or a least mess with) the settings of normal enumerate, we give them here using the optional argument (recall that they share most counters and settings).

We use a trivial align mode to ensure enumitem doesn't try to apply indention, alignment or spacing effects to the label. ⟨*afterlabel*⟩ is a hook that enumitem calls after closing the `\hbox` with item content, after typesetting the label, but before unboxing the content. We use `\__gatherenum_boxitem:` defined before to ensure the content stays boxed.

```
        \penalty \predisplaypenalty
        \gatherenum[
            itemjoin=\skip_horizontal:n{1em plus 1fil},
            #1,
            mode=boxed,
            align=none,
            afterlabel=\__gatherenum_boxitem:,
        ]
    }{
        \__gatherenum_save_enumerate:w[#1]
    }
}{
```

Finish the environment, either by closing the standard `enumerate`, or by ending our horizontal enumeration, and closing the paragraph. We are less strict on widows and clubs than normal.

```
    \clist_if_in:nnTF { #1 } { gathered } {
        \endgatherenum
        \unskip
        \int_set_eq:NN \clubpenalty \interlinepenalty
        \int_set_eq:NN \widowpenalty \interlinepenalty
        \use:c{@ @ par}% avoid l3docstrip replacement of @
    }{
        \__gatherenum_save_endenumerate:w
    }
}
```

Last touches: a `gathered` key for enumitem so that it doesn't complain when it sees it (this enables us to keep the key in the list instead of filtering it with `\clist_remove_all:Nn`). Next is the trivial align mode we talked about earlier, and at last some shorthands for common layouts.

```
\SetEnumitemKey{gathered}{}
\SetLabelAlign{none}{#1}
\SetEnumitemKey{centered}{before*=\centering}
\SetEnumitemKey{alignleft}{before*=\raggedright}
```

⟨/package⟩