The `exercisesheets` Package

Sebastian Kuhnert Frank Fuhlbrück

Version 0.17, 2024-04-23

The `exercisesheets` package provides a way to typeset exercise sheets as used in university courses and school classes. It evolved from a set of macros an environments that were finally combined into this package. Starting from Version `0.7`, there was an alternative variant designed for use with the beamer class, which is integrated into the main package since version `0.11`. Not all combinations of options have been tested with the beamer variant. From version `0.16` on, there is an embedded mode that allows using exercise in lecture notes etc. without requiring page breaks. Since the package includes a loading mechanism for exercises in external files, the same exercises can be reused in different contexts.

Licence

1 Reading this Document

- All macros and environments defined by this package are printed in bold like this: **`\points`** and `\begin{`**`sheet`**`}\end{`**`sheet`**`}`.
- Key-value style options with a mandatory argument are printed like this: `only`=⟨*list of ranges*⟩, while optional (usually boolean) arguments look like this: `embedded`=⟨*true/false*⟩.

2 Related Classes and Packages

2.1 Packages with Similar Functionality

- The `exercise` package offers similar functionality, though the concept is a bit different: That package provides explicit commands for sub-exercises

---

[1] frank@fuhlbrueck.net

while `exercisesheets` (in its standard setting) relies on other means like the `enumerate` environment for this. An advantage of the `exercise` package is that answers can be delayed. On the other hand `exercisesheets`'s environment based user interface is a bit cleaner and you have to care less about package internals.

- The `xsim` (and its predecessor `exsheets`) package also offers a similar set of features for the average user. It also has a mechanism for hiding certain exercises and solutions that works a bit differently than the mechanism of this package. There is (to my knowledge) no explicit beamer support.

2.2 Recommended Additional Packages

- The use of `hyperref` is encouraged. Please use the options `pdfusetitle`, `plainpages=false` and `pdfpagelabels` for optimal results.
- If the `babel` package is loaded, the appropriate `language=`⟨*lang*⟩-option is automatically derived and can be omitted.
- The use of `enumitem` or `paralist` is recommended. Top level enumerating lists are then modified to have the form (a), (b), . . . , which is useful in exercise definitions (you can override this).
- If the `varioref` package is used, appropriate label formats are installed for exercises and sub-exercises.
- If the `xcolor` package is loaded, to-do markers generated with **`\TODO`** are printed in red.

2.3 Packages Loaded by exercisesheets

- `etoolbox` (at least version 1.7)
- `scrlfile` (part of KOMA-Script)
- `pgfkeys` (part of PGF 2.0)
- `ifthen`
- `amsmath` (for `\numberwithin`)
- `ifluatex`
- `refcount` (for `\getrefnumber`)

3 Basic Usage

3.1 Package Loading

The beamer variant is integrated now and automatically activated if the current document class is beamer.

`\usepackage[`⟨*options*⟩`]{exercisesheets}`

The following options are available:

`only=`⟨*list of ranges*⟩ When given, only the mentioned sheets will be included in the output. This is useful to speed up compilation times for big documents. Some efforts are made that sheet and exercise numbering remain consistent and references to exercises and enumerated lists on skipped sheets still work as expected. References to other objects are broken though (they currently point to the exercise or item containing them). The ⟨*list of ranges*⟩ is a comma separated list of sheet numbers and sheet ranges. If a comma is included in the range, it has to be protected with a pair of braces. The special sheet "number" `last` stands for the last sheet in the document (this may require an additional LaTeX run). Examples:

- `only=3` Only include the 3rd sheet of the document.
- `only={1,3,5,7}` Only include the sheets 1, 3, 5 and 7.
- `only=last` Only include the last sheet of the document.
- `only={3-5,8-last}` Exclude the sheets 1, 2, 6 and 7.
- `only={-2,5-}` Exclude the sheets 3 and 4.

`all` Typeset all sheets (equivalent to `only=-`). This is the default.

`solutions=`⟨*true/false/oral/nonoral/sample oral/sample nonoral/sample all*⟩ By default, solutions (provided in the `solution` environment) are not included in the output. By providing the option `solutions=true` (or just `solutions`) this can be changed. Choosing `solutions=oral` shows only solutions for exercises marked `oral` (useful for printouts taken to class). Choosing `solutions=nonoral` shows only solutions for exercises that are not marked `oral` (useful for correcting). If you want to provided sample solutions for your students but only for some of your exercises you can use the `sample` key to select those exercises along with any options starting with sample here. Note that non-sample exercises are skipped completely (not only their solutions) if you choose an option starting with sample.

`gradingguides=`⟨*true/false*⟩ Within `gradingguide` environments (usually used inside `solution`) you can specify e.g. the amount of points to assign for certain solutions. This option controls whether or not to include this guides in the output (e.g. solutions for correcting contain them, but sample solutions for students don't).

`solutionsby=`⟨*name/names*⟩ Use this to give the authors of the solutions. They are credited at the beginning of each sheet, if the solutions are included in the output.

`language=`⟨*lang*⟩ Set the language of the sheet and exercise heads to ⟨*lang*⟩. If the babel package is loaded this is not necessary, as the main document language will be automatically detected. Currently `english`, `german` and `ngerman` are supported. Other translations are welcome: Please contact the author.

`pointsfloatright` Use an alternative mechanism to place points for Sub-Exercises. If this option is used, the points label for a Sub-Exercise is placed at the (right) end of the current line. It is especially useful if used in a context like `\item\points{3}`. This option exploits (and partially breaks) RTL-Support.

`exercisespath` Set the (relative) path of the directory containing exercises to be included via **`\includeexercise`**.

`patchenumerate=`⟨*true/false*⟩ Sets enumerate item labels for the first four levels, starting with a), b) etc. for the top-level (sub-exercises). Depending on other loaded packages (`enumitem`, `paralist`) the mechanism slightly differs. This option is active by default.

`settitle=`⟨*true/false*⟩ If set to true (which is the default), the document `\title` will be automatically set to the value of `strings/sheets` and the value of `strings/solutions` will be appended if `solutions` is not `false`.

`patchpagenumbers=`⟨*true/false*⟩ If true (default), pages will be numbered within sheets and page numbers get a prefix of the form "⟨*sheet number*⟩`-`".

`usestartsection=`⟨*true/false*⟩ If true (default), the command `\@startsection` is used to create the header of any exercise as a section, which (among other things) adds an entry to the TOC (if existent) at level 1 (usually corresponding to `\section`).

`embedded=`⟨*true/false*⟩ Default: false. Embedded mode allows for exercise "sheets" which do not interfere with a surrounding document. More explicitly, it does not interfere with `secnumdepth`, defines special counters `exshsheet` and `exshexercise` to use them for `sheet counter` and `exercise counter`, respectively and sets all of `patchenumerate`, `settitle`, `patchpagenumbers` and `usestartsection` to `false`. An internally defined macro which basically does no extra formatting or adding anything to the TOC is used as `sheet sectioning cmd` instead of the usual `\part`. Finally, the `sheet start page action` and the `sheet end page action` are both set to `\relax`. If you want to override any of this, do it after setting `embedded` to `true`.

`minskips=`⟨*true/false*⟩ Default: false. If true, the values of `above sheet title skip`, `below sheet header skip`, `above first exercise skip` and `below exercise title skip` are all set to 0pt.

`showtodos / hidetodos` **`\TODO`** markers are hidden by default, this option switches their behavior. This option is usually set within **`\ifsamplesolutions`** or similar.

`beamercompatibility=`⟨*true/false*⟩ This option (which only is effective in the non-beamer version) defines several beamer macros with as trivial effects as possible, e.g., `\pause` becomes `\relax` and overlay specifications are mostly ignored. Be careful to use this option after loading `enumitem` etc. as the enumerate and itemize environment are defined to swallow overlay specifications without effect, this also holds for `\item<1->`. If you load `exercisesheets` before them use `\exshset` afterwards. The current list of redefinitions is as follows:

```
\let\pause\relax
\def\frame{}
\renewcommand{\frame}[1][]{}
\let\endframe\relax
\def\onslide<##1>{}
\def\only<##1>##2{##2}
\def\uncover<##1>##2{##2}
\def\visible<##1>##2{##2}
\def\invisible<##1>##2{##2}
\long\def\alt<##1>##2##3{##2}
\long\def\temporal<##1>##2##3##4{##3}
\let\exsh@origitem\item
\def\exsh@ovrlitem<##1>{\exsh@origitem}
\def\item{\@ifnextchar<\exsh@ovrlitem\exsh@origitem}
\let\exsh@origitemize\itemize
\def\exsh@ovrlitemize[##1]{\exsh@origitemize}
\def\itemize{\@ifnextchar[\exsh@ovrlitemize\exsh@origitemize}
\let\exsh@origenumerate\enumerate
\def\exsh@ovrlenumerate[##1]{\exsh@origenumerate}
\def\enumerate{%
  \@ifnextchar[\exsh@ovrlenumerate\exsh@origenumerate}
```

`filenameasexercisename=`⟨*true/false*⟩ This option (set to false by default) causes **`\includeexercise`** to set the name of an exercise as the filename (without extension). This has two purposes: If you have nice file names you can automatically name the exercise. On the other hand this is helpful during exercise sheet composition because it shows the name of the corresponding file in the compiled file.

3.2 Supplying Meta-Data

The following commands are enhanced (or provided) to set the options controlling the sheet headers (see Section **??**):

`\subject{`⟨*subject*⟩`}`
`\author{`⟨*author*⟩`}`
`\date{`⟨*semester*⟩`}`

If one of these commands is omitted (and the corresponding option is not used either), a warning is issued.

Please do not include a `\title` in your document, as `exercisesheets` will automatically generate an appropriate one (this may require an additional LaTeX run). If you want to set the title yourself, set the option `settitle` to `false`.

All this information is included in the PDF meta-data if the `hyperref` package is loaded with the option `pdfusetitle`.

There is also a related option:

`exauthor=`⟨*list of names*⟩ If you want to use the `\author` macro for the general author of a course but there are different authors for individual exercises or all exercises in general, you can set this key. Currently only exercisesheets-baemer uses this for the footline, while `author` is used for the headline.

3.3 Defining Exercise Sheets

```
\begin{sheet}[⟨options⟩]
  ⟨sheet contents⟩
\end{sheet}
```

Insert a sheet into the document. This environment can be repeated to combine several sheets in a single LaTeX file. For each sheet, a new page is started and an appropriate header is generated. The ⟨*sheet contents*⟩ can be anything but will usually consist of several `exercise` environments (see Section **??**).

The following ⟨*options*⟩ are supported:

`date=`⟨*date*⟩ Set the date the sheet was/will be issued. This information is included in the sheet header. By default, this information is omitted. See also the `semester` option.

`note=`⟨*note*⟩ Include ⟨*note*⟩ in the sheet header. Useful to inform students when the sheet is due. If you want a note consisting of more than one line split at a particular position, use `\protect\linebreak`.

`title=`⟨*title*⟩ Directly set the sheet title. When this option is used, different page numbering conventions are used. This is useful to typeset exams (combined with the next option).

`number within sheet=`⟨*true/false*⟩ Deviate from the usual numbering theme and restart from one for exercises on the sheet. The previous counter value is restored after the sheet, so you can insert a special sheet.

`author=`⟨*author*⟩ Set the author included in the sheet head. By default, the value passed to `\author` is used.

`exauthor=`⟨*exauthor*⟩ Only used by the beamer variant, see above.

`subject=`⟨*subject*⟩ Set the subject included in the sheet head. By default, the value passed to `\subject` is used.

`semester=`⟨*semester*⟩ Set the semester included in the sheet head. By default, the value passed to `\date` is used.

`beamerwithheadline=`⟨*true/false*⟩ Controls whether a headline with author, subject etc. is shown on beamer slides, similar to the regular sheets. This options is off by default as headlines (and footers) take a considerable amount off space. If you specify the embedded option the current default headers and footers will be used if you omit this option. Otherwise omitting it will imply an empty headline.

`beamerwithfootline=`⟨*true/false*⟩ The same for the footer.

`beameruseblocks=`⟨*true/false*⟩ Controls whether the exercise title is shown inside a beamer block (⟨*true*⟩) or a simple colorbox (⟨*false*⟩, the default). Depending on your style a block might look fancier, but it usually consume more space.

3.4 Defining Exercises and Solutions

```
\begin{exercise}[⟨options⟩]
  ⟨exercise text⟩
  \begin{solution}[⟨solution options⟩]
    ⟨solution text⟩
  \end{solution}
  \begin{beamersolution}[⟨solution options⟩]
    ⟨solution text⟩
  \end{beamersolution}
\end{exercise}
```

This inserts an exercise into the current document. All **`beamersolution`**s are ignored, if the `exercisesheets` package with any class but beamer, but using beamer `exercisesheets` also processes normal solutions by default (set option `beamersolution` to turn this off).

The following options are supported:

`name=`⟨*text*⟩ Use ⟨*text*⟩ as the name of the exercise. Useful for exercises that prove a famous theorem.

`firstline=`⟨*text*⟩ Save some space by text ⟨*text*⟩ behind the exercise title.

`savetasks=`⟨*true/false*⟩ Saves the main task and each sub task for later use with **`\restatetask`**[⟨*which*⟩], where ⟨*which*⟩ is either `main` (default) or the number of a sub task.

`points=`⟨*number/oral/sum*⟩ Assign this exercise ⟨*number*⟩ points. By default, exercises are unlabelled. ⟨*oral*⟩ works the same way as the option `oral`. The value ⟨*sum*⟩ displays the sum of all occurrences of **`\points`** within the exercise: Ordinary and bonus points are treated separately. If there are only bonus points, the option `bonus` is automatically triggered. To undo this, either delete the `.aux` file or explicitly use `\points[bonus=false]{\param{number}}` for at least one subexercise. Points can be summed up also during a single pass by using the Lua interface.

`oral` Label this exercise as *oral*. This supersedes and is superseded by the option `points`.

`pointsinfo=`⟨*text*⟩ Supply ⟨*text*⟩ as additional information to be displayed after the points.

`optional` Shortcut for `pointsinfo=optional` with automatic translation and abbreviation (if requested).

`bonus` Change “points” to “bonus points” (with automatic translation and abbreviation).

`abbrev` Use abbreviated labels.

`exercisemark=`⟨*symbol*⟩ Mark the exercise with ⟨*symbol*⟩ in the left margin.

`difficult` Shortcut for `exercisemark=`$*$.

`solutions=`⟨*true/false/oral/nonoral/...*⟩ Use this to override the document (or sheet) default.

`sample` Include this exercise (and its solution) in while compiling sample solutions.

`beamersolution` For beamer variant only. If this option is set all normal solutions will not be included and only **`beamersolution`**s are typeset.

`framed` For beamer variant only, simply ignored elsewhere. If this option is set for a normal solution, its content will be put on one or more frames. Use **`\newframe`** to start a new frame. This option is especially helpful if you provide your own definitions of `\only` etc. and/or use the `beamercompatibility`. for non-beamer compilation. **`\newframe`** is already defined as `\relax` in non-beamer `exercisesheets`, even without invoking `beamercompatibility`.

`fragile` For beamer variant only and only used combined with framed, simply ignored elsewhere. Frames are declared as `fragile` und parsing of **`\newframe`** etc. is done via LuaLaTeX. Thus LuaLaTeXis required (and this requires font setup via fontspec etc.), but you do not have to use any Lua code yourself. Mostly useful for listings and other verbatim stuff within solutions.

Solutions are only typeset, if the `solutions` option is in effect. There can be multiple solution environments within a single exercise environment; this is useful if the exercise consists of several sub-exercises. Sub-exercises can simply be defined with an `enumerate` or `compactenum` environment.

4 Utilities

4.1 Loading Excercises From Files

`\includeexercise`⟨*file name*⟩
`\includeexercise*`⟨*file name*⟩

Both load the exercise from exercisespath/⟨*file name*⟩. Note that the exercise environment must be contained in the file. The starred version is helpful for faster skipping of unused exercises. In this case, the file is not opened to search for labels. Whenever the exercise is not skipped and you use the starred version and also use labels within the exercise file, `exercisesheets` outputs an error. This is done to ensure that references are not overlooked when the exercise is actually skipped later.

4.2 Including Hints

```
\begin{hint}[⟨options⟩]
  ⟨hint text⟩
\end{hint}
```

```
\begin{hint*}[⟨options⟩]
  ⟨hint text⟩
\end{hint*}
```

These environments include hints in the exercise definition. The first form starts a new paragraph, the second one puts the hint in parenthesis.

4.3 TODO Markers

You can use the following to include a red TODO marker in your document. This is useful to mark places where work is still in progress. A warning is issued at each place. TODO markers can also contain an optional description of the task that needs to be done. If you switch off displaying TODO markers (s.a.) the warning will be issued nevertheless.

```
\TODO[what needs to be done]
```

4.4 Annotating Points for Sub-Exercises

```
\begin{exercise}[points=sum]%sum produces 4+6
  \begin{enumerate}
   \item Part 1 \points{oral}
   \item Part 2 \points[optional]{oral}
   \item Part 3 \points{4}
   \item Part 4 \points[bonus]{6}
  \end{enumerate}
  Sub-Exercises within continuous text can be annotated like this
      \points[inplace]{3} without adding space.
\end{exercise}
```

4.5 Explicitly Stating Tasks

In principle, `exercisesheets` aims to be very lightweight and there is not much mandatory structure inside exercises. Furthermore, the `enumi` counter is used for sub-exercise. There are, however, situations where we want more explicit structure, for instance to have special font for the main task or the subtasks or to reuse some of the tasks later. This is especially handy if using the beamer variant where the original task and a part of the solution might be on different slides. In the future it might be possible to use other counters then `enumi`, which is also only possible if sub-exercises are not simply `\items` in `enumerate`. If you want to reuse the main or any subtasks you may use `\restatetask` or `\restatetaskbeamer` (this shows noting if not compiled with beamer). There is also a starred version (`\restatetask*` / `\restatetaskbeamer*`) for each of them, which adds a newline. All four have an optional argument that allows to restate a subtask instead of the main task. This argument is either `main`, a number of a **previous** subtask or `cur` for the current subtask. Using `headerrestate` for your solution will automatically restate the current task within the beamer solution header (does noting without beamer).

```
\begin{exercise}[points=oral]
  \begin{maintask}
    A and B are true.
  \end{maintask}
  \begin{subtasks}
    \subtask{A is true}
      \begin{solution}[framed]
        We want to show: \restatetask %A and B are true.
        \newframe
        Out first step is to show: \restatetask[cur] %A is true
      \end{solution}
    \subtask{If A is true, B is true.}
      \begin{solution}[framed]
        Obvious (i.e.: homework).
      \end{solution}
    \subtask{B is true}
      \begin{solution}[framed]
        We already know: \restatetask[1] and
        \restatetask[2], therefore B is also true.
      \end{solution}
  \end{subtasks}
\end{exercise}
```

4.6 Labels and References within an Exercise

While the global reference labels set via varioref (if loaded) are nice for references to subexercises far away, always mentioning the exercise number seems superfluous. Furthermore, for usage in indices etc. you might want to use the arabic representation instead. This is what the commands `\subexnref` and `\subexlref` are designed for. However, both of them still require an ordinary label, which is not allowed in exercises loaded with `\loadexercise*`. For this purpose `\inexlabel` can be used which is just a normal label that is not redefined to produce an error. Furthermore, we define a counter `subex` (which is currently just an alias for enumi, but this might change). You can use this counter (alias) with common commands like `\arabic` or `\alph`.

```
\begin{exercise}
  \begin{enumerate}
```

```
    \item \inexlabel{subex:xisseven} Let $x_{\thesubex}=7$.
    \item Compute $x_{\subexnref{subex:xisseven}}+3$.
    \begin{solution}
      From \subexlref{subex:xisseven} %(a)
      we know that $x_{\subexnref{subex:xisseven}}$ is $7$ and
      thus the sum is $10$.
    \end{solution}
  \end{enumerate}
\end{exercise}
```

5 Advanced Usage

5.1 Setting Options

Options can be given at different places.

1. As local options to one of the environments.
2. As package options: This is convenient for global options but suffers from shortcomings in way LaTeX processes options: Macros are expanded and spaces are stripped.
3. By the independent **`\exshset`** command. This is especially useful in the preamble to set options that would be garbled by the LaTeX option handling routine. It also allows to change an option for the rest of the current scope.

Example: Change the solution authors of the following sheets:

⟨*some sheets*⟩
`\exshset{solutionsby=`⟨*other authors*⟩`}`
⟨*more sheets*⟩

5.2 Changing Strings

For some languages, predefined sets of strings are provided and automatically activated. If your language is not supported or if you want to change (some of) the used strings, you can do so with the following options:

`strings/sheet=`⟨*string*⟩

`strings/sheets=`⟨*string*⟩

`strings/solutions=`⟨*string*⟩

`strings/solutionsby=`⟨*string*⟩

`strings/exercise=`⟨*string*⟩

`strings/solution=`⟨*string*⟩

`strings/hint=`⟨*string*⟩

`strings/oral=`⟨*string*⟩

`strings/oral abbrev=`⟨*string*⟩

`strings/point=`⟨*string*⟩

`strings/points=`⟨*string*⟩

`strings/points abbrev=`⟨*string*⟩

`strings/bonus point=`⟨*string*⟩

`strings/bonus points=`⟨*string*⟩

`strings/bonus points abbrev=`⟨*string*⟩

`strings/optional=`⟨*string*⟩

`strings/optional abbrev=`⟨*string*⟩

Example: Give an introduction that should only be included in the version with solutions:

```
\begin{solution}[strings/solution=Introduction]
  ⟨introduction text⟩
\end{solution}
```

5.3 Changing Fonts

The package `exercisesheets` comes with its own way to change the used fonts. Each font can be changed in the following way:

```
\exshset{⟨font element⟩=⟨font specification⟩}
```

The available ⟨*font element*⟩s are listed below together with their default values:

`sheet header font` The basic font for subject, author, semester, date, note and solution authors in the sheet headers.
Default: `\normalfont\normalsize`

`subject font` The font for the subject in the sheet header.
Default: `\scshape`

`author font` The font for the author in the sheet header.
Default: `\scshape`

`semester font` The font for the semester in the sheet header.
Default: empty, i. e. no change.

`date font` The font for the date in the sheet header.
Default: empty, i. e. no change.

`solutionsby font` The font for the information who has produced the solutions provided below the sheet title.
Default: `\itshape`

`sheet note font` The font for the note provided below the sheet title.
Default: `\itshape\bfseries`

`sheet title font` The font for the sheet title itself.
Default: `\Large\bfseries`

`exercise title font` The font for the exercise title.
Default: `\bfseries`

`points font` The font for the number of points in the exercise head (relative to the exercise title) and for **`\points`**.
Default: `\itshape`

`main task font` The font for the **`maintask`** environment.
Default: empty, i. e. no change.

`subtask font` The font for each **`\subtask`**.
Default: empty, i. e. no change.

`task restate font` The font used for **`\restatetask`**.
Default: `\itshape`

`hint font` The font for hints.
Default: empty, i. e. no change.

`hint title font` The font for the string “Hint:”.
Default: `\itshape`

`solution font` The font for solutions.
Default: empty, i. e. no change.

`solution title font` The font for the string “Solution:”
Default: `\bfseries`

`grading guide font` The font for grading guides.
Default: `\itshape`

`todo marker font` The font for the string “TODO”.
Default: `\ifdef{\color}{\color{red}}{}\bfseries`

5.4 Controlling the Spacing, Page Handling and Separations between Elements

The following options allow fine-tuning of the spacing:

`below slide headline skip=`⟨*dimen*⟩ The distance between haedline (if present) and slide content. Only used by beamer variant, ignored elsewhere.
Default: `0mm`

`above sheet title skip=`⟨*dimen*⟩ The distance between author/date and sheet title.
Default: 4ex

`above sheet note skip=`⟨*dimen*⟩ The distance above the sheet note.
Default: `.7ex`

`above solutionsby skip=`⟨*dimen*⟩ The distance above the solution author.
Default: 1ex

`below sheet header skip=`⟨*dimen*⟩ The distance below the sheet header.
Default: `4ex plus 1ex minus .5ex`

`above exercise skip=`⟨*dimen*⟩ The distance above exercises.
Default: `3ex plus 1ex minus .5ex`

`below exercise title skip=`⟨*dimen*⟩ The distance below exercise titles.
Default: `\parskip`

`above solution skip=`⟨*dimen*⟩ The distance above solutions.
Default: 1ex

`above hint skip=`⟨*dimen*⟩ The distance above hints.
Default: 1ex

The following two options control the page handling at the beginning and at the end of each sheet:

`sheet start page action=`⟨*macro*⟩ Executed at the beginning of each sheet.
Default: `\clearpage`

`sheet end page action=`⟨*macro*⟩ Executed at the end of each sheet.
Default: `\clearpage`

The following option controls what follows after the string `strings/solution` within a solution header. Using this, you can disable a solution header completely or, for instance, add a horizontal bar below each solution title.

`solution title separator=`⟨*content*⟩ Inserted directly after `strings/solution` in a solution title:
Default: `:\par`

5.5 Controlling the Sectioning (beta)

By default, exercises are sections and sheets are parts. They use the respective counters and commands. This is impractical, if you want to include exercise “sheets” as, e.g., a subsection within a section of a book or article. While the page handling options already allow to dismiss page breaks, sectioning commands might still break your layout or structure. Since version `0.15` you can use your own counters and commands to start sheets and (to a certain degree) exercises. Exercises are currently still started by `\@startsection` with the counter set to `exercise counter`, but the level set to 1 (e.g. section). This will likely change with one of the next versions, but needs some testing. Thus, if you use any of these options, watch for undesired consequences and report them via gitlab if possible.

`sheet sectioning cmd=`⟨*macro*⟩ This macro is called only in one place in the following way: `\macro[short title]{long title}`, so it must take one argument in [brackets] and one normal argument.
Default: `\part`

`sheet counter=`⟨*counter*⟩ This counter will be incremented wit every sheet. There was always an internal counter `sheetid` used to distinguish sheets for hyperref even if `part`s value changed. Do not use this counter here, as it will have strange effects (incremented twice).
Default: `part`

`exercise counter=`⟨*counter*⟩ This counter will be incremented wit every exercise (usually within each sheet). Default: `section`

5.6 Special Code for Solutions

```
\ifsolutions{⟨if true⟩}{⟨if false⟩}
\iforalsolutions{⟨if true⟩}{⟨if false⟩}
\ifnonoralsolutions{⟨if true⟩}{⟨if false⟩}
\ifsamplesolutions{⟨if true⟩}{⟨if false⟩}
```

There are also special options for conditionals to be used in the arguments of the environments defined by this package:

`ifsolutions={`⟨*options if true*⟩`}{`⟨*options if false*⟩`}` Execute ⟨*options if true*⟩ if solutions are included in the current document, ⟨*options if false*⟩ otherwise.

`iforalsolutions={`⟨*options if true*⟩`}{`⟨*options if false*⟩`}` Execute ⟨*options if true*⟩ if solutions for oral exercises are included in the current document, ⟨*options if false*⟩ otherwise.

`ifnonoralsolutions={`⟨*options if true*⟩`}{`⟨*options if false*⟩`}` Execute ⟨*options if true*⟩ if solutions for non-oral exercises are included in the current document, ⟨*options if false*⟩ otherwise.

`ifsamplesolutions={`⟨*options if true*⟩`}{`⟨*options if false*⟩`}` Execute ⟨*options if true*⟩ if sample solutions for marked exercises are included in the current document, ⟨*options if false*⟩ otherwise.

Example 1: Only include points for sub-exercises when solutions are typeset:

```
\ifsolutions{}{\renewcommand{\points}[2][]{}}
```

Example 2: Modify the sheet header spacing in the non-solution version:

```
\begin{sheet}[ifsolutions={}{above title skip=2ex}]% usually 4ex
  \dots
\end{sheet}
```

5.7 Using Hooks

There are several hooks used by `exercisesheets`:

`every sheet` This is used at the beginning of every sheet.

`every exercise` This is used at the beginning of every exercise.

`every solution` This is used at the beginning of every solution.

`every hint` This is used at the beginning of every hint.

Hooks can be used to influence the behaviour of the respective environments. Users of `tikz` should be familiar with the concept.

5.8 Control Skipping of Custom Macros in Skipped Exercises

`custom skip macro=`⟨*macro*⟩

If an exercise is not printed, some macros like `\label{}` are processed nevertheless. However, if you define your own macro using one of these, then this macro will be completely ignored if the exercise containing it is skipped. This option allows to define a custom handler for your macros. The most common use checks for your custom macros with nested `\ifstrequal`s and then either executes a custom skipper, replaces your macro with the standard version or uses `\expandafter` to deliver the expanded version of your macro to `\exshskipcontinue`.

Example:

```
\def\mylabel#1{...}
\def\myitem{...}
\def\myitemtwo{...}
\def\skipmylabel#1{...\exshskipcontinue}
\long\def\customskip#1{
  \ifstrequal{#1}{\mylabel}
    {\skipmylabel}
    {\ifstrequal{#1}{\myitem}
      {\exshskipcontinue\item}
      {\ifstrequal{#1}{\myitemtwo}
        {\expandafter\exshskipcontinue\myitemtwo}
        {\exshskipcontinue}
      }
    }
}
\exshset{custom skip macro={\customskip}}
```

6 The Lua Interface

Since version `0.11` there is a Lua interface for the exercise environment. This interface will offer roughly the same features. Its main purpose will be allowing to reorder sub-exercises more easily and to offer more dynamic options for the display of solutions (one combined solution or single after each sub-exercise).

```
\begin{Lexercise}
  --use [[]] if you need \ or escape it in "": "\\"
  firstline = [[Assume $\pi=4$.]],
  points=10,
  name="Pragmatic",
  options=[[main task font={\itshape}]],
  task = [[
    This is the main task specified via the Lua
    interface.
  ]],
  solution=[[
    This is a solution for the main task.
  ]],
\end{Lexercise}
```

7 Usage Tips

7.1 Seperate Solution File

If you do not want to temporarily comment out the `solutions` option in your main file, say `exercises.tex`, you can create an additional file `solutions.tex` with the following contents:

```
\PassOptionsToPackage{solutions}{exercisesheets}
\input{exercises.tex}
```

If you leave out the `solutions` option in you main file, running `pdflatex exercises.tex` will create `exercises.pdf` without solutions and `pdflatex solutions.tex` will create `solutions.pdf` with solutions.

This also works well in combination with a makefile that generates `solutions.tex` as a temporary file.

7.2 Compatibilty with make4ht and tex4ebook

Since version `0.15` this package can be used with make4ht and tex4ebook in non-beamer mode. Some code for the sheet and exercise header already provides a similar alignment as when producing PDF files. If your goal is an exercise (or solution) document within a single HTML file that looks as similar to the PDF as possible, consider `https://tex.stackexchange.com/questions/605478/`, `https://tex.stackexchange.com/questions/662445` and `https://tex.stackexchange.com/questions/630992`.

8 Changelog

v. `0.17`: 2024-04-23

- Make embedded mode work better with beamer. This includes not overriding headlines and footlines and incrementing the frame counter for non-frames.
- issue an explicit warning if \subtaskis outside of a subtasks environment
- clearify documentation and use consistent styling
- additional example for embedding exercises

v. `0.16`: 2024-03-07

- Embedded mode and most of the additional options it automatically sets.

v. `0.15.1`: 2023-12-14

- bugfix: beamer support was broken in `0.15` due to a missing macro.

v. `0.15`: 2023-11-30

- new option `solution title separator` (initially set to`:\par`). Among other things, this allows for solutions without any caption or header by setting `solution title separator`, `solution title font` as well as `strings/solution` to `\relax`.
- initial support for `make4ht` and `tex4ebook` by avoiding certain behavior when these modes are used. Note that both are incompatible with beamer mode.
- initial decoupling of exercise and sheet from sections and parts using the new options (defaults in parenthesis) `sheet sectioning cmd` (`\part`), `sheet counter` (`part`) and `exercise counter` (`section`).

v. `0.14`: 2023-02-06

- new option `headerrestate` to restate the current (if in subexercise) or main task within a solutions header
- bugfix: **`\restatetask`** etc. give meaningful error if `savetasks` is not set.

v. `0.13.1`: 2022-11-22

- bugfix: `\detokenize` filenames when `filenameasexercisename` is active as otherwise, e.g., underscores would not be allowed in filenames.

v. `0.12.2`: 2022-10-20

v. `0.13`: 2022-11-07

- bugfix: `\begingroup` and `\endgroup` within **`\subtask`** now contain everything up to the next **`\subtask`**.

- `\restatetask` has now a starred version that adds a newline and beamer versions for both the regular and the starred version. Furthermore, there is an argument `cur` to restate the current subtask (the one just given before). This is only defined within the aforementioned group, i.e., throws an error after `\end{subtasks}`.

v. `0.12.2`: 2022-10-20
- bugfix: `dofile` with `kpse.find_file`
- docs: document option `fragile`

v. `0.12.1`: 2022-10-19
- bugfix for `\begin{enumerate}[<+->]`

v. `0.12`: 2022-09-02
- new option `beamerwithfootline`

v. `0.11`: 2022-02-11
- Local references and the subex counter
- Lua interface (not documented yet, see example and example file)
- Framed solutions can now be fragile, however this also requires LuaLaTeX(but not using the Lua Interface).
- beamer version is now included (no separate package)
- some bug fixes