# Table of Contents

1. Licence, Requirements and readme .................................................. 2

2. Motivation, Background ..................................................................... 3

3. How to Read the Manual ................................................................... 3
   3.1. Nomenclature ............................................................................. 3
   3.2. Package Options ......................................................................... 4
   3.3. Setting Options ........................................................................... 5
   3.4. Command descriptions ............................................................... 5

4. Exercises and Solutions ..................................................................... 6

5. How the Exercise Environments Work .............................................. 7

6. New Exercise Types ........................................................................... 9

7. Exercise Properties ............................................................................ 12
   7.1. Predefined Properties ............................................................... 12
   7.2. Declaring Own Properties ........................................................ 14
   7.3. Exercise Goals .......................................................................... 15
   7.4. Exercise Tags ............................................................................. 18

8. Using and Printing an Exercise .......................................................... 19
   8.1. What the Environments do ......................................................... 19
   8.2. Environment Options & Hooks ................................................... 20

8.3. (Re-) Inserting a Certain Exercise .................................................. 21

9. Collecting Exercises ......................................................................... 22
   9.1. Background .............................................................................. 22
   9.2. Usage ........................................................................................ 22
   9.2.1. Declaring collections ............................................................ 22
   9.2.2. Activating collections ............................................................ 23
   9.2.3. Printing collections ................................................................. 23

10. Printing Random Exercises From a Collection .................................. 26

11. Printing Solutions ............................................................................. 27

12. Grading Tables ................................................................................ 29

13. Styling the Exercises – Templates .................................................. 31
   13.1. Background ............................................................................. 31
   13.2. Templates Provided by the Package ......................................... 31
   13.3. Commands for Usage in Template Definitions ........................ 33
   13.3.1. Goals .................................................................................. 33
   13.3.2. Properties .......................................................................... 34
   13.3.3. Parameters ......................................................................... 35
   13.3.4. Tags .................................................................................... 36
1. Licence, Requirements and README

Permission is granted to copy, distribute and/or modify this software under the terms of the \TeX{} Project Public License (\lppl{}), version 1.3c or later (http://www.latex-project.org/lppl.txt). The software has the status "maintained."\texttt{xsim} loads the packages expl3 [L3Pa], xparse [L3Pb], array [MC21] booktabs [Fea20] and translations [Nie22]. All of these packages are present on a modern and up to date \TeX{} distribution such as \TeX{} Live or MiK\TeX{} so no further action should be needed. When you are using \texttt{xsim} you should be using an up to date \TeX{} distribution, anyway.

1. Licence, Requirements and README

13.3.5. Further Commands for Usage in Template Definitions . . . 36
13.4. Declaring Templates . . . . 39
13.4.1. Environment Templates 39
13.4.2. Heading Templates . . 40
13.4.3. Grading Table Templates . . . . . . . . 40
13.5. Create and Use \texttt{xsim} Style Files 40
13.6. Examples . . . . . . . . . . . . 41
13.6.1. The default Exercise Template . . . . 41
13.6.2. A New Exercise Type Using \texttt{tocolorbox} . . . . 42
13.6.3. Mimicking exsheets’ runin Template . . . . 43
13.6.4. Mimicking exsheets’ margin Template . . . . 44
13.6.5. A minimal Template . . 44
13.6.6. The Headings Templates . . . . . . . . 45
13.6.7. The default Table Template . . . . . . . . 45
13.6.8. The default* Table Template . . . . . . . . 46

14. Exercise Translations 48

15. Cloze Tests and Blank Lines 51

A. Future Plans 53

B. FAQ & How to... 53
B.1. ...Know if \texttt{xsim} Needs Another Compilation? ... 53
B.2. ...Resolve Getting Repeatedly Wrong Exercise Properties or Wrong Exercise Lists? ... 54
B.3. ...Resolve Strange Errors After Updating? ... 54
B.4. ! \TeX{} capacity exceeded, sorry [text input levels=15]. Why? ... 54
B.5. Runaway argument? !File ended while scanning use of ^^M. Why? ... 54
B.6. ...Put a Star (or Another Symbol) in Headings of Exercises That Are Special? ... 54
B.7. ...Print All Solutions Grouped by Section? ... 55

C. The \texttt{xsimverb} package 57

D. All Exercise Examples 58

E. All Solution Examples 60

F. Example Documents Coming With This Package 61

G. References 82

H. Index 84
Please be aware that \texttt{xsim} is in an experimental state and actively developed. Many aspects may change from one update to another until a stable version 1 will be reached. However, I will try my best to keep the interface stable. Newer versions of \texttt{xsim} may depend on newer versions of the support packages. Remember: it is always dangerous to update single packages. Always update your \LaTeX{} distribution if you want an up to date version of a package. Be careful: if you’re in the middle of an important project it might be better to wait with the update until you’ve finished the project. Every update might be breaking some things.

## 2. Motivation, Background

It has been quite a while since I first published \texttt{exsheets} [Nie19] in June 2012. Since then it has gained a user base and a little bit of popularity as the number of questions on tex.sx shows (151 at the time of writing) [var]. User questions, bug reports and feature requests improved it over the time. It still has a version number starting with a zero, though, which in my versioning system means I still consider it experimental. This is due to several facts. It lacks a few features which I consider essential for a full version 1. For one thing it is not possible to have several kinds of exercises numbered independently. Using verbatim material such as listings inside exercises and solutions is not possible and the current workaround isn’t that ideal either. One request which dates back quite a while now was to have different types of points to exercises…

All of those aren’t easy to add due to the way \texttt{exsheets} is implemented right now. As a consequence I wanted to re-implement \texttt{exsheets} for a long time. This is what lead to \texttt{xsim}. Internally the package works completely different.

\texttt{xsim} will be the official successor of \texttt{exsheets} which is now considered obsolete but will stay alive and will still receive bugfix releases. However, new features will not be added to \texttt{exsheets} any more.

## 3. How to Read the Manual

### 3.1. Nomenclature

Throughout this manual certain terms are used. This section explains their meaning in this manual.

\textbf{collection} A \textit{collection} bundles a number of exercises of one type or all types of exercises within certain barriers in the document. Those exercise collections can be printed at any place in the document.

\textbf{goal} \textit{Goals} are a certain type of properties with a numerical value the sum of which is available throughout the document.
3. How to Read the Manual

**parameter**  *Parameters* are options of exercise types which are the same for each exercise of a type and can be retrieved and used in exercise templates.

**property**  *Properties* are options of exercises which are individual for each exercise and can be retrieved and used in exercise templates.

**tag**  *Tags* are a certain type of properties with a csv list as value which can be used for selective usage of exercises.

**template**  *Templates* are generic code frameworks which are used for typesetting *xsim*’s objects such as exercises, solutions, or grading tables.

### 3.2. Package Options

*xsim* has these package options:

**verbose**
Writes extensive information about what *xsim* is doing into the log file.

**final**
If used the exercise and solution environments will not rewrite the environment body files.

**clear-aux**
If used every time the total number of exercise changes *xsim* will write less information to the auxiliary file on the next run and only if the number of exercises stays stable between compilations the needed information will be written to the auxiliary file. *This needs more compilations until everything stabilizes but should reduce the probability of possibly faulty exercises after changes to the document.* The final option automatically disables this option. See also sections 5 on page 7 and B.2 on page 54.

**no-files**
This option prevents *xsim* from writing the exercises and solutions to external files. This will keep your working folder “clean” but will also prevent using verbatim material in exercises and solutions. 

**use-files**
This is the opposite of the option no-files.

**use-aux**
With this option enabled *xsim* will use the regular auxiliary file \jobname.aux instead of its own auxiliary file \jobname.xsim.

**blank**
With this option enabled *xsim* will not define the default environments exercise and solution.

Those options are load-time options and are used the usual way as package options:

---

*Introduced in version 0.15 (Nov 2, 2019)*

*Introduced in version 0.20 (Jan 31, 2021)*

*Introduced in version 0.13 (Oct 6, 2019)*

*Introduced in version 0.17 (Feb 21, 2020)*
3. How to Read the Manual

\usepackage[use-files]{xsim}

Although those options technically belong to the package module (see also section 3.3) it is not possible to set them via \xsimsetup.

3.3. Setting Options

Apart from the package options already described in section 3.2 on the previous page \texttt{xsim} has further options. All those options are set using the following command:

\xsimsetup{⟨options⟩}

Set up \texttt{xsim}'s package options and all other options described at other places in the manual.

Options can be “toplevel” options or options belonging to a module:

\texttt{toplevel = ⟨value⟩}

A toplevel option.

\texttt{module/sublevel = ⟨value⟩} \texttt{module} A sublevel option belonging to the module \texttt{module}

Both kinds of options are set with the setup command:

\begin{verbatim}
\xsimsetup{
  toplevel = {value} ,
  module/sublevel = {value}
}
\end{verbatim}

3.4. Command descriptions

Some commands do have a ∗ symbol printed next to their names. This indicates that the command is expandable, \textit{i.e.}, it is usable in an \texttt{edef} or \texttt{write} context and will expand according to its description. All other commands are engine protected, \textit{i.e.}, in the sense of \texttt{e-\TeX}'s \texttt{protected}.

Some command name descriptions end with \texttt{TF}.

\texttt{SomeCommandTF⟨arguments⟩}{⟨true⟩}{⟨false⟩}

A command with maybe some arguments and ending with the two arguments ⟨true⟩ and ⟨false⟩.

This means two things: the command is a conditional which tests something and depending on the outcome of the test leaves either the ⟨true⟩ argument (T) or the ⟨false⟩ argument (F) in the input stream. It also means two additional commands exist:
4. Exercises and Solutions

The two predefined environments for exercises and solutions are the following ones:

\begin{exercise} \langle \text{properties} \rangle \end{exercise}

Input and typeset an exercise. See section 7 on page 12 for details on exercise properties.

\begin{solution} \langle \text{options} \rangle \end{solution}

Input and typeset the solution to the exercise of the previous exercise environment. See section 11 on page 27 for details on options of solutions.

Exercise 1

A first example for an exercise.

As can be seen in the example a solution is not printed with the default setup. This can be changed using the following option.

solution/print = \text{true}|\text{false} \\
\text{Default: false}

Set if solutions are printed or not.

The option (belonging to the module solution) can either be set locally as option to the solution environment

\begin{solution}[print=true] \end{solution}

or with the setup command for all following solutions:

\begin{solution}[\text{print=true}] \end{solution}

---

1. When you load \texttt{xsim} with the \texttt{blank} those environments will \textit{not} be defined!
5. How the Exercise Environments Work

There is an completely analogous option for the exercise environment:

\begin{verbatim}
\xsimsetup{
  solution/print = true
}
\end{verbatim}

exercise/print = true|false [Default: true]

Set if exercises are printed or not.

More details on those two environments can be found in section 8 on page 19.

5. How the Exercise Environments Work

Depending on the options no-files/use-files the bodies of exercises and solutions is either
just another property (see section 7 on page 12) – this is the default behavior – or is written to
an external file.

If the option no-files = {true} is set the exercise and solution bodies are saved as properties
exercise-body and solution-body. This is the default behavior of \texttt{xsim} since version 0.20
and should work fine in most cases. One obvious downsise is that verbatim material cannot be
used in exercises and solutions.

If the option use-files = {true} is set both the exercise and the solution environments
write the contents of their bodies verbatim to external files following a certain naming structure:

- \langle\text{jobname}\rangle-\langle\text{type}\rangle-\langle\text{id}\rangle-exercise|solution-body.tex

The name starts with the name of the job (which is the name of the document itself) followed
by type and id of the corresponding exercise and then followed by the environment type. For
example both environments from the first example have been written to files named

- \texttt{xsim-manual-exercise-1-exercise-body.tex}
- \texttt{xsim-manual-exercise-1-solution-body.tex}

These external files are input when the respective exercise or solution is printed. An advantage
of using external files is that verbatim material is allowed inside the environments. Details
on the \langle\text{type}\rangle of an exercise will be given in section 6 on page 9. The \langle\text{id}\rangle of an exercise is a
positive integer unique to each exercise environment regardless if the exercise is being printed or
used at all.

Each of those files contains some information about itself and where and why it was

defined:

\footnote{In this example the sourcecode line number is misleading as the example where the file was generated itself
was an external file where the exercise environment indeed was on line 1.}
5. How the Exercise Environments Work

Arguably one downside of the approach using external files for each exercise and its solution is that your project folder will be cluttered with files. In order to deal with this somehow `xsim` offers the following option:

**path = \{⟨path name⟩\}**

(Initially empty)

With this option a subfolder or path within the main project folder can be given. Exercises will be written to and included from this path. *The path must exist on your system before you can use it!* This document uses `path = \{exercises\}`.

**file-extension = \{⟨string⟩\}**

This option lets you choose the extension of the external files. Default: `tex`

Another thing to keep in mind: the environment in many ways works the same way as the `filecontents` environment. *This also means that you cannot have comments or labels or anything else on the first line of the environments!*

```latex
\begin{exercise}[points=2] % this comment will cause trouble
Lorem ipsum
\end{exercise}
```
6. **New Exercise Types**

\texttt{xsim} writes a lot of stuff to an auxiliary file called \texttt{⟨file name⟩.xsim} (or the common \texttt{⟨file name⟩.aux} if you use option \texttt{use-aux}) for re-using information on subsequent compilations. If you add exercises, change properties \textit{etc.} it might happen that wrong information is staying in the auxiliary file and is wrongly used by \texttt{xsim}. In such cases deleting the auxiliary file and doing a few fresh compilations may resolve your problems.

Sometimes the existence of exercise or solution files from earlier compilations may lead to wrong lists of exercises or solutions. In such cases it can be useful to delete all those files and doing a fresh compilation. It may be helpful to use a subfolder for those external files which will make deleting them a little bit easier. (Don’t forget to both create the subfolder and set \texttt{path} accordingly then.)

Using the \texttt{clear-aux} option might help to reduce erroneous exercises.

A lot of the lines \texttt{xsim} writes to the auxiliary file and reads in a subsequent run look like this:

\begin{verbatim}
  \texttt{xsim(points){exercise-2==4||exercise-10==2.5||problem-11==5}}
\end{verbatim}

As you can see different entries of the various properties of exercises are separated with \texttt{||}. This means that you cannot use this symbol combination inside properties. For this reason \texttt{xsim} provides an option to change the marker.

\begin{verbatim}
\texttt{split-aux-lists }{\langle \text{string}\rangle}
\end{verbatim}

\texttt{split-aux-lists} = \texttt{\{⟨string⟩\}}

Set the string that is used to separate the property entries in the auxiliary file.

Introduced in version 0.11 (Feb 12, 2018)

6. **New Exercise Types**

It is easy to define new exercise environments together with a corresponding solution environment using the following command:

\begin{verbatim}
\DeclareExerciseType{⟨type⟩}{⟨parameters⟩}
\end{verbatim}

Declare a new exercise type analogous to the exercise and solution environments.

Declaring a new exercise type will also define a new command:

\begin{verbatim}
\numberof⟨exercise-env⟩\texttt{s}
\end{verbatim}

These commands hold the absolut number of used exercises of type \texttt{⟨type⟩}. The meaning of \texttt{⟨exercise-env⟩} will become clear below when the exercise parameters are explained. It is always the same as the exercise environment name.
6. New Exercise Types

There are \texttt{\numberofexercises-exercises} and \texttt{\numberofproblems-problem} in this manual.

There are 7 exercises and 1 problem in this manual.

\texttt{\textsc{xsim}’s pre-defined environment pair has been defined as follows:}

\begin{verbatim}
\DeclareExerciseType{exercise}{
exercise-env = exercise, 
solution-env = solution, 
exercise-name = \XSIMtranslate{exercise}, 
exercises-name = \XSIMtranslate{exercises}, 
solution-name = \XSIMtranslate{solution}, 
solutions-name = \XSIMtranslate{solutions}, 
exercise-template = default, 
solution-template = default, 
exercise-heading = \subsection{}, 
solution-heading = \subsection{}
}
\end{verbatim}

The above already is an example for almost all parameters that can (and often must) be set. Here is the complete list:

\texttt{\texttt{exercise-env} = \{⟨exercise environment name⟩\}}

The name for the environment used for the exercises of type \texttt{⟨type⟩}. \textit{This parameter is mandatory.} It can’t be changed afterwards.

\texttt{\texttt{solution-env} = \{⟨solution environment name⟩\}}

The name for the environment used for the solutions of type \texttt{⟨type⟩}. \textit{This parameter is mandatory.} It can’t be changed afterwards.

\texttt{\texttt{exercise-name} = \{⟨exercise name⟩\}}

The name of the exercises of type \texttt{⟨type⟩} – used for typesetting. \textit{This parameter is mandatory.}

\texttt{\texttt{exercises-name} = \{⟨exercises name⟩\}}

The plural name of the exercises of type \texttt{⟨type⟩} – used for typesetting. If this is not set explicitly an \texttt{s} is appended to the singular name.

\texttt{\texttt{solution-name} = \{⟨solution name⟩\}}

The name of the solutions of type \texttt{⟨type⟩} – used for typesetting. \textit{This parameter is mandatory.}

\texttt{\texttt{solutions-name} = \{⟨solutions name⟩\}}

The plural name of the solutions of type \texttt{⟨type⟩} – used for typesetting. If this is not set explicitly an \texttt{s} is appended to the singular name.
6. New Exercise Types

**exercise-template** = {⟨exercise template⟩}

The template used for typesetting the exercises of type ⟨type⟩. *This parameter is mandatory.*

See section 13 on page 31 for details on templates.

**solution-template** = {⟨solution template⟩}

The template used for typesetting the exercises of type ⟨type⟩. *This parameter is mandatory.*

See section 13 on page 31 for details on templates.

**counter** = {⟨counter name⟩}

The counter used for the exercises of type ⟨type⟩. If not explicitly set the counter with the same name as exercise-env is used. Otherwise the specified counter is used. This enables to have different types of exercises sharing a common counter. *This parameter can’t be changed afterwards.* If the explicit or implicit counter does not exist, yet, it will be defined.

**within** = {⟨counter name⟩}

Adds counter to the reset list of ⟨counter name⟩ by applying \counterwithin. ( Introduced in version 0.20 (Jan 31, 2021) )

**the-counter** = {⟨code⟩}

Redefines the corresponding representation command \the... of counter to ⟨code⟩. ( Introduced in version 0.20 (Jan 31, 2021) )

**solution-counter** = {⟨counter name⟩}

The counter used for the solutions of type ⟨type⟩. If not explicitly set the counter with the same name as solution-env is used. Otherwise the specified counter is used. This enables to have different types of solutions sharing a common counter although this doesn’t actually make much sense. But it can be useful to avoid using an already existing counter. *This parameter can’t be changed afterwards.* If the explicit or implicit counter does not exist, yet, it will be defined. The sole purpose of this counter is to be able to label solutions so they can be \pageref. ( Introduced in version 0.14 (Oct 13, 2019) )

**number** = {⟨integer⟩}

An internal parameter that is used to keep track of the number of exercises of a type. This parameter cannot be set or changed by the user. ( Introduced in version 0.14 (Oct 13, 2019) )

**exercise-heading** = {⟨exercise heading command⟩}

The command used for typesetting of the heading of exercises of type ⟨type⟩ – used for typesetting with the command \GetExerciseHeadingF. ( Introduced in version 0.14 (Oct 13, 2019) )

**solution-heading** = {⟨solution heading command⟩}

The command used for typesetting of the heading of solutions of type ⟨type⟩ – used for typesetting with the command \GetExerciseHeadingF.

It is possible to change some of the parameters after an exercise type has been defined. Those include exercise-name, solution-name, exercise-template, and solution-template. It is also possible to define new parameters.

\DeclareExerciseParameter*{⟨parameter⟩}

Declares the new parameter ⟨parameter⟩. The optional star declares a fixed parameter which cannot be changed once it is set. *You probably will never need this command. Most tasks can be solved using properties (see section 7 on the next page) instead.*
7. Exercise Properties

\SetExerciseParameter{\langle type\rangle}{\langle parameter\rangle}{\langle value\rangle}
Usable to set a single parameter to a new value.

\SetExerciseParameters{\langle type\rangle}{\langle parameters\rangle}
Set several parameters at once. \langle parameters\rangle is a csv list of key/value pairs.

If you try to set an already set but fixed parameter like exercise-env a warning will be written to the log file. For all parameters that can be changed also options exist which can be set via \xsimsetup. They are explained in section 8.2 on page 20.

All exercises of a type use the parameters (e.g., exercise-template) that are currently active. If you want exercises with a different look or different names in the same document you should use different exercises types.

7. Exercise Properties

7.1. Predefined Properties

Exercise like the exercise environment and possibly others defined with \DeclareExerciseType have a number of predefined properties:

id = \langle integer\rangle
Holds the internal id of an exercise. Cannot be set by the user.

ID = \langle text\rangle
Holds the user id of an exercise if defined. Otherwise it is equal to id.

counter = \langle text\rangle
Holds the counter value representation of an exercise (i.e., what you usually know as \the\{counter\}). Cannot be set by the user.

counter-value = \langle integer\rangle
Holds the counter value of an exercise (i.e., what you usually know as \the\{value\}\{counter\}). Cannot be set by the user.

subtitle = \langle text\rangle
Holds the subtitle of an exercise.

points = \langle number\rangle
Holds the reachable points of an exercise.

bonus-points = \langle number\rangle
Holds the reachable bonus-points of an exercise.

print = true|false
Holds the print boolean of an exercise.
7. Exercise Properties

print! = true|false
Holds a special print boolean of an exercise, see page 19.

use = true|false
Holds the usage boolean of an exercise.

use! = true|false
Holds a special usage boolean of an exercise, see page 19.

used = true|false
True if an exercise has been used at least once. For an existing exercise this is only false for exercises that have been collected (cf. section 9 on page 22).

solution = true|false
Holds the solution boolean of an exercise. If this is true then a solution has the same text/environment body as the corresponding exercise. (This might be useful for multiple choice questions for example.)

tags = ⟨csv list of tags⟩
Holds the list of tags the exercise should be associated with.

topics = ⟨csv list of topics⟩
Holds the list of topics the exercise should be associated with.

page = ⟨text⟩
Holds the page counter value representation of an exercise (i.e., what you usually know as \thepage).

page-value = ⟨integer⟩
Holds the page counter value of an exercise (i.e., what you usually know as \the\value{page}).

section = ⟨text⟩
Holds the section counter value representation of an exercise (i.e., what you usually know as \thessection).

section-value = ⟨integer⟩
Holds the section counter value of an exercise (i.e., what you usually know as \the\value{section}).

chapter = ⟨text⟩
Holds the chapter counter value representation of an exercise (i.e., what you usually know as \thechapter).
Only if a command \chapter and a counter chapter exist.

chapter-value = ⟨integer⟩
Holds the chapter counter value of an exercise (i.e., what you usually know as \the\value{chapter}).
Only if a command \chapter and a counter chapter exist.
\section*{7. Exercise Properties}

\texttt{sectioning = \{(section numbers)\}}

Holds five brace groups which in turn hold the section numbers (integers) of the exercise in the order \{(chapter)\}{\{(section)\}}{\{(subsection)\}}{\{(subsubsection)\}}{\{(paragraph)\}}.

\texttt{exercise-body = \{\LaTeX code\}}

When the package option \texttt{no-files} is set this property is defined and holds the environment body of an exercise.

\texttt{solution-body = \{\LaTeX code\}}

When the package option \texttt{no-files} is set this property is defined and holds the environment body of the corresponding solution.

Some of these properties are fixed and cannot be set by the user. Those include \texttt{id}, \texttt{counter}, and \texttt{counter-value}. The others can be set using the optional argument of the exercise environment.

\begin{exercise}[subtitle={This is a subtitle},points=4,bonus-points=1]
An exercise where some properties have been set.
\end{exercise}

\subsection*{Exercise 2 This is a subtitle}
An exercise where some properties have been set.

\section*{7.2. Declaring Own Properties}

\texttt{xsim} offers the possibility to declare additional exercise properties:

\texttt{\textbackslash DeclareExerciseProperty!*-\{(property)\}}

Declares the property \texttt{(property)}.

If used with the optional \texttt{!} a \texttt{unique property} is defined which means that each exercise must have a property value distinct from all other exercises (all means all – \textit{independent from the exercise type}).

If used with the optional \texttt{*} a \texttt{boolean property} is defined which means that it only should get the values \texttt{true} or \texttt{false} and if used without value it gets the value \texttt{true} instead of an empty value. If any other value is used the property is set to \texttt{false}. A boolean property obviously cannot be unique. The optional \texttt{*} takes precedence over the optional \texttt{!}, \textit{i.e.}, if both are present the property is boolean \textit{but not} unique.

If used with the optional \texttt{-} a property is defined which won’t get updated through subsequent compilation runs but is only set when the exercise is used.

\texttt{\textbackslash DeclareExercisePropertyAlias\{(property 1)\}\{(property 2)\}}

Declares \texttt{(property 1)} to be an alias of \texttt{(property 2)}. This means that each time \texttt{(property 2)} is set \texttt{(property 1)} will be set to the same value \textit{unless} it has been set already. As an example: property \texttt{ID} is an alias of property \texttt{id}.
This is better demonstrated with an example:

\begin{exercise}
\verb+\GetExerciseProperty{id}+: \GetExerciseProperty{id} \par
\verb+\GetExerciseAliasProperty{ID}+: \GetExerciseAliasProperty{ID} \par
\verb+\GetExerciseProperty{ID}+: \GetExerciseProperty{ID} \\
\end{exercise}

\begin{exercise}[ID=foo-bar]
\verb+\GetExerciseProperty{id}+: \GetExerciseProperty{id} \par
\verb+\GetExerciseAliasProperty{ID}+: \GetExerciseAliasProperty{ID} \par
\verb+\GetExerciseProperty{ID}+: \GetExerciseProperty{ID} \\
\end{exercise}

Exercise 3
\GetExerciseProperty{id}: 3
\GetExerciseAliasProperty{ID}: 3
\GetExerciseProperty{ID}: 3

Exercise 4
\GetExerciseProperty{id}: 4
\GetExerciseAliasProperty{ID}: 4
\GetExerciseProperty{ID}: foo-bar

The power of properties will get more clear when reading section 13 on page 31 about templates.

7.3. A Special Kind of Property: Exercise Goals

Exercise goals are a generic concept in xsim for exercise properties like points or bonus-points. Those are properties which can (only) get a decimal number as value the sum of which is calculated and available (after a compilation) throughout the document.

\DeclareExerciseGoal{⟨goal⟩}
Declare a new exercise goal named ⟨goal⟩ and also a property called ⟨goal⟩.

\TotalExerciseTypeGoal{⟨type⟩}{⟨goal⟩}{⟨singular⟩}{⟨plural⟩}
Get the sum of goal ⟨goal⟩ for all exercises of type ⟨type⟩. ⟨singular⟩ and ⟨plural⟩ are placed after the sum in the input stream depending on whether the sum equals 1 or not.

\TotalExerciseTypeGoals{⟨type⟩}{⟨list of goals⟩}{⟨singular⟩}{⟨plural⟩}
Get the sum of goal all goals in ⟨list of goals⟩ for all exercises of type ⟨type⟩. The goal names in ⟨list of goals⟩ must be separated with +. ⟨singular⟩ and ⟨plural⟩ are placed after the sum in the input stream depending on whether the sum equals 1 or not.
7. Exercise Properties

\TotalExerciseGoal\{\langle goal\rangle\}\{\langle singular\rangle\}\{\langle plural\rangle\}
Get the sum of goal \langle goal\rangle for all exercises. \langle singular\rangle and \langle plural\rangle are placed after the sum in the input stream depending on whether the sum equals 1 or not.

\TotalExerciseGoals\{\langle list of goals\rangle\}\{\langle singular\rangle\}\{\langle plural\rangle\}
Get the sum of goal all goals in \langle list of goals\rangle for all exercises. The goal names in \langle list of goals\rangle must be separated with +. \langle singular\rangle and \langle plural\rangle are placed after the sum in the input stream depending on whether the sum equals 1 or not.

\AddtoExerciseTypeGoal\{\langle type\rangle\}\{\langle goal\rangle\}\{\langle value\rangle\}
Adds \langle value\rangle to the goal \langle goal\rangle of exercise type \langle type\rangle. (To be used within exercises.)

\AddtoExerciseTypeGoal\Print\{\langle type\rangle\}\{\langle goal\rangle\}\{\langle value\rangle\}\{\langle singular\rangle\}\{\langle plural\rangle\}
Adds \langle value\rangle to the goal \langle goal\rangle of exercise type \langle type\rangle. The value and – depending on wether the value equals 1 or not – \langle singular\rangle or \langle plural\rangle are left in the input stream. (To be used within exercises.)

\AddtoExerciseGoal\{\langle goal\rangle\}\{\langle value\rangle\}
Adds \langle value\rangle to the goal \langle goal\rangle of the current exercise type. (To be used within exercises.)

\AddtoExerciseGoal\Print\{\langle goal\rangle\}\{\langle value\rangle\}\{\langle singular\rangle\}\{\langle plural\rangle\}
Adds \langle value\rangle to the goal \langle goal\rangle of the current exercise type. The value and – depending on wether the value equals 1 or not – \langle singular\rangle or \langle plural\rangle are left in the input stream. (To be used within exercises.)

\ExerciseGoalValuePrint\{\langle value\rangle\}\{\langle singular\rangle\}\{\langle plural\rangle\}
Print \langle value\rangle and – depending on wether the value equals 1 or not – \langle singular\rangle or \langle plural\rangle.

\Printgoals\{\langle value\rangle\}
Print \langle value\rangle according to option goal-print. Defined in terms of \ExerciseGoalValuePrint.

\Printpoints\{\langle type\rangle\}
Print the sum of points for all exercises of type \langle type\rangle followed by an appropriate translation of the words “point” or “points”, respectively. Defined in terms of \TotalExerciseTypeGoal.

\PrintTotalPoints
Print the sum of points for all exercises followed by an appropriate translation of the words “point” or “points”, respectively. Defined in terms of \TotalExerciseGoal.

\AddPoints\{\langle value\rangle\}
Adds \langle value\rangle to the points of the current exercise type. (To be used within exercises.) Prints the value followed by an appropriate translation of the words “point” or “points”, respectively. The starred version prints nothing. Defined in terms of \AddtoExerciseGoal and \AddtoExerciseGoalPrint.

\Points\{\langle value\rangle\}
Print \langle value\rangle followed by an appropriate translation of the words “point” or “points”, respectively. Defined in terms of \ExerciseGoalValuePrint.

---

3. See section 14 on page 48 for details on the definition and usage of language dependent words.
7. Exercise Properties

\printbonus{⟨type⟩}
Print the sum of bonus points for all exercises of type ⟨type⟩ followed by an appropriate translation of the words “point” or “points”, respectively. Defined in terms of \TotalExerciseTypeGoal.

\printtotalbonus
Print the sum of bonus points for all exercises followed by an appropriate translation of the words “point” or “points”, respectively. Defined in terms of \TotalExerciseGoal.

\addbonus*{⟨value⟩}
Adds ⟨value⟩ to the bonus points of the current exercise type. (To be used within exercises.) Prints the value followed by an appropriate translation of the words “point” or “points”, respectively. The starred version prints nothing. Defined in terms of \AddtoExerciseGoal and \AddtoExerciseGoalPrint.

The two existing goals are defined with

\begin{verbatim}
1 \DeclareExerciseGoal{points}
2 \DeclareExerciseGoal{bonus-points}
\end{verbatim}

When goal values are printed the decimal number is fed to a function which can be changed using the following option:

\begin{verbatim}
goal-print = {⟨code⟩}
\end{verbatim}

Default: #1

How to format goal values. Use #1 to refer to the actual number.

At last some examples for a custom command: let’s say you want a command which prints the complete sum for all exercises of all exercise types of both points and bonus-points added up:

\begin{verbatim}
\NewDocumentCommand\printsumofpointsandbonus{}{%
\TotalExerciseGoals{points+bonus-points}
\,\XSIMtranslate{point}
\,\XSIMtranslate{points}%
}
\end{verbatim}

Here is how you could mimick the command \totalpoints from exsheets:

\begin{verbatim}
\NewDocumentCommand\pointsandbonus{}{%
\TotalExerciseGoal{points}{}{}%
\IfExerciseGoalsSumF{bonus-points}{=0}
\,+(\,\TotalExerciseGoal{bonus-points}{}{})%
\,\XSIMtranslate{points}%
}
\end{verbatim}
7. Exercise Properties

7.4. A Special Kind of Property: Exercise Tags

Exercise tags are a generic concept in `xsim` for exercise properties like `tags` or `topics`. Those are properties which can (only) get a csv list of strings as value. Those strings can be used to selectively use exercises. See section 8 on the following page for details on usage of exercises and the difference to printing an exercise and how to use exercise tags for selection.

\DeclareExerciseTagging{⟨tag⟩}
This defines an exercise tagging group named ⟨tag⟩. It also defines a property named ⟨tag⟩. In addition two options are defined: an option named ⟨tag⟩ which can be used for selection and an boolean option ⟨tag⟩/ignore-untagged.

\ProvideExerciseTagging{⟨tag⟩}
The same as \DeclareExerciseTagging but does nothing when ⟨tag⟩ already exists.

The two existing tagging groups have been defined and preset with the following code:

\begin{verbatim}
\DeclareExerciseTagging{tags}
\DeclareExerciseTagging{topics}
\xsimsetup{tags/ignore-untagged=false}
\end{verbatim}

For each new tag new options are defined and available. They are described for the two predefined tags:

\begin{itemize}
\item \textbf{tags} = \{⟨csv list of tags⟩\}
Choose the set of tags whose associated exercises should be printed.
\item \textbf{topics} = \{⟨csv list of topics⟩\}
Choose the set of topics whose associated exercises should be printed.
\item \textbf{tags/ignore-untagged} = \{true|false\} Default: false
If set to true exercises with no tags will be printed even if tags have been chosen with the option \textit{tags}.
\item \textbf{topics/ignore-untagged} = \{true|false\} Default: true
If set to true exercises with no topics will be printed even if topics have been chosen with the option \textit{topics}.
\item \textbf{tags/use-unmatched} = \{true|false\} Default: false
Introduced in version 0.21 (Feb 12, 2022)
If set to true exercises with no matching tags will be used (i.e., for example the counter gets stepped).
\item \textbf{topics/use-untagged} = \{true|false\} Default: true
Introduced in version 0.21 (Feb 12, 2022)
If set to true exercises with no matching topics will be used (i.e., for example the counter gets stepped).
\end{itemize}
8. Using and Printing an Exercise

It may happen that you choose certain tags for printing and want one or two exercises to be printed or used even if they don’t match the tagging criteria. For this reason two additional properties exist which can be set to an exercise:

\[ \text{print!} = \text{true}|\text{false} \]
If set to true the exercise will be printed (and thus used) regardless of other conditions.

\[ \text{use!} = \text{true}|\text{false} \]
If set to true the exercise will be used regardless of other conditions.

8. Using and Printing an Exercise

8.1. What the Environments do

When an exercise is started with \begin{exercise} (or other environments defined through \DeclareExerciseType) then three things happen depending on the setup:

1. where the environment is placed in the document the base setup happens: this sets the id of the exercise and sets the properties chosen in the optional argument.

2. in the default setting also the usage setup happens: this increments the counter and sets the counter related properties like counter. Also the properties related to section numbers and the page are set.

3. in the default setting also the print setup happens: this mostly does nothing. It initiates recording goals that are set inside of the exercise body.

Depending on wether the option exercise/use has been set to false or wether the property use has been set to false the usage setup can be delayed. The usage the happens when the exercise is inserted the first time elsewhere in the document, for example as part of a collection, see section 9 on page 22.

Depending on wether the option exercise/print has been set to false or if the property print has been set to false the print setup can be suppressed. It happens each time an exercise is inserted.

\begin{exercise}[print=false,ID=invisible]
This exercise will not be printed but the exercise counter will be incremented nonetheless. Its solution will be printed in the list of solutions.
\end{exercise}
\begin{solution}
The solution of the exercise that has not been printed.
\end{solution}

The schematic structure of an exercise is shown in figure 1 on the following page.
8. Using and Printing an Exercise

pre hook

begin template code

begin hook

environment body

end hook

end template code

post hook

Figure 1: Schematic structure of an exercise or solution.

8.2. Environment Options & Hooks

For each exercise type there are the following options for both environments, the environments’ names are the module names for the options (here using the “exercise” type):

**exercise/print** = true|false  
Default: true  
Determines if exercises of type “exercise” are printed.

**exercise/use** = true|false  
Default: true  
Determines if exercises of type “exercise” are used.

**exercise/collect** = true|false  
Default: false  
Sets both print and use to false when collect = {true} is set and vice versa.

**exercise/within** = {⟨counter⟩}  
(Initially empty)  
Adds the exercise counter to the reset list of the counter ⟨counter⟩ using \counterwithin. Beware that if the counter is a shared counter this will affect all objects using this counter!

**exercise/the-counter** = {⟨code⟩}  
An interface for redefining the counter representation command \the(⟨counter⟩).

**exercise/template** = {⟨template⟩}  
An interface for \SetExerciseParameter{exercise}{exercise-template}{⟨template⟩}.

**solution/template** = {⟨template⟩}  
An interface for \SetExerciseParameter{exercise}{solution-template}{⟨template⟩}.

**exercise/name** = {⟨name⟩}  
An interface for \SetExerciseParameter{exercise}{exercise-name}{⟨name⟩}.

20
8. Using and Printing an Exercise

solution/name = \{\langle name\rangle\}

An interface for \SetExerciseParameter{exercise}{solution-name}{\langle name\rangle}.

exercise/heading = \{\langle heading command\rangle\}

An interface for \SetExerciseParameter{exercise}{exercise-heading}{\langle heading command\rangle}.

solution/heading = \{\langle heading command\rangle\}

An interface for \SetExerciseParameter{exercise}{solution-heading}{\langle heading command\rangle}.

exercise/pre-hook = \{\langle code\rangle\}

The code for the pre exercise hook for exercises of the type “exercise”. (initially empty)

exercise/begin-hook = \{\langle code\rangle\}

The code for the begin exercise hook for exercises of the type “exercise”. (initially empty)

exercise/end-hook = \{\langle code\rangle\}

The code for the end exercise hook for exercises of the type “exercise”. (initially empty)

exercise/post-hook = \{\langle code\rangle\}

The code for the post exercise hook for exercises of the type “exercise”. (initially empty)

solution/print = true|false

Determines if solutions of type “exercise” are printed. Default: false

solution/pre-hook = \{\langle code\rangle\}

The code for the pre solution hook for solutions of the type “exercise”. (initially empty)

solution/begin-hook = \{\langle code\rangle\}

The code for the begin solution hook for solutions of the type “exercise”. (initially empty)

solution/end-hook = \{\langle code\rangle\}

The code for the end solution hook for solutions of the type “exercise”. (initially empty)

solution/post-hook = \{\langle code\rangle\}

The code for the post solution hook for solutions of the type “exercise”. (initially empty)

8.3. (Re-) Inserting a Certain Exercise

If you know type and id of an exercise you can (re-)insert every existing exercise, i.e., every exercise whose external file exists.

\printexercise{\langle type\rangle}{\langle csv of ids\rangle}

Inserts the exercise or exercises of type \langle type\rangle with the \texttt{id}s or \texttt{IDs} given in \langle csv of ids\rangle.

\xprintexercise{\langle type\rangle}{\langle csv of ids\rangle}

The same as \printexercise but expands \langle type\rangle and the items of \langle csv of ids\rangle before it uses them.
9. Collecting Exercises

This exercise will not be printed but the exercise counter will be incremented nonetheless. Its solution will be printed in the list of solutions.

9. Collecting Exercises

The whole collection mechanism has been changed completely with version 0.20. Please check this section carefully and update and adapt your collections.

9.1. Background

\textsc{xsim} knows the concept of \textit{exercise collections}. A collection of exercises can be used in two cases, basically:

1. When you want to print a certain group of exercises several times. In this case you add the exercise to a collection while also using it in the running text. Somewhere later (or earlier actually) in the document you can the reprint the collection.\footnote{There is another way to do that, but...}

2. If you only want to print exercises which have certain tags. Maybe you have some kind of exercise library in an external which you include into your document and want to print only those exercises which fit certain conditions.

Each collection must have a unique name with which you can refer to the corresponding collection. A collection is realized by declaring the collection in the preamble and by activating the collection.

9.2. Usage

9.2.1. Declaring collections

\textit{A collection must be declared in the preamble}. Using a pair of commands explained below exercises between those commands are added to the corresponding collection but not printed. After a collection is completed the collection can be printed as often as needed.

\begin{verbatim}
\DeclareExerciseCollection[\{tag values\}]{\{collection name\}}
\end{verbatim}

Define a new collection \texttt{(collection name)} in the document preamble. An example could look like this: \texttt{\DeclareExerciseCollection[topics=algebra,level=1]{easy-algebra}}. Choosing tags automatically activates the collection which means that all exercises which fit the tags are automatically added to the new collection.
9. Collecting Exercises

You have a choice whether any or all of the tags should match. So \{topics=algebra, level=1\} should probably match all conditions but the choice is up to you. For each defined collection an option exists:

\[\text{collection/(name)/match-all = true|false}\]

If set to true all conditions must be met.

\[\text{collection/(name)/match-any = true|false}\]

The inverse option to match-all. If set to true any of the conditions must be met.

9.2.2. Activating collections

If a collection is not activated per default in can be activated (or deactivated in the opposite case) with these commands:

\[\text{\activatecollection{⟨collection name⟩}}\]

Activates the collection ⟨collection name⟩ which now collects all exercises until the collection is deactivated. ⟨collection name⟩ can be a csv list of collection names.

\[\text{\deactivatecollection{⟨collection name⟩}}\]

Deactivates the collection ⟨collection name⟩. ⟨collection name⟩ can be a csv list of collection names.

These commands do not influence exercises in other ways other than adding them to collections.

\[\text{\collectexercises{⟨collection name⟩}}\]

Activates the collectiοcollectexercisesn ⟨collection name⟩ which now collects all exercises until the collection is deactivated, starts a group with \begingroup and sets the option collect, see below.

\[\text{\collectexercisesstop{⟨collection name⟩}}\]

Deactivates the collection ⟨collection name⟩ and ends a group with \endgroup.

\[\text{collect = true|false}\]

A shortcut option which sets the options ⟨type⟩/collect for each exercise type.

9.2.3. Printing collections

\[\text{\printcollection{⟨options⟩}{⟨collection name⟩}}\]

Prints the collection ⟨collection name⟩, i.e., all exercises collected earlier. This command cannot be used before the corresponding collection has been closed correctly.

Valid options are the following:

\[\text{print-collection/headings = true|false}\]

If true a heading for each exercise type is inserted.
9. Collecting Exercises

\texttt{print-collection/headings-template} = \texttt{\{\{template\}\}} \quad \text{Default: collection}

The heading template used when \texttt{headings} = \texttt{true}.

\texttt{print-collection/print} = \texttt{exercises|solutions|both} \quad \text{Default: exercises}

Determines whether \texttt{\textbackslash printcollection} prints the exercises or the solutions of the collection. When you choose both exercises and solutions are printed alternately.

Those options can also be set via \texttt{\textbackslash xsimsetup} using the module \texttt{print-collection}.

Please be aware that exercises are not used or printed while they are collected. Nonetheless the property \texttt{use} is set to \texttt{true} (so that solutions can be printed even if the exercises are not) and the property \texttt{print} is set to \texttt{false}. Also their counters are \texttt{not stepped} during the process. This only happens when they are printed the first time, \textit{cf.} the \texttt{used} property. At that time also the properties \texttt{page}, \texttt{section} and \texttt{chapter} are set and the property \texttt{print} is set to \texttt{true}.

The usage should be clear:

\begin{verbatim}
\collectexercises{foo}
\begin{exercise}
This exercise is added to the collection `foo'.
\end{exercise}
\begin{exercise}
This exercise is also added to the collection `foo'.
\end{exercise}
\begin{exercise}
So is this.
\end{exercise}
\begin{exercise}
As well as this one.
\end{exercise}
\collectexercisestop{foo}
\end{verbatim}

Once the collection is closed it can be printed:

\begin{verbatim}
\printcollection{foo}
\end{verbatim}

\textbf{Exercise}

This exercise is added to the collection `foo'.
9. Collecting Exercises

Exercise
This exercise is also added to the collection ‘foo’.

Exercise
So is this.

Exercise
As well as this one.

You can open several collections at the same time:

\begin{verbatim}
\collectexercises{foo}
\ ...
\activatecollection{bar}
\ ...
\deactivatecollection{foo}
\ ...
\collectexercisestop{bar}
\end{verbatim}

Exercises will be added to each open collection.
There is one generic collection called "all exercises". As the name already suggests it will hold all exercises. So if you say

\begin{verbatim}
\printcollection{all exercises}
\end{verbatim}

all exercises will be printed.

If you use $\textcolor{red}{\texttt{\textbackslash label}}$s inside of exercises and you print exercises more than once in your document (by reusing a collection for example) you will get

\begin{verbatim}
LaTeX Warning: There were multiply-defined labels.
\end{verbatim}

Equally if you have environments like $\textcolor{red}{\texttt{\textbackslash begin\{equation\}}}$ which step a counter inside an exercise or solution the counter will be stepped each time the exercise is used.

At last now an example using external files, collections and tags:
10. Printing Random Exercises From a Collection

\texttt{xsim} provides the possibility of selecting random exercises from a collection (cf. section 9 on page 22).

\begin{itemize}
\item Please be aware that this feature is \textit{not} available in \texttt{X\LaTeX}!
\end{itemize}

\texttt{\textbackslash printrandomexercises}\{\langle\textit{options}\rangle\}\{\langle\textit{number}\rangle\}

This command prints \textit{(number)} random exercises from the collection chosen with option \texttt{collection}, see below. When this command is used it generates a random list of integers which is written to the aux file. On the subsequent compilations the according exercises are printed. \textit{If you want to regenerate the random list you have to delete the aux file before compiling.}

Valid options for this command are:
11. Printing Solutions

\begin{verbatim}
random/sort = true|false  
  Default: true
  Determines whether the random chosen exercises should be sorted according to their order of
  definition in the collection or not.

random/collection = \{(collection)\}  
  Default: all exercises
  The collection from which the exercises are to be chosen from.

random/exclude = \{(csv list of ids)\}  
  A list of ids or IDs of exercises not to be chosen.

random/print = exercises|solutions|both  
  Default: exercises
  Determines whether \printrandomexercises prints the exercises or the solutions. When you
  choose both exercises and solutions are printed alternately.
\end{verbatim}

\printrandomexercises[collection=foo]{2}

Exercise

This exercise is also added to the collection ‘foo’.

Exercise

So is this.

The example above of course doesn’t make much sense but if you have a collection which
collects exercises from an external file and the exercises haven’t been printed in the document
before then you will get a list of subsequently numbered exercises.

11. Printing Solutions

There are different commands for printing the solutions to exercises:

\begin{verbatim}
\printsolutionstype*[\{options\}]{\{exercise type\}}
  Prints the solutions of all used exercises of type \{exercise type\}. The starred version only
  prints the solutions of all printed exercises of type \{exercise type\}.

\printsolutions*[\{options\}]
  Prints the solutions of all used exercises of all types ordered by type. The starred version only
  prints the solutions of all printed exercises of all types.

\printallsolutions*[\{options\}]
  Prints the solutions of all used exercises of all types ordered by appearance in the document.
  The starred version only prints the solutions of all printed exercises of all types.

\printsolution*[\{options\}]{\{type\}{\{id\}]}
  Prints the solution of the exercise of type \{type\} with the id \{id\}.
\end{verbatim}
11. Printing Solutions

\printsolution{(type)}{(id)}

The same as \printsolution but expands (type) and (id) before it uses them.

\printsolutionstype{exercise}

Solutions to the Exercises

Solution 1
A first example for a solution.

Solution 5
The solution of the exercise that has not been printed.

Solution 6
Try to fill in these blanks. All of them are created by using the \blank command.

The options can be divided into two groups. The ones in the first group modify the layout.

**print-solutions/headings = true|false**
If true a heading for each exercise type is inserted.

**print-solutions/headings-template = \{template\}**
The heading template used when headings = {true}.

The ones in the second group set conditions selecting which solutions are printed. If you combine those conditions a solution is printed if it meets either of the conditions.

**print-solutions/section = true|false|integer**
If you set section = {true} only solutions of exercises of the current section are printed. If you set section = {4} only solutions of exercises in a section with number 4 are printed.

**print-solutions/chapter = true|false|integer**
If you set chapter = {true} only solutions of exercises of the current chapter are printed. If you set chapter = {4} only solutions of exercises in a chapter with number 4 are printed.

**print-solutions/collection = false|collection name**
If used only solutions of exercises belonging to collection \{collection name\} are printed.

The conditions can be combined. The following call will only print solutions from exercises in section 3 of chapter 2:

\printsolutions[chapter=2,section=3]
The selection per section or per chapter relies on the counter numbers of the sections or chapters, respectively. This means if section numbers are reset (e.g. by \chapter or \appendix) and you have exercises from different sections with the same section number the solutions of all those exercises will be printed. This means you only should use the section selection when section are the top document level headings (apart from parts) and you have no exercises in the appendix. Similar considerations are valid for the chapter selection.

All options can also be set via \xsimsetup using the module print-solutions.

\printsolution{exercise}{5}

Solution 5
The solution of the exercise that has not been printed.

12. Grading Tables

When you create exercises it may not only be desirable to be able to add points and bonus-points to a question (see section 7.3 on page 15 about exercise goals) but also to be able to output a grading table. \texttt{xsim} has built-in means for this.

\texttt{\gradingtable[(options)]}
Print a grading table.

Valid options for this command are

- \texttt{template = \{\langle template\rangle\}}
  Choose the template used for the grading table.
  Default: \texttt{default}

- \texttt{type = \{\langle exercise type\rangle\}}
  Choose the exercise type for which the table is printed.
  Initially empty

Both option defaults can be changed with \xsimsetup setting the options using \texttt{grading-table}:
12. Grading Tables

An example:

```latex
\texttt{\xsimsetup{
  grading-table/template = default*
}}
```

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Points reached</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td><strong>4</strong></td>
</tr>
</tbody>
</table>

Or using the "default*" template:

```latex
\texttt{\xsimgradingtable[type=exercise,template=default*,type=exercise]}
```

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Points reached</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 7 1 2 3 4 5 total</td>
<td>0 0 0 4 0 0 0 4</td>
</tr>
</tbody>
</table>

Available templates and how to define new ones are explained in sections 13.4.3 on page 40 and 13.6 on page 41. \texttt{XSIM} per default provides two templates "default" and "default*", the first one has a vertical layout, the second a horizontal layout. Both templates can be used per type like in the examples above or for all types at once by leaving the specification \texttt{type} away:
13. Styling the Exercises – Templates

13.1. Background

Whenever \texttt{xsim} outputs something to be typeset it uses so-called templates for the task. \texttt{xsim} knows of three different kinds of templates:

- environment templates (see section 13.4.1 on page 39),
- heading templates (see section 13.4.2 on page 40) and
- grading table templates (see section 13.4.3 on page 40)

The most important one for the styling of the exercises are the environment templates. Those templates give you complete control over the look and arrangement of an exercise. To be able to do this \texttt{xsim} provides a large number of commands which can be used only inside template definitions.\footnote{The last sentence is wrong: those commands can be used anywhere but most of them only give useful results inside of templates.} Those commands are explained in the next section. Their usage will hopefully become clear in the examples in section 13.6 on page 41. Having full control over the layout comes at a price: you need to be able to program yourself in order to achieve certain layouts.\footnote{I plan to incorporate the most common layouts – and maybe some fancy ones, too – in the examples section 13.6 on page 41 but at the time of writing this is still up in the air.}

### 13.2. Templates Provided by the Package

\texttt{xsim} comes with a few predefined layouts:

- \texttt{default} The template activated per default and the only one available without further action.
runin  A layout rather similar to the one by package exsheets, see section 13.6.3 on page 43. Available through the style file \texttt{layouts} (see section 13.5 on page 40 for more information on style files).

margin  A layout rather similar to the one by package exsheets, see section 13.6.4. Available through the style file \texttt{layouts}.

minimal  A minimalistic layout, see section 13.6.5. As the others inspired by an exsheets layout. Available through the style file \texttt{layouts}.

inline  A minimalistic layout, the same as \texttt{minimal} but doesn’t add \texttt{par} at the beginning and end. Available through the style file \texttt{layouts}.

centered  A layout with a centered heading. Available through the style file \texttt{layouts}.

---

**Layout “default”**

Exercise  The Subtitle


---

**Layout “runin”**

Exercise  The Subtitle  


---

**Layout “margin”**

Exercise  

13. Styling the Exercises – Templates

## Layout “inline”


## Layout “minimal” (Like “inline” but as own paragraph.)


## Layout “centered”

### Exercise  The Subtitle


### 13.3. Commands for Usage in Template Definitions

#### 13.3.1. Goals

\IfExerciseGoalTF{⟨goal⟩}{⟨relation and value⟩}{⟨true⟩}{⟨false⟩}

Checks the sum of goal ⟨goal⟩ against ⟨relation and value⟩.

\IfExerciseGoalSingularTF{⟨goal⟩}{⟨true⟩}{⟨false⟩}

Checks if the value of the goal ⟨goal⟩ of the current exercise equals 1. This is the same as \IfExerciseGoalTF{⟨goal⟩}{=1}{⟨true⟩}{⟨false⟩}.

\IfExerciseTypeGoalsSumTF{⟨type⟩}{⟨list of goals⟩}{⟨relation and value⟩}{⟨true⟩}{⟨false⟩}

Checks the sum of all goals in ⟨list of goals⟩ for the exercises of type ⟨type⟩ against ⟨relation and value⟩.

\IfExerciseGoalsSumTF{⟨type⟩}{⟨list of goals⟩}{⟨relation and value⟩}{⟨true⟩}{⟨false⟩}

Checks the sum of all goals in ⟨list of goals⟩ for all exercises of all types against ⟨relation and value⟩.
13. Styling the Exercises – Templates

\TotalExerciseTypeGoal{⟨goal⟩}{⟨type⟩}{⟨singular⟩}{⟨plural⟩}
Print the sum of goal ⟨goal⟩ for the exercises of type ⟨type⟩ and append ⟨singular⟩ or ⟨plural⟩ depending on whether the sum equals 1 or not.

\TotalExerciseGoal{⟨goal⟩}{⟨singular⟩}{⟨plural⟩}
Print the sum of goal ⟨goal⟩ for all exercises of all types and append ⟨singular⟩ or ⟨plural⟩ depending on whether the sum equals 1 or not.

13.3.2. Properties

∗ \IfExercisePropertyExistTF{⟨property⟩}{⟨true⟩}{⟨false⟩}
Tests whether an exercise property with the name ⟨property⟩ is defined.

∗ \IfExercisePropertySetTF{⟨property⟩}{⟨true⟩}{⟨false⟩}
Tests whether the exercise property ⟨property⟩ has been set for the current exercise.

∗ \GetExerciseProperty{⟨property⟩}
Retrieves the value of the property ⟨property⟩ for the current exercise.

\GetExercisePropertyTF{⟨property⟩}{⟨true⟩}{⟨false⟩}
Tests whether the exercise property ⟨property⟩ has been set for the current exercise. Inside the ⟨true⟩ branch you can refer to the retrieved value either with \#1 or with \PropertyValue. This command expands its contents inside a group.

\GetExerciseBody{exercise|solution}
Introduced in version 0.10 (Sep 19, 2017)
Retrieves the environment body of either the exercise or the corresponding solution of the current exercise.

∗ \GetExerciseIdForProperty{⟨property⟩}{⟨value⟩}
Retrieves the property id of the exercise where the property ⟨property⟩ has the value ⟨value⟩. This only works for unique properties!

\GetExerciseTypeForProperty{⟨property⟩}{⟨value⟩}
Retrieves the property type of the exercise where the property ⟨property⟩ has the value ⟨value⟩. This only works for unique properties!

\SetExerciseProperty{⟨property⟩}{⟨value⟩}
Changed in version 0.9 (Jun 20, 2017)
Set the property ⟨property⟩ of the current exercise to ⟨value⟩.

\SetExpandedExerciseProperty{⟨property⟩}{⟨value⟩}
Introduced in version 0.9 (Jun 20, 2017)
Expand ⟨value⟩ \edef-like and set the property ⟨property⟩ of the current exercise to the result of the expansion.

\ExerciseSetProperty{⟨type⟩}{⟨id⟩}{⟨property⟩}{⟨value⟩}
Introduced in version 0.9 (Jun 20, 2017)
Set the property ⟨property⟩ of the exercise of type ⟨type⟩ and id ⟨id⟩ to ⟨value⟩.
13. Styling the Exercises – Templates

\ExerciseSetExpandedProperty{{type}}{\{id\}}{\{property\}}{\{value\}}
Expand \{value\} \edef-like and set the property \{property\} of the exercise of type \{type\} and id \{id\} to the result of the expansion.

\* \IfExerciseBooleanPropertyTF{\{property\}}{\{true\}}{\{false\}}
Checks whether the boolean property \{property\} has value \{true\} or \{false\} and leaves the corresponding argument in the input stream. Gives an error if \{property\} is not a boolean property.

\* \GetExerciseAliasProperty{\{property\}}
Retrieves the value of the property of which \{property\} is an alias of for the current exercise.

\SaveExerciseProperty{\{property\}}{\{macro\}}
Saves the value of the property \{property\} for the current exercise in macro \{macro\}.

\GlobalSaveExerciseProperty
Globally saves the value of the property \{property\} for the current exercise in macro \{macro\}.

\ExercisePropertyIfSetTF{{type}}{\{id\}}{\{property\}}{\{true\}}{\{false\}}
Test if the property \{property\} has been set for the exercise of type \{type\} with id \{id\}.

\* \ExercisePropertyGet{{type}}{\{id\}}{\{property\}}
Retrieves the value of the property \{property\} for the exercise of type \{type\} with id \{id\}.

\* \ExercisePropertyGetAlias{{type}}{\{id\}}{\{property\}}
Retrieves the value of the property of which \{property\} is an alias of for the exercise of type \{type\} with id \{id\}.

\ExercisePropertySave{{type}}{\{id\}}{\{property\}}{\{macro\}}
Saves the value of the property \{property\} for the exercise of type \{type\} with id \{id\} in macro \{macro\}.

\ExercisePropertyGlobalSave{{type}}{\{id\}}{\{property\}}{\{macro\}}
Globally saves the value of the property \{property\} for the exercise of type \{type\} with id \{id\} in macro \{macro\}.

13.3.3. Parameters

\* \GetExerciseParameter{\{parameter\}}
Retrieves the value of the parameter \{parameter\} for the current exercise type.

\GetExerciseParameterTF{\{parameter\}}{\{true\}}{\{false\}}
Retrieves the value of the parameter \{parameter\} for the current exercise type. Inside the \{true\} branch you can refer to the retrieved value either with \#1 or with \ParameterValue. This command expands its contents inside a group.
13. Styling the Exercises – Templates

\* \GetExerciseName
   Retrieves the value of the parameter exercise-name for the current exercise or of the parameter solution-name for the current solution.

\* \GetExerciseHeadingF{{false}}
   Retrieves the value of the parameter exercise-heading for the current exercise or of the parameter solution-heading for the current solution. Inserts ⟨false⟩ if the corresponding parameter has not been set.

\* \ExerciseParameterGet{{type}}{{parameter}}
   Retrieves the value of the parameter ⟨parameter⟩ for the exercise of type ⟨type⟩ with id ⟨id⟩.

\* \IfExerciseParameterSetTF{{parameter}}{{true}}{{false}}
   Introduced in version 0.9 (Jun 20, 2017)
   Test if the parameter ⟨parameter⟩ has been set for the current exercise type.

\* \ExerciseParameterIfSetTF{{type}}{{parameter}}{{true}}{{false}}
   Introduced in version 0.9 (Jun 20, 2017)
   Test if the parameter ⟨parameter⟩ has been set for the exercise type ⟨type⟩.

13.3.4. Tags

\ForEachExerciseTag{{type}}{{code}}
   Loops over all tags of tag type ⟨type⟩ for the current exercise applying ⟨code⟩ each time. Inside ⟨code⟩ you can refer to the corresponding tag with #1.

\ListExerciseTags{{type}}{{between}}
   Lists all tags of tag type ⟨type⟩ for the current exercise using ⟨between⟩ as a separator.

\UseExerciseTags{{type}}{{between two}}{{between}}{{between last two}}
   Lists all tags of tag type ⟨type⟩ for the current exercise using ⟨between⟩ as a separator and ⟨between last two⟩ as separator between the last two tags of the list. If the list only consists of two tags ⟨between two⟩ is used as separator.

\IfExerciseTagSetTF{{value}}{{true}}{{false}}
   Introduced in version 0.11 (Feb 12, 2018)
   In order to insert text (also outside of exercises) depending on the chosen tags this command lets you check if value ⟨value⟩ has been set for tags.

\IfExerciseTopicSetTF{{value}}{{true}}{{false}}
   Introduced in version 0.11 (Feb 12, 2018)
   In order to insert text (also outside of exercises) depending on the chosen tags this command lets you check if value ⟨value⟩ has been set for topics.

13.3.5. Further Commands for Usage in Template Definitions

\UseExerciseTemplate{{type}}{{name}}
   Retrieve template ⟨name⟩ of type ⟨type⟩. This can be useful if you want to define a template which just adds some code to an existing template (an automated \label, say).

\ExerciseType
   Can be used to refer to the current exercise type.

36
13. Styling the Exercises – Templates

* \ExerciseID
  Can be used to refer to the current exercise id.

* \ExerciseText
  Can be used inside solutions to retrieve the text of the corresponding solution. This is probably seldom useful as in most use cases the exercise property `solution` is the easier alternative.

* \ExerciseCollection
  Can be used in certain templates to refer to the collection that is currently inserted.

* \numberofusedexercises
  Holds the total number of used exercises. Useful in table template definitions.

* \ExerciseTableType{⟨code⟩}
  In table template definitions this macro either expands to the given exercise type or – if no type has been given – to `⟨code⟩`.

* \IfInsideSolutionTF{⟨true⟩}{⟨false⟩}
  Tests if the template is used inside a solution environment or not.

* \IfSolutionPrintTF{⟨true⟩}{⟨false⟩}
  Tests if the option `print` for the solutions of the current \ExerciseType is set to `true` or `false`.

* \IfExistSolutionTF{⟨true⟩}{⟨false⟩}
  Tests if a solution for the current exercise exists.

\ForEachPrintedExerciseByType{⟨code⟩}
  Loops over each `printed` exercise ordered by the exercise types and within each type by id. Inside `⟨code⟩` you can refer to several properties of the corresponding exercise:
  
  • #1: the type of the exercise
  • #2: the id of the exercise
  • #3: the `counter` property of the exercise
  • #4: the `subtitle` property of the exercise
  • #5: the `points` property of the exercise
  • #6: the `bonus-points` property of the exercise

\ForEachUsedExerciseByType{⟨code⟩}
  Loops over each `used` exercise ordered by the exercise types and within each type by id. Inside `⟨code⟩` you can refer to several properties of the corresponding exercise:
  
  • #1: the type of the exercise
  • #2: the id of the exercise
  • #3: the `counter` property of the exercise
13. Styling the Exercises – Templates

• #4: the `subtitle` property of the exercise
• #5: the `points` property of the exercise
• #6: the `bonus-points` property of the exercise

\ForEachUsedExerciseByOrder{⟨code⟩}
Loops over each *used* exercise ordered by the exercise types and within each type by the order they have been used in the document. Inside `<code>` you can refer to several properties of the corresponding exercise:

• #1: the type of the exercise
• #2: the id of the exercise
• #3: the `counter` property of the exercise
• #4: the `subtitle` property of the exercise
• #5: the `points` property of the exercise
• #6: the `bonus-points` property of the exercise

\ForEachPrintedExerciseByID{⟨code⟩}
Loops over each *printed* exercise order by the exercise id. Inside `<code>` you can refer to several properties of the corresponding exercise:

• #1: the type of the exercise
• #2: the id of the exercise
• #3: the `counter` property of the exercise
• #4: the `subtitle` property of the exercise
• #5: the `points` property of the exercise
• #6: the `bonus-points` property of the exercise

\ForEachUsedExerciseByID{⟨code⟩}
Loops over each *used* exercise order by the exercise id. Inside `<code>` you can refer to several properties of the corresponding exercise:

• #1: the type of the exercise
• #2: the id of the exercise
• #3: the `counter` property of the exercise
• #4: the `subtitle` property of the exercise
• #5: the `points` property of the exercise
• #6: the `bonus-points` property of the exercise

\XSIMprint{exercise|solution}{{⟨type⟩}}{{⟨id⟩}}
Inserts the either the exercise or the solution of type `<type>` with the id or ID `<id>`. 

38
13. Styling the Exercises – Templates

\texttt{\textbackslash XSIM\textbackslash print\{exercise|solution\}\{\langle type\rangle\}\{\langle id\rangle\}}

The same as \texttt{\textbackslash XSIM\textbackslash print} but expands \langle type\rangle and \langle id\rangle before it uses them.

* \texttt{\textbackslash XSIM\textbackslash translate\{\langle keyword\rangle\}}

Delivers the translation of \langle keyword\rangle according to the current document language (in the meaning of a babel [Bra22] or polyglossia [Cha21] language). Existing keywords and keyword translations (and how to add new ones) are explained in section 14 on page 48.

\texttt{\textbackslash XSIM\textbackslash expand\textbackslash code\{\langle code\rangle\}}

Expands \langle code\rangle like \texttt{\edef} does and leaves the result in the input stream.

* \texttt{\textbackslash XSIM\textbackslash if\textbackslash chapter\textbackslash TF\{\langle true\rangle\}\{\langle false\rangle\}}

Returns \langle true\rangle if both a macro \texttt{\chapter} and a counter chapter are defined and \langle false\rangle otherwise.

\texttt{\textbackslash XSIM\textbackslash mixed\textbackslash case\{\langle code\rangle\}}

Converts the full expansion of \langle code\rangle to mixed case:
\texttt{\textbackslash XSIM\textbackslash mixed\textbackslash case\{this is some text\}} This is some text
This command expands \langle code\rangle before converting it.

\texttt{\textbackslash XSIM\textbackslash put\textbackslash right\{\langle macro\rangle\}\{\langle code\rangle\}}

Extends the macro definition of \langle macro\rangle with \langle code\rangle putting it to the right. This is more or less a local version of the LaTeX kernel macro \texttt{\g@addto@macro}.

* \texttt{\textbackslash XSIM\textbackslash if\textbackslash eq\textbackslash TF\{\langle code 1\rangle\}\{\langle code 2\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}}

Checks if the full expansion of \langle code 1\rangle and \langle code 2\rangle is the same tokenlist.

* \texttt{\textbackslash XSIM\textbackslash if\textbackslash blank\textbackslash TF\{\langle code\rangle\}\{\langle true\rangle\}\{\langle false\rangle\}}

Checks if the full expansion of \langle code\rangle is blank (\textit{i.e.}, if it is empty or only consists of spaces).

\texttt{\textbackslash XSIM\textbackslash at\textbackslash begin\textbackslash document\{\langle code\rangle\}}

Adds \langle code\rangle to \texttt{\xsim}'s begin document hook. Should be used inside style files instead of \texttt{\textbackslash AtBeginDocument}.

\texttt{\textbackslash XSIM\textbackslash at\textbackslash end\textbackslash document\{\langle code\rangle\}}

Adds \langle code\rangle to \texttt{\xsim}'s end document hook. Should be used inside style files instead of \texttt{\textbackslash AtEndDocument}.

13.4. Declaring Templates

13.4.1. Environment Templates

\texttt{\textbackslash Declare\textbackslash Exercise\textbackslash Environment\textbackslash Template\{\langle name\rangle\}\{\langle begin\ code\rangle\}\{\langle end\ code\rangle\}}

Declare the environment template \langle name\rangle.

Environment templates are used by the exercise and solution environments. Those are the templates set with the parameters \texttt{exercise\textbackslash template} and \texttt{solution\textbackslash template}.

The predefined template is called “default”, see section 13.6.1 on page 41.
13. Styling the Exercises – Templates

13.4.2. Heading Templates

\DeclareExerciseHeadingTemplate{⟨name⟩}{⟨code⟩}
Declare the heading template ⟨name⟩.

Heading templates are used by \printsolutions, \printsolutionstype and \printcollection. Those are the templates set with the option headings-template of the modules print-solutions and print-collection.

The predefined templates are “default”, “collection”, “per-section” and “per-chapter” see section 13.6.6 on page 45.

13.4.3. Grading Table Templates

\DeclareExerciseTableTemplate{⟨name⟩}{⟨code⟩}
Declare the grading table template ⟨name⟩.

Table templates are used by \gradingtable. Those are the templates set with the option template of module grading-table.

The predefined templates are “default” and “default*”, see sections 13.6.7 on page 45 and 13.6.8 on page 46.

13.5. Create and Use xsim Style Files

xsim offers you the possibility to create own style files. Let’s say you want to have a style called math-exam. Then you need to save all necessary definitions in a file called:

xsim.style.math-exam.code.tex

The first command in the file should be \xsimstyle{math-exam}. This file can now be loaded into your document using \loadxsimstyle{math-exam} or by using \xsimsetup{load-style=math-exam}:

\documentclass[DIV=18,parskip=half]{scrartcl}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[clear-aux]{xsim}
\loadxsimstyle{math-exam}
\title{Math Exam \#3}
\date{2017-03-28}

In this style file stuff like template and property definitions should happen. This is more or less a convenient way to

• keep the preamble “clean” and
13. Styling the Exercises – Templates

- define re-usable styles without the need of copying the document preamble to another document.

A style file is like a package or class file, i.e., @ has category code 11 (letter).

The formal description of the commands:

\xsimstyle*{(style name)}

The first command in a XSIM style file called xsim.style.(style name).code.tex which defines the XSIM style ⟨style name⟩. The starred version activates expl3 syntax. 7

\loadxsimstyle{(csv list of style names)}

Load one or more styles into the document.

\loadstyle = {(csv list of style names)}

Another interface for \loadxsimstyle{(csv list of style names)}.

At the moment this mechanism offers no advantages over creating a custom package or simply \inputing a file. Future versions might provide additional features.

13.6. Examples

The repository of this package currently includes 42 example documents demonstrating how different aspects of this package work or how different kinds of problems can be solved or how different kinds of layouts can be achieved as well as how solve concrete problems that have come up in different L\LaTeX/ forums, see section F on page 61.

13.6.1. The default Exercise Template

Below the definition of the default exercise template provided by XSIM is shown:

![](image)

7. Those users who want this will know what it means. If you don’t know what it means you will not need it.
13. Styling the Exercises – Templates

13.6.2. A New Exercise Type Using \texttt{tcolorbox}

Let’s say we want exercises to be put in a \texttt{tcolorbox}. We want a bold title and, if given, an italic subtitle. Exercises should also have the points after the subtitle in parentheses if given. Let’s also say we want those to be an additional exercise type in addition to the ones \texttt{xsim} already provides. This is shown with the following code which is also how the problems in this manual have been defined:

\begin{verbatim}
\DeclareExerciseEnvironmentTemplate{tcolorbox}
{\tcolorbox[}
  colback = red!5!white ,
  colframe = red!75!black ,
  colbacktitle = yellow!50!red ,
  coltitle = red!25!black ,
  breakable ,
  drop shadow ,
  beforeafter skip = .5\baselineskip ,
  title =
  \textbf{\GetExerciseName~\GetExerciseProperty{counter}}%
  \GetExercisePropertyT{subtitle}{ \textit{\PropertyValue}}%
  \IfInsideSolutionF{%\ Gets the points after the title if given.
    \GetExercisePropertyT{points}{ % notice the space
      \printgoal{\PropertyValue}
      \IfExerciseGoalSingularTF{points}{\XSIMtranslate{point}}{\XSIMtranslate{points}}%
    }%
  }%}
\end{verbatim}
13. Styling the Exercises – Templates

\begin{exercise}[subtitle=My subtitle,points=5]
This is a problem using a subtitle and points.
\end{exercise}

\begin{answer}
This is the answer to problem~\GetExerciseProperty{counter}.
\end{answer}

\begin{problem}
My subtitle (5 points)
This is a problem using a subtitle and points.
\end{problem}

13.6.3. Mimicking exsheets’ \texttt{runin} Template

The following example shows how you could mimick exsheets’ \texttt{runin} template. The outcome isn’t exactly the same since exsheets doesn’t use \texttt{\marginpar} but the result should look very similar. A safer definition would use a real sectioning command for the title.

\usepackage{needspace}
\DeclareExerciseEnvironmentTemplate{runin}{%
  \par\vspace{\baselineskip}
  \Needspace*{2\baselineskip}
  \noindent
  \textbf{\XSIMmixedcase{\GetExerciseName}~\GetExerciseProperty{counter}}% <<< notice the space
  \IfInsideSolutionF{%
    \GetExercisePropertyT{subtitle}{\textit{#1}}% %
  }%
  \GetExercisePropertyT{points}{%}
  \marginpar{%}
13. Styling the Exercises – Templates

13.6.4. Mimicking exsheets’ margin Template

The following example shows how you could mimick exsheets’ margin template.

\DeclareExerciseEnvironmentTemplate{margin}
\trivlist
\item\llap{\smash{\begin{tabular}[t]{@{}r@{}}
\textbf{\XSIMmixedcase{\GetExerciseName} ~ \GetExerciseProperty{counter}}
\IfExercisePropertySetT{points}{\textbf{\ifExercisePoints singularTF{points}{\ifExercisePoints translate{point}{\ifExercisePoints translate{points}{\textbf{\ifExercisePoints translate{points}{\(\printgoal{\GetExerciseProperty{points}}\text{\ifExercisePoints translate{point-abbr}{\(\printgoal{#1}\text{\ifExercisePoints translate{point-abbr}{\(\XSIMtranslate{point-abbr}{\XSIMtranslate{points}}\)}}\end{tabular}}\relax\endtrivlist}}}}}}}}}}}}}}

13.6.5. A minimal Template

This shows the implementation of the minimal template:

\DeclareExerciseEnvironmentTemplate{minimal}
\par
13. Styling the Exercises – Templates

13.6.6. The Headings Templates

\textbf{xsim} defines four heading templates which only differ by which text they output:

\begin{verbatim}
\DeclareExerciseHeadingTemplate{default}{\section*{\XSIMtranslate{default-heading}}}
\DeclareExerciseHeadingTemplate{collection}{\section*{\XSIMtranslate{collection-heading}}}
\DeclareExerciseHeadingTemplate{per-section}{\section*{\XSIMtranslate{per-section-heading}}}
\DeclareExerciseHeadingTemplate{per-chapter}{\section*{\XSIMtranslate{per-chapter-heading}}}
\end{verbatim}

Section 14 on page 48 shows how the translations are defined.

13.6.7. The default Table Template

This template is the one used for grading tables per default. It has a vertical layout.

\begin{verbatim}
\DeclareExerciseTableTemplate{default}{%
  \XSIMputright\ExerciseTableCode{%
    \toprule
    \XSIMifblankF{\ExerciseType}{%\XSIMmixedcase{\GetExerciseParameter{exercise-name}}} &
    \XSIMmixedcase{\XSIMtranslate{points}} &
    \XSIMtranslate{reached} \\
    \midrule
    \ForEachUsedExerciseByType{%
      \XSIMifeqT{#1}{\ExerciseTableType{#1}}{%
        \XSIMmixedcase{\XSIMtranslate{points}} &
        \XSIMtranslate{reached} \ \\
      }
    }
  }
\end{verbatim}
13. Styling the Exercises – Templates

The part

\begin{tabular}{lcc}
\ExerciseTableCode{...} & \ExerciseTableCode{...} & \\
\midrule
\end{tabular}

repeatedly checks if an exercise type has been given for the table. This makes it possible to
design the table differently if it is for one exercise type only (the true case) or for all exercise
types (the false case). \ExerciseTableType{⟨code⟩} either expands to the given exercise
type or to ⟨code⟩.

13.6.8. The default* Table Template

The second of the predefined grading table templates. It has a horizontal layout.

If you have a lot of exercises the width of a table with this layout may exceed the text
width of the document!


\toprule
\XSimifblankF{\ExerciseType}
{\XSimmixedcase{\GetExerciseParameter{exercise-name}}} &
\% \\
\ForEachUsedExerciseByType{%
\XSimifeqT{#1}{\ExerciseTableType{#1}}{
\XSimifblankT{\ExerciseTableType{}}
{\
\XSimputright{\ExerciseTableCode}{
\XSimmixedcase{\ExerciseParameterGet{#1}{exercise-name} }
}\%
}
}\%
\XSimputright{\ExerciseTableCode{#3 \&}}
}%
\% \\
\XSimputright{\ExerciseTableCode{\XSimtranslate{total} \midrule \XSimmixedcase{\XSimtranslate{points}} \&}}
\% \\
\ForEachUsedExerciseByType{%
\XSimifeqT{#1}{\ExerciseTableType{#1}}{%
\XSimputright{\ExerciseTableCode{\XSimifblankTF{#5}{\printgoal{0}}{\printgoal{#5}} \&}}
}%
\% \\
\XSimputright{\ExerciseTableCode{\XSimifblankTF{\ExerciseType}{\TotalExerciseGoal{points}{}{}{}}\TotalExerciseTypeGoal{\ExerciseType}{points}{}{}}\midrule \XSimtranslate{reached} &}
\% \\
\ForEachUsedExerciseByType{%
\XSimifeqT{#1}{\ExerciseTableType{#1}}{\XSimputright{\ExerciseTableCode{}}}
}%
\% \\
\edef\numberofcolumns{\XSimifblankTF{\ExerciseType}{\numberofusedexercises}{\csname numberof \ExerciseType s\endcsname}}
\% \\
\edef\numberofcolumns\{0\}

47
14. Exercise Translations

\begin{tabular}{l*{\numberofcolumns}{c}c}
\ExerciseTableCode
\end {tabular}%

The part
\XSIMifblankTF{\ExerciseType}{ ... }{ ... }

repeatedly checks if an exercise type has been given for the table. This makes it possible to design the table differently if it is for one exercise type only (the true case) or for all exercise types (the false case). \ExerciseTableType{(code)} either expands to the given exercise type or to (code).

14. Exercise Translations

\DeclareExerciseTranslation{(language)}{(keyword)}{(translation)}
Declare the translation of (keyword) for language (language).

\DeclareExerciseTranslations{(keyword)}{(translations)}
Declare the translations of (keyword) for several languages at once. See an example of the usage below.

\XSIMtranslate{(keyword)}
Delivers the translation of (keyword) according to the current document language (in the meaning of a babel [Bra22] or polyglossia [Cha21] language).

\ForEachExerciseTranslation{(code)}
Loops over all translations of all keywords known to \texttt{xsim}. Inside (code) you can refer to the keyword with #1, to the language with #2, and to the translation with #3.

As an example how to use \texttt{\DeclareExerciseTranslations} here is how the translations for exercise have been defined:

\begin{verbatim}
\DeclareExerciseTranslations{exercise}{
  Fallback = exercise ,
  English = exercise ,
  French = exercice ,
  German = "Ubung
}
\end{verbatim}
Table 1 shows all existing keywords with all predefined translations.

<table>
<thead>
<tr>
<th>keyword</th>
<th>language</th>
<th>translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>exercise</td>
<td>Fallback</td>
<td>exercise</td>
</tr>
<tr>
<td>exercise</td>
<td>English</td>
<td>exercise</td>
</tr>
<tr>
<td>exercise</td>
<td>French</td>
<td>&quot;exercice&quot;</td>
</tr>
<tr>
<td>exercises</td>
<td>Fallback</td>
<td>exercises</td>
</tr>
<tr>
<td>exercises</td>
<td>English</td>
<td>exercises</td>
</tr>
<tr>
<td>exercises</td>
<td>French</td>
<td>&quot;exercices&quot;</td>
</tr>
<tr>
<td>exercises</td>
<td>German</td>
<td>&quot;Ubung&quot;</td>
</tr>
<tr>
<td>question</td>
<td>Fallback</td>
<td>question</td>
</tr>
<tr>
<td>question</td>
<td>English</td>
<td>question</td>
</tr>
<tr>
<td>question</td>
<td>French</td>
<td>question</td>
</tr>
<tr>
<td>question</td>
<td>German</td>
<td>Aufgabe</td>
</tr>
<tr>
<td>questions</td>
<td>Fallback</td>
<td>questions</td>
</tr>
<tr>
<td>questions</td>
<td>English</td>
<td>questions</td>
</tr>
<tr>
<td>questions</td>
<td>French</td>
<td>questions</td>
</tr>
<tr>
<td>questions</td>
<td>German</td>
<td>Aufgaben</td>
</tr>
<tr>
<td>solution</td>
<td>Fallback</td>
<td>solution</td>
</tr>
<tr>
<td>solution</td>
<td>English</td>
<td>solution</td>
</tr>
<tr>
<td>solution</td>
<td>French</td>
<td>solution</td>
</tr>
<tr>
<td>solution</td>
<td>German</td>
<td>&quot;Ubungen&quot;</td>
</tr>
<tr>
<td>solutions</td>
<td>Fallback</td>
<td>solutions</td>
</tr>
<tr>
<td>solutions</td>
<td>English</td>
<td>solutions</td>
</tr>
<tr>
<td>solutions</td>
<td>French</td>
<td>solutions</td>
</tr>
<tr>
<td>solutions</td>
<td>German</td>
<td>&quot;Osungen&quot;</td>
</tr>
<tr>
<td>point-abbr</td>
<td>Fallback</td>
<td>p.</td>
</tr>
<tr>
<td>point-abbr</td>
<td>English</td>
<td>p.</td>
</tr>
<tr>
<td>point-abbr</td>
<td>French</td>
<td>p.</td>
</tr>
<tr>
<td>point-abbr</td>
<td>German</td>
<td>P.</td>
</tr>
<tr>
<td>point</td>
<td>Fallback</td>
<td>point</td>
</tr>
<tr>
<td>point</td>
<td>English</td>
<td>point</td>
</tr>
<tr>
<td>point</td>
<td>French</td>
<td>point</td>
</tr>
<tr>
<td>point</td>
<td>German</td>
<td>Punkt</td>
</tr>
<tr>
<td>points</td>
<td>Fallback</td>
<td>points</td>
</tr>
<tr>
<td>points</td>
<td>English</td>
<td>points</td>
</tr>
<tr>
<td>points</td>
<td>French</td>
<td>points</td>
</tr>
<tr>
<td>points</td>
<td>German</td>
<td>Punkte</td>
</tr>
<tr>
<td>reached</td>
<td>Fallback</td>
<td>reached</td>
</tr>
</tbody>
</table>

continues
14. Exercise Translations

<table>
<thead>
<tr>
<th>keyword</th>
<th>language</th>
<th>translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>reached</td>
<td>English</td>
<td>reached</td>
</tr>
<tr>
<td>reached</td>
<td>French</td>
<td>obtenus</td>
</tr>
<tr>
<td>reached</td>
<td>German</td>
<td>erreicht</td>
</tr>
<tr>
<td>total</td>
<td>Fallback</td>
<td>total</td>
</tr>
<tr>
<td>total</td>
<td>English</td>
<td>total</td>
</tr>
<tr>
<td>total</td>
<td>French</td>
<td>total</td>
</tr>
<tr>
<td>total</td>
<td>German</td>
<td>insgesamt</td>
</tr>
<tr>
<td>default-heading</td>
<td>Fallback</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solutions-name}} to the \XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>default-heading</td>
<td>English</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solutions-name}} to the \XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>default-heading</td>
<td>French</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solutions-name} des \GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>default-heading</td>
<td>German</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solutions-name}} zu den \XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>collection-heading</td>
<td>Fallback</td>
<td>\XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>collection-heading</td>
<td>English</td>
<td>\XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>collection-heading</td>
<td>French</td>
<td>\XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
<tr>
<td>collection-heading</td>
<td>German</td>
<td>\XSIMmixedcase {\GetExerciseParameter {exercises-name}}</td>
</tr>
</tbody>
</table>
| per-section-heading | Fallback | \XSIMmixedcase {\GetExerciseParameter {solutions-name}} to the \XSIMmixedcase {\GetExerciseParameter {exercises-name}} of Section
| per-section-heading | English | \XSIMmixedcase {\GetExerciseParameter {solutions-name}} to the \XSIMmixedcase {\GetExerciseParameter {exercises-name}} of Section
| per-section-heading | French  | \XSIMmixedcase {\GetExerciseParameter {solutions-name} des \GetExerciseParameter {exercises-name} de la section} }
15. Cloze Tests and Blank Lines

Similar to exsheets \texttt{xsim} provides a command \texttt{\textbackslash blank}:

\texttt{\textbackslash blank[\langle options\rangle\{\langle text\ to\ be\ filled\ in\rangle\}]

Creates a blank in normal text or in an exercise but fills the text of its argument if inside a solution. If used at the begin of a paragraph \texttt{\textbackslash blank} will do two things: it will set the linespread according to an option explained below and will insert \texttt{\textbackslash par} after the lines. The starred version doesn’t do these things.

Those are the options for customization:

\begin{itemize}
\item \texttt{\textbackslash blank/blank-style = {\langle code\rangle}}
  \hspace{1cm} Default: \texttt{\underline{\#1}}
  
  Instructions for typesetting the blank cloze. Refer to the filled in space with \#1.

\item \texttt{\textbackslash blank/filled-style = {\langle code\rangle}}
  \hspace{1cm} Default: \texttt{\underline{\#1}}
  
  Instructions for typesetting the filled cloze. Refer to the filled in text with \#1

\item \texttt{\textbackslash style = {\langle code\rangle}}
  
  Shortcut for setting both \texttt{blank-style} and \texttt{filled-style} at once.

\item \texttt{\textbackslash blank/scale = {\langle decimal\ number\rangle}}
  \hspace{1cm} Default: 1
  
  Scales the blank to \langle decimal\ number\rangle times its natural width.
\end{itemize}
15. Cloze Tests and Blank Lines

\texttt{\textbf{blank/width = \{\langle dim \rangle\}}}

Sets the blank to a width of \langle dim \rangle. This takes precedence over \texttt{scale}.

\texttt{\textbf{blank/linespread = \{\langle decimal number \rangle\}}}

Default: 1

Set the linespread for the blank lines. This only has an effect if \texttt{\blank} is used at the begin of a paragraph.

\texttt{\textbf{blank/line-increment = \{\langle dim \rangle\}}}

Default: 0.001\texttt{\linewidth}

The blank line is built in multiples of this value. If the value is too large you may end up with uneven lines. If the value is too small you may end up with a non-ending compilation. Experiment with values to find the suiting one for your use case.

\texttt{\textbf{blank/line-minimum-length = \{\langle dim \rangle\}}}

Default: 2em

The minimal length a line must have before it is built step by step.

\texttt{\textbf{blank/fill = true|false}}

Default: false

If set to true, this will show the correct answers in the blanks within an exercise as well.

---

This is a \texttt{\blank\{blank\}} outside in normal text.

\begin{exercise}
\texttt{Try to fill in \blank[width=4cm]{these} blanks. All of them \texttt{\blank} are created by using the \texttt{\cs{blank} \texttt{\blank}} command.}
\end{exercise}

\begin{solution}[print]
\texttt{Try to fill in \blank[width=4cm]{these} blanks. All of them \texttt{\blank} are created by using the \texttt{\cs{blank} \texttt{\blank}} command.}
\end{solution}

This is a _____ outside in normal text.

**Exercise 6**

Try to fill in ______________ blanks. All of them ______ by using the \texttt{\blank ______}.

**Solution 6**

Try to fill in these blanks. All of them are created by using the \texttt{\blank command}.

A number of empty lines are easily created by setting the \texttt{width} option:

\begin{exe}
\texttt{Write up the pros and cons of \texttt{xsim} over \texttt{pkg(exsheets)}:}
\end{exe}

\begin{exe}
\texttt{\blank[width=4.8\texttt{\linewidth},linespread=1.5]{}}
\end{exe}
A. Future Plans

Write up the pros and cons of \texttt{xsim} over \texttt{exsheets}:

\begin{itemize}
  \item a document class \texttt{xsim-exam} for creating exams; this class should itself feature the possibility of creating different versions of an exam, maybe already provide multiple choice questions and so on; one could also think about automatic creation of running headers and footers, \textit{i.e.}, means for changing the layout of the exam; following the spirit of \texttt{xsim} this should probably be done using templates as well.
\end{itemize}

I am very open to suggestions regarding features, both in general and specifically regarding the document class.

B. FAQ & How to...

This section serves as a kind of gallery showing solutions to common problems. I expect this section to grow over the years. Some examples especially regarding other layouts are also shown in example files added to this package.

B.1. ...Know if \texttt{xsim} Needs Another Compilation?

If \texttt{xsim} wants you to recompile your document it issues a warning and writes the following to the logfile:

```
1 Package xsim Warning: Exercise properties may have changed. Rerun to get
them
2 (xsim) synchronized.
```

So just check the logfile regularly (which you should be doing anyway) and keep your eyes open.
B. FAQ & How to...

B.2. ...Resolve Repeatedly Wrong Exercise Properties or Wrong Exercise Lists?

XSIM writes a lot of stuff to an auxiliary file called \langle file name⟩.xsim (or the common \langle file name⟩.aux if you use option use-aux) for re-using information on subsequent compilations. If you add exercises, change properties etc. it might happen that wrong information is staying in the auxiliary file and is wrongly used by XSIM. In such cases deleting the auxiliary file and doing a few fresh compilations may resolve your problems.

Sometimes the existence of exercise or solution files from earlier compilations may lead to wrong lists of exercises or solutions. In such cases it can be useful to delete all those files and doing a fresh compilation. It may be helpful to use a subfolder for those external files which will make deleting them a little bit easier. (Don’t forget to both create the subfolder and set path accordingly then.)

Using the clear-aux option might help to reduce erroneous exercises.

B.3. ...Resolve Strange Errors After Updating?

XSIM writes a lot of stuff to the auxiliary file. An update may well change how this is done so deleting the auxiliary file and doing a few fresh compilations may resolve your problems.

B.4. ! TeX capacity exceeded, sorry [text input levels=15]. Why?

Did you try to use an exercise or solution in a macro of some sort? This generally will fail if you also use the option use-files. But there should never be the need to hide the environments inside of a macro, anyway.

B.5. Runaway argument? !File ended while scanning use of ^M. Why?

Did you try to use an exercise or solution in a macro of some sort? This generally will fail if you also use the option use-files. But there should never be the need to hide the environments inside of a macro, anyway.

B.6. ...Put a Star (or Another Symbol) in Headings of Exercises That Are Special?

The code below shows one possible modification of an exercise template which allows to easily create bonus exercises:

\begin{verbatim}
\% preamble:
\usepackage{amsymb}
\% declare boolean property:
\DeclareExerciseProperty*(bonus)
\DeclareExerciseEnvironmentTemplate{bonus}
\end{verbatim}

9. The reasons are similar to the ones given here: https://tex.stackexchange.com/a/295422/.
**B. FAQ & How to...**

The usage is now as follows:

```latex
\xsimsetup{exercise/template = bonus}
% set the boolean property to true
\begin{exercise}[bonus]
A bonus question.
\end{exercise}
```

**★ Bonus Exercise 7**

A bonus question.

**B.7. ...Print All Solutions Grouped by Section?**

Here is an idea how to get a list of all solutions grouped by the section the corresponding exercises are appearing in.
Solutions to the Exercises of Section 4

Solution 1
A first example for a solution.

Solutions to the Exercises of Section 8

Solution 5
The solution of the exercise that has not been printed.

Answers to the Problems of Section 13

Answer 1 My subtitle
This is the answer to problem 1.

Solutions to the Exercises of Section 15

Solution 6
Try to fill in these blanks. All of them are created by using the \blank command.

---

10. Taking care of the fact that we’re in the appendix now which means we can’t use \value{section}. Therefore this manual does \edef\lastsection{arabic{section}} right before \appendix
C. The xsimverb package

xsimverb version 0.4 (Feb 12, 2022)

xsim comes bundled with another package called xsimverb. This package loads a very small subset of \texttt{xsim} which allows to create environments that write their contents verbatim to external files. It provides the following commands (which of course are also available in \texttt{xsim}, too):

\begin{verbatim}
xSIMfilewritestart*{⟨file name⟩}
\end{verbatim}
Start writing to the file named \texttt{⟨file name⟩}. This should be the \textit{last} command in the \texttt{begin} definition of an environment. If is is used in an environment with arguments where the \texttt{last} argument is optional you should check if the optional argument is given and use the starred version if the test is negative. This is demonstrated in an example below using \texttt{xparse}'s \texttt{\textbackslash NewDocumentEnvironment}. \textit{If you want an environment with only an optional argument you should use \texttt{xparse}'s commands to define it. Due to the way how \texttt{\textbackslash newenvironment} scans for optional arguments you'll otherwise may end up with leading spaces gobbled from the first line in your environment.}

\begin{verbatim}
xSIMfilewritestop
\end{verbatim}
Stop writing to the file. This should be the \textit{first} command in the \texttt{end} definition of an environment.

\begin{verbatim}
xSIMsetfilebegin{⟨code⟩}
\end{verbatim}
This command can be used to write something to the external file \textit{before} the environment contents. Must be set before \texttt{xSIMfilewritestart} in the \texttt{begin} definition.

\begin{verbatim}
xSIMsetfilebeginX{⟨code⟩}
\end{verbatim}
Introduced in version 0.21 (Feb 12, 2022)
The same as \texttt{xSIMsetfilebegin} but expands its argument first.

\begin{verbatim}
xSIMsetfileend{⟨code⟩}
\end{verbatim}
This command can be used to write something to the external file \textit{after} the environment contents. Must be set before \texttt{xSIMfilewritestart} in the \texttt{begin} definition.

\begin{verbatim}
xSIMsetfileendX{⟨code⟩}
\end{verbatim}
Introduced in version 0.21 (Feb 12, 2022)
The same as \texttt{xSIMsetfileend} but expands its argument first.

\begin{verbatim}
xSIMgobblechars{⟨integer⟩}
\end{verbatim}
Determines how many characters are cut off of the beginning of each line of the environment body before it is written to the file. The default value is 0.

The following code shows an example of how to use those commands:

\begin{verbatim}
\documentclass{article}
\usepackage{xsimverb,listings}
\makeatletter
\NewDocumentEnvironment{example}{o}
\makeatother
\end{verbatim}
D. All Exercise Examples

The tmp file produced by the above example will contain the following three lines (if the file itself was called test.tex):

```
% file `test.tmp'
bla bla \LaTeX
% bye bye
```

D. All Exercise Examples

You will notice that some exercises from section 13.6 on page 41 look differently in this section. That is because all exercises of a type use the template that’s currently active. If you want exercises with a different look you should use different exercises types.

The following list is created with this code:

```
xsimsetup{exercise/template = bonus}
\printcollection[headings]{all exercises}
```
**Exercises**

**Exercise 1**
A first example for an exercise.

**Exercise 2** *This is a subtitle*
An exercise where some properties have been set.

**Exercise 3**
\GetExerciseProperty{id}: 3
\GetExerciseAliasProperty{ID}: 3
\GetExerciseProperty{ID}: 3

**Exercise 4**
\GetExerciseProperty{id}: 4
\GetExerciseAliasProperty{ID}: 4
\GetExerciseProperty{ID}: foo-bar

**Exercise 5**
This exercise will not be printed but the exercise counter will be incremented nonetheless. Its solution will be printed in the list of solutions.

**Exercise**
This exercise is added to the collection ‘foo’.

**Exercise**
This exercise is also added to the collection ‘foo’.

**Exercise**
So is this.

**Exercise**
As well as this one.
E. All Solution Examples

Exercise  The Subtitle

Exercise 6
Try to fill in ________________ blanks. All of them _______ by using the \blank command.

★ Bonus Exercise 7
A bonus question.

Problems

<table>
<thead>
<tr>
<th>Problem 1 My subtitle (5 points)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is a problem using a subtitle and points.</td>
</tr>
</tbody>
</table>

E. All Solution Examples

Solutions to the Exercises

Solution 1
A first example for a solution.

Solution 5
The solution of the exercise that has not been printed.

Solution 6
Try to fill in these blanks. All of them are created by using the \blank command.
Answers to the Problems

**Answer 1** *My subtitle*

This is the answer to problem 1.

### F. Example Documents Coming With This Package

The repository of this package\(^\text{11}\) currently includes 42 example documents demonstrating how different aspects of this package work or how different kinds of problems can be solved or how different kinds of layouts can be achieved as well as how to solve concrete problems that have come up in different \LaTeX{} forums.

Besides showing excerpts of the code and the resulting pdf the examples below also link to both the \texttt{tex} source the resulting pdf.

**Example 1: Create blank lines**

Links: \([\TeX]\) \([\text{PDF}]\)  

File: \texttt{xsim.blanks.tex}

```latex
7  solution/print = true ,
8  blank/filled-style = \underline{\textcolor{red}{\textcolor{#1}}}
9  }
10
11 \begin{document}
12 \begin{exercise}[points=3]
13 Erklären Sie den Begriff.
14
15 \end{exercise}
```

\[\text{Solution}][\text{Exercise}][\text{PDF}]

---

Example 2: Put headings in a box
Links: [tex] [PDF] File: xsim.boxed-headings.tex

\DeclareExerciseEnvironmentTemplate{custom}
{%
  \Needspace*{5\baselineskip}
  \begin{tcolorbox}
    \textbf{\XSIMmixedcase{\GetExerciseName}~\GetExerciseProperty{counter}.}\
    \textit{\GetExercisePropertyT{subtitle}{\textit{#1}}}\%
  \end{tcolorbox}
  \noindent
%
\}

Exercise 1.

Foo bar baz


Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Example 3: Create code examples
Links: [tex] [PDF] File: xsim.code-and-output.tex

\makeatletter
\NewDocumentEnvironment{example}{!o}
{%
  \XSIMgobblechars{2}%
\}

columns = fullflexible ,
commentstyle = \color{gray!70} ,
keywordstyle = \color{red!70!black}
\}
\makeatletter
\NewDocumentEnvironment{example}{!o}
{%
  \XSIMgobblechars{2}%
\}
Example 4: How to use collections

Links: [\TeX] [PDF]
File: xsim.collections.tex

\begin{exercise}[use,print]
outside before
\end{exercise}

Exercise 1
outside before
Easy

Exercise 2

foo one
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

Exercise 3

foo two
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

Example 5: Crossreferencing between problems and answers

Links: [\TeX] [PDF]
File: xsim.crossref.tex

\begin{exercise}

\subsection*{Solution 1}

\begin{exercise}

foo two
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo three
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo one
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo two
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo three
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo one
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo two
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo three
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo one
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo two
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.

foo three
Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst.
Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.
Example 6: Exercises as a description list

Links: [\LaTeX] [PDF]  
File: xsim.description-list.tex

\begin{verbatim}
\xsimsetup{
  exercise/template=item,
  solution/template=item,
  print-solutions/headings-template=none
}
\newenvironment{exercises}{\section{Exercises}\description}{\enddescription}

\begin{exercises}
\end{exercises}

\begin{exercises}
\end{exercises}

\end{verbatim}

Example 7: A custom point scheme

Links: [\LaTeX] [PDF]  
File: xsim.different-point-types.tex

\begin{verbatim}
\newcommand*{\printA}{\TotalExerciseGoal{A}{~A point}{~A points}}
\newcommand*{\printC}{\TotalExerciseGoal{C}{~C point}{~C points}}
\newcommand*{\printE}{\TotalExerciseGoal{E}{~E point}{~E points}}
\usepackage{needspace}
\DeclareExerciseEnvironmentTemplate{custom}{% 
  \par\vspace{\baselineskip}}
\newenvironment{custom}{\begin{custom}}{\end{custom}}

\begin{custom}
\item Differentiate $y = 3x^2 + 5x + 3$. (1/0/0)
\item Find the equation of the tangent line to the function $y = \frac{x}{2}$ at $x = 2$. (2/1/0)
\item Prove that the derivative of a constant is zero. (0/1/2)
\end{custom}
\end{verbatim}
Example 8: Difficulty levels

Links: [\TeX] [PDF]

Example Documents Coming With This Package

Example 8: Difficulty levels

\DeclareExerciseEnvironmentTemplate{custom}
{
\subsection*
\XSIMmixedcase {\GetExerciseName}\nobreakspace
\GetExerciseProperty{counter}\%
\IfExercisePropertySetT{difficulty}

Exercise 1 (easy)
An easy question.

Exercise 3 (hard)
Now let's see if you can solve this one.

Example 9: Floating exercises and a list of exercises

Links: [\TeX] [PDF]

Example 9: Floating exercises and a list of exercises

\listname={List of Exercises},
\name=Exercise,
\placement=htp,
\{ex\}

\DeclareExerciseEnvironmentTemplate{float}
{%
\ex
\captionsetup{labelformat=empty,
singlelinecheck=false,listformat=empty}
Example 10: Using the grade distribution macros

\begin{verbatim}
\DeclareExerciseProperty{grade}
\newcommand\printgrades{\begin{enumerate}
\ForEachUsedExerciseByType{\GetExercisePropertyT{grade}}{
\item \XSIMmixedcase{GetExerciseName} \#1
}\end{enumerate}}
\end{verbatim}

\begin{verbatim}
\begin{enumerate}
\item[\textbf{Exercise 1}] \texttt{Pythagoras}
\item[\textbf{Exercise 2}] \texttt{Another Problem}
\item[\textbf{Exercise 3}] \texttt{Yet Another Problem}
\end{enumerate}
\end{verbatim}

Example 11: Give hints

\begin{verbatim}
\DeclareExerciseProperty{hint}
\% we'll use a description list for the hints
\newcommand\printhints{\begin{description}
\ForEachUsedExerciseByType{\GetExercisePropertyT{hint}}{
\item \XSIMmixedcase{\GetExerciseName} \#1
}\end{description}}
\end{verbatim}

\begin{verbatim}
\begin{description}
\item[\textbf{Exercise 1}] This is a hint to the first problem.
\item[\textbf{Exercise 3}] This is a hint to the third problem.
\end{description}
\end{verbatim}
Example 12: Use listings in exercises
Links: [\TeX] [PDF] File: xsim.listings.tex

\lstset{
  frame=single,
  xleftmargin=20pt,
  numbers=left,
  numberstyle=\small,
  tabsize=2,
  breaklines,
  showspaces=false,
}

Exercise 1 /6 p.
Consider the following C program.
1 # include <stdio .h>
2
3 int main ( int argc , char * argv [] ) {
4   printf ( " hello , world \n" );
5 }

Example 13: A custom list of exercises
Links: [\TeX] [PDF] File: xsim.listofexercises.tex

\exercise/within=chapter,
\exercise/template=theorem ,
\exercise/the-counter=\thechapter.\arabic{exercise}
}

\DeclareExerciseEnvironmentTemplate{theorem}
{
\par addvspace{\baselineskip}
\noindent

Chapter 1
kinetic
1.1
1.2 (Foo Bar)
1.3
1.4
...
Example 14: Multiplechoice exercises

Example 15: Sum of points
Example 16: Random exercises from a collection

Links: [TEX] [PDF]  
File: xsim.randomexercises.tex

\begin{exercise}[ID=A]  
exercise A
\end{exercise}

\begin{solution}  
solution A
\end{solution}

\begin{exercise}[ID=B]  
exercise B
\end{exercise}

Example 17: Various aspects of

Links: [TEX] [PDF]  
File: xsim.various.tex

\begin{exercise}
\begin{solution}
\begin{exercise}
\begin{solution}
\begin{exercise}
\begin{solution}
\begin{solution-env = hint ,
exercise-name = Question ,
solution-name = Hint ,
exercise-template = default ,
solution-template = default ,
counter = exercise % shares a counter with
the 'exercise' type
\end{exercise}
\end{exercise}
\end{exercise}
\end{exercise}
\end{exercise}
\end{exercise}
\end{solution}
\end{solution}
\end{solution}
\end{solution}
\end{solution}
\end{exercise}
\end{exercise}

\DeclareExerciseType{problem}{

Points reached
Exercise 1 4
Exercise 2 5
Exercise 4 0
Question 3 0
Problem 1 0
Problem 2 2
Problem 3 1
total 12

Total: 12 points
from questions: 0 points
from exercises: 9 points
from problems: 3 points
Total bonus: 1 point

Exercise 1
This is the solution to exercise 1.

F. Example Documents Coming With This Package

Example 18: Exercises like theorems
Links: [\TeX] [PDF] [forum]  
File: xsim.texsx-13635.tex

\begin{document}
\begin{exercise}
foo
\end{exercise}
\begin{exercise}
bar
\end{exercise}
\end{document}

Example 19: Random/custom order of exercises
Links: [\TeX] [PDF] [forum]  
File: xsim.texsx-155630.tex

\begin{document}
\begin{exercise}
foo
\end{exercise}
\begin{exercise}
bar
\end{exercise}
\end{document}

1 Prime Numbers
A prime number is a positive integer other than 1 that is only divisible by 1
and itself.

As you will show in Exercise 1.1, there are infinitely many primes.
The number of primes that are smaller than a given natural number \(n\) is
denoted \(\pi(n)\).

Exercises
Exercise 1.1 (Euclid's Theorem). Show that there are infinitely many prime
numbers.

Exercise 2.1. Find an asymptotic formula for \(\pi(n)\).
Hint: You might find Exercise 2.1 helpful.

2 Zeta function
The zeta function is given by
\[\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s},\]
where \(s\) is a complex number with real part bigger than 1. For example
\[\zeta(2) = \frac{\pi^2}{6}.\]
F. Example Documents Coming With This Package

Example 20: Exercises and solutions in a tcolorbox

\begin{exercise}[subtitle = \texttt{Codeless Question},points=10]
A question without code, worth 10 points. Subtitle and point values are in correct place.
\end{exercise}

\begin{solution}
Solution 1
\end{solution}

Example 21: Using pythontex

\begin{exercise}[subtitle = \texttt{Codeful Question},points=15]
Now with PythonTeX:
print("hello, world!")
sum = 0
for j in range(0,100):
    sum += j
print(sum)
\end{exercise}
Example 22: Print solutions per chapter/section

Links: [\TeX] [PDF] [forum]  
File: xsim.texp--305110.tex

\begin{document}
\part{EXCERCISES}
\chapter{Topic 1}
\section{Section}
\end{document}

Example 23: Adapt how points are printed

Links: [\TeX] [PDF] [forum]  
File: xsim.texp--308883.tex

\begin{document}
\begin{exercise}[points=2.5]
foo
\end{exercise}
\end{document}
F. Example Documents Coming With This Package

Example 24: Another tcolorbox example
Links: [TeX] [PDF] [forum]

\usepackage{tasks}
\usepackage{xsim}
\usepackage{tcolorbox}
\tcbuselibrary{breakable, skins}
\settasks{ label = \arabic*. }
\begin{exercise}
\begin{tcolorbox}
\begin{align*}
    & E = \{1, 2, 3, 4, 5, 6, 7\} \\
    & A = \{1, 2, 3, 4\}, \quad B = \{4, 5, 6, 7\}, \\
    & C = \{1, 3, 5, 7\}, \quad D = \{2, 3, 4, 5, 6\}.
\end{align*}
\end{tcolorbox}
\end{exercise}

1. Compute the derivative of the following function:
   \( f(x) = \sin((\sin x)^2) \)

2. Compute the derivative of the following function:
   \( f(x) = \sin((\sin x)^2) \)

3. Compute the derivative of the following function:
   \( f(x) = \sin((\sin x)^2) \)

4. Solution:
   \{5, 6, 7\}

Example 25: Fancy tcolorbox and crossreferencing
Links: [TeX] [PDF] [forum]

\begin{exercise}
\begin{tcolorbox}
\begin{align*}
    & E = \{1, 2, 3, 4, 6, 7\} \\
    & A = \{3, 4, 5\}.
\end{align*}
\end{tcolorbox}
\end{exercise}

Chapter one
The First Chapter

Exercise 1
Compute the derivative of the following function:
\( f(x) = \sin((\sin x)^2) \)

The solution of this exercise is on page 4.

Exercise 2
Compute the derivative of the following function:
\( f(x) = \sin((\sin x)^2) \)

The solution of this exercise is on page 4.
Example 26: Custom layout

```
exercise/the-counter = \thesection.\arabic{exercise},
exercise/template=cyan-box,
exercise/name=Example,
solution/template=red,
solution/print=true
```

```
\DeclareExerciseEnvironmentTemplate{cyan-box}{\begin{tcolorbox}[arc=0mm,boxrule=1pt,
colback=white,colframe=blue,leftrule=3mm]
My first set of exercises

EXAMPLE 1.1 Prove that
\begin{align*}
\Delta(f_{ij} f_{ij}) &= \nabla_k f_{ij} + f_{ij} f_k \\
&\quad \nabla_i R_{jk} - \nabla_k R_{ij}
\end{align*}

SOLUTION From ...

EXAMPLE 1.2 Prove that Paulinho is smart.

SOLUTION All ducks are smart. Paulinho is a duck. Therefore, Paulinho is smart.

EXAMPLE 1.3 Prove that Paulinho is smart.

SOLUTION All ducks are smart. Paulinho is a duck. Therefore, Paulinho is smart.
```

Example 27: An empty box for points

```
\usepackage{tgpagella}
\usepackage{xsim,needspace,adjustbox,scrextend}
\xsimsetup{
exercise/the-counter = \arabic{exercise},
exercise/template = square
}
```

```


1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing ...

4. My first set of exercises

1.1. Prove that ... is smart.

1.2. Prove that Paulinho is smart.

1.3. Prove that Paulinho is smart.
```

1. My first set of exercises

EXAMPLE 1.1 Prove that ...

SOLUTION From ...

EXAMPLE 1.2 Prove that Paulinho is smart.

SOLUTION All ducks are smart. Paulinho is a duck. Therefore, Paulinho is smart.

EXAMPLE 1.3 Prove that Paulinho is smart.

SOLUTION All ducks are smart. Paulinho is a duck. Therefore, Paulinho is smart.

2. My second set of exercises

EXAMPLE 2.1 Prove that Paulinho is smart.

SOLUTION All ducks are smart. Paulinho is a duck. Therefore, Paulinho is smart.

EXAMPLE 2.2 Prove that Paulinho is smart.

SOLUTION All ducks are smart. Paulinho is a duck. Therefore, Paulinho is smart.
```

1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing ...


1. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing ...

Example 28: Layout adjustments

Links: [TeX] [PDF] [forum]
File: xsim.texsx-369803.tex

```latex
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage[left=2cm,right=2.5cm,top=2.5cm, bottom=2cm]{geometry}
\usepackage[xsim,siunitx]
\DeclareExerciseTagging{difficulty}
\DeclareExerciseEnvironmentTemplate{custom}{%
\subsection*{Aufgabe 1 (Widerstandswürfel)}
Gegeben ist ein Würfel, wobei jede der Kanten einen Widerstand von $R = 1 \, \Omega$ hat.
Wie groß ist der Widerstand entlang einer Raumdiagonalen?
Lösung 1
Wir wollen den Widerstand zwischen den Punkten $X$ und $Y$ bestimmen, also entlang der Raum-
diagonalen (siehe Abb. ??). Weil die Raumdiagonale eine Symmetrieachse ist, sollte das Problem
symmetrisch sein, und deswegen eine recht einfache Lösung haben.
```

Example 29: Minimalistic layout

Links: [TeX] [PDF] [forum]
File: xsim.texsx-370642.tex

```latex
\par
\xsimsetup{exercise/template=simple}
\begin{document}
\begin{exercise}\label{eq1}
Let $X$ be such that\ldots
\end{exercise}
```

1. Let $X$ be such that.
2. In this exercise consider $Y = 2$.
3. Consider $X$ as in exercise 1 (I would like to see the issue number 1).
F. Example Documents Coming With This Package

Example 30: Exercises and sub-exercises

Links: [TeX] [PDF] [forum]  
File: xsim.texsx-391530.tex

\begin{example}
\begin{verbatim}
7    solution-env = answer ,
8    exercise-name = Question ,
9    solution-name = Answer ,
10   exercise-template = item ,
11   solution-template = item
12 \} \end{verbatim}

\end{example}

\begin{example}
\begin{verbatim}
14 \DeclareExerciseProperty{title}
\end{verbatim}
\end{example}

Example 31: Different aspects of exercises, highlighted solutions

Links: [TeX] [PDF] [forum]  
File: xsim.texsx-395273.tex

\begin{example}
\begin{verbatim}
7 \DeclareExerciseTagging{level}
8 % declare a template which typesets exercises differently according to given properties:
9 \DeclareExerciseEnvironmentTemplate{exercise}
10 \} \%
11 \renewcommand*{theenumi\{theexercise.\arabic{enumi}}}\%
12 \par\addvspace{\baselineskip}
13 \Needspace*(2\baselineskip)
\end{verbatim}
\end{example}
Example 32: Flushright Solutions

Links: [TeX] [PDF] [forum]

File: xsim.texsx-466584.tex

\DeclareExerciseEnvironmentTemplate{exandsol}{{}%
\par\vspace{\baselineskip}\Needspace*{2\baselineskip}
\noindent\sffamily\textbf{\XSIMmixedcase{\GetExerciseName}~\GetExerciseProperty{counter}}%
\GetExercisePropertyT{subtitle}{\hspace{3em}{\small#1}}\par
\normalfont
}

Exercise 1-31

Factorize as much as possible the following expressions.

(a) \((2x - 3)^2 - (3x - 2)^2\)

(b) \((x^2 - 25) - 2(5 - x)(x + 6)\)

(c) \(2x(x + 2) + (x + 1)^2 + 2\)

Solutions:

(a) \(- (x + 5)(5x - 1)\)

(b) \((x + 5)(3x - 17)\)

(c) \(3(x + 1)^2\)

Example 33: Multiple choice questions with automated solutions (1)

Links: [TeX] [PDF] [forum]

File: xsim.texsx-498299.tex

{\item \GetExerciseProperty{counter}}

\DeclareExerciseProperty{answer}

\newcommand*{\answer}[1]{% 
\XSIMexpandcode{\SetExerciseProperty{answer}{\unexpanded{\textit{(#1)}}}}

1. What is the product of \(-2\) and \(3\)?
   \(-6\) (a) \(6\) (b) \(5\) (c) \(-5\) (d)

2. What is the sum of the sides of a polygon called?
   \(\text{Perimeter}\) (a) \(\text{Leg}\) (b) \(\text{Volume}\) (c)

3. What is the sum of \(-2\) and \(-3\)?
   \(-6\) (a) \(6\) (b) \(5\) (c) \(-5\) (d)

2 Answers

1. (a) \(-6\) 2. (b) \(\text{Perimeter}\) 3. (d) \(-5\)
F. Example Documents Coming With This Package

Example 34: Multiple choice questions with automated solutions (2)

Links: [TeX] [PDF] [forum] File: xsim.texsx-549540.tex

\geometry{ a4paper, left=5mm, right=5mm, top =15mm }
\usepackage{amsmath,amssymb}
\usepackage{multicol}
\setlength{\columnsep}{1mm}
\setlength{\columnseprule}{0.2pt}
\usepackage[most]{tcolorbox}

Test 1

Angles in parallel lines

1. If the ratio of two supplementary angles is \( \frac{4}{11} \), what is the measure of the small angle?
   - (A) 36°
   - (B) 44°
   - (C) 48°
   - (D) 52°
   - (E) 60°

2. Is this really a question?
   - (A) yes
   - (B) no
   - (C) maybe
   - (D) no idea

3. This really is a question!
   - (A) yes
   - (B) no
   - (C) maybe
   - (D) no idea

Answers
1. (C) 48°
2. (A) yes
3. (B) no

Example 35: Exercises at the end of section and sectionwise solutions

Links: [TeX] [PDF] [forum] File: xsim.texsx-576998.tex

\textbf{Chapter 14}

14.1 Cats as pets

Text text text. . .
Text text text. . .
Text text text. . .

Exercises 14.1

1. Draw a cat.
2. Bathe a cat.
3. Trace the history of cats through time.

14.2 Frogs as pets

Text text text. . .
Text text text. . .
Text text text. . .

Exercises 14.2

1. Lick a frog.
2. Bathe a frog.
3. Feed a frog.
Example 36: Multiple hints per exercise with backlinks

\DeclareExerciseProperty{hints}
\newcounter{hint}
\renewcommand\theHhint{\ExerciseID.\arabic{hint}}
\newcounter{step}
\NewDocumentEnvironment{hint}{}{%
\stepcounter{hint}%
\XSIMsetfilebegin{Exercises
Exercise1.1: Hints: 1, 2
Description of the exercise. Run the following command
print("Hello world")
Verbatim &^%$&
Exercise1.2:
Hint: 1
Another exercise
2 Hints
Hint 1 to exercise 1.1
Back to exercise 1.1
Run the example from the command line with the python command.
Hint 2 to exercise 1.1
Back to exercise 1.1
The solution is 42.
Hint 1 to exercise 1.2
Back to exercise 1.2
The first ten million years were the worst. Some
.

Example 37: Custom list of exercises

\xsimsetup{
exercise/name = Aufgabe ,
solution/name = Lösung ,
exercise/within = section ,
exercise/the-counter = \thesection.\arabic{exercise} ,
exercise/template = mine
}
\newif\iflist

\begin{Exercises
Exercise 1.1
Eine erste Aufgabe
Exercise 1.2
Eine zweite Aufgabe
1.1 Erstes Unterkapitel
Aufgabe 1.3
Eine Aufgabe in einem Unterkapitel
Aufgabe 1.4
Noch eine Aufgabe in einem Unterkapitel
1.1.1 Tiefer geschachteltes Unterkapitel
Aufgabe 1.5
Noch eine Aufgabe
1.1.2 Weiter geschachtelt
Aufgabe 1.6
Eine weitere Aufgabe
1.2 Zweites Unterkapitel
Aufgabe 1.7
Und eine weitere Aufgabe
\end{Exercises

79
Example 38: Indicate difficulty level

Links: [\TeX] [PDF] [forum]

File: xsim.texwelt-15093.tex

\DeclareExerciseTagging{AFB}
\DeclareExerciseEnvironmentTemplate{myexam}
\{
  \par\vspace{\baselineskip}
  \Needspace*{3\baselineskip}
  \noindent
  \textbf{Aufgabe} \GetExerciseProperty{counter}.
  \textit{textit{#1}} \hfill
  \GetExercisePropertyT{subtitle}{\quad}
  \textbf{Schwierigkeit: 1 3 P.}
  \Das ist eine sehr tolle Frage.
\}
\Needspace*{3\baselineskip}
\noindent
\textbf{Aufgabe 1.}
\Das ist eine sehr tolle Frage.
\textbf{Aufgabe 2.}
\Das ist eine sehr tolle Frage.
\textbf{Aufgabe 3.}
\Das ist eine sehr tolle Frage.
\textbf{Aufgabe 4.}
\Das ist eine sehr tolle Frage.

Example 39: Long and short solutions

Links: [\TeX] [PDF] [forum]

File: xsim.texwelt-23968.tex

% new environment:
\NewDocumentEnvironment{shortsolution}{+b}
{\edef\ExerciseType{\csname g_xsim_
  exercise_type_tl\endcsname}\
  \edef\ExerciseID{\csname g_xsim_exercise_
  _id_tl\endcsname}\
  \SetExerciseProperty{shortsolution}{#1}\
}
\NewDocumentEnvironment{shortsolution}{+b}
{\edef\ExerciseType{\csname g_xsim_
  exercise_type_tl\endcsname}\
  \edef\ExerciseID{\csname g_xsim_exercise_
  _id_tl\endcsname}\
  \SetExerciseProperty{shortsolution}{#1}\
}
\NewDocumentEnvironment{shortsolution}{+b}
{\edef\ExerciseType{\csname g_xsim_
  exercise_type_tl\endcsname}\
  \edef\ExerciseID{\csname g_xsim_exercise_
  _id_tl\endcsname}\
  \SetExerciseProperty{shortsolution}{#1}\
}
\NewDocumentEnvironment{shortsolution}{+b}
{\edef\ExerciseType{\csname g_xsim_
  exercise_type_tl\endcsname}\
  \edef\ExerciseID{\csname g_xsim_exercise_
  _id_tl\endcsname}\
  \SetExerciseProperty{shortsolution}{#1}\
}
Example 40: Different versions for students and teachers

Links: [\TeX] [PDF] [forum]  
File: xsim.golatex-80640.tex

\begin{verbatim}
\setlength{\breite}{160mm}
\setlength{\hoehe}{80mm}
\usepackage{geometry}
\usepackage[bitstream-charter]{mathdesign}
\newlength{\hoehe}
\setlength{\hoehe}{80mm}
\setlength{\breite}{160mm}
\newlength{\hoehe}
\setlength{\hoehe}{80mm}
\usepackage{geometry}
\usepackage[bitstream-charter]{mathdesign}
\end{verbatim}

Für die Schülerausgabe sollen Häuschen (Grid) mit Seitenlänge 4 mm gesetzt werden.  
Für die Lehrerausgabe sollen statt Häuschen die Lösung in z. B. einer Box geschrieben werden. Dafür soll die Lösung (bezogen auf dieses Beispiel) auch in einer Box mit der exakten Breite 160 mm und der exakten Höhe 80 mm gesetzt werden. Weiter soll natürlich die Position der Lösungsbox und der Häuschenbox exakt identisch sein.

Lösung: Hier soll die Lösung stehen:

\[ E = mc^2. \]

Example 41: Another custom layout with rules

Links: [\TeX] [PDF] [forum]  
File: xsim.golatex-91339.tex

\begin{verbatim}
\xsimsetup{
exercise/within = section ,
exercise/the-counter = \thesection.\arabic{exercise} ,
print-solutions/headings-template=none
}
\SetExerciseParameters{exercise}{
exercise-template = mine ,
solution-template = mine
}
\end{verbatim}

Wellenausbreitung im Vakuum und in Materie

1.1 Maxwellsche Gleichungen

\[ \nabla \cdot \vec{E}(\vec{r}, t) = \rho (\vec{r}, t) \]
\[ \varepsilon_0 \]

Aufgabe 1.1  

Eine weitere Aufgabe

Lösungen

Lösung 1.2  

Solution Number 2

Wellenausbreitung im Vakuum und in Materie

2.1 Maxwellsche Gleichungen

\[ \nabla \cdot \vec{E}(\vec{r}, t) = \rho (\vec{r}, t) \]
\[ \varepsilon_0 \]

Aufgabe 2.1  

Eine weitere Aufgabe

Lösungen

Lösung 2.2  

Solution Number 2

1 Wellenausbreitung im Vakuum und in Materie

1.1 Maxwellsche Gleichungen

\[ \nabla \cdot \vec{E}(\vec{r}, t) = \rho (\vec{r}, t) \]
\[ \varepsilon_0 \]

Aufgabe 1.1  

Eine weitere Aufgabe

Lösungen

Lösung 1.2  

Solution Number 2

Example 42: Different ideas for exams

Links: [텍스] [PDF] [github]

File: xsim.issues-49.tex

G. References

url: https://www.ctan.org/pkg/babel/.

polyglossia. version 1.53, Apr. 12, 2021 (or newer).
url: https://www.ctan.org/pkg/polyglossia/.

url: https://www.ctan.org/pkg/booktabs/.

[L3Pa] The \LaTeX\3 Project Team. l3kernel. Feb. 5, 2022 (or newer).
url: https://www.ctan.org/pkg/l3kernel/.

[L3Ph] The \LaTeX\3 Project Team. l3packages. Feb. 5, 2022 (or newer).
url: https://www.ctan.org/pkg/l3packages/.

url: https://www.ctan.org/pkg/array/.

[Nie19] Clemens Niederberger. exsheets. version 0.21k, Sept. 30, 2019 (or newer).
url: https://ctan.org/pkg/exsheets.

[Nie22] Clemens Niederberger. translations. version 1.12, Feb. 5, 2022 (or newer).
url: https://www.ctan.org/pkg/translations/.

82
G. References

VARIOUS. Questions tagged ‘exsheets’.
URL: http://tex.stackexchange.com/questions/tagged/exsheets (visited on 02/21/2020).
# H. Index

## Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>⟨type⟩ (option class)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>activatecollection</td>
<td></td>
<td>23, 25</td>
</tr>
<tr>
<td>addbonus</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>addpoints</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>AddtoExerciseGoal</td>
<td></td>
<td>16f.</td>
</tr>
<tr>
<td>AddtoExerciseGoalPrint</td>
<td></td>
<td>16f.</td>
</tr>
<tr>
<td>AddtoExerciseTypeGoal</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>AddtoExerciseTypeGoalPrint</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>array (package)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>babel (package)</td>
<td></td>
<td>39, 48</td>
</tr>
<tr>
<td>bezos, Javier</td>
<td></td>
<td>39, 48</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>4, 6</td>
</tr>
<tr>
<td>blank</td>
<td></td>
<td>28, 51 f., 56, 60f.</td>
</tr>
<tr>
<td>blank-style (property)</td>
<td></td>
<td>12, 15, 17, 37f.</td>
</tr>
<tr>
<td>booktabs (package)</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>braams, Johannes</td>
<td></td>
<td>39, 48</td>
</tr>
<tr>
<td>Carlisle, David</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>chapter</td>
<td></td>
<td>13, 24</td>
</tr>
<tr>
<td>chapter (property)</td>
<td></td>
<td>28 f.</td>
</tr>
<tr>
<td>chapter-value (property)</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Charette, François</td>
<td></td>
<td>39, 48</td>
</tr>
<tr>
<td>clear-aux</td>
<td></td>
<td>4, 9, 54</td>
</tr>
<tr>
<td>collect (option class)</td>
<td></td>
<td>23</td>
</tr>
<tr>
<td>collect</td>
<td></td>
<td>20, 23</td>
</tr>
<tr>
<td>collectexercises</td>
<td></td>
<td>23–26</td>
</tr>
<tr>
<td>collectexercisesstop</td>
<td></td>
<td>23–26</td>
</tr>
<tr>
<td>collection</td>
<td></td>
<td>26f.</td>
</tr>
<tr>
<td>counter (parameter)</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>counter (property)</td>
<td></td>
<td>12, 14, 19, 37f.</td>
</tr>
<tr>
<td>counter-value (property)</td>
<td></td>
<td>12, 14</td>
</tr>
<tr>
<td>deactivatecollection</td>
<td></td>
<td>23, 25</td>
</tr>
<tr>
<td>DeclareExerciseCollection</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>DeclareExerciseEnvironmentTemplate</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>DeclareExerciseGoalValuePrint</td>
<td></td>
<td>15, 17</td>
</tr>
<tr>
<td>DeclareExerciseHeadingTemplate</td>
<td></td>
<td>40, 45</td>
</tr>
<tr>
<td>DeclareExerciseIfSetTF</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>DeclareExerciseProperty</td>
<td></td>
<td>14, 54, 66, 68, 76f., 79</td>
</tr>
<tr>
<td>DeclareExercisePropertyAlias</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>DeclareExerciseTableAlias</td>
<td></td>
<td>40, 45f.</td>
</tr>
<tr>
<td>DeclareExerciseTagging</td>
<td></td>
<td>18, 63, 75f., 80</td>
</tr>
<tr>
<td>DeclareExerciseTranslation</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>DeclareExerciseTranslations</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>DeclareExerciseType</td>
<td></td>
<td>9f., 12, 19, 43, 69</td>
</tr>
<tr>
<td>end-hook</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td>equation (environment)</td>
<td></td>
<td>25</td>
</tr>
<tr>
<td>exclude</td>
<td></td>
<td>27</td>
</tr>
<tr>
<td>exercise (environment)</td>
<td></td>
<td>4, 6f., 9, 12, 19</td>
</tr>
<tr>
<td>exercise (option class)</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>exercise-body (property)</td>
<td></td>
<td>7, 14</td>
</tr>
<tr>
<td>exercise-env (parameter)</td>
<td></td>
<td>10ff.</td>
</tr>
<tr>
<td>exercise-heading (parameter)</td>
<td></td>
<td>11, 36</td>
</tr>
<tr>
<td>exercise-name (parameter)</td>
<td></td>
<td>10f., 36</td>
</tr>
<tr>
<td>exercise-template (parameter)</td>
<td></td>
<td>11f., 39</td>
</tr>
<tr>
<td>\ExerciseCollection</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>\ExerciseGoalValuePrint</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>\ExerciseID</td>
<td></td>
<td>37, 63, 79f.</td>
</tr>
<tr>
<td>\ExerciseParameterGet</td>
<td></td>
<td>36, 46f.</td>
</tr>
<tr>
<td>\ExerciseParameterIfSetTF</td>
<td></td>
<td>36</td>
</tr>
<tr>
<td>\ExercisePropertyGet</td>
<td></td>
<td>35, 78</td>
</tr>
<tr>
<td>\ExercisePropertyGetAlias</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>\ExercisePropertyGlobalSave</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>\ExercisePropertyIfSetTF</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>\ExercisePropertySave</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>exercises-name (parameter)</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>\ExerciseSetExpandedProperty</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>\ExerciseSetProperty</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>\ExerciseTableCode</td>
<td></td>
<td>45–48</td>
</tr>
<tr>
<td>\ExerciseTableType</td>
<td></td>
<td>37, 45ff.</td>
</tr>
</tbody>
</table>
INDEX

\ExerciseText .................................. 37
\ExerciseType ............................... 36 f., 45–48, 80
expl3 (package) ......................... 2
exsheets (package) ............... 3, 17, 32, 43, 51, 53

F
F A R, Simon .............................. 2
file-extension ............................... 8
filecontents (environment) ....... 8
fill .............................................. 52
filled-style ............................... 51
final ........................................ 4
\ForEachExerciseTag ...................... 36
\ForEachExerciseTranslation ......... 48
\ForEachPrintedExerciseByID ... 38
\ForEachPrintedExerciseByType .. 37
\ForEachUsedExerciseByID ......... 38
\ForEachUsedExerciseByOrder ..... 38
\ForEachUsedExerciseByType 37, 45, 47, 66, 78

G
\GetExerciseAliasProperty .......... 15, 35
\GetExerciseBody ...................... 34
\GetExerciseHeadingF ............... 11, 36, 41
\GetExerciseIdForProperty ....... 34
\GetExerciseName ................. 36, 41–44, 55, 62 f., 65 f., 70 f., 77
\GetExerciseParameter .......... 35, 45, 47
\GetExerciseParameterTF .......... 35
\GetExerciseProperty ........... 15, 34, 41–45, 55, 62, 65, 70 f., 77 f., 80
\GetExercisePropertyTF .......... 34
\GetExercisePropertyT41–45, 55, 62, 66, 70, 77 f., 80
\GetExerciseTypeForProperty .......................... 34
\GlobalSaveExerciseProperty .... 35
goal-print ............................. 16 f.
grading-table (option class) .... 29, 40
\gradingtable ......................... 29 f., 40

H
heading ........................................ 21
headings .................................. 23 f., 28
headings-template ..................... 24, 28, 40

I
ID (property) ......................... 12, 14, 21, 27, 38
id (property) ......................... 12, 14, 19, 21, 27, 34, 38
\IfExerciseBooleanPropertyTF ..... 35
\IfExerciseBooleanPropertyT ..... 55
\IfExerciseBooleanPropertyTF ..... 68
\IfExerciseGoalTF ............... 33
\IfExerciseGoalSingularTF ..... 33
\IfExerciseGoalSingularTF 42, 44 f.
\IfExerciseGoalsSumTF .......... 33
\IfExerciseGoalsSumF ............. 17
\IfExerciseGoalsSumTF .......... 68
\IfExerciseGoalsSumTF .......... 68
\IfExerciseGoalTF ............... 33
\IfExerciseParameterSetTF .... 36
\IfExercisePropertyExistTF .... 34
\IfExercisePropertySetTF .... 34
\IfExercisePropertySetT ........ 34
\IfExerciseTagSetTF ............. 36
\IfExerciseTopicSetTF ......... 36
\IfExerciseTypeGoalsSumTF .... 33
\IfExistSolutionTF ............. 37
\IfInsideSolutionTF ............. 37
\IfInsideSolutionF ............. 41 ff., 45, 55, 70
\IfInsideSolutionTF ............. 63, 71, 80
\IfSolutionPrintTF .............. 37
ignore-untagged .................. 18

L
l3kernel (bundle) ............... 2
l3packages (bundle) .......... 2
line-increment ....................... 52
line-minimum-length .......... 52
linespread .......................... 52
\ListExerciseTags .............. 36
load-style .......................... 41
\loadxsimstyle .................... 40 f.
LPPL ........................................ 2

M
match-all ................................ 23
match-any ................................ 23
MITTELBA CH, Frank ............... 2
INDEX

N
\name .............................................. 20 f.
\NewDocumentEnvironment .................. 57
\Niederberger, Clemens ..................... 2 f.
\numberof (exercise-env) .................... 9
\numberofexercises ......................... 10
\numberofusedexercises ..................... 37, 47

P
\package (option class) .................... 5
\page (property) .......................... 13, 19, 24
\page-value (property) .................... 13
\ParameterValue .......................... 8 f., 54
\points (property) ......................... 12, 15, 17, 37 f.
\points8 f., 14, 16 f., 41–47, 55, 61, 64, 68, 71 f.
\polyglossia (package) ..................... 39, 48
\post-hook .................................. 21
\pre-hook .................................. 21
\print .................................. 6 f., 19 ff., 24, 27, 37
\print (property) ......................... 12, 19, 24
\print-collection (option class) ........... 24, 40
\print-solutions (option class) ............ 29, 40
\print! (property) ..................... 13, 19
\printallsolutions ......................... 27
\printbonus .................................. 17
\printcollection ......................... 23 ff., 40, 58
\printexercise ................................ 21 f.
\printgoal ................................ 16, 42, 44–47
\printpoints ................................ 16
\printrandomexercises .................... 26 f.
\printsolution .............................. 27 ff.
\printsolutions ............................ 27 ff., 40, 56
\printsolutionstype ....................... 27 f., 40
\printtotalbonus ......................... 17
\printtotalpoints ......................... 16
\PropertyValue .......................... 34, 41 f., 44 f., 55
\ProvideExerciseTagging ................. 18

Q
Questions tagged 'exsheets' .................. 3

R
\Reutenauer, Arthur ......................... 39, 48

S
\SaveExerciseProperty ...................... 35
\scale .................................... 51 f.
\section (property) ...................... 13, 24
\section-value (property) .............. 13
\sectioning (property) .................... 14
\SetExerciseParameter .................... 12, 20 f.
\SetExerciseParameters ................. 12, 81
\SetExerciseProperty .................... 34, 77, 80, 82
\SetExpandedExerciseProperty .......... 34
\solution (environment) .......... 4, 6, 9
\solution (property) ................... 13, 37
\solution-body (property) ............ 7, 14
\solution-counter (parameter) ........ 11
\solution-env (parameter) ............ 10 f.
\solution-heading (parameter) ......... 11, 36
\solution-name (parameter) .......... 10 f., 36
\solution-name (parameter) .......... 10 f., 36
\solution-template (parameter) ...... 11, 39
\solutions-name (parameter) ........ 10
\sort .................................. 27
\split-aux-lists ......................... 9
\style .................................. 51
\style file ......................... 32, 40 f.
\subtitle (property) .................. 12, 37 f.

T
\tags .................................. 18
\tags (property) .................... 13, 18, 36
\template ................................ 20, 29, 40
\The \LaTeX Project Team ............. 2
\the-counter ................. 12, 20 f.
\the-counter (parameter) ...... 11
\theexercise ......................... 76
\topics ................................ 18
\topics (property) .................. 13, 18, 36
\TotalExerciseGoal .................... 16 f., 34, 46 f., 64, 68
\TotalExerciseGoals ................... 16 f.
\TotalExerciseTypeGoal ............. 15 ff., 34, 46 ff.
\TotalExerciseTypeGoals .............. 15

translations (package) .................... 2
INDEX

type ........................................... 29 f.
type (property) ............................. 34

U
use ............................................. 19 f.
use (property) ............................... 13, 19, 24
use-aux ........................................ 4, 9, 54
use-files ....................................... 4, 7, 54
use-unmatched ................................ 18
use-unmatched ................................ 18
use! (property) ............................... 13, 19
used (property) .............................. 13, 24
\UseExerciseTags ......................... 36
\UseExerciseTemplate .................... 36, 68

V
VARIOUS ........................................ 3
verbose ........................................ 4

W
width ........................................... 52
within .......................................... 20
within (parameter) ......................... 11

X
xparse (package) ........................... 2, 57
\xprintexercise ......................... 21
\xprintsolution .......................... 28
\XSIMatbegindocument .................. 39
\XSIMatenddocument ..................... 39
\XSIMexpandcode ......................... 39, 46, 77
\XSIMfilewritestart ..................... 57 f.
\XSIMfilewritestop ....................... 57 f.
\XSIMgobblechars ......................... 57 f.
\XSIMifblankTF ............................. 39
\XSIMifblankF ............................... 45, 47
\XSIMifblankT ............................... 46 f.
\XSIMifblankTF ............................. 46 f.
\XSIMifchapterTF ......................... 39
\XSIMifeqTF ................................ 39
\XSIMifeqF ................................ 47
\XSIMifeqT ................................ 45, 47
\XSIMmixedcase ......................... 39, 41, 43-47, 55, 62 f., 65 f.,
70 f., 77
\XSIMprint .................................. 38 f.
\XSIMputright .............................. 39, 45 ff.
\XSIMsetfilebegin ......................... 57, 79
\XSIMsetfilebeginX ....................... 57 f.
\XSIMsetfileend ............................ 57
\XSIMsetfileendX ......................... 57 f.
\simsetup ................................. 5, 7, 12, 18, 24, 26, 29 f., 40, 52, 55,
58, 63 f., 74 f., 79, 81
\xsimstyle .................................. 40 f.
\XSIMtranslate ......................... 10, 17, 39, 42, 44-48, 55, 68
xsimverb (package) ...................... 57
\XSIMxprint ............................... 39