Table of Contents

1. Licence, Requirements and README .................................................. 2
2. Motivation and Background ................................................................. 3
3. How to Read the Manual ..................................................................... 3
   3.1. Nomenclature ............................................................................. 3
   3.2. Package Options ........................................................................ 4
   3.3. Setting Options ......................................................................... 4
   3.4. Command descriptions .............................................................. 5
4. Exercises and Solutions ....................................................................... 5
5. How the Exercise Environments Work ................................................ 6
6. New Exercise Types ............................................................................. 8
7. Exercise Properties ............................................................................. 10
   7.1. Predefined Properties ............................................................... 10
   7.2. Declaring Own Properties ......................................................... 12
   7.3. Exercise Goals ......................................................................... 14
   7.4. Exercise Tags ........................................................................... 16
8. Using and Printing an Exercise ......................................................... 17
   8.1. What the Environments do ......................................................... 17
   8.2. Environment Options & Hooks .................................................... 18
9. Collecting Exercises .......................................................................... 20
   9.1. Background ............................................................................. 20
   9.2. Usage ...................................................................................... 20
10. Printing Random Exercises From a Collection .................................... 24
11. Printing Solutions ............................................................................. 25
12. Grading Tables .................................................................................. 27
13. Styling the Exercises – Templates .................................................... 28
   13.1. Background ............................................................................ 28
   13.2. Templates Provided by the Package ......................................... 29
   13.3. Commands for Usage in Template Definitions .......................... 30
      13.3.1. Goals .............................................................................. 30
      13.3.2. Properties ....................................................................... 31
      13.3.3. Parameters ...................................................................... 32
      13.3.4. Tags ............................................................................... 33
      13.3.5. Further Commands for Usage in Template Definitions .......... 33
14. Using and Printing Random Exercises From a Collection .................. 33
15. Printing Solutions From a Collection ................................................ 34
16. Grading Tables .................................................................................. 36
17. Styling the Exercises – Templates .................................................... 37
18. Using and Printing Random Exercises From a Collection .................. 39
19. Printing Solutions From a Collection ................................................ 40
20. Grading Tables .................................................................................. 42
21. Styling the Exercises – Templates .................................................... 43
22. Using and Printing Random Exercises From a Collection .................. 45
23. Printing Solutions From a Collection ................................................ 46
24. Grading Tables .................................................................................. 48
25. Styling the Exercises – Templates .................................................... 49
26. Using and Printing Random Exercises From a Collection .................. 51
27. Printing Solutions From a Collection ................................................ 52
28. Grading Tables .................................................................................. 54
29. Styling the Exercises – Templates .................................................... 55
30. Using and Printing Random Exercises From a Collection .................. 57
31. Printing Solutions From a Collection ................................................ 58
32. Grading Tables .................................................................................. 60
33. Styling the Exercises – Templates .................................................... 61
34. Using and Printing Random Exercises From a Collection .................. 63
35. Printing Solutions From a Collection ................................................ 64
36. Grading Tables .................................................................................. 66
37. Styling the Exercises – Templates .................................................... 67
38. Using and Printing Random Exercises From a Collection .................. 69
39. Printing Solutions From a Collection ................................................ 70
40. Grading Tables .................................................................................. 72
41. Styling the Exercises – Templates .................................................... 73
42. Using and Printing Random Exercises From a Collection .................. 75
43. Printing Solutions From a Collection ................................................ 76
44. Grading Tables .................................................................................. 78
45. Styling the Exercises – Templates .................................................... 79
46. Using and Printing Random Exercises From a Collection .................. 81
47. Printing Solutions From a Collection ................................................ 82
48. Grading Tables .................................................................................. 84
49. Styling the Exercises – Templates .................................................... 85
50. Using and Printing Random Exercises From a Collection .................. 87
51. Printing Solutions From a Collection ................................................ 88
52. Grading Tables .................................................................................. 90
Permission is granted to copy, distribute and/or modify this software under the terms of the \LaTeX\ Project Public License (\LPPL), version 1.3 or later (http://www.latex-project.org/lppl.txt). The software has the status "maintained." \textsc{xsim} loads the packages expl3 [L3Pa], xparse [L3Pb], etoolbox [Leh15], array [MC08] booktabs [Fea05] and translations [Nie15]. 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 \textsc{xsim} you should be using an up to date \TeX\ distribution, anyway.
2. Motivation and Background

It has been quite a while since I first published exsheets [Nie17] 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 (119 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 exsheets is implemented right now. As a consequence I wanted to re-implement exsheets for a long time. This is what lead to XSSIM. Internally the package works completely different.

XSIM will be the official successor of exsheets which is now considered obsolete but will stay alive and will still receive bugfix releases. However, new features will not be added to 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.

collection A 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.

goal Goals are a certain type of properties with a numerical value the sum of which is available throughout the document.

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.
3. How to Read the Manual

**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 auxfile on the next run and only if the number of exercises stays stable between compilations the needed information will be written to the auxfile. *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 6 and B.2 on page 48.

Those options are used the usual way as package option

\usepackage[verbose]{xsim}

or as global option

\documentclass[verbose]{article}

or via the setup command:

\xsimsetup{(options)}

Set up *xsim*’s package options and all other options described at other places in the manual.

### 3.3. Setting Options

Apart from the package options already described in section 3.2 *xsim* has further options. Those can be “toplevel” options or options belonging to a module.
4. Exercises and Solutions

toplevel = {{value}}
A toplevel option.

module/sublevel = {{value}} A sublevel option belonging to the module module

Both kinds of options are set with \xsimsetup:

\xsimsetup{
  toplevel = {value},
  module/sublevel = {value}
}

3.4. Command descriptions

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

Some command name descriptions end with TF.

\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:

\SomeCommandT{arguments}{true}
The same as \SomeCommandTF but only with the (true) argument and no (false) argument.

\SomeCommandF{arguments}{false}
The same as \SomeCommandTF but only with the (false) argument and no (true) argument.

4. Exercises and Solutions

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

\begin{exercise}{properties}
Input and typeset an exercise. See section 7 on page 10 for details on exercise properties.

\begin{solution}{options}
Input and typeset the solution to the exercise of the previous exercise environment. See
section 11 on page 25 for details on options of solutions.
5. How the Exercise Environments Work

Both the exercise and the solution environments write the contents of their bodies verbatim to external files following a certain naming structure:

- \((\text{jobname})-(\text{type})-(\text{id})-\text{exercise|solution}\)-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

- `xsim_manual-exercise-1-exercise-body.tex` and
- `xsim_manual-exercise-1-solution-body.tex`, respectively.

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 \textit{type} of an exercise will be given in section 6 on the following page. The \textit{id} 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 generated:

```
\[
\begin{verbatim}
\% file `xsim_manual-exercise-1-exercise-body.tex'
\% in folder `exercises/'
\% exercise of type `exercise' with id `1'
\% generated by the `exercise' environment of the
\% `xsim' package v0.11 (2018/02/12)
\% from source `xsim_manual' on 2018/02/14 on line 1
\%
\end{verbatim}
\[
A first example for an exercise.
```

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 \texttt{xsim} offers the following option:

\texttt{path = \{(path name)\}} \hspace{1cm} (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 \textit{path must exist on your system before you can use it!} This document uses \texttt{path = \{exercises\}}.

\texttt{file-extension = \{(string)\}} \hspace{1cm} Default: \texttt{tex}

This option let's you choose the extension of the auxiliary files.

Another thing to keep in mind: the environment in many ways works the same way as the \texttt{filecontents} environment. \textit{This also means that you cannot have comments on the first line of the environments:}

\footnote{1. 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.}
6. New Exercise Types

\begin{exercise}[points=2] % this comment will cause trouble
Lorem ipsum
\end{exercise}

\texttt{x.s.i.m} writes a lot of stuff to the auxfile for re-using information on subsequent compilations. If you add exercises, change properties etc. it might happen that wrong information is staying in the auxfile and is wrongly used by \texttt{x.s.i.m}. In such cases deleting the auxfile 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 \texttt{clear-aux} option might help to reduce erroneous exercises.

A lot of the lines \texttt{x.s.i.m} writes to the auxfile and reads in a subsequent run look like this:

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

As you can see different entries of the various properties of exercises are separated with a |. This means that you cannot use this symbol inside properties. This is why \texttt{x.s.i.m} provides an option to change this marker.

\texttt{split-aux-lists} = \{\langle string \rangle\}  \hspace{1cm} \text{Default: |}

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

6. New Exercise Types

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

\texttt{\DeclareExerciseType{\langle type \rangle}{\langle parameters \rangle}}

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

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

\texttt{\numberof\langle exercise-env \rangle s}

These commands hold the absolut number of used exercises of type \langle type \rangle. The meaning of \langle exercise-env \rangle will become clear below when the exercise parameters are explained. It is always the same as the exercise environment name.

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

There are \numberofexercises\ exercises and \numberofproblems\ problem in this manual.

There are 12 exercises and 1 problem in this manual.

\texttt{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} ,
  solution-name = \XSIMtranslate{solution} ,
  exercise-template = default ,
  solution-template = default
}
\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{exercise-env} = \{⟨exercise environment name⟩\}

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

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

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

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

The name of the exercises of type \langle \text{type} \rangle – used for typesetting. \textit{This parameter is mandatory.}

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

The name of the solutions of type \langle \text{type} \rangle – used for typesetting. \textit{This parameter is mandatory.}

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

The template used for typesetting the exercises of type \langle \text{type} \rangle. \textit{This parameter is mandatory.} See section 13 on page 28 for details on templates.

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

The template used for typesetting the exercises of type \langle \text{type} \rangle. \textit{This parameter is mandatory.} See section 13 on page 28 for details on templates.

\texttt{counter} = \{⟨counter name⟩\}

The counter used for the exercises of type \langle \text{type} \rangle. If not explicitly set the counter with the same name as \texttt{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.

\[\text{solution-counter} = \{\langle \text{counter name} \rangle\}\]

The counter used for the solutions of type \(\langle \text{type} \rangle\). If not explicitly set the counter with the same name as \text{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{...}.

\[\text{number} = \{\langle \text{integer} \rangle\}\]

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.

It is possible to change some of the parameters after an exercise type has been defined. Those include \text{exercise-name}, \text{solution-name}, \text{exercise-template}, and \text{solution-template}:

\[\\SetExerciseParameter{\langle \text{type} \rangle}{\langle \text{parameter} \rangle}{\langle \text{value} \rangle}\]

Usable to set a single parameter to a new value.

\[\\SetExerciseParameters{\langle \text{type} \rangle}{\langle \text{parameters} \rangle}\]

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

If you try to set an already set but fixed parameter like \text{exercise-env} a warning will be written to the log file. For all parameters that can be changed also options exist wich can be set via \texttt{\xsimsetup}. They are explained in section \ref{sec:options} on page \pageref{sec:options}.

All exercises of a type use the parameters (e.g., \texttt{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.

\section{Exercise Properties}

\subsection{Predefined Properties}

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

\[\text{id} = \{\langle \text{integer} \rangle\}\]

Holds the internal id of an exercise. *Cannot be set by the user.*

\[\text{ID} = \{\langle \text{text} \rangle\}\]

Holds the user id of an exercise if defined. Otherwise it is equal to \text{id}.
7. Exercise Properties

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

subtitle = {{text}}
Holds the subtitle of an exercise.

points = {{number}}
Holds the reachable points of an exercise.

bonus-points = {{number}}
Holds the reachable bonus-points of an exercise.

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

print! = true|false
Holds a special print boolean of an exercise, see page 17.

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

use! = true|false
Holds a special usage boolean of an exercise, see page 17.

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 20).

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 \arabic{page}).

section = {{text}}
Holds the section counter value representation of an exercise (i.e., what you usually know as \thesection).
7. Exercise Properties

**section-value** = {\langle integer \rangle}

Holds the section counter value of an exercise (i.e., what you usually know as \texttt{\arabic{section}}).

**chapter** = {\langle text \rangle}

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

**chapter-value** = {\langle integer \rangle}

Holds the chapter counter value of an exercise (i.e., what you usually know as \texttt{\arabic{chapter}}). Only if a command \texttt{\chapter} and a counter \texttt{chapter} exist.

**sectioning** = {\langle section numbers \rangle}

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

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}

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

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

7.2. Declaring Own Properties

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

\texttt{\DeclareExerciseProperty!*-\{property\}}

Declares the property \texttt{\{property\}}.

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

If used with the optional \texttt{*} a 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{!}, i.e., if both are present the property is boolean 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.
7. Exercise Properties

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

This is better demonstrated with an example:

\begin{exercise}
\lipsum[4]
\verb+\GetExerciseProperty{id}+: \GetExerciseProperty{id} \par
\verb+\GetExerciseAliasProperty{ID}+: \GetExerciseAliasProperty{ID} \par
\verb+\GetExerciseProperty{ID}+: \GetExerciseProperty{ID}
\end{exercise}

\begin{exercise}[ID=foo-bar]
\lipsum[4]
\verb+\GetExerciseProperty{id}+: \GetExerciseProperty{id} \par
\verb+\GetExerciseAliasProperty{ID}+: \GetExerciseAliasProperty{ID} \par
\verb+\GetExerciseProperty{ID}+: foo-bar
\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 28 about
7. Exercise Properties

templates.

7.3. A Special Kind of Property: Exercise Goals

Exercise goals are a generic concept in \texttt{xsim} for exercise properties like \texttt{points} or \texttt{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.

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

\TotalExerciseGoals{⟨list of goals⟩}{⟨singular⟩}{⟨plural⟩}
Get the sum of goal all goals in ⟨list of goals⟩ for all exercises. 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.

\AddtoExerciseTypeGoal{⟨type⟩}{⟨goal⟩}{⟨value⟩}
Adds ⟨value⟩ to the goal ⟨goal⟩ of exercise type ⟨type⟩.

\AddtoExerciseTypeGoalPrint{⟨type⟩}{⟨goal⟩}{⟨value⟩}{⟨singular⟩}{⟨plural⟩}
Adds ⟨value⟩ to the goal ⟨goal⟩ of exercise type ⟨type⟩. The value and – depending on whether the value equals 1 or not – ⟨singular⟩ or ⟨plural⟩ are left in the input stream.

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

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

\ExerciseGoalValuePrint{⟨value⟩}{⟨singular⟩}{⟨plural⟩}
Print ⟨value⟩ and – depending on whether the value equals 1 or not – ⟨singular⟩ or ⟨plural⟩.

\endinput
7. Exercise Properties

\printgoal\{\langle value\rangle\}
Print \langle value\rangle according to option \texttt{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.

\printbonus\{\langle type\rangle\}
Print the sum of bonus 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.

\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*\{\langle value\rangle\}
Adds \langle value\rangle 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{itemize}
  \item \DeclareExerciseGoal\{points\}
  \item \DeclareExerciseGoal\{bonus-points\}
\end{itemize}

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

\texttt{goal-print} = \{(\langle code\rangle)\}
Default: \#1

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

\textsuperscript{2} See section 14 on page 43 for details on the definition and usage of language dependent words.
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:

```latex
\NewDocumentCommand{\printsumofpointsandbonus}{}{%
  \TotalExerciseGoals{points+bonus-points}
  \\XSIMtranslate{point}{}
  \\XSIMtranslate{points}{}
}%
```

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

```latex
\NewDocumentCommand{\pointsandbonus}{}{%
  \TotalExerciseGoal{points}{}
  \IfExerciseGoalsSumF{bonus-points}{=0}{%\,+\,\TotalExerciseGoal{bonus-points}{}
  \\XSIMtranslate{points}{}
}%
```

### 7.4. A Special Kind of Property: Exercise Tags

Exercise tags are a generic concept in \texttt{xsim} for exercise properties like \texttt{tags} or \texttt{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 next page for details on usage of exercises and the difference to printing an exercise and how to use exercise tags for selection.

\texttt{\DeclareExerciseTagging{⟨tag⟩}}

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

\texttt{\ProvideExerciseTagging{⟨tag⟩}}

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

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

```latex
\DeclareExerciseTagging{tags}
\DeclareExerciseTagging{topics}
\xsimsetup{tags/ignore-untagged=false}
```

This means that these options are available:

\texttt{tags} = \texttt{⟨csv list of tags⟩}

Choose the set of tags whose associated exercises should be printed.
Choose the set of topics whose associated exercises should be printed.

\texttt{ignore-untagged = true|false} \hspace{2cm} \text{Default: false}

If set to true exercises with no tags will be printed even if tags have been chosen with the option \texttt{tags}.

\texttt{ignore-untagged = true|false} \hspace{2cm} \text{Default: true}

If set to true exercises with no topics will be printed even if topics have been chosen with the option \texttt{topics}.

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:

\texttt{print! = true|false}

If set to true the exercise will be printed (and thus used) regardless of other conditions.

\texttt{use! = true|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 \texttt{\begin{exercise}} (or other environments defined through \texttt{\DeclareExerciseType}) then different things happen depending on different settings:

- If the \texttt{insert mode} is active nothing happens, see section 9 on page 20 for details on this.
- Else the id integer is incremented.
- If the exercise is \texttt{used} the corresponding counter is stepped and the exercise is added to the “use list”. The properties \texttt{counter} and \texttt{use} are updated accordingly.
- If an exercise is \texttt{printed} then it is also \texttt{used}. An exercise that isn’t used cannot be printed. Being printed means two things: being added to the “print list” and being typeset at the position where the exercise is placed in the source file. If an exercise is \texttt{not printed but used} it means that the counter will be stepped. This can be useful for creating an exercise sheet only containing the solutions for some exercises.
- If an exercise is printed certain hooks and template code is inserted around the environment body.

1. \texttt{\begin{exercise}[print=false]}
2. This exercise will not be printed but the exercise counter will be stepped.

\begin{exercise}[print=false]
8. Using and Printing an Exercise

```
\begin{environment}
\begin{templateCode}
\begin{hook}
\begin{environmentBody}
\end{hook}
\end{templateCode}
\end{environment}
\end{templateCode}
\begin{hook}
\end{templateCode}
```

**Figure 1:** Schematic structure of an exercise or solution.

```
\begin{exercise}
\begin{solution}
\end{solution}
\end{exercise}
\end{solution}
```

The schematic structure of an exercise is shown in figure 1.

### 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/within** = {⟨counter⟩}
  - (initially empty)
  - Adds the exercise counter to the reset list of the counter ⟨counter⟩. 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 \( \text{\textbackslash the(counter)} \).
8. Using and Printing an Exercise

**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)}}.

**solution/name** = {{(name)}
    An interface for \SetExerciseParameter{exercise}{solution-name}{{(name)}}.

**exercise/pre-hook** = {{(code)}
    The code for the pre exercise hook for exercises of the type “exercise”. (initially empty)

**exercise/begin-hook** = {{(code)}
    The code for the begin exercise hook for exercises of the type “exercise”. (initially empty)

**exercise/end-hook** = {{(code)}
    The code for the end exercise hook for exercises of the type “exercise”. (initially empty)

**exercise/post-hook** = {{(code)}
    The code for the post exercise hook for exercises of the type “exercise”. (initially empty)

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

**solution/pre-hook** = {{(code)}
    The code for the pre solution hook for solutions of the type “exercise”. (initially empty)

**solution/begin-hook** = {{(code)}
    The code for the begin solution hook for solutions of the type “exercise”. (initially empty)

**solution/end-hook** = {{(code)}
    The code for the end solution hook for solutions of the type “exercise”. (initially empty)

**solution/post-hook** = {{(code)}
    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{(type)}{(id)}
    Inserts the exercise of type (type) with the id (id).


9. Collecting Exercises

9.1. Background

xsim knows the concept of “exercise collections”. A collection of exercises can be useful when you want to print a certain group of exercises several times. Each collection must have a unique name with which you can refer to the corresponding collection. A collection is realized by declaring the collection and by surrounding the exercises belonging to the collection with a certain pair of commands (this is explained in the next section).

Let’s say you have several files of math exercises where one only contains geometry exercises and another only calculus exercises and so on. Surrounding the \input of each file with said pair of commands for a certain collection all exercises of the corresponding file now are a collection which then can be printed at once wherever you want the collection of exercises to be printed. By choosing certain tags (see section 7.4 on page 16) inside each collection you could even cherry-pick exercises from the external file.

9.2. Usage

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.

\DeclareExerciseCollection{(collection name)}
Define a new collection (collection name) in the document preamble.

\collectexercisestype{(collection name)}{(exercise type)}
Opens the collection (collection name) which now collects all exercises of type (exercise type) until the collection is closed with \collectexercisesstop. Collections of other types are not collected.

\collectexercises{(collection name)}
Opens the collection (collection name) which now collects all exercises until the collection is closed with \collectexercisesstop.

---

3. This command starts a group with \begingroup!
4. This command starts a group with \begingroup!
9. Collecting Exercises

\collectexercisesstop\{collection name\}
Closes the collection \{collection name\}.

\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:

\texttt{\printcollection/headings} = \texttt{true}|false \hspace{2cm} \text{Default: false}
If true a heading for each exercise type is inserted.

\texttt{\printcollection/headings-template} = \{\texttt{\template}\} \hspace{2cm} \text{Default: collection}
The heading template used when \texttt{headings} = \{true\}.

\texttt{\printcollection/print} = \texttt{exercises}|\texttt{solutions}|\texttt{both} \hspace{2cm} \text{Default: exercises}
Determines whether \texttt{\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{\xsimsetup} using the module \texttt{print-collection}.

\begin{itemize}
  \item 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}.
\end{itemize}

The usage should be clear:

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

\footnote{This command ends a group with \texttt{\endgroup}!}
Once the collection is closed it can be printed:

```
\printcollection{foo}
```

**Exercise 6**

This exercise is added to the collection ‘foo’.

**Exercise 7**

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

**Exercise 8**

So is this.

**Exercise 9**

As well as this one.

You can open several collections at the same time:

```
\collectexercises{foo}
\collectexercisestype{bar}{exercises}
\collectexercisesstop{bar}
\collectexercisesstop{foo}
```

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

```
\printcollection{all exercises}
```

all exercises will be printed.
9. Collecting Exercises

If you use \texttt{\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 \texttt{\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:

\begin{verbatim}
% preamble:
\\% \DeclareExerciseCollection{foo-easy}
\\% \DeclareExerciseCollection{foo-medium}
\\% \DeclareExerciseTagging{difficulty}
\%
\% document:
\\% \collectexercises{foo-easy}
\\% \xsimsetup{difficulty=easy}
\\% \input{foo.tex}
\\% \collectexercisestop{foo-easy}
\% collection `foo-easy' now contains all exercises of file `foo.tex' tagged with `difficulty=easy'
\\% \collectexercises{foo-medium}
\\% \xsimsetup{difficulty=medium}
\\% \input{foo.tex}
\\% \collectexercisestop{foo-medium}
\% collection `foo-medium' now contains all exercises of file `foo.tex' tagged with `difficulty=medium'
\end{verbatim}

The recommended usage is similar to the last example. Actually a collection can be printed \textit{before} it is opened, too. (This needs at least two compilations, though.) However, it is safer printing a collection only once and only \textit{after it has been collected}. No guaranties are given that properties are set correctly if you use the collection before. You usually also will make sure that the exercises in a collection are unique, \textit{i.e.}, that an exercise is not part of several collections – at least not if both collections are printed in the same document.
10. Printing Random Exercises From a Collection

XSIM provides the possibility of selecting random exercises from a collection (cf. section 9 on page 20).

Please be aware that this feature is not available in XeLaTeX!

\printrandomexercises\[\langle\text{options}\rangle\]{\langle\text{number}\rangle}

This command prints \langle\text{number}\rangle random exercises from the collection chosen with option \textbf{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:

\begin{itemize}
  \item \textbf{sort} = true|false \textit{Default: true}
      \begin{itemize}
        \item Determines whether the random chosen exercises should be sorted according to their order of definition in the collection or not.
      \end{itemize}
  \item \textbf{collection} = \{\langle\text{collection}\rangle\} \textit{Default: all exercises}
      \begin{itemize}
        \item The collection from which the exercises are to be chosen from.
      \end{itemize}
  \item \textbf{exclude} = \{\langle\text{csv list of ids}\rangle\}
      \begin{itemize}
        \item A list of ids or IDs of exercises not to be chosen.
      \end{itemize}
  \item \textbf{print} = exercises|solutions|both \textit{Default: exercises}
      \begin{itemize}
        \item Determines whether \texttt{printrandomexercises} prints the exercises or the solutions. When you choose both exercises and solutions are printed alternately.
      \end{itemize}
\end{itemize}

```
\printrandomexercises[\text{collection=}foo]{2}
```

\textbf{Exercise 7}

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

\textbf{Exercise 9}

As well as this one.

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:

\printsolutionstype*\{\langle options\rangle\}\{\langle exercise type\rangle\}
  Prints the solutions of all used exercises of type \langle exercise type\rangle. The starred version only prints the solutions of all printed exercises of type \langle exercise type\rangle.

\printsolutions*\{\langle options\rangle\}
  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*\{\langle options\rangle\}
  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\{\langle options\rangle\}\{\langle type\rangle\}\{\langle id\rangle\}
  Prints the solution of the exercise of type \langle type\rangle with the id \langle id\rangle.

---

\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 11**
Try to fill in these blanks. All of them are created by using the \texttt{\blank} command.

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

**print-solutions/headings** = \texttt{true}|\texttt{false} \quad \text{Default: true}
If true a heading for each exercise type is inserted.

**print-solutions/headings-template** = \texttt{\{\langle template\rangle\}} \quad \text{Default: default}
The heading template used when headings = \texttt{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.
11. Printing Solutions

\textbf{print-solutions/section} = \textbf{true}|false|(integer) \quad \text{Default: false}

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.

\textbf{print-solutions/chapter} = \textbf{true}|false|(integer) \quad \text{Default: false}

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.

\textbf{print-solutions/collection} = false|\langle\text{collection name}\rangle \quad \text{Default: false}

If used only solutions of exercises belonging to collection \langle collection name \rangle 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 \texttt{\chapter} or \texttt{\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 \texttt{xsimsetup} using the module \texttt{print-solutions}.

```
\printsolutions[section=4,headings-template=per-section]
```

\section*{Solutions to the Exercises of Section 4}

\section*{Solution 1}

A first example for a solution.

```
\printsolution{exercise}{5}
```

\section*{Solution 5}

The solution of the exercise that has not been printed.
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 14 about exercise goals) but also to be able to output a grading table. **XSIM** has built-in means for this.

\texttt{\textbackslash gradingtable[(options)\]}

Print a grading table.

Valid options for this command are

\texttt{template = \{\textit{template}\} }\quad \text{Default: default}

Choose the template used for the grading table.

\texttt{type = \{\textit{exercise type}\} }\quad \text{Initially empty}

Choose the exercise type for which the table is printed.

Both option defaults can be changed with \texttt{\textbackslash simsetup} setting the options using \texttt{grading-table}:

\begin{verbatim}
\texttt{\textbackslash simsetup{\textbackslash grading-table/template = default*}}
\end{verbatim}

An example:

\begin{verbatim}
\texttt{\textbackslash gradingtable[type=exercise]}
\end{verbatim}

\begin{tabular}{ll}
\hline
Exercise & Points reached \\
\hline
1 & 0 \\
2 & 4 \\
3 & 0 \\
4 & 0 \\
5 & 0 \\
6 & 0 \\
7 & 0 \\
8 & 0 \\
9 & 0 \\
10 & 2.5 \\
11 & 0 \\
12 & 0 \\
\hline
total & 6.5 \\
\hline
\end{tabular}

Or using the “default*” template:
Available templates and how to define new ones are explained in sections 13.4.3 on page 36 and 13.5 on page 36. \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:

\begin{gradingtable}
\begin{tabular}{rrrrrrrrrrr}
Exercise & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & total \\
Points & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 2.5 & 0 & 0 & 6.5 & reached \\
\end{tabular}
\end{gradingtable}

\textbf{13. Styling the Exercises – Templates}

\subsection*{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:

\begin{itemize}
\item environment templates (see section 13.4.1 on page 35),
\end{itemize}
13. Styling the Exercises – Templates

- heading templates (see section 13.4.2 on page 36) and
- grading table templates (see section 13.4.3 on page 36)

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 13.5 on page 36. 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 13.5 on page 36 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:

\begin{description}
\item[default] The template activated per default and the only one available without further action.
\item[runin] A layout rather similar to the one by package exsheets, see section 13.5.3 on page 38. Available through the style file \texttt{layouts} (see section B.7 on page 50 for more information on style files).
\item[margin] A layout rather similar to the one by package exsheets, see section 13.5.4. Available through the style file \texttt{layouts} (see section B.7 on page 50 for more information on style files).
\end{description}

\textbf{Layout “default”}

\textbf{Exercise 10 The Subtitle}

13. Styling the Exercises – Templates

Layout "runin"


Layout "margin"


13.3. Commands for Usage in Template Definitions

13.3.1. Goals

\IfExerciseGoalTF{goal}{relation and value}{true}{false}
Checks the sum of goal \(\text{goal}\) against \(\text{relation and value}\).

\IfExerciseGoalSingularTF{goal}{true}{false}
Checks if the value of the goal \(\text{goal}\) 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 \(\text{list of goals}\) for the exercises of type \(\text{type}\) against \(\text{relation and value}\).

\IfExerciseGoalsSumTF{type}{list of goals}{relation and value}{true}{false}
Checks the sum of all goals in \(\text{list of goals}\) for all exercises of all types against \(\text{relation and value}\).

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

\TotalExerciseGoal{goal}{singular}{plural}
Print the sum of goal \(\text{goal}\) for all exercises of all types and append \(\text{plural}\) or \(\text{plural}\) depending on whether the sum equals 1 or not.
13. Styling the Exercises – Templates

13.3.2. Properties

* \IfExercisePropertyExist\{\langle property \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}
  Tests wether an exercise property with the name \langle property \rangle is defined.

* \IfExercisePropertySet\{\langle property \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}
  Tests wether the exercise property \langle property \rangle has been set for the current exercise.

* \GetExerciseProperty\{\langle property \rangle\}
  Retrieves the value of the property \langle property \rangle for the current exercise.

* \GetExerciseProperty\{\langle property \rangle\}\{\langle true \rangle\}\{\langle false \rangle\}
  Tests wether the exercise property \langle property \rangle has been set for the current exercise. Inside the \langle true \rangle 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\}
  Retrieves the environment body of either the exercise or the corresponding solution of the current exercise.

* \GetExerciseIdForProperty\{\langle property \rangle\}\{\langle value \rangle\}
  Retrieves the property id of the exercise where the property \langle property \rangle has the value \langle value \rangle. This only works for unique properties!

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

* \SetExerciseProperty\{\langle property \rangle\}\{\langle value \rangle\}
  Set the property \langle property \rangle of the current exercise to \langle value \rangle.

* \SetExpandedExerciseProperty\{\langle property \rangle\}\{\langle value \rangle\}
  Expand \langle value \rangle \edef-like and set the property \langle property \rangle of the current exercise to the result of the expansion.

* \ExerciseSetProperty\{\langle type \rangle\}\{\langle id \rangle\}\{\langle property \rangle\}\{\langle value \rangle\}
  Set the property \langle property \rangle of the exercise of type \langle type \rangle and id \langle id \rangle to \langle value \rangle.

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

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

* \GetExerciseAliasProperty\{\langle property \rangle\}
  Retrieves the value of the property of which \langle property \rangle 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 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.

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

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

\IfExerciseParameterSetTF{⟨parameter⟩}{⟨true⟩}{⟨false⟩}
Test if the parameter ⟨parameter⟩ has been set for the current exercise type.

\ExerciseParameterIfSetTF{⟨type⟩}{⟨parameter⟩}{⟨true⟩}{⟨false⟩}
Test if the parameter ⟨parameter⟩ has been set for the exercise type ⟨type⟩.
13. Styling the Exercises – Templates

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.

\IfExerciseTagSet{⟨value⟩}{⟨true⟩}{⟨false⟩}
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.

\IfExerciseTopicSet{⟨value⟩}{⟨true⟩}{⟨false⟩}
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.

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

• \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.
13. Styling the Exercises – Templates

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

*\ForEachPrintedExerciseByID*
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*
Loops over each `used` exercise order by the exercise id. Inside `(code)` you can refer to several properties of the corresponding exercise:
13. Styling the Exercises – Templates

• #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

\XSIMtranslate{(keyword)}
Delivers the translation of ⟨keyword⟩ according to the current document language (in the meaning of a babel [Bra16] or polyglossia [Cha15] language). Existing keywords and keyword translations (and how to add new ones) are explained in section 14 on page 43.

\XSIMexpandcode{(code)}
Expands ⟨code⟩ like \edef does and leaves the result in the input stream.

\XSIMifchapterTF{(true)}{(false)}
Returns ⟨true⟩ if both a macro \chapter and a counter chapter are defined and ⟨false⟩ otherwise.

\XSIMmixedcase{(code)}
Converts the full expansion\(^8\) of ⟨code⟩ to mixed case:
\XSIMmixedcase{this is some text} This is some text
This command expands ⟨code⟩ before converting it.

\XSIMputright{(macro)}{(code)}
Extends the macro definition of ⟨macro⟩ with ⟨code⟩ putting it to the right. This is more or less a local version of the \LaTeX{} kernel macro \g@addto@macro.

\XSIMifeqTF{(code 1)}{(code 2)}{(true)}{(false)}
Checks if the full expansion\(^8\) of ⟨code 1⟩ and ⟨code 2⟩ is the same tokenlist.

\XSIMifblankTF{(true)}{(false)}
Checks if the full expansion\(^8\) of ⟨code⟩ is blank (i.e., if it is empty or only consists of spaces).

13.4. Declaring Templates

13.4.1. Environment Templates

\DeclareExerciseEnvironmentTemplate{(name)}{(begin code)}{(end code)}
Declare the environment template ⟨name⟩.

Environment templates are used by the exercise and solution environments. Those are the templates set with the parameters exercise-template and solution-template.

The predefined template is called “default”, see section 13.5.1 on the following page.

\(^8\) This is a \romannumerals expansion [Flo].
13. Styling the Exercises – Templates

13.4.2. Heading Templates

\DeclareExerciseHeadingTemplate\langle name\rangle\{\langle code\rangle\}

Declare the heading template \langle name\rangle.

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 13.5.5 on page 39.

13.4.3. Grading Table Templates

\DeclareExerciseTableTemplate\langle name\rangle\{\langle code\rangle\}

Declare the grading table template \langle name\rangle.

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 13.5.6 on page 40 and 13.5.7 on page 41.

13.5. Examples

The repository of this package currently includes 35 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 \LaTeX forums, see section F on page 56.

13.5.1. The \texttt{default} Exercise Template

Below the definition of the default exercise template provided by \texttt{XSIM} is shown:

```latex
\begin{verbatim}
\DeclareExerciseEnvironmentTemplate\langle default\rangle{% 
  \subsection* \% 
  \XSIMmixedcase{\GetExerciseName}nobreakspace 
  \GetExerciseProperty{counter}\% 
  \IfInsideSolutionF \% 
    \GetExercisePropertyT{subtitle}\% 
    \{ \normalfont\itshape\PropertyValue\}\% \% 
  \} \% 
  \GetExercisePropertyT{points}\% 
\end{verbatim}
```

13. Styling the Exercises – Templates

\marginpar
{\IfInsideSolutionF{\rule{1.2cm}{1pt}\slash}\
\printgoal{\PropertyValue}\
\GetExercisePropertyT{bonus-points}{-(+\printgoal{\PropertyValue})}\
}\
~\XSIMtranslate{point-abbr}\
}%
\}

13.5 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:

1 \DeclareExerciseEnvironmentTemplate{tcolorbox}
2 {\
3  \tcolorbox[
4   colback = red!5!white ,
5   colframe = red!75!black ,
6   colbacktitle = yellow!50!red ,
7   coltitle = red!25!black ,
8   breakable ,
9   drop shadow ,
10  beforeafter skip = .5\baselineskip ,
11  title =
12  \textbf{\GetExerciseName~\GetExerciseProperty{counter}}\
13  \GetExercisePropertyT{subtitle}{\textit{\PropertyValue}}\
14  \IfInsideSolutionF{\
15   \GetExercisePropertyT{points}{ % notice the space
16   (\printgoal{\PropertyValue}
17   \IfExerciseGoalSingularTF{points}
18     (\XSIMtranslate{point})
19     (\XSIMtranslate{points})%
20   )%
21   )%
22   )%
23  }%
24  }%
25  }
13. Styling the Exercises – Templates

See it in action:

\begin{problem}[subtitle=My subtitle,points=5]
This is a problem using a subtitle and points.
\end{problem}
\begin{answer}
This is the answer to problem~\GetExerciseProperty{counter}.
\end{answer}

13.5.3. Mimicking exsheets’ runin Template

The following example shows how you could mimick exsheets’ runin template. The outcome isn’t exactly the same since exsheets doesn’t use \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{% %
\printgoal{\PropertyValue}}}%
13. Styling the Exercises – Templates

\getexercisepropertyt{bonus-points}{+\printgoal{\propertyvalue}}%
\ifexercisegoalsingulartf{points}{\xsimtranslate{point}}{\xsimtranslate{points}}%
%
%
}
}

13.5.4. Mimicking exsheets’ margin Template

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

\declareexerciseenvironmenttemplate{margin}{
% \trivlist
% \itemllap{% \smash{% \tabular[t]{@{}r@{}}% \textbf{\xsimmixedcase{\getexercisename} \getexerciseproperty{counter}}% \ifexercisepropertysett{points}{% \tabularnewline\% \printgoal{\getexerciseproperty{points}}% \getexercisepropertyt{bonus-points}{+\printgoal{#1}}% \,\xsimtranslate{point-abbr} % \xsimtranslate{points}}% }% \endtabular % \relax % }\relax % }\endtrivlist

13.5.5. The Headings Templates

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

\declareexerciseheadingtemplate{default}{{\section*{\xsimtranslate{default-heading}}}}
\declareexerciseheadingtemplate{collection}{{\section*{\xsimtranslate{collection-heading}}}}
13. Styling the Exercises – Templates

Section 14 on page 43 shows how the translations are defined.

13.5.6. The default Table Template

This template is the one used for grading tables per default. It has a vertical layout.
13. Styling the Exercises – Templates

The part

\noexpand\begin{tabular}{\XSIMifblankTF{\ExerciseType}{l}{c}cc}
\noexpand\ExerciseTableCode
\noexpand\end{tabular}%
%}
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
%
The part

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 [Bra] or polyglossia [Cha] language).

\ForEachExerciseTranslation{(code)}
Loops over all translations of all keywords known to \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 \DeclareExerciseTranslations here is how the translations for exercise have been defined:

```
\DeclareExerciseTranslations{exercise}{
  Fallback = exercise ,
  English = exercise ,
  French = exercice ,
  German = "Ubung
}
```

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>exercice</td>
</tr>
<tr>
<td>exercise</td>
<td>German</td>
<td>&quot;Ubung</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>
</tbody>
</table>

continues
<table>
<thead>
<tr>
<th>keyword</th>
<th>language</th>
<th>translation</th>
</tr>
</thead>
<tbody>
<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>L&quot;osung</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>
<tr>
<td>reached</td>
<td>English</td>
<td>reached</td>
</tr>
<tr>
<td>reached</td>
<td>French</td>
<td>atteint</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>totalement</td>
</tr>
<tr>
<td>total</td>
<td>German</td>
<td>insgesamt</td>
</tr>
</tbody>
</table>

default-heading

| default-heading | Fallback | \XSIMmixedcase {\GetExerciseParameter {solution-name}s} to the \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} |
| default-heading | English  | \XSIMmixedcase {\GetExerciseParameter {solution-name}s} to the \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} |
| default-heading | German   | \XSIMmixedcase {\GetExerciseParameter {solution-name}en} zu den \XSIMmixedcase {\GetExerciseParameter {exercise-name}en} |

collection-heading

| collection-heading | Fallback | \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} |
| collection-heading | English  | \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} |
| collection-heading | German   | \XSIMmixedcase {\GetExerciseParameter {exercise-name}en} |
### 15. Cloze Tests and Blank Lines

Similar to exsheets, `xsim` provides a command `\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` `\blank` will do two things: it will set the linespread according to an option explained below and will insert `\par` after the lines. The starred version doesn’t do these things.

Those are the options for customization:

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

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

<table>
<thead>
<tr>
<th>keyword</th>
<th>language</th>
<th>translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>per-section-heading</td>
<td>Fallback</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solution-name}s} to the \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} of Section\nobreakspace \ExerciseSection</td>
</tr>
<tr>
<td>per-section-heading</td>
<td>English</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solution-name}s} to the \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} of Section\nobreakspace \ExerciseSection</td>
</tr>
<tr>
<td>per-section-heading</td>
<td>German</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solution-name}en} zu den \XSIMmixedcase {\GetExerciseParameter {exercise-name}en} in Abschnitt\nobreakspace \ExerciseSection</td>
</tr>
<tr>
<td>per-chapter-heading</td>
<td>Fallback</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solution-name}s} to the \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} of Chapter\nobreakspace \ExerciseChapter</td>
</tr>
<tr>
<td>per-chapter-heading</td>
<td>English</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solution-name}s} to the \XSIMmixedcase {\GetExerciseParameter {exercise-name}s} of Chapter\nobreakspace \ExerciseChapter</td>
</tr>
<tr>
<td>per-chapter-heading</td>
<td>German</td>
<td>\XSIMmixedcase {\GetExerciseParameter {solution-name}en} zu den \XSIMmixedcase {\GetExerciseParameter {exercise-name}en} in Kapitel\nobreakspace \ExerciseChapter</td>
</tr>
</tbody>
</table>
15. Cloze Tests and Blank Lines

style = {{code}}
Shortcut for setting both blank-style and filled-style at once.

blank/scale = {{decimal number}}
Scales the blank to (decimal number) times its natural width.

blank/width = {{dim}}
Sets the blank to a width of (dim). This takes precedence over scale.

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

blank/line-increment = {{dim}}
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.

blank/line-minimum-length = {{dim}}
The minimal length a line must have before it is built step by step.

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

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

\xsimsetup{blank/filled-style=\textcolor{red}{#1}}

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

This is a _____ outside in normal text.

Exercise 11
Try to fill in _________________ blanks. All of them ________ by using the \blank ________.

Solution 11
Try to fill in these blanks. All of them are created by using the \blank command.

A number of empty lines are easily created by setting the width option:
A. Future Plans

\texttt{xsim} is complete in so far as it is perfectly usable to create exams or exercise and solution sections in books with the most freedom in layout already. But still there are features which would be useful additions. Below I list all ideas that I currently plan to add to \texttt{xsim}:

- 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.

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

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

```latex
\begin{verbatim}
\verbatiminput{verbatim}
\end{verbatim}
```
B. FAQ & How to…

* Exercise properties may have changed. Rerun to get them synchronized.

*************************************************

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

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

\texttt{xsim} writes a lot of stuff to the auxfile for re-using information on subsequent compilations. If you add exercises, change properties etc. it might happen that wrong information is staying in the auxfile and is wrongly used by \texttt{xsim}. In such cases deleting the auxfile 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.

B.3. …Resolve Strange Errors After Updating?

\texttt{xsim} writes a lot of stuff to the auxfile. An update may well change how this is done so deleting the auxfile 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. 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. 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{quote}
\texttt{\textbullet} The reasons are similar to the ones given here: \url{https://tex.stackexchange.com/a/295422/}.
\end{quote}
The usage is now as follows:

\begin{exercise}[bonus]
A bonus question.
\end{exercise}
B. FAQ & How to…

B.7. ...Create and Use \texttt{xsim} Style Files?

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

\begin{verbatim}
xsim.style.math-exam.code.tex
\end{verbatim}

The first command in the file should be \texttt{xsimstyle(math-exam)}. This file can now be loaded into your document using \texttt{\loadxsimstyle(math-exam)}:

\begin{verbatim}
\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}
\end{verbatim}

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
- 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, \textit{i.e.}, \texttt{@} has category code 11 (letter).

The formal description of the commands:

\begin{verbatim}
\xsimstyle*(⟨style name⟩)
\loadxsimstyle{⟨csv list of style names⟩}
\end{verbatim}

The first command in a \texttt{xsim} style file called \texttt{xsim.style.(style name).code.tex} which defines the \texttt{xsim} style \texttt{(style name)}. The starred version activates expl3 syntax.\footnote{11. Those users who want this will know what it means. If you don’t know what it means you will not need it.}

\begin{verbatim}
\loadxsimstyle{⟨csv list of style names⟩}
\end{verbatim}

Load one or more styles into the document.

\begin{verbatim}
At the moment this mechanism offers no advantages over creating a custom package or simply \texttt{\input}ing a file. Future versions might provide additional features.
\end{verbatim}

B.8. ...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.
For this manual we then get the following list.\footnote{12}

### 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 11

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

\footnote{12. Taking care of the fact that we’re in the appendix now which means we can’t use \texttt{value(section)}. Therefore this manual does \texttt{\edef\arabic{section}} right before \texttt{\appendix}}
C. The xsimverb package

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

\texttt{\XSIMfilewritestart*{(file name)}}

Start writing to the file named \textit{(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 \textit{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{\NewDocumentEnvironment}. 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{\newenvironment} scans for optional arguments you’ll otherwise may end up with leading spaces gobbled from the first line in your environment.

\texttt{\XSIMfilewritestop}

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

\texttt{\XSIMsetfilebegin{(code)}}

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.

\texttt{\XSIMsetfileend{(code)}}

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.

\texttt{\XSIMgobblechars{(integer)}}

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.

An example of how to use those commands:

\begin{verbatim}
\documentclass{article}
\usepackage{xsimverb,listings}
\makeatletter
\NewDocumentEnvironment{example}{o}{{\XSIMsetfilebegin{(@percentchar\space file `\jobname.tmp')}\%
\XSIMsetfileend{(@percentchar\space bye bye)\%
\IfNoValueTF{#1}{\XSIMfilewritestart*{\jobname.tmp}}\%
\XSIMfilewritestart{\jobname.tmp}\%
\end{example}
\end{verbatim}
The \tmp file produced by the above example will contain the following three lines (if the file itself was called test.tex):

\begin{verbatim}
% file `test.tmp'
bla bla \LaTeX
% bye bye
\end{verbatim}

\section*{D. All Exercise Examples}

You will notice that some exercises from section 13.5 on page 36 look differently in this section. That is because all exercises of a type use the template that’s \textit{currently active}. If you want exercises with a different look you should use different exercises types.

The following list is created with this code:

\begin{verbatim}
\xsimsetup{exercise/template = bonus}
\printcollection[headings]{all exercises}
\end{verbatim}

\section*{Exercises}

\subsection*{Exercise 1}

A first example for an exercise.
D. All Exercise Examples

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 6
This exercise is added to the collection ‘foo’.

Exercise 7
This exercise is also added to the collection ‘foo’.

Exercise 8
So is this.
E. All Solution Examples

**Exercise 9**
As well as this one.

**Exercise 10 The Subtitle**

**Exercise 11**
Try to fill in ______________ blanks. All of them ________ by using the \texttt{\textbackslash blank} command.

★ **Bonus Exercise 12**
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 11**
Try to fill in these blanks. All of them are created by using the \texttt{\textbackslash 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 currently includes 35 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.

---

### Example 1 “blanks” [\TeX{}][PDF]

```latex
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{xsim,lipsum,xcolor}
\xsimsetup{
  solution/print = true ,
  blank/filled-style = \underline{\textcolor{red}{#1}}
}
```

---

### Example 2 “boxed-headings” [\TeX{}][PDF]

```latex
\documentclass{article}
\usepackage{tcolorbox}
\usepackage{xsim}
\usepackage{needspace}
\DeclareExerciseEnvironmentTemplate{custom}{\Needspace*{5\baselineskip}
\begin{tcolorbox}}
\begin{tcolorbox}
```

---

F. Example Documents Coming With This Package

Example 3 “code-and-output” [TeX] [PDF]

\documentclass{article}
\usepackage{xsimverb, listings, xcolor}
\lstdefinestyle{mystyle}{
  language = [All]TeX ,
  basicstyle = \ttfamily ,
  columns = fullflexible ,
  commentstyle = \color{gray!70} ,
  keywordstyle = \color{red!70!black} 
}

% file 'code-and-output.tmp'
bla bla \LaTeX
% bye bye
bla bla LATEX
blubber \LaTeX

Example 4 “collections” [TeX] [PDF]

\documentclass{article}
\usepackage{xsim, lipsum, hyperref}
\DeclareExerciseCollection{foo-easy}
\DeclareExerciseCollection{foo-medium}
\DeclareExerciseTagging{difficulty}
\usepackage{filecontents, lipsum}
\begin{filecontents*}{foo.tex}
\begin{exercise}[difficulty=easy,points=1]
Exercise 1
outside before
1 Easy
Exercise 2 /1 p.
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. Integer tempus convallis augue. Etiam
facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet,
enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus
eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis
tortor vitae risus porta vehicula.

Exercise 3 /1 p.
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. Integer tempus convallis augue. Etiam
facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet,
enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus
eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis
tortor vitae risus porta vehicula.

2 Medium
Exercise 4 /1 p.
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. Integer tempus convallis augue. Etiam
facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet,
enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus
eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis
tortor vitae risus porta vehicula.

Solutions to the Exercises
Solution 4 /1 p.
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. Integer tempus convallis augue. Etiam

Example 5 “crossref” [TeX] [PDF]

\documentclass{article}
\usepackage{xsim, lipsum, hyperref}
\DeclareExerciseHeadingTemplate{custom}
\ DeclareExerciseEnvironmentTemplate{custom}
\ IfInsideSolutionTF
\ \{\label{sol:\ExerciseID}\}

1 Exercises
Exercise 1
Solution on page 2.
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. Integer tempus convallis augue. Etiam facilisis.
Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim.
Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae
risus porta vehicula.
F. Example Documents Coming With This Package

Example 6 "description-list" [\TeX] [PDF]
\begin{verbatim}
documentclass{article}
usepackage{xsim,lipsum}

\DeclareExerciseEnvironmentTemplate{item}
{\item[\XSIMmixedcase{\GetExerciseName}~\GetExerciseProperty{counter}]}
\xsimsetup{
  exercise/template=item,
  solution/template=item,
  print-solutions/headings-template=none
}
\end{verbatim}

Example 7 "different-point-types" [\TeX] [PDF]
\begin{verbatim}
documentclass{article}
usepackage{xsim}

\DeclareExerciseGoal{A}
\DeclareExerciseGoal{C}
\DeclareExerciseGoal{E}

\newcommand*{\printA}{\TotalExerciseGoal{A}{~A~point}{~A~points}}
\newcommand*{\printC}{\TotalExerciseGoal{C}{~C~point}{~C~points}}
\newcommand*{\printE}{\TotalExerciseGoal{E}{~E~point}{~E~points}}
\end{verbatim}

Example 8 "difficulties" [\TeX] [PDF]
\begin{verbatim}
documentclass{article}
usepackage{clear-aux}{xsim}

\DeclareExerciseTagging{difficulty}
\xsimsetup{
  difficulty={easy,hard}
}
\DeclareExerciseEnvironmentTemplate{custom}
\end{verbatim}

1 Exercises
Exercise 1 (easy)
An easy question.
Exercise 2 (hard)
Now let's see if you can solve this one.
1. Differentiate \( y = 3x^2 + 5x + 3 \). (1/0/0)
2. Find the equation of the tangent line to the function \( y = \frac{x}{2} \) at \( x = 2 \). (2/1/0)
3. Prove that the derivative of a constant is zero. (0/1/2)

1. \( \frac{dy}{dx} = 6x + 5 \)
2. \( y = \frac{1}{2}(x+2)^2 \)
3. \( y' = 0 \) for \( y = C \), where \( C \) is a constant.
F. Example Documents Coming With This Package

Example 9 “hints” [TeX] [PDF]

```latex
\documentclass{article}
\usepackage{xsim}
\usepackage{needspace}
\DeclareExerciseProperty{hint}
% we'll use a description list for the hints:
\newcommand\printhints{
1 Problems
Exercise 1 Pythagoras
This is the first problem.
Exercise 2 Another Problem
This is the second problem.
Exercise 3 Yet Another Problem
This is the third problem.
2 Hints
Exercise 1 This is a hint to the first problem.
Exercise 3 This is a hint to the third problem.
3 Solutions
Solution 1 This is the solution to the first problem.
Solution 2 This is the solution to the second problem.
Solution 3 This is the solution to the third problem.
```

Example 10 “floating” [TeX] [PDF]

```latex
\documentclass{article}
\usepackage{xsim,newfloat,caption,lipsum}
\DeclareFloatingEnvironment[fileext=loe,listname={List of Exercises},name=Exercise,placement=htp]{ex}
'''
```

Example 11 “grade-distribution” [TeX] [PDF]

```latex
\documentclass{article}
\usepackage{xsim}
\xsimsetup{grades/split=;}
\DeclareGradeDistribution{
1 = 1 ;
1,5 = .9167 ;
2 = .8333 ;
2,5 = .75 ;
```
F. Example Documents Coming With This Package

Example 12 "listings" [TeX] [PDF]

```tex
\documentclass{article}
\usepackage{xcolor}
\usepackage{listings}
\lstset{
frame=single,
xleftmargin=20pt,
}
```

Example 13 "multiplechoice" [TeX] [PDF]

```tex
\documentclass{scrartcl}
\usepackage{amssymb,fmtcount}
\newlist{choices}{itemize}{1}
\setlist[choices]{label=$\Box$}
\newcommand\choice{\item}
\DeclareExerciseProperty{choices}
\DeclareExerciseProperty*{multiple}
```

Example 14 "listofexercises" [TeX] [PDF]

```tex
\documentclass[a4paper,10pt]{book}
\usepackage[utf8]{inputenc}
\usepackage{xsim}
\xsimsetup{
exercise/within=chapter,
exercise/template=theorem ,
exercisethe-counter=\thechapter.\arabic{exercise}
}
```

Exercise 1
Consider the following C program.

```
#include <stdio.h>

int main ( int argc , char * argv [] ) {
    printf ( " hello , world \n" );
}
```

Question 1
Select one correct answer.

□ one
□ two
□ three
□ four

Question 2
Answer this question on a separate sheet.

Question 3
Select one or more correct answers

□ one
□ two
□ three
□ four

Question 4
Select two correct answers.

□ one
□ two
□ three
□ four

Chapter 1
kinetic
1.1435-1
1.2 (Foo Bar)435-2
1.3435-3
F. Example Documents Coming With This Package

Example 15 “pointsums” [\TeX] [PDF]

\documentclass{article}
\usepackage{xsim, lipsum}
\NewDocumentCommand\printcompletepoints{\{}{\%}
\TotalExerciseGoals{points + bonus-points}
{\%}
{\%}
{\%}
{\%}
\NewDocumentCommand\pointsandbonus{\}{\%}

Example 16 “randomexercises” [\TeX] [PDF]

\documentclass{article}
\usepackage{xsim}
\DeclareExerciseCollection{foo}
\usepackage{filecontents}
\begin{filecontents*}{random.tex}
\begin{exercise}[ID=A]
\begin{filecontents*}{random.tex}
\end{exercise}

Example 17 “various” [\TeX] [PDF]

\documentclass{article}
\usepackage{xsim, lipsum, tcolorbox}
\DeclareExerciseType{question}{
exercise-env = question,
solution-env = hint,
exercise-name = Question,
solution-name = Hint,
exercise-template = default,
solution-template = default,
F. Example Documents Coming With This Package

Example 18 “texsx-13635” [TeX] [PDF] [forum]

```
\documentclass{article}
\usepackage{xsim}
\DeclareExerciseEnvironmentTemplate{theorem:remark}{\par\addvspace{\baselineskip}\noindent\textit{\IfInsideSolutionF{%\XSIMmixedcase{\GetExerciseName}}}~}%
```

Example 19 “texsx-155630” [TeX] [PDF] [forum]

```
\documentclass{article}
\usepackage{xsim}
\DeclareExerciseCollection{foo}
\begin{document}
\collectexercises{foo}
\begin{exercise}
Exercise 1
foo
Exercise 2
baz
Exercise 3
bar
```

Example 20 “texsx-199360” [TeX] [PDF] [forum]

```
\documentclass{scrartcl}
\usepackage[clear-aux]{xsim}
\usepackage{tcolorbox,blindtext}
\DeclareExerciseEnvironmentTemplate{custom}{%\begin{tcolorbox}[width = \textwidth ,
```

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

2 Zeta function
The zeta function is given by \( \zeta(s) = \sum_{n=1}^{\infty} \frac{n^{-s}}{n^s} \), where \( s \) is a complex number with real part bigger than 1. For example \( \zeta(2) = \pi^2 / 6 \).

Exercises
Exercise 2.1. Extend \( \zeta \) as far as possible and find all zeros of the function.

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

2 Zeta function
The zeta function is given by \( \zeta(s) = \sum_{n=1}^{\infty} \frac{n^{-s}}{n^s} \), where \( s \) is a complex number with real part bigger than 1. For example \( \zeta(2) = \pi^2 / 6 \).

Exercises
Exercise 2.1. Extend \( \zeta \) as far as possible and find all zeros of the function.
F. Example Documents Coming With This Package

Example 21 “texsx-299534” [TeX] [PDF] [forum]

```latex
\documentclass{article}
\usepackage{pythontex}
\begin{document}
\section{Test}
\begin{exercise}[subtitle = Codeless Question, points = 10]
1 Test Exercise 1 Codeless Question 10 p.
A question without code, worth 10 points. Subtitle and point values are in correct place.
\begin{exercise}[subtitle = Codeful Question, points = 15]
Now with PythonTeX:
\begin{verbatim}
print("hello, world!")
sum = 0
for j in range(0,100):
    sum += j
print(sum)
\end{verbatim}
1
```

Example 22 “texsx-305110” [TeX] [PDF] [forum]

```latex
\documentclass[12pt, a4paper]{book}
\usepackage{xsim}
\xsimsetup{
    exercise/within = section ,
    exercise/the-counter = \thesection.\arabic{exercise}
}
\begin{document}
Part I
EXCERCISES
1
```

Example 23 “texsx-308883” [TeX] [PDF] [forum]

```latex
\documentclass{article}
\usepackage{xsim, siunitx}
\xsimsetup{goal-print={\num{#1}}}
\sisetup{output-decimal-marker=,}
\begin{document}
\begin{exercise}[points=2.5]
foo
```

1
```

63
The solution of this exercise is on page 4.

The First Chapter

Chapter one

Exercise 2
Compute the derivative of the following function:

Example /

Exercise 1
Compute the derivative of the following function:

Example /
F. Example Documents Coming With This Package

Example 27 “tексx-369636” [TeX] [PDF] [forum]

```
1. % https://tex.stackexchange.com/q/369686/
2. \documentclass[a4paper,11pt]{article}
3. \usepackage[top=2cm, bottom=3cm, left=4cm, right=4cm]{geometry}
4. \usepackage[T1]{fontenc}
5. \usepackage{fourier}
6. \usepackage{tgpagella}
7. \usepackage[utf8]{inputenc}
8. \usepackage{xsim,needspace,adjustbox,scrextend}
```

Example 28 “tексx-369803” [TeX] [PDF] [forum]

```
1. % https://tex.stackexchange.com/q/369803
2. \documentclass[a4paper,parskip=half]{scrartcl}
3. \usepackage{utf8}{inputenc}
4. \usepackage[ngerman]{babel}
5. \usepackage{amsmath}
6. \usepackage{amsthm}
7. \usepackage{amsfonts}
8. \usepackage{amssymb}
```

Example 29 “tексx-370642” [TeX] [PDF] [forum]

```
1. % https://tex.stackexchange.com/q/370642/
2. \documentclass[a4paper,12pt]{article}
3. \usepackage{xsim}
4. \DeclareExerciseEnvironmentTemplate{simple}
5. {\parindent\textbf{\GetExerciseProperty{counter}}.}
6. {\par}
7. \xsimsetup{exercise/template=simple}
8. \begin{document}
```

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 “texsx-391530” [TeX] [PDF] [forum]

1. % https://tex.stackexchange.com/q/391530
2. \documentclass{article}
3. \usepackage{xsim}
4. \DeclareExerciseType{subquestion}{
5. exercise-env = question ,
6. solution-env = answer ,
7. exercise-name = Question ,
8. solution-name = Answer ,
9. exercise-template = item ,

Example 31 “texsx-395273” [TeX] [PDF] [forum]

1. \documentclass{article}
2. \usepackage{xsim,tcolorbox,needspace}
3. \usepackage[utf8]{inputenc}
4. \usepackage[ngerman]{babel}
5. \DeclareExerciseType{subquestion}{
6. exercise-env = question ,
7. solution-env = answer ,
8. exercise-name = Question ,
9. solution-name = Answer ,
10. exercise-template = item ,

Example 32 “texwelt-6698” [TeX] [PDF] [forum]

1. % http://texwelt.de/wissen/fragen/6698/
2. \documentclass{article}
3. \usepackage{utf8, inputenc}
4. \usepackage[ngerman]{babel}
5. \usepackage{needspace}
6. \usepackage{xsim}
7. \xsimsetup{
8. exercise/name = Aufgabe ,
9. solution/name = Lösung ,

1 Erstes Kapitel
1.1 Aufgabe 1.1
Eine erste Aufgabe
1.1 Aufgabe 1.2
Eine zweite Aufgabe
1.1.1 Aufgabe 1.3
Eine Aufgabe in einem Unterkapitel
1.1.2 Aufgabe 1.4
Noch eine Aufgabe in einem Unterkapitel
1.2 Zweites Unterkapitel
1.2.1 Aufgabe 1.5
Noch eine Aufgabe
1.2.2 Aufgabe 1.6
Eine weitere Aufgabe
1.2.3 Aufgabe 1.7
Und eine weitere Aufgabe
F. Example Documents Coming With This Package

Example 33 “texwelt-15093” [TeX] [PDF] [forum]

\documentclass[paper=a4]{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage[xsim,needspace]{xsim}
\DeclareExerciseTagging{AFB}
\DeclareExerciseEnvironmentTemplate{myexam}

Aufgabe 1.
Eine Frage
Schwierigkeit: 1 3 P.
Das ist eine sehr tolle Frage.

Aufgabe 2.
Schwierigkeit: 2 3 P.
Das ist eine sehr tolle Frage.

Aufgabe 3.
Das ist eine sehr tolle Frage.

Aufgabe 4.
Eine andere Frage
Schwierigkeit: 4 3 P.
Das ist eine sehr tolle Frage.

Aufgabe 5.
Eine Frage
Schwierigkeit: 2
Das ist eine sehr tolle Frage.

Example 34 “golatex-80640” [TeX] [PDF] [forum]

\documentclass[fontsize=12pt,parskip=half]{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}
\newlength{breite}
\setlength{breite}{160mm}
\newlength{hoehe}
\setlength{hoehe}{80mm}

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 = m \cdot c^2.

Example 35 “golatex-91339” [TeX] [PDF] [forum]

\documentclass[18pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{xsim,tcolorbox}
\usepackage{amsmath}
\xsimsetup{
exerwithin = section ,
exercounter = \thesection.arabic{exercise}}
## G. References

babel. version 3.9q, Feb. 24, 2016 (or newer).
url: http://mirror.ctan.org/macros/latex/required/babel/.

polyglossia. version 1.42.0, Aug. 6, 2015 (or newer).
url: http://mirror.ctan.org/macros/latex/contrib/polyglossia/.

url: http://mirror.ctan.org/macros/latex/contrib/booktabs/.

[Flo] Bruno Le Floch. Cunning (La)TeX tricks.
url: http://tex.stackexchange.com/a/19769/ (visited on 03/02/2017).

[L3Pa] The \LaTeX{}3 Project Team.
l3kernel. version SVN 6377, Jan. 19, 2016 (or newer).
url: http://mirror.ctan.org/macros/latex/contrib/l3kernel/.

[L3Pb] The \LaTeX{}3 Project Team.
l3packages. version SVN 6377, Jan. 19, 2016 (or newer).
url: http://mirror.ctan.org/macros/latex/contrib/l3packages/.

etoolbox. version 2.2a, Aug. 2, 2015 (or newer).
url: http://mirror.ctan.org/macros/latex/contrib/etoolbox/.

[MCo8] Frank Mittelbach and David Carlisle.
array. version 2.4c, Sept. 9, 2008 (or newer).
url: http://mirror.ctan.org/macros/latex/required/tools/.

[Nie15] Clemens Niederberger. translations. version 1.2e, Nov. 7, 2015 (or newer).
url: http://mirror.ctan.org/macros/latex/contrib/translations/.

[Nie17] Clemens Niederberger. exsheets. version 0.21i, Feb. 8, 2017 (or newer).
url: http://mirror.ctan.org/macros/latex/contrib/exsheets/.

[var] Various. Questions tagged 'exsheets'.
H. Index

A
\addbonus ....................................... 15
\addpoints .................................... 15
\AddtoExerciseGoal .............................. 14 f.
\AddtoExerciseGoalPrint ........................ 15
\AddtoExerciseGoalType ......................... 14
\AddtoExerciseGoalTypePrint ..................... 14
array (package) ................................... 2

B
babel (package) ................................... 35, 43
begin-hook ...................................... 19
Bezos, Javier ...................................... 35, 43
\blank ........................................ 25, 45 ff., 51, 55 f.
\blank-style ..................................... 45 f.
\bonus-points (property) ......................... 11, 14, 16, 34 f.
\booktabs (package) ................................ 2
Braams, Johannes .................................. 35, 43

C
Carlisle, David ..................................... 2
\chapter ........................................... 26
\chapter (property) ................................ 12, 21
\chapter-value (property) ....................... 12
Charrette, François ................................ 35, 43
clear-aux ......................................... 4, 8, 48
\collectexercises ................................. 20–23, 62
\collectexercisesstop ......................... 20–23
\collectexercisestype ......................... 20, 22
collection .......................................... 24, 26
counter (parameter) ................................ 9
counter (property) .................................. 11 f., 17, 34 f.
counter-value (property) ......................... 11 f.
Cunning (La)TeX tricks ............................ 35

D
\DeclareExerciseCollection ....................... 20, 57, 61 f.
\DeclareExerciseEnvironmentTemplate .......... 35–39, 49, 56 ff., 62, 64 f., 67
\DeclareExerciseGoal ............................ 14 f., 58
\DeclareExerciseHeadingTemplate ............. 36, 39 f., 57
\DeclareExerciseParameterIfSetTF ............ 32
\DeclareExerciseParameterGet .................. 32, 40 f.
\DeclareExerciseProperty ....................... 32
\DeclareExercisePropertyAlias ................ 32
\DeclareExercisePropertyGet .................. 32

E
end-hook ........................................ 19
equation (environment) .......................... 23
etoolbox (package) ............................... 2
\exclude ....................................... 24
\exercise (environment) ......................... 5–8, 10, 17
\exercise-env (parameter) ...................... 9 f.
\exercise-name (parameter) ..................... 9 f., 32
\exercise-template (parameter) ................ 9 f., 35
\ExerciseCollection ................................ 33
\ExerciseGoalValuePrint ....................... 14 f.
\ExerciseID ....................................... 33, 57
\ExerciseParameterGet .......................... 32, 40 f.
\ExerciseParameterIfSetTF .................... 32
\ExerciseProperty ................................ 32
\ExercisePropertyGet .......................... 32
\ExercisePropertyGetAlias .................... 32
\ExercisePropertyGlobalSave ................. 32
\ExercisePropertyIfSetTF .................... 32
\ExercisePropertySave ........................ 32
\ExerciseSetExpandedProperty ............... 31
\ExerciseSetProperty .......................... 31
\ExerciseTableCode ............................ 40 ff.
\ExerciseTableType ............................. 33, 40 ff.
\ExerciseType ................................... 33 f., 40 f.
expl3 (package) .................................. 2
\exsheets (package) .............................. 3, 16, 29, 38, 45, 47

F
Fear, Simon ....................................... 2
\fileextension .................................... 2
filecontents (environment) ..................... 7
\filled-style ................................... 45 f.
INDEX

final .................................................. 4
Floch, Bruno Le ................................... 35
\ForEachExerciseTag ............................. 33
\ForEachExerciseTranslation ................. 43
\ForEachPrintedExerciseByID ............... 34
\ForEachPrintedExerciseByType ............. 34
\ForEachUsedExerciseByID .................... 34
\ForEachUsedExerciseByType ................. 34, 40 ff.
G
\GetExerciseAliasProperty ..................... 13, 31
\GetExerciseBody ................................. 31
\GetExerciseIdForProperty .................... 31
\GetExerciseName ................................ 32, 36–39, 49, 58, 62
\GetExerciseParameter ......................... 32, 40 f.
\GetExerciseParameterTF ....................... 32
\GetExerciseProperty ............................ 13, 31, 36–39, 49, 58, 65
\GetExercisePropertyTF ....................... 31
\GetExercisePropertyForProperty .......... 36–39, 49
\GetExerciseTypeForProperty ................. 31
\GlobalSaveExerciseProperty ................ 32
goal-print ....................................... 15
grading-table (option class) .............. 27, 36
\gradingtable ....................................... 27 f., 36
H
headings ............................................. 21, 25
headings-template ............................... 21, 25, 36
I
ID (property) .................................... 10, 13, 24
id (property) ...................................... 10, 12 f., 19, 24 f., 31
\IfExerciseBooleanPropertyTF ............. 31
\IfExerciseGoalTF .............................. 49
\IfExerciseGoalSingularTF .................. 30
\IfExerciseGoalSingularTF .................. 30
\IfExerciseGoalSumTF ......................... 37, 39
\IfExerciseGoalSumTF ......................... 37, 39
\IfExerciseGoalsSumTF ...................... 30
\IfExerciseGoalsSumF ......................... 16
\IfExerciseGoalTF .............................. 30
\IfExerciseParameterSetTF .................. 32
\IfExercisePropertyExistTF ............... 31
\IfExercisePropertySetTF ................... 31
\IfExerciseSetTF ............................... 39, 49
\IfExerciseTagSetTF ......................... 39, 49
\IfExerciseTopicSetTF ....................... 33
\IfExerciseTypeGoalsSumTF ................. 33
\IfExistSolutionTF ......................... 34
\IfInsideSolutionTF ............................ 33
\IfInsideSolutionTF ............................ 36 ff., 49, 62
\IfSolutionPrintTF ........................... 34
ignore-untagged ............................... 17
L
\kernel (bundle) ................................. 2
\packages (bundle) ............................ 2
Lehman, Philipp .................................. 2
line-increment .................................. 46
line-minimum-length ......................... 46
linespread ...................................... 46
\ListExerciseTags .............................. 33
\loadxsimstyle ................................. 50
LPPL ............................................. 2
M
Mittelbach, Frank ............................... 2
N
name ............................................... 19
\NewDocumentEnvironment .................. 52
Niederberger, Clemens ......................... 2 f.
number (parameter) ............................. 10
\numberofhAexercise-environments ........ 8
\numberofexercises ........................... 9
\numberoffusedexercises ................... 33, 42
P
page (property) ................................. 11, 21
page-value (property) ......................... 11
\ParameterValue ............................... 32
path ............................................. 7 f., 48
points (property) ............................... 11, 14, 16, 34 f.
\points 8, 12, 15 f., 36–40, 42, 49, 57 f., 61, 63
polyglossia (package) ......................... 35, 43
post-hook ...................................... 19
pre-hook ...................................... 19
print ........................................... 6, 18 f., 21, 24, 34
# INDEX

| print (property) | 11, 21 |
| print-collection (option class) | 21, 36 |
| print-solutions (option class) | 26, 36 |
| print! (property) | 11, 17 |
| \printallsolutions | 25 |
| \printbonus | 15 |
| \printcollection | 21, 36, 53 |
| \printexercise | 19 |
| \printgoal | 15, 37–40, 42 |
| \printpoints | 15 |
| \printrandomexercises | 24 |
| \printsolutions | 25, 36, 51 |
| \printsolutions-type | 25, 36 |
| \printtotalbonus | 15 |
| \printtotalpoints | 15 |
| PropertyValue | 31, 36–39, 49 |
| \ProvideExerciseTagging | 16 |

| Q | Questions tagged ‘exsheets’ | 3 |

| R | Reutenauer, Arthur | 35, 43 |

| S | \SaveExerciseProperty | 32 |
| scale | 46 |
| section | 26 |
| section (property) | 11, 21 |
| section-value (property) | 12 |
| sectioning (property) | 12 |
| \SetExerciseParameter | 10, 19 |
| \SetExerciseParameters | 10 |
| \SetExerciseProperty | 31 |
| \SetExpandedExerciseProperty | 31 |
| solution (environment) | 5, 8 |
| solution-counter (parameter) | 10 |
| solution-env (parameter) | 9 |
| solution-name (parameter) | 9, 32 |
| solution-template (parameter) | 9, 35 |
| sort | 24 |
| split-aux-lists | 8 |
| style | 46 |
| style file | 29, 50 |
| subtitle (property) | 11, 34 |

| T | tags (property) | 16 |
| tags (property) | 11, 16, 33 |
| template | 19, 27, 36 |
| \The \LaTeX\mbox{3} Project Team | 2 |
| the-counter | 18 |
| topics (property) | 11, 16, 17 |
| topics (property) | 11, 16, 33 |
| \TotalExerciseGoal | 14, 30, 40, 42, 58 |
| \TotalExerciseGoals | 14, 16, 61 |
| \TotalExerciseTypeGoal | 14, 30, 40, 42 |
| \TotalExerciseTypeGoals | 14 |
| translations (package) | 2 |
| type | 27 |
| type (property) | 31 |

| U | use | 18 |
| use (property) | 11, 17, 21 |
| use! (property) | 11, 17 |
| used (property) | 11, 21 |
| \UseExerciseTags | 33 |
| \UseExerciseTemplate | 33 |

| V | \VARious | 3 |
| \\VARious | 4 |
| verbose | 4 |

| W | width | 46 |
| within | 18 |
| \\WRIGHT, Joseph | 2 |

| X | xparse (package) | 2, 52 |
| \XSI\ExpandCode | 35, 40 |
| \XSI\filewritestart | 52 |
| \XSI\filewritestop | 52 |
| \XSI\gobblechars | 52 |
| \XSI\if\chapterTF | 35 |
| \XSI\if\ifTF | 40 |
| \XSI\if\ifTF | 35 |
INDEX

\XSIMifeqTF...............................35  \xsimsetup4 ff., 10, 16, 21, 23, 26 ff., 46, 49, 53.
\XSIMifeqF.................................42  56, 58 ff., 63–67
\XSIMifeqTF.................................40 ff. \xsimstyle.................................50
\XSIMmixedcase.....................35 ff., 38–42, 49, 58, 62 \XSIMtranslate..9, 16, 35, 37, 39 ff., 42 ff., 49,
\XSIMputright......................35, 40 ff.  57, 61
\XSIMsetfilebegin...............52  xsimverb (package).......................52
\XSIMsetfileend.....................52