

ExSHEETS

vo.2b 2012/10/08

Yet another package for the creation of exercise sheets and exams.

Clemens NIEDERBERGER

<https://bitbucket.org/cgnieder/exsheets/>
contact@mychemistry.eu

English documentation

ExSHEETS provides means to create exercises or questions and their corresponding solutions. The questions can be divided into classes and/or topics and can be printed selectively. Meta-data to questions can be added and recovered.

The solutions may be printed where they are, can be collected and printed at a later point in the document altogether or section-wise or selectively by ID.

ExSHEETS provides a comprehensive interface for styling the headings of questions and solutions.

Contents

		5.2	Package Options	5
		5.3	Create Questions/Exercises and their Solutions	6
I Preliminary	3	5.3.1	The question Environment	6
1 Licence and Requirements	3	5.3.2	Options to the question Environment	7
2 Motivation	3	5.3.3	The solution Environment	8
3 Additional Packages	4	5.3.4	Options to the solution Environment	9
4 Installation	4	5.3.5	Setting the Counter . .	10
		5.3.6	Language Settings . .	10
II The ExSHEETS package	4	5.4	Counting Points	11
		5.4.1	The Commands	11
5 Package Usage	4	5.4.2	Options	12
5.1 Setup	4	5.5	A Grade Distribution	13

Contents

5.6	Printing Solutions	14	6.2.19	The ‘margin-nr’ Instance	40
5.6.1	Print all	15	6.2.20	The ‘raggedleft’ Instance	41
5.6.2	Print per chapter/section	16	6.2.21	The ‘fancy’ Instance .	41
5.6.3	Print by ID	18	6.2.22	The ‘fancy-wp’ Instance	42
5.7	Using Classes	19	6.3	The exsheets-tasks Object .	42
5.8	Using Topics	20	6.3.1	Available Options . . .	42
5.9	Adding and Using Additional Information to Questions . . .	20	6.3.2	Predefined Instances .	43
5.10	Selectively Include Questions from External Files	22	6.4	Load Custom Configurations .	43
5.11	The tasks Environment . . .	24	III The TRANSLATIONS package	44	
5.11.1	How it works	24	7 Motivation	44	
5.11.2	Available Options . . .	26	8 Usage	44	
5.11.3	Available Instances . .	28	IV The CNTFORMATS package	46	
5.11.4	Custom Labels	29	9 Motivation	46	
5.12	Filling in the Blanks	29	10 Example	46	
6 Styling your Exercise/Exam Sheets	31		11 Usage	46	
6.1	Background	31	12 Predefined Patterns and Formats	48	
6.2	The exsheets-headings Object	31	V Appendix	49	
6.2.1	Available Options . . .	31	A List of all Solutions used in this Manual	49	
6.2.2	The ‘block’ Instance .	33	Solution 8.	49	
6.2.3	The ‘runin’ Instance .	33	Solution 9.	49	
6.2.4	The ‘simple’ Instance .	34	Fancy name 10.	49	
6.2.5	The ‘block-rev’ Instance	34	Solution 16.	49	
6.2.6	The ‘block-wp’ Instance	35	Solution 17.	49	
6.2.7	The ‘block-wp-rev’ In- stance	35	Solution 18.	49	
6.2.8	The ‘block-nr’ Instance	36	Solution 22.	49	
6.2.9	The ‘block-nr-wp’ In- stance	36	Solution 23.	49	
6.2.10	The ‘runin-rev’ Instance	36	Solution 24.	49	
6.2.11	The ‘runin-wp’ Instance	37	Solution 25.	50	
6.2.12	The ‘runin-wp-rev’ In- stance	37	References	50	
6.2.13	The ‘runin-nr’ Instance	38	Index	51	
6.2.14	The ‘runin-fixed-nr’ Instance	38			
6.2.15	The ‘runin-nr-wp’ In- stance	39			
6.2.16	The ‘centered’ Instance	39			
6.2.17	The ‘centered-wp’ In- stance	39			
6.2.18	The ‘margin’ Instance	40			

Part I

Preliminary

1 Licence and Requirements

ExSHEETS is placed under the terms of the LaTeX Project Public License, version 1.3 or later (<http://www.latex-project.org/lppl.txt>). It has the status “maintained.”

ExSHEETS loads and needs the following packages: `l3kernel`, `xparse`,¹ `xtemplate`,² `l3sort`,³ `l3keys2e`,⁴ `xcolor`,⁵ `ulem`,⁶ `etoolbox`,⁷ `environ`,⁸ `lcg`⁹ and `silence`.¹⁰ **ExSHEETS** calls `\normalem` (from the `ulem` package).

2 Motivation

There are already quite a number of packages that allow the creation of exercise sheets or written exams. Just to name the most common ones: `exam`¹¹ [4], `examdesign`¹² [1], `exercise`¹³ [12], `probsoln`¹⁴ [14], `answers`¹⁵ [13] (and many more).

One thing I missed in all packages that I’ve tried out¹⁶ was a high flexibility in choosing which questions and solutions should be printed, where which solutions should be printed and so on, combined with the possibility to assign questions to different classes so one could for example create two versions of an exam out of the box. And – I can’t get enough – I also want to be able to use/design different layouts for questions additional to a standard section-like format. All these points are realized in **ExSHEETS**.

Additionally one should be able to assign some sort of meta-data to questions that of course should be easily reusable. How this can be done is explained in section 5.9.

Then there is – at least in Germany – the habit of having lists of exercises aligned in columns but counting from the left to the right instead from up to down. **ExSHEETS** provides a possibility for that (see section 5.11). I am not quite content with it as it works now, though.¹⁷

On the other hand **ExSHEETS** doesn’t – and probably won’t – offer a real possibility for creating multiple choice questions. As a fact it doesn’t provide many (if any) means to specify the *type* of question or the structure. If these are your needs take a look at `examdesign`, for example. Or exploit the possibilities `enumitem`¹⁸ [2] gives you.

I had the idea for this package in 2008. Back then my \TeX skills were by far not good enough to write it. Actually, even today I wouldn’t have been able to realize it without all the `l3` packages like `l3kernel` and `l3packages`. I actively began to develop **ExSHEETS** in spring 2011 but it wasn’t until now (September 2012) that I consider it stable enough for wider usage. At the time of writing (October 8, 2012) there still are probably lots of rough edges let alone bugs so I am very interested in all kinds of feedback.

¹ CTAN: `xparse` ² CTAN: `xtemplate` ³ CTAN: `l3sort` ⁴ CTAN: `l3keys2e` ⁵ CTAN: `xcolor` ⁶ CTAN: `ulem` ⁷ CTAN: `etoolbox` ⁸ CTAN: `environ` ⁹ CTAN: `lcg` ¹⁰ CTAN: `silence` ¹¹ CTAN: `exam`
¹² CTAN: `examdesign` ¹³ CTAN: `exercise` ¹⁴ CTAN: `probsoln` ¹⁵ CTAN: `answers` ¹⁶ Well, probably I didn’t try hard enough... ¹⁷ There are still other possibilities, for example take a look here: <http://tex.stackexchange.com/questions/67966/enumerate-in-multicols> or at the `multienum` package [5]. ¹⁸ CTAN: `enumitem`

3 Additional Packages

ExSHEETS actually bundles three packages: **ExSHEETS**, **TRANSLATIONS** and **CNTFORMATS**. **TRANSLATIONS** is described in part III and **CNTFORMATS** in part IV. These two packages provide functionality that is used by **ExSHEETS**. They can, however, be used independently from **ExSHEETS**.

4 Installation

If you install **ExSHEETS** manually beware to put the files

```
exsheets_headings.def
exsheets_headings.cfg
exsheets_tasks.def
exsheets_tasks.cfg
```

in the same directory as the `exsheets.sty` file.¹⁹ You *can* install the other two packages, **TRANSLATIONS** and **CNTFORMATS**, in different locations but since they belong to **ExSHEETS** they probably should be placed in the same directory.

As with every manual package installation you need to make sure to put the files in a directory where \TeX can find them and afterwards update the database.

Part II

The **ExSHEETS** package

5 Package Usage

5.1 Setup

The **ExSHEETS** package has three different types of options, kind of. The first type are the classic package options which are used when you load **ExSHEETS**:

```
1 \usepackage[<options>]{exsheets}
```

These options are described in section 5.2.

The second type are options that belong to a specific environment or command. These options are either used directly with the environment/command

¹⁹ That is, a directory like `texmf-local/tex/latex/exsheets`, probably

```

1 \begin{env}[<options>]
2   ...
3 \end{env}

```

or can be set with the setup command:

► `\SetupExSheets[<module>]{<options>}`

The options of the second type all belong to `modules`. Let's say you want to specify some options of the question environment. You can then say the following:

```

1 \SetupExSheets[question]{option1,option2=value2}
2 % or:
3 \SetupExSheets{question/option1,question/option2=value2}

```

The `module` an option belongs to is written in the left margin next to the when the option is described.

The third type aren't options at all, actually. However, thanks to the great `xtemplate` package you are able to define your own instances of some of the objects used by `ExSHEETS`. This is explained in a little more detail in part 6 on page 31 ff.

5.2 Package Options

The package `ExSHEETS` has some options, namely the following ones:

- `color` = <colour> Default: `exsheetsblue`
A custom colour that is used in certain but very rare circumstances.
- `counter-format` = <counter-format> Default: `qu.`
Formatting of the counter of the questions. This option takes a special kind of string that is described in section 5.3.5.
- `counter-within` = <counter> (initially empty)
Resets the question counter with every step of <counter>.
- `headings` = <instance> Default: `block`
Choose the style of the questions' and solutions' headings. There are two predefined styles: `block` and `runin`.
- `headings-format` = <code> Default: `\normalsize\bfseries`
This code is placed immediately before the headings of the questions and solutions.
- `load-headings` = `true|false` Default: `false`
Loads additional styles for the headings. More on this is described in section 6.2.

- ▶ `load-tasks = true|false` Default: false
Loads additional styles for the tasks environment. See section 5.11.
- ▶ `totoc = true|false` Default: false
This option adds the questions and solutions with their names and numbers to the table of contents.
- ▶ `questions-totoc = true|false` Default: false
This option adds the questions with their names and numbers to the table of contents.
- ▶ `solutions-totoc = true|false` Default: false
This option adds the solutions with their names and numbers to the table of contents.
- ▶ `toc-level = <toc level>` Default: subsection
This option sets the level in which questions and solutions should appear in the table of contents.
- ▶ `questions-toc-level = <toc level>` Default: subsection
This option sets the level in which questions should appear in the table of contents.
- ▶ `solutions-toc-level = <toc level>` Default: subsection
This option sets the level in which solutions should appear in the table of contents.
- ▶ `use-ref = true|false` Default: false
enable referencing to sections and chapters in a way that the references can be used with `\printsolutions`, see section 5.6.2 for details.

The toc options are demonstrated with section V and the solutions printed there being listed in the table of contents.

5.3 Create Questions/Exercises and their Solutions

Now, let's start with the most important part: the questions and (possibly) their respective solutions.

5.3.1 The question Environment

Questions are written inside the question environment:

- ▶ `\begin{question}[<options>]{<points>} ... \end{question}`

```

1 \begin{question}
2   This is our very first very difficult to solve question!
3 \end{question}

```

Exercise 1.

This is our very first very difficult to solve question!

5 Package Usage

As you can see a heading is automatically created and the question is numbered. You can of course change both the numbering and the naming, but more on that later.

The question environment takes an optional argument `{<points>}` that can be used to assign points to the question (as is common in written exams):

```
1 \begin{question}{3}
2   This is our first difficult question that is worth 3 points!
3 \end{question}
```

Exercise 2.

3 P.

This is our first difficult question that is worth 3 points!

These points are saved internally (see section 5.4 for reasons why) and are written to the right margin next to the question heading in the default setting.

You can also assign bonus points by inserting `<point>+<bonus points>` as argument.

```
1 \begin{question}{1+1}
2   This question is worth 1 point and 1 bonus point.
3 \end{question}
4 \begin{question}{+3}
5   This question is a bonus question. It is worth 3 bonus points.
6 \end{question}
```

Exercise 3.

1 (+1) P.

This question is worth 1 point and 1 bonus point.

Exercise 4.

(+3 P.)

This question is a bonus question. It is worth 3 bonus points.

5.3.2 Options to the question Environment

The question environment takes one or more of the following options:

- `question` ► `type` = `exam|exercise` Default: `exercise`
determines the type of question and changes the default name of a question from “Exercise” to “Question”. These default names are language dependent.
If you use `\usepackage[ngerman]{babel}`, for example, then the names are “Übung and “Aufgabe”.
- `question` ► `name` = `<name>` (initially empty)
sets a custom name. All predefined names are discarded.
- `question` ► `print` = `true|false` Default: `true`
prints or hides the question.

- `question` ► `ID = <id>` (initially empty)
 assigns a custom ID to the question. See section 5.6.3 for further information.
- `question` ► `class = <class>` (initially empty)
 assigns a class to the question. See section 5.7 for further information.
- `question` ► `topic = <topic>` (initially empty)
 assigns a topic to the question. See section 5.8 for further information.
- `question` ► `use = true|false` Default: true
 discards the question. Or not.

```

1 \begin{question}[type=exam]
2   This question has the type \texttt{exam}. The default name has changed
   from
3   ‘‘Exercise’’ to ‘‘Question’’.
4 \end{question}
5 \begin{question}[name=Fancy name]
6   This question has a custom name.
7 \end{question}
8 \begin{question}[print=false]
9   This question is not printed.
10 \end{question}

```

Question 5.

This question has the type exam. The default name has changed from “Exercise” to “Question”.

Fancy name 6.

This question has a custom name.

The difference between `print` and `use` lies behind the scenes: with `print = false` the question is not printed, but it still gets an individual ID, is numbered, and a possible solution is saved. This is for example useful when you want to print a sample solution for an exam. With `use = false` it is fully discarded which means it is not accessible through an ID and a possible solution will not be saved.

5.3.3 The solution Environment

If you want to save/print (more on the exact usage in section 5.6) a solution you have to use the solution environment *after* the question it belongs to and *before* the next question.

► `\begin{solution}[<options>] ... \end{solution}`

5 Package Usage

```
1 \begin{question}[ID=first]\label{qu:question_with_solution}
2   This is our first question that gets a solution!
3 \end{question}
4 \begin{solution}
5   This is the solution to exercise~\ref{qu:question_with_solution}!
6 \end{solution}
```

Exercise 8.

This is our first question that gets a solution!

You can see that in the default settings the solution is *not* written to the document. It has been saved, though, for possible later usage. We will see the solution later!

5.3.4 Options to the solution Environment

The solutions environment also has options, namely these:

`solution ► name = <name>` (initially empty)
sets a custom name.

`solution ► print = true|false` Default: false
prints or hides the solution.

Their meaning is the same as those for the question environment.

```
1 \begin{question}{5}
2   The solution to this questions gets printed where it is.
3 \end{question}
4 \begin{solution}[print]
5   See? This solution gets printed where you have put it in the code of
6   your document.
7 \end{solution}
8 \begin{question}{2.5}
9   The solution to this questions gets printed where it is \emph{and}
10  has a fancy name. Have you noticed that you can assign partial
11  points?
12 \end{question}
13 \begin{solution}[print,name=Fancy name]
14   See? This solution gets printed where you have put it and has a fancy
15   name!
16 \end{solution}
```

Exercise 9.

5 P.

The solution to this questions gets printed where it is.

Solution 9.

See? This solution gets printed where you have put it in the code of your document.

Exercise 10.

2.5 P.

The solution to this questions gets printed where it is *and* has a fancy name. Have you noticed that you can assign partial points?

Fancy name 10.

See? This solution gets printed where you have put it and has a fancy name!

5.3.5 Setting the Counter

The package option `counter-format` allows you to specify how the question counter is formatted.

The input is an arbitrary string which means you can have anything as counter number. However, the letter combinations `ch`, `se`, `qu` and `tsk` are replaced with the counters for the chapter, section, question or tasks (see section 5.11), respectively. While the last one is not really useful in this case the others allow for a combined numbering. Each of these letter combinations can have an optional argument that specifies the format of the respective counter. 1: `\arabic`, a: `\alph`, A: `\Alph`, r: `\roman` and R: `\Roman`.

```
1 \SetupExSheets{counter-format=Nr~se~(qu[a])}
2 \begin{question}
3   A question with a differently formatted number.
4 \end{question}
```

Exercise Nr 5 (k)

A question with a differently formatted number.

5.3.6 Language Settings

The names of the questions and solutions are language dependent. If you use `babel`²⁰ or `polyglossia`²¹ **ExSHEETS** will adapt to the document language. Exsheets has a number of translations but surely not all! If you miss a language please drop me a line in an email²² containing the babel language name and the correct translations for questions (possibly distinguishing between exercises and exam questions) and solutions.

Until I implement it you can add something like this to your preamble (example for Danish) and try if it works:

²⁰ CTAN: babel ²¹ CTAN: polyglossia ²² contact@mychemistry.eu

```

1 \DeclareTranslation{Danish}{exsheets-exercise-name}
2   {\0{ }velse}
3 \DeclareTranslation{Danish}{exsheets-question-name}
4   {\0gave}
5 \DeclareTranslation{Danish}{exsheets-solution-name}
6   {\0pl\0{ }sning}

```

If this isn't working it means that the language you're using is unknown to the **TRANSLATIONS** package (see part III). In this case please notify me, too. You then can still use the **name** options.

5.4 Counting Points

5.4.1 The Commands

You have seen in section 5.3.1 that you can assign points to a question. If you do so these points are printed into the margin²³ and are counted internally. But there are additional commands to assign points or bonus points and a number of commands to retrieve the sum of points and/or bonus points.

- ▶ **\addpoints*{<num>}**
This command can be used to add points assigned to subquestions. **\addpoints** will print the points (with “unit”) *and* add them to the sum of all points, **\addpoints*** will only add them but print nothing.
- ▶ **\points*{<num>}**
This command will only print the points (with “unit”) but won't add them to the sum of points.
- ▶ **\addbonus*{<num>}**
This command can be used to add bonus points assigned to subquestions. **\addbonus** will print the points (with “unit”) *and* add them to the sum of all bonus points, **\addbonus*** will only add them but print nothing.
- ▶ **\bonus*{<num>}**
This command will only print the bonus points (with “unit”) but won't add them to the sum of bonus points.
- ▶ **\pointssum***
Prints the sum of all points with or without (starred version) “unit”: 63.75 P.
- ▶ **\currentpointssum***
Prints the current sum of points with or without (starred version) “unit”: 11.5 P.
- ▶ **\bonussum***
Prints the sum of all bonus points with or without (starred version) “unit”: 4 P.

²³ Well, not necessarily. It depends on the heading style you have chosen.

► `\currentbonusum*`

Prints the current sum of bonus points with or without (starred version) “unit”: 4 P.

► `\totalpoints*`

prints the sum of the points *and* the sum of the bonus points with “unit”: 63.75 (+4) P. The starred version prints the sum of the points without “unit”: 63.75 (+4).

The last three commands need at least *two* \LaTeX runs to get the sum right.

Suppose you have an exercise worth 4 P. which consists of four questions listed with an `enumerate` environment that are all worth 1 P. each. You have two possibilities to display and count them:

	Exercise 12.	4 P.
1	<code>% uses package 'enumitem'</code>	
2	<code>\begin{question}{4}</code>	
3	<code>\begin{enumerate}[label=\alph*]</code>	a) blah (1 P.)
4	<code>\item blah (\points{1})</code>	b) blah (1 P.)
5	<code>\item blah (\points{1})</code>	c) blah (1 P.)
6	<code>\item blah (\points{1})</code>	d) blah (1 P.)
7	<code>\item blah (\points{1})</code>	
8	<code>\end{enumerate}</code>	
9	<code>\end{question}</code>	
	Exercise 13.	
10	<code>\begin{question}</code>	
11	<code>\begin{enumerate}[label=\alph*]</code>	
12	<code>\item blah (\addpoints{1})</code>	a) blah (1 P.)
13	<code>\item blah (\addpoints{1})</code>	b) blah (1 P.)
14	<code>\item blah (\addpoints{1})</code>	c) blah (1 P.)
15	<code>\item blah (\addpoints{1})</code>	d) blah (1 P.)
16	<code>\end{enumerate}</code>	
17	<code>\end{question}</code>	

5.4.2 Options

`points` ► `name = <name>`

Default: P.

Choose the “unit” for the points. If you like to differentiate between a single point and more than one point you can give a plural ending separated with a slash: `name = point/s`. This sets also the name of the bonus points.

`points` ► `name-plural = <plural form of name>`

(initially empty)

Instead of forming the plural form with an ending to the singular form this option allows to set an extra word for it. This sets also the plural form for the bonus points.

`points` ► `bonus-name = <name>`

Default: P.

Choose the “unit” for the bonus points. If you like to differentiate between a single point and more than one point you can give a plural ending separated with a slash: `bonus-name = point/s`.

5 Package Usage

points ► **bonus-plural** = <plural form of name> (initially empty)
Instead of forming the plural form with an ending to the singular form this option allows to set an extra word for it.

points ► **use-name** = true|false Default: true
Don't display the name at all. Or do.

points ► **number-format** = <any code> (initially empty)
This option allows formatting of the number, e.g. italics: **number-format** = `\textit`.

points ► **bonus-format** = <any code> (initially empty)
This option allows formatting of the number of the bonus points, e.g. italics: **bonus-format** = `\textit`.

points ► **parse** = true|false Default: true
If set to false the points are not counted and the `\totalpoints`, `\pointssum` and `\bonusum` commands won't know their value.

points ► **separate-bonus** = true|false Default: false
This option determines whether points and bonus points each get their own unit when they appear together (in the margin or with `\totalpoints`).

points ► **pre-bonus** = <tokens> Default: (+)
Code to be inserted before the bonus points when they follow normal points.

points ► **post-bonus** = <tokens> Default:)
Code to be inserted after the bonus points when they follow normal points.

```
1 \SetupExSheets[points]{name=point/s,number-format=\color{red}}
2 \begin{question}{1}
3   This one's easy so only 1 point can be earned.
4 \end{question}
5 \begin{question}{7.5}
6   But this one's hard! 7.5 points are in there for you!
7 \end{question}
```

Exercise 14. 1 point

This one's easy so only 1 point can be earned.

Exercise 15. 7.5 points

But this one's hard! 7.5 points are in there for you!

5.5 A Grade Distribution

Probably this is a rather esoteric feature but it could prove useful in some cases. Suppose you are a German math teacher and want to grade exactly corresponding to the number of points

5 Package Usage

relative to the sum of total points, regardless of how big that might be. You could do something like this to present your grading decisions for the exam:

```

1 % preamble:
2 % \DeclareRelGrades{
3 %   1      = 1 ,
4 %   {1,5} = .9167 ,
5 %   2      = .8333 ,
6 %   {2,5} = .75 ,
7 %   3      = .6667 ,
8 %   {3,5} = .5833 ,
9 %   4      = .5
10 % }
11 \small\setlength\tabcolsep{2pt}
12 \begin{tabular}{r|*8c}
13 Punkte
14   & $\grade*{1}$      & & $\le\grade*{1}$ & $\le\grade*{1,5}$ & $\le\grade
15   & $\le\grade*{2,5}$ & $\le\grade*{3}$ & $\le\grade*{3,5}$ & $\le\grade
16   & $\grade*{4}$ \\
17 Note
18   & 1 & 1--2 & 2 & 2--3 & 3 & 3--4 & 4 & 5
19 \end{tabular}

```

Punkte	$63.75 \leq 63.75 \leq 58 \leq 53 \leq 48 \leq 43 \leq 37 < 32$
Note	1 1-2 2 2-3 3 3-4 4 5

These are the available commands and options:

- `\DeclareRelGrades{<grade>=<num>, ... }`

This command is used to define grades and assign the percentage of total points to them.

- `\grade*{<grade>}`

Gives the number of points corresponding to a grade depending on the value of `\pointssum` with or without (starred version) “unit”.

`grades` ► `round = <num>`

Default: 0

The number of decimals the points of a grade are rounded to. This doesn’t apply to the maximum number of points if the rounded number would be bigger than the actual sum.

`grades` ► `half = true|false`

Default: false

If set to true points are rounded either to full or to half points.

5.6 Printing Solutions

You have already seen that you can print solutions where they are using the `print` option. But `ExSHEETS` offers you quite more possibilities.

In the next subsections the usage of the command

► `\printsolutions[<setting>]`

is discussed.

Before we do that a hint: remember that you can set the option `print` globally:

```
1 % in the preamble
2 \SetupExSheets{solution/print=true}
```

Now if you want to typeset some text depending on the option being true or not you can use the following commands:

► `\PrintSolutionsTF{<true code>}{<false code>}`

► `\PrintSolutionsT{<true code>}`

► `\PrintSolutionsF{<false code>}`

They might come in handy if you want two versions of an exercise sheet, one with the exercises and one with the solutions, and you want to add different titles to these versions, for instance.

5.6.1 Print all

The first and easiest usage of `\printsolutions` is the following:

```
1 \printsolutions
```

There is nothing more to say, really. It prints all solutions you have specified except those belonging to a question with option `use = false`. Yes, there's one more point: `\printsolutions` only knows the solutions that have been set *before* its usage! This is also true for every usage explained in the next sections.

```
1 \printsolutions
```

Solution 8.

This is the solution to exercise 8!

Solution 9.

See? This solution gets printed where you have put it in the code of your document.

Fancy name 10.

See? This solution gets printed where you have put it and has a fancy name!

5.6.2 Print per chapter/section

Current chapter/section

If you are not creating an exercise sheet or an exam but are writing a textbook you maybe want a section at the end of each chapter showing the solution to the exercises presented in that chapter. In this case use the command as follows:

```
1 \printsolutions[section]
2 % or
3 \printsolutions[chapter]
```

Again, this is pretty much self-explaining. The solutions to the questions of the current chapter²⁴ or section are printed.

```
1 \begin{question}
2   This is the first and only question in this section.
3 \end{question}
4 \begin{solution}
5   This will be one of a few solutions printed by the following call of
6   \cmd{printsolutions}.
7 \end{solution}
8 And now:
9 \printsolutions[section]
```

Exercise 16.

This is the first and only question in this section.

And now:

Solution 8.

This is the solution to exercise 8!

Solution 9.

See? This solution gets printed where you have put it in the code of your document.

Fancy name 10.

See? This solution gets printed where you have put it and has a fancy name!

Solution 16.

This will be one of a few solutions printed by the following call of `\printsolutions`.

Specific chapter/section

You can also print only the solutions from chapters or sections other than the current ones. The syntax is fairly easy:

²⁴ Only if the document class you're using *has* chapters, of course!

```

1 \printsolutions[section={1-7,10}]
2 % the same for chapters:
3 % \printsolutions[chapter={1-7,10}]

```

Solution 8.

This is the solution to exercise 8!

Solution 9.

See? This solution gets printed where you have put it in the code of your document.

Fancy name 10.

See? This solution gets printed where you have put it and has a fancy name!

Solution 16.

This will be one of a few solutions printed by the following call of `\printsolutions`.

Don't forget that `\printsolutions` cannot know the solutions from section 10 yet. It is just used to demonstrate the syntax. You can also use an open range, e.g. something like

```

1 \printsolutions[section={-4,10-}]

```

This would print the solutions from sections 1–4 and from all sections with number 10²⁵ and greater.

There is an obvious disadvantage: you have to know the section numbers! But there is a solution: use the package option `use-ref = true`. Then you can do something like

```

1 % in the preamble:
2 \usepackage[use-ref]{exsheets}
3 % somewhere in your code after \section{A really cool section
   title}:
4 \label{sec:ReallyCool}
5 % somewhere later in your code:
6 \printsolutions[section={-\S{sec:ReallyCool}}]
7 % which will print all solutions from questions up to and
8 % including the really cool section

```

With the package option `use-ref = true` each usage of `\label` will create additional labels (one preceded with `exse:` and another one with `exch:`) which store the section number and the chapter number, respectively. These are used internally by two commands `\S` and `\C` which refer to the section number and the chapter number the label was created in. *These commands are only available as arguments of `\printsolutions`.*

²⁵ Or rather where `\value{section}` is 10 or greater – the actual counter formatting is irrelevant.

5 Package Usage

Since some packages like the well known `hyperref`²⁶ for example redefine `\label` `use-ref` won't work in together with it. In this case don't use `use-ref` and set `\exlabel{<label>}` instead to remember the section/the chapter number. Its usage is just like `\label`. So the safest way is as follows:

```
1 % in the preamble:
2 \usepackage{exsheets}
3 % somewhere in your code after \section{A really cool section
   title}:
4 \exlabel{sec:ReallyCool}
5 % somewhere later in your code:
6 \printsolutions[section={-\S{sec:ReallyCool}}]
7 % which will print all solutions from questions up to and
8 % including the really cool section
```

Please be aware that the labels must be processed in a previous \LaTeX run before `\S` and `\C` can pass them on to `\printsolutions`.

5.6.3 Print by ID

Now comes the best part: you can also print selected solutions! Every question has an ID. To see which ID a question has you can call the following command:

► `\DebugExSheets{true|false}`

Let's create some more questions and take a look what this command does:

```
1 \DebugExSheets{true}
2 \begin{question}[ID=nice!]
3   A question with a nice \acs{id}!
4 \end{question}
5 \begin{solution}
6   The solution to the question with the nice \acs{id}.
7 \end{solution}
8 \begin{question}{3.75}
9   Yet another question. But this time with quarter points!
10 \end{question}
11 \begin{solution}
12   Yet another solution.
13 \end{solution}
```

ID: nice! Exercise 17.
A question with a nice ID!

ID: 18 Exercise 18.
Yet another question. But this time with quarter points!

3.75 P.

²⁶ CTAN: hyperref

So now we can call some specific solutions:

```
1 \printsolutions[byID={first,nice!,10,14}]
```

Solution 8.

This is the solution to exercise 8!

Fancy name 10.

See? This solution gets printed where you have put it and has a fancy name!

Solution 17.

The solution to the question with the nice ID.

This makes use of the `l3sort` package which at the time of writing is still considered experimental. In case you wonder where solution 14 is: question 14 has no solution given.

If you don't want that the solutions are sorted automatically but appear in the order given you can use the option

`solution ► sorted = true|false` Default: true
Sort solutions given by ID or don't.

5.7 Using Classes

For creating different variants of a written exam or different difficulty levels of an exercise sheet it comes in handy if one can assign certain classes to questions and then tell `ExSHEETS` only to use one or more specific classes.

► `use-classes = <list of classes>` (initially empty)
When this option is used only the questions belonging to the specified classes are printed and have their solutions saved.

```
1 \SetupExSheets{use-classes={easy,hard}}
2 \begin{question}[class=easy]
3   This one's easy.
4 \end{question}
5 \begin{question}[class=middle]
6   This one's a little bit harder.
7 \end{question}
8 \begin{question}[class=hard]
9   Let's see if you can solve this one!
10 \end{question}
```

Exercise 19.

This one's easy.

Exercise 20.

Let's see if you can solve this one!

Questions of classes that are not used are fully discarded. *This also means that questions that don't have a class assigned are discarded.*

5.8 Using Topics

Similarly to classes one can assign topics to questions. The usage is practically identical, the semantic meaning is different.

► `use-topics = <list of topics>` (initially empty)

When this option is used only the questions belonging to the specified topics are printed and have their solutions saved.

```

1 \SetupExSheets{use-topics={trigonometry}}
2 \begin{question}[topic=trigonometry]
3   A trigonometry question.
4 \end{question}
5 \begin{question}[topic=arithmetics]
6   A arithmetics question
7 \end{question}
```

Exercise 21.

A trigonometry question.

Questions of topics that are not used are fully discarded. *This also means that questions that don't have a topic assigned are discarded.*

If you set both `use-classes` and `use-topics` then only questions will be used that *match both categories*.

Ideally one could assign more than one topic to a question but this is *not* supported yet.

5.9 Adding and Using Additional Information to Questions

For managing lots of questions and corresponding solutions it can be very useful to be able to save and recover additional information to the questions. This is possible with the following commands. First the ones for saving:

► `\DeclareQuestionProperty{<name>}`

This command defines a question property `<name>`. It can only be used in the document preamble.

► `\SetQuestionProperties{<name>=<value>,...}`

Set the properties for a specific question. this command can only be used inside the question environment.

Now the commands for recovering the properties:

► `\QuestionNumber{<id>}`

Recover the number of the question with the ID <id>. The number is displayed according to the format set with `counter-format`.

► `\GetQuestionProperty{<name>}{<id>}`

Recover the property <name> of the question with the ID <id>. Of course the property must have been declared before.

Let's say we have declared the properties points, notes, reference and topic. Then we can do the following:

```

1 % uses 'biblatex'
2 \begin{question}[ID=center,topic=LaTeX]{3}
3 Explain how you could center text in a \LaTeX\ document.
4 \SetQuestionProperties{
5   points    = 3,
6   topic     = \TeX/\LaTeX ,
7   notes     = {How to center text.},
8   reference = {\textcite{companion}}}
9 \end{question}
10 \begin{solution}
11 To center a short part of the text body one can use the \texttt{center}
12 environment (\points{1}). Inside an environment like \texttt{table} one
13 should use \texttt{\string\centering} (\points{1}). For single lines
14 there is also the \texttt{\string\centerline} command (\points{1}).
15 \end{solution}
16 \begin{question}[ID=knuthbooks,topic=LaTeX]{2}
17 Name two books by D.\,E.\,Knuth.
18 \SetQuestionProperties{
19   points    = 2,
20   topic     = \TeX/\LaTeX ,
21   notes     = {Books by Knuth.},
22   reference = {\textcite{knuth:ct:a,knuth:ct:b,knuth:ct:c,knuth:ct:d,
23   knuth:ct:e}}}
24 \end{question}
25 \begin{solution}
26 For example two volumes from \citetitle{knuth:ct:}:
27 \citetitle{knuth:ct:a,knuth:ct:b,knuth:ct:c,knuth:ct:d,knuth:ct:e}. Each
28 valid
29 answer is worth \points{1}
30 \end{solution}

```

Exercise 22.

3 P.

Explain how you could center text in a \LaTeX document.**Exercise 23.**

2 P.

Name two books by D. E. Knuth.

It is now possible to recover these values later:

```

1 % uses 'booktabs'
2 \begin{center}
3 \begin{tabular}{lll}
4 \toprule
5 Question & Property & \\
6 \midrule
7 \QuestionNumber{center}
8 & Points & \GetQuestionProperty{points}{center} \\
9 & Topic & \GetQuestionProperty{topic}{center} \\
10 & References & \GetQuestionProperty{reference}{center} \\
11 & Note & \GetQuestionProperty{notes}{center} \\
12 \midrule
13 \QuestionNumber{knuthbooks}
14 & Points & \GetQuestionProperty{points}{knuthbooks} \\
15 & Topic & \GetQuestionProperty{topic}{knuthbooks} \\
16 & References & \GetQuestionProperty{reference}{knuthbooks} \\
17 & Note & \GetQuestionProperty{notes}{knuthbooks} \\
18 \bottomrule
19 \end{tabular}
20 \end{center}

```

Question	Property	
22.	Points	3
	Topic	\LaTeX / \LaTeX
	References	Goossens, Mittelbach, and Samarin [3]
	Note	How to center text.
23.	Points	2
	Topic	\LaTeX / \LaTeX
	References	Knuth [7, 8, 9, 10, 11]
	Note	Books by Knuth.

5.10 Selectively Include Questions from External Files

Suppose you have one or more files with questions prepared to use them as a kind of database. One for class A, say, one for class B, one for class C and so one, something like this:

5 Package Usage

```
1 % this is file classA.tex
2 \begin{question}[class=A,ID=A1,topic=X]
3   First question of class A, topic X.
4 \end{question}
5 \begin{solution}
6   First solution of class A.
7 \end{solution}
8 \begin{question}[class=A,ID=A2,topic=Y]
9   Second question of class A, topic Y.
10 \end{question}
11 \begin{solution}
12   Second solution of class A.
13 \end{solution}
14 ...
15 % end of file classA.tex
16 \endinput
```

You can of course just `\input` or `\include` it but that would of course include the whole file into your document. But wouldn't it be nice to just include selected questions? Or maybe a five random questions from the file? That is possible with the following command:

► `\includequestions[<options>]{<list of filenames>}`

If you use it without options it will have the same effect as `\input`. There are however the following options:

`include` ► `all` = `true`|`false`

`include` ► `IDs` = <list of IDs> (initially empty)
Includes only the specified questions.

`include` ► `random` = <num> (initially empty)
Includes <num> randomly selected questions. This option uses the `lcg` package to create the pseudo-random numbers.

`include` ► `seed` = <num> Default: -1
Sets the seed for the pseudo-random numbers.

The usage should be self-explainable:

```
1 % include questions A1, A3 and A4:
2 \includequestions[IDs={A1,A3,A4}]{classA.tex}
3 % or include 3 random questions:
4 \includequestions[random=3]{classA}
```

In order to be able to select the questions `ExSHEETS` needs to `\input` the file twice. The first time the available questions are determined, the second time the selected questions are used.

This unfortunately means that anything that is *not* part of a question or solution is also input twice. Either don't do that or use one of the following commands for control:

- `\PrintIfIncludeActiveTF{<true code>}{<false code>}`
- `\PrintIfIncludeActiveT{<true code>}`
- `\PrintIfIncludeActiveF{<false code>}`

If you use the `random` option you need to be aware of the following facts: the options `use-classes`, `use-topics` and `question/print` are not considered when selecting the questions which means you might end up with less questions included than you've specified. The option `question/use` is also not taken into consideration. In fact it is completely ignored and the questions are included anyway. I'm still not sure this is a feature or a bug...

5.11 The tasks Environment

5.11.1 How it works

The tasks environment is similar to a list like `enumerate` but not the same. The first difference: there is no pagebreak possible inside an item but only between items. The second difference: the enumeration default is a), b), c) ... The third difference: the input syntax:

- `\begin{tasks}[<options>][<num of columns>]` list of tasks, each ended by a semicolon `\end{tasks}`

Let's use some sample text in the next few examples:

```

1 \def\s{This is some sample text we will use to create a
   somewhat longer text
2 spanning a few lines.}
3 \def\sample{\s\ \s\par\s}
```

```

1 Some text before the list.
2 \begin{tasks}
3   \sample ;
4   \sample ;
5   \sample ;
6 \end{tasks}
7 And also some text after it.
```

5 Package Usage

Some text before the list.

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

And also some text after it.

The environment takes the optional argument `{<num of columns>}` with which the number of columns used by the environment is specified.

```
1 \begin{tasks}{2}  
2 \sample ;  
3 \s \s ;  
4 \s ;  
5 \sample ;  
6 \s\par\s ;  
7 \end{tasks}
```

- | | |
|---|---|
| <p>a) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.
This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | <p>b) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> |
| <p>c) This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | <p>d) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.
This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> |
| <p>e) This is some sample text we will use to create a somewhat longer text spanning a few lines.
This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | |

5.11.2 Available Options

The environment has some options, namely these:

- tasks** ► **style** = <instance> (initially empty)
Choose the instance to be used. Read more on this in section 6.3.
- tasks** ► **label** = <label specs> (initially empty)
Sets a custom label. The letters tsk are replaced with the task-counter. An optional argument directly following these letters specifies the counter format: 1: \arabic, a: \alph, A: \Alph, r: \roman and R: \Roman.
- tasks** ► **label-width** = <dim> Default: 3ex
Sets the width of the item labels.
- tasks** ► **before-skip** = <skip> Default: .33\baselineskip
Sets the skip before the list.
- tasks** ► **after-skip** = <skip> Default: .33\baselineskip
Sets the skip after the list.

Now the same list as above but with three columns and a different label:

```

1 \begin{tasks}[label=(tsk[r]),label-width=4ex]{3}
2   \sample ;
3   \s\ \s ;
4   \s ;
5   \sample ;
6   \s\par\s ;
7 \end{tasks}

```

- | | | |
|---|---|--|
| <p>(i) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | <p>(ii) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | <p>(iii) This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> |
| <p>(iv) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | <p>(v) This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.</p> | |

Let's use it inside a question:

```

1 \SetupExSheets[tasks]{label=qu.tsk,label-width=5ex}
2 \begin{question}[type=exam]{4}
3 I have these two tasks for you. Shall we begin?
4 \begin{tasks}{2}
5   The first task: easy! ;
6   The second task: even more so! ;
7 \end{tasks}
8 \end{question}
9 \begin{solution}[print]
10 Now, let's see\ldots\ ah, yes:
11 \begin{tasks}
12   This is the first solution. Told you it was easy. ;
13   This is the second solution. And of course you knew that! ;
14 \end{tasks}
15 \end{solution}

```

Question 24.

4 P.

I have these two tasks for you. Shall we begin?

24.1 The first task: easy!

24.2 The second task: even more so!

Solution 24.

Now, let's see... ah, yes:

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

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

5.11.3 Available Instances

When you use the package option `load-tasks` there are currently three additional instances for the tasks are available:

itemize uses `\labelitemi` as labels.

enumerate enumerates the items with 1., 2., ...

multiplechoice a – well – ‘multiple choice’ list.

```

1 \begin{tasks}[style=itemize]{2}
2   that's just how\ldots ;
3   \ldots we expected ;
4 \end{tasks}
5 \begin{tasks}[style=enumerate]{2}
6   that's just how\ldots ;
7   \ldots we expected ;
8 \end{tasks}
9 \begin{tasks}[style=multiplechoice]{2}
10  that's just how\ldots ;
11  \ldots we expected ;
12 \end{tasks}

```

- | | |
|---|---|
| • that's just how... | • ...we expected |
| 1. that's just how... | 2. ...we expected |
| <input type="checkbox"/> that's just how... | <input type="checkbox"/> ...we expected |

5.11.4 Custom Labels

If you want to change a single label inside a list, you can use

► `\task{<label>}`

in one of the tasks. This will temporarily overwrite the default label.

1 \begin{tasks}[style=itemize]	• a standard item
2 a standard item ;	• another one
3 another one ;	
4 \task{+}a different one ;	+ a different one
5 and another one ;	
6 \end{tasks}	• and another one

5.12 Filling in the Blanks

Both in exercise sheets and in exams it is sometimes desirable to be able to create _____ that have to be filled in. Or may be some more lines: _____

- `\blank[<options>]{<text to be filled in>}`
 creates a blank in normal text or in a question but fills the text of its argument if inside a solution.

The options are these:

`blank` ► `style` = `line|wave|dline|dotted|dashed` Default: `line`
 The style of the line.

`blank` ► `width` = `<dim>` (initially empty)
 The width of the line. If it is not used the width of the filled in text is used.

```

1 \begin{question}
2   Try to fill in \blank[width=4cm]{these} blanks. All of them
3   \blank[style=dotted]{are created} by using the \cmd{blank}
4   \blank[style=dashed]{command}.
5 \end{question}
6 \begin{solution}[print]
7   Try to fill in \blank[width=4cm]{these} blanks. All of them
8   \blank[style=dotted]{are created} by using the \cmd{blank}
9   \blank[style=dashed]{command}.
10 \end{solution}
```

Exercise 25.

Try to fill in _____ blanks. All of them by using
 the `\blank` _____.

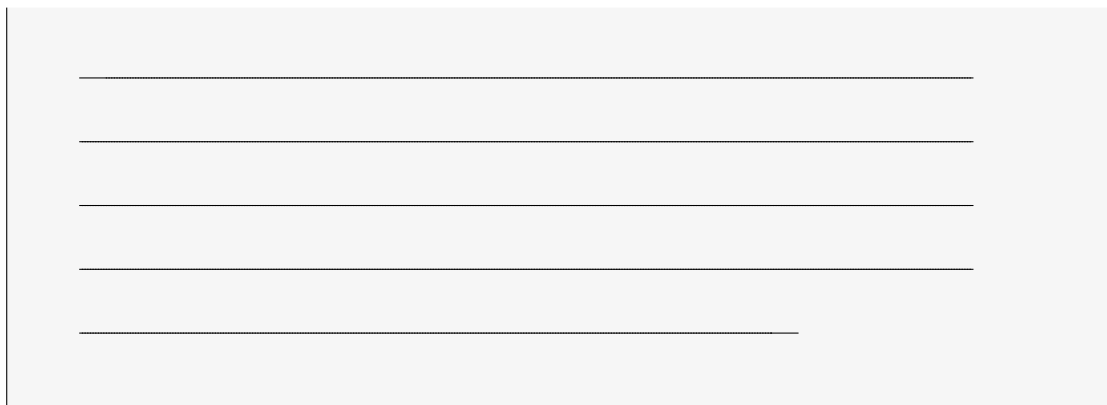
Solution 25.

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:

```

1 % \usepackage{setspace}
2 \spacing{2}\blank[width=4.8\linewidth]{}\endspacing
```



6 Styling your Exercise/Exam Sheets

6.1 Background

The **ExSHEETS** package makes extensive use of \LaTeX ’s coffins²⁷ as well as its templates concept.²⁸ The latter allows a rather easy extension and customization of some of **ExSHEETS**’ environments. To be more precise: you can define your own instances for the headings used for questions and solutions and for the tasks environment.

What this package doesn’t provide is changing the background of questions or framing them. But this is easily possible using the `mdframed`²⁹ package and its `\surroundwithmdframed` command.

6.2 The `exsheets-headings` Object

ExSHEETS defines the object `exsheets-headings` and one template for it, the ‘default’ template. The package also defines two instances of this template, the ‘block’ instance and the ‘runin’ instance.

<pre> 1 \SetupExSheets{headings=block} 2 \begin{question}{1} 3 a ‘block’ heading 4 \end{question} 5 \SetupExSheets{headings=runin} 6 \begin{question}{1} 7 a ‘runin’ heading 8 \end{question} </pre>	<p>Exercise 26.</p> <p>a ‘block’ heading</p> <p>Exercise 27. a ‘runin’ heading</p>	<p>1 P.</p> <p></p> <p>1 P.</p>
--	--	---------------------------------

6.2.1 Available Options

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

²⁷ See the documentation to the `xcoffins` package for more information on that. ²⁸ Have a look into the documentation to the `xtemplate` package. ²⁹ CTAN: `mdframed`

6 Styling your Exercise/Exam Sheets

listed in the definition for the template interface:

```
1 \DeclareTemplateInterface{exsheets-heading}{default}{3}
2 {
3   % option      : type      = default
4   runin         : boolean   = false ,
5   indent-first  : boolean   = false ,
6   toc-reversed  : boolean   = false ,
7   vscale        : real      = 1     ,
8   above         : length    = 2pt   ,
9   below         : length    = 2pt   ,
10  main           : tokenlist =       ,
11  pre-code       : tokenlist =       ,
12  post-code      : tokenlist =       ,
13  title-format   : tokenlist =       ,
14  title-pre-code : tokenlist =       ,
15  title-post-code : tokenlist =      ,
16  number-format  : tokenlist =       ,
17  number-pre-code : tokenlist =      ,
18  number-post-code : tokenlist =     ,
19  points-format  : tokenlist =       ,
20  points-pre-code : tokenlist =      ,
21  points-post-code : tokenlist =     ,
22  join           : tokenlist =       ,
23  attach         : tokenlist =       ,
24 }
```

Each heading is built with at most four coffins available with the names ‘main’, ‘title’, ‘number’ and ‘points’. Those coffins place possibly the whole heading, the title, the question number and the assigned points. The only coffin that’s always typeset is the ‘main’ coffin, which is empty per default.

Coffins can be joined (two become one, the first extends its bounding box to contain the second) using the following syntax:

```
1 join = coffin1[handle11,handle12]coffin2[handle21,handle22](x-
   offset,y-offset)
```

The syntax for attaching (two become one, the first does *not* extend its bounding box around the second) the syntax is the same.

More on coffin handles is described in the documentation for the xcoffins. Figure 1 briefly demonstrates the available handle pairs.

The following subsections will show all definitions of the instances available with the package option `load-headings` and how they look. This will hopefully give you enough ideas to create your own instance if you want to have another heading style than the ones available.

Of you use the option `load-headings` each of the following instances is available through the option `headings = <instance>`.

6 Styling your Exercise/Exam Sheets

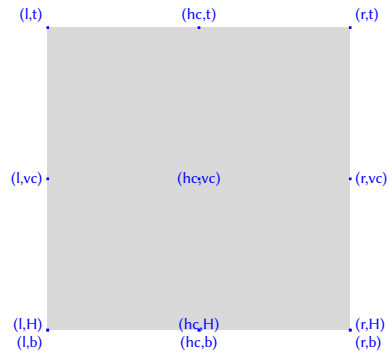


FIGURE 1: Available handles for a horizontal coffin.

6.2.2 The ‘block’ Instance

```

1 \DeclareInstance{exsheets-heading}{block}{default}
2 {
3   join          = { title[r,B]number[l,B](1ex,0pt) } ,
4   attach        =
5   {
6     main[l,vc]title[l,vc](0pt,0pt) ;
7     main[r,vc]points[l,vc](\marginparsep,0pt)
8   }
9 }

```

Exercise 28.

1P.

A ‘block’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.3 The ‘runin’ Instance

```

1 \DeclareInstance{exsheets-heading}{runin}{default}
2 {
3   runin          = true ,
4   number-post-code = \space ,
5   attach         =
6   { main[l,vc]points[l,vc](\linewidth+\marginparsep,0pt) }
7   ,
8   join           =
9   {
10    main[r,vc]title[r,vc](0pt,0pt) ;
11    main[r,vc]number[l,vc](1ex,0pt)
12  }
13 }

```

Exercise 29. A ‘runin’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. 1 P.

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

6.2.4 The ‘simple’ Instance

```

1  \DeclareInstance{exsheets-heading}{simple}{default}
2  {
3    title-format      = \normalsize ,
4    points-pre-code   = ( ,
5    points-post-code  = ) ,
6    attach            = { main[l,t]number[l,t](0pt,0pt) } ,
7    join              =
8      {
9        number[r,b]title[l,b](1ex,0pt) ;
10       main[l,b]points[l,t](1em,0pt)
11     }
12  }
```

30. Exercise (1 P.)

A ‘simple’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.5 The ‘block-rev’ Instance

```

1  \DeclareInstance{exsheets-heading}{block-rev}{default}
2  {
3    toc-reversed      = true ,
4    join              = { number[r,B]title[l,B](1ex,0pt) } ,
5    attach            =
6      {
7        main[l,vc]number[l,vc](0pt,0pt) ;
8        main[r,vc]points[l,vc](\marginparsep,0pt)
9      }
10  }
```

31. Exercise

1 P.

A ‘block-rev’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

6 Styling your Exercise/Exam Sheets

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

6.2.6 The ‘block-wp’ Instance

```
1  \DeclareInstance{exsheets-heading}{block-wp}{default}
2  {
3    points-pre-code = ( ,
4    points-post-code = ) ,
5    join           =
6    {
7      title[r,B]number[l,B](1ex,0pt) ;
8      title[r,B]points[l,B](1ex,0pt)
9    } ,
10   attach          = { main[l,vc]title[l,vc](0pt,0pt) }
11 }
```

Exercise 32. (1P.)

A ‘block-wp’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.7 The ‘block-wp-rev’ Instance

```
1  \DeclareInstance{exsheets-heading}{block-wp-rev}{default}
2  {
3    toc-reversed    = true ,
4    points-pre-code = ( ,
5    points-post-code = ) ,
6    join           =
7    {
8      number[r,B]title[l,B](1ex,0pt) ;
9      number[r,B]points[l,B](1ex,0pt)
10   } ,
11   attach          = { main[l,vc]number[l,vc](0pt,0pt) }
12 }
```

33. Exercise (1P.)

A ‘block-wp-rev’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.8 The ‘block-nr’ Instance

```

1 \DeclareInstance{exsheets-heading}{block-nr}{default}
2 {
3   attach      =
4   {
5     main[l,vc]number[l,vc](0pt,0pt) ;
6     main[r,vc]points[l,vc](\marginparsep,0pt)
7   }
8 }

```

34.

1P.

A ‘block-nr’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.9 The ‘block-nr-wp’ Instance

```

1 \DeclareInstance{exsheets-heading}{block-nr-wp}{default}
2 {
3   points-pre-code = ( ,
4   points-post-code = ) ,
5   join           = { number[r,vc]points[l,vc](1ex,0pt) } ,
6   attach         = { main[l,vc]number[l,vc](0pt,0pt) }
7 }

```

35. (1P.)

A ‘block-nr-wp’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.10 The ‘runin-rev’ Instance

```

1 \DeclareInstance{exsheets-heading}{runin-rev}{default}
2 {
3   toc-reversed = true ,
4   runin        = true ,
5   title-post-code = \space ,
6   attach       =
7   { main[l,vc]points[l,vc](\linewidth+\marginparsep,0pt) }
8   ,
9   join         =

```

```

9      {
10         main[r,vc]number[r,vc](0pt,0pt) ;
11         main[r,vc]title[l,vc](1ex,0pt)
12     }
13 }

```

36. Exercise A ‘runin-rev’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. 1 P.

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

6.2.11 The ‘runin-wp’ Instance

```

1  \DeclareInstance{exsheets-heading}{runin-wp}{default}
2  {
3      runin          = true ,
4      points-pre-code = ( ,
5      points-post-code = )\space ,
6      join           =
7      {
8          main[r,vc]title[r,vc](0pt,0pt) ;
9          main[r,vc]number[l,vc](1ex,0pt) ;
10         main[r,vc]points[l,vc](1ex,0pt)
11     }
12 }

```

Exercise 37. (1 P.) A ‘runin-wp’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.12 The ‘runin-wp-rev’ Instance

```

1  \DeclareInstance{exsheets-heading}{runin-wp-rev}{default}
2  {
3      toc-reversed    = true ,
4      runin           = true ,
5      points-pre-code = ( ,
6      points-post-code = )\space ,
7      join            =
8      {
9          main[r,vc]number[r,vc](0pt,0pt) ;
10         main[r,vc]title[l,vc](1ex,0pt) ;
11         main[r,vc]points[l,vc](1ex,0pt)

```

```

12     }
13 }

```

38. Exercise (1P.) A ‘runin-wp-rev’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.13 The ‘runin-nr’ Instance

```

1  \DeclareInstance{exsheets-heading}{runin-nr}{default}
2  {
3    runin          = true ,
4    number-post-code = \space ,
5    attach         =
6      { main[l,vc]points[l,vc](\linewidth+\marginparsep,0pt) }
7    ,
8    join           = { main[r,vc]number[l,vc](0pt,0pt) }
9  }

```

39. A ‘runin-nr’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. 1P.

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

6.2.14 The ‘runin-fixed-nr’ Instance

```

1  \DeclareInstance{exsheets-heading}{runin-fixed-nr}{default}
2  {
3    runin          = true ,
4    number-pre-code = \hbox to 2em \bgroup ,
5    number-post-code = \hfil\egroup ,
6    attach         =
7      { main[l,vc]points[l,vc](\linewidth+\marginparsep,0pt) }
8    ,
9    join           = { main[r,vc]number[l,vc](0pt,0pt) }
10 }

```

40. A ‘runin-fixed-nr’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. 1P.

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

6.2.15 The ‘runin-nr-wp’ Instance

```

1  \DeclareInstance{exsheets-heading}{runin-nr-wp}{default}
2  {
3      runin          = true ,
4      points-pre-code = ( ,
5      points-post-code = )\space ,
6      join           =
7      {
8          main[r,vc]number[l,vc](0pt,0pt) ;
9          main[r,vc]points[l,vc](1ex,0pt)
10     }
11 }

```

41. (1P.) A ‘runin-nr-wp’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.16 The ‘centered’ Instance

```

1  \DeclareInstance{exsheets-heading}{centered}{default}
2  {
3      join           = { title[r,B]number[l,B](1ex,0pt) } ,
4      attach         =
5      {
6          main[hc,vc]title[hc,vc](0pt,0pt) ;
7          main[r,vc]points[l,vc](\marginparsep,0pt)
8      }
9  }

```

Exercise 42.

1P.

A ‘centered’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.17 The ‘centered-wp’ Instance

```

1  \DeclareInstance{exsheets-heading}{centered-wp}{default}
2  {
3      points-pre-code = ( ,
4      points-post-code = ) ,
5      join           =

```

6 Styling your Exercise/Exam Sheets

```
6      {
7          title[r,B]number[l,B](1ex,0pt) ;
8          title[r,B]points[l,B](1ex,0pt)
9      } ,
10     attach          = { main[hc,vc]title[hc,vc](0pt,0pt) }
11 }
```

Exercise 43. (1P.)

A ‘centered-wp’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.18 The ‘margin’ Instance

```
1  \DeclareInstance{exsheets-heading}{margin}{default}
2  {
3      runin          = true ,
4      number-post-code = \space ,
5      points-pre-code  = ( ,
6      points-post-code = )\space ,
7      join            = { title[r,b]number[l,b](1ex,0pt) } ,
8      attach          =
9      {
10         main[l,vc]title[r,vc](0pt,0pt) ;
11         main[l,b]points[r,t](0pt,0pt)
12     }
13 }
```

Exercise 44. A ‘margin’ heading. This is some sample text we will use to create a somewhat longer text (1P.) spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.19 The ‘margin-nr’ Instance

```
1  \DeclareInstance{exsheets-heading}{margin-nr}{default}
2  {
3      runin = true ,
4      attach =
5      {
6          main[l,vc]number[r,vc](-1ex,0pt) ;
7          main[r,vc]points[l,vc](\linewidth+\marginparsep,0pt)
8      }
9  }
```

45. A ‘margin-nr’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines. 1 P.

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

6.2.20 The ‘raggedleft’ Instance

```

1 \DeclareInstance{exsheets-heading}{raggedleft}{default}
2 {
3   join          = { title[r,B]number[l,B](1ex,0pt) } ,
4   attach        =
5   {
6     main[r,vc]title[r,vc](0pt,0pt) ;
7     main[r,vc]points[l,vc](\marginparsep,0pt)
8   }
9 }

```

Exercise 46. 1 P.

A ‘raggedleft’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.21 The ‘fancy’ Instance

```

1 \DeclareInstance{exsheets-heading}{fancy}{default}
2 {
3   toc-reversed   = true ,
4   indent-first   = true ,
5   vscale         = 2 ,
6   pre-code       = \rule{\linewidth}{1pt} ,
7   post-code      = \rule{\linewidth}{1pt} ,
8   title-format   = \large\scshape\color{exsheetsred} ,
9   number-format  = \large\bfseries\color{exsheetsblue} ,
10  points-format   = \itshape ,
11  join           = { number[r,B] title[l,B] (1ex,0pt) } ,
12  attach         =
13  {
14    main[hc,vc]number[hc,vc](0pt,0pt) ;
15    main[l,vc]points[r,vc](\marginparsep,0pt)
16  }
17 }

```

1P.

47. EXERCISE

A ‘fancy’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.2.22 The ‘fancy-wp’ Instance

```

1  \DeclareInstance{exsheets-heading}{fancy-wp}{default}
2  {
3      toc-reversed      = true ,
4      indent-first      = true ,
5      vscale            = 2 ,
6      pre-code          = \rule{\linewidth}{1pt} ,
7      post-code         = \rule{\linewidth}{1pt} ,
8      title-format      = \large\scshape\color{exsheetsred} ,
9      number-format     = \large\bfseries\color{exsheetsblue} ,
10     points-format     = \itshape ,
11     points-pre-code    = ( ,
12     points-post-code   = ) ,
13     join               =
14     {
15         number[r,B]title[l,B](1ex,0pt) ;
16         number[r,B]points[l,B](1ex,0pt)
17     } ,
18     attach             = { main[hc,vc]number[hc,vc](0pt,0pt) }
19 }

```

48. EXERCISE (1P.)

A ‘fancy-wp’ heading. This is some sample text we will use to create a somewhat longer text spanning a few lines. This is some sample text we will use to create a somewhat longer text spanning a few lines.

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

6.3 The exsheets-tasks Object

The second object that’s defined by **ExSHEETS** is the ‘exsheets-tasks’ object. This time there are four instances available for the one template (again ‘default’) that was defined.

6.3.1 Available Options

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

6 Styling your Exercise/Exam Sheets

```
1 \DeclareTemplateInterface{exsheets-tasks}{default}{3}
2 {
3   % option      : type      = default
4   enumerate     : boolean   = true      ,
5   label         : tokenlist
6   indent        : length    = 2.5em     ,
7   label-format  : tokenlist = tsk[a])   ,
8   label-width   : length    = 3ex       ,
9   after-item-skip : length   = 2ex
10 }
```

6.3.2 Predefined Instances

This is rather brief this time:

```
1 % ALPHABETIZE: a) b) c)
2 \DeclareInstance{exsheets-tasks}{alphabetize}{default}{}
3 % available when 'load-tasks=true':
4 % ITEMIZE:
5 \DeclareInstance{exsheets-tasks}{itemize}{default}
6 { enumerate = false }
7 % ENUMERATE:
8 \DeclareInstance{exsheets-tasks}{enumerate}{default}
9 { label-format = tsk. }
10 % MULTIPLECHOICE:
11 \DeclareInstance{exsheets-tasks}{multiplechoice}{default}
12 {
13   enumerate = false ,
14   label     = \choicebox ,
15 }
```

6.4 Load Custom Configurations

If you have custom configurations you want to be loaded automatically then save them in a file `exsheets_configurations.cfg`. If this file is present it will be loaded `\AtBeginDocument`.

Part III

The **TRANSLATIONS** package

7 Motivation

I wrote this package when I was in need for an expandable version of translator's `\translate`. If you don't need that then there is no need whatsoever for this package as translator from the beamer³⁰ bundle provides all the functionality **TRANSLATIONS** does (except for: translator's `\translate` isn't expandable, `\GetTranslation` is.).

8 Usage

These are the commands provided by **TRANSLATIONS**:

- ▶ `\DeclareLanguage{<lang>}`
Declare a language that can be used by **TRANSLATIONS**. If the language already exists it will be silently redefined. This command can only be used in the preamble.
- ▶ `\DeclareLanguageAlias{<lang2>}{<lang1>}`
Declares `<lang2>` to be an alias of `<lang1>`. If `<lang1>` doesn't exist yet a warning will be raised and it will be defined. This command can only be used in the preamble.
- ▶ `\NewTranslation{<lang>}{<key>}{<translation>}`
Defines a translation of key `<key>` for the language `<lang>`. An error will be raised if a translation of `<key>` already exists. This command can only be used in the preamble.
- ▶ `\RenewTranslation{<lang>}{<key>}{<translation>}`
Redefines a translation of key `<key>` for the language `<lang>`. An error will be raised if no translation of `<key>` exists. This command can only be used in the preamble.
- ▶ `\DeclareTranslation{<lang>}{<key>}{<translation>}`
Defines a translation of key `<key>` for the language `<lang>`. No error will be raised if a translation of `<key>` already exists. This command can only be used in the preamble.
- ▷ `\GetTranslationFor{<lang>}{<key>}`
Fetches and prints the translation of `<key>` for the language `<lang>`. This command is expandable.
- ▷ `\GetTranslation{<key>}`
Fetches and prints the translation of `<key>` for the currently active language (as for example set by babel). This command is expandable.
- ▶ `\SaveTranslationFor{<cmd>}{<lang>}{<key>}`
Fetches and saves the translation of `<key>` for the language `<lang>` in the macro `<cmd>`.

³⁰ CTAN: beamer

- `\SaveTranslation{<cmd>}{<key>}`
Fetches and saves the translation of <key> or the currently active language (as for example set by babel) in the macro <cmd>.
- `\LoadDictionary{<name>}`
Loads a file named <name>-<lang>.trsl where <lang> corresponds to the lowercase name of the current language as defined with `\DeclareLanguage`. This file should contain the translations for the specified language.
- `\LoadDictionaryFor{<lang>}{<name>}`
Loads a file named <name>-<lang>.trsl.

Quite a number of languages already are defined, either directly or via an alias. So, before you define a language you should take a look in `translations.sty` if the language doesn't already exist.

Here is a small example of usage:

```

1 % in the preamble:
2 % \DeclareTranslation{English}{Kueche}{kitchen}
3 % \DeclareTranslation{German}{Kueche}{K\"uche}
4 % \DeclareTranslation{Spanish}{Kueche}{cocina}
5 % \DeclareTranslation{French}{Kueche}{cuisine}
6
7 \GetTranslation{Kueche}
8 \SaveTranslation\kitchen{Kueche}
9 \SaveTranslationFor\cuisine{french}{Kueche}
10
11 \selectlanguage{ngerman}
12 \GetTranslation{Kueche} \kitchen \GetTranslationFor{spanish}{Kueche}
13 \cuisine

kitchen
Küche kitchen cocina cuisine

```

A typical dictionary file should look as follows:

```

1 % this is file housing-german.trsl
2 \ProvideDictionaryFor{german}{housing}[<version info>]
3 \DeclareTranslation{German}{kitchen (housing)}{K\"uche}
4 \DeclareTranslation{German}{bathroom (housing)}{Bad}
5 \DeclareTranslation{German}{living room (housing)}{Wohnzimmer}

6 \DeclareTranslation{German}{bedroom (housing)}{Schlafzimmer}
7 ...
8 \endinput

```

Part IV

The **CNTFORMATS** package

9 Motivation

CNTFORMATS provides a way to format counters with what I will call patterns. This does not in any way effect the usual \LaTeX way of treating counters and does not use `\the<counter>` nor is it affected by the redefinition of them.

This package is aimed at package or class authors and probably not very useful for document authors.

10 Example

A use case typically looks as follows:

```
1 \ReadCounterPattern{se.sse} 10.0
```

where the key `se` stands for the current value of the section counter and `sse` for subsection, respectively. `se.sse` is an example for what will be called *pattern*. The keys for the counters can have optional arguments that specify the format:

```
1 \stepcounter{subsection}
2 \ReadCounterPattern{se[A](sse[r])} J(i)
```

`A` stands for `\Alph` and `r` for `\roman`. A complete list can be found in table 2 on page 48. As you can see you can insert arbitrary other tokens in a pattern that won't be changed.

11 Usage

In order to make counters known to **CNTFORMATS** the following commands are used:

- `\AddCounterPattern* [<module>] {<counter>} {<pattern>}`
This command will make the (existing) counter `<counter>` known to **CNTFORMATS** and assign the pattern `<pattern>` to it.
- `\NewCounterPattern* [<module>] {<counter>} {<pattern>}`
This command will create a new counter `<counter>`, make it known to **CNTFORMATS** and assign the pattern `<pattern>` to it.
- `\ReadCounterFrom [<module>] {<counter>} {<internal cmd>}`
If you use one of the commands above with the starred version the number for the pattern is not automatically fetched from the internal `\c@<counter>`. This can now be assigned with `\ReadCounterFrom` where `<internal cmd>` is the macro that holds the number.

The commands above can only be used in the document preamble.

After the creation of these pattern markers one wants to be able to use them. There are a number of macros that allow different aspects of usage.

- ▶ `\ReadCounterPattern[<module>]{<pattern>}`
Reads, interprets and prints a pattern.
- ▶ `\@cntfmts@parsed@pattern`
After `\ReadCounterPattern` has been used the current pattern interpretation is stored in this macro. The *interpretation* is *not* what is printed. See the examples below for details.
- ▶ `\ReadCounterPatternFrom[<module>]{<macro that holds pattern>}`
Reads, interprets and prints a pattern that's stored in a macro.
Otherwise the same as `\ReadCounterPattern`.
- ▶ `\SaveCounterPattern{<cmd a>}{<cmd b>}{<pattern>}`
Saves the `<pattern>` in `<cmd a>` and the interpreted pattern in `<cmd b>`.
- ▶ `\eSaveCounterPattern[<module>]{<cmd a>}{<cmd b>}{<pattern>}`
Saves the `<pattern>` in `<cmd a>` and the expanded pattern in `<cmd b>`.
- ▶ `\SaveCounterPatternFrom[<module>]{<cmd a>}{<cmd b>}{<macro that holds pattern>}`
Like `\SaveCounterPattern` but reads the pattern from a macro.
- ▶ `\eSaveCounterPatternFrom[<module>]{<cmd a>}{<cmd b>}{<macro that holds pattern>}`
Like `\eSaveCounterPattern` but reads the pattern from a macro.

The optional argument `<module>` should be specific for a package, say, so that different patterns for the section for example don't interfere with each other. If you leave the argument the default module `cntfmts` is used.

The **ExSHEETS** packages uses the commands with the module `exsheets`. You can find the following lines in **ExSHEETS**' code:

```

1 \AddCounterPattern*[exsheets]{section}{se}
2 \ReadCounterFrom[exsheets]{section}
3   \l__exsheets_counter_sec_int
4
5 \NewCounterPattern*[exsheets]{question}{qu}
6 \ReadCounterFrom[exsheets]{question}
7   \l__exsheets_counter_qu_int

```

Now let's see a short example that hopefully explains what the macros do:

```

1 % preamble
2 % \NewCounterPattern{testa}{ta}
3 \setcounter{testa}{11}
4 \ReadCounterPattern{ta}
5 \ReadCounterPattern{ta[a]} \
6 \ttfamily\makeatletter
7 \meaning\@cntfmts@parsed@pattern
8
9 \bigskip
10 \SaveCounterPattern\tmpa\tmpb{ta[a]}
11 \meaning\tmpa \
12 \meaning\tmpb
13
14 \bigskip
15 \eSaveCounterPattern\tmpa\tmpb{ta[a]}
16 \meaning\tmpa \
17 \meaning\tmpb

11 k
macro:->{}\csuse {@cntfmts@read@ta@counter}[a]\relax

macro:->ta[a]
macro:->{}\csuse {@cntfmts@read@ta@counter}[a]\relax

macro:->ta[a]
macro:->{}k\relax

```

You can see that somehow additional (empty) groups and a `\relax` found their way into the interpreted and thus the expanded pattern. This is due to the fact that reading optional arguments expandably isn't easy and must have some safety net.

12 Predefined Patterns and Formats

CNTFORMATS predefines a number of pattern keys. These are listed in table 1.

TABLE 1: Predefined Patterns for the module cntfmts.

counter	pattern
chapter	ch
section	se
subsection	sse
subsubsection	ssse
paragraph	pg

TABLE 2: Predefined Formats

key	format
1	\arabic
a	\alph
A	\Alph
r	\roman
R	\Roman

Table 2 lists the predefined formats. If you want you can add own formats.

- `\NewPatternFormat{<pattern>}{<format>}`
`<format>` is a number presentation command like `\@alph`. This command can only be used in the preamble.

```

1 % preamble
2 % \usepackage{alphalph}
3 % \makeatletter
4 % \NewPatternFormat{aa}{\alphalph@alph}
5 % \makeatother
6 \ReadCounterPattern{se[aa]}

```

1

Part V

Appendix

A List of all Solutions used in this Manual

Solution 8.

This is the solution to exercise 8!

Solution 9.

See? This solution gets printed where you have put it in the code of your document.

Fancy name 10.

See? This solution gets printed where you have put it and has a fancy name!

Solution 16.

This will be one of a few solutions printed by the following call of `\printsolutions`.

Solution 17.

The solution to the question with the nice ID.

Solution 18.

Yet another solution.

Solution 22.

To center a short part of the text body one can use the center environment (1 P.). Inside an environment like table one should use `\centering` (1 P.). For single lines there is also the `\centerline` command (1 P.).

Solution 23.

For example two volumes from *Computers & Typesetting*: *T_EXbook*, *T_EX*, *METAFONTbook*, *META-FONT*, *Computer Modern Typefaces*. Each valid answer is worth 1 P.

References

Solution 24.

Now, let's see... ah, yes:

- a) This is the first solution. Told you it was easy.
- b) This is the second solution. And of course you knew that!

Solution 25.

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

References

- [1] Jason Alexander. examdesign. Version 1.1, Mar. 26, 2001. URL: <http://www.ctan.org/pkg/examdesign/>.
- [2] Javier Bezos. enumitem. Version 3.5.2, Sept. 28, 2011. URL: <http://www.ctan.org/pkg/enumitem/>.
- [3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. 1st ed. Reading, Mass.: Addison-Wesley, 1994. 528 pp.
- [4] Philip Hirschhorn. exam. Version 2.4, May 22, 2011. URL: <http://www.ctan.org/pkg/exam/>.
- [5] Dennis Kletzing. multienum. Version n.n. May 19, 2005. URL: <http://www.ctan.org/pkg/multienum/>.
- [6] Donald E. Knuth. *Computers & Typesetting*. 5 vols. Reading, Mass.: Addison-Wesley, 1984–1986.
- [7] Donald E. Knuth. *Computers & Typesetting*. Vol. A: *The T_EXbook*. Reading, Mass.: Addison-Wesley, 1984.
- [8] Donald E. Knuth. *Computers & Typesetting*. Vol. B: *T_EX: The Program*. Reading, Mass.: Addison-Wesley, 1986.
- [9] Donald E. Knuth. *Computers & Typesetting*. Vol. C: *The METAFONTbook*. Reading, Mass.: Addison-Wesley, 1986.
- [10] Donald E. Knuth. *Computers & Typesetting*. Vol. D: *METAFONT: The Program*. Reading, Mass.: Addison-Wesley, 1986.
- [11] Donald E. Knuth. *Computers & Typesetting*. Vol. E: *Computer Modern Typefaces*. Reading, Mass.: Addison-Wesley, 1986.
- [12] Paul Pichaureau. exercise. Version 1.58, May 8, 2012. URL: <http://www.ctan.org/pkg/answers/>.
- [13] Mike Piff, *current maintainer*: Joseph Wright. answers. Version 2.13, Oct. 11, 2010. URL: <http://www.ctan.org/pkg/answers/>.
- [14] Nicola L. C. Talbot. probsoln. Version 3.02, Dec. 10, 2011. URL: <http://www.ctan.org/pkg/probsolns/>.

Index

A

<code>\addbonus</code>	11
<code>\AddCounterPattern</code>	46
Additional Information to Questions	
20 ff.	
<code>\addpoints</code>	11
<code>after-skip</code>	26
<code>all</code>	23
<code>\Alph</code>	10, 26, 46, 48
<code>\alph</code>	10, 26, 48
answers.....	3
<code>\arabic</code>	10, 26, 48

B

<code>babel</code>	10, 44 f.
<code>\baselineskip</code>	26
<code>beamer</code>	44
<code>before-skip</code>	26
<code>\bfseries</code>	5
<code>blank</code>	
<code>style</code>	30
<code>width</code>	30
<code>\blank</code>	30, 50
Blanks	29 ff.
<code>\bonus</code>	11
<code>bonus-format</code>	13
<code>bonus-name</code>	12
<code>bonus-plural</code>	13
<code>\bonussum</code>	11, 13

C

<code>\C</code>	17 f.
<code>class</code>	8
Classes	19 f.
CNTFORMATS	46–49
Example.....	46
Motivation.....	46
Predefined Patterns.....	48
Usage.....	46
<code>color</code>	5
<code>counter-format</code>	5, 10, 21
<code>counter-within</code>	5

Counting Points	11 ff.
options.....	12
<code>\currentbonussum</code>	12
<code>\currentpointssum</code>	11
Custom Configurations	43

D

<code>\DebugExSheets</code>	18
<code>\DeclareLanguage</code>	44 f.
<code>\DeclareLanguageAlias</code>	44
<code>\DeclareQuestionProperty</code>	20
<code>\DeclareRelGrades</code>	14
<code>\DeclareTranslation</code>	44

E

<code>enumitem</code>	3
<code>environ</code>	3
Environments	
<code>question</code>	6
<code>solution</code>	8
<code>tasks</code>	24
<code>\eSaveCounterPattern</code>	47
<code>\eSaveCounterPatternFrom</code>	47
<code>etoolbox</code>	3
<code>exam</code>	3
<code>examdesign</code>	3
<code>exercise</code>	3
<code>\exlabel</code>	18
EXSHEETS	4–43

G

<code>\GetQuestionProperty</code>	21
<code>\GetTranslation</code>	44
<code>\GetTranslationFor</code>	44
<code>\grade</code>	14
Grade Distribution	13 f.
<code>grades</code>	
<code>half</code>	14
<code>round</code>	14

H

<code>half</code>	14
-------------------------	----

INDEX

headings 5, 32
 headings-format 5
 hyperref 18

I

ID 8
 IDs 23
 include
 all 23
 IDs 23
 random 23
 seed 23
 \include 23
Include from External Files 22 ff.
 \includequestions 23
 \input 23

L

l3kernel 3
 l3keys2e 3
 l3packages 3
 l3sort 3, 19
 label 26
 \label 17 f.
 label-width 26
 \labelitemi 28
Language Settings 10 f.
 lcg 3, 23
 load-headings 5, 32
 load-tasks 6, 28
 \LoadDictionary 45
 \LoadDictionaryFor 45

M

mdframed 31
 multienum 3

N

name 7, 9, 11 f.
 name-plural 12
 \NewCounterPattern 46
 \NewPatternFormat 48
 \NewTranslation 44
 \normalem 3
 \normalsize 5

number-format 13

P

Package Options 5 f.

parse 13

points

 bonus-format 13
 bonus-name 12
 bonus-plural 13
 name 12
 name-plural 12
 number-format 13
 parse 13
 post-bonus 13
 pre-bonus 13
 separate-bonus 13
 use-name 13

\points 11

\pointssum 11, 13 f.

polyglossia 10

post-bonus 13

pre-bonus 13

Preliminary 3 f.

 Installation 4

 Licence 3

 Motivation 3

 Requirements 3

print 7 ff., 14 f.

\PrintIfIncludeActiveF 24

\PrintIfIncludeActiveT 24

\PrintIfIncludeActiveTF 24

\printsolutions 6, 15–18, 49

\PrintSolutionsF 15

\PrintSolutionsT 15

\PrintSolutionsTF 15

probsoln 3

Q

question

 class 8

 ID 8

 name 7

 print 7

 topic 8

INDEX

type.....	7	per section/chapter	16
use.....	8	solutions-toc-level.....	6
question (env.)	6	solutions-totoc	6
question/print	24	sorted.....	19
question/use.....	24	style	26, 30
\QuestionNumber.....	21	T	
questions	8	\task.....	29
options	7	tasks	
questions-toc-level.....	6	after-skip.....	26
questions-totoc	6	before-skip	26
R		label	26
random	23 f.	label-width	26
\ReadCounterFrom.....	46	style	26
\ReadCounterPattern.....	47	tasks	24–29
\ReadCounterPatternFrom.....	47	Custom Labels	29
\RenewTranslation	44	Options.....	26
\Roman	10, 26, 48	tasks (env.).....	24
\roman	10, 26, 46, 48	\textit.....	13
round.....	14	The ‘exsheets-headings’ Object ... 31–42	
S		The ‘exsheets-tasks’ Object	42 f.
\S.....	17 f.	toc-level	6
\SaveCounterPattern.....	47	topic.....	8
\SaveCounterPatternFrom.....	47	Topics	20
\SaveTranslation	45	\totalpoints.....	12 f.
\SaveTranslationFor.....	44	totoc.....	6
seed.....	23	\translate.....	44
separate-bonus.....	13	TRANSLATIONS	44 f.
\SetQuestionProperties.....	21	Motivation.....	44
Setting the Counter	10	Usage	44
Setup	4 f.	translator	44
\SetupExSheets.....	5	type.....	7
silence	3	U	
solution		ulem.....	3
name.....	9	use	8, 15
print	9, 14	use-classes	19 f., 24
sorted	19	use-name	13
solution (env.)	8	use-ref	6, 17 f.
solutions	8 ff.	use-topics	20, 24
options	9	V	
print.....	14	\value.....	17
all	15		
by ID	18		

INDEX

W

width 30

X

xcoeffs 31f.

xcolor 3

xparse 3

xtemplate 3, 5, 31