

The exercises package*

Roger Jud
rogerjud@outlook.com

May 17, 2020

Abstract

The package `exercises` defines the environments `exercise` and `solution`. They can be used to typeset exercises and solutions. The package allows you to customize the layout of those environments. The optional points in exercises can be added automatically. In addition you're able to hide the solutions.

Contents

1 Usage	2	4.3	Formatierung der Nummerierung	8
1.1 Basic example	2	4.4	Positionierung der Überschriften	9
1.2 Optional arguments	3	4.5	Positionierung der Punktzahlen	10
1.3 Sum of points	3	4.6	Formatierung der Punktzahlen	10
2 Options	4	4.7	Klammern formatieren . .	11
3 Special commands	5	4.8	Definition von Zählern . .	12
3.1 Sum of points	5	4.9	Umgebung <code>exercise</code> . . .	12
3.2 Special page breaks	6	4.10	Berechnung und Ausgabe der Gesamtpunktzahl . . .	13
3.3 Print text depending on whether the solution is printed or not	6	4.11	Umgebung <code>solution</code> . . .	14
4 Implementation	6	4.12	Befehle für die Seitenumbrüche	15
4.1 Benötigte Pakete	6	4.13	Befehl für die bedingte Textausgabe	15
4.2 Paket-Optionen definieren	7			

*This document corresponds to `exercises v1.1`, dated 2000/05/17.

1 Usage

Install the package and put `\usepackage{exercises}` in your preamble. Notice that `exercises` loads `verbatim`, `ifthen`, `kvoptions` and `xparse`. If you have already loaded these packages using `\usepackage{verbatim, ifthen, ...}`, you should unload these therefore.

Now you are able to use the environments `exercise` and `solution`. The two following sections show some code examples and the corresponding output. This is followed by some explanations and a list of all the options.

1.1 Basic example

The following example shows the basic usage of the two environments. There are two exercises and two solutions. Notice that the exercises and the solutions are automatically numbered.

It is important to put `printsolution=true` as an option of the package, because otherwise the solutions won't be printed. (The default value is `false`.)

```
\usepackage[printsolution=true]{exercises}

\begin{exercise}
  Calculate  $1 + 1 =$ 
\end{exercise}

\begin{solution}
  The result is  $1 + 1 = 2$ 
\end{solution}

\begin{exercise}
  Is this  $2 + 3 = 6$  true or false. Give a reason why.
\end{exercise}

\begin{solution}
  It is false, because  $2 + 3 = 5$ 
\end{solution}
```

produces

Exercise 1:

Calculate $1 + 1 =$

Solution 1:

The result is $1 + 1 = 2$

Exercise 2:

Is this $2 + 3 = 6$ true or false. Give a reason why.

Solution 2:

It is false, because $2 + 3 = 5$

1.2 Optional arguments

The exercise-environment has a optional argument – the points. The solution-environment has also a optional argument – the space which is left blank, if the solutions are hidden.

```
\usepackage[printsolution=true]{exercises}

\begin{exercise}[2]
  Calculate  $1 + 1 =$ 
\end{exercise}

\begin{solution}[3cm]
  The result is  $1 + 1 = 2$ 
\end{solution}

\begin{exercise}[2]
  Is this  $2 + 3 = 6$  true or false. Give a reason why.
\end{exercise}

\begin{solution}[2cm]
  It is false, because  $2 + 3 = 5$ 
\end{solution}
```

produces

Exercise 3: (1 Pt.)

Calculate $1 + 1 =$

Solution 3:

The result is $1 + 1 = 2$

Exercise 4: (2 Pt.)

Is this $2 + 3 = 6$ true or false. Give a reason why.

Solution 4:

It is false, because $2 + 3 = 5$

1.3 Sum of points

The points of all exercises are added. The command `\totalpoints` can be used to print the sum of the points. The word Pt. will be added automatically. You can customize it using the package option `totalpointsname=<string>`. (See also the following section.)

```
Sum of points: \totalpoints
```

produces

Sum of points: 3 Pt.

2 Options

This package uses the key-value-format for the package options. You can choose them while loading the package.

```
\usepackage[option]{exercises}
```

The following options are available:

- `printsolution=boolean`:
 - `printsolution=false` (*default*) The content between `\begin{solution}` and `\end{solution}` is not printed.
 - `printsolution=true` The solutions are printed.
- `exercisepaceabove=length` Space above the exercise-environment.
default value: 2.0 ex plus 1.0 ex minus 0.5 ex
- `exercisepacebelow=length` Space below the exercise-environment.
default value: 1.5 ex plus 1.0 ex minus 0.5 ex
- `solutionspaceabove=length` Space above the solution-environment.
default value: 2.0 ex plus 1.0 ex minus 0.5 ex
- `solutionspacebelow=length` Space below the solution-environment.
default value: 1.5 ex plus 1.0 ex minus 0.5 ex
- `blockspace=length` Space below exercisename respectively solutionname, if `exercisenameposition` respectively `solutionnameposition` is set to `block`.
default value: 0.25 ex
- `inlinespace=length` Space after exercisename respectively solutionname, if `exercisenameposition` respectively `solutionnameposition` is set to `inline`.
default value: 0.5 em
- `exercisenumformat=string` It defines the number format for the numbering of the exercises and the solutions. You can choose from the following:
 - `exercisenumformat=arabic` (*default*) arabic numbers
 - `exercisenumformat=roman` small roman numbers
 - `exercisenumformat=Roman` large roman numbers
 - `exercisenumformat=alph` small latin letters
 - `exercisenumformat=Alph` large latin letters
- `pointspostion=string` If you use points for the exercise-Environment, you can tell the package where to put them. You can choose from the following:
 - `pointspostion=aftername` (*default*) after the double point (Exercise 1: (3 Pt.))
 - `pointspostion=margin` at the right margin
- `pointssize=string` You can choose between two fontsizes for the points:
 - `pointssize=normalsize` (*default*) normal fontsize (like `\normalsize`)

- `pointsfontsize=footnotesize` smaller fontsize (like `\footnotesize`)
- `pointsname=<string>` You can select which text is displayed after the points of an exercise. (*default: Pt.*)
- `exercisename=<string>` You can select which text is displayed at the beginning of an exercise. (*default: Exercise*)
- `exercisenameposition=<string>` There are two ways to position the `exercisename`:
 - `exercisenameposition=block` (*default*) There is a line break after `exercisename` and optionally an additional vertical space (`blockspace`).
 - `exercisenameposition=inline` There is only a horizontal space (`inlinespace`) after `exercisename`.
- `solutionname=<string>` You can select which text is displayed at the beginning of a solution. (*default: Solution*)
- `solutionnameposition=<string>` There are two ways to position the `solutionname`:
 - `solutionnameposition=block`: (*default*) There is a line break after `solutionname` and optionally an additional vertical space (`blockspace`).
 - `solutionnameposition=inline` There is only a horizontal space (`inlinespace`) after `solutionname`.
- `totalpointsname=<string>` You can select which text is displayed after the points using the command `\totalpoints`. (*default: Pt.*)
- `braces=<string>` You can choose between three types of braces embracing the points of an exercise:
 - `braces=nobraces` no braces
 - `braces=round` (*default*) round braces like (1 Pt.)
 - `braces=square` square braces like [1 Pt.]
- `starpoints=<boolean>`:
 - `starpoints=false` (*default*) The points are printed as normal arabic numbers.
 - `starpoints=true` Stars (for example **** for five points) are printed instead of numbers.

3 Special commands

3.1 Sum of points

You can print the sum of points anywhere in the document using `\totalpoints`. \TeX needs two runs to calculate the sum.

3.2 Special page breaks

Custom page breaks using `\newpage` will move depending on whether or not to print the solutions. Because of that two new commands are available.

`\exercisewpage` inserts only a page break if `printsolution=false`, so if the solutions are *not* printed. `\solutionwpage` – on the other hand – inserts a page break if `printsolution=true`, so if the solutions are printed.

3.3 Print text depending on whether the solution is printed or not

With the command `\ifsolutionthenelse{ }{ }` you can print text depending on whether the solution is printed or not.

```
\usepackage[printsolution=true]{exercises}
\ifsolutionthenelse{%
  \subsection*{This is the header for the solutions}%
}%
{%
  \subsection*{This is the header for the exercises}%
}
```

produces

This is the header for the solutions

while

```
\usepackage[printsolution=false]{exercises}
\ifsolutionthenelse{%
  \subsection*{This is the header for the solutions}%
}%
{%
  \subsection*{This is the header for the exercises}%
}
```

produces

This is the header for the exercises

4 Implementation – how I’ve done it (only in German)

4.1 Benötigte Pakete

Zuerst lade ich alle benötigten Pakete. Das Paket `verbatim` benutze ich, um die Lösungen mit den Befehlen `\comment` und `\endcomment` auszukommentieren.

```
8 \RequirePackage{verbatim}
9 %
```

Das Paket `ifthen` hilft mit Befehlen wie `\ifthenelse`, `\equal`, `\boolean` und anderen.

```
10 \RequirePackage{ifthen}
11 %
```

Um eine Komma-separierte Liste von $\langle key \rangle = \langle value \rangle$ -Optionen erstellen zu können, benötige ich das Paket kvoptions.

```
12 \RequirePackage{kvoptions}
13 %
```

Das Paket xparse erlaubt mir das erstellen von Befehlen und Umgebungen mit mehreren (optionalen) Argumenten.

```
14 \RequirePackage{xparse}
15 %
```

Das Paket marginnote erlaubt mir die Erstellung von Bemerkungen am Rand. Ich benutze dieses Pakte anstelle des Befehls marginpar, weil marginpar die Notiz manchmal auf einer Seite platziert. (Paket-Warnung: marginpar on page xy moved.)

```
16 \RequirePackage{marginnote}
17 %
```

In einem ersten Schritt weise ich den Wert $\langle family \rangle$ (am besten der Name des Pakets) sowie $\langle prefix \rangle$ (das Präfix, welches vor jeden Variablennamen gesetzt wird) zu. Weitere Informationen können der Dokumentation von kvoptions entnommen werden.

```
18 \SetupKeyvalOptions{
19   family=exercises,
20   prefix=exercises@}
21 %
```

4.2 Paket-Optionen definieren

Jetzt definiere und initialisiere ich alle Paket-Optionen. Wenn der Benutzer keine Option wählt, wir der Standard-Wert (in eckigen Klammern) benutzt. Dann benutze ich den Befehl \ProcessKeyvalOptions um die Optionen dem Paket exercises zugänglich zu machen.

```
22 \DeclareBoolOption{printsolution}
23 \DeclareStringOption[2.0ex plus 1.0ex minus 0.5ex]{exercisesspaceabove}
24 \DeclareStringOption[1.5ex plus 1.0ex minus 0.5ex]{exercisesspacebelow}
25 \DeclareStringOption[2.0ex plus 1.0ex minus 0.5ex]{solutionspaceabove}
26 \DeclareStringOption[1.5ex plus 1.0ex minus 0.5ex]{solutionspacebelow}
27 \DeclareStringOption[0.25ex]{blockspace}
28 \DeclareStringOption[0.5em]{inlinespace}
29 \DeclareStringOption[arabic]{exercisenumbersformat}
30 \DeclareStringOption[aftername]{pointsposition}
31 \DeclareStringOption[normalsize]{pointsfontsize}
32 \DeclareStringOption[Pt.]{pointsname}
33 \DeclareStringOption[Exercise]{exercisename}
34 \DeclareStringOption[block]{exercisenameposition}
35 \DeclareStringOption[Solution]{solutionname}
36 \DeclareStringOption[block]{solutionnameposition}
37 \DeclareStringOption[Pt.]{totalpointsname}
```

```

38 \DeclareStringOption[round]{braces}
39 \DeclareBoolOption[false]{starpoints}
40 \ProcessKeyvalOptions{exercises}
41 %

```

4.3 Formatierung der Nummerierung

`\exercises@formatnumbers` Ich definiere einen Befehl um die Nummerierung der Übungen zu formatieren.

Wird für die Option `\exercisenumberformat` `\arabic` verwendet, so wird `\exercises@formatnumbers` so definiert, dass die Nummerierung als arabische Ziffern ausgegeben wird.

```

42 \ifthenelse{\equal{\exercises@exercisenumberformat}{arabic}}%
43   {\newcommand{\exercises@formatnumbers}[1]{\arabic{#1}}}{}%
44 %

```

Wird für die Option `\exercisenumberformat` `\roman` verwendet, so wird `\exercises@formatnumbers` so definiert, dass die Nummerierung als kleine römische Ziffern ausgegeben wird.

```

45 \ifthenelse{\equal{\exercises@exercisenumberformat}{roman}}%
46   {\newcommand{\exercises@formatnumbers}[1]{\roman{#1}}}{}%
47 %

```

Wird für die Option `\exercisenumberformat` `\Roman` verwendet, so wird `\exercises@formatnumbers` so definiert, dass die Nummerierung als grosse römische Ziffern ausgegeben wird.

```

48 \ifthenelse{\equal{\exercises@exercisenumberformat}{Roman}}%
49   {\newcommand{\exercises@formatnumbers}[1]{\Roman{#1}}}{}%
50 %

```

Wird für die Option `\exercisenumberformat` `\alph` verwendet, so wird `\exercises@formatnumbers` so definiert, dass die Nummerierung als Buchstaben ausgegeben wird.

```

51 \ifthenelse{\equal{\exercises@exercisenumberformat}{alph}}%
52   {\newcommand{\exercises@formatnumbers}[1]{\alph{#1}}}{}%
53 %

```

Wird für die Option `\exercisenumberformat` `\Alph` verwendet, so wird `\exercises@formatnumbers` so definiert, dass die Nummerierung als grosse Buchstaben ausgegeben wird.

```

54 \ifthenelse{\equal{\exercises@exercisenumberformat}{Alph}}%
55   {\newcommand{\exercises@formatnumbers}[1]{\Alph{#1}}}{}%
56 %

```

Wird für die Option `\exercisenumberformat` weder `\arabic` noch `\roman` noch `\Roman` noch `\alph` noch `\Alph` verwendet, so wird eine Fehlermeldung ausgegeben und `\exercisenumberformat` auf `\arabic` gesetzt.

```

57 \ifthenelse{\not\equal{\exercises@exercisenumberformat}{arabic}%
58   \and \not\equal{\exercises@exercisenumberformat}{roman}%
59   \and \not\equal{\exercises@exercisenumberformat}{Roman}%
60   \and \not\equal{\exercises@exercisenumberformat}{alph}%
61   \and \not\equal{\exercises@exercisenumberformat}{Alph}}%
62   {\PackageError{exercises}%
63     {"\exercises@exercisenumberformat" ist als Option von "exercisenumberformat"
nicht erlaubt. %

```



```

64     Stattdessen wurde "arabic" verwendet.}%
65     {"\exercises@exercisenumbersformat" ist als Option von "exercisenumbersformat"
nicht erlaubt. }%
66     Stattdessen wurde "arabic" verwendet.}%
67     \newcommand{\exercises@formatnumbers}[1]{\arabic{#1}}}%
68 %

```

4.4 Positionierung der Überschriften

```

69 \ifthenelse{\equal{\exercises@exercisenameposition}{block}}%
70   {\newcommand{\exercises@placeexercise}{\noindent\par\vspace{\
exercises@blockspace}}}%
71 %

```

```

72 \ifthenelse{\equal{\exercises@exercisenameposition}{inline}}%
73   {\newcommand{\exercises@placeexercise}{\hspace{\exercises@inlinespace}}}%
74 %

```

```

75 \ifthenelse{\not\equal{\exercises@exercisenameposition}{block}%
76   \and \not\equal{\exercises@exercisenameposition}{inline}}%
77   {\PackageError{exercises}%
78     {"\exercises@exercisenameposition" ist als Option von "exercisenameposition"
nicht erlaubt. }%
79     Stattdessen wurde "block" verwendet.}%
80     {"\exercises@exercisenameposition" ist als Option von "exercisenameposition"
nicht erlaubt. }%
81     Stattdessen wurde "block" verwendet.}%
82     \newcommand{\exercises@placeexercise}{\noindent\par\vspace{\
exercises@blockspace}}}%
83 %

```

```

84 \ifthenelse{\equal{\exercises@solutionnameposition}{block}}%
85   {\newcommand{\exercises@placesolution}{\noindent\par\vspace{\
exercises@blockspace}}}%
86 %

```

```

87 \ifthenelse{\equal{\exercises@solutionnameposition}{inline}}%
88   {\newcommand{\exercises@placesolution}{\hspace{\exercises@inlinespace}}}%
89 %

```

```

90 \ifthenelse{\not\equal{\exercises@solutionnameposition}{block}%
91   \and \not\equal{\exercises@solutionnameposition}{inline}}%
92   {\PackageError{exercises}%
93     {"\exercises@solutionnameposition" ist als Option von "solutionnameposition"
nicht erlaubt. }%
94     Stattdessen wurde "block" verwendet.}%
95     {"\exercises@solutionnameposition" ist als Option von "solutionnameposition"
nicht erlaubt. }%
96     Stattdessen wurde "block" verwendet.}%
97     \newcommand{\exercises@placesolution}{\noindent\par\vspace{\
exercises@blockspace}}}%
98 %

```

4.5 Positionierung der Punktzahlen

`\exercis@placepoints` Nun werden die Befehle zur Positionierung der Punkte (falls vorhanden) definiert. Falls für die Option `\pointsplace` keine der Werte `\aftername` (Position der Punkte rechts des Doppelpunktes nach *Exercise*) oder `\margin` (am rechten Rand) verwendet wird, gibt das Paket eine Fehlermeldung aus und verwendet die Option `\aftername`.

Wird für die Option `\aftername` verwendet, so wird `\exercis@placepoints` so definiert, dass lediglich die Punktzahl ausgegeben wird.

```
99 \ifthenelse{\equal{\exercis@pointsposition}{aftername}}{%
100   \newcommand{\exercis@placepoints}[1]{ #1}}{%
101   %
```

Wird für die Option `\margin` verwendet, so wird `\exercis@placepoints` so definiert, dass die Punktzahl mit Hilfe von `\marginnote` am rechten Rand ausgegeben wird.

```
102 \ifthenelse{\equal{\exercis@pointsposition}{margin}}{%
103   \newcommand{\exercis@placepoints}[1]{%
104     \marginnote{#1}}}%
105   %
```

Wird für die Option `\pointsposition` weder `\aftername` noch `\margin` verwendet, so wird eine Fehlermeldung ausgegeben und `\pointsposition` auf `\aftername` gesetzt.

```
106 \ifthenelse{\not\equal{\exercis@pointsposition}{aftername}%
107   \and \not\equal{\exercis@pointsposition}{margin}}{%
108   {\PackageError{exercis}%
109     {"\exercis@pointsposition" ist als Option von "pointsposition" nicht erlaubt.
110     %
111     Stattdessen wurde "aftername" verwendet.}%
112     {"\exercis@pointsposition" ist als Option von "pointsposition" nicht erlaubt.
113     %
114     Stattdessen wurde "aftername" verwendet.}%
115     \newcommand{\exercis@placepoints}[1]{ #1}}}%
116   %
```

4.6 Formatierung der Punktzahlen

```
115 \ifthenelse{\equal{\exercis@pointsfontsize}{normalsize}}{%
116   \newcommand{\exercis@formatpoints}[1]{\normalsize #1}}{%
117   %
```

```
118 \ifthenelse{\equal{\exercis@pointsfontsize}{footnotesize}}{%
119   \newcommand{\exercis@formatpoints}[1]{\footnotesize #1 \normalsize}}{%
120   %
```

```
121 \ifthenelse{\not\equal{\exercis@pointsfontsize}{normalsize}%
122   \and \not\equal{\exercis@pointsfontsize}{footnotesize}}{%
123   {\PackageError{exercis}%
124     {"\exercis@pointsfontsize" ist als Option von "pointsfontsize" nicht erlaubt.
125     %
126     Stattdessen wurde "normalsize" verwendet.}%
```

```

126 {"\exercises@pointsfontsize" ist als Option von "pointsfontsize" nicht erlaubt.
    %
127   Stattdessen wurde "normalsize" verwendet.}%
128   \newcommand{\exercises@formatpoints}[1]{\normalsize #1}}{}%
129 %

```

4.7 Klammern formatieren

`\exercises@braces` Nun werden die Befehle zur Ausgabe der Klammern (falls vorhanden) definiert. Falls für die Option `<braces>` keine der Werte `<nobraces>`, `<round>` oder `<square>` verwendet wird, gibt das Paket eine Fehlermeldung aus und verwendet die Option `<round>`.

Wird für die Option `<braces>` `<nobraces>` verwendet, so wird `\exercises@braces` so definiert, dass keine Klammern ausgegeben werden.

```

130 \ifthenelse{\equal{\exercises@braces}{nobraces}}%
131   {\newcommand{\exercises@openbrace}{}%
132   \newcommand{\exercises@closebrace}{}{}}%
133 %

```

Wird für die Option `<braces>` `<round>` verwendet, so wird `\exercises@braces` so definiert, dass runde Klammern ausgegeben werden.

```

134 \ifthenelse{\equal{\exercises@braces}{round}}%
135   {\newcommand{\exercises@openbrace}{(%
136   \newcommand{\exercises@closebrace}{)%}}%
137 %

```

Wird für die Option `<braces>` `<square>` verwendet, so wird `\exercises@braces` so definiert, dass eckige Klammern ausgegeben werden.

```

138 \ifthenelse{\equal{\exercises@braces}{square}}%
139   {\newcommand{\exercises@openbrace}{[%
140   \newcommand{\exercises@closebrace}{]%}}%
141 %

```

Wird für die Option `<braces>` weder `<nobraces>` noch `<round>` noch `<square>` verwendet, so wird eine Fehlermeldung ausgegeben und `<braces>` auf `<round>` gesetzt.

```

142 \ifthenelse{\not\equal{\exercises@braces}{nobraces}}%
143   \and \not\equal{\exercises@braces}{round}}%
144   \and \not\equal{\exercises@braces}{square}}%
145   {%
146   \PackageError{exercises}%
147     {"\exercises@braces" ist als Option von "braces" nicht erlaubt. %
148     Stattdessen wurde "round" verwendet.}%
149     {"\exercises@braces" ist als Option von "braces" nicht erlaubt. %
150     Stattdessen wurde "round" verwendet.}%
151     \newcommand{\exercises@openbrace}{(%
152     \newcommand{\exercises@closebrace}{)%}
153   }{}%
154 %

```

4.8 Definition von Zählern

Nun werden die Zähler zur Nummerierung der Aufgaben ($\langle exercises@exercisenummer \rangle$) und Lösungen ($\langle exercises@solutionnummer \rangle$) sowie zur Zählung der Punkte ($\langle exercises@totalpoints \rangle$ und $\langle exercises@finaltotalpoints \rangle$) definiert und initialisiert.

```
155 \newcounter{exercises@totalpoints}
156 \setcounter{exercises@totalpoints}{0}
157 \newcounter{exercises@finaltotalpoints}
158 \newcounter{exercises@exercisenummer}
159 \setcounter{exercises@exercisenummer}{1}
160 \newcounter{exercises@solutionnummer}
161 \setcounter{exercises@solutionnummer}{1}
162 \newcounter{i}
163 \newlength{\exercises@parskipsave}
164 %
```

4.9 Umgebung exercise

exercise Der eigentliche Code folgt nun. Die Umgebung `exercise` definiert ein optionales Argument (die Punktzahl).

```
165 \DeclareDocumentEnvironment{exercise}{o}{%
166 \setlength{\exercises@parskipsave}{0cm}
167 \setlength{\exercises@parskipsave}{\parskip}
168 \setlength{\parskip}{0cm}
169 %
```

Der obere Abstand wird eingefügt.

```
170 \vspace{\exercises@exercisespaceabove}%
171 %
```

Ist das optionale Argument leer (`\IfNoValueTF` also wahr), wird lediglich $\langle exercisename \rangle$ (standarmässig Exercise) sowie $\langle exercisenummer \rangle$ ausgegeben.

```
172 \IfNoValueTF{#1}%
173 {\textbf{\exercises@exercisename\ \exercises@formatnumbers{
exercises@exercisenummer}:}\exercises@placeexercise}%
174 %
```

Ist das optionale Argument vorhanden (`\IfNoValueTF` also nicht wahr), wird zuerst $\langle totalpoints \rangle$ um die Punktzahl erhöht und anschliessend $\langle exercisename \rangle$ sowie $\langle exercisenummer \rangle$ ausgegeben.

```
175 {\addtocounter{exercises@totalpoints}{#1}%
176 \textbf{\exercises@exercisename\ \exercises@formatnumbers{
exercises@exercisenummer}:}%
177 %
```

Falls $\langle starpoints \rangle$ wahr ist, wird eine Schleife zur Ausgabe der Sterne – entsprechend der Anzahl Punkt – gestartet.

```

178 \ifthenelse{\boolean{exercises@starpoints}}%
179   {\exercises@placepoints{\exercises@openbrace%
180     \setcounter{i}{0}\whiledo{#1>\thei}{*\stepcounter{i}}%
181     \exercises@closebrace}}%
182 %

```

Ist $\langle starpoints \rangle$ nicht wahr ist, wird die Punktzahl ausgegeben.

```

183   {\exercises@placepoints{\exercises@formatpoints{\exercises@openbrace#1 \
exercises@pointsname\exercises@closebrace}}\exercises@placeexercise%
184   }%
185 \setlength{\parskip}{\exercises@parskipsave}%
186 }{%
187 \setlength{\parskip}{0cm}%
188 %

```

Der Zähler $\langle exercisenummer \rangle$ wird um 1 hochgezählt und der untere Abstand wird eingefügt.

```

189 \stepcounter{exercises@exercisenummer}%
190 \vspace{\exercises@exercisepacebelow}%
191 \setlength{\parskip}{\exercises@parskipsave}%
192 }%
193 %

```

4.10 Berechnung und Ausgabe der Gesamtpunktzahl

`\totalpoints` Der Befehle `\totalpoints` zur Ausgabe der Gesamtpunktzahl im Dokument wird definiert. Es wird eine Referenz auf den Zähler $\langle exercises@finaltotalpoints \rangle$ erstellt, welcher am Ende des Dokuments mit Hilfe von `\AtEndDocument` in die aux-Datei geschrieben wird.

```

194 \newcommand\totalpoints{\theexercises@finaltotalpoints\ \
exercises@totalpointsname}%
195 %

```

`\immediate\write\@auxout` sorgt dafür, dass der String `\setcounter{exercises@finaltotalpoints}{\theexercises@totalpoints}` in die aux-Datei geschrieben wird. So wird dem Zähler $\langle exercises@finaltotalpoints \rangle$ der Wert von $\langle exercises@totalpoints \rangle$ übergeben. Die direkte Speicherung von $\langle exercises@totalpoints \rangle$ in der aux-Datei ist nicht möglich, denn dann würde bei einem \LaTeX Lauf folgendes passieren:

1. Das Paket `exercises` wird geladen und $\langle exercises@totalpoints \rangle$ wird mit 0 initialisiert.
2. Bei `\begin{document}` wird die aux-Datei gelesen. Beim ersten Lauf ändert sich $\langle exercises@totalpoints \rangle$ nicht.
3. Zu $\langle exercises@totalpoints \rangle$ werden die Punktzahlen der einzelnen Aufgaben addiert.
4. Am Ende des Dokumentes wird mit `\AtEndDocument` bzw. `\immediate\write\@auxout` der aktuelle Wert von $\langle exercises@totalpoints \rangle$ in die aux-Datei geschrieben.

5. Beim zweiten \LaTeX -Lauf würde wiederum das Paket `exercises` geladen und $\langle exercises@totalpoints \rangle$ mit 0 initialisiert werden.
6. Bei `\begin{document}` wird wiederum die aux-Datei gelesen. Beim diesem zweiten \LaTeX -Lauf würde $\langle exercises@totalpoints \rangle$ nun den Wert der Gesamtpunktzahl aus dem ersten Lauf bekommen.
7. Zu $\langle exercises@totalpoints \rangle$ werden die Punktzahlen der einzelnen Aufgaben addiert.
8. usw.

Man erkennt, dass sich so die Gesamtpunktzahl bei jedem \LaTeX -Lauf verdoppeln, verdreifachen usw. würde. Entsprechend wird $\langle exercises@totalpoints \rangle$ jeweils am Schluss des Dokuments an $\langle exercises@finaltotalpoints \rangle$ übergeben.

```

196 \AtEndDocument{\immediate\write\@auxout%
197 {\string\setcounter{exercises@finaltotalpoints}{\theexercises@totalpoints}}}%
198 %

```

4.11 Umgebung `solution`

solution Die Umgebung `solution` definiert ein optionales Argument: der Abstand, welcher freigelassen werden soll, falls die Lösung nicht ausgegeben wird.

```

199 \DeclareDocumentEnvironment{solution}{0}{%
200 \setlength{\exerciseparskip}{0cm}
201 \setlength{\exerciseparskip}{\parskip}
202 \setlength{\parskip}{0cm}
203 %

```

Der obere Abstand wird eingefügt, falls die Lösungen ausgegeben werden.

```

204 \ifthenelse{\boolean{exercises@printsolution}}{%
205   {\vspace{\exerciseparskip}}}%
206 %

```

$\langle solutionname \rangle$ und die Lösungsnummer werden ausgegeben, sofern die Lösungen überhaupt ausgegeben. Andernfalls wird der Leerraum (das optionale Argument) sowie `\comment` ausgegeben.

```

207 \ifthenelse{\boolean{exercises@printsolution}}{%
208   {\textbf{\exerciseparskip\ \exerciseparskip\ \exerciseparskip\
exerciseparskip:\exerciseparskip}}}%
209   {\IfNoValueTF{#1}{\vspace{#1}}{\comment}}%
210 \setlength{\parskip}{\exerciseparskip}%
211 }%
212 %

```

Werden die Lösungen ausgegeben, so wird $\langle solutionspacebelow \rangle$ ausgegeben. Anders falls folgt die Ausgabe von `\endcomment`. Zu guter Letzt wird der Zähler $\langle solutionnumber \rangle$ eins hochgezählt.

```

213 {%
214 \setlength{\parskip}{0cm}%
215 \ifthenelse{\boolean{exercises@printsolution}}{\vspace{\
exercises@solutionspacebelow}}%
216 {\endcomment}\stepcounter{exercises@solutionnumber}\setlength{\parskip}{\
exercises@parskipsave}%
217 \setlength{\parskip}{\exercises@parskipsave}%
218 }%
219 %

```

4.12 Befehle für die Seitenumbrüche

`\exercisewpage` Die Befehle `exercisewpage` und `solutionwpage` erlauben es, einen Seitenumbruch in Abhängigkeit der Lösungsausgabe einzufügen.

```

220 \newcommand{\exercisewpage}{\ifthenelse{\boolean{exercises@printsolution}}{\
newpage}}
221 %
222 \newcommand{\solutionwpage}{\ifthenelse{\boolean{exercises@printsolution}}{\
newpage}}
223 %

```

4.13 Befehl für die bedingte Textausgabe

`\ifsolutionthenelse` Der Befehl `ifsolutionthenelse{}{}` erlaubt es, Text in Abhängigkeit der Lösungsausgabe auszugeben.

```

224 \newcommand{\ifsolutionthenelse}[2]{\ifthenelse{\boolean{exercises@printsolution
}}{#1}{#2}}
225 %

```

Change History

v1.0	v1.1
General: Initial version 1	General: Add the command ifsolutionthenelse 1