

The exercisepoints Package*

2019/01/03 v1.2.3

Henning Kerstan

The *exercisepoints* package provides an exercise environment and several macros to count points for exercises. The actual typesetting of an exercise can be defined by the user.

1 Introduction and Basic Usage

This package can be used to facilitate exercise counting and exercise point counting in a $\text{\LaTeX}$ -document. It counts the number of exercises (for example, it counted that this document has 5 exercises) and it sums all the points of all exercises (for example, the package determined that the exercises in this document have a total of 22.5 points).

Especially for exams it is also common to have an overview of all exercises and their maximal points. This is also supported by this package by providing a macro to retrieve the points of each exercise. For example, the exercise overview for this document is given in Figure 1 below.

Exercise	Points	Max. Points
1		2.5
2		4.5
3		4.5
4		4
5		7
Sum		22.5
Bonus		10
Sum with Bonus		32.5

Figure 1: Overview of all exercises

This package only provides functionality for counting and adding points, the actual typesetting of the exercise can be entirely determined by the user.

1.1 Load the Package

To use this package type

```
\usepackage{exercisepoints}
```

anywhere in your document's preamble. If this does not work because the package is not contained in your $\text{\LaTeX}$ -distribution, simply copy the file 'exercisepoints.sty' into the same folder as your document (or, if you plan to use it more often, copy it into your local texmf directory).

1.2 Typeset a First Exercise

Now you can typeset your first exercise using the following code.

```
\begin{exercise}[Simple Addition]
  What is  $1+1$ ? \points{2.5}
\end{exercise}
```

If you typeset this without further modifications it will yield the following result.

*<https://github.com/henningkerstan/exercisepoints>

Exercise 0: Simple Addition

(2.5 Points)

What is $1 + 1$?

As you can see, inside the code we used `\points{2.5}` to assign this exercise a total of 2.5 points. These points were printed flushed right in the exercise header.

1.3 Play with the Points

If you issue the `\points{...}` macro multiple times within an exercise environment, the points will add up. Thus let us consider the following code.

```
\begin{exercise}[Simple Equation]
  Determine a number  $x$  such that  $3 \cdot x = 15$  \points
    {1.5} and explain how you did that. \points{3}
\end{exercise}
```

This code will typeset as follows.

Exercise 1: Simple Equation

(4.5 Points)

Determine a number x such that $3 \cdot x = 15$ and explain how you did that.

As expected, the points were added to obtain the total $1.5 + 3 = 4.5$ points which again is printed in the exercise's header.

This feature is especially handy if you have an exercise with several subparts (items). In this case you can also use the `\itempoints{...}` macro which does the same as the `\points{...}` macro but additionally also prints the supplied points flush right. We rewrite our previous exercise with an enumerate environment as follows.

```
\begin{exercise}[Simple Equation]\vspace{-1.5em}
  \begin{enumerate}
    \item Determine  $x$  such that  $3 \cdot x = 15$ . \itempoints{1.5}
    \item Explain how you arrived at your solution in the
          previous part. \itempoints{3}
  \end{enumerate}
\end{exercise}
```

This code results in the following output.

Exercise 2: Simple Equation

(4.5 Points)

1. Determine x such that $3 \cdot x = 15$. (1.5)
2. Explain how you arrived at your solution in the previous part. (3)

Note that by default this only displays the number of points in parentheses. If you want to add a unit, you can use

```
\setitempointsunit{pt}{pts}
```

anywhere in your document which sets *pt* (singular form) and *pts* (plural form) as units for the `itempoints`. Afterwards, all `itempoints` will display this unit right of the number.

In fact, if we do this and assign the first part of the above exercise just one point instead of 1.5 (replacing `\itempoints{1.5}` by `\itempoints{1}`), it will be displayed as follows.

Exercise 3: Simple Equation (4 Points)

1. Determine x such that $3 \cdot x = 15$. (1 pt)
2. Explain how you arrived at your solution in the previous part. (3 pts)

1.4 Obtain the Total Points

If you have typeset your exercises, there are a few commands to obtain the total number of points.

- ▷ `\numberofexercises` expands to the number of all exercises in the document.
- ▷ `\totalpoints` expands to the sum of all points of all exercises in the document.
- ▷ `\getpoints{x}` expands to the total points of the exercise number x (starting with 0).

Note that the information for these macros is retrieved from the .aux file so to get the right numbers here, you have to compile your document twice. As a simple example for the above macros we take the following code.

```
Exercise 0 has \getpoints{0} points, there are \
numberofexercises\ exercises with a total of \
totalpoints\ points.
```

This compiles to the following sentence.

Exercise 0 has 2.5 points, there are 5 exercises with a total of 22.5 points.

1.5 Use Bonus Points

In some courses it is possible to obtain a bonus for the exam. This is supported by providing the commands

```
\bonuspoints{x}
```

which will set the total amount of bonus points to x bonuspoints (in this document we have used it with $x = 10$) and

```
\getbonuspoints
```

which retrieves the current bonus points (in this document it expands to 10). Please note that the bonuspoint command differs from the `\points` and `\itempoint` commands in two aspects:

- ▷ Bonus points are *not additive*, i.e., if you use `\bonuspoints` multiple times in your document, only the last occurrence will set the bonus points.
- ▷ Bonus points are *not added* to the total points. However, there is an additional macro

```
\totalpointswithbonus
```

which will expand to the sum of `\totalpoints` and `\getbonuspoints`. In this document it thus expands to 32.5.

1.6 Subexercises

If you have an exercise with many different parts (for example, if you have one overall topic with several areas), you can use the subexercise environment to structure your exercise. Subexercises work exactly as exercises, but can only be used within an exercise environment. The following is a compiled example of such an exercise.

Exercise 4: Calculus (7 Points)

Exercise 4.0: Derivatives (4 Points)

Determine the derivatives of the following functions.

1. $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^2 + 2x + 3$ (1 pt)
2. $g: \mathbb{R} \rightarrow \mathbb{R}, g(x) = \exp(x^2)$ (3 pts)

Exercise 4.1: Maxima and Minima (3 Points)

Determine the local maximum or minimum of the function

$$f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^2 + 2x + 3.$$

It was typeset using the following code.

```
\begin{exercise}[Calculus]
\vspace{-1em}
\begin{subexercise}[Derivatives]
Determine the derivatives of the following functions.
\begin{enumerate}
\item  $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^2 + 2x + 3$  \itempoints{1}
\item  $g: \mathbb{R} \rightarrow \mathbb{R}, g(x) = \exp(x^2)$  \itempoints{3}
\end{enumerate}
\end{subexercise}\vspace{-1em}
\begin{subexercise}[Maxima and Minima]
Determine the local maximum or minimum of the function
 $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = x^2 + 2x + 3$ . \points{3}
\end{subexercise}
\end{exercise}
```

Note that it is only possible to use the subexercise environment within an exercise environment and it is not possible to nest subexercise environments.

2 Exercise Layout

This package only provides a minimal layout style for an exercise which is *not meant to be actually used*. It is just included to test the package functionality.

2.1 Disable the Built In Layout

If you define your own layout, you can load the package (but it is not necessary) with the `customlayout` option, i.e. by issuing the command

```
\usepackage[customlayout]{exercisepoints}
```

which disables the built in exercise style. Note that in this case your document will only compile exercises or subexercises if you have defined the respective layout. Otherwise compilation will stop with a package error message.

2.2 Define Your Own Layout

In order to define how your exercise looks like, you can use the two commands listed below.

```
\AtBeginExercise{%
% your code
}

\AtEndExercise{%
% your code
}
```

For example, the minimal example style is implemented using the following code.

```

\AtBeginExercise{%
  ~\smallskip\\\sffamily%
  \ifthenelse{\equal{\currentexercisetitle}{}}{%
    {\textbf{Exercise-\currentexercisenum}}%
    {%
      \textbf{Exercise-\currentexercisenum:} %
      \currentexercisetitle%
    }%
  },~\hfill\textbf{(\currentexercisepoints~Points)}%
  \smallskip\\\noindent\normalfont%
}
\AtEndExercise{~\smallskip\\}%

```

As you can see, there are three useful macros that can be used for defining the exercise layout:

- ▷ `\currentexercisetitle` expands to the exercise title supplied as (optional) argument to the exercise environment.
- ▷ `\currentexercisenum` expands to the current internal exercise number (starting at 0).
- ▷ `\currentexercisepoints` expands to the current exercise total points. Note that this is retrieved from the .aux file so to get the right number here, you have to compile your document twice.

For subexercises you have essentially the same commands, i.e., you have the commands

- ▷ `\AtBeginSubexercise{...}`,
- ▷ `\AtEndSubexercise{...}`,
- ▷ `\currentsubexercisenum`,
- ▷ `\currentsubexercisepoints` and
- ▷ `\currentsubexercisetitle`

which work as their counterparts for exercises.

2.3 Example Layout Using (Sub-)Sections

One option for a custom layout could be to use sections for exercises and subsections for subexercises. This can be achieved using the following code.

```

\usepackage[customlayout]{exercisepoints}
\AtBeginExercise{%
  \section{\texorpdfstring{%
    \currentexercisetitle\hfill%
    (\currentexercisepoints~Points)}}{%
    \currentexercisetitle (\currentexercisepoints~Points)}%
  }
}
\AtEndExercise{\clearpage}
\AtBeginSubexercise{%
  \subsection{\texorpdfstring{%
    \currentsubexercisetitle\hfill%
    (\currentsubexercisepoints~Points)}}{%
    \currentsubexercisetitle %
    (\currentsubexercisepoints~Points)}%
  }
}
\AtEndSubexercise{}

```

3 Copyright and License

Copyright © 2017–2019 Henning Kerstan.

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in

<http://www.latex-project.org/lppl.txt>

and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is Henning Kerstan.

This work consists of the files ‘exercisepoints.sty’ and ‘exercisepoints.tex’.

4 Version History

2019/01/03 v1.2.3 Revised README.md.

2019/01/02 v1.2.2 Slight revision of documentation.

2018/08/15 v1.2.1 Renamed the package from *hkexercise* to *exercisepoints* to comply with CTAN rules (see <https://ctan.org/file/help/ctan/CTAN-upload-addendum>)

2017/02/27 v1.2.0 Added the `\setitempointsunit{...}` command.

2017/07/02 v1.1.0 This version introduces subexercises. Moreover several checks were added to avoid use of commands like `\currentexercisepoints` outside of exercise environments and to prevent nesting of exercises. New option `customlayout` disables built in exercise layout. Added `\bonuspoint` commands.

2017/06/26 v1.0.1 Redefined `\currentexercisepoints` so it can be used within section command.

2017/06/26 v1.0 The initial version of this package is published on GitHub.