

# The homeworkassignment\*class<sup>†</sup>

Adrian C Hinrichs  
adrian.hinrichs@rwth-aachen.de

April 23, 2019

## Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Dependencies</b>	<b>2</b>
2.1	Mandatory Dependencies . . . . .	2
2.2	Recommended Dependencies . . . . .	3
<b>3</b>	<b>Options</b>	<b>4</b>
<b>4</b>	<b>Layout</b>	<b>8</b>
4.1	Headers & Footers . . . . .	8
4.2	Enhance Mathenvironments . . . . .	9
4.3	fonts . . . . .	9
4.4	Serife (Default) . . . . .	9
4.4.1	San Serife . . . . .	9
4.4.2	Monospace . . . . .	9
<b>5</b>	<b>Commands</b>	<b>10</b>
5.1	Constants . . . . .	10
5.2	Document Informations . . . . .	10
5.2.1	Inherited from <code>article</code> . . . . .	10
5.3	Sectioning . . . . .	10
5.3.1	internal Sectioning . . . . .	11
5.3.2	‘better’ Sectioning . . . . .	12
5.3.3	Even Better Sectioning-Environments . . . . .	14
5.4	Useful Macros . . . . .	15
5.4.1	QUOD ERAT DEMUNSTRANDUM, End of Proof . . . . .	15
5.4.2	QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST . . . . .	15

---

\*The name was changed with version v3.0, to become compatible with CTANs guidelines and to maintain a degree of backwards compatibility. The class was called `HomeworkAssignment` prior to v3.0

<sup>†</sup>This document corresponds to `homeworkassignment` v4.1, dated 2019/04/23.

5.4.3	Stolen Goods . . . . .	15
5.4.4	Rounding . . . . .	18
5.4.5	ToDoS . . . . .	18
5.5	Grading Table . . . . .	19
5.5.1	Internal commands . . . . .	20
5.6	Title . . . . .	20
5.7	Counters . . . . .	21
<b>6</b>	<b>Environments</b>	<b>22</b>
6.1	Proof . . . . .	22
6.2	Proof by contradiction . . . . .	22
<b>7</b>	<b>Development and support</b>	<b>23</b>
<b>8</b>	<b>Changelog</b>	<b>23</b>
8.1	Version-Scheme . . . . .	25
<b>9</b>	<b>Translations</b>	<b>26</b>

## 1 Abstract

This class provides a relative simple document-type for homework, mainly created for assignments at the University This class is inherited from `article`, it is not perfect, but I am trying my very best.

## 2 Dependencies

### 2.1 Mandatory Dependencies

This class is build upon `article`, so of course the first dependency is:

`article` 1992 LESLIE LAMPORT, 1994-97 FRANK MITTELBACH JOHANNES BRAAMS, THE L<sup>A</sup>T<sub>E</sub>X-TEAM, <https://www.ctan.org/pkg/kvoptions>,

Because I am very lazy, the `homeworkassignment` is “a little bit” bloated. These are all required packages:

`kvoptions` HEIKO OBERDIEK, <https://www.ctan.org/pkg/kvoptions>,  
for key=value-style options

`suffix` DAVID KASTRUP, <https://www.ctan.org/pkg/suffix>,  
Makes it easy to define `\macro*` commands

`xifthen` JOSSELIN NOIREL, <https://www.ctan.org/pkg/xifthen>,  
For if-else-structures

**translations** CLEMENS NIEDERBERGER, <https://www.ctan.org/pkg/translations>,  
Implements an easy method of translations.

**amsmath** THE L<sup>A</sup>T<sub>E</sub>X-TEAM, FRANK MITTELBACH RAINER SCHÖPF, ET AL.,  
<https://www.ctan.org/pkg/amsmath>,  
For better math-typesetting

**amssymb** AMERICAN MATHEMATICAL SOCIETY, [mirror.ctan.org/fonts/amssymb/doc/amssymb.pdf](https://mirror.ctan.org/fonts/amssymb/doc/amssymb.pdf),  
For more mathematical symbols

**etoolbox** PHILIPP LEHMAN (INACTIVE), JOSEPH WRIGHT, <https://www.ctan.org/pkg/etoolbox>,  
The package is a toolbox of programming facilities geared primarily towards L<sup>A</sup>T<sub>E</sub>Xclass and package authors

**array** FRANK MITTELBACH, DAVID CARLISLE, THE L<sup>A</sup>T<sub>E</sub>X-TEAM, <https://www.ctan.org/pkg/array>,  
A new implementations for tables and arrays

**xparse** FRANK MITTELBACH, CHRIS ROWLEY, DAVID CARLISLE, THE L<sup>A</sup>T<sub>E</sub>X3 PROJECT, <https://ctan.org/pkg/xparse>,  
The package provides a high-level interface for producing documentlevel commands. In that way, it offers a replacement for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s `\newcommand` macro, with significantly improved functionality.

**gillius** BOB TENNENT, <https://ctan.org/pkg/gillius>,  
A Gillian Sans inspired font, used for all sans serifes fonts

**hyperref** [HTTPS://CTAN.ORG/PKG/HYPERREF](https://CTAN.ORG/PKG/HYPERREF), SebastianRahtz, HeikoOberdiek,  
For hyperrefs, obviously

**xcolor** DR. UWE KERN, <https://www.ctan.org/pkg/xcolor>,  
For coloring of ToDos

## 2.2 Recommended Dependencies

These are not loaded automatically, but require a switch as option (see section 3). The switch is typically the name of the package.

**tikz** TILL TANTAU, MARK WIBROW, CHRISTIAN FEUERSÄNGER ET AL., <https://www.ctan.org/pkg/pgf>,  
An incredible powerfull image tool. When loading TikZ, the homeworkassignment automatically loads a shipload of TikZ-librarys and own styles.  
See section 3 for more informations

**listings** CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN, <https://www.ctan.org/pkg/listings>,  
For source-code. Sourcecode in the homeworkassignment is automatically framed, printed in `scriptsize`, and linebeals will be introduced

array possibly can be removed

I intend to move these styles to a own package, so that they are usable without the homeworkassignment

Loads required Packages

```

1 \RequirePackage{suffix}
2 \RequirePackage{fancyhdr}
3 \RequirePackage{xifthen}
4 \RequirePackage{translations}
5 \PassOptionsToPackage{fleqn}{amsmath}
6 \RequirePackage{amsmath}
7 \RequirePackage{amssymb}
8 \RequirePackage{etoolbox}
9 \RequirePackage{array}
10 \RequirePackage{xparse}
11 \RequirePackage{ifxetex}
12
13 \RequirePackage{wasysym}
14 \RequirePackage{adjustbox}
15
16 \RequirePackage{eso-pic}
17
18 \RequirePackage{xcolor}

```

### 3 Options

KV-Options is essential for this.

```

19 \RequirePackage{kvoptions}
20 \SetupKeyvalOptions{ family=hwa,
21   prefix=hwa@ }
22 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

```

`problemstyle=<1>` These options allow the customizatuion of the displayed numbers. For Example, if  
`subproblemstyle=<1>` `problemstyle=Roman`, `subproblemstyle=arabic`, `subsubproblemstyle=roman`  
`subsubproblemstyle=<1>` is passed, The first subsubproblem of the first subproblem of the first problem  
would be labeled as **i)** of **Problem I.1**.

Available options are `arabic`, `Alph`, `alph`, `Roman`, and `roman`. Standard values are:  
`problemstyle=arabic`, `subproblemstyle=alph`, `subsubproblemstyle=roman`.

```

23 \DeclareStringOption[arabic]{problemsty}
24 \DeclareStringOption[alph]{subproblemsty}
25 \DeclareStringOption[roman]{subsubproblemsty}

```

**tikz** Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science  
and mathematics. See 3 for more informations

```

26 \DeclareBoolOption[false]{tikz}

```

**listings** Loads Listings Package and sets listing-layout to use a small fontsize. Adds  
indication for linebreaks.

```

27 \DeclareBoolOption[false]{listings}

```

**oneside, twoside** Changes layout. `oneside` is the complementary option to `twoside`  
Standard layout is `twopaged`.

```

28 \DeclareBoolOption[true]{twoside}
29 \DeclareComplementaryOption{oneside}{twoside}

```

`onecolumn,twocolumn` Changes layout. `onecolumn` is the complementary option to `twocolumn`. Standard Layout has one columns

```
30 \DeclareBoolOption[false]{twocolumn}
31 \DeclareComplementaryOption{onecolumn}{twocolumn}
```

`punchmark` Adds a mark for an hole puncher. Standard Layout has no marking.

```
32 \DeclareBoolOption[false]{punchmark}
```

`hlines=<1>` Key-Value-option. Takes the level of `hlines`. Available are `all`, `decreased`, `header`, `none`, with decreasing number of lines; `none` displays none, `header` only the one under headers and `decreased` adds the big line in the title, while `all` displays all.

```
33 \DeclareStringOption[all]{hlines}
```

`todos=<1>` Key-Value-option. Takes which `ToDo`s shall be displayed. Available are `all` (default), `nolist`, `none`. See subsection 5.4.5 for explanation of the levels.

```
34 \DeclareStringOption[all]{todos}
```

`unicode-math` Loads the `unicode-math`-package and overwrites the damn `\QED-Command` `unicode-math` introduces, that creates a filled out box and only works in `math-`mode, but not telling you that it only works in `math-`mode or overwrites an already existing command. For a reason, that currently (06<sup>th</sup> of December 2018) slips my mind completely, `unicode-math` needs to be loaded after `article`, because it needs to be defined

ATTENTION: Please do never, never, never, never, never ever load `unicode-math` your self, because this breaks **everything**<sup>1</sup>

`\end{rant}`

If XeTeX is used, the default option for this is `true`, otherwise it is false.  
For the handling of the option, see 5.4.1

```
35 \ifxetex
36 \DeclareBoolOption[true]{unicodemath}
37 \else
38 \DeclareBoolOption[false]{unicodemath}
39 \fi

Loads article and processes the options
40 \ProcessKeyvalOptions*
41 \ifhwa@twoside
42 \PassOptionsToClass{twoside}{article}
43 \else
44 \PassOptionsToClass{oneside}{article}
45 \fi
46 \ifhwa@twocolumn
47 \PassOptionsToClass{twocolumn}{article}
48 \else
49 \PassOptionsToClass{onecolumn}{article}
```

```

50 \fi
51 \LoadClass{article}
52
53 \newboolean{hwa@todos@inplace}
54 \newboolean{hwa@todos@list}
55 \setboolean{hwa@todos@inplace}{true}
56 \setboolean{hwa@todos@list}{true}
57 \ifthenelse{\equal{\hwa@todos}{all}}{
58 }{
59   \ifthenelse{\equal{\hwa@todos}{nolist}}{
60     \ClassWarning{homeworkassignment}{You specified todos=none,
61       there will be no list of TODO}
62     \setboolean{hwa@todos@list}{false}
63   }{
64     \ifthenelse{\equal{\hwa@todos}{none}}{
65       \ClassWarning{homeworkassignment}{You specified todos=none,
66         there will be no TODOs printed in the final document}
67       \setboolean{hwa@todos@list}{false}
68       \setboolean{hwa@todos@inplace}{false}
69     }{
70       \ClassError{homeworkassignment}{\hwa@todos is not a valid value
71         for the option ‘todos’}
72     }
73   }
74 }

Load Hyperref (breaks if it is loaded before article
75 \RequirePackage{hyperref}

Loads listings, if wanted
76 \ifhwa@listings
77 \RequirePackage{listings}
78 \lstset{
79   frame = single,
80   breaklines = true,
81   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
82   basicstyle=\scriptsize
83 }
84 \else
85 \empty
86 \fi

```

`\hwa@hline@L...` Defines new commands to output desired lines and change the constant `\hwa@headrulewidth`

ATTENTION: `\hwa@hline@LONE` breaks the line automatically, in opposite to `\hwa@hline@LTWO`

```

88 \newcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
89 \vspace{.25cm}}
90 \newcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
91 \newcommand{\hwa@headrulewidth}{.7pt}
92 \ifthenelse{\equal{\hwa@hlines}{all}}{
93 \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
94 \vspace{.25cm}}
95 \renewcommand{\hwa@headrulewidth}{.7pt}
96 \renewcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
97 }{
98 \ifthenelse{\equal{\hwa@hlines}{decreased}}{
99 \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
100 \vspace{.25cm}}
101 \renewcommand{\hwa@headrulewidth}{.7pt}
102 }{\ifthenelse{\equal{\hwa@hlines}{header}}{
103 \renewcommand{\hwa@headrulewidth}{.7pt}
104 }{\ifthenelse{\equal{\hwa@hlines}{none}}{
105 \renewcommand{\hwa@headrulewidth}{0pt}
106 }{
107 \ClassError{homeworkassignment}{Value '\hwa@lines' for key 'hlines'
108 is not known}{The option hlines takes an argument to set which
109 hlines are drawn. Possible values are 'all','decreased' , 'header', and
110 'none'. 'all' is standard.}
111 }
112 }
113 \renewcommand{\hwa@hline@LONE}{~\vspace{.5cm}}
114 }
115 \renewcommand{\hwa@hline@LTWO}{\vspace{.75cm}}
116 }

```

If tikz is Wanted, load Usefull Styles

```

117 \ifhwa@tikz
118 \RequirePackage{tikz}
119 \usetikzlibrary{shapes,arrows,positioning,decorations,
120 automata,backgrounds,petri,bending,
121 shapes.multipart}
122 \tikzset{
123 treenode/.style = {shape=circle, rounded corners,
124 draw, align=center},
125 graynode/.style = {fill=gray},
126 normalnode/.style = {treenode, font=\Large, bottom color=white},
127 array/.style = {rectangle split,
128 rectangle split horizontal,
129 rectangle split,
130 draw}
131 }
132 \fi

```

Make sure that this is the last Package loaded

```

133 \RequirePackage{geometry}

```

```

134 \ifhwa@twocolumn
135 \geometry{top=2cm, bottom=2cm, left=2cm,
136     headsep=14pt,hmarginratio={1:1}}
137 \else
138 \geometry{top=2cm, bottom=2cm, width=35em,
139     headsep=14pt,hmarginratio={4:3}}
140 \fi

```

## 4 Layout

Initially, the homeworkassignment had a verry *special* appereance, which became much more customizable with version 3.0, see ?? if you want to know how.

### 4.1 Headers & Footers

Sets the page-headers.

All headers are cleared before they get any Text — just to be sure.

The headers have the date on the subject and the author on the right side, the tutorial, sheat-title and deadline on the left side, the pagenumber is displayed in the right footer.

If the document is twopaged, the informations in the headers are splittet, so that author and subject are displayed only on odd pages and the title on even, the pagenumber is displayed on the right side on odd pages and on the left side on even pages.

On the first page, only the date and tutorial will be displayed in the header, the rest of infomration should be in the title.

```

141 \fancypagestyle{firstpage}{
142     %
143     \fancyhf{}
144     % clear all six fields
145     \renewcommand{\headrulewidth}{\hwa@headrulewidth}
146     \renewcommand{\footrulewidth}{0pt}
147     \fancyfoot[R]{\thepage}
148     \fancyhead[L]{\hwa@tutorium}
149     \fancyhead[R]{\@date } }
150 \fancypagestyle{followingpage}{
151     \fancyhf{}
152     \ifhwa@twoside % IF
153     \fancyhead[RO]{\@author}
154     \fancyhead[LO]{\hwa@kurs\
155         \hwa@tutorium}
156     \fancyhead[LE]{
157         \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
158         \GetTranslation{abgabe}: \hwa@abgabe
159     }
160     \fancyfoot[R0,LE]{\thepage}

```



```

161
162 \else %ELSE
163
164 \fancyhead[R]{\hwa@kurs\
165 \@author}
166 \fancyhead[L]{\hwa@tutorium\
167 \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\}
168 \GetTranslation{abgabe}: \hwa@abgabe}
169 \fancyfoot[R]{\thepage}
170 \fi %ENDIF
171 \renewcommand{\headrulewidth}{\hwa@headrulewidth}
172 \renewcommand{\footrulewidth}{0pt}
173 }
174 \pagestyle{followingpage}

```

## 4.2 Enhance Mathenvironments

A couple of thigns, to make math-environments more beautifull and compact.

`\theequation` Displays equation-numbers as upper-case roman numbers.  
175 `\renewcommand{\theequation}{\Roman{equation}}`

`\allowdisplaybreaks` Allow pagebreaks in Mathmode  
176 `\allowdisplaybreaks`

## 4.3 fonts

### 4.4 Serife (Default)

#### 4.4.1 San Serife

I fancy the Gillius-Font-Family, so that is the default Sans-Serif font, when using XeTeX, The template does default to Gillius ADF, which is available for free, licensed under the GNU License.

```

177 \ifthenelse{\boolean{xetex}}{
178 \RequirePackage{fontspec}
179 \setsansfont{TeX Gyre Adventor}
180 \setmainfont{TeX Gyre Pagella}
181 \setmonofont{Fira Mono}
182 }{
183 \RequirePackage{tgadventor}
184 \RequirePackage{tgpagella}
185 \RequirePackage{FiraMono}
186 }

```

#### 4.4.2 Monospace

## 5 Commands

### 5.1 Constants

Defines some constants

`\hwa@pointboxsize` Explains it self.  
187 `\newcommand{\hwa@pointboxsize}{3em}`

### 5.2 Document Informations

`\subject, \kurs` Sets the subject of the document. Takes the subject as argument. Standard Value is “Kein Kurs”  
`\kurs` is deprecated.

```
188 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??} % To store the value
189 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
190 \newcommand{\kurs}[1]{\subject{#1}}
```

`\tutorial, \tutorium` Sets the tutorial of the author. Takes it as an argument. Standard Value is empty, so that this command can be omitted.  
`\tutorium` is deprecated.

```
191 \newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}??} % To store the value
192 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}
193 \newcommand{\tutorium}[1]{\tutorial{#1}}
```

`\deadline, \abgabe` Sets the deadline of the document. Takes it as an argument. Standard value is `\today`.  
`\abgabe` is deprecated

```
194 \newcommand{\hwa@abgabe}{\today} % To store the value
195 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
196 \newcommand{\abgabe}[1]{\deadline{#1}}
```

`\sheetTitle` Sets a descriptonal Title of the Sheet, will be written in the header of every page.  
197 `\newcommand{\hwa@sheetTitle}{}  
198 \newcommand{\sheetTitle}[1]{\def\hwa@sheetTitle{#1}}`

#### 5.2.1 Inherited from article

`\author` Sets the author of the document.  
`\date` Sets the date of the document.

### 5.3 Sectioning

Because the class is designed for Assignments, the sectioning-commands are different from Article

### 5.3.1 internal Sectioning

`\hwa@problem`  
`\hwa@subproblem` These commands work like their counterpart in article, except that there will be  
`\hwa@subsubproblem` no number, nor will they increase a counter. Nevertheless, they will be shown in  
the table of contents. With version 4.0 they were made private, because I figured  
that they are not usefull enough and I am now able to introduce environments  
with the old names

```

199 \DeclareDocumentCommand\hwa@problem{m o}{\@startsection{problem}%Name
200   {1}%Level
201   {\z@}%indent
202   {-2em \@plus -1em \@minus -1em}%beforeskip
203   {1ex \@plus .5ex}%afterskip
204   {\normalfont\Large\sffamily\bfseries}%style
205   *{#1
206     \IfNoValueF{#2}{
207       \hfill
208       \frame{\framebox[\hwa@pointboxsize]{
209         \hfill \normalfont{\large/\small{#2}}}}
210     }
211   }
212   \addcontentsline{toc}{section}{#1}
213 }
214
215 \DeclareDocumentCommand\hwa@subproblem{m o}{\@startsection{subproblem}%Name
216   {2}%Level
217   {\z@}%indent
218   {-1em \@plus -.5em \@minus -.5em}%beforeskip
219   {.5ex \@plus .5ex}%afterskip
220   {\normalfont\large\sffamily\bfseries}%style
221   *{#1
222     \IfNoValueF{#2}{
223       \hfill \framebox[\hwa@pointboxsize]{
224         \hfill \normalfont\large/\small{#2}}
225     }
226   }
227   \addcontentsline{toc}{subsection}{#1}
228 }
229
230 \DeclareDocumentCommand\hwa@subsubproblem{m o}{\@startsection{subsubproblem}%Name
231   {3}%Level
232   {\z@}%indent
233   {-.5em}%beforeskip
234   {.5em}%afterskip
235   {\normalfont\sffamily\bfseries}%style
236   *{#1
237     \IfNoValueF{#2}{
238       \hfill \framebox[\hwa@pointboxsize]{

```

```

239         \hfill\normalfont\large/\scriptsize{#2}}
240     }
241 }
242 }
243

```

**\keyword** Creates a new Paragraph ,which will start with the Argument in Bold, followed by two non-breaking spaces.

```

244 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
245     {4}%Level
246     {\parindent}%indent
247     {-1em}%beforeskip
248     {\z@}%afterskip
249     {\normalfont \sffamily\bfseries}%style
250     *{#1~~}
251 }

```

The following Macros make use of **\keyword**, so it is suggested to use them instead.

<b>\solution</b>	They work like <b>\keyword</b> , but take only an optional Argument print out “Solution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respectively
<b>\proof</b>	2, via <b>\keyword</b> . If an argument is passed, they print out this argument after the
<b>\given</b>	<b>\keyword</b> . They are not mentioned in the table of contents.
<b>\toShow</b>	
<b>\toDisprove</b>	252 \newcommand{\solution}[1] [] {\keyword{\GetTranslation{loesung}\ifstrempy{#1}{~{#1}:}}}
<b>\assumption</b>	253 \newcommand{\toShow}[1] [] {\keyword{\GetTranslation{zuZeigen}\ifstrempy{#1}{~{#1}:}}}
<b>\supposeThat</b>	254 \newcommand{\toDisprove}[1] [] {
	255 \keyword{\GetTranslation{zuWiderlegen}\ifstrempy{#1}{~{#1}:}}
	256 \newcommand{\given}[1] [] {\keyword{\GetTranslation{gegeben}\ifstrempy{#1}{~{#1}:}}
	257 \newcommand{\assumption}[1] [] {\keyword{\GetTranslation{Annahme}\ifstrempy{#1}{~{#1}:}}
	258 \newcommand{\supposeThat}[1] [] {\keyword{\GetTranslation{Angenommen-dass}\ifstrempy{#1}{~{#1}:}}

### 5.3.2 ‘better’ Sectioning

The following commands are an augmented version of the “plain” commands.

<b>\newproblem</b>	
<b>\newproblem*</b>	These commands require no argument, and automatically create a numbered title. They have two optional arguments: <b>\newproblem</b> [#1]{#2} where #1 is the (sub(sub))problem-number and #2 are the points. If there is a number of Points assigned to a (sub(sub))problem, then the command will generate a box to write the reched number of points down next to it.
<b>\newsproblem</b>	
<b>\newsproblem*</b>	

---

<sup>2</sup>As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed.  
See section 9 for all Translations

Normally, `\newproblem` adds the new Created Problem to the grading-table (see subsection 5.5), `\newproblem*` does not do this.

They use counters, of course:

```

259 \newcounter{problem} \setcounter{problem}{0}
260 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
261 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
262
263 \DeclareDocumentCommand\newproblem{0}{ g}{
264   \stepcounter{problem}% to reset the lower counters
265   \ifthenelse{\equal{#1}{}}{
266     % empty
267   }{
268     \setcounter{problem}{#1}
269   }
270
271   \IfNoValueTF{#2}{
272     \hwa@problem{\GetTranslation{aufgabe} \hwa@problemno}
273     \addToGradingTable{\# \hwa@problemno}
274   }{
275     \hwa@problem{\GetTranslation{aufgabe} \hwa@problemno}[#2]
276     \addToGradingTable{\# \hwa@problemno}{/#2}
277   }
278 }
279
280 \WithSuffix\newcommand\newproblem*[1] []{\stepcounter{problem}
281 \ifthenelse{\equal{#1}{}}{ } {
282   \stepcounter{problem}% to reset the lower counters
283   \setcounter{problem}{#1}
284   \hwa@problem{\GetTranslation{aufgabe} \hwa@problemno}
285 }
286
287 \DeclareDocumentCommand\newsubproblem{0}{ g}{
288   \stepcounter{subproblem}
289   \ifthenelse{\equal{#1}{}}{ } {
290     \setcounter{subproblem}{#1}
291   }
292   \IfNoValueTF{#2}{
293     \hwa@subproblem{\GetTranslation{aufgabe}
294       \hwa@problemno{}}.\hwa@subproblemno}
295   }{
296     \hwa@subproblem{\GetTranslation{aufgabe}
297       \hwa@problemno{}}.\hwa@subproblemno}[#2]
298   }
299 }
300
301 \DeclareDocumentCommand\newsbproblem{0}{ g}{
302   \stepcounter{subsubproblem}
303   \ifthenelse{\equal{#1}{}}{ } {\setcounter{subsubproblem}{#1}}

```

```

304 \IfNoValueTF{#2}{
305   \hwa@subsubproblem{\hwa@subsubproblemno)}
306 }
307 {
308   \hwa@subsubproblem{\hwa@subsubproblemno)}[#2]
309 }
310 }
311

```

### 5.3.3 Even Better Sectioning-Environments

```

h jk
312 \NewDocumentEnvironment{problem}{0}{ g}{
313   \newproblem[#1]{#2}
314   \newcommand{\task}[1]{
315     \begin{framed}
316       \keyword{Problem:} ##1
317     \end{framed}
318   }
319 }{}
320 \NewDocumentEnvironment{problem*}{0}{ g}{
321   \newproblem*[#1]{#2}
322   \newcommand{\task}[1]{
323     \begin{framed}
324       \keyword{Problem:} ##1
325     \end{framed}
326   }
327 }{}
328 \NewDocumentEnvironment{subproblem}{0}{ g}{
329   \newsubproblem[#1]{#2}
330   \newcommand{\task}[1]{
331     \begin{framed}
332       \keyword{Problem:} ##1
333     \end{framed}
334   }
335 }{}
336 \NewDocumentEnvironment{subsubproblem}{0}{ g}{
337   \newsubsubproblem[#1]{#2}
338   \newcommand{\task}[1]{
339     \begin{framed}
340       \keyword{Problem:} ##1
341     \end{framed}
342   }
343 }{}

```

## 5.4 Useful Macros

### 5.4.1 QUOD ERAT DEMUNSTRANDUM, End of Proof

`\QED`  
`\EOP` Display a flushed-right *QED*,  $\square$ , or  $\blacksquare$ , respectively. `\qed` is not implemented, to keep compatibility to several Math-packages, which define the later.  
`\eop`

```
344 \newcommand{\hwa@QED}{\begin{flushright}}
345   \textsc{Qed}
346 \end{flushright}
347 }
348 \newcommand{\QED}{\hwa@QED}
349
350 \ifhwa@unicodemath
351 \RequirePackage{unicode-math}
352 \AtBeginDocument{\let\QEDSymbol\QED
353 \renewcommand{\QED}{\hwa@QED}
354 }
355 \fi
356
357 \newcommand{\EOP}{\begin{flushright}
358   \(\square\)
359 \end{flushright}
360 }
361 \newcommand{\eop}{\hfill\(\blacksquare\)}
```

### 5.4.2 QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST

`\QNED`  
`\qned` Display a flushed-right *triangle*. `\QNED` displays it in a new line, `\qned` at the end of the same line.  
In Mathematical proofs this symbol is used to mark things, which we did not intend to proof, but are interesting anyway or things wich are not proofed mathematically, but are explained in a ay, whcih lets no doubt on their correctness.

```
362 \newcommand{\QNED}{\begin{flushright} \(\triangle\)
363 \end{flushright}
364 }
365 \newcommand{\qned}{\hfill\(\triangle\)}
```

### 5.4.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's<sup>3</sup>

---

<sup>3</sup>“Occloxiium” on GitHub:<https://github.com/occloxiium>

#### amath-Class<sup>4</sup>

$\backslash N$			
$\backslash Z$	Defines a set of mathematical sets, which are very usefull (see Table 1)		
$\backslash R$			
$\backslash Q$	Command	Output	Description
$\backslash Compl$	$\backslash N$	$\mathbb{N}$	Natural Numbers
$\backslash F$	$\backslash Z$	$\mathbb{Z}$	Whole Numbers
$\backslash Primes$	$\backslash Q$	$\mathbb{Q}$	Rational Numbers
	$\backslash R$	$\mathbb{R}$	Real Numbers
	$\backslash Compl$	$\mathbb{C}$	Complex Numbers
	$\backslash F_n$	$\mathbb{F}_n$	Prime Field to base $n$
	$\backslash Primes^5$	$\mathbb{P}$	Set of all Primes

Table 1: Field-Commands

```

366 \newcommand{\N}{\ensuremath{\mathbb{N}}}
367 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
368 \newcommand{\R}{\ensuremath{\mathbb{R}}}
369 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
370 \newcommand{\Compl}{\ensuremath{\mathbb{C}}}
371 \newcommand{\F}{\ensuremath{\mathbb{F}}}
372 % The last one is mine
373 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}

```

$\backslash GL$			
$\backslash id$	Output usefull Plaintext-Operators and Functions. See table 2. Require		
$\backslash Var$	Mathmode		
$\backslash Perm$			
$\backslash MComb$			
$\backslash Comb$	Command	Output	
$\backslash Pot$	$\backslash GL$	GL	
$\backslash Map$	$\backslash id$	id	
$\backslash Hom$	$\backslash Var$	Var	
$\backslash Ker$	$\backslash Perm$	Perm	
$\backslash Intpol$	$\backslash Comb$	Comb	
$\backslash Pol$	$\backslash MComb$	MComb	
$\backslash Sol$	$\backslash Pot$	Pot	
$\backslash Bin$	$\backslash Map$	Map	
$\backslash charakteristik$	$\backslash Hom$	Hom	
$\backslash fo$	$\backslash Intpol$	Intpol	
$\backslash first$	$\backslash Pol$	Pol	
$\backslash la$	$\backslash Sol$	Sol	
$\backslash diff$	<sup>4</sup> amath.sty is part of Alexander Bartolomey's Alphabet Classes: <a href="https://github.com/occlonium/AlphabetClasses">https://github.com/occlonium/AlphabetClasses</a>		
$\backslash partdiff$	<sup>5</sup> Has to be $\backslash Primes$ , because $\backslash P$ is already in use		
$\backslash dx$			
$\backslash divides$			
$\backslash property$			
$\backslash dim$			
$\backslash Im$			
$\backslash excup$			
$\backslash falls$			



<code>\Bin</code>	Bin
<code>\charakteristik</code>	char
<code>\diff{&lt;1&gt;}</code>	$\frac{d}{d<1>}$
<code>\partdiff{&lt;1&gt;}</code>	$\frac{\partial}{\partial<1>}$
<code>\divides</code> and <code>property</code>	Prints a vertical line
<code>\dx</code>	$dx$
<code>\excup</code>	$\dot{\cup}$
<code>\fo</code>	fo
<code>\first</code>	fi
<code>\la</code>	la

Table 2: Common Functions

`\falls` prints out »falls«<sup>6</sup>

```

374 \DeclareMathOperator{\GL}{GL}
375 \DeclareMathOperator{\id}{id}
376 \DeclareMathOperator{\Var}{Var}
377 \DeclareMathOperator{\Perm}{Perm}
378 \DeclareMathOperator{\MComb}{MComb}
379 \DeclareMathOperator{\Comb}{Comb}
380 \DeclareMathOperator{\Pot}{Pot}
381 \DeclareMathOperator{\Map}{Map}
382 \DeclareMathOperator{\Hom}{Hom}
383 \DeclareMathOperator{\Ker}{Ker}
384 \DeclareMathOperator{\Intpol}{Intpol}
385 \DeclareMathOperator{\Pol}{Pol}
386 \DeclareMathOperator{\Sol}{Sol}
387 \DeclareMathOperator{\Bin}{Bin}
388 \DeclareMathOperator{\charakteristik}{char}
389 \DeclareMathOperator{\fo}{fo}
390 \DeclareMathOperator{\first}{fi}
391 \DeclareMathOperator{\la}{la}
392
393 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d#1}}}
394 \newcommand{\partdiff}[1]{\ensuremath{\frac{\partial}{\partial#1}}}
395 \newcommand{\dx}{\mathop{}\!\mathrm{d}x}
396 \newcommand{\divides}{\mathop{}\!\mathrm{\mid}}
397 \newcommand{\property}{\mathop{}\!\mathrm{\mid}}
398
399 \renewcommand{\dim}[1][\text]{\ensuremath{\text{dim}_{#1}}}
400 \renewcommand{\Im}{\mathop{}\!\mathrm{Im}}
401
402 \newcommand{\excup}{\mathop{}\!\mathrm{\dot{\cup}}}
403 \newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }

```

<sup>6</sup>In German, actual Translation may differ

### 5.4.4 Rounding

Require Mathmode

Command	Output	Meaning
<code>\floor{&lt;1&gt;}</code>	$\lfloor <1> \rfloor$	floor <1>
<code>\ceil{&lt;1&gt;}</code>	$\lceil <1> \rceil$	ceil <1>
<code>\roundHU{&lt;1&gt;}</code>	$\lfloor <1> + \frac{1}{2} \rfloor$	Round <1> “half up” ( $\lfloor <1> + \frac{1}{2} \rfloor$ )
<code>\roundHD{&lt;1&gt;}</code>	$\lfloor <1> - \frac{1}{2} \rfloor$	Round <1> “half down” ( $\lfloor <1> - \frac{1}{2} \rfloor$ )

Table 3: Rounding Functions

```

404 \newcommand{\floor}[1]{\ensuremath{\left\lfloor\right. #1 \right\rfloor}}
405 \newcommand{\ceil}[1]{\ensuremath{\left\lceil\right. #1 \right\rceil}}
406 \newcommand{\roundHU}[1]{\ensuremath{\left\lfloor\right. #1 \right\rfloor}}
407 \newcommand{\roundHD}[1]{\ensuremath{\left\lfloor\right. #1 \right\rfloor}}

```

`\bigforall`

`\bigexists` Redefines big versions of quantors, adds an h-skip to normal version.

```

408 \let\forall\forallforall\forallforall
409 \let\exists\existsexists\existsexists
410 \renewcommand{\forall}{\ensuremath{\hskip 2pt \forall\hskip 2pt}}
411 \renewcommand{\exists}{\ensuremath{\hskip 2pt \exists\hskip 2pt}}
412 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\Large $\mathsurround4pt\forall$}}}
413 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\Large $\mathsurround4pt\exists$}}}

```

### 5.4.5 ToDos

Utility for the Documentation of ToDos

`\todo` Creates a todo at the location of the command, highlighted in red. The ToDos will be listed after maketitle, unless the option `todos=nolist` or `todos=none` is specified.

```

414 \DeclareDocumentCommand\todo{G}{ }{
415   \ifthenelse{\boolean{hwa@todos@inplace}}{
416     {\color{red}\textbf{~\label{TODO\alph{todoNum}}TODO~\#1~}}
417     \xdef\hwa@todoList@aux{\hwa@todoList@aux
418       \string\item\string\hyperref[TODO\alph{todoNum}]{TODO \#1}
419     }
420     \stepcounter{todoNum}
421   }{ }
422 }

```

Uses the internal `hwa@todo-counter`

```

423 \newcounter{todoNum} \setcounter{todoNum}{1}

```

`\hwa@tableOfTodos` Prints all ToDos

```

424 \DeclareDocumentCommand\hwa@tableOfTodos{}{
425   \ifthenelse{\boolean{hwa@todos@list}}{

```

```

426 \ifthenelse{\equal{\hwa@todoList}{}}{\%Nothing
427 }{
428   {\color{red}
429     \hwa@problem{Table of Todos}
430     \begin{itemize}
431       \hwa@todoList
432     \end{itemize}}
433 }
434 }{}
435 }

```

## 5.5 Grading Table

This Document-Class is still mainly designed for Homework, so it would be nice, if there was a table to write Grades into, wouldn't it?

`\addToGradingTable` Adds the given parameter as an exercise to the Grading-Table. All Problems, created with `\newproblem` are added automatically.

```

436 \DeclareDocumentCommand\addToGradingTable{m g}{
437   \xdef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
438   \xdef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{#1} &}
439   \IfNoValueTF{#2}{
440     \xdef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
441   }{
442     \xdef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill
443       {\string\small #2} &}
444   }
445 }

```

`\makeGradingTable` Outputs a table to fill in the reached Points. Only shows Problems created by `\newproblem`.

Displays the according number of maximum points for each problem, if specified.

Displays the total number of maximum Problems, if given by Argument Like `\tableofcontent`, it needs a second run of L<sup>A</sup>T<sub>E</sub>X, until all are added.

It will never overflow the Line-Width, thanks to an `adjustbox`. [ #1 ]: *Optional*.

The total number of points reachable.

```

446 \DeclareDocumentCommand\makeGradingTable{o}{
447   \begin{table}[hb]
448     \centering
449     \large
450     \begin{adjustbox}{max width=\linewidth}
451       \expandafter\table\expandafter{\hwa@gradingtbl@defs ||p{\hwa@pointboxsize}}\hline
452       \hwa@gradingtbl@lineOne \(\Sigma\) \\\hline\small
453       \hwa@gradingtbl@lineTwo \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm}\\\hline
454     \end{table}
455     \end{adjustbox}
456   \end{table}
457 }

```

See example documents for output

### 5.5.1 Internal commands

`\hwa@gradingtbl@...` Defines macros whose contents will be written to the AUX-File and read in the next run, and the usable commands. The later will contain the information, but have to be defined (incase the aux-file does not exist)

```
458 \edef\hwa@gradingtbl@aux@defs{}
459 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
460 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
461
462 \edef\hwa@gradingtbl@defs{}
463 \newcommand{\hwa@gradingtbl@lineOne}{}
464 \newcommand{\hwa@gradingtbl@lineTwo}{}

```

`\hwa@todoList@...` See `\hwa@gradingtbl@...`

```
465 \newcommand{\hwa@todoList}{}
466 \newcommand\hwa@todoList@aux{}
467 % \end{macro}
468 % \begin{macro}{\write\@auxout}
469 % Write to aux
470 % \begin{macrocode}
471 \AtEndDocument{%
472 \immediate\write\@auxout{%
473 \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
474 }
475 \immediate\write\@auxout{%
476 \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
477 }
478 \immediate\write\@auxout{%
479 \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
480 }
481 \immediate\write\@auxout{%
482 \gdef\string\hwa@todoList{\hwa@todoList@aux}%
483 }
484 }

```

## 5.6 Title

`\maketitle` Overrides `\maketitle`.

```
485 \renewcommand{\maketitle} {
486 \thispagestyle{firstpage}
487 \ifhwa@twocolumn{
488 \twocolumn[
489 \hwa@maketitletext
490 ]}
491 }\else{
492 \hwa@maketitletext
493 }\fi

```

```

494 \hwa@tableOfTodos
495 }

```

`\hwa@maketitletext` Prints out the title with author etc. Used to reduce code duplication for two- and onecolumn styles

```

496 \newcommand{\hwa@maketitletext}{
497   \begin{centering}
498     \huge{\textsf{\textbf{\hwa@kurs}}}\hwa@hline@LONE \large
499     \ifthenelse{\equal{\hwa@sheetTitle}{}}{\textbf{\hwa@sheetTitle}}{\}
500     \GetTranslation{abgabe}: \hwa@abgabe\
501     \hwa@hline@LTWO
502     \normalsize{\@author}\
503     \hwa@hline@LTWO \normalsize
504   \end{centering}
505 }
506 \ifthenelse{\boolean{\hwa@punchmark}}{
507   \newcommand{\hwa@punchmarkRad}{3mm}
508   \newcommand{\hwa@punchmarkDistanceX}{12mm}
509   \newcommand{\hwa@punchmarkDistanceY}{40mm}
510   \AtBeginDocument{
511     % Where will the punch be?
512     \AddToShipoutPictureBG*{\AtPageUpperLeft{
513       \put(\LenToUnit{\hwa@punchmarkDistanceX}-\hwa@punchmarkRad*2},\LenToUnit{-.5\paperheight-\hwa@punchmarkRad*2}){\tikz{\draw (0,0) -- (5mm,0);}}}
514       \put(\LenToUnit{\hwa@punchmarkDistanceX}-\hwa@punchmarkRad*2},\LenToUnit{-.5\paperheight+\hwa@punchmarkRad*2}){\tikz{\draw (0,0) -- (5mm,0);}}}
515     % Punch-Positioningmark
516     \AddToShipoutPictureBG*{\AtPageUpperLeft{
517       \put(\LenToUnit{5mm},\LenToUnit{-.5\paperheight}){\tikz{\draw (0,0) -- (5mm,0);}}}
518     }
519 }{
520 }

```

## 5.7 Counters

The actual counters are defined in subsubsection 5.3.2.

`Counter-Commands` These are used to output the Exercise numbers in the desired style

```

521 \newcommand{\hwa@problemno}{\arabic{problem}}
522 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
523 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}

```

`\hwa@parseCounterStyle` This takes a style-input (#1), one of the three previous defined commands (#2) and the corresponding counter (#3) to redefine #1, so that it corresponds to #2. See ?? for example usement.

```

524 \newcommand{\hwa@parseCounterStyle}[3]{
525   \ifthenelse{\equal{#1}{arabic}}{\renewcommand{#2}{\arabic{#3}}}
526   \ifthenelse{\equal{#1}{roman}}{\renewcommand{#2}{\roman{#3}}}
527   \ifthenelse{\equal{#1}{alph}}{\renewcommand{#2}{\alph{#3}}}
528   \ifthenelse{\equal{#1}{Alph}}{\renewcommand{#2}{\Alph{#3}}}
529   \ifthenelse{\equal{#1}{Roman}}{\renewcommand{#2}{\Roman{#3}}}

```

```

530          \renewcommand{#2}{\Roman{#3}} }{
531          \ClassError{homeworkassignment}{Invalid Value #1 for
532            option Counter-Styling}{Possible Values are alph,
533            arabic, Arabic, roman or Roman.} } } } } }

```

Redefines the three counter-commands:

```

534 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
535 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
536 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}

```

## 6 Environments

### 6.1 Proof

Used for proofes. Starts bth proof and ends with a End-Of-Proof symbol.

```

537 \NewDocumentEnvironment{proof}{G{\GetTranslation{beweis}} O{\QED}}
538 {
539   \keyword{#1:~~}
540 }
541 {
542   #2
543 }

```

### 6.2 Proof by contradiction

Used for proofes. Starts bth proof and ends with a End-Of-Proof symbol.

```

544 \NewDocumentEnvironment{contradiction}{}
545 {
546   \begin{proof}{\GetTranslation{beweis}~\GetTranslation{per}~\GetTranslation{Widerspruch}}[\hfi
547   }
548   {
549     \end{proof}
550 }

```

## 7 Development and support

The package is developed at *GitHub*:

<https://github.com/ACHinrichs/LaTeX-templates>

Please refer to that site for any bug report or development information.

## 8 Changelog

**v1.0 - 2016/10/23** Initial

**v1.1 - 2016/11/02** ...

**v1.2 - 2016/11/03** ...

**v1.3 - 2016/12/01**    • Provide the Class as .dtx file and more

**v1.4 - 2017/04/29**    • “Minor” bugfixes

**v1.5 - 2017/04/29**    • Problems are displayed in the table of contents. Type of numeration is now configurable.

**v1.5.1 - 2017/04/29**    • Bugfix

**v1.5.2 - 2017/04/29**    • Add version-number

**v1.6 - 2017/05/02**    • Add Translations (German and English)

    • Add `\given` and `\toShow`

    • Add `\QED`, `\EOP`, and `\eop`

**v1.6.3 - 2017/05/05**    • Bugfixes

**v1.6.4 - 2017/05/09**    • Change `\eop` to be in the same line

**v1.7 - 2017/05/09**    • Add `\QED`

**v2.0 - 2017/05/23** “Layout 2.0”

    • Change Margins

    • Add Option to select older Page-Style

    • Change `standardlayout` to `twocolumn` and `twoside`

    • ~~Steal~~ Use Macros by Alexander Bartolomey (See 5.4.3)

    • Add some TikZ-Styles

    • Add round functions

**v2.2 - 2017/06/17**    • Add Grading-table

    • Add `\keyword`, `\assumption`, and `\supposeThat`

- Add `\newproblem*`
  - Add `\sheetTitle`
  - Change equation-numbering to uppercase roman
- v2.2.1 - 2017/06/20**    • Fix error with commands like `\solution` and `\keyword`.
- v2.4 - 2017/04/07**    • Fix math alignment
- Add option for flushed left equations
  - Update `amath` port to use
- v3.0 - 2017/12/26** “WS 2017”
- Rename to `homeworkassignment`
  - Add Environment for various proofs
  - Add points for exercises and a place to fill them in
  - Add option to remove or decrease or remove the hlines
  - Remove legacy styles
  - Rework the documentation
  - Beautify Maths
  - Fix `OneColumn-Maktitle-Bug`
  - Fix `Subproblem-Counter` not beeing reset
  - Merry Christmas!
- v3.2 - 2018/12/06** Nikolaus Release
- Make XeLaTeX-Compatible
  - Fix `\newproblem` requiring a Problem-Number
  - Add `\toDisprove` macro similar to the `\toShow` macro
  - Add option for punchmarks
  - Add option to load `unicode-math` and work around a incompatibility
- v4.0 - 2019/04/17** Pre-Easter Release
- Add option to use `unicode-math`
  - Rename `\C` to `\Comp1`, because of a `unicode-math` incompatibility
  - Rework Fonts
  - Make `onecolumn` the default
  - Implement `ToDo`s
  - Replace `\problem`-commands by `problem` environments (which behave like `\newproblem`)
  - Add Task-Command



## 8.1 Version-Scheme

Since Version 2.0 the following version-scheme applies:

**Major Version** has to be increased, if

- there are changes, which create visible changes in the output of existing documents (except for bugfixes), or
- a command is removed or changed in a way, that existing documents do not compile with the new version.

**Minor Version** has to be increased, if

- new backwards compatible commands are introduced
  - Bugfixes may be introduced too.

The minor version of stable releases is always even, the minor version of development versions is always odd. (0 counts as even).

**Patches** May be introduced on Stable Branch. With every non-document-breaking bugfix, the patch-number has to be incremented.

Because Fixing Bugs is a part of development, development-versions do not have numeric patch-numbers, but alphabetic identifiers, directly after the minor-version.

## 9 Translations

Homeworkassignment currently supports English and German, fallback language is German. Unfortunately these two are the only Languages I am capable of translating reliable, so if you want to use an other language, I would be very happy if you would help me to translate homeworkassignment! Please open an issue, author a pull-request or send me an e-mail.

```
551 \DeclareTranslationFallback{aufgabe}{Aufgabe}
552 \DeclareTranslationFallback{loesung}{L\"osung}
553 \DeclareTranslationFallback{beweis}{Beweis}
554 \DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
555 \DeclareTranslationFallback{abgabe}{Abgabe}
556 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
557 \DeclareTranslationFallback{zuWiderlegen}{Zu widerlegen}
558 \DeclareTranslationFallback{gegeben}{Gegeben}
559 \DeclareTranslationFallback{falls}{falls}
560 \DeclareTranslationFallback{Annahme}{Annahme}
561 \DeclareTranslationFallback{Angenommen-dass}{Angenommen, dass}
562 \DeclareTranslationFallback{per}{per}
563 \DeclareTranslationFallback{Widerspruch}{Widerspruch}
564
565 \DeclareTranslation{German}{aufgabe}{Aufgabe}
566 \DeclareTranslation{German}{loesung}{L\"osung}
567 \DeclareTranslation{German}{beweis}{Beweis}
568 \DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
569 \DeclareTranslation{German}{abgabe}{Abgabe}
570 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
571 \DeclareTranslation{German}{zuWiderlegen}{Zu widerlegen}
572 \DeclareTranslation{German}{gegeben}{Gegeben}
573 \DeclareTranslation{German}{falls}{falls}
574 \DeclareTranslation{German}{Falls}{Falls}
575 \DeclareTranslation{German}{Annahme}{Annahme}
576 \DeclareTranslation{German}{Angenommen-dass}{Angenommen, dass}
577 \DeclareTranslation{German}{per}{per}
578 \DeclareTranslation{German}{Widerspruch}{Widerspruch}
579
580 \DeclareTranslation{English}{aufgabe}{Problem}
581 \DeclareTranslation{English}{loesung}{Solution}
582 \DeclareTranslation{English}{beweis}{Proof}
583 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
584 \DeclareTranslation{English}{abgabe}{Deadline}
585 \DeclareTranslation{English}{zuZeigen}{To show}
586 \DeclareTranslation{English}{zuWiderlegen}{To disprove}
587 \DeclareTranslation{English}{gegeben}{Given}
588 \DeclareTranslation{English}{falls}{if}
589 \DeclareTranslation{English}{Falls}{If}
590 \DeclareTranslation{English}{Annahme}{Assumption}
591 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}
592 \DeclareTranslation{English}{per}{by}
```

593 \DeclareTranslation{English}{Widerspruch}{contradiction}

**End**

*The End*

594 \endinput