

# The HomeworkAssignment class\*

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

May 20, 2017

## Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Options</b>	<b>2</b>
2.1	Inherited options . . . . .	2
<b>3</b>	<b>Commands</b>	<b>2</b>
3.1	Document Informations . . . . .	2
3.1.1	Inherited from <code>article</code> . . . . .	3
3.2	Sectioning . . . . .	3
3.2.1	‘plain’ Sectioning . . . . .	3
3.2.2	‘better’ Sectioning . . . . .	3
3.3	Useful Macros . . . . .	3
3.3.1	QUOD ERAT DEMUNSTARNDUM, End of Proof . . . . .	3
3.3.2	QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST . . . . .	3
3.4	Stolen Goods . . . . .	4
<b>4</b>	<b>Pagestyle</b>	<b>5</b>
4.1	Headers . . . . .	5
<b>5</b>	<b>Development and support</b>	<b>6</b>
<b>6</b>	<b>Changelog</b>	<b>6</b>
<b>7</b>	<b>Examples</b>	<b>7</b>
<b>8</b>	<b>Implementation</b>	<b>7</b>
8.1	Packages & Options . . . . .	7
8.2	TikZ-Styles . . . . .	8
8.3	Translations . . . . .	9
8.4	Headers & Footers . . . . .	9

---

\*This document corresponds to `HomeworkAssignment` v2.0,dated 2017/05/20.

8.5	Internal commands . . . . .	11
8.5.1	Counter-Commands . . . . .	11
8.5.2	Counter-Style Parser . . . . .	11
8.5.3	Counter-Commands II . . . . .	11
8.6	Commands . . . . .	11

## 1 Abstract

This class provides a relative simple docuemnt-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 verry best.

## 2 Options

<code>problemstyle=&lt;1&gt;</code>	These options allow the customizatuion of the displayed numbers. For Example, if
<code>subproblemstyle=&lt;1&gt;</code>	<code>problemstyle=Roman</code> , <code>subproblemstyle=arabic</code> , <code>subsubproblemstyle=roman</code>
<code>subsubproblemstyle=&lt;1&gt;</code>	is passed, The first subsubproblem of the first subproblem of the first problem would be labled as <b>i</b> ) of <b>Problem I.1</b> . Available options are <code>arabic</code> , <code>Alph</code> , <code>alph</code> , <code>Roman</code> , and <code>roman</code> . Standard values are: <code>problemstyle=arabic</code> , <code>subproblemstyle=alph</code> , <code>subsubproblemstyle=roman</code> .
<code>design=&lt;1&gt;</code>	Allows the User to select an older page-style, for backwards compatibility. Recognized values are <code>v1</code> and <code>v2</code> . Everytime a version Changes the default look, a new possible value will be added. Only set this if you really need to get an old look, <i>older styles are not going to be maintained!</i>
<code>tikz</code>	Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science and ;athematics. See 8.2 for more informations

### 2.1 Inherited options

Because the class is inherited by `article`, every Option that can be passed to `article`, will be passed to `article`.

## 3 Commands

### 3.1 Document Informations

<code>\subject</code>	Sets the subject of the document. Takes the subject as argument. Standard Value is "Kein Kurs"
<code>\kurs</code>	<code>\kurs</code> is deprecated.
<code>\tutorial</code>	Sets the tutorial of the author. Takes it as an argument. Stamdard Value is empty, so that this command can be omitted.
<code>\tutorium</code>	

`\tutorium` is deprecated.

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

### 3.1.1 Inherited from article

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

## 3.2 Sectioning

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

### 3.2.1 ‘plain’ Sectioning

`\problem` These commands work like their counterpart in article, except that there will be  
`\subproblem` no number, nor will they increase a counter. Nevertheless, they will be shown in  
`\subsubproblem` the table of contents.

`\solution` They work like Paragraph, but do not take an argument, instead they print  
`\proof` out “Lösung”, “Beweis” “Gegeben”, and “Zu zeigen”, respectively<sup>1</sup>. They are not  
`\given` mentioned in the table of contents.  
`\toShow`

### 3.2.2 ‘better’ Sectioning

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

`\newproblem` These commands require no argument, and automatically create a numbered  
`\newsproblem` title. The optional Argument is the new value for the corresponding counter.  
`\newsproblem`

## 3.3 Useful Macros

### 3.3.1 Quod Erat Demunstarndum, End of Proof

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

### 3.3.2 Quod Non Erat Demunstarndum at iucundum est

`\QNEd` Display a flushed-right  $\triangle$ . `\QNEd` displays it in a new line, `\qed` at the end of  
`\qed`

---

<sup>1</sup>As of v1.6, Translations are added, depending on the chosen Language, there may be an other Text displayed.  
See 8.3 for all Translations

the same line.

In Mathematical proofs this symbol is used to mark things, which we did not intend to proof, but are interesting anyway.

### 3.4 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's<sup>2</sup> `amath-Class`<sup>3</sup>

		<sup>45</sup> Defines a set of mathematical fields, which are verry usefull (see Table 1)	
<code>\N</code>	Command	Output	Description
<code>\Z</code>	<code>\N</code>	$\mathbb{N}$	Natural Numbers
<code>\R</code>	<code>\Z</code>	$\mathbb{Z}$	Whole? Numbers
<code>\Q</code>	<code>\Q</code>	$\mathbb{Q}$	Rational Numbers
<code>\C</code>	<code>\R</code>	$\mathbb{R}$	Real Numbers
<code>\F</code>	<code>\C</code>	$\mathbb{C}$	Complex Numbers
<code>\Primes</code>	<code>\F</code>	$\mathbb{F}$	Prime-Field?
	<code>\Primes</code>	$\mathbb{P}$	Set of all Primes

Table 1: Field-Commands

**Functions and Operators** Output usefull Plaintext-Operators and Functions. See table 2.

	Command	Output
<code>\divides</code>	and <code>property</code>	Prints a vertical Bar
	<code>\Var</code>	Var
	<code>\Perm</code>	Perm
	<code>\Comb</code>	Comb
	<code>\MComb</code>	MComb
	<code>\Im</code>	Im
	<code>\Pot</code>	Pot
	<code>\Map</code>	Map
	<code>\Bin</code>	Bin
	<code>\GL</code>	GL
	<code>\id</code>	id
	<code>\dx</code>	$dx$
	<code>\excup</code>	$\dot{\cup}$

<sup>2</sup>“Occloxiium” on GitHub:<https://github.com/occloxiium>

<sup>3</sup>`Amath.sty` is part of Alexander Bartolomey's Alphabet Classes: <https://github.com/occloxiium/AlphabetClasses>

<sup>4</sup>Has to be `\Primes`, because `\P` is already in use

<sup>5</sup>not a Field

<code>\dim[&lt;1&gt;]</code>	$\dim_{<1>}$
<code>\diff{&lt;1&gt;}</code>	$\frac{d}{d<1>}$

Table 2: Text-like Functions

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

## 4 Pagestyle

## 4.1 Headers

To do

---

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

## 5 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.

## 6 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 `\QNEd`

**v2.0 - 2017/05/20** Change Margins,

Add Option to select older Page-Style,

Change standardlayout to twocolumn and twoside

~~Steal~~ Use Macros by Alexander Bartolomey (See 3.4)

Add some TikZ-Styles

## 7 Examples

For examples, please see the Git-Repo at <https://github.com/ACHinrichs/LaTeX-templates>

## 8 Implementation

The following part is very boring, but I have not found a solution to create a .cls-file without including the implementation into the document. Loads L<sup>A</sup>T<sub>E</sub>X<sub>2</sub><sub>ε</sub> and sets the Version Loads the `article`, which is the base-class.

### 8.1 Packages & Options

```
1 \RequirePackage{kvoptions}
2 \SetupKeyvalOptions{ family=hwa,
3   prefix=hwa@ }
4 \DeclareStringOption[arabic]{problemsty}
5 \DeclareStringOption[alph]{subproblemsty}
6 \DeclareStringOption[roman]{subsubproblemsty}
7 \DeclareBoolOption[false]{listings}
8 \DeclareStringOption[v2]{design}
9 \DeclareBoolOption[true]{twoside}
10 \DeclareComplementaryOption{oneside}{twoside}
11 \DeclareBoolOption[true]{twocolumn}
12 \DeclareComplementaryOption{onecolumn}{twocolumn}
13 \DeclareBoolOption[false]{tikz}
14 % Redefine the article-options
15 %   \begin{macrocode}
16 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

    Processes the Options and loads article
17 \ProcessKeyvalOptions*
18 \ifhwa@twoside
19 \PassOptionsToClass{twoside}{article}
20 \else
21 \PassOptionsToClass{oneside}{article}
22 \fi
23 \ifhwa@twocolumn
24 \PassOptionsToClass{twocolumn}{article}
25 \else
26 \PassOptionsToClass{onecolumn}{article}
27 \fi
28 \LoadClass{article}

    Loads required Packages
29 \RequirePackage{suffix}
30 \RequirePackage{fancyhdr}
31 \RequirePackage{ifthen}
32 \RequirePackage{translations}
```

```

33 \RequirePackage{amssymb}
34 \RequirePackage{ams,ath}
35 \ifhwa@listings
36 \RequirePackage{listings}
37 \lstset{
38   frame = single,
39   breaklines = true,
40   postbreak=\raisebox{0ex}[0ex]{0ex}{\ensuremath{\hookrightarrow\space}},
41   basicstyle=\scriptsize
42 }
43 \else
44 \empty
45 \fi
46 \ifhwa@tikz
47 \RequirePackage{tikz}
48 \fi
49 % Make sure that this is the last Package loaded
50 \ifthenelse{\equal{\hwa@design}{v2}}{
51   \RequirePackage{geometry}
52   \ifhwa@twocolumn
53     \geometry{top=2cm, bottom=2cm, left=2cm,
54       headsep=14pt,hmarginratio={1:1}}
55   \else
56     \geometry{top=2cm, bottom=2cm, width=35em,
57       headsep=14pt,hmarginratio={4:3}}
58   \fi
59 }{
60   \ifthenelse{\equal{\hwa@design}{v1}}{
61     \empty
62   }{
63     \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
64       is not known}{The option design takes an argument to set the
65       Pagestyle to the one of a previous version. Acceptable values are
66       'v1', or 'v2'}
67   }
68 }

```

## 8.2 TikZ-Styles

If tikz is Wanted, load Usefull Styles

```

69 \ifhwa@tikz
70 \usetikzlibrary{shapes,arrows,positioning,decorations,
71   automata,backgrounds,petri,bending,
72   shapes.multipart
73 }
74 \tikzset{
75   treenode/.style = {shape=circle, rounded corners,
76     draw, align=center},
77   graynode/.style = {fill=gray},
78   normalnode/.style = {treenode, font=\Large, bottom color=white},

```



```

79 array/.style = {rectangle split,
80     rectangle split horizontal,
81     rectangle split,
82     draw}
83 }
84 \fi

```

### 8.3 Translations

Load translations, currently supports English and German, Fallback is German

```

85 \DeclareTranslationFallback{aufgabe}{Aufgabe}
86 \DeclareTranslationFallback{loesung}{L\ "osung}
87 \DeclareTranslationFallback{beweis}{Beweis}
88 \DeclareTranslationFallback{uebungsgruppe}{\ "Ubungsgruppe}
89 \DeclareTranslationFallback{abgabe}{Abgabe}
90 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
91 \DeclareTranslationFallback{gegeben}{Gegeben}
92 \DeclareTranslationFallback{falls}{falls}
93 \DeclareTranslationFallback{Falls}{Falls}
94
95 \DeclareTranslation{German}{aufgabe}{Aufgabe}
96 \DeclareTranslation{German}{loesung}{L\ "osung}
97 \DeclareTranslation{German}{beweis}{Beweis}
98 \DeclareTranslation{German}{uebungsgruppe}{\ "Ubungsgruppe}
99 \DeclareTranslation{German}{abgabe}{Abgabe}
100 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
101 \DeclareTranslation{German}{gegeben}{Gegeben}
102 \DeclareTranslation{German}{falls}{falls}
103 \DeclareTranslation{German}{Falls}{Falls}
104
105 \DeclareTranslation{English}{aufgabe}{Problem}
106 \DeclareTranslation{English}{loesung}{Solution}
107 \DeclareTranslation{English}{beweis}{Proof}
108 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
109 \DeclareTranslation{English}{abgabe}{Deadline}
110 \DeclareTranslation{English}{zuZeigen}{To show}
111 \DeclareTranslation{English}{gegeben}{Given}
112 \DeclareTranslation{English}{falls}{if}
113 \DeclareTranslation{English}{Falls}{If}

```

### 8.4 Headers & Footers

Sets the page-headers.

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

The headers look like specified above (4.1). Also inserts the Titlepage.

```

114 \fancypagestyle{firstpage}{
115     %
116     \fancyhf{}
117     % clear all six fields

```

```

118 \renewcommand{\headrulewidth}{.7pt}
119 \renewcommand{\footrulewidth}{0pt}
120 \fancyfoot[R]{\thepage}
121 \fancyhead[L]{\hwa@tutorium}
122 \fancyhead[R]{\@date } }
123 \fancypagestyle{followingpage}{
124 %
125 \fancyhf{}
126
127 \ifthenelse{\equal{\hwa@design}{v2}}{
128   \ifhwa@twoside % IF
129
130     \fancyhead[RO]{\@author}
131     \fancyhead[LO]{\hwa@kurs\
132       \hwa@tutorium}
133     \fancyhead[LE]{\GetTranslation{abgabe}: \hwa@abgabe}
134     \fancyfoot[RO,LE]{\thepage}
135
136     \else %ELSE
137
138     \fancyhead[R]{\hwa@kurs\
139       \@author}
140     \fancyhead[L]{\hwa@tutorium\
141       \GetTranslation{abgabe}: \hwa@abgabe}
142     \fancyfoot[R]{\thepage}
143     \fi %ENDIF
144 }{
145   \ifthenelse{\equal{\hwa@design}{v1}}{
146     \fancyhead[RE,LO]{\@author}
147     \fancyhead[LE,RO]{\hwa@kurs\
148       \GetTranslation{abgabe}: \hwa@abgabe}
149     \fancyfoot[RE,LO]{\thepage}
150   }{
151     \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
152       is not known}{The option design takes an argument to set the
153       Pagestyle to the one of a previous version. Acceptable values are
154       'v1', or 'v2'}
155   }
156 }
157
158
159 \renewcommand{\headrulewidth}{0.7pt}
160 \renewcommand{\footrulewidth}{0pt} } \pagestyle{followingpage}
161 \AtBeginDocument{ \thispagestyle{firstpage}
162 \setlength{\headheight}{25pt} }

```

## 8.5 Internal commands

### 8.5.1 Counter-Commands

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

```
163 \newcommand{\hwa@problemno}{\arabic{problem}}
164 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
165 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}
```

### 8.5.2 Counter-Style Parser

**Counter--Style Parser** 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 8.5.3 for example usement.

```
166 \newcommand{\hwa@parseCounterStyle}[3]{
167   \ifthenelse{\equal{#1}{arabic}}{\renewcommand{#2}{\arabic{#3}} }{
168     \ifthenelse{\equal{#1}{roman}}{\renewcommand{#2}{\roman{#3}} }{
169       \ifthenelse{\equal{#1}{alph}}{\renewcommand{#2}{\alph{#3}} }{
170         \ifthenelse{\equal{#1}{Alph}}{\renewcommand{#2}{\Alph{#3}} }{
171           \ifthenelse{\equal{#1}{Roman}}{
172             \renewcommand{#2}{\Roman{#3}} }{
173             \ClassError{HomeworkAssignment}{Invalid Value #1 for
174               option Counter-Styling}{Possible Values are alph,
175               arabic, Arabic, roman or Roman.} } } } } }
```

### 8.5.3 Counter-Commands II

**Counter--Style ParserCommands II** Redefines the three counter-commands

```
176 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
177 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
178 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}
```

## 8.6 Commands

**\subject** Defines \kurs. \subject equals \kurs

```
179 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??}
180 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
181 \newcommand{\kurs}[1]{\subject{#1}}
```

**\tutorial** Defines \tutorial. \tutorial equals \tutorial

```
182 \newcommand{\hwa@tutorial}{?\GetTranslation{uebungsgruppe}??}
183 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorial}{#1}}
184 \newcommand{\tutorial}[1]{\tutorial{#1}}
```

**\deadline** Defines \deadline. \abgabe equals \deadline

```
185 \newcommand{\hwa@abgabe}{\today}
186 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
187 \newcommand{\abgabe}[1]{\deadline{#1}}
```

`\maketitle` Overrides maketitle.

```

188
189 \renewcommand{\maketitle} {
190   \twocolumn[{\%
191     \begin{centering}
192       \huge{\textbf{\hwa@kurs}} \vspace{.25cm} {\hrule height 2pt}
193       \vspace{.25cm} \large
194       \GetTranslation{abgabe}: \hwa@abgabe\\
195       \vspace{.5cm} \hrule \vspace{.25cm}
196       \normalsize{\@author}\\
197       \vspace{.25cm} \hrule \vspace{.25cm} \normalsize
198     \end{centering}
199   }]
200 }
```

Defines and initialize all counters.

```

201 \newcounter{problem} \setcounter{problem}{0}
202 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
203 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
204
```

Defines ‘plain’ sectioning-commands. See 3.2 for more informations.

```

205 \newcommand{\problem}[1]{\@startsection{problem}%Name
206   {1}%Level
207   {\z@}%indent
208   {-2em \@plus -1em \@minus -1em}%beforeskip
209   {1ex \@plus .5ex}%afterskip
210   {\normalfont\Large\bfseries}%style
211   *{#1} \addcontentsline{toc}{section}{#1} }
212
213 \newcommand{\subproblem}[1]{\@startsection{subproblem}%Name
214   {2}%Level
215   {\z@}%indent
216   {-1em \@plus -.5em \@minus -.5em}%beforeskip
217   {.5ex \@plus .5ex}%afterskip
218   {\normalfont\large\bfseries}%style
219   *{#1} \addcontentsline{toc}{subsection}{#1} }
220
221 \newcommand{\subsubproblem}[1]{\@startsection{subsubproblem}%Name
222   {3}%Level
223   {\z@}%indent
224   {-.5em}%beforeskip
225   {.5em}%afterskip
226   {\normalfont\bfseries}%style
227   *{#1} }
228
229 \newcommand{\solution}[1][ ]{\@startsection{solution}%Name
230   {4}%Level
231   {\parindent}%indent
232   {-.1em}%beforeskip
```

```

233 {\z@}%afterskip
234 {\normalfont\bfseries}%style
235 *{\GetTranslation{loesung}\ifthenelse{\equal{#1}{}} {} { #1}:~~ } }
236
237 \newcommand{\proof}[1] [] {\@startsection{proof}%Name
238 {4}%Level
239 {\parindent}%indent
240 {- .1em}%beforeskip
241 {\z@}%afterskip
242 {\normalfont\bfseries}%style
243 *{\GetTranslation{beweis}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }
244
245 \newcommand{\toShow}[1] [] {\@startsection{to show}%Name
246 {4}%Level
247 {\parindent}%indent
248 {- .1em}%beforeskip
249 {\z@}%afterskip
250 {\normalfont\bfseries}%style
251 *{\GetTranslation{zuZeigen}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }
252
253 \newcommand{\given}[1] [] {\@startsection{given}%Name
254 {4}%Level
255 {\parindent}%indent
256 {- .1em}%beforeskip
257 {\z@}%afterskip
258 {\normalfont\bfseries}%style
259 *{\GetTranslation{gegeben}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }
260

```

Defines ‘better’ sectioning commands. See 3.2 and 3.2.2 for more informations.

```

261 \newcommand{\newproblem}[1] [] {\stepcounter{problem}
262 \ifthenelse{\equal{#1}{}} {} {} {\setcounter{problem}{#1}}
263 \problem{\GetTranslation{aufgabe} \hwa@problemno} }
264
265 \newcommand{\newsproblem}[1] [] {\stepcounter{subproblem}
266 \ifthenelse{\equal{#1}{}} {} {} {\setcounter{subproblem}{#1}}
267 \subproblem{\GetTranslation{aufgabe} \hwa@problemno}{.\hwa@subproblemno} }
268
269 \newcommand{\newsproblem}[1] [] {\stepcounter{subsubproblem}
270 \ifthenelse{\equal{#1}{}} {} {} {\setcounter{subsubproblem}{#1}}
271 \subsubproblem{\hwa@subsubproblemno}} }
272

```

End of Proof

```

273 \newcommand{\QED}{\begin{flushright}
274 \textit{QED}
275 \end{flushright}}
276 }
277 \newcommand{\EOP}{\begin{flushright}
278 $\square$

```

```

279 \end{flushright}
280 }
281 \newcommand{\eop}{\hfill$\blacksquare$}

```

c demonstrandum at iucundum est

```

282 \newcommand{\QED}{\begin{flushright}
283   $\triangle$
284 \end{flushright}
285 }
286 \newcommand{\qed}{\hfill$\triangle$}

```

The following Macros are all stolen (and adapted) from occloxiun (see 3.4)

#### Math Common Set Symbols

```

287 \newcommand{\N}{\ensuremath{\mathbb{N}}}
288 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
289 \newcommand{\R}{\ensuremath{\mathbb{R}}}
290 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
291 \newcommand{\C}{\ensuremath{\mathbb{C}}}
292 \newcommand{\F}{\ensuremath{\mathbb{F}}}
293 % The last one is mine
294 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}

```

#### Mathematical Functions

```

295 \newcommand{\GL}{\ensuremath{\text{GL}}}
296 \newcommand{\id}{\ensuremath{\text{id}}}
297 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d\#1}}}
298 \newcommand{\dx}{\text{d}x}
299
300 \newcommand{\divides}{\ensuremath{\mid}}
301 \newcommand{\property}{\ensuremath{\mid}}
302
303 \newcommand{\Var}{\ensuremath{\text{Var}}}
304 \newcommand{\Perm}{\ensuremath{\text{Perm}}}
305 \newcommand{\MComb}{\ensuremath{\text{MComb}}}
306 \newcommand{\Comb}{\ensuremath{\text{Comb}}}
307
308 \renewcommand{\dim}[1][\text{dim}]{\ensuremath{\text{dim}_{\#1}}}
309 \renewcommand{\Im}{\ensuremath{\text{Im}}}
310
311 \newcommand{\Pot}{\ensuremath{\text{Pot}}}
312 \newcommand{\Map}{\ensuremath{\text{Map}}}
313
314 \newcommand{\excup}{\ensuremath{\stackrel{\cdot}{\cup}}}
315
316 \newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }
317
318 \newcommand{\Bin}{\ensuremath{\text{Bin}}}

```

#### Math Big Quantors

```

319 \let\forall\forall
320 \let\exists\exists
321 \renewcommand{\forall}{\ensuremath{\hskip 2pt \forall \hskip 2pt}}
322 \renewcommand{\exists}{\ensuremath{\hskip 2pt \exists \hskip 2pt}}
323 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\height}{\Large $\mathsurround4pt\forall$}}}
324 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\height}{\Large $\mathsurround4pt\exists$}}}

The End
325 \endinput

```