

# tikzsymbols\*

Ben Vitecek  
b.vitecek@gmx.at

February 28, 2013

## Abstract

Just some symbols created with “tikz”.  
English is not my native language. So there might be some errors ☺

## Contents

<b>1</b>	<b>Short Introduction</b>	<b>2</b>
<b>2</b>	<b>Options</b>	<b>2</b>
<b>3</b>	<b>Symbols</b>	<b>2</b>
3.1	cooking-symbols 🍳 . . . . .	2
3.2	Emoticons ☺ . . . . .	3
3.2.1	“normal” Emoticons 🐼 . . . . .	3
3.2.2	“3D” Emoticons 🍌 . . . . .	4
3.3	other Symbol(s) 🍷 . . . . .	5
3.4	Trees 🌳 . . . . .	5
3.5	Something to redefine . . . . .	6
<b>4</b>	<b>Warnings and Errors</b>	<b>6</b>
4.1	Warnings . . . . .	6
4.2	and errors . . . . .	7
<b>5</b>	<b>Nobody is perfect</b>	<b>7</b>
<b>6</b>	<b>Code</b>	<b>7</b>
6.1	Cookingsymbolcode . . . . .	7
6.2	Emoticonscode . . . . .	12
<b>7</b>	<b>Other symbols(s)</b>	<b>22</b>
7.1	Trees . . . . .	25

---

\*This document corresponds to tikzsymbols v1.7, dated 2013/02/28.

# 1 Short Introduction

There are about two emoticons available in L<sup>A</sup>T<sub>E</sub>X: Smiley and Frowny. But why aren't there more? Or why did nobody make cooking-symbols? I thought about this questions and during a project I developed some (cooking)symbols. Developing them was real fun and so I made some more, reworked some etc. And here they are.

## 2 Options

These options are for the commands in the section "Trees" 3.4. The trees look pretty nice, but have one drawback: L<sup>A</sup>T<sub>E</sub>X needs extremely long to produce them. So these options come in handy: by turning **tree=off** or using **draft** the trees will be replaced by squares (for an example see section "Trees" 3.4). Those squares are fast produced by L<sup>A</sup>T<sub>E</sub>X and have almost the same size as the trees, they are "spacefillers". In your final document you can turn **tree=on**, delete it or write **final** and the trees will be produced.

Options to produce normal trees: 🌳	Options for "spacefillers": ☐
<code>\usepackage{tikzsymbols}</code>	
<code>\usepackage{tree=on}{tikzsymbols}</code>	<code>\usepackage{tree=off}{tikzsymbols}</code>
<code>\usepackage{final}{tikzsymbols}</code>	<code>\usepackage{draft}{tikzsymbols}</code>
<code>\documentclass[final]{class}</code>	<code>\documentclass[draft]{class}</code>
<code>\usepackage{tikzsymbols}</code>	<code>\usepackage{tikzsymbols}</code>

Note: you shouldn't use both, **tree=on/off** and class-option **draft**. It's just unnecessary.

While working on this document I used the **draft** syntax and in the final output I deleted it.

But there are no warnings, if you misspell something e.g. **tree=onf**. You will see what happens.

## 3 Symbols

In this section the symbols are introduced. They 🍷 all 🍷 change 🍷 automatically 🌳 with 🍷 the textsize 🍷.














### 3.1 cooking-symbols 🍷

At the following table the cooking-symbols are listed.

The first column shows the Command (at first the german at second the english). The second are the optional keys.

`<scale>` can be a number between 0 and (not exactly) 1400<sup>1</sup>, default is 1. The optional parameter(s) are for both, the german and the english commands the same.

Da Umlaute nicht angezeigt werden können, werden die Umlaute ö, ä, ü ersetzt durch: o, a, u.

German & English Commands	Optional parameter(s)	Output (defaultsize)
<code>\Kochtopf</code>	<code>\pot</code>	<code>[\scale]</code> 
<code>\Bratpfanne</code>	<code>\fryingpan</code>	<code>[\scale]</code> 
<code>\Schneebeesen</code>	<code>\eggbeater</code>	<code>[\scale]</code> 
<code>\Sieb</code>	<code>\sieve</code>	<code>[\scale]</code> 
<code>\Purierstab</code>	<code>\blender<sup>2</sup></code>	<code>[\scale]</code> 
<code>\Dreizack</code>	<code>\trident</code>	<code>[\scale]</code> 
<code>\Backblech</code>	<code>\bakingplate</code>	<code>[\scale]</code> 
<code>\Ofen</code>	<code>\oven</code>	<code>[\scale]</code> 
<code>\Pfanne</code>	<code>\pan</code>	<code>[\scale]</code> 
<code>\Herd</code>	<code>\cooker</code>	<code>[\scale]</code> 
<code>\Saftpresse</code>	<code>\squeezer</code>	<code>[\scale]</code> 
<code>\Schussel</code>	<code>\bowl</code>	<code>[\scale]</code> 
<code>\Schaler</code>	<code>\peeler</code>	<code>[\scale]</code> 

## 3.2 Emoticons ☺

### 3.2.1 “normal” Emoticons 🐱

First column shows the commands, the second the optional parameter(s), the third the default-output.

`<scale>` can be a number between 0 and not exactly 2000<sup>3</sup>, default is 1.











`<color>` can be every defined color.








Commands	Optional parameter(s)	Output (default)
<code>\Sadey</code>	<code>[\scale][\color]</code>	☹
<code>\Smiley</code>	<code>[\scale][\color]</code>	☺
<code>\Laughy</code>	<code>[\scale][\color][\mouth color]</code>	😄
<code>\Annoey</code>	<code>[\scale][\color]</code>	☹
<code>\Neutrey</code>	<code>[\scale][\color]</code>	☹
<code>\Winkey</code>	<code>[\scale][\color]</code>	☺
<code>\Sey</code>	<code>[\scale][\color]</code>	☺

<sup>1</sup>Well, you can also use a number between 0 and −1400 which will lead to something like this: ☹, ☹, ☹. Note: Some symbols like “Kochtopf/pot” can’t be scaled in such a way. Furthermore the symbols are not designed for negative numbers and will have a bad line width.

<sup>2</sup>I know that “Purierstab” should be translated as “immersion blender”, but I’m just using “blender”

<sup>3</sup>Do you even need so large symbols?















Commands	Optional parameter(s)	Output (default)
<code>\Innocey</code>	<code>[\scale][\color][\halo color]</code>	
<code>\wInnocey</code>	<code>[\scale]</code>	
<code>\Cooley</code>	<code>[\scale][\color]</code>	
<code>\Tongey</code>	<code>[\scale][\color][\tongue color]</code>	
<code>\Nursey<sup>4</sup></code>	<code>[\scale][\color][\cap color][\cross color]</code>	
<code>\Vomey</code>	<code>[\scale][\color][\vomit color]</code>	
<code>\Walley</code>	<code>[\scale][\color][\wall color]</code>	
<code>\Cat</code>	<code>[\scale]</code>	
<code>\Ninja</code>	<code>[\scale][\color][\headband color][\eye color]</code>	
<code>\NiceReapey</code>	<code>[\scale]</code>	

Examples: `\Sadey[] [red]`  `\Cooley[3] [cyan]`   
`\Vomey[1.5] [green!80!black] [olive]`   
`\Nursey[] [yellow] [blue] [red]`   
`\Ninja[1.3] [] [violet] [red]`   
`\colorbox{yellow}{\Winkey \Annoey \Neutrey}`   
`{\color{blue}\Sey}` 

### 3.2.2 “3D” Emoticons

First column shows the commands (note: the “3D” Emoticons begin with `\d...`), the second the optional parameter(s), the third the default-output.

`<scale>` can be a number between 0 and a large number<sup>5</sup>, default is 1. `<color>` can be every defined color (see examples below).

Commands	Optional parameter(s)	Output (default)
<code>\dSadey</code>	<code>[\scale][\color]</code>	
<code>\dSmiley</code>	<code>[\scale][\color]</code>	
<code>\dLaughhey</code>	<code>[\scale][\color][\mouth color]</code>	
<code>\dAnnoey</code>	<code>[\scale][\color]</code>	
<code>\dNeutrey</code>	<code>[\scale][\color]</code>	
<code>\dWinkey</code>	<code>[\scale][\color]</code>	
<code>\dSey</code>	<code>[\scale][\color]</code>	
<code>\dInnocey</code>	<code>[\scale][\color][\halo color]</code>	
<code>\dCooley</code>	<code>[\scale][\color]</code>	
<code>\dTongey</code>	<code>[\scale][\color][\tongue color]</code>	
<code>\dNursey<sup>4</sup></code>	<code>[\scale][\color][\cap color][\cross color]</code>	
<code>\dVomey</code>	<code>[\scale][\color][\vomit color]</code>	
<code>\dWalley</code>	<code>[\scale][\color][\wall color]</code>	
<code>\dNinja</code>	<code>[\scale][\color][\headband color][\eye color]</code>	

<sup>4</sup>The cross has nothing to do with religion meanings.

<sup>5</sup>over 500 for sure

Examples: `\dSadey[] [red]` 🍷 `\dCooley[3] [cyan]` 😎  
`\dVomey[1.5] [green!70!black] [olive]` 🌿  
`\dNursey[] [yellow] [blue] [red]` 🌈  
`\dNinja[1.3] [] [violet] [red]` 🌺.

### 3.3 other Symbol(s) 📖

Commands	Optional parameter(s)	Output (default)
<code>\Person</code>	<code>[\scale]</code>	👤
<code>\Candle</code>	<code>[\scale]</code>	🕯️
<code>\Fire</code>	<code>[\scale]</code>	🔥
<code>\Coffeecup</code>	<code>[\scale]</code>	☕
<code>\Chair</code>	<code>[\scale]</code>	🪑
<code>\Bed</code>	<code>[\scale]</code>	🛏️
<code>\Moai</code>	<code>[\scale]</code>	🗿

### 3.4 Trees 🌳

“Hey, these trees look exactly like the ones in the tikzmanual” – “NO! Not “exactly”, they look pretty a like... Well I changed them a bit... Hey! The best ideas are stolen ...”

`<scale>` can be a number between 0 and 128<sup>6</sup>, default is 1.

`<color>` can be every defined color.

`{leaf}` uses the colors of `{\leaf color a}` and `{\leaf color b}`, you can leave this one empty, if you don’t want leaves (`\Wintertree` is without *leave*, see examples below).

If you are using those trees, L<sup>A</sup>T<sub>E</sub>X needs longer to produce the pdf. So you may use the package option `tree=off` or `draft` (see section 2) to make L<sup>A</sup>T<sub>E</sub>X faster.

Commands	Optional/Needed parameter(s)	Output
<code>\BasicTree</code>	<code>[\scale]{\trunk color}{\leaf color a}{\leaf color b}{leaf}</code>	see below
<code>\Springtree</code>	<code>[\scale]</code>	🌳
<code>\Summertree</code>	<code>[\scale]</code>	🌳
<code>\Autumntree</code>	<code>[\scale]</code>	🌳
<code>\Wintertree</code>	<code>[\scale]</code>	🌳
<code>\WorstTree</code>	<code>[\scale]</code>	🌳

`\BasicTree` examples:


`\BasicTree{red}{orange}{yellow}{leaf}` 🌳

`\BasicTree[5]{orange!95!black}{orange!80!black}{orange!70!black}{leaf}`




<sup>6</sup>if it is larger it uses too much of L<sup>A</sup>T<sub>E</sub>X memory and an error message appears. Of course there are also negative numbers allowed, but a number smaller than -2 doesn’t look good anymore.

```
\BasicTree[2]{blue!65!white}{cyan!50!white}{cyan!50!white}{}
```



\BasicTree example with option `tree=off` or `draft` (see section 2):


```
\BasicTree{red}{orange}{yellow}{leaf}
```



```
\BasicTree[5]{orange!95!black}{orange!80!black}{orange!70!black}{leaf}
```



```
\BasicTree[2]{blue!65!white}{cyan!50!white}{cyan!50!white}{}
```



I think it's best if you define your own tree using `\newcommand` and `\BasicTree` (don't forget `\xspace`):

```
\newcommand{\Myicetree}[1][1]{%
  \BasicTree[#1]{blue!65!white}{cyan!50!white}{cyan!50!white}{}\xspace}
```

### 3.5 Something to redefine

At the end of the code I am using the command:

```
\newcommand{\tikzsymbolsaftersymbolinput}{\xspace}
```

You may change this (for some reasons I don't know). If you want “...” after every symbol you can define: `\renewcommand{\tikzsymbolsaftersymbolinput}{\dots}` which will lead to: ☺...🌳...!...

Well that's it.

## 4 Warnings and Errors

### 4.1 Warnings

You can use this symbols in chapters, sections, subsections, etc. But the log file will print a warning, something like:

```
Package hyperref Warning: Token not allowed in a PDF string (PDF-
DocEncoding): (hyperref) removing '\Smiley' on input line 137.
```

You can avoid those messages by putting the symbol into this command:

```
\texorpdfstring{\Smiley}{Smiley}
```

For example you may use something like that:

```
\subsubsection{'3D' Emoticons \texorpdfstring{\dSmiley}{dSmiley}}
```

or

```
\subsection{Emoticons \texorpdfstring{\Smiley}{Smiley}}
```

or

```
\subsubsection{'normal' Emoticons \texorpdfstring{\Cat}{Cat}}
```

If you misspell `tree=on` or `tree=off` the output will be something like: “1redor-angeyellowleaf”. If that happens, you misspelled something (“on” or “off”). I have no idea how I can solve this (it was hard enough to make an option that works).

## 4.2 and errors

If you load the package “marvosym” make sure you load “tikzsymbols” after this package because both packages define `\Smiley`, “marvosym” via `\newcommand` “tikzsymbols” via `\DeclareRobustCommand`.

If you load “marvosym” *after* “tikzsymbols”, L<sup>A</sup>T<sub>E</sub>X generates an error message because “Smiley” has already been defined.

If you load “marvosym” *before* “tikzsymbols”, “tikzsymbols” will overwrite marvosym’s Smiley and no error message is generated.

## 5 Nobody is perfect

If you are sure that you found a bug, please send me a mail involving a *minimal example* of the code which shows the bug. And a description would be nice.

## 6 Code (do you really need this section?)

There is not much to see, all this symbols were created with “tikz”. But it may helps you (somehow).

The first lines are always the same: what do I need, how is the package named:

```
1 \NeedsTeXFormat{LaTeX2e}[2011/06/27]
2 \ProvidesPackage{tikzsymbols}
3 [2013/02/28 v1.7 Some symbols created using tikz.]
4 \RequirePackage{tikz}
5 \RequirePackage{xargs}
6 \RequirePackage{xcolor}
7 \RequirePackage{xspace}
```

Furthermore we need to load some libraries from tikz:

```
8 \usetikzlibrary{arrows,decorations.pathmorphing,trees}
```

`\tikzsymbolsaftersymbolinput` Now we define this strange named macro. This macro is inserted after the tikz-code, and is defined as `\xspace` (there may be some changes in future, and to write less I define this macro)

```
9 \newcommand{\tikzsymbolsaftersymbolinput}{\xspace}
```

### 6.1 Cookingsymbolcode

`\Kochtopf = \pot` I am using `\DefineRobustCommand` so that the symbols can be used in `\section{}`, `\footnote`, `\index{}`, etc. You can either use the german commands or the english ones:

```
10 \DeclareRobustCommand{\Kochtopf}[1][1]{%
11 \begin{tikzpicture}[x=2ex,y=2.2ex, line width=0.07ex*#1,scale=#1]
12 \draw[rounded corners=0.2ex*#1] (0,0.5) -- (0,0) -- (1,0) -- (1,0.5);
13 \draw(0,0.4) arc (90:270:0.1);
14 \draw(1,0.4) arc (90:-90:0.1);
```

```

15 \draw (0,0.5) -- (1,0.5) .. controls (1,0.6) and (0,0.6) .. (0,0.5);
16 \draw (0.6,0.585) arc (0:180:0.1);
17 \draw[decorate,decoration={snake,amplitude=.12ex*#1,segment length=0.93ex*#1}]
18 (0,0.35) -- (1,0.35);
19 \draw (0.1,0.25) circle (0.04);
20 \draw (0.3,0.2) circle (0.04);
21 \draw (0.13, 0.125) circle (0.04);
22 \draw (0.6,0.25) circle (0.04);
23 \draw (0.45,0.1) circle (0.04);
24 \draw (0.88,0.2) circle (0.04);
25 \draw (0.7,0.11) circle (0.04);
26 \end{tikzpicture}%
27 \tikzsymbolsaftersymbolinput%
28 }
29 \let\pot\Kochtopf

```

\Bratpfanne = \fryingpan You may wonder why I am writing something like: `amplitude=.12ex*#1`. Well it's hard to explain in english, but I try my best: After being scaled these symbols would look not so good without `*#1`. The lines would be too thin, the corners not rounded enough, etc. To prevent too thin lines due to scaling I am multiplying the line width and the corners etc. so that they look with `scale=5` a like with `scale=1`.

```

30 \DeclareRobustCommand{\Bratpfanne}[1][1]{%
31 \begin{tikzpicture}[x=0.7ex,y=1.4ex, line width=0.07ex*#1, scale=#1]
32 \draw[rounded corners=0.07ex*#1] (-1,0) -- (1,0) -- (1.5,0.4) -- (-1.5,0.4) -- cycle;
33 \draw[ line width=0.037ex*#1, rounded corners=0.023ex*#1]
34 (-1.4,0.3) -- (-3.5,0.3) -- (-3.5,0.25) -- (-1.3,0.25);
35 \draw[line width=0.023ex*#1] (-1.1,0.1) -- (1.1,0.1);
36 \draw[line width=0.035ex*#1, decorate,
37   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
38 (-0.3,0.5) -- (-0.3,1);
39 \draw[line width=0.035ex*#1, decorate,
40   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
41 (0.3,0.5) -- (0.3,1);
42 \draw[line width=0.035ex*#1, decorate,
43   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
44 (-1,0.5) -- (-1,1);
45 \draw[line width=0.035ex*#1, decorate,
46   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
47 (1,0.5) -- (1,1);
48 \end{tikzpicture}%
49 \tikzsymbolsaftersymbolinput%
50 }
51 \let\fryingpan\Bratpfanne

```

\Schneebesen = \eggbeater The next one:

```

52 \DeclareRobustCommand{\Schneebesen}[1][1]{%
53 \begin{tikzpicture}[y=2.1ex,x=1.4ex, scale=#1]
54 \draw[line width=0.01ex*(#1-.#1*3)]
55 (0,0) .. controls (0.2,0.0) and (0.2,0.2) .. (0,0.4);

```



```

56 \draw[line width=0.01ex*(#1-.#1*3)]
57   (0,0) .. controls (-0.2,0.0) and (-0.2,0.2) .. (0,0.4);
58 \draw[line width=0.01ex*(#1-.#1*3)]
59   (0,0) .. controls (-0.1,0.0) and (-0.1,0.2) .. (0,0.4);
60 \draw[line width=0.01ex*(#1-.#1*3)]
61   (0,0) .. controls (0.1,0.0) and (0.1,0.2) .. (0,0.4);
62 \draw[line width=0.01ex*(#1-.#1*3)]
63   (0,0) .. controls (-0.15,0.0) and (-0.15,0.2) .. (0,0.4);
64 \draw[line width=0.01ex*(#1-.#1*3)]
65   (0,0) .. controls (0.15,0.0) and (0.15,0.2) .. (0,0.4);
66 \draw[line width=0.01ex*(#1-.#1*3)]
67   (0,0) .. controls (-0.05,0.0) and (-0.05,0.2) .. (0,0.4);
68 \draw[line width=0.01ex*(#1-.#1*3)]
69   (0,0) .. controls (0.05,0.0) and (0.05,0.2) .. (0,0.4);
70 \draw[line width=0.01ex*(#1-.#1*3)]
71   (0,0) --(0,0.4);
72 \fill[line width=0.05ex*#1, rounded corners=0.07ex*#1]
73   (-0.05,0.37) -- (0.05,0.37) -- (0.05,0.75) -- (-0.05,0.75) -- cycle;
74 \end{tikzpicture}%
75 \tikzsymbolsaftersymbolinput%
76 }
77 \let\eggbeater\Schneebesen

```

\Sieb = \sieve Now a long one;

```

78 \DeclareRobustCommand{\Sieb}[1][1]{%
79 \begin{tikzpicture}[x=2.8ex, y=2.8ex, line width=0.02ex*#1 ,scale=#1]
80 \draw[line width=0.09ex*#1] (-0.2,0) -- (1.01,0);
81 \draw (0.2,0) arc (180:360:0.4);
82 \draw (0.25,0) arc (180:360:0.35);
83 \draw (0.3,0) arc (180:360:0.3);
84 \draw (0.35,0) arc (180:360:0.25);
85 \draw (0.4,0) arc (180:360:0.2);
86 \draw (0.45,0) arc (180:360:0.15);
87 \draw (0.5,0) arc (180:360:0.1);
88 \draw (0.55,0) arc (180:360:0.05);
89 \draw (.95,0) -- (0.95,-0.194);
90 \draw (.9,0) -- (0.9,-0.265);
91 \draw (.85,0) -- (0.85,-0.313);
92 \draw (.8,0) -- (0.8,-0.345);
93 \draw (.75,0) -- (0.75,-0.37);
94 \draw (.7,0) -- (0.7,-0.39);
95 \draw (.65,0) -- (0.65,-0.4);
96 \draw (.6,0) -- (0.6,-0.4);
97 \draw (.55,0) -- (0.55,-0.4);
98 \draw (.5,0) -- (0.5,-0.39);
99 \draw (.45,0) -- (0.45,-0.37);
100 \draw (.4,0) -- (0.4,-0.348);
101 \draw (.35,0) -- (0.35,-0.314);
102 \draw (.3,0) -- (0.3,-0.265);
103 \draw (.25,0) -- (0.25,-0.194);

```

```

104 \draw (0.2,-0.05) -- (1,-0.05);
105 \draw (0.21,-0.1) -- (0.99,-0.1);
106 \draw (0.23,-0.15) -- (0.97,-0.15);
107 \draw (0.255,-0.2) -- (0.945,-0.2);
108 \draw (0.289,-0.25) -- (0.911,-0.25);
109 \draw (0.335,-0.3) -- (0.865,-0.3);
110 \draw (0.406,-0.35) -- (0.794,-0.35);
111 \end{tikzpicture}%
112 \tikzsymbolsaftersymbolinput%
113 }
114 \let\sieve\Sieb

```

`\Purierstab = \blender` Da es keine Umlaute gibt, werden ä, ü, ö einfach zu: a, u, o. This symbol is far from perfect. And I know that the correct translation of “Pürrierstab” would be “immersion blender”, but I am just using “blender”:

```

115 \DeclareRobustCommand{\Purierstab}[1][1]{%
116 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.07ex*#1,scale=#1]
117 \draw[rounded corners=0.07ex*#1] (0,0) -- (0.3,0) -- (0.15,0.1) --cycle;
118 \fill[rounded corners=0.07ex*#1] (0.15,0.3) -- (0.24,0.4) -- (0.24,0.7) --
119 (0.06,0.7) -- (0.06,0.4) -- cycle;
120 \draw (0.15,0.4) -- (0.15,0.1);
121 \end{tikzpicture}%
122 \tikzsymbolsaftersymbolinput%
123 }
124 \let\blender\Purierstab

```

`\Dreizack = \trident` Important for cooking:

```

125 \DeclareRobustCommand{\Dreizack}[1][1]{%
126 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.035ex*#1,scale=#1]
127 \fill[rounded corners=0.07ex*(#1-.#1)] (0,0) -- (0,0.4) -- (0.1,0.4)
128 -- (0.1,0.0) -- cycle;
129 \draw (0.05,0) -- (0.05,0.7);
130 \draw[rounded corners=0.07ex*(#1-.#1*#1*2)] (0,0.7) -- (0,0.55)
131 -- (0.05,0.55) -- (0.1,0.55) -- (0.1,0.7);
132 \end{tikzpicture}%
133 \tikzsymbolsaftersymbolinput%
134 }
135 \let\trident\Dreizack

```

`\Backblech = \bakingplate` I may overdo it a little bit ...:

```

136 \DeclareRobustCommand{\Backblech}[1][1]{%
137 \begin{tikzpicture}[x=6.53ex,y=5.ex, line width=0.07ex*#1,scale=#1]
138 \filldraw[rounded corners=0.09ex*#1] (0,0) rectangle (0.3,0.3);
139 \draw[rounded corners=0.07ex*#1, line width=0.03ex*#1]
140 (0.1,0) -- (-0.025,0) -- (-0.025,0.3) -- (0.1,0.3);
141 \draw[rounded corners=0.07ex*#1, line width=0.03ex*#1]
142 (0.2,0) -- (.325,0) -- (.325,0.3) -- (0.2,0.3);
143 \foreach \@BackblechlochX in {0.007,0.293}
144 \foreach \@BackblechlochY in {0.007,0.293}

```

```

145 \fill[white] (\@BackblechlochX,
146 \@BackblechlochY) circle (0.02ex);
147 \end{tikzpicture}%
148 \tikzsymbolsaftersymbolinput%
149 }
150 \let\bakingplate\Backblech

\Ofen = \oven I may overdo it a little bit 2:
151 \DeclareRobustCommand{\Ofen}[1][1]{%
152 \begin{tikzpicture}[x=0.50ex,y=.5ex, line width=0.07ex*#1,scale=#1]
153 \draw (0,0) rectangle (4,3);
154 \draw (0.25,0.25) rectangle (3.75,2);
155 \foreach \Ofenschalter in {0.5,1.1,2.9,3.5}
156 \fill (\Ofenschalter,2.5) circle (0.22);
157 \draw (1.5,2.28) rectangle (2.5,2.72);
158 \draw[line width=0.05ex*#1] (1,1.75) -- (3,1.75);
159 \end{tikzpicture}%
160 \tikzsymbolsaftersymbolinput%
161 }
162 \let\oven\Ofen

\Pfanne = \pan I can't think of a better word ...:
163 \DeclareRobustCommand{\Pfanne}[1][1]{%
164 \begin{tikzpicture}[x=2.3ex,y=2.3ex, line width=0.09ex*#1,scale=#1]
165 \draw [rounded corners=0.023ex*#1] (0,0) -- (0.9,0) -- (1,0.3) -- (-0.1,0.3) -- cycle;
166 \draw (-0.2,0.22) -- (-0.08,0.22);
167 \draw (0.97,0.22) -- (1.08,0.22);
168 \draw[decorate,decoration={snake,amplitude=.046ex*#1,
169 segment length=0.82ex*#1},line width=0.05ex*#1] (-0.05,0.1) -- (0.95,0.1);
170 \end{tikzpicture}%
171 \tikzsymbolsaftersymbolinput%
172 }
173 \let\pan\Pfanne

\Herd = \cooker I hope it' the right translation:
174 \DeclareRobustCommand{\Herd}[1][1]{%
175 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.04ex*#1,scale=#1]
176 \draw[line width=0.08ex*#1] (0,0) rectangle (2,1.5);
177 \draw (0.5,0.45) circle (0.35);
178 \draw (0.5,0.45) circle (0.2);
179 \draw (1.45,0.45) circle (0.3);
180 \draw (0.5,1.15) circle (0.21);
181 \draw (1.05,0.95) rectangle (1.85,1.35);
182 \draw (1.45,1.15) circle (0.15);
183 \end{tikzpicture}%
184 \tikzsymbolsaftersymbolinput%
185 }
186 \let\cooker\Herd

```

```

\Saftpresse = \squeezer It's an old squeezer:
187 \DeclareRobustCommand{\Saftpresse}[1][1]{%
188 \begin{tikzpicture}[x=1.2ex,y=1ex,line width=0.07ex*#1,scale=#1]
189 \draw[rounded corners=0.1ex*#1] (0,0.85) -- (0,0) -- (1.5,0) -- (1.5,0.85) -- cycle;
190 \draw (0,0.7) -- (1.5,0.7);
191 \draw[rounded corners=0.1ex*#1] (0.3,0.7) -- (0.75,1.55) -- (1.2,0.7);
192 \draw[rounded corners=0.1ex*#1] (0.45,0.7) -- (0.75,1.55) -- (1.05,0.7);
193 \draw[rounded corners=0.1ex*#1] (0.65,0.7) -- (0.75,1.55) -- (0.85,0.7);
194 \draw[line width=0.05ex*#1, decorate,
195   decoration={snake,amplitude=.05ex*#1,segment length=0.48ex*#1}] (0,0.3) -- (1.5,0.3);
196 \end{tikzpicture}%
197 \tikzsymbolsaftersymbolinput%
198 }
199 \let\squeezer\Saftpresse

```

\Schussel = \bowl It may looks a bit queery, but I like it. Wieder dasselbe mit den Umlauten: ü=u.

```

200 \DeclareRobustCommand{\Schussel}[1][1]{%
201 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex*#1, scale=#1]
202 \draw[rounded corners=0.5ex*#1]
203   (-0.02,1.4) -- (0,1.4) -- (0,0.05) -- (1.5,0.05) -- (1.5,1.4) -- (1.52,1.4);
204 \draw (0.35,0) -- (1.15,0);
205 \end{tikzpicture}
206 \tikzsymbolsaftersymbolinput%
207 }
208 \let\bowl\Schussel

```

\Schaler = \peeler I can't believe I forgot this command. I made it and forgot to copy and paste it inside this document!!!! Jedenfalls wieder ä=a:

```

209 \DeclareRobustCommand{\Schaler}[1][1]{%
210 \begin{tikzpicture}[x=2.7ex,y=2.3ex, line width=0.07ex*#1,scale=#1]
211 \draw[rounded corners=0.07ex*#1] (0,0.4) -- (0,0.1) arc (0:180:-0.1) -- (0.2,0.4)
212 -- (0.3,0.5) -- (0.3,0.65) -- (0.2,0.65) -- (0.2,0.5) -- (0,0.5) -- (0,0.65) --
213 (-0.1,0.65) -- (-0.1,0.5) -- cycle;
214 \draw[line width=0.03ex*#1] (0,0.6) -- (0.2,0.6);
215 \draw[line width=0.03ex*#1] (0,0.58) -- (0.2,0.58);
216 \end{tikzpicture}%
217 \tikzsymbolsaftersymbolinput%
218 }
219 \let\peeler\Schaler

```

## 6.2 Emoticonscode

\Sadey \dSadey An other name of Sadey is Frowny, but I named it Sadey because there are enough Frownys in the world. All “3D” Emoticons start with \d..., and all Emoticons end with an “ey” (exception: “Cat”). The “default color” of the 2D Emoticons is opacity=0, with that default option, it's useful for \colorbox{yellow}{\Sadey} which leads to ☹ instead of ☹ (with default=white).

```

220 \DeclareRobustCommandx{\Sadey}[2][1=1, 2={opacity=0}, usedefault]{%

```

```

221 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
222 \fill[#2, line width=0.12ex*#1] (0,0) circle (0.33);
223 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
224 \fill (0.1,0.1) circle (0.05);
225 \fill (-0.1,0.1) circle (0.05);
226 \draw (-0.2,-0.11) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.11);
227 \end{tikzpicture}%
228 \tikzsymbolsaftersymbolinput%
229 }
230 \DeclareRobustCommand{\dSadey}[2][1=1,2=yellow,usedefault]{%
231 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
232 \shade[ball color=#2] (0,0) circle (0.33);
233 \shade[ball color=black] (0.1,0.1) circle (0.05);
234 \shade[ball color=black] (-0.1,0.1) circle (0.05);
235 \draw[black] (-0.2,-0.11) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.11);
236 \end{tikzpicture}%
237 \tikzsymbolsaftersymbolinput%
238 }

```

\Annoey \dAnnoey An annoyed Smiley -\_-

```

239 \DeclareRobustCommand{\Annoey}[2][1=1,2={opacity=0},usedefault]{%
240 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
241 \fill[#2, line width=0.12ex*#1] (0,0) circle (0.33);
242 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
243 \draw (0.08,0.1) -- (0.22,0.1);
244 \draw (-0.08,0.1) -- (-0.22,0.1);
245 \draw (-0.2,-0.1) -- (0.2,-0.1);
246 \end{tikzpicture}%
247 \tikzsymbolsaftersymbolinput%
248 }
249 \DeclareRobustCommand{\dAnnoey}[2][1=1,2=yellow,usedefault]{%
250 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
251 \shade[ball color=#2] (0,0) circle (0.33);
252 \draw[black] (0.08,0.1) -- (0.22,0.1);
253 \draw[black] (-0.08,0.1) -- (-0.22,0.1);
254 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
255 \end{tikzpicture}%
256 \tikzsymbolsaftersymbolinput%
257 }

```

\Smiley \dSmiley A normal Smiley

```

258 \DeclareRobustCommand{\Smiley}[2][1=1,2={opacity=0},usedefault]{%
259 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
260 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.33);
261 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
262 \fill (-0.1,0.1) circle (0.05);
263 \fill (0.1,0.1) circle (0.05);
264 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
265 \end{tikzpicture}%
266 \tikzsymbolsaftersymbolinput%

```

```

267 }
268 \DeclareRobustCommand{\dSmiley}[3][1=1,2=yellow,3=yellow,usedefault]{%
269 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
270 \shade[ball color=#2] (0,0) circle (0.33);
271 \shade[ball color=black] (-0.1,0.1) circle (0.05);
272 \shade[ball color=black] (0.1,0.1) circle (0.05);
273 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
274 \end{tikzpicture}%
275 \tikzsymbolsaftersymbolinput%
276 }

```

\Laughey \dLaughey A laughing Smiley

```

277 \DeclareRobustCommand{\Laughey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
278 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
279 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.33);
280 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
281 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
282 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
283 \fill[#3,rounded corners=0.1ex*#1, yshift=-0.5]
284 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
285 \draw[rounded corners=0.1ex*#1, yshift=-0.5]
286 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
287 \end{tikzpicture}%
288 \tikzsymbolsaftersymbolinput%
289 }
290 \DeclareRobustCommand{\dLaughey}[3][1=1,2=yellow, 3=red ,usedefault]{%
291 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
292 \fill[ball color=#2,line width=0.12ex*#1] (0,0) circle (0.33);
293 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
294 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
295 \shade[ball color=#3, rounded corners=0.1ex*#1, yshift=-0.3]
296 (-0.25,-0.0) .. controls (-0.13,-0.26) and (0.13,-0.26) .. (0.25,-0.0) -- cycle;
297 \end{tikzpicture}%
298 \tikzsymbolsaftersymbolinput%
299 }

```

\Neutrey \dNeutrey neutral Smiley :|

```

300 \DeclareRobustCommand{\Neutrey}[2][1=1, 2={opacity=0}, usedefault]{%
301 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
302 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.33);
303 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
304 \fill (0.1,0.1) circle (0.05);
305 \fill (-0.1,0.1) circle (0.05);
306 \draw (-0.2,-0.1) -- (0.2,-0.1);
307 \end{tikzpicture}%
308 \tikzsymbolsaftersymbolinput%
309 }
310 \DeclareRobustCommand{\dNeutrey}[2][1=1,2=yellow,usedefault]{%
311 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
312 \shade[ball color=#2] (0,0) circle (0.33);

```

```

313 \shade[ball color=black] (0.1,0.1) circle (0.05);
314 \shade[ball color=black] (-0.1,0.1) circle (0.05);
315 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
316 \end{tikzpicture}%
317 \tikzsymbolsaftersymbolinput%
318 }

\Winkey \dWinkey ;)

319 \DeclareRobustCommand{\Winkey}[2][1,2={opacity=0} ,usedefault]{%
320 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
321 \fill[#2, line width=0.12ex*#1] (0,0) circle (0.33);
322 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
323 \draw(0.17,0.1) -- (0.05,0.1);
324 \fill (-0.1,0.1) circle (0.05);
325 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
326 \end{tikzpicture}%
327 \tikzsymbolsaftersymbolinput%
328 }

329 \DeclareRobustCommand{\dWinkey}[2][1,2=yellow,usedefault]{%
330 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
331 \shade[ball color=#2] (0,0) circle (0.33);
332 \draw(0.17,0.1) -- (0.05,0.1);
333 \shade[ball color=black] (-0.1,0.1) circle (0.05);
334 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
335 \end{tikzpicture}%
336 \tikzsymbolsaftersymbolinput%
337 }

\Sey \dSey I can't think of a better name :S

338 \DeclareRobustCommand{\Sey}[2][1,2={opacity=0} ,usedefault]{%
339 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
340 \fill[#2, line width=0.12ex*#1] (0,0) circle (0.33);
341 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
342 \fill (0.1,0.1) circle (0.05);
343 \fill (-0.1,0.1) circle (0.05);
344 \draw (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
345 \end{tikzpicture}%
346 \tikzsymbolsaftersymbolinput%
347 }

348 \DeclareRobustCommand{\dSey}[2][1,2=yellow ,usedefault]{%
349 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1]
350 \shade[ball color=#2] (0,0) circle (0.33);
351 \shade[ball color=black] (0.1,0.1) circle (0.05);
352 \shade[ball color=black] (-0.1,0.1) circle (0.05);
353 \draw[black] (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
354 \end{tikzpicture}%
355 \tikzsymbolsaftersymbolinput%
356 }

\Innocey \dInnocey An innocent Smiley

```

```

357 \DeclareRobustCommand{\Innocey}[3][1=1,2={opacity=0},3=yellow,usedefault]{%
358 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
359 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.33);
360 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
361 \fill (-0.1,0.1) circle (0.05);
362 \fill (0.1,0.1) circle (0.05);
363 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
364 \draw[#3, line width=0.095ex*#1] (0.32,0.31) arc (0:360:0.32 and 0.1);
365 \draw[line width=0.05ex*#1] (0.3,0.31) arc (0:360:0.3 and 0.07);
366 \draw[line width=0.05ex*#1] (0.35,0.31) arc (0:360:0.35 and 0.12);
367 \end{tikzpicture}}%
368 \tikzsymbolsaftersymbolinput%
369 }
370 \DeclareRobustCommand{\wInnocey}[1][1]{\Innocey[#1][opacity=0][white]}
371 \DeclareRobustCommand{\dInnocey}[3][1=1,2=yellow,3=yellow,usedefault]{%
372 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
373 \shade[ball color=#2] (0,0) circle (0.33);
374 \shade[ball color=black] (-0.1,0.1) circle (0.05);
375 \shade[ball color=black] (0.1,0.1) circle (0.05);
376 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
377 \draw[color=#3!97!black, line width=0.09ex*#1] (0.32,0.31) arc (0:360:0.32 and 0.1);
378 \draw[line width=0.05ex*#1] (0.3,0.31) arc (0:360:0.3 and 0.07);
379 \draw[line width=0.05ex*#1] (0.35,0.31) arc (0:360:0.35 and 0.12);
380 \end{tikzpicture}}%
381 \tikzsymbolsaftersymbolinput%
382 }

```

\Cooley \dCooley Don't know what I shall write here.

```

383 \DeclareRobustCommand{\Cooley}[2][1=1,2={opacity=0},usedefault]{%
384 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
385 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.33);
386 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
387 \fill[rounded corners=0.1ex*#1]
388 (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
389 \fill[rounded corners=0.1ex*#1]
390 (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
391 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
392 \draw (0.02,0.1) -- (-0.02,0.1);
393 \draw (-0.2,0.1) -- (-0.3,0.13);
394 \draw (0.2,0.1) -- (0.3,0.13);
395 \end{tikzpicture}}%
396 \tikzsymbolsaftersymbolinput%
397 }
398 \DeclareRobustCommand{\dCooley}[2][1=1,2=yellow,usedefault]{%
399 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
400 \shade[ball color=#2] (0,0) circle (0.33);
401 \draw[black] (0.02,0.1) -- (-0.02,0.1);
402 \draw[black] (-0.2,0.1) -- (-0.295,0.146);
403 \draw[black] (0.2,0.1) -- (0.295,0.146);
404 \shade[ball color=black,rounded corners=0.1ex*#1]

```



```

405 (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
406 \shade[ball color=black,rounded corners=0.1ex*#1]
407 (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
408 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
409 \end{tikzpicture}%
410 \tikzsymbolsaftersymbolinput%
411 }

```

\Tongey \dTongey :P

```

412 \DeclareRobustCommand{\Tongey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
413 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
414 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.33);
415 \draw[line width=0.12ex*#1] (0,0) circle (0.33);
416 \fill (-0.1,0.1) circle (0.05);
417 \fill (0.1,0.1) circle (0.05);
418 \fill[#3,line width=0.058ex*#1, rounded corners=0.12ex*#1]
419 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
420 \draw[line width=0.07ex*#1, yshift=0.21ex]
421 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
422 \draw[line width=0.058ex*#1, rounded corners=0.11ex*#1]
423 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
424 \end{tikzpicture}%
425 \tikzsymbolsaftersymbolinput%
426 }
427 \DeclareRobustCommand{\dTongey}[3][1=1,2=yellow,3=red,usedefault]{%
428 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*#1,scale=#1]
429 \shade[ball color=#2] (0,0) circle (0.33);
430 \shade[ball color=black] (-0.1,0.1) circle (0.05);
431 \shade[ball color=black] (0.1,0.1) circle (0.05);
432 \shade[ball color=#3,line width=0.058ex*#1, rounded corners=0.12ex*#1]
433 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
434 \draw[black, line width=0.058ex*#1, rounded corners=0.12ex*#1]
435 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
436 \draw[black, line width=0.07ex*#1, yshift=0.21ex]
437 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
438 \end{tikzpicture}%
439 \tikzsymbolsaftersymbolinput%
440 }

```

\Nursesey \dNursesey a Nurse (the cross has nothing to do with religion).

```

441 \DeclareRobustCommand{\Nursesey}[4][1=1,2={opacity=0},3={opacity=0},4=black,usedefault]{%
442 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*#1,scale=#1]
443 \fill[#3,rounded corners=.023ex*#1]
444 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
445 \fill[#2,line width=0.12ex*#1] (0,0) circle (0.3);
446 \draw[line width=0.12ex*#1] (0,0) circle (0.3);
447 \fill (-0.1,0.1) circle (0.05);
448 \fill (0.1,0.1) circle (0.05);
449 \draw[line width=0.09ex*#1, yshift=0.07ex]
450 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);

```

```

451 \draw[rounded corners=.023ex*#1]
452   (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
453 \draw[#4,line width=.046ex*#1] (0,0.35) -- (0,0.5);
454 \draw[#4,line width=.046ex*#1] (-0.05,0.45) -- (0.05,0.45);
455 \end{tikzpicture}%
456 \tikzsymbolsaftersymbolinput%
457 }
458 \DeclareRobustCommand{\dNursey}[4][1=1,2=yellow,3=white,4=red,usedefault]{%
459 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*#1,scale=#1]
460 \shade[ball color=#2] (0,0) circle (0.3);
461 \shade[ball color=black] (-0.1,0.1) circle (0.05);
462 \shade[ball color=black] (0.1,0.1) circle (0.05);
463 \draw[black, line width=0.09ex*#1, yshift=0.07ex]
464   (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
465 \shade[ball color=#3, rounded corners=.023ex*#1,yshift=-0.09ex]
466   (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0) arc(0:180:0.3);
467 \shade[ball color=#4,line width=.046ex*#1]
468   (-0.01,0.31) -- (-0.01,0.46) -- (0.01,0.46) -- (0.01,0.31)--cycle;
469 \shade[ball color=#4,line width=.046ex*#1]
470   (-0.05,0.4) -- (0.05,0.4) -- (0.05,0.42)--(-0.05,0.42) -- cycle;
471 \end{tikzpicture}%
472 \tikzsymbolsaftersymbolinput%
473 }

\Vomey \dVomey *Bläärgh*
474 \DeclareRobustCommand{\Vomey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
475 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*#1,scale=#1]
476 \fill[#2,rounded corners=0.05ex*#1] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
477 \draw[rounded corners=0.05ex*#1] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
478 \draw[line width=0.05ex*#1] (-0.5,0.3) -- (-0.3,0.1);
479 \fill (-0.45,0.27) arc (100:350:0.1);
480 \fill[#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
481   controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
482   controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
483 \fill[#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) ..
484   controls (1.7,-1.2) and (1.3,-1.2) .. (1,-1) ..
485   controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
486 \draw (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1);
487 \draw (0,-0.4) .. controls (0.5,-0.5) and (0.8,-0.7) .. (1,-1);
488 \draw (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
489   controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2)
490   and (1,-2) .. (1,-1.5) .. controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
491 \end{tikzpicture}%
492 \tikzsymbolsaftersymbolinput%
493 }
494 \DeclareRobustCommand{\dVomey}[3][1=1,2=yellow,3={brown!10!olive},usedefault]{%
495 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*#1,scale=#1]
496 \shade[ball color=#2!90!brown,rounded corners=0.03ex*#1]
497   (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
498 \draw[black, line width=0.05ex*#1] (-0.5,0.3) -- (-0.3,0.1);

```

```

499 \shade[ball color=black] (-0.45,0.27) arc (100:350:0.1);
500 \shade[ball color=#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
501 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
502 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
503 \shade[ball color=#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) .. controls
504 (1.7,-1.2) and (1.3,-1.2) .. (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
505 \end{tikzpicture}%
506 \tikzsymbolsaftersymbolinput%
507 }

```

\Walley \dWalley Well ... this Emoticon should be the visualization of the german saying “Gegen eine Wand rennen”, which means something like: Not being able to solve a problem.

```

508 \DeclareRobustCommand{\Walley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
509 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1,
510 decoration={random steps,segment length=0.15ex*#1, amplitude=0.1ex*#1}]
511 \fill[#2, line width=0.08ex*#1] (0,0) circle (0.28);
512 \draw[line width=0.08ex*#1] (0,0) circle (0.28);
513 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
514 \draw (0.28,-0.33) rectangle (0.66,0.33);
515 \draw[line width=0.06ex*#1] (0.28,0) --++(0.05,0.07) --++(0.03,0.02) --+
516 +(0.03,-0.02) --++(0.03,0.1) --++(0.03,0.02) -- (0.5,0.25);
517 \draw[line width=0.06ex*#1] (0.28,0) --++(0.06,-0.02) --++(0.04,0.06) --+
518 +(0.0,-0.08) --++(0.08,0.06) --++(0.03,-0.02) --+(0.08,0.02) -- (0.6,0.0);
519 \draw[line width=0.06ex*#1] (0.28,0) --++(0.03,-0.02) --++(0.03,-0.07) --+
520 +(0.03,-0.01) --++(0.01,-0.07) --++(0.06,0.01) --++(0.03,-0.08) --
521 (0.5,0.-0.25);
522 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
523 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
524 \end{tikzpicture}%
525 \tikzsymbolsaftersymbolinput%
526 }
527 \DeclareRobustCommand{\rWalley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
528 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1,
529 decoration={random steps,segment length=0.15ex*#1, amplitude=0.1ex*#1}]
530 \fill[#2, line width=0.08ex*#1] (0,0) circle (0.28);
531 \draw[line width=0.08ex*#1] (0,0) circle (0.28);
532 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
533 \draw (0.28,-0.33) rectangle (0.66,0.33);
534 \draw[decorate, line width=0.06ex*#1] (0.28,0) -- (0.5,0.25);
535 \draw[decorate,line width=0.06ex*#1] (0.28,0) -- (0.6,0.0);
536 \draw[decorate,line width=0.06ex*#1] (0.28,0) -- (0.5,-0.25);
537 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
538 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
539 \end{tikzpicture}%
540 \tikzsymbolsaftersymbolinput%
541 }
542 \DeclareRobustCommand{\dWalley}[2][1=1, 2=yellow, usedefault]{%
543 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1,
544 decoration={random steps,segment length=0.15ex*#1, amplitude=0.1ex*#1}]
545 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);

```

```

546 \draw[line width=0.06ex*#1] (0.28,0) --++(0.05,0.07) --++(0.03,0.02) --+
547 +(0.03,-0.02) --++(0.03,0.1) --++(0.03,0.02) -- (0.5,0.25);
548 \draw[line width=0.06ex*#1] (0.28,0) --++(0.06,-0.02) --++(0.04,0.06) --+
549 +(0.0,-0.08) --++(0.08,0.06) --++(0.03,-0.02) --++(0.08,0.02) -- (0.6,0.0);
550 \draw[line width=0.06ex*#1] (0.28,0) --++(0.03,-0.02) --++(0.03,-0.07) --+
551 +(0.03,-0.01) --++(0.01,-0.07) --++(0.06,0.01) --++(0.03,-0.08) -- (0.5,0.-0.25);
552 \shade[ball color=#2, line width=0.08ex*#1] (-0.01,0) circle (0.31);
553 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
554 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
555 \end{tikzpicture}%
556 \tikzsymbolsaftersymbolinput%
557 }
558 \DeclareRobustCommandx{\drWalley}[2][1=1, 2=yellow, usedefault]{%
559 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1,
560 decoration={random steps,segment length=0.15ex*#1, amplitude=0.1ex*#1}]
561 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
562 \draw[decorate, line width=0.06ex*#1] (0.298,0) -- (0.5,0.25);
563 \draw[decorate,line width=0.06ex*#1] (0.298,0) -- (0.6,0.0);
564 \draw[decorate,line width=0.06ex*#1] (0.298,0) -- (0.5,-0.25);
565 \shade[ball color=#2, line width=0.08ex*#1] (-0.01,0) circle (0.31);
566 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
567 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
568 \end{tikzpicture}%
569 \tikzsymbolsaftersymbolinput%
570 }

```

\Cat \*Miau\*

```

571 \DeclareRobustCommandx{\Cat}[1][1=1,usedefault]{%
572 \begin{tikzpicture}[x=2.33ex,y=2.33ex, line width=0.093ex*#1,scale=#1]
573 \draw (0,0) circle (0.3);
574 \draw[rounded corners=0.163ex*#1] (-0.3,0) -- (-0.35,0.5) -- (0,0.3);
575 \draw[rounded corners=0.163ex*#1] (0,0.3) -- (0.35,0.5) -- (0.3,0);
576 \fill (-0.15,.15) circle (0.05);
577 \fill (0.15,.15) circle (0.05);
578 \draw[rounded corners=0.175ex*#1,yshift=-0.12ex]
579 (0,0) -- (0,-0.1) -- (-0.1,-0.095);
580 \draw[rounded corners=0.175ex*#1,yshift=-0.12ex]
581 (0,0) -- (0,-0.1) -- (0.1,-0.095);
582 \draw[rounded corners=.12ex*#1,yshift=-.15ex, line width=0.03em*(#1-.#1)]
583 (-0.1,0.1) -- (0,0) -- (0.1,0.1) -- cycle ;
584 \draw[line width=0.035ex*#1](-0.1,-0.05)..controls(-0.25,0)and(-0.35,0).. (-0.4,-0.05);
585 \draw[line width=0.035ex*#1, ](-0.1,-0.05)..
586 controls(-0.25,-0.01)and(-0.35,-0.09).. (-0.4,-0.14);
587 \draw[line width=0.035ex*#1, ](-0.1,-0.05)..
588 controls(-0.25,-0.045)and(-0.35,-0.13).. (-0.4,-0.23);
589 \draw[line width=0.035ex*#1](0.1,-0.05)..controls(0.25,0)and(0.35,0).. (0.4,-0.05);
590 \draw[line width=0.035ex*#1 ]
591 (0.1,-0.05)..controls(0.25,-0.01)and(0.35,-0.09).. (0.4,-0.14);
592 \draw[line width=0.035ex*#1]
593 (0.1,-0.05)..controls(0.25,-0.045)and(0.35,-0.13).. (0.4,-0.23);

```

```

594 \end{tikzpicture}%
595 \tikzsymbolsaftersymbolinput%
596 }

\Ninja \dNinja A Ninja.

597 \DeclareRobustCommand{\Ninja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
598 \def\Black@is@Black{black}%
599 \def\Black@or@not@Black{#2}%
600 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1,
601 decoration={random steps,segment length=0.1ex*#1, amplitude=0.1ex*#1}]
602 \fill[#2, line width=0.08ex*#1] (0,0) circle (0.33);
603 \draw (-0.2,-0.125) -- ++(0.4,0);
604 \fill[decoration={random steps,segment length=0.1ex*#1, amplitude=0.01ex*#1}, decorate,#3]
605 (-0.33,0) -- (0.33,0) -- (0.23,0.23) -- (-0.23,0.23) -- cycle;
606 \ifx\Black@or@not@Black\Black@is@Black
607 \draw[line width=0.08ex*#1] (0,0) circle (0.33);\fi
608 \fill[#3] (0,0.1) -- (-0.33,0) -- (-0.26,0.23);
609 \fill[#3] (0.3465,0) arc (0:42:0.34 and 0.345) --
610 (0.2,0.23)-- (0.31,0.0) -- cycle;
611 \fill[#3] (-0.3465,0) arc (0:-42:-0.34 and -0.345) --
612 (-0.2,0.23)-- (-0.31,0.0) -- cycle;
613 \fill[#4] (0.129,0.1425) arc (55:-180:.05);
614 \fill[#4] (-0.129,0.1425) arc (-55:180:-.05);
615 \draw[ decorate,decoration={snake,amplitude=.1ex*#1,segment length=0.55ex*#1},decorate, #3]
616 (0.26,0.21) -- (0.5,0.35);
617 \draw[ decorate,decoration={snake,amplitude=.1ex*#1,segment length=0.5ex*#1},decorate, #3]
618 (0.26,0.21) -- (0.53,0.1);
619 \ifx\Black@or@not@Black\Black@is@Black
620 \else\draw[line width=0.08ex*#1] (0,0) circle (0.33);\fi
621 \end{tikzpicture}%
622 \tikzsymbolsaftersymbolinput%
623 }

624 \DeclareRobustCommand{\dNinja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
625 \def\Black@is@Black{black}%
626 \def\Black@or@not@Black{#2}%
627 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*#1,scale=#1,
628 decoration={random steps,segment length=0.1ex*#1, amplitude=0.1ex*#1}]
629 \draw[ decorate,decoration={snake,amplitude=.1ex*#1,
630 segment length=0.55ex*#1},decorate, #3!50!black] (0.26,0.21) -- (0.5,0.35);
631 \draw[ decorate,decoration={snake,amplitude=.1ex*#1,
632 segment length=0.5ex*#1},decorate, #3!50!black] (0.26,0.21) -- (0.53,0.1);
633 \shade[ball color=#2, line width=0.08ex*#1] (0,0) circle (0.347);
634 \draw (-0.2,-0.125) -- ++(0.4,0);
635 \ifx\Black@or@not@Black\Black@is@Black
636 \draw[line width=0.08ex*#1] (0,0) circle (0.33);\fi
637 \fill[decoration={random steps,segment length=0.1ex*#1, amplitude=0.01ex*#1},ball color=#3]
638 decorate {(-0.33,0) -- (0.3465,0) }
639 {arc (0:42:0.34 and 0.345)}
640 decorate {-- (-0.25,0.24)}
641 { arc (-42:0:-0.375 and -0.345)};

```

```

642 \shade[ball color=#4] (0.129,0.1425) arc (55:-180:.05);
643 \shade[ball color=#4] (-0.129,0.1425) arc (-55:180:-.05);
644 \shade[top color=#4!80!black, bottom color=#4] (0.129,0.1425) arc (55:-180:.05);
645 \shade[top color=#4!80!black, bottom color=#4] (-0.129,0.1425) arc (-55:180:-.05);
646 \end{tikzpicture}%
647 \tikzsymbolsaftersymbolinput%
648 }

```

\NiceReapey Not very well made. But it's better than nothing

```

649 \DeclareRobustCommand{\NiceReapey}[1][1=1,usedefault]{%
650 \begin{tikzpicture}[x=0.11em,y=0.11em, line width=0.07ex*#1,scale=#1]
651 \draw (1.7,-1) arc (360:180:1.7 and 2)
652   arc (260:110:1.5 and 2) .. controls (-1,3.3) and (1,3.3) .. (1.9,2.97)
653   arc (260:100:-1.3 and -2) -- cycle;
654 \fill[black!20!white] (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5) -- cycle;
655 \draw (3,-3) -- (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5);
656 \draw (0,-1.5) circle (1 and 0.5);
657 \draw[line width=0.04ex*#1] (-0.2,-1) -- (-0.2,-2);
658 \draw[line width=0.04ex*#1] (0.2,-1) -- (0.2,-2);
659 \draw[line width=0.04ex*#1] (0.6,-1) -- (0.6,-2);
660 \draw[line width=0.04ex*#1] (-0.6,-1) -- (-0.6,-2);
661 \draw[line width=0.04ex*#1] (-1,-1.5) -- (1,-1.5);
662 \fill (1.25,1.25) circle (0.5 and 0.75);
663 \fill (-1.25,1.25) circle (0.5 and 0.75);
664 \end{tikzpicture}%
665 \tikzsymbolsaftersymbolinput%
666 }

```

## 7 Other symbols(s)

\Person My first symbol: a person. In german it would be called "Strichmaxerl".

```

667 \DeclareRobustCommand{\Person}[1][1]{%
668 \begin{tikzpicture}[line width=0.12ex*#1,scale=#1,x=1.35ex,y=1.35ex]%
669 \draw (0,-0.1) -- (0.15,0.2) -- (0.3,-0.1);
670 \draw (.15,.2) -- (.15,.4);
671 \draw (.15,.4) -- (.4,.5);
672 \draw (.15,.4) -- (-0.1,.5);
673 \draw (.15,.4) -- (.15,.53);
674 \draw (.15,.8) circle (0.18);
675 \end{tikzpicture}%
676 \tikzsymbolsaftersymbolinput%
677 }

```

\candle A burning candle

```

678 \DeclareRobustCommand{\Candle}[1][1]{%
679 \begin{tikzpicture}[x=1ex, y=1ex, scale=#1, line width=0.07ex*#1]
680 \draw[rounded corners=0.04ex*#1] (0,0) -- (0.2,0) -- +(0,1) -- (0,1) -- cycle;
681 \draw[line width=0.05ex*#1] (0.1,1) -- (0.1,1.2);

```

```

682 \draw[xshift=0.95, yshift=2.2, line width=0.04ex*#1]
683   (-0.1,0.6) .. controls (-0.4,0.8) and (-0.1,1) .. (-0.1,1.2);
684 \draw [xshift=0.95, yshift=2.2, line width=0.04ex*#1]
685   (-0.1,0.6) .. controls (0.2,0.8) and (-0.1,1) .. (-0.1,1.2);
686 \end{tikzpicture}%
687 \tikzsymbolsaftersymbolinput%
688 }

```

\fire Just a fire.

```

689 \DeclareRobustCommand{\Fire}[1][1]{%
690 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.07ex*#1,rotate=45, scale=#1]
691 \fill (-0.05,0) -- (0.05,0) -- (0.05,0.95) -- (-0.05,0.95) -- cycle;
692 \fill (-0.74,0.7) -- (0.19,0.7) -- (0.19,0.8) -- (-0.74,0.8) -- cycle;
693 \fill[rotate=-20, xshift=-1.3, yshift=-0.1]
694   (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
695 \fill[rotate=-70, xshift=-3.3, yshift=-2.3]
696   (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
697 \fill[rotate=135, xshift=2.5, yshift=-3.8]
698   (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
699 \draw[rotate=-45, xshift=-2.6, yshift=1.5,line width=0.04ex*#1, x=0.5ex, y=0.5ex]
700 (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0.05,1.7);
701 \draw[rotate=-45, xshift=-2.1,yshift=1.5,line width=0.04ex*#1, x=0.5ex, y=0.5ex]
702 (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.15,1.7);
703 \draw[rotate=-45, xshift=-2.5] (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0,1.5);
704 \draw[rotate=-45, xshift=-2] (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.1,1.5);
705 \end{tikzpicture}%
706 \tikzsymbolsaftersymbolinput%
707 }

```

\cofeecup Just a cup of coffee.

```

708 \DeclareRobustCommand{\Coffeecup}[1][1]{%
709 \begin{tikzpicture}[x=0.7ex,y=0.7ex, scale=#1, line width=0.07ex*#1]
710 \draw (0,0) arc (180:269:0.8 and 1) -- ++(0.5,0) arc (269:360:0.8 and 1) -- cycle;
711 \draw (2.1,-0.15) -- (2.2,-0.15) arc (90:-90:0.3) -- (1.8, -0.75);
712 \draw[line width=0.05ex*#1, decorate,
713   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
714   (0.4,0.3) -- +(0,1);
715 \draw[line width=0.05ex*#1, decorate,
716   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
717   (1,0.3) -- +(0,1);
718 \draw[line width=0.05ex*#1, decorate,
719   decoration={snake,amplitude=.05ex*#1,segment length=0.408ex*#1}]
720   (1.6,0.3) -- +(0,1);
721 \draw (0,-1.05) -- (2.1,-1.05);
722 \end{tikzpicture}%
723 \tikzsymbolsaftersymbolinput%
724 }

```

\chair A chair.

```

725 \DeclareRobustCommand{\Chair}[1][1]{%
726 \begin{tikzpicture}[x=0.9ex,y=0.9ex, scale=#1, line width=0.07ex*#1]
727 \draw (0,-0.5) -- (0,0.7) -- (0.5,1) -- (0.5,0.25);
728 \draw[line width=0.06ex*#1] (0,0.4) -- (0.5,0.7);
729 \draw (0,0) -- (0.5,0.3) -- (1,0) --(1,-0.5);
730 \draw (0.5,-0.3) -- (0.5,-0.8);
731 \draw (1,0) -- (0.5,-0.3) -- (0,0);
732 \end{tikzpicture}%
733 \tikzsymbolsaftersymbolinput%
734 }

```

\bed A chair.

```

735 \DeclareRobustCommand{\Bed}[1][1]{%
736 \begin{tikzpicture}[x=1ex,y=1ex, scale=#1, line width=0.08ex*#1]
737 \draw (0,0) -- (0,1.6);
738 \draw (3,0) -- (3,1.2);
739 \draw (0,0.5) -- (3,0.5);
740 \draw (0,0.35) -- (3,0.35);
741 \draw (0.7,0.5) arc (0:90:0.7);
742 \draw (0.7,0.5) arc(180:30:1.231 and 0.6);
743 \end{tikzpicture}%
744 \tikzsymbolsaftersymbolinput%
745 }

```

\tikzsymbolsMoaiscale \*sight\* \ifnum cannot compare decimals and to make it possible to scale Moai with decimals (e.g. 0.6, 2.6, 9.345234) I had to define lengths, because L<sup>A</sup>T<sub>E</sub>X can compare decimals in lengths.

\tikzsymbolsMoaiCheckI At first we define the length \tikzsymbolsMoaiscale: it will contain the scaling number.

\tikzsymbolsMoaiCheckII At second \tikzsymbolsMoaiCheckI: the line width of the Moai; depends on \tikzsymbolsMoaiscale.

\tikzsymbolsMoaiCheckI and \tikzsymbolsMoaiCheckII: to be able to “check” if \tikzsymbolsMoaiscale is greater than 2 or 5 or not.

```

746 \newlength{\tikzsymbolsMoaiscale}
747 \newlength{\tikzsymbolsMoaiCheckI}
748 \newlength{\tikzsymbolsMoaiCheckII}\setlength{\tikzsymbolsMoaiCheckI}{2ex}
749 \newlength{\tikzsymbolsMoaiCheckII}\setlength{\tikzsymbolsMoaiCheckII}{5ex}

```

\Moai From the Easter Island: a Moai.

```

750 \DeclareRobustCommand{\Moai}[1][1=1,usedefault]{%
751 \setlength{\tikzsymbolsMoaiscale}{#1ex}%
752 \ifnum \tikzsymbolsMoaiscale<\tikzsymbolsMoaiCheckI%
753 \setlength{\tikzsymbolsMoaiCheckI}{0.05ex}%
754 \else%
755 \ifnum \tikzsymbolsMoaiscale<\tikzsymbolsMoaiCheckII%
756 \setlength{\tikzsymbolsMoaiCheckII}{0.035ex}%
757 \else%
758 \setlength{\tikzsymbolsMoaiCheckII}{0.03ex}%
759 \fi\fi%

```



```

760 \begin{tikzpicture}[x=.13ex, y=.13ex, rounded corners=0.01ex,scale=#1,
761   line width=\tikzsymbolsMoaithickness*#1]
762 \draw (-2.6,-4.25) -- (-2.5,-5.8)
763 ..controls (-2,-6.8) and (1.5,-6.8) .. (2.2,-5.8) -- (2.4,-3.95);
764 \draw(-2.5,2.5) .. controls (-2.9,4.6) and (2,5) .. (3.3,2.5) -- (2.9,-3.4)
765 .. controls (2,-5) and (-4,-5) .. (-3.1,-3) -- cycle;
766 \draw (-2.5,3) -- (-2,5) .. controls (0,6) and (2,5.8) .. (3.1,4.7) -- (3.3,2.5);
767 \draw[line width=0.02ex*#1] (-2.2,-1.8) .. controls (-1,-1.3) and (0,-1.7) .. (1,-2);
768 \draw[line width=0.02ex*#1] (-2.2,-1.8) .. controls (-1,-1) and (0,-1.4) .. (1,-2);
769 \draw[line width=0.02ex*#1] (-2.2,-1.8) .. controls (-1,-2) and (0,-2) .. (1,-2);
770 \draw (-0.8,4) .. controls (-0.8,3) and (-0.8,2) .. (-1.6,0.5) -- (-1.8,-0.4)
771 .. controls (-1,0.2) and (0,0.2) .. (0.6,-0.4) -- (0.7,0.4)
772 .. controls (0,1) and (0,2) .. (0.8,4);
773 \draw (-1.8,-0.4) .. controls (-0.5,-0.5) and (0,-0.5) .. (0.6,-0.4);
774 \draw (3.2,3.5) -- (3.7,3.5) .. controls (3.5,2) and (3.5,2) .. (3.6,-1.5) -- (3,-1.9);
775 \draw (-2.5,3) .. controls (-2.7,2) and (-3,1) .. (-2.88,-1);
776 \draw (-2.5,2.8) .. controls (-2,2.5) and (-1,3) .. (-0.8,3.1);
777 \draw (0.5,3.3) .. controls (1,3) and (1,2.5) .. (3.3,2.4);
778 \end{tikzpicture}%
779 \tikzsymbolsaftersymbolinput%
780 }

```

## 7.1 Trees

Many great ideas are stolen. Don't know who said that, but it's true.

`\@leaf@is@leaf` I this name ... However, We need this command for creating an error message if the last paramter of `BasicTree` is neither “leaf” nor empty.

```
781 \def\@leaf@is@leaf{leaf}
```

`\if@draft \if@final` We need them for some package warnings.

```
782 \newif\if@tikzsymbols@draft
```

```
783 \newif\if@tikzsymbols@final
```

`\@Tree@SetUp` First we define our `@Tree@SetUp` (how the trees will look like) (I used the code from the “tikz” manual and changed it a little bit):

```

784 \DeclareRobustCommand{\@Tree@SetUp}[1][1]{\tikzset{%
785   ld/.style={level distance=##1ex},lw/.style={line width=##1ex},%
786   level 1/.style={ld=0.60,   trunk,           lw=0.1*#1 ,sibling angle=60},%
787   level 2/.style={ld=0.20,   trunk!80!leaf a,   lw=.073*#1,sibling angle=70},%
788   level 3/.style={ld=0.25,   trunk!60!leaf a,   lw=.05*#1,sibling angle=70},%
789   level 4/.style={ld=0.10,   trunk!40!leaf a,   lw=.025*#1,sibling angle=60},%
790   level 5/.style={ld=0.15,   trunk!20!leaf a,   lw=.02*#1,sibling angle=60},%
791   level 6/.style={ld=0.08,   leaf a,           lw=.021*#1,sibling angle=60},%
792 }}%

```

`\Basic@Tree` Now we define our `\Basic@Tree`. We will need it later for our package option (basic code is also from “tikz” manual).

```
793 \DeclareRobustCommandx{\Basic@Tree}[5][1=1, usedefault]{\%
```

```

794 \def\leaf@or@not@leaf{#5}%
795 \@Tree@SetUp[#1]%
796 \pgfarrowsdeclare{leaf}{leaf}%
797 {\pgfarrowslefttextend{-.1ex} \pgfarrowsrighttextend{-0.05ex}}%
798 {%
799 \pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
800 \pgfpatharc{150}{30}{0.16ex*(#1/2)}% dicke
801 \pgfpatharc{-30}{-150}{0.16ex*(#1/2)}%
802 \pgfusepathqfill%
803 }%
804 \colorlet{trunk}{#2}%
805 \colorlet{leaf a}{#3}%
806 \colorlet{leaf b}{#4}\begin{tikzpicture}[x=1ex,y=1ex, scale=#1+0.2, line width=0.07ex*#1]%
807 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
808 \draw[opacity=0] (-0.82-0.1* #1/100,0) rectangle (0.82+0.1*#1/100, 1.415+0.01*#1/100);
809 \else
810 \draw[opacity=0] (-0.75,0.1) rectangle (0.75,1.3);
811 \fi
812 \coordinate (root) [grow cyclic,rotate=90] child {
813 child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
814 child foreach \c in {0,1,2} { child foreach \d in {0,1} {
815 child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5] }
816 }}} } edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
817 };%
818 \end{tikzpicture}%
819 }}

```

**draft** If the class option `draft`, then Squares instead of trees. Furthermore we set `\@drafttrue` and `\@finalfalse` for some warnings:

```

820 \DeclareOption{draft}{\@tikzsymbols@drafttrue\@tikzsymbols@finalfalse
821 \def\Basic@Tree{\Basic@Tree@off}}

```

**final** If the class option `final`, then trees. Same as before:

```

822 \DeclareOption{final}{\@tikzsymbols@draftfalse\@tikzsymbols@finaltrue
823 \def\Basic@Tree{\Basic@Tree@on}}

```

It's extremely annoying: you are working almost a day to find out how this package recognizes `draft` and at the end there are just two lines of code.

**tree** Now we declare the name of our option: “tree” (I could have named it stone, or wood, etc. but I used “tree”). This code is copy & pasted from this site: <http://tex.stackexchange.com/>. Plus some warnings, if you use class option `draft` or `final` with package option `tree=on/off`:

```

824 \DeclareOption{tree}{%
825 \def\Basic@Tree{\csname Basic@Tree@#1\endcsname}%
826 \if@tikzsymbols@draft%
827 \PackageWarningNoLine{tikzsymbols}{You can use class option \MessageBreak
828 draft with package option tree=on/off;
829 \MessageBreak but I think it would be better if you
830 \MessageBreak delete tree=on/off}\fi%

```

```

831 \if@tikzsymbols@final%
832 \PackageWarningNoLine{tikzsymbols}{You can use class option \MessageBreak
833 final with package option tree=on/off;
834 \MessageBreak but I think it would be better if you
835 \MessageBreak delete tree=on/off}\fi%
836 }

\Basic@Tree@off We define \Basic@Tree@off; it will be shown if tree=off. I looks a bit confusing,
but this syntax provides a square, which is as large as the tree. Furthermore, we
check if the last parameter is "leaf":

837 \DeclareRobustCommand{\Basic@Tree@off}[5][1=1, usedefault]{\%
838 \def\leaf@or@not@leaf{#5}%
839 \def\@y@x@y@x{0.095*#1/100}
840 \begin{tikzpicture}[scale=#1+0.2,x=1.3ex,y=1.3ex, line width=0.07ex*#1]
841 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
842 \draw[#2] (0-\@y@x@y@x,0) -- (0-\@y@x@y@x,1.08+0.05*#1/100);
843 \draw[#3] (0-\@y@x@y@x,1.08+0.05*#1/100) -- (1.2+\@y@x@y@x,1.08+0.05*#1/100);
844 \draw[#4] (1.2+\@y@x@y@x,1.08+0.05*#1/100) -- (1.2+\@y@x@y@x,0);
845 \draw[#3] (1.2+\@y@x@y@x,0) -- (0.5,0);
846 \draw[#4] (0.5+0.4*#1/100,0) -- (0-\@y@x@y@x,0);
847 \else
848 \draw[#2] (0,0) -- (0,1);
849 \draw[#3] (0,1) -- (1.15,1);
850 \draw[#4] (1.15,1) -- +(0,-1);
851 \fi%
852 \end{tikzpicture}%
853 }}

\Basic@Tree@on We define \Basic@Tree@off; it will be shown if tree=on:

854 \DeclareRobustCommand{\Basic@Tree@on}[5][1=1, usedefault]{\%
855 \def\leaf@or@not@leaf{#5}%
856 \@Tree@SetUp[#1]%
857 \pgfarrowsdeclare{leaf}{leaf}%
858 {\pgfarrowsleftextend{-.1ex} \pgfarrowsrightextend{-0.05ex}}%
859 {%
860 \pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
861 \pgfpatharc{150}{30}{0.16ex*(#1/2)}% dicke
862 \pgfpatharc{-30}{-150}{0.16ex*(#1/2)}%
863 \pgfusepathqfill%
864 }%
865 \colorlet{trunk}{#2}%
866 \colorlet{leaf a}{#3}%
867 \colorlet{leaf b}{#4}\begin{tikzpicture}[x=1ex,y=1ex, scale=#1+0.2, line width=0.07ex*#1]%
868 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
869 \draw[opacity=0] (-0.82-0.1* #1/100,0) rectangle (0.82+0.1*#1/100, 1.415+0.01*#1/100);
870 \else
871 \draw[opacity=0] (-0.75,0.1) rectangle (0.75,1.3);
872 \fi
873 \coordinate (root) [grow cyclic,rotate=90] child {

```

```

874 child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
875 child foreach \c in {0,1,2} { child foreach \d in {0,1} {
876 child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5] }
877 }}} } edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
878 };%
879 \end{tikzpicture}%
880 }}

```

`\ProcessOptionsX*` Again a code from the internet (don't know what `\relax` does):

```

881 \ProcessOptions*\relax

```

`\BasicTree` We define our `\BasicTree`. We check if the last paramter is “leaf”, if not we check if the last paramter is empty, if not we generate an error meassge:

```

882 \newcommand\BasicTree[5][1]{%
883 \def\leaf@or@not@leaf{#5}%
884 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
885 \Basic@Tree[#1]{#2}{#3}{#4}{#5}\tikzsymbolsaftersymbolinput%
886 \else%
887 \ifx\#5\%
888 \Basic@Tree[#1]{#2}{#3}{#4}{#5}\tikzsymbolsaftersymbolinput%
889 \else%
890 \PackageError{tikzsymbols}{The last parameter has either to be \MessageBreak
891 ‘leaf’ or has to be empty}{See the documentation. Section ‘Treesñ.}%
892 \fi\fi%
893 }

```

`\WorstTree` An extremly bad Tree. It's really worst.

```

894 \DeclareRobustCommand{\WorstTree}[1][1]{%
895 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.04ex*#1,scale=#1]
896 \fill[brown] (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
897 controls (0.5,0.7) and (0.5,0.3) .. (1,0);
898 \draw (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
899 controls (0.5,0.7) and (0.5,0.3) .. (1,0) ;
900 \fill[green] (0.2,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
901 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
902 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) ..
903 controls (1.1, 1.6) and (1,1.4) .. (1,1.2) ..
904 controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.5,0.8);
905 \draw (0.214,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
906 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
907 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) .. controls (1.1, 1.6) and (1,1.4) ..
908 (1,1.2) .. controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.486,0.8);
909 \fill[red] (0,1) circle (0.1);
910 \fill[red] (0.4,1.2) circle (0.1);
911 \fill[red] (0.8,1.1) circle (0.1);
912 \end{tikzpicture}%
913 \tikzsymbolsaftersymbolinput%
914 }

```

“Hey! That is the code from the tikzmanuel!!!” – “Yes, I know.”

```

\Springtree Some predefined Trees.
\Summertree “Hey that look like the trees in the ...” – “Yes, Yes, I know!”.
\Autumntree We don’t need \tikzsymbolsaftersymbolinput because it is already defined
\Wintertree in \BasicTree.

915 \DeclareRobustCommand{\Springtree}[1][1=1, usedefault]%
916 {\BasicTree[#1]{brown!70!black}{green!90!black}{green!80!black}{leaf}}
917 \DeclareRobustCommand{\Summertree}[1][1=1, usedefault]%
918 {\BasicTree[#1]{brown!50!black}{green!80!black}{red!80!green}{leaf}}
919 \DeclareRobustCommand{\Autumntree}[1][1=1, usedefault]%
920 {\BasicTree[#1]{red!30!black}{red!75!black}{orange}{leaf}}
921 \DeclareRobustCommand{\Wintertree}[1][1=1, usedefault]%
922 {\BasicTree[#1]{black!80!}{black!50}{black!25}{}}

```

Well that’s it. Happy T<sub>E</sub>Xing!

PS. Something went wrong with the Change History, but I don’t know what.

## Change History

v1.0	General: Initial version . . . . .	1	Now an error message is generated if the last parameter of “BasicTree” is neither “leaf” nor empty. . . . .	1	
v1.05	General: Deleted a “t” in the BasicTree-code, shortened the trunk from the tree a bit, renamed some codes, made an index . . . . .	1	v1.61	General: Made an invisible box in BasicTree. . . . .	1
v1.6	General: on/off. . . . .	1	v1.65	General: Improved BasicTree; New symbols “Schaler/peeler”, Laughey, Walley, Ninja; but didn’t improve the source-description . . . . .	1
	Renamed “tikzsymbolsaftersymbolinput” to “tikzsymbolsaftersymbolinput” . . . . .	1	v1.7	General: New symbols, etc. . . . .	1
	Now “Person” can be used in sections, etc. . . . .	1			

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>Symbols</b>	807, 841, 868, 884	.....	820
\@BackblechlochX 143, 145	\@tikzsymbols@draftfalse	\@tikzsymbols@finaltrue	
\@BackblechlochY 144, 146	.....	822	.....
\@Ofenschalter 155, 156	\@tikzsymbols@drafttrue		
\@Tree@SetUp 784, 795, 856	.....	820	\@y@x@y@x . 839, 842–846
\@leaf@is@leaf . 781,	\@tikzsymbols@finalfalse	\\	.....
			887

<b>A</b>		<b>E</b>	<b>\Moai</b> ..... <u>750</u>
<b>\a</b> ..... 813, 874		<b>\else</b> ..... 620,	
<b>\Annoey</b> ..... 239		754, 757, 809,	
<b>\Annoey_\dAnnoey</b> .. <u>239</u>		847, 870, 886, 889	
<b>\Autumntree</b> ..... <u>915</u>		<b>\endcsname</b> ..... 825	
<b>B</b>		<b>F</b>	
<b>\b</b> ..... 813, 874		<b>\fi</b> ... 607, 620, 636,	
<b>\Backblech</b> .... 136, 150		759, 811, 830,	
<b>\Basic@Tree</b> <u>793</u> , 821,		835, 851, 872, 892	
823, 825, 885, 888		<b>\filldraw</b> ..... 138	
<b>\Basic@Tree@off</b> 821, <u>837</u>		<b>\final</b> ..... <u>822</u>	
<b>\Basic@Tree@on</b> 823, <u>854</u>		<b>\Fire</b> ..... 689	
<b>\BasicTree</b> ..... <u>882</u> ,		<b>\fire</b> ..... 689	
916, 918, 920, 922		<b>\foreach</b> .. 143, 144, 155	
<b>\Bed</b> ..... 735			
<b>\bed</b> ..... <u>735</u>		<b>H</b>	
<b>\Black@is@Black</b> 598,		<b>\Herd</b> ..... 174, 186	
606, 619, 625, 635			
<b>\Black@or@not@Black</b>		<b>I</b>	
..... 599,		<b>\if@draft_\if@final</b> <u>782</u>	
606, 619, 626, 635		<b>\if@tikzsymbols@draft</b>	
<b>\Bratpfanne</b> ..... 30, 51		..... 782, 826	
		<b>\if@tikzsymbols@final</b>	
<b>C</b>		..... 783, 831	
<b>\c</b> ..... 814, 875		<b>\ifnum</b> ..... 752, 755	
<b>\Candle</b> ..... 678		<b>\ifx</b> 606, 619, 635, 807,	
<b>\candle</b> ..... <u>678</u>		841, 868, 884, 887	
<b>\Cat</b> ..... <u>571</u>		<b>\Innocey</b> ..... 357, 370	
<b>\Chair</b> ..... 725		<b>\Innocey_\dInnocey</b> . <u>357</u>	
<b>\chair</b> ..... <u>725</u>			
<b>\cofeecup</b> ..... <u>708</u>		<b>K</b>	
<b>\Coffeecup</b> ..... 708		<b>\Kochtopf</b> ..... 10, 29	
<b>\Cooley</b> ..... 383			
<b>\Cooley_\dCooley</b> .. <u>383</u>		<b>L</b>	
<b>\coordinate</b> ... 812, 873		<b>\Laughey</b> ..... 277	
<b>\csname</b> ..... 825		<b>\Laughey_\dLaughey</b> . <u>277</u>	
		<b>\leaf@or@not@leaf</b> ..	
<b>D</b>		..... 794,	
<b>\d</b> ..... 814, 875		807, 838, 841,	
<b>\DeclareOption</b> ....		855, 868, 883, 884	
..... 820, 822, 824		<b>\leafcolor</b> .... 815, 876	
<b>\def</b> ... 598, 599, 625,		<b>\let</b> ..... 29, 51,	
626, 781, 794,		77, 114, 124, 135,	
821, 823, 825,		150, 162, 173,	
838, 839, 855, 883		186, 199, 208, 219	
<b>\draft</b> ..... <u>820</u>		<b>M</b>	
<b>\Dreizack</b> .... 125, 135		<b>\MessageBreak</b> .....	
<b>\drWalley</b> ..... 558		.. 827, 829, 830,	
		832, 834, 835, 890	
		<b>N</b>	
		<b>\NeedsTeXFormat</b> .... 1	
		<b>\Neutrey</b> ..... 300	
		<b>\Neutrey_\dNeutrey</b> . <u>300</u>	
		<b>\newcommand</b> ..... 9, 882	
		<b>\newif</b> ..... 782, 783	
		<b>\newlength</b> .... 746–749	
		<b>\NiceReapey</b> ..... 649	
		<b>\Ninja</b> ..... 597	
		<b>\Ninja_\dNinja</b> .... <u>597</u>	
		<b>\Nursey</b> ..... 441	
		<b>\Nursey_\dNursey</b> .. <u>441</u>	
		<b>O</b>	
		<b>\Ofen</b> ..... 151, 162	
		<b>P</b>	
		<b>\PackageError</b> ..... 890	
		<b>\PackageWarningNoLine</b>	
		..... 827, 832	
		<b>\Person</b> ..... <u>667</u>	
		<b>\Pfanne</b> ..... 163, 173	
		<b>\pgfarrowsdeclare</b> ..	
		..... 796, 857	
		<b>\pgfarrowslefttextend</b>	
		..... 797, 858	
		<b>\pgfarrowsrighttextend</b>	
		..... 797, 858	
		<b>\pgfpatharc</b> .....	
		.. 800, 801, 861, 862	
		<b>\pgfpathmoveto</b> 799, 860	
		<b>\pgfpoint</b> .... 799, 860	
		<b>\pgfusepathqfill</b> 802, 863	
		<b>\ProcessOptions</b> ... 881	
		<b>\ProcessOptionsX*</b> .. <u>881</u>	
		<b>\ProvidesPackage</b> ... 2	
		<b>\Purierstab</b> ... 115, 124	
		<b>R</b>	
		<b>\relax</b> ..... 881	
		<b>\RequirePackage</b> ... 4–7	
		<b>\rWalley</b> ..... 527	
		<b>S</b>	
		<b>\Sadey</b> ..... 220	
		<b>\Sadey_\dSadey</b> .... <u>220</u>	
		<b>\Saftpresse</b> ... 187, 199	
		<b>\Schaler</b> ..... 209, 219	

\Schneebesen . . . .	52, 77	247, 256, 266,	753, 756, 758, 761
\Schussel . . . .	200, 208	275, 288, 298,	
\setlength .	748, 749,	308, 317, 327,	\Tongey . . . . . 412
	751, 753, 756, 758	336, 346, 355,	\Tongey_\dTongey .. 412
\Sey . . . . .	338	368, 381, 396,	\tree . . . . . 824
\Sey_\dSey . . . . .	338	410, 425, 439,	
\Sieb . . . . .	78, 114	456, 472, 492,	<b>U</b>
\Smiley . . . . .	258	506, 525, 540,	\usetikzlibrary . . . . 8
\Smiley_\dSmiley ..	258	556, 569, 595,	
\Springtree . . . . .	915	622, 647, 665,	<b>V</b>
\Summertree . . . . .	915	676, 687, 706,	\Vomey . . . . . 474
		723, 733, 744,	\Vomey_\dVomey .... 474
		779, 885, 888, 913	
<b>T</b>			<b>W</b>
\tikzset . . . . .	784	\tikzsymbolsMoaiCheckI	\Walley . . . . . 508
\tikzsymbolsaftersymbolinput		746, 752	\Walley_\dWalley .. 508
..... 9, 27, 49,		\tikzsymbolsMoaiCheckII	\Winkey . . . . . 319
75, 112, 122, 133,		746, 755	\Winkey_\dWinkey .. 319
148, 160, 171,		\tikzsymbolsMoaiscale	\wInnocey . . . . . 370
184, 197, 206,		746, 751, 752, 755	\Wintertree . . . . . 915
217, 228, 237,		\tikzsymbolsMoaitickness	\WorstTree . . . . . 894