

tikzsymbols*

Ben Vitecek
b.vitecek@gmx.at

April 18, 2013

Abstract

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

Contents

1	Short Introduction	2
2	Options	2
2.1	tree=on/off, draft, final	2
2.2	marvosym	2
2.3	draftabsolute	3
3	Symbols	3
3.1	cooking-symbols 🍷	3
3.2	Emoticons ☺	4
3.2.1	“normal” Emoticons 🐱	4
3.2.2	“3D” Emoticons 🤪 🤩	5
3.3	other Symbol(s) 🍷	5
3.4	Trees 🌳	6
3.5	Something to redefine	7
4	Warnings and Errors	7
4.1	Warnings	7
4.2	and errors	8
5	Nobody is perfect	8
6	Code	8
6.1	Cookingsymbolcode	12
6.2	Emoticonscode	19
6.3	Other symbols(s)	34
6.4	Trees	38

*This document corresponds to tikzsymbols v2.5, dated 2013/04/18.

1 Short Introduction

There are about two emoticons available in L^AT_EX: 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

2.1 tree=on/off, draft, final

These options are for the commands in the section “Trees” 3.4. The trees look pretty nice, but have one drawback: L^AT_EX 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^AT_EX 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.

2.2 marvosym

Package “marvosym” defines the partly the same commands as “tikzsymbols”. *You should always load “tikzsymbols” after “marvosym”!!* If you do that, “tikzsymbols” redefines for example marvosyms “Smiley” and “Coffeecup”. But if you like marvosyms “Smiley” more than the “Smiley” from “tikzpicture”, you should use the option “marvosym”:






Without option “marvosym” ☹️ ☹️	With option “marvosym” ☹️ ☹️
<code>\usepackage{marvosym}</code>	<code>\usepackage{marvosym}</code>
<code>\usepackage{tikzsymbols}</code>	<code>\usepackage{marvosym}{tikzsymbols}</code>

If you use the option **marvosym** without loading the package, L^AT_EX will produce an error message.

2.3 draftabsolute

Another problem with tikz is: if you use tikz often, L^AT_EX becomes extremely slow, same here. To avoid it, I made the option **draftabsolute**. If you use this option, tikz is not used at all. Instead of the symbols some text appears. So tikz is not used and L^AT_EX becomes faster again. If you have many of those tikzsymbols, use this option (I won't guarantee that the text has the same width and height as the symbols, but I did my best). At the symbol tables, you will see the replacements of the symbols if you use **draftabsolute**. *If you use many symbols of this package, it is recommended to use this option*

3 Symbols

In this section the symbols are introduced. They  all  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 (not exactly) -1400 and (also not exactly) 1400¹, 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	draftabsolute
\Kochtopf	\pot	[<scale>]		kochen
\Bratpfanne	\fryingpan	[<scale>]		braten
\Schneebeesen	\eggbeater	[<scale>]		X
\Sieb	\sieve	[<scale>]		sieve
\Purierstab	\blender ²	[<scale>]		X
\Dreizack	\trident	[<scale>]		I
\Backblech	\bakingplate	[<scale>]		Backblech
\Ofen	\oven	[<scale>]		oven
\Pfanne	\pan	[<scale>]		Pfanne
\Herd	\cooker	[<scale>]		cooker
\Saftpresse	\squeezer	[<scale>]		X
\Schussel	\bowl	[<scale>]		bowl
\Schaler	\peeler	[<scale>]		X

¹Since version 2.2 you can use negative numbers as well (see examples)

²I know that "Purierstab" should be translated as "immersion blender", but I'm just using "blender"

3.2 Emoticons ☺

3.2.1 “normal” Emoticons 🐱

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

<scale> can be a number between (not exactly) -2000 and (not exactly) 2000³, default is 1.

<color> can be every defined color.

Commands	Optional parameter(s)	Output	draftabsolute
\Sadey	[\scale] [\color]	☹	Sad
\Smiley	[\scale] [\color]	☺	Smile
\Laughey	[\scale] [\color] [\mouth color]	😄	laugh
\Annoey	[\scale] [\color]	😞	Ann
\Neutrey	[\scale] [\color]	😐	Neut
\Winkey	[\scale] [\color]	😉	Waik
\Sey	[\scale] [\color]	😏	S
\Xey	[\scale] [\color]	😬	Xe
\Innocey	[\scale] [\color] [\halo color]	😇	Immo
\wInnocey	[\scale]	😇	Immo
\Cooley	[\scale] [\color]	😎	Cool
\Tongey	[\scale] [\color] [\tongue color]	😜	ton
\Nursey ⁴	[\scale] [\color] [\cap color] [\cross color]	👩	Nurse
\Vomey	[\scale] [\color] [\vomit color]	🤮	vomit
\Walley	[\scale] [\color] [\wall color]	🏠	Wall
\rWalley ⁵	[\scale] [\color] [\wall color]	🏠	rWall
\Cat	[\scale]	🐱	Cat
\Ninja	[\scale] [\color] [\headband color] [\eye color]	🥷	ssh
\NiceReapey	[\scale]	👻	Ann

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[-1] \Neutrey} 😄 😞 😐

{\color{blue}\Sey} 😏

³Do you even need so large symbols?

















⁴The cross has nothing to do with religion meanings.

⁵“r” stands “random” and means that the cracks in the wall are generated randomly, but it takes some time to generate it.

3.2.2 “3D” Emoticons 🤪🤨

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

`<scale>` can be a number between a small number⁶ and a large number⁷, default is 1. `<color>` can be every defined color (see examples below).

Commands	Optional parameter(s)	Output	<code>draft</code> <code>absolute</code>
<code>\dSadey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dSad</code>
<code>\dSmiley</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dSmile</code>
<code>\dLaughy</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle mouth color \rangle]</code>		<code>dLaug</code>
<code>\dAnnoey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dAnn</code>
<code>\dNeutrey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dNeut</code>
<code>\dWinkey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dWin</code>
<code>\dSey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dS</code>
<code>\dXey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dXe</code>
<code>\dInnocey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle halo color \rangle]</code>		<code>dInno</code>
<code>\dCooley</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>		<code>dCool</code>
<code>\dTongey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle tongue color \rangle]</code>		<code>dTun</code>
<code>\dNursey</code> ⁴	<code>[\langle scale \rangle] [\langle color \rangle] [\langle cap color \rangle] [\langle cross color \rangle]</code>		<code>dNur</code>
<code>\dVomey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle vomit color \rangle]</code>		<code>dvomit</code>
<code>\dWalley</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle wall color \rangle]</code>		<code>dWall</code>
<code>\drWalley</code> ⁸	<code>[\langle scale \rangle] [\langle color \rangle] [\langle wall color \rangle]</code>		<code>drWall</code>
<code>\dNinja</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle headband color \rangle] [\langle eye color \rangle]</code>		<code>dsnh</code>







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) 📖

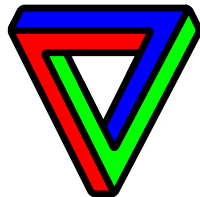
Commands	Optional parameter(s)	Output	<code>draft</code> <code>absolute</code>
<code>\Person</code>	<code>[\langle scale \rangle] [\langle left arm \rangle] [\langle right arm \rangle] [\langle left leg \rangle] [\langle right arm \rangle]</code>		P
<code>\Candle</code>	<code>[\langle scale \rangle]</code>		C
<code>\Fire</code>	<code>[\langle scale \rangle]</code>		fire
<code>\Coffeecup</code>	<code>[\langle scale \rangle]</code>		coffe
<code>\Chair</code>	<code>[\langle scale \rangle]</code>		chair
<code>\Bed</code>	<code>[\langle scale \rangle]</code>		Bed


⁶under 500 for sure

⁷over 500 for sure

⁸“r” stands for “random” and means that the cracks in the wall are generated randomly, but it takes some time.

Commands	Optional parameter(s)	Output	draftabsolute
<code>\Moai</code>	<code>[\scale]</code>		M
<code>\Tribar</code>	<code>[\scale] [\color 1] [\color 2] [\color 3]</code>		T
<code>\Snowman</code>	<code>[\scale]</code>		Snow



`\Tribar[-10] [blue] [red] [green]`  `\Tribar[2.1] [blue] [blue!50] [blue!20]`

`\Person[1] [10] [30] [40] [4]%, \Person[1.4] [210] [310] [10] [90]%,`
`\Person[2] [510] [110] [190] [990]%, \Person[0.9] [54] [28] [95] [16]%`

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 (not exactly) -900 and (again not exactly) 900 ⁹, 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^AT_EX needs longer to produce the pdf. So you may use the package option `tree=off` or `draft` (see section 2) to make L^AT_EX faster.


Commands	Optional/Needed parameter(s)	Output	draftabsolute
<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>		tree

`\BasicTree` examples:

`\colorbox{green}{\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} {}` 

`\BasicTree[-1.54] {green!90!black} {green!95!black} {green!99!black} {leaf}` 

⁹if it is larger (or less) it uses too much of L^AT_EX memory and an error message appears.

`\colorbox{black}{\BasicTree[3.75]{gray!80}{gray!50}{gray!40}{leaf}}`



`\colorbox{green}{\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}{}`



`\BasicTree[-1.54]{green!90!black}{green!95!black}{green!99!black}{leaf}`



`\colorbox{black}{\BasicTree[3.75]{gray!80}{gray!50}{gray!40}{leaf}}`



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^AT_EX 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 (if you like the “Smiley” from marvosym more, use the tikzsymbols option `marvosym`).

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}[2011/06/31]
2 \ProvidesPackage{tikzsymbols}
3 [2013/04/18 v2.5 Some symbols created using tikz.]
4 \@ifpackageloaded{tikz}{}{\RequirePackage{tikz}}
5 \@ifpackageloaded{xargs}{}{\RequirePackage{xargs}}
6 \@ifpackageloaded{xcolor}{}{\RequirePackage{xcolor}}
7 \@ifpackageloaded{xspace}{}{\RequirePackage{xspace}}
8 \@ifpackageloaded{xkeyval}{}{\RequirePackage{xkeyval}}
9 \@ifpackageloaded{calc}{}{\RequirePackage{calc}}
```

Furthermore we need to load some libraries from tikz:

```
10 \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)

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

`\@leaf@is@leaf`

We need this command for creating an error message if the last paramter of BasicTree is neither “leaf” nor empty.

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



```

\tkzsymlbsscl .
13 \newlength{\tkzsymlbsscl}
14 \def\set@tkzsymlbsscl#1{\setlength{\tkzsymlbsscl}{#1pt}}

\tikzsymbols@resizebox We define a box.
15 \def\tikzsymbols@resizebox#1#2#3{%
16 \resizebox{#1}{\ifdim\tkzsymlbsscl<0pt-\fi #2}{#3}%
17 }

\if@draft \if@final We need draft and final for some package warnings. “marvosym” for the option
\if@tkzssmbles@neg “marvosym” and “neg” (negative) if something is negativ.
\if@tikzsymbols@marvosym
18 \newif\if@tikzsymbols@draft
19 \newif\if@tikzsymbols@final
20 \newif\if@tkzssmbles@neg
21 \newif\if@tikzsymbols@marvosym
22 \newif\if@tikzsymbols@absolute@draft

marvosym I define the option “marvosym”: you should use it, if you load the package
“marvosym”
23 \DeclareOptionX{marvosym}{\@tikzsymbols@marvosymtrue}

\@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):
24 \def\@Tree@SetUp{\tikzset{%
25 ld/.style={level distance=##1ex},lw/.style={line width=##1ex},%
26 level 1/.style={ld=0.60, trunk,lw=0.1 ,sibling angle=60},%
27 level 2/.style={ld=0.20,trunk!80!leaf a,lw=.073,sibling angle=70},%
28 level 3/.style={ld=0.25,trunk!60!leaf a,lw=.05,sibling angle=70}, %
29 level 4/.style={ld=0.10,trunk!40!leaf a,lw=.025,sibling angle=60},%
30 level 5/.style={ld=0.15,trunk!20!leaf a,lw=.02,sibling angle=60},%
31 level 6/.style={ld=0.08,leaf a,lw=.021,sibling angle=60},%
32 }}%

\Basic@Tree Now we define our \Basic@Tree. We will need it later for our package option
(basic code is also from “tikz” manual).
33 \DeclareRobustCommand{\Basic@Tree}[5][1=1, usedefault]{%
34 \set@tkzsymlbsscl{#1}\ifdim\tkzsymlbsscl<0pt \@tkzssmbles@negtrue\fi%
35 \def\leaf@or@not@leaf{#5}%
36 \@Tree@SetUp%
37 \pgfarrowsdeclare{leaf}{leaf}%
38 {\pgfarrowsleftextend{-.1ex} \pgfarrowsrightextend{-0.05ex}}%
39 {%
40 \pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
41 \pgfpatharc{150}{30}{0.08ex}% dicke
42 \pgfpatharc{-30}{-150}{0.08ex}%
43 \pgfusepathqfill%
44 }%
45 \colorlet{trunk}{#2}%
46 \colorlet{leaf a}{#3}%

```

```

47 \colorlet{leaf b}{#4}%
48 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
49 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
50   \if@tkzssmbles@neg%
51     \draw[opacity=0,scale=#1+0.1*#1] (-0.82+0.1*#1/100,0) rectangle
52     (0.82-0.1*#1/100, 1.4-0.01*#1/100);
53   \else
54     \draw[opacity=0,scale=#1+0.1*#1] (-0.82-0.1*#1/100,0) rectangle
55     (0.82+0.1*#1/100, 1.4+0.01*#1/100);
56   \fi%
57 \else
58   \draw[opacity=0,scale=#1+0.1*#1] (-0.75,-0.01) rectangle (0.75,1.3);
59 \fi
60 \pgflowlevel{\pgftransformscale{#1+0.02ex}}{%
61 \coordinate (root) [grow cyclic,rotate=90] child {
62 child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
63 child foreach \c in {0,1,2} { child foreach \d in {0,1} {
64 child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5] }
65 }}} } edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
66 };}%
67 \end{tikzpicture}%
68 \@tkzssmbles@negfalse%
69 }}

```

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

```

70 \DeclareOptionX{draft}{\@tikzsymbols@drafttrue\@tikzsymbols@finalfalse
71 \def\Basic@Tree{\Basic@Tree@off}}

```

draft=absolute I define the option “`draft=absolute`”.

```

72 \DeclareOptionX{draftabsolute}{\def\Basic@Tree{\Basic@Tree@off}%
73 \@tikzsymbols@absolute@drafttrue}

```

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

```

74 \DeclareOptionX{final}{\@tikzsymbols@draftfalse\@tikzsymbols@finaltrue
75 \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`:

```

76 \DeclareOptionX{tree}{%
77   \def\Basic@Tree{\csname Basic@Tree@#1\endcsname}%
78   \if@tikzsymbols@draft%
79 \PackageWarningNoLine{tikzsymbols}{You can use class option \MessageBreak
80 draft with package option tree=on/off;}

```

```

81 \MessageBreak but I think it would be better if you
82 \MessageBreak delete tree=on/off}\fi%
83 \if@tikzsymbols@final%
84 \PackageWarningNoLine{tikzsymbols}{You can use class option \MessageBreak
85 final with package option tree=on/off;
86 \MessageBreak but I think it would be better if you
87 \MessageBreak delete tree=on/off}\fi%
88 }

```

`\Basic@Tree@off` We define `\Basic@Tree@off`; it will be shown if `tree=off`. It 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”:

```

89 \DeclareRobustCommand{\Basic@Tree@off}[5][1=1, usedefault]{%
90 \set@tkzsymlsscl{#1}%
91 \if@tikzsymbols@absolute@draft%
92 \tikzsymbols@resizebox{#1ex*174/100}{#1ex*15/10}{BT}%
93 \else%
94 \ifdim\tkzsymlsscl<0pt \set@tkzsymlsscl{-#1}\fi%
95 \def\leaf@or@not@leaf{#5}%
96 \def\Tree@Off@line{0.095*#1/100}%
97 \begin{tikzpicture}[scale=#1+0.01*#1,x=1.35ex,y=1.3ex, line width=0.07ex*\tkzsymlsscl]
98 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
99 \draw[#2] (0-\Tree@Off@line,0) -- (0-\Tree@Off@line,1.08+0.05*#1/100);
100 \draw[#3] (0-\Tree@Off@line,1.08+0.05*#1/100) -- (1.2+\Tree@Off@line,1.08+0.05*#1/100);
101 \draw[#4] (1.2+\Tree@Off@line,1.08+0.05*#1/100) -- (1.2+\Tree@Off@line,0);
102 \draw[#3] (1.2+\Tree@Off@line,0) -- (0.5,0);
103 \draw[#4] (0.5+0.4*#1/100,0) -- (0-\Tree@Off@line,0);
104 \else
105 \draw[#2] (0,0) -- (0,1);
106 \draw[#3] (0,1) -- (1.15,1);
107 \draw[#4] (1.15,1) -- +(0,-1);
108 \fi%
109 \end{tikzpicture}%
110 \fi%
111 }}

```

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

```

112 \DeclareRobustCommand{\Basic@Tree@on}[5][1=1, usedefault]{%
113 \set@tkzsymlsscl{#1}\ifdim\tkzsymlsscl<0pt \tkzssmbles@negtrue\fi%
114 \def\leaf@or@not@leaf{#5}%
115 \@Tree@SetUp%
116 \pgfarrowsdeclare{leaf}{leaf}%
117 {\pgfarrowsleftextend{-.1ex} \pgfarrowsrightextend{-0.05ex}}%
118 {%
119 \pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
120 \pgfpatharc{150}{30}{0.08ex}% dicke
121 \pgfpatharc{-30}{-150}{0.08ex}%
122 \pgfusepathqfill%
123 }%

```

```

124 \colorlet{trunk}{#2}%
125 \colorlet{leaf a}{#3}%
126 \colorlet{leaf b}{#4}%
127 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
128 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
129   \if@tkzssmbles@neg%
130     \draw[opacity=0,scale=#1+0.1*#1] (-0.82+0.1*#1/100,0) rectangle
131     (0.82-0.1*#1/100, 1.4-0.01*#1/100);
132   \else
133     \draw[opacity=0,scale=#1+0.1*#1] (-0.82-0.1*#1/100,0) rectangle
134     (0.82+0.1*#1/100, 1.4+0.01*#1/100);
135   \fi%
136 \else
137   \draw[opacity=0,scale=#1+0.1*#1] (-0.75,-0.01) rectangle (0.75,1.3);
138 \fi
139 \pgflowlevel{\pgftransformscale{#1+0.02ex}}{%
140 \coordinate (root) [grow cyclic,rotate=90] child {
141 child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
142 child foreach \c in {0,1,2} { child foreach \d in {0,1} {
143 child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5] }
144 }}} } edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
145 };}%
146 \end{tikzpicture}%
147 \@tkzssmbles@negfalse%
148 }}%

```

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

```

149 \ProcessOptionsX*\relax

```

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:

```

150 \DeclareRobustCommand{\Kochtopf}[1][1]{%
151 \set@tkzsymlsscl{#1}%
152 \if@tkzsymbols@absolute@draft%
153 \tikzsymbols@resizebox{#1ex*247/100}{#1ex*151/100}{kochen}%
154 \else%
155 \ifdim\tikzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
156 \begin{tikzpicture}[x=2ex,y=2.2ex, line width=0.07ex*\tikzsymlsscl,scale=#1]
157 \draw[rounded corners=0.2ex*\tikzsymlsscl] (0,0.5) -- (0,0) -- (1,0) -- (1,0.5);
158 \draw(0,0.4) arc (90:270:0.1);
159 \draw(1,0.4) arc (90:-90:0.1);
160 \draw (0,0.5) -- (1,0.5) .. controls (1,0.6) and (0,0.6) .. (0,0.5);
161 \draw (0.6,0.585) arc (0:180:0.1);
162 \draw[decorate,
163 decoration={snake,amplitude=.12ex*\tikzsymlsscl,segment length=0.93ex*\tikzsymlsscl}]
164 (0,0.35) -- (1,0.35);

```

```

165 \draw (0.1,0.25) circle (0.04);
166 \draw (0.3,0.2) circle (0.04);
167 \draw (0.13, 0.125) circle (0.04);
168 \draw (0.6,0.25) circle (0.04);
169 \draw (0.45,0.1) circle (0.04);
170 \draw (0.88,0.2) circle (0.04);
171 \draw (0.7,0.11) circle (0.04);
172 \end{tikzpicture}%
173 \fi%
174 \tikzsymbolsaftersymbolinput%
175 }
176 \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 the symbols would not look 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 the same, no matter how you scale it.

```

177 \DeclareRobustCommand{\Bratpfanne}[1][1]{%
178 \set@tkzsymlsscl{#1}%
179 \if@tikzsymbols@absolute@draft%
180 \tikzsymbols@resizebox{#1ex*355/100}{#1ex*14/10}{braten}%
181 \else%
182 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
183 \begin{tikzpicture}[x=0.7ex,y=1.4ex, line width=0.07ex*\tkzsymlsscl, scale=#1,
184 decoration={snake,amplitude=.05ex*\tkzsymlsscl,segment length=0.408ex*\tkzsymlsscl}]
185 \draw[rounded corners=0.07ex*\tkzsymlsscl]
186 (-1,0) -- (1,0) -- (1.5,0.4) -- (-1.5,0.4) -- cycle;
187 \draw[line width=0.037ex*\tkzsymlsscl, rounded corners=0.023ex*\tkzsymlsscl]
188 (-1.4,0.3) -- (-3.5,0.3) -- (-3.5,0.25) -- (-1.3,0.25);
189 \draw[line width=0.023ex*\tkzsymlsscl] (-1.1,0.1) -- (1.1,0.1);
190 \draw[line width=0.035ex*\tkzsymlsscl, decorate]
191 (-0.3,0.5) -- (-0.3,1);
192 \draw[line width=0.035ex*\tkzsymlsscl, decorate]
193 (0.3,0.5) -- (0.3,1);
194 \draw[line width=0.035ex*\tkzsymlsscl, decorate]
195 (-1,0.5) -- (-1,1);
196 \draw[line width=0.035ex*\tkzsymlsscl, decorate]
197 (1,0.5) -- (1,1);
198 \end{tikzpicture}%
199 \fi%
200 \tikzsymbolsaftersymbolinput%
201 }
202 \let\fryingpan\Bratpfanne

```

\Schneebeesen = \eggbeater The next one:

```

203 \DeclareRobustCommand{\Schneebeesen}[1][1]{%
204 \set@tkzsymlsscl{#1}%

```

```

205 \if@tikzsymbols@absolute@draft%
206 \tikzsymbols@resizebox{#1ex*57/100}{#1ex*155/100}{x}%
207 \else%
208 \ifdim\tikzsymbolsscl<0pt\set\tikzsymbolsscl{-#1}\fi%
209 \begin{tikzpicture}[y=2.1ex,x=1.4ex, scale=#1]
210 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
211   (0,0) .. controls (0.2,0.0) and (0.2,0.2) .. (0,0.4);
212 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
213   (0,0) .. controls (-0.2,0.0) and (-0.2,0.2) .. (0,0.4);
214 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
215   (0,0) .. controls (-0.1,0.0) and (-0.1,0.2) .. (0,0.4);
216 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
217   (0,0) .. controls (0.1,0.0) and (0.1,0.2) .. (0,0.4);
218 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
219   (0,0) .. controls (-0.15,0.0) and (-0.15,0.2) .. (0,0.4);
220 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
221   (0,0) .. controls (0.15,0.0) and (0.15,0.2) .. (0,0.4);
222 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
223   (0,0) .. controls (-0.05,0.0) and (-0.05,0.2) .. (0,0.4);
224 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
225   (0,0) .. controls (0.05,0.0) and (0.05,0.2) .. (0,0.4);
226 \draw[line width=0.01ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*3)]
227   (0,0) --(0,0.4);
228 \fill[line width=0.05ex*\tikzsymbolsscl, rounded corners=0.07ex*\tikzsymbolsscl]
229   (-0.05,0.37) -- (0.05,0.37) -- (0.05,0.75) -- (-0.05,0.75) -- cycle;
230 \end{tikzpicture}%
231 \fi%
232 \tikzsymbolsaftersymbolinput%
233 }
234 \let\eggbeater\Schneebesen

\Sieb = \sieve Now a long one;
235 \DeclareRobustCommand{\Sieb}[1][1]{%
236 \set\tikzsymbolsscl{#1}%
237 \if@tikzsymbols@absolute@draft%
238 \tikzsymbols@resizebox{#1ex*3475/1000}{#1ex*112/100}{sieve}%
239 \else%
240 \ifdim\tikzsymbolsscl<0pt\set\tikzsymbolsscl{-#1}\fi%
241 \begin{tikzpicture}[x=2.8ex, y=2.8ex,line width=0.02ex*\tikzsymbolsscl ,scale=#1]
242 \draw[line width=0.09ex*\tikzsymbolsscl] (-0.2,0) -- (1.01,0);
243 \draw (0.2,0) arc (180:360:0.4);
244 \draw(0.25,0) arc (180:360:0.35);
245 \draw (0.3,0) arc (180:360:0.3);
246 \draw (0.35,0) arc (180:360:0.25);
247 \draw (0.4,0) arc (180:360:0.2);
248 \draw (0.45,0) arc (180:360:0.15);
249 \draw (0.5,0) arc (180:360:0.1);
250 \draw (0.55,0) arc (180:360:0.05);
251 \draw (.95,0) -- (0.95,-0.194);
252 \draw (.9,0) -- (0.9,-0.265);

```

```

253 \draw (.85,0) -- (0.85,-0.313);
254 \draw (.8,0) -- (0.8,-0.345);
255 \draw (.75,0) -- (0.75,-0.37);
256 \draw (.7,0) -- (0.7,-0.39);
257 \draw (.65,0) -- (0.65,-0.4);
258 \draw (.6,0) -- (0.6,-0.4);
259 \draw (.55,0) -- (0.55,-0.4);
260 \draw (.5,0) -- (0.5,-0.39);
261 \draw (.45,0) -- (0.45,-0.37);
262 \draw (.4,0) -- (0.4,-0.348);
263 \draw (.35,0) -- (0.35,-0.314);
264 \draw (.3,0) -- (0.3,-0.265);
265 \draw (.25,0) -- (0.25,-0.194);
266 \draw (0.2,-0.05) -- (1,-0.05);
267 \draw (0.21,-0.1) -- (0.99,-0.1);
268 \draw (0.23,-0.15) -- (0.97,-0.15);
269 \draw (0.255,-0.2) -- (0.945,-0.2);
270 \draw (0.289,-0.25) -- (0.911,-0.25);
271 \draw (0.335,-0.3) -- (0.865,-0.3);
272 \draw (0.406,-0.35) -- (0.794,-0.35);
273 \end{tikzpicture}%
274 \fi%
275 \tikzsymbolsaftersymbolinput%
276 }
277 \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 “Pürierstab” would be “immersion blender”, but I am just using “blender”:

```

278 \DeclareRobustCommand{\Purierstab}[1][1]{%
279 \set@tkzsymbolsscl{#1}%
280 \if@tkzsymbols@absolute@draft%
281 \tikzsymbols@resizebox{#1ex*076/100}{#1ex*155/100}{x}%
282 \else%
283 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
284 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.07ex*\tkzsymbolsscl,scale=#1]
285 \draw[rounded corners=0.07ex*\tkzsymbolsscl] (0,0) -- (0.3,0) -- (0.15,0.1) --cycle;
286 \fill[rounded corners=0.07ex*\tkzsymbolsscl] (0.15,0.3) -- (0.24,0.4) -- (0.24,0.7) --
287 (0.06,0.7) -- (0.06,0.4) -- cycle;
288 \draw (0.15,0.4) -- (0.15,0.1);
289 \end{tikzpicture}%
290 \fi%
291 \tikzsymbolsaftersymbolinput%
292 }
293 \let\blender\Purierstab

```

`\Dreizack = \trident` Important for cooking:

```

294 \DeclareRobustCommand{\Dreizack}[1][1]{%
295 \set@tkzsymbolsscl{#1}%

```

```

296 \if@tikzsymbols@absolute@draft%
297 \tikzsymbols@resizebox{#1ex*27/100}{#1ex*155/100}{x}%
298 \else%
299 \ifdim\tikzsymbolsscl<0pt\set\tikzsymbolsscl{-#1}\fi%
300 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.035ex*\tikzsymbolsscl,scale=#1]
301 \fill[rounded corners=0.07ex*(\tikzsymbolsscl-\tikzsymbolsscl/100)]
302 (0,0) -- (0,0.4) -- (0.1,0.4) -- (0.1,0.0) -- cycle;
303 \draw (0.05,0) -- (0.05,0.7);
304 \draw[rounded corners=0.07ex*(\tikzsymbolsscl-\tikzsymbolsscl/100*\tikzsymbolsscl*2]
305 (0,0.7) -- (0,0.55) -- (0.05,0.55) -- (0.1,0.55) -- (0.1,0.7);
306 \end{tikzpicture}%
307 \fi%
308 \tikzsymbolsaftersymbolinput%
309 }
310 \let\trident\Dreizack

```

\Backblech = \bakingplate I may have too many strange words:

```

311 \DeclareRobustCommand{\Backblech}[1][1]{%
312 \set\tikzsymbolsscl{#1}%
313 \if@tikzsymbols@absolute@draft%
314 \tikzsymbols@resizebox{#1ex*232/100}{#1ex*155/100}{\Backblech}%
315 \else%
316 \ifdim\tikzsymbolsscl<0pt\set\tikzsymbolsscl{-#1}\fi%
317 \begin{tikzpicture}[x=6.53ex,y=5.ex, line width=0.07ex*\tikzsymbolsscl,scale=#1]
318 \filldraw[rounded corners=0.09ex*\tikzsymbolsscl] (0,0) rectangle (0.3,0.3);
319 \draw[rounded corners=0.07ex*\tikzsymbolsscl, line width=0.03ex*\tikzsymbolsscl]
320 (0.1,0) -- (-0.025,0) -- (-0.025,0.3) -- (0.1,0.3);
321 \draw[rounded corners=0.07ex*\tikzsymbolsscl, line width=0.03ex*\tikzsymbolsscl]
322 (0.2,0) -- (.325,0) -- (.325,0.3) -- (0.2,0.3);
323 \foreach \@BackblechlochX in {0.007,0.293}
324 \foreach \@BackblechlochY in {0.007,0.293}
325 \fill[white] (\@BackblechlochX,
326 \@BackblechlochY) circle (0.02ex);
327 \end{tikzpicture}%
328 \fi%
329 \tikzsymbolsaftersymbolinput%
330 }
331 \let\bakingplate\Backblech

```

\Ofen = \oven I may have again too many strange words:

```

332 \DeclareRobustCommand{\Ofen}[1][1]{%
333 \set\tikzsymbolsscl{#1}%
334 \if@tikzsymbols@absolute@draft%
335 \tikzsymbols@resizebox{#1ex*208/100}{#1ex*155/100}{\oven}%
336 \else%
337 \ifdim\tikzsymbolsscl<0pt\set\tikzsymbolsscl{-#1}\fi%
338 \begin{tikzpicture}[x=0.50ex,y=.5ex, line width=0.07ex*\tikzsymbolsscl,scale=#1]
339 \draw (0,0) rectangle (4,3);
340 \draw (0.25,0.25) rectangle (3.75,2);
341 \foreach \@Ofenschalter in {0.5,1.1,2.9,3.5}

```



```

342 \fill (\Ofenschalter,2.5) circle (0.22);
343 \draw (1.5,2.28) rectangle (2.5,2.72);
344 \draw[line width=0.05ex*\tkzsymlbsscl] (1,1.75) -- (3,1.75);
345 \end{tikzpicture}%
346 \fi%
347 \tikzsymbolsaftersymbolinput%
348 }
349 \let\oven\Ofen

\Pfanne = \pan I can't think of a better word ...:
350 \DeclareRobustCommand{\Pfanne}[1][1]{%
351 \set@tkzsymlbsscl{#1}%
352 \if@tikzsymbols@absolute@draft%
353 \tikzsymbols@resizebox{#1ex*303/100}{#1ex*74/100}{\Pfanne}%
354 \else%
355 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
356 \begin{tikzpicture}[x=2.3ex,y=2.3ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
357 \draw [rounded corners=0.023ex*\tkzsymlbsscl]
358 (0,0) -- (0.9,0) -- (1,0.3) -- (-0.1,0.3) -- cycle;
359 \draw (-0.2,0.22) -- (-0.08,0.22);
360 \draw (0.97,0.22) -- (1.08,0.22);
361 \draw[decorate,decoration={snake,amplitude=.046ex*\tkzsymlbsscl,
362 segment length=0.82ex*\tkzsymlbsscl},line width=0.05ex*\tkzsymlbsscl]
363 (-0.05,0.1) -- (0.95,0.1);
364 \end{tikzpicture}%
365 \fi%
366 \tikzsymbolsaftersymbolinput%
367 }
368 \let\pan\Pfanne

\Herd = \cooker I hope it' the right translation:
369 \DeclareRobustCommand{\Herd}[1][1]{%
370 \set@tkzsymlbsscl{#1}%
371 \if@tikzsymbols@absolute@draft%
372 \tikzsymbols@resizebox{#1ex*208/100}{#1ex*157/100}{\cooker}%
373 \else%
374 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
375 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.04ex*\tkzsymlbsscl,scale=#1]
376 \draw[line width=0.08ex*\tkzsymlbsscl] (0,0) rectangle (2,1.5);
377 \draw (0.5,0.45) circle (0.35);
378 \draw (0.5,0.45) circle (0.2);
379 \draw (1.45,0.45) circle (0.3);
380 \draw (0.5,1.15) circle (0.21);
381 \draw (1.05,0.95) rectangle (1.85,1.35);
382 \draw (1.45,1.15) circle (0.15);
383 \end{tikzpicture}%
384 \fi%
385 \tikzsymbolsaftersymbolinput%
386 }
387 \let\cooker\Herd

```

\Saftpresse = \squeezer It's an old squeezer:

```

388 \DeclareRobustCommand{\Saftpresse}[1][1]{%
389 \set@tkzsymbolsscl{#1}%
390 \if@tikzsymbols@absolute@draft%
391 \tikzsymbols@resizebox{#1ex*187/100}{#1ex*155/100}{x}%
392 \else%
393 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
394 \begin{tikzpicture}[x=1.2ex,y=1ex,line width=0.07ex*\tkzsymbolsscl,scale=#1]
395 \draw[rounded corners=0.1ex*\tkzsymbolsscl]
396   (0,0.85) -- (0,0) -- (1.5,0) -- (1.5,0.85) -- cycle;
397 \draw (0,0.7) -- (1.5,0.7);
398 \draw[rounded corners=0.1ex*\tkzsymbolsscl] (0.3,0.7) -- (0.75,1.55) -- (1.2,0.7);
399 \draw[rounded corners=0.1ex*\tkzsymbolsscl] (0.45,0.7) -- (0.75,1.55) -- (1.05,0.7);
400 \draw[rounded corners=0.1ex*\tkzsymbolsscl]
401   (0.65,0.7) -- (0.75,1.55) -- (0.85,0.7);
402 \draw[line width=0.05ex*\tkzsymbolsscl, decorate,
403   decoration={snake,amplitude=.05ex*\tkzsymbolsscl,
404     segment length=0.48ex*\tkzsymbolsscl}] (0,0.3) -- (1.5,0.3);
405 \end{tikzpicture}%
406 \fi%
407 \tikzsymbolsaftersymbolinput%
408 }
409 \let\squeezer\Saftpresse

```

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

```

410 \DeclareRobustCommand{\Schussel}[1][1]{%
411 \set@tkzsymbolsscl{#1}%
412 \if@tikzsymbols@absolute@draft%
413 \tikzsymbols@resizebox{#1ex*232/100}{#1ex*148/100}{bowl}%
414 \else%
415 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
416 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex*\tkzsymbolsscl, scale=#1]
417 \draw[rounded corners=0.5ex*\tkzsymbolsscl]
418   (-0.02,1.4) -- (0,1.4) -- (0,0.05) -- (1.5,0.05) -- (1.5,1.4) -- (1.52,1.4);
419 \draw (0.35,0) -- (1.15,0);
420 \draw[opacity=0] (-0.4,0) -- (1.85,0);
421 \end{tikzpicture}%
422 \fi%
423 \tikzsymbolsaftersymbolinput%
424 }
425 \let\bowl\Schussel

```

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

```

426 \DeclareRobustCommand{\Schaler}[1][1]{%
427 \set@tkzsymbolsscl{#1}%
428 \if@tikzsymbols@absolute@draft%
429 \tikzsymbols@resizebox{#1ex*116/100}{#1ex*155/100}{X}%
430 \else%



```

```

431 \ifdim\tkzsymlbsscl<0pt\set\tkzsymlbsscl{-#1}\fi%
432 \begin{tikzpicture}[x=2.7ex,y=2.3ex, line width=0.07ex*\tkzsymlbsscl,scale=#1]
433 \draw[rounded corners=0.07ex*\tkzsymlbsscl]
434   (0,0.4) -- (0,0.1) arc (0:180:-0.1) -- (0.2,0.4)
435   -- (0.3,0.5) -- (0.3,0.65) -- (0.2,0.65) -- (0.2,0.5) -- (0,0.5) -- (0,0.65) --
436   (-0.1,0.65) -- (-0.1,0.5) -- cycle;
437 \draw[line width=0.03ex*\tkzsymlbsscl] (0,0.6) -- (0.2,0.6);
438 \draw[line width=0.03ex*\tkzsymlbsscl] (0,0.58) -- (0.2,0.58);
439 \end{tikzpicture}%
440 \fi%
441 \tikzsymbolsaftersymbolinput%
442 }
443 \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”, “Ninja”, and else). The “default color” of the 2D Emoticons is `opacity=0`, it’s useful for `\colorbox{yellow}{\Sadey}` which leads to  instead of  (with default=white).

```

444 \DeclareRobustCommand{\Sadey}[2][1=1, 2={opacity=0}, usedefault]{%
445 \set\tkzsymlbsscl{#1}%
446 \if\tikzsymbols@absolute@draft%
447 \tikzsymbols@resizebox{#1ex*172/100}{#1ex*165/100}{Sad}%
448 \else%
449 \ifdim\tkzsymlbsscl<0pt\set\tkzsymlbsscl{-#1}\fi%
450 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
451 \fill[#2, line width=0.1ex*\tkzsymlbsscl] (0,0) circle (0.33);
452 \draw[line width=0.12ex*\tkzsymlbsscl] (0,0) circle (0.33);
453 \fill (0.1,0.1) circle (0.05);
454 \fill (-0.1,0.1) circle (0.05);
455 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
456 \end{tikzpicture}%
457 \fi%
458 \tikzsymbolsaftersymbolinput%
459 }
460 \DeclareRobustCommand{\dSadey}[2][1=1,2=yellow,usedefault]{%
461 \set\tkzsymlbsscl{#1}%
462 \if\tikzsymbols@absolute@draft%
463 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dSad}%
464 \else%
465 \ifdim\tkzsymlbsscl<0pt\set\tkzsymlbsscl{-#1}\fi%
466 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
467 \shade[ball color=#2] (0,0) circle (0.33);
468 \shade[ball color=black] (0.1,0.1) circle (0.05);
469 \shade[ball color=black] (-0.1,0.1) circle (0.05);
470 \draw[black] (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
471 \end{tikzpicture}%

```

```

472 \fi%
473 \tikzsymbolsaftersymbolinput%
474 }

\Annoey \dAnnoey  An annoyed Smiley --
475 \DeclareRobustCommand{\Annoey}[2][1,2={opacity=0},usedefault]{%
476 \set@tkzsymbolsscl{#1}%
477 \if@tikzsymbols@absolute@draft%
478 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{Ann}%
479 \else%
480 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
481 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
482 \fill[#2, line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
483 \draw[line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
484 \draw (0.08,0.1) -- (0.22,0.1);
485 \draw (-0.08,0.1) -- (-0.22,0.1);
486 \draw (-0.2,-0.1) -- (0.2,-0.1);
487 \end{tikzpicture}%
488 \fi%
489 \tikzsymbolsaftersymbolinput%
490 }

491 \DeclareRobustCommand{\dAnnoey}[2][1,2=yellow,usedefault]{%
492 \set@tkzsymbolsscl{#1}%
493 \if@tikzsymbols@absolute@draft%
494 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dAnn}%
495 \else%
496 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
497 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
498 \shade[ball color=#2] (0,0) circle (0.33);
499 \draw[black] (0.08,0.1) -- (0.22,0.1);
500 \draw[black] (-0.08,0.1) -- (-0.22,0.1);
501 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
502 \end{tikzpicture}%
503 \fi
504 \tikzsymbolsaftersymbolinput%
505 }

\Smiley \dSmiley  A normal Smiley
506 \if@tikzsymbols@marvosym\relax\else%
507 \DeclareRobustCommand{\Smiley}[2][1,2={opacity=0},usedefault]{%
508 \set@tkzsymbolsscl{#1}%
509 \if@tikzsymbols@absolute@draft%
510 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{Smile}%
511 \else%
512 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
513 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbolsscl,scale=#1]
514 \fill[#2] (0,0) circle (0.33);
515 \draw (0,0) circle (0.33);
516 \fill (-0.1,0.1) circle (0.05);
517 \fill (0.1,0.1) circle (0.05);

```

```

518 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
519 \end{tikzpicture}%
520 \fi%
521 \tikzsymbolsaftersymbolinput%
522 }%
523 \fi
524 \DeclareRobustCommand{\dSmiley}[3][1=1,2=yellow,3=yellow,usedefault]{%
525 \set@tkzsymlsscl{#1}%
526 \if@tikzsymbols@absolute@draft%
527 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dSmiley}%
528 \else%
529 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
530 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.1ex*\tkzsymlsscl,scale=#1]
531 \shade[ball color=#2] (0,0) circle (0.33);
532 \shade[ball color=black] (-0.1,0.1) circle (0.05);
533 \shade[ball color=black] (0.1,0.1) circle (0.05);
534 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
535 \end{tikzpicture}%
536 \fi%
537 \tikzsymbolsaftersymbolinput%
538 }

```

\Laughey \dLaughey A laughing Smiley

```

539 \DeclareRobustCommand{\Laughey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
540 \set@tkzsymlsscl{#1}%
541 \if@tikzsymbols@absolute@draft%
542 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{lachen}%
543 \else%
544 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
545 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
546 \fill[#2,line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
547 \draw[line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
548 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
549 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
550 \fill[#3,rounded corners=0.1ex*\tkzsymlsscl, yshift=-0.5]
551 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
552 \draw[rounded corners=0.1ex*\tkzsymlsscl, yshift=-0.5]
553 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
554 \end{tikzpicture}%
555 \fi%
556 \tikzsymbolsaftersymbolinput%
557 }
558 \DeclareRobustCommand{\dLaughey}[3][1=1,2=yellow, 3=red ,usedefault]{%
559 \set@tkzsymlsscl{#1}%
560 \if@tikzsymbols@absolute@draft%
561 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{lachen}%
562 \else%
563 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
564 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
565 \fill[ball color=#2,line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);

```

```

566 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
567 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
568 \shade[ball color=#3, rounded corners=0.1ex*\tkzsymlbsscl, yshift=-0.3]
569 (-0.25,-0.0) .. controls (-0.13,-0.26) and (0.13,-0.26) .. (0.25,-0.0) -- cycle;
570 \end{tikzpicture}%
571 \fi%
572 \tikzsymbolsaftersymbolinput%
573 }

\Neutrey \dNeutrey neutral Smiley :|
574 \DeclareRobustCommand{\Neutrey}[2][1,2={opacity=0},usedefault]{%
575 \set@tkzsymlbsscl{#1}%
576 \if@tikzsymbols@absolute@draft%
577 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{Neutral}%
578 \else%
579 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
580 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
581 \fill[#2,line width=0.12ex*\tkzsymlbsscl] (0,0) circle (0.33);
582 \draw[line width=0.12ex*\tkzsymlbsscl] (0,0) circle (0.33);
583 \fill (0.1,0.1) circle (0.05);
584 \fill (-0.1,0.1) circle (0.05);
585 \draw (-0.2,-0.1) -- (0.2,-0.1);
586 \end{tikzpicture}%
587 \fi%
588 \tikzsymbolsaftersymbolinput%
589 }

590 \DeclareRobustCommand{\dNeutrey}[2][1,2=yellow,usedefault]{%
591 \set@tkzsymlbsscl{#1}%
592 \if@tikzsymbols@absolute@draft%
593 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dneutral}%
594 \else%
595 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
596 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
597 \shade[ball color=#2] (0,0) circle (0.33);
598 \shade[ball color=black] (0.1,0.1) circle (0.05);
599 \shade[ball color=black] (-0.1,0.1) circle (0.05);
600 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
601 \end{tikzpicture}%
602 \fi%
603 \tikzsymbolsaftersymbolinput%
604 }

\Winkey \dWinkey ;)
605 \DeclareRobustCommand{\Winkey}[2][1,2={opacity=0},usedefault]{%
606 \set@tkzsymlbsscl{#1}%
607 \if@tikzsymbols@absolute@draft%
608 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{Wink}%
609 \else%
610 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
611 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlbsscl,scale=#1]

```

```

612 \fill[#2] (0,0) circle (0.33);
613 \draw(0,0) circle (0.33);
614 \draw(0.17,0.1) -- (0.05,0.1);
615 \fill (-0.1,0.1) circle (0.05);
616 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
617 \end{tikzpicture}%
618 \fi%
619 \tikzsymbolsaftersymbolinput%
620 }
621 \DeclareRobustCommand{\dWinkey}[2][1=1,2=yellow,usedefault]{%
622 \set@tkzsymlsscl{#1}%
623 \if@tikzsymbols@absolute@draft%
624 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dWink}%
625 \else%
626 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
627 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlsscl,scale=#1]
628 \shade[ball color=#2] (0,0) circle (0.33);
629 \draw(0.17,0.1) -- (0.05,0.1);
630 \shade[ball color=black] (-0.1,0.1) circle (0.05);
631 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
632 \end{tikzpicture}%
633 \fi%
634 \tikzsymbolsaftersymbolinput%
635 }

```

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

```

636 \DeclareRobustCommand{\Sey}[2][1=1,2={opacity=0} ,usedefault]{%
637 \set@tkzsymlsscl{#1}%
638 \if@tikzsymbols@absolute@draft%
639 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{S}%
640 \else%
641 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
642 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
643 \fill[#2, line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
644 \draw[line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
645 \fill (0.1,0.1) circle (0.05);
646 \fill (-0.1,0.1) circle (0.05);
647 \draw (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
648 \end{tikzpicture}%
649 \fi%
650 \tikzsymbolsaftersymbolinput%
651 }
652 \DeclareRobustCommand{\dSey}[2][1=1,2=yellow ,usedefault]{%
653 \set@tkzsymlsscl{#1}%
654 \if@tikzsymbols@absolute@draft%
655 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dS}%
656 \else%
657 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
658 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
659 \shade[ball color=#2] (0,0) circle (0.33);

```

```

660 \shade[ball color=black] (0.1,0.1) circle (0.05);
661 \shade[ball color=black] (-0.1,0.1) circle (0.05);
662 \draw[black] (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
663 \end{tikzpicture}%
664 \fi
665 \tikzsymbolsaftersymbolinput%
666 }

```

\Key \dKey I can't think of a better name again.

```

667 \DeclareRobustCommand{\Xkey}[2][1=1, 2={opacity=0}, usedefault]{%
668 \set@tkzsymlsscl{#1}%
669 \if@tikzsymbols@absolute@draft%
670 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{Xe}%
671 \else%
672 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{#1}\fi%
673 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
674 \fill[#2, line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
675 \draw[line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
676 \draw (0.05,0.05) -- ++ (0.1,0.1);
677 \draw (0.15,0.05) -- ++ (-0.1,0.1);
678 \draw (-0.05,0.05) -- ++ (-0.1,0.1);
679 \draw (-0.15,0.05) -- ++ (0.1,0.1);
680 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
681 \end{tikzpicture}%
682 \fi%
683 \tikzsymbolsaftersymbolinput%
684 }
685 \DeclareRobustCommand{\dXkey}[2][1=1, 2={yellow}, usedefault]{%
686 \set@tkzsymlsscl{#1}%
687 \if@tikzsymbols@absolute@draft%
688 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dXe}%
689 \else%
690 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{#1}\fi%
691 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
692 \fill[ball color=#2, line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
693 \draw (0.05,0.05) -- ++ (0.1,0.1);
694 \draw (0.15,0.05) -- ++ (-0.1,0.1);
695 \draw (-0.05,0.05) -- ++ (-0.1,0.1);
696 \draw (-0.15,0.05) -- ++ (0.1,0.1);
697 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
698 \end{tikzpicture}%
699 \fi%
700 \tikzsymbolsaftersymbolinput%
701 }

```

\Innocey \dInnocey An innocent Smiley

```

702 \DeclareRobustCommand{\Innocey}[3][1=1,2={opacity=0},3=yellow ,usedefault]{%
703 \set@tkzsymlsscl{#1}%
704 \if@tikzsymbols@absolute@draft%
705 \tikzsymbols@resizebox{#1ex*173/100}{#1ex*19/10}{Inno}%

```



```

706 \else%
707 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
708 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlblsscl,scale=#1]
709 \fill[#2] (0,0) circle (0.33);
710 \draw (0,0) circle (0.33);
711 \fill (-0.1,0.1) circle (0.05);
712 \fill (0.1,0.1) circle (0.05);
713 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
714 \draw[#3, line width=0.095ex*\tkzsymlblsscl] (0.32,0.31) arc (0:360:0.32 and 0.1);
715 \draw[line width=0.05ex*\tkzsymlblsscl] (0.3,0.31) arc (0:360:0.3 and 0.07);
716 \draw[line width=0.05ex*\tkzsymlblsscl] (0.35,0.31) arc (0:360:0.35 and 0.12);
717 \end{tikzpicture}%
718 \fi%
719 \tikzsymbolsaftersymbolinput%
720 }
721 \DeclareRobustCommand{\wInnocey}[1][1]{\Innocey[#1][opacity=0][white]}
722 \DeclareRobustCommand{\dInnocey}[3][1=1,2=yellow,3=yellow,usedefault]{%
723 \set@tkzsymlblsscl{#1}%
724 \if@tikzsymbols@absolute@draft%
725 \tikzsymbols@resizebox{#1ex*173/100}{#1ex*185/100}{\dInno}%
726 \else%
727 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
728 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlblsscl,scale=#1]
729 \shade[ball color=#2] (0,0) circle (0.33);
730 \shade[ball color=black] (-0.1,0.1) circle (0.05);
731 \shade[ball color=black] (0.1,0.1) circle (0.05);
732 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
733 \draw[color=#3!97!black, line width=0.09ex*\tkzsymlblsscl]
734 (0.32,0.31) arc (0:360:0.32 and 0.1);
735 \draw[line width=0.05ex*\tkzsymlblsscl] (0.3,0.31) arc (0:360:0.3 and 0.07);
736 \draw[line width=0.05ex*\tkzsymlblsscl] (0.35,0.31) arc (0:360:0.35 and 0.12);
737 \end{tikzpicture}%
738 \fi%
739 \tikzsymbolsaftersymbolinput%
740 }

```

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

```

741 \DeclareRobustCommand{\Cooley}[2][1=1,2={opacity=0} ,usedefault]{%
742 \set@tkzsymlblsscl{#1}%
743 \if@tikzsymbols@absolute@draft%
744 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*166/100}{\Cool}%
745 \else%
746 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
747 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlblsscl,scale=#1]
748 \fill[#2] (0,0) circle (0.33);
749 \draw (0,0) circle (0.33);
750 \fill[rounded corners=0.1ex*\tkzsymlblsscl]
751 (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
752 \fill[rounded corners=0.1ex*\tkzsymlblsscl]
753 (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;

```

```

754 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
755 \draw (0.02,0.1) -- (-0.02,0.1);
756 \draw (-0.2,0.1) -- (-0.3,0.13);
757 \draw (0.2,0.1) -- (0.3,0.13);
758 \end{tikzpicture}%
759 \fi%
760 \tikzsymbolsaftersymbolinput%
761 }
762 \DeclareRobustCommand{\dCooley}[2][1=1,2=yellow,usedefault]{%
763 \set@tkzsymbolsscl{#1}%
764 \if@tkzsymbols@absolute@draft%
765 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dCool}%
766 \else%
767 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
768 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbolsscl,scale=#1]
769 \shade[ball color=#2] (0,0) circle (0.33);
770 \draw[black] (0.02,0.1) -- (-0.02,0.1);
771 \draw[black] (-0.2,0.1) -- (-0.295,0.146);
772 \draw[black] (0.2,0.1) -- (0.295,0.146);
773 \shade[ball color=black,rounded corners=0.1ex*\tkzsymbolsscl]
774 (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
775 \shade[ball color=black,rounded corners=0.1ex*\tkzsymbolsscl]
776 (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
777 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
778 \end{tikzpicture}%
779 \fi%
780 \tikzsymbolsaftersymbolinput%
781 }

```

\Tongey \dTongey :P

```

782 \DeclareRobustCommand{\Tongey}[3][1=1,2={opacity=0},3={opacity=0} ,usedefault]{%
783 \set@tkzsymbolsscl{#1}%
784 \if@tkzsymbols@absolute@draft%
785 \tikzsymbols@resizebox{#1ex*171/100}{#1ex*165/100}{ton}%
786 \else%
787 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
788 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbolsscl,scale=#1]
789 \fill[#2] (0,0) circle (0.33);
790 \draw (0,0) circle (0.33);
791 \fill (-0.1,0.1) circle (0.05);
792 \fill (0.1,0.1) circle (0.05);
793 \fill[#3,line width=0.058ex*\tkzsymbolsscl, rounded corners=0.12ex*\tkzsymbolsscl]
794 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
795 \draw[line width=0.07ex*\tkzsymbolsscl, yshift=0.21ex]
796 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
797 \draw[line width=0.058ex*\tkzsymbolsscl, rounded corners=0.12ex*\tkzsymbolsscl]
798 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
799 \end{tikzpicture}%
800 \fi%
801 \tikzsymbolsaftersymbolinput%

```

```

802 }
803 \DeclareRobustCommand{\dTongey}[3][1=1,2=yellow,3=red,usedefault]{%
804 \set@tkzsymlblsscl{#1}%
805 \if@tikzsymbols@absolute@draft%
806 \tikzsymbols@resizebox{#1ex*159/100}{#1ex*156/100}{dton}%
807 \else%
808 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
809 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlblsscl,scale=#1]
810 \shade[ball color=#2] (0,0) circle (0.33);
811 \shade[ball color=black] (-0.1,0.1) circle (0.05);
812 \shade[ball color=black] (0.1,0.1) circle (0.05);
813 \shade[ball color=#3,line width=0.058ex*\tkzsymlblsscl, rounded corners=0.12ex*\tkzsymlblsscl]
814 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
815 \draw[black, line width=0.058ex*\tkzsymlblsscl, rounded corners=0.12ex*\tkzsymlblsscl]
816 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
817 \draw[black, line width=0.07ex*\tkzsymlblsscl, yshift=0.21ex]
818 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
819 \end{tikzpicture}%
820 \fi%
821 \tikzsymbolsaftersymbolinput%
822 }

```

\Nursey \dNursey a Nurse (the cross has nothing to do with religion).

```

823 \DeclareRobustCommand{\Nursey}[4][1=1,2={opacity=0},3={opacity=0},4=black,usedefault]{%
824 \set@tkzsymlblsscl{#1}%
825 \if@tikzsymbols@absolute@draft%
826 \tikzsymbols@resizebox{#1ex*15/10}{#1ex*215/100}{Nurse}%
827 \else%
828 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
829 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymlblsscl,scale=#1]
830 \fill[#3,rounded corners=.023ex*\tkzsymlblsscl]
831 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
832 \fill[#2] (0,0) circle (0.3);
833 \draw (0,0) circle (0.3);
834 \fill (-0.1,0.1) circle (0.05);
835 \fill (0.1,0.1) circle (0.05);
836 \draw[line width=0.09ex*\tkzsymlblsscl, yshift=0.07ex]
837 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
838 \draw[rounded corners=.023ex*\tkzsymlblsscl]
839 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
840 \draw[#4,line width=.046ex*\tkzsymlblsscl] (0,0.35) -- (0,0.5);
841 \draw[#4,line width=.046ex*\tkzsymlblsscl] (-0.05,0.45) -- (0.05,0.45);
842 \end{tikzpicture}%
843 \fi%
844 \tikzsymbolsaftersymbolinput%
845 }
846 \DeclareRobustCommand{\dNursey}[4][1=1,2=yellow,3=white,4=red,usedefault]{%
847 \set@tkzsymlblsscl{#1}%
848 \if@tikzsymbols@absolute@draft%
849 \tikzsymbols@resizebox{#1ex*14/10}{#1ex*2}{dNurse}%

```

```

850 \else%
851 \ifdim\tkzsymlbsscl<Opt\set@tkzsymlbsscl{-#1}\fi%
852 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymlbsscl,scale=#1]
853 \shade[ball color=#2] (0,0) circle (0.3);
854 \shade[ball color=black] (-0.1,0.1) circle (0.05);
855 \shade[ball color=black] (0.1,0.1) circle (0.05);
856 \draw[black, line width=0.09ex*\tkzsymlbsscl, yshift=0.07ex]
857 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
858 \shade[ball color=#3, rounded corners=.023ex*\tkzsymlbsscl,yshift=-0.09ex]
859 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0) arc (0:180:0.3);
860 \shade[ball color=#4,line width=.046ex*\tkzsymlbsscl]
861 (-0.01,0.31) -- (-0.01,0.46) -- (0.01,0.46) -- (0.01,0.31)--cycle;
862 \shade[ball color=#4,line width=.046ex*\tkzsymlbsscl]
863 (-0.05,0.4) -- (0.05,0.4) -- (0.05,0.42)--(-0.05,0.42) -- cycle;
864 \end{tikzpicture}%
865 \fi%
866 \tikzsymbolsaftersymbolinput%
867 }

\Vomey \dVomey *Bläärgh*

868 \DeclareRobustCommand{\Vomey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
869 \set@tkzsymlbsscl{#1}%
870 \if@tikzsymbols@absolute@draft%
871 \tikzsymbols@resizebox{#1ex*305/100}{#1ex*165/100}{vomit}%
872 \else%
873 \ifdim\tkzsymlbsscl<Opt\set@tkzsymlbsscl{-#1}\fi%
874 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
875 \fill[#2,rounded corners=0.05ex*\tkzsymlbsscl] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
876 \draw[rounded corners=0.05ex*\tkzsymlbsscl] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
877 \draw[line width=0.05ex*\tkzsymlbsscl] (-0.5,0.3) -- (-0.3,0.1);
878 \fill (-0.45,0.27) arc (100:350:0.1);
879 \fill[#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
880 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
881 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
882 \fill[#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) ..
883 controls (1.7,-1.2) and (1.3,-1.2) .. (1,-1) ..
884 controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
885 \draw (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1);
886 \draw (0,-0.4) .. controls (0.5,-0.5) and (0.8,-0.7) .. (1,-1);
887 \draw (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
888 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2)
889 and (1,-2) .. (1,-1.5) .. controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
890 \end{tikzpicture}%
891 \fi%
892 \tikzsymbolsaftersymbolinput%
893 }

894 \DeclareRobustCommand{\dVomey}[3][1=1,2=yellow,3={brown!10!olive},usedefault]{%
895 \set@tkzsymlbsscl{#1}%
896 \if@tikzsymbols@absolute@draft%
897 \tikzsymbols@resizebox{#1ex*295/100}{#1ex*156/100}{dvomit}%

```

```

898 \else%
899 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
900 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
901 \shade[ball color=#2!90!brown,rounded corners=0.03ex*\tkzsymlbsscl]
902 (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
903 \draw[black, line width=0.05ex*\tkzsymlbsscl] (-0.5,0.3) -- (-0.3,0.1);
904 \shade[ball color=black] (-0.45,0.27) arc (100:350:0.1);
905 \shade[ball color=#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
906 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
907 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
908 \shade[ball color=#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) .. controls
909 (1.7,-1.2) and (1.3,-1.2) .. (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
910 \end{tikzpicture}%
911 \fi%
912 \tikzsymbolsaftersymbolinput%
913 }

\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.
914 \DeclareRobustCommand{\Walley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
915 \set@tkzsymlbsscl{#1}%
916 \if@tikzsymbols@absolute@draft%
917 \tikzsymbols@resizebox{#1ex*235/100}{#1ex*165/100}{Wall}%
918 \else%
919 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
920 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1,
921 decoration={random steps,segment length=0.15ex*\tkzsymlbsscl, amplitude=0.1ex*\tkzsymlbsscl}]
922 \fill[#2, line width=0.08ex*\tkzsymlbsscl] (0,0) circle (0.28);
923 \draw[line width=0.08ex*\tkzsymlbsscl] (0,0) circle (0.28);
924 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
925 \draw (0.28,-0.33) rectangle (0.66,0.33);
926 \draw[line width=0.06ex*\tkzsymlbsscl]
927 (0.28,0) ---+(0.05,0.07) ---+(0.03,0.02) ---
928 +(0.03,-0.02) ---+(0.03,0.1) ---+(0.03,0.02) -- (0.5,0.25);
929 \draw[line width=0.06ex*\tkzsymlbsscl]
930 (0.28,0) ---+(0.06,-0.02) ---+(0.04,0.06) ---
931 +(0.0,-0.08) ---+(0.08,0.06) ---+(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);
932 \draw[line width=0.06ex*\tkzsymlbsscl]
933 (0.28,0) ---+(0.03,-0.02) ---+(0.03,-0.07) ---
934 +(0.03,-0.01) ---+(0.01,-0.07) ---+(0.06,0.01) ---+(0.03,-0.08) --
935 (0.5,0.-0.25);
936 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
937 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
938 \end{tikzpicture}%
939 \fi%
940 \tikzsymbolsaftersymbolinput%
941 }

942 \DeclareRobustCommand{\rWalley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
943 \set@tkzsymlbsscl{#1}%
944 \if@tikzsymbols@absolute@draft%

```

```

945 \tikzsymbols@resizebox{#1ex*235/100}{#1ex*165/100}{rWall}%
946 \else%
947 \ifdim\tkzsymlbsscl<Opt\set\tkzsymlbsscl{-#1}\fi%
948 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1,
949 decoration={random steps,segment length=0.15ex*\tkzsymlbsscl, amplitude=0.1ex*\tkzsymlbsscl}]
950 \fill[#2, line width=0.08ex*\tkzsymlbsscl] (0,0) circle (0.28);
951 \draw[line width=0.08ex*\tkzsymlbsscl] (0,0) circle (0.28);
952 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
953 \draw (0.28,-0.33) rectangle (0.66,0.33);
954 \draw[decorate, line width=0.06ex*\tkzsymlbsscl] (0.28,0) -- (0.5,0.25);
955 \draw[decorate,line width=0.06ex*\tkzsymlbsscl] (0.28,0) -- (0.6,0.0);
956 \draw[decorate,line width=0.06ex*\tkzsymlbsscl] (0.28,0) -- (0.5,-0.25);
957 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
958 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
959 \end{tikzpicture}%
960 \fi%
961 \tikzsymbolsaftersymbolinput%
962 }
963 \DeclareRobustCommand{\dWalley}[2][1=1, 2=yellow, usedefault]{%
964 \set\tkzsymlbsscl{#1}%
965 \if\tikzsymbols@absolute@draft%
966 \tikzsymbols@resizebox{#1ex*242/100}{#1ex*156/100}{dWall}%
967 \else%
968 \ifdim\tkzsymlbsscl<Opt\set\tkzsymlbsscl{-#1}\fi%
969 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1,
970 decoration={random steps,segment length=0.15ex*\tkzsymlbsscl, amplitude=0.1ex*\tkzsymlbsscl}]
971 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
972 \draw[line width=0.06ex*\tkzsymlbsscl]
973 (0.28,0) ---+(0.05,0.07) ---+(0.03,0.02) ---+
974 +(0.03,-0.02) ---+(0.03,0.1) ---+(0.03,0.02) -- (0.5,0.25);
975 \draw[line width=0.06ex*\tkzsymlbsscl]
976 (0.28,0) ---+(0.06,-0.02) ---+(0.04,0.06) ---+
977 +(0.0,-0.08) ---+(0.08,0.06) ---+(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);
978 \draw[line width=0.06ex*\tkzsymlbsscl]
979 (0.28,0) ---+(0.03,-0.02) ---+(0.03,-0.07) ---+
980 +(0.03,-0.01) ---+(0.01,-0.07) ---+(0.06,0.01) ---+(0.03,-0.08) -- (0.5,0.-0.25);
981 \shade[ball color=#2, line width=0.08ex*\tkzsymlbsscl] (-0.01,0) circle (0.31);
982 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
983 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
984 \end{tikzpicture}%
985 \fi%
986 \tikzsymbolsaftersymbolinput%
987 }
988 \DeclareRobustCommand{\drWalley}[2][1=1, 2=yellow, usedefault]{%
989 \set\tkzsymlbsscl{#1}%
990 \if\tikzsymbols@absolute@draft%
991 \tikzsymbols@resizebox{#1ex*242/100}{#1ex*156/100}{drWall}%
992 \else%
993 \ifdim\tkzsymlbsscl<Opt\set\tkzsymlbsscl{-#1}\fi%
994 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1,

```

```

995 decoration={random steps,segment length=0.15ex*\tkzsymlblsscl, amplitude=0.1ex*\tkzsymlblsscl}}
996 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
997 \draw[decorate, line width=0.06ex*\tkzsymlblsscl] (0.298,0) -- (0.5,0.25);
998 \draw[decorate,line width=0.06ex*\tkzsymlblsscl] (0.298,0) -- (0.6,0.0);
999 \draw[decorate,line width=0.06ex*\tkzsymlblsscl] (0.298,0) -- (0.5,-0.25);
1000 \shade[ball color=#2, line width=0.08ex*\tkzsymlblsscl] (-0.01,0) circle (0.31);
1001 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1002 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1003 \end{tikzpicture}%
1004 \fi
1005 \tikzsymbolsaftersymbolinput%
1006 }

```

\Cat *Miau*

```

1007 \DeclareRobustCommand{\Cat}[1][1=1,usedefault]{%
1008 \set@tkzsymlblsscl{#1}%
1009 \if@tikzsymbols@absolute@draft%
1010 \tikzsymbols@resizebox{#1ex*19/10}{#1ex*19/10}{Cat}%
1011 \else%
1012 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
1013 \begin{tikzpicture}[x=2.33ex,y=2.33ex, line width=0.093ex*\tkzsymlblsscl,scale=#1]
1014 \draw (0,0) circle (0.3);
1015 \draw[rounded corners=0.163ex*\tkzsymlblsscl] (-0.3,0) -- (-0.35,0.5) -- (0,0.3);
1016 \draw[rounded corners=0.163ex*\tkzsymlblsscl] (0,0.3) -- (0.35,0.5) -- (0.3,0);
1017 \fill (-0.15,.15) circle (0.05);
1018 \fill (0.15,.15) circle (0.05);
1019 \draw[rounded corners=0.175ex*\tkzsymlblsscl,yshift=-0.12ex]
1020 (0,0) -- (0,-0.1) -- (-0.1,-0.095);
1021 \draw[rounded corners=0.175ex*\tkzsymlblsscl,yshift=-0.12ex]
1022 (0,0) -- (0,-0.1) -- (0.1,-0.095);
1023 \draw[rounded corners=.12ex*\tkzsymlblsscl,yshift=-.15ex, line width=0.03em*(#1-.#1)]
1024 (-0.1,0.1) -- (0,0) -- (0.1,0.1) -- cycle ;
1025 \draw[line width=0.035ex*\tkzsymlblsscl]
1026 (-0.1,-0.05)..controls(-0.25,0)and(-0.35,0).. (-0.4,-0.05);
1027 \draw[line width=0.035ex*\tkzsymlblsscl](-0.1,-0.05)..
1028 controls(-0.25,-0.01)and(-0.35,-0.09).. (-0.4,-0.14);
1029 \draw[line width=0.035ex*\tkzsymlblsscl](-0.1,-0.05)..
1030 controls(-0.25,-0.045)and(-0.35,-0.13).. (-0.4,-0.23);
1031 \draw[line width=0.035ex*\tkzsymlblsscl]
1032 (0.1,-0.05)..controls(0.25,0)and(0.35,0).. (0.4,-0.05);
1033 \draw[line width=0.035ex*\tkzsymlblsscl]
1034 (0.1,-0.05)..controls(0.25,-0.01)and(0.35,-0.09).. (0.4,-0.14);
1035 \draw[line width=0.035ex*\tkzsymlblsscl]
1036 (0.1,-0.05)..controls(0.25,-0.045)and(0.35,-0.13).. (0.4,-0.23);
1037 \end{tikzpicture}%
1038 \fi%
1039 \tikzsymbolsaftersymbolinput%
1040 }

```

\Ninja \dNinja A Ninja.

```

1041 \DeclareRobustCommand{\Ninja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1042 \set@tkzsymbolsscl{#1}%
1043 \if@tikzsymbols@absolute@draft%
1044 \tikzsymbols@resizebox{#1ex*215/100}{#1ex*165/100}{sshh}%
1045 \else%
1046 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1047 \def\Black@is@Black{black}%
1048 \def\Black@or@not@Black{#2}%
1049 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1,
1050 decoration={random steps,segment length=0.1ex*\tkzsymbolsscl, amplitude=0.1ex*\tkzsymbolsscl}]
1051 \fill[#2, line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.33);
1052 \draw (-0.2,-0.125) -- ++(0.4,0);
1053 \fill[decoration={random steps,segment length=0.1ex*\tkzsymbolsscl,
1054 amplitude=0.01ex*\tkzsymbolsscl}, decorate,#3]
1055 (-0.33,0) -- (0.33,0) -- (0.23,0.23) -- (-0.23,0.23) -- cycle;
1056 \ifx\Black@or@not@Black\Black@is@Black
1057 \draw[line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.33);\fi
1058 \fill[#3] (0,0.1) -- (-0.33,0) -- (-0.26,0.23);
1059 \fill[#3] (0.3465,0) arc (0:42:0.34 and 0.345) --
1060 (0.2,0.23)-- (0.31,0.0) -- cycle;
1061 \fill[#3] (-0.3465,0) arc (0:-42:-0.34 and -0.345) --
1062 (-0.2,0.23)-- (-0.31,0.0) -- cycle;
1063 \fill[#4] (0.129,0.1425) arc (55:-180:.05);
1064 \fill[#4] (-0.129,0.1425) arc (-55:180:-.05);
1065 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1066 segment length=0.55ex*\tkzsymbolsscl}, #3]
1067 (0.26,0.21) -- (0.5,0.35);
1068 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1069 segment length=0.55ex*\tkzsymbolsscl}, #3]
1070 (0.26,0.21) -- (0.53,0.1);
1071 \ifx\Black@or@not@Black\Black@is@Black
1072 \else\draw[line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.33);\fi
1073 \end{tikzpicture}%
1074 \fi%
1075 \tikzsymbolsaftersymbolinput%
1076 }
1077 \DeclareRobustCommand{\dNinja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1078 \set@tkzsymbolsscl{#1}%
1079 \if@tikzsymbols@absolute@draft%
1080 \tikzsymbols@resizebox{#1ex*215/100}{#1ex*17/10}{dsshh}%
1081 \else%
1082 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1083 \def\Black@is@Black{black}%
1084 \def\Black@or@not@Black{#2}%
1085 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1,
1086 decoration={random steps,segment length=0.1ex*\tkzsymbolsscl, amplitude=0.1ex*\tkzsymbolsscl}]
1087 \draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1088 segment length=0.55ex*\tkzsymbolsscl},decorate, #3!50!black]
1089 (0.26,0.21) -- (0.5,0.35);
1090 \draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,

```



```

1091 segment length=0.5ex*\tkzsymlblsscl},decorate, #3!50!black]
1092 (0.26,0.21) -- (0.53,0.1);
1093 \shade[ball color=#2, line width=0.08ex*\tkzsymlblsscl] (0,0) circle (0.347);
1094 %\draw (-0.2,-0.125) -- ++(0.4,0);
1095 \ifx\Black@or@not@Black\Black@is@Black
1096 \draw[line width=0.08ex*\tkzsymlblsscl] (0,0) circle (0.33);\fi
1097 \fill[decoration={random steps,segment length=0.1ex*\tkzsymlblsscl,
1098 amplitude=0.01ex*\tkzsymlblsscl},ball color=#3]
1099 decorate {(-0.33,0) -- (0.3465,0) }
1100 {arc (0:42:0.34 and 0.345)}
1101 decorate {-- (-0.25,0.24)}
1102 { arc (-42:0:-0.375 and -0.345)};
1103 \shade[ball color=#4] (0.129,0.1425) arc (55:-180:.05);
1104 \shade[ball color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1105 \shade[top color=#4!80!black, bottom color=#4] (0.129,0.1425) arc (55:-180:.05);
1106 \shade[top color=#4!80!black, bottom color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1107 \end{tikzpicture}%
1108 \fi%
1109 \tikzsymbolsaftersymbolinput%
1110 }

```

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

```

1111 \DeclareRobustCommand{\NiceReapey}[1][1=1,usedefault]{%
1112 \set@tkzsymlblsscl{#1}%
1113 \if@tikzsymbols@absolute@draft%
1114 \tikzsymbols@resizebox{#1ex*264/100}{#1ex*155/100}{Ann}%
1115 \else%
1116 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
1117 \begin{tikzpicture}[x=0.11em,y=0.11em, line width=0.07ex*\tkzsymlblsscl,scale=#1]
1118 \draw (1.7,-1) arc (360:180:1.7 and 2)
1119 arc (260:110:1.5 and 2) .. controls (-1,3.3) and (1,3.3) .. (1.9,2.97)
1120 arc (260:100:-1.3 and -2) -- cycle;
1121 \fill[black!20!white] (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5) -- cycle;
1122 \draw (3,-3) -- (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5);
1123 \draw (0,-1.5) circle (1 and 0.5);
1124 \draw[line width=0.04ex*\tkzsymlblsscl] (-0.2,-1) -- (-0.2,-2);
1125 \draw[line width=0.04ex*\tkzsymlblsscl] (0.2,-1) -- (0.2,-2);
1126 \draw[line width=0.04ex*\tkzsymlblsscl] (0.6,-1) -- (0.6,-2);
1127 \draw[line width=0.04ex*\tkzsymlblsscl] (-0.6,-1) -- (-0.6,-2);
1128 \draw[line width=0.04ex*\tkzsymlblsscl] (-1,-1.5) -- (1,-1.5);
1129 \fill (1.25,1.25) circle ( 0.5 and 0.75);
1130 \fill (-1.25,1.25) circle ( 0.5 and 0.75);
1131 \end{tikzpicture}%
1132 \fi%
1133 \tikzsymbolsaftersymbolinput%
1134 }

```

6.3 Other symbols(s)

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

```

1135 \DeclareRobustCommand{\Person}[5][1=1,2=-22,3=22,4=27,5=-27,usedefault]{%
1136 \set@tkzsymbolsscl{#1}%
1137 \if@tikzsymbols@absolute@draft%
1138 \tikzsymbols@resizebox{#1ex*78/100}{#1ex*155/100}{P}%
1139 \else%
1140 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1141 \begin{tikzpicture}[line width=0.12ex*\tkzsymbolsscl, scale=#1, x=1.35ex, y=1.35ex]
1142 \draw[rotate around={#5:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1143 \draw[rotate around={#4:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1144 \draw (.15,.2) -- (.15,.4);
1145 \draw[rotate around={#3:(.15,.4)}] (.15,.4) -- (.42,.4);
1146 \draw[rotate around={#2:(.15,.4)}] (.15,.4) -- (-0.12,.4);
1147 \draw (.15,.4) -- (.15,.53);
1148 \draw (.15,.8) circle (0.18);
1149 \end{tikzpicture}%
1150 \fi%
1151 \tikzsymbolsaftersymbolinput%
1152 }
```

\Candle A burning candle

```

1153 \DeclareRobustCommand{\Candle}[1][1]{%
1154 \set@tkzsymbolsscl{#1}%
1155 \if@tikzsymbols@absolute@draft%
1156 \tikzsymbols@resizebox{#1ex*63/100}{#1ex*165/100}{C}%
1157 \else%
1158 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1159 \begin{tikzpicture}[x=1ex, y=1ex, scale=#1, line width=0.07ex*\tkzsymbolsscl]
1160 \draw[rounded corners=0.04ex*\tkzsymbolsscl] (0,0) -- (0.2,0) -- +(0,1) -- (0,1) -- cycle;
1161 \draw[line width=0.05ex*\tkzsymbolsscl] (0.1,1) -- (0.1,1.2);
1162 \draw[xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymbolsscl]
1163 (-0.1,0.6) .. controls (-0.4,0.8) and (-0.1,1) .. (-0.1,1.2);
1164 \draw[xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymbolsscl]
1165 (-0.1,0.6) .. controls (0.2,0.8) and (-0.1,1) .. (-0.1,1.2);
1166 \end{tikzpicture}%
1167 \fi%
1168 \tikzsymbolsaftersymbolinput%
1169 }
```

\Fire Just a fire.

```

1170 \DeclareRobustCommand{\Fire}[1][1]{%
1171 \set@tkzsymbolsscl{#1}%
1172 \if@tikzsymbols@absolute@draft%
1173 \tikzsymbols@resizebox{#1ex*158/100}{#1ex*16/10}{fire}%
1174 \else%
1175 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1176 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.07ex*\tkzsymbolsscl,rotate=45, scale=#1]
1177 \fill (-0.05,0) -- (0.05,0) -- (0.05,0.95) -- (-0.05,0.95) -- cycle;
```

```

1178 \fill (-0.74,0.7) -- (0.19,0.7) -- (0.19,0.8) -- (-0.74,0.8) -- cycle;
1179 \fill[rotate=-20, xshift=-1.3, yshift=-0.1]
1180 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1181 \fill[rotate=-70, xshift=-3.3, yshift=-2.3]
1182 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1183 \fill[rotate=135, xshift=2.5, yshift=-3.8]
1184 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1185 \draw[rotate=-45, xshift=-2.6, yshift=1.5, line width=0.04ex*\tkzsymbolsscl, x=0.5ex, y=0.5ex]
1186 (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0.05,1.7);
1187 \draw[rotate=-45, xshift=-2.1, yshift=1.5, line width=0.04ex*\tkzsymbolsscl, x=0.5ex, y=0.5ex]
1188 (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.15,1.7);
1189 \draw[rotate=-45, xshift=-2.5] (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0,1.5);
1190 \draw[rotate=-45, xshift=-2] (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.1,1.5);
1191 \end{tikzpicture}%
1192 \fi%
1193 \tikzsymbolsaftersymbolinput%
1194 }

```

\Coffeecup Just a cup of coffee.

```

1195 \if@tikzsymbols@marvosym\relax\else%
1196 \DeclareRobustCommand{\Coffeecup}[1][1]{%
1197 \set@tkzsymbolsscl{#1}%
1198 \if@tikzsymbols@absolute@draft%
1199 \tikzsymbols@resizebox{#1ex*181/100}{#1ex*165/100}{coffe}%
1200 \else%
1201 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1202 \begin{tikzpicture}[x=0.7ex,y=0.7ex, scale=#1, line width=0.07ex*\tkzsymbolsscl,
1203 decoration={snake,amplitude=.05ex*\tkzsymbolsscl,segment length=0.408ex*\tkzsymbolsscl}]
1204 \draw (0,0) arc (180:269:0.8 and 1) -- ++(0.5,0) arc (269:360:0.8 and 1) -- cycle;
1205 \draw (2.1,-0.15) -- (2.2,-0.15) arc (90:-90:0.3) -- (1.8,-0.75);
1206 \draw[line width=0.05ex*\tkzsymbolsscl, decorate]
1207 (0.4,0.3) -- +(0,1);
1208 \draw[line width=0.05ex*\tkzsymbolsscl, decorate]
1209 (1,0.3) -- +(0,1);
1210 \draw[line width=0.05ex*\tkzsymbolsscl, decorate]
1211 (1.6,0.3) -- +(0,1);
1212 \draw (0,-1.05) -- (2.1,-1.05);
1213 \end{tikzpicture}%
1214 \fi%
1215 \tikzsymbolsaftersymbolinput%
1216 }%
1217 \fi

```

\Chair A chair.

```

1218 \DeclareRobustCommand{\Chair}[1][1]{%
1219 \set@tkzsymbolsscl{#1}%
1220 \if@tikzsymbols@absolute@draft%
1221 \tikzsymbols@resizebox{#1ex*112/100}{#1ex*165/100}{chair}%
1222 \else%
1223 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1} \@tkzssmbles@negtrue\fi%

```

```

1224 \begin{tikzpicture}[x=0.9ex,y=0.9ex, scale=#1, line width=0.07ex*\tkzsymbolsscl]
1225 \draw (0,-0.5) -- (0,0.7) -- (0.5,1) -- (0.5,0.25);
1226 \draw[line width=0.06ex*\tkzsymbolsscl] (0,0.4) -- (0.5,0.7);
1227 \draw (0,0) -- (0.5,0.3) -- (1,0) --(1,-0.5);
1228 \if@tkzssmbles@neg\draw (0.5,0.3) -- +(0,-0.5);\fi
1229 \draw (0.5,-0.3) -- (0.5,-0.8);
1230 \draw (1,0) -- (0.5,-0.3) -- (0,0);
1231 \end{tikzpicture}%
1232 \fi%
1233 \@tkzssmbles@negfalse%
1234 \tikzsymbolsaftersymbolinput%
1235 }

```

\Bed A bed.

```

1236 \DeclareRobustCommand{\Bed}[1][1]{%
1237 \set@tkzsymbolsscl{#1}%
1238 \if@tikzsymbols@absolute@draft%
1239 \tikzsymbols@resizebox{#1ex*309/100}{#1ex*162/100}{\Bed}%
1240 \else%
1241 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1242 \begin{tikzpicture}[x=1ex,y=1ex, scale=#1, line width=0.08ex*\tkzsymbolsscl]
1243 \draw (0,0) -- (0,1.6);
1244 \draw (3,0) -- (3,1.2);
1245 \draw (0,0.5) -- (3,0.5);
1246 \draw (0,0.35) -- (3,0.35);
1247 \draw (0.7,0.5) arc (0:90:0.7);
1248 \draw (0.7,0.5) arc(180:30:1.231 and 0.6);
1249 \end{tikzpicture}%
1250 \fi%
1251 \tikzsymbolsaftersymbolinput%
1252 }

```

\Tribar Also called Penrose-Triangle

```

1253 \DeclareRobustCommand{\Tribar}[4][1=1,2={opacity=0},3={opacity=0},4={opacity=0},usedefault]
1254 {%
1255 \set@tkzsymbolsscl{#1}%
1256 \if@tikzsymbols@absolute@draft%
1257 \tikzsymbols@resizebox{#1ex*17/10}{#1ex*167/100}{\T}%
1258 \else%
1259 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1260 \begin{tikzpicture}[x=0.65ex,y=0.65ex,scale=#1,
1261   rounded corners=0.03ex*\tkzsymbolsscl, line width=0.06ex*\tkzsymbolsscl]
1262 \fill[#2] (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1263   -- (-0.65,-0.6) -- (0.35,1.3) -- +(0.15,-0.3);
1264 \fill[#3] (0,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- (-0.1,-0.3);
1265 \fill[#4] (1,0) -- (0.35,1.3) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (1.3,0);
1266 \draw (0,0) -- (1,0) -- (0.5,1) -- cycle;
1267 \draw (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1268   -- (-0.65,-0.6) -- (0.35,1.3) -- (0.8,.4);
1269 \draw (0.9,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- +(-.05,-0.1);

```

```

1270 \draw (-0.6,-0.6) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (0.4,1.9);
1271 \end{tikzpicture}%
1272 \fi%
1273 \tikzsymbolsaftersymbolinput%
1274 }

\Moai From the Easter Island: a Moai.
1275 \DeclareRobustCommand{\Moai}[1][1=1,usedefault]{%
1276 \set@tkzsymlsscl{#1}%
1277 \if@tkzsymbols@absolute@draft%
1278 \tikzsymbols@resizebox{#1ex*103/100}{#1ex*161/100}{M}%
1279 \else%
1280 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
1281 \ifdim\tkzsymlsscl<2pt%
1282 \def\tikzsymbolsMoaitickness{0.05ex}%
1283 \else%
1284 \ifdim\tkzsymlsscl<5pt%
1285 \def\tikzsymbolsMoaitickness{0.035ex}%
1286 \else%
1287 \def\tikzsymbolsMoaitickness{0.03ex}%
1288 \fi\fi%
1289 \begin{tikzpicture}[x=.13ex, y=.13ex, rounded corners=0.01ex*\tkzsymlsscl, scale=#1,
1290 line width=\tikzsymbolsMoaitickness*\tkzsymlsscl]
1291 \draw (-2.6,-4.25) -- (-2.5,-5.8)
1292 ..controls (-2,-6.8) and (1.5,-6.8) .. (2.2,-5.8) -- (2.4,-3.95);
1293 \draw(-2.5,2.5) .. controls (-2.9,4.6) and (2,5) .. (3.3,2.5) -- (2.9,-3.4)
1294 .. controls (2,-5) and (-4,-5) .. (-3.1,-3) -- cycle;
1295 \draw (-2.5,3) -- (-2,5) .. controls (0,6) and (2,5.8) .. (3.1,4.7) -- (3.3,2.5);
1296 \draw[line width=0.02ex*\tkzsymlsscl]
1297 (-2.2,-1.8) .. controls (-1,-1.3) and (0,-1.7) .. (1,-2);
1298 \draw[line width=0.02ex*\tkzsymlsscl]
1299 (-2.2,-1.8) .. controls (-1,-1) and (0,-1.4) .. (1,-2);
1300 \draw[line width=0.02ex*\tkzsymlsscl]
1301 (-2.2,-1.8) .. controls (-1,-2) and (0,-2) .. (1,-2);
1302 \draw (-0.8,4) .. controls (-0.8,3) and (-0.8,2) .. (-1.6,0.5) -- (-1.8,-0.4)
1303 .. controls (-1,0.2) and (0,0.2) .. (0.6,-0.4) -- (0.7,0.4)
1304 .. controls (0,1) and (0,2) .. (0.8,4);
1305 \draw (-1.8,-0.4) .. controls (-0.5,-0.5) and (0,-0.5) .. (0.6,-0.4);
1306 \draw (3.2,3.5) -- (3.7,3.5) .. controls (3.5,2) and (3.5,2) .. (3.6,-1.5) -- (3,-1.9);
1307 \draw (-2.5,3) .. controls (-2.7,2) and (-3,1) .. (-2.88,-1);
1308 \draw (-2.5,2.8) .. controls (-2,2.5) and (-1,3) .. (-0.8,3.1);
1309 \draw (0.5,3.3) .. controls (1,3) and (1,2.5) .. (3.3,2.4);
1310 \end{tikzpicture}%
1311 \fi%
1312 \tikzsymbolsaftersymbolinput%
1313 }

\Snowman A snowman. I think his smile is scary.
1314 \DeclareRobustCommand{\Snowman}[1][1]{%
1315 \set@tkzsymlsscl{#1}%

```

```

1316 \if@tikzsymbols@absolute@draft%
1317 \tikzsymbols@resizebox{#1ex*153/100}{#1ex*169/100}{Snow}%
1318 \else%
1319 \ifdim\tikzsymbolsscl<0pt\set\tikzsymbolsscl{-#1}\fi%
1320 \begin{tikzpicture}[x=0.9ex,y=0.9ex,line width=0.07ex*\tikzsymbolsscl, scale=#1]
1321 \draw (0,0) circle (0.4 and 0.35);
1322 \draw[line width=0.06ex*\tikzsymbolsscl] (0,0.64) circle (0.3 and 0.28);
1323 \draw[line width=0.05ex*\tikzsymbolsscl] (0,1.14) circle (0.2 and 0.2);
1324 \draw[rounded corners=0.1ex*\tikzsymbolsscl,line width=0.05ex*\tikzsymbolsscl,
1325 rotate around={-30:(0,1.14)}]
1326 (-0.2,1.15) -- ++(0,0.35) -- +(0.4,0) -- (0.2,1.14);
1327 \draw[rounded corners=0.07ex*\tikzsymbolsscl,line width=0.05ex*\tikzsymbolsscl,
1328 rotate around={-30:(0,1.14)}]
1329 (-0.2,1.19) arc (270:90:0.1);
1330 \fill (0,0.78) circle (0.04);
1331 \fill (0,0.63) circle (0.04);
1332 \fill (0,0.48) circle (0.04);
1333 \fill (0,0.2) circle (0.05);
1334 \fill (0,0) circle (0.05);
1335 \fill (0,-0.2) circle (0.05);
1336 \fill (-0.06,1.18) circle (0.045);
1337 \fill (0.06,1.18) circle (0.045);
1338 \fill (0.1,1.08) circle (0.015);
1339 \fill (-0.1,1.08) circle (0.015);
1340 \fill (0.06,1.055) circle (0.015);
1341 \fill (-0.06,1.055) circle (0.015);
1342 \fill (0.02,1.039) circle (0.015);
1343 \fill (-0.02,1.039) circle (0.015);
1344 \draw (-0.3,0.7) -- (-0.6,0.8);
1345 \draw (-0.6,0.8) -- (-0.75,0.7);
1346 \draw (-0.6,0.8) -- (-0.55,1);
1347 \draw (-0.6,0.8) -- (-0.8,0.9);
1348 \draw[line width=0.06ex*\tikzsymbolsscl] (-0.65,0) -- (-0.65,1);
1349 \foreach\x in {-0.85, -0.75,-0.65,-0.55,-0.45}
1350 \draw[line width=0.05ex*\tikzsymbolsscl] (-0.65,1) -- (\x,1.3);
1351 \draw (0.3,0.7) -- (0.6,0.8);
1352 \draw (0.6,0.8) -- (0.75,0.7);
1353 \draw (0.6,0.8) -- (0.6,1);
1354 \draw (0.6,0.8) -- (0.8,0.9);
1355 \end{tikzpicture}%
1356 \fi%
1357 \tikzsymbolsaftersymbolinput%
1358 }

```

6.4 Trees

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

`\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:

```

1359 \newcommand\BasicTree[5][1]{%
1360 \def\leaf@or@not@leaf{#5}%
1361 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
1362 \Basic@Tree[#1]{#2}{#3}{#4}{#5}\tikzsymbolsaftersymbolinput%
1363 \else%
1364 \ifx\@#5\%
1365 \Basic@Tree[#1]{#2}{#3}{#4}{#5}\tikzsymbolsaftersymbolinput%
1366 \else%
1367 \PackageError{tikzsymbols}{The last\MessageBreak parameter has either to be \MessageBreak
1368 ‘leaf’ or has to be empty}{See the tikzsymbols documentation. Section ‘‘Trees’’}%
1369 \fi\fi%
1370 }

```

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

```

1371 \DeclareRobustCommand{\WorstTree}[1][1]{%
1372 \set@tkzsymlbsscl{#1}%
1373 \if@tikzsymbols@absolute@draft%
1374 \tikzsymbols@resizebox{#1ex*165/100}{#1ex*165/100}{tree}%
1375 \else%
1376 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
1377 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.04ex*\tkzsymlbsscl,scale=#1]
1378 \fill[brown] (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1379 controls (0.5,0.7) and (0.5,0.3) .. (1,0);
1380 \draw (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1381 controls (0.5,0.7) and (0.5,0.3) .. (1,0) ;
1382 \fill[green] (0.2,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1383 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1384 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) ..
1385 controls (1.1, 1.6) and (1,1.4) .. (1,1.2) ..
1386 controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.5,0.8);
1387 \draw (0.214,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1388 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1389 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) .. controls (1.1, 1.6) and (1,1.4) ..
1390 (1,1.2) .. controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.486,0.8);
1391 \fill[red] (0,1) circle (0.1);
1392 \fill[red] (0.4,1.2) circle (0.1);
1393 \fill[red] (0.8,1.1) circle (0.1);
1394 \end{tikzpicture}%
1395 \fi%
1396 \tikzsymbolsaftersymbolinput%
1397 }

```

`\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`.

```

1398 \DeclareRobustCommand{\Springtree}[1][1=1, usedefault]%
1399 {\BasicTree[#1]{brown!70!black}{green!90!black}{green!80!black}{leaf}}

```

```

1400 \DeclareRobustCommand{\Summertree}[1][1=1, usedefault]%
1401   {\BasicTree[#1]{brown!50!black}{green!80!black}{red!80!green}{leaf}}
1402 \DeclareRobustCommand{\Autumntree}[1][1=1, usedefault]%
1403   {\BasicTree[#1]{red!30!black}{red!75!black}{orange}{leaf}}
1404 \DeclareRobustCommand{\Wintertree}[1][1=1, usedefault]%
1405   {\BasicTree[#1]{black!80!}{black!50}{black!25}{}}

1406 \AtBeginDocument{
1407   \if@tikzsymbols@marvosym
1408   \@ifpackageloaded{marvosym}{\%
1409     \PackageError{tikzsymbols}{Use option 'marvosym' only}{\MessageBreak
1410       if you load package 'marvosym'}
1411     \if load package 'marvosym' or \MessageBreak
1412     delete the tikzsymbols option 'marvosym'}}
1413 \fi
1414 }

```

Well that's it. Happy T_EXing!

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

Change History

v1.0		v1.65
General: Initial version	1	General: Improved BasicTree; New symbols “Schaler/peeler”, Laughy, Walley, Ninja; but didn't improve the source- description
v1.05		1
General: Deleted a “t” in the BasicTree-code, shortened the trunk from the tree a bit, re- named some codes, made an in- dex	1	v1.7
v1.6		General: New symbols, etc.
General: on/off.	1	1
Renamed “tikzsymbolsaftersym- bolinput” to “tikzsymbolsafter- symbolinput”	1	v2.0
Now “Person” can be used in sec- tions, etc.	1	General: Fixed Bugs, improved BasicTree, new option “mar- vosym”, new symbol
Now an error message is gener- ated if the last parameter of “BasicTree” is neither “leaf” nor empty.	1	1
v1.61		v2.2
General: Made an invisible box in BasicTree.	1	General: Now you can use negative scaling. Include ifpackageloaded. Did something else, I can't re- member
		1
		v2.5
		General: New option: draftabsolute, changed the documentation a bit
		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.

Symbols	<code>\Bed</code> 1236	528, 543, 562,
<code>\@BackblechlochX</code> 323, 325	<code>\Black@is@Black</code> . . .	578, 594, 609,
<code>\@BackblechlochY</code> 324, 326	. . . 1047, 1056,	625, 640, 656,
<code>\@Ofenschalter</code> 341, 342	1071, 1083, 1095	671, 689, 706,
<code>\@Tree@SetUp</code> <u>24</u> , 36, 115	<code>\Black@or@not@Black</code>	726, 745, 766,
<code>\@ifpackageloaded</code> 1048, 1056,	786, 807, 827,
. 4-9, 1408	1071, 1084, 1095	850, 872, 898,
<code>\@leaf@is@leaf</code>	<code>\Bratpfanne</code> . . . 177, 202	918, 946, 967,
<u>12</u> , 49, 98, 128, 1361		992, 1011, 1045,
<code>\@tikzsymbols@absolute@drafttrue</code>	C	1072, 1081, 1115,
. 73	<code>\c</code> 63, 142	1139, 1157, 1174,
<code>\@tikzsymbols@draftfalse</code>	<code>\Candle</code> 1153	1195, 1200, 1222,
. 74	<code>\Cat</code> 1007	1240, 1258, 1279,
<code>\@tikzsymbols@drafttrue</code>	<code>\Chair</code> 1218	1283, 1286, 1318,
. 70	<code>\Cofeecup</code> 1195	1363, 1366, 1375
<code>\@tikzsymbols@finalfalse</code>	<code>\Coffeecup</code> 1196	<code>\endcsname</code> 77
. 70	<code>\Cooley</code> 741	
<code>\@tikzsymbols@finaltrue</code>	<code>\Cooley_\dCooley</code> . . 741	F
. 74	<code>\coordinate</code> 61, 140	<code>\fi</code> 16, 34, 56, 59, 82,
<code>\@tikzsymbols@marvosymtrue</code>	<code>\csname</code> 77	87, 94, 108, 110,
. 23		113, 135, 138,
<code>\@tkzssmbles@negfalse</code>	D	155, 173, 182,
. . . . 68, 147, 1233	<code>\d</code> 63, 142	199, 208, 231,
<code>\@tkzssmbles@negtrue</code>	<code>\DeclareOptionX</code> . . .	240, 274, 283,
. . . . 34, 113, 1223	. 23, 70, 72, 74, 76	290, 299, 307,
<code>\@</code> 1364	<code>\def</code> 12,	316, 328, 337,
	14, 15, 24, 35, 71,	346, 355, 365,
A	72, 75, 77, 95, 96,	374, 384, 393,
<code>\a</code> 62, 141	114, 1047, 1048,	406, 415, 422,
<code>\Annoey</code> 475	1083, 1084, 1282,	431, 440, 449,
<code>\Annoey_\dAnnoey</code> . . 475	1285, 1287, 1360	457, 465, 472,
<code>\AtBeginDocument</code> . 1406	<code>\draft</code> 70	480, 488, 496,
<code>\Autumntree</code> 1398	<code>\Dreizack</code> 294, 310	503, 512, 520,
	<code>\drWalley</code> 988	523, 529, 536,
B		544, 555, 563,
<code>\b</code> 62, 141	E	571, 579, 587,
<code>\Backblech</code> 311, 331	<code>\else</code> . 53, 57, 93, 104,	595, 602, 610,
<code>\Basic@Tree</code> <u>33</u> , 71, 72,	132, 136, 154,	618, 626, 633,
75, 77, 1362, 1365	181, 207, 239,	641, 649, 657,
<code>\Basic@Tree@off</code> . . .	282, 298, 315,	664, 672, 682,
. 71, 72, <u>89</u>	336, 354, 373,	690, 699, 707,
<code>\Basic@Tree@on</code> . 75, <u>112</u>	392, 414, 430,	718, 727, 738,
<code>\BasicTree</code> <u>1359</u> , 1399,	448, 464, 479,	746, 759, 767,
1401, 1403, 1405	495, 506, 511,	779, 787, 800,

808, 820, 828,	<code>\if@tikzsymbols@draft</code>	<code>M</code>
843, 851, 865, 18, 78	<code>\marvosym</code> <u>23</u>
873, 891, 899,	<code>\if@tikzsymbols@final</code>	<code>\MessageBreak</code> .. 79,
911, 919, 939, 19, 83	81, 82, 84, 86,
947, 960, 968,	<code>\if@tikzsymbols@marvosym</code>	87, 1367, 1409, 1411
985, 993, 1004,	<u>18, 506, 1195, 1407</u>	<code>\Moai</code> <u>1275</u>
1012, 1038, 1046,	<code>\if@tkzssmbles@neg</code> ..	N
1057, 1072, 1074,	. <u>18, 50, 129, 1228</u>	<code>\NeedsTeXFormat</code>
1082, 1096, 1108,	<code>\ifdim</code> 16, 34,	<code>\Neutrey</code> 574
1116, 1132, 1140,	94, 113, 155, 182,	<code>\Neutrey_\dNeutrey</code> . <u>574</u>
1150, 1158, 1167,	208, 240, 283,	<code>\newcommand</code> ... 11, 1359
1175, 1192, 1201,	299, 316, 337,	<code>\newif</code> 18–22
1214, 1217, 1223,	355, 374, 393,	<code>\newlength</code> 13
1228, 1232, 1241,	415, 431, 449,	<code>\NiceReapey</code> <u>1111</u>
1250, 1259, 1272,	465, 480, 496,	<code>\Ninja</code> 1041
1280, 1288, 1311,	512, 529, 544,	<code>\Ninja_\dNinja</code> ... <u>1041</u>
1319, 1356, 1369,	563, 579, 595,	<code>\Nursey</code> 823
1376, 1395, 1413	610, 626, 641,	<code>\Nursey_\dNursey</code> .. <u>823</u>
<code>\filldraw</code> 318	657, 672, 690,	O
<code>\final</code> <u>74</u>	707, 727, 746,	<code>\Ofen</code> 332, 349
<code>\Fire</code> <u>1170</u>	767, 787, 808,	P
<code>\foreach</code>	828, 851, 873,	<code>\PackageError</code> 1367, 1409
323, 324, 341, 1349	899, 919, 947,	<code>\PackageWarningNoLine</code>
	968, 993, 1012, 79, 84
H	1046, 1082, 1116,	<code>\Person</code> <u>1135</u>
<code>\Herd</code> 369, 387	1140, 1158, 1175,	<code>\Pfanne</code> 350, 368
	1201, 1223, 1241,	<code>\pgfarrowsdeclare</code> 37, 116
I	1259, 1280, 1281,	<code>\pgfarrowslefttextend</code>
<code>\if@draft_\if@final</code> <u>18</u>	1284, 1319, 1376 38, 117
<code>\if@tikzsymbols@absolute@draft</code> 49, 98,	<code>\pgfarrowsrighttextend</code>
..... 22,	128, 1056, 1071, 38, 117
91, 152, 179, 205,	1095, 1361, 1364	<code>\pgflowlevel</code> ... 60, 139
237, 280, 296,	<code>\Innocey</code> 702, 721	<code>\pgfpatharc</code>
313, 334, 352,	<code>\Innocey_\dInnocey</code> . <u>702</u>	.. 41, 42, 120, 121
371, 390, 412,	K	<code>\pgfpathmoveto</code> . 40, 119
428, 446, 462,	<code>\Kochtopf</code> 150, 176	<code>\pgfpoin</code> 40, 119
477, 493, 509,		<code>\pgftransformscale</code> .
526, 541, 560,	L 60, 139
576, 592, 607,	<code>\Laughey</code> 539	<code>\pgfusepathqfill</code> 43, 122
623, 638, 654,	<code>\Laughey_\dLaughey</code> . <u>539</u>	<code>\ProcessOptionsX</code> .. 149
669, 687, 704,	<code>\leaf@or@not@leaf</code> ..	<code>\ProcessOptionsX*</code> .. <u>149</u>
724, 743, 764,	. 35, 49, 95, 98,	<code>\ProvidesPackage</code> ... 2
784, 805, 825,	114, 128, 1360, 1361	<code>\Purierstab</code> ... 278, 293
848, 870, 896,	<code>\leafcolor</code> 64, 143	
916, 944, 965,	<code>\let</code> ... 176, 202, 234,	R
990, 1009, 1043,	277, 293, 310,	<code>\relax</code> ... 149, 506, 1195
1079, 1113, 1137,	331, 349, 368,	<code>\RequirePackage</code> ... 4–9
1155, 1172, 1198,	387, 409, 425, 443	
1220, 1238, 1256,		
1277, 1316, 1373		

\resizebox 16	\setlength 14	1151, 1168, 1193,
\rWalley 942	\Sey 636	1215, 1234, 1251,
	\Sey_\dSey 636	1273, 1312, 1357,
	\Sieb 235, 277	1362, 1365, 1396
S		
\Sadey 444	\Smiley 507	\tikzsymbolsMoaithickness
\Sadey_\dSadey 444	\Smiley_\dSmiley . . 506 1282,
\Saftpresse . . . 388, 409	\Snowman 1314	1285, 1287, 1290
\Schaler 426, 443	\Springtree 1398	\tikzsymlsscl 13, 16,
\Schneebeesen . . 203, 234	\Summertree 1398	34, 94, 97, 113,
\Schussel 410, 425		155–157, 163,
\set@tkzsymlsscl 14,	T	
34, 90, 94, 113,	\tikzset 24	182–185, 187,
151, 155, 178,	\tikzsymbols@resizebox	189, 190, 192,
182, 204, 208, 15,	194, 196, 208,
236, 240, 279,	92, 153, 180, 206,	210, 212, 214,
283, 295, 299,	238, 281, 297,	216, 218, 220,
312, 316, 333,	314, 335, 353,	222, 224, 226,
337, 351, 355,	372, 391, 413,	228, 240–242,
370, 374, 389,	429, 447, 463,	283–286, 299–
393, 411, 415,	478, 494, 510,	301, 304, 316–
427, 431, 445,	527, 542, 561,	319, 321, 337,
449, 461, 465,	577, 593, 608,	338, 344, 355–
476, 480, 492,	624, 639, 655,	357, 361, 362,
496, 508, 512,	670, 688, 705,	374–376, 393–
525, 529, 540,	725, 744, 765,	395, 398–400,
544, 559, 563,	785, 806, 826,	402–404, 415–
575, 579, 591,	849, 871, 897,	417, 431–433,
595, 606, 610,	917, 945, 966,	437, 438, 449–
622, 626, 637,	991, 1010, 1044,	452, 465, 466,
641, 653, 657,	1080, 1114, 1138,	480–483, 496,
668, 672, 686,	1156, 1173, 1199,	497, 512, 513,
690, 703, 707,	1221, 1239, 1257,	529, 530, 544–
723, 727, 742,	1278, 1317, 1374	547, 550, 552,
746, 763, 767,	\tikzsymbolsaftersymbolinput	563–565, 568,
783, 787, 804,	. . 11, 174, 200,	579–582, 595,
808, 824, 828,	232, 275, 291,	596, 610, 611,
847, 851, 869,	308, 329, 347,	626, 627, 641–
873, 895, 899,	366, 385, 407,	644, 657, 658,
915, 919, 943,	423, 441, 458,	672–675, 690–
947, 964, 968,	473, 489, 504,	692, 707, 708,
989, 993, 1008,	521, 537, 556,	714–716, 727,
1012, 1042, 1046,	572, 588, 603,	728, 733, 735,
1078, 1082, 1112,	619, 634, 650,	736, 746, 747,
1116, 1136, 1140,	665, 683, 700,	750, 752, 767,
1154, 1158, 1171,	719, 739, 760,	768, 773, 775,
1175, 1197, 1201,	780, 801, 821,	787, 788, 793,
1219, 1223, 1237,	844, 866, 892,	795, 797, 808,
1241, 1255, 1259,	912, 940, 961,	809, 813, 815,
1276, 1280, 1315,	986, 1005, 1039,	817, 828–830,
1319, 1372, 1376	1075, 1109, 1133,	836, 838, 840,
		841, 851, 852,

856, 858, 860,	1116, 1117, 1124–	U	
862, 873–877,	1128, 1140, 1141,	\usetikzlibrary ...	10
899–901, 903,	1158–1162, 1164,	V	
919–923, 926,	1175, 1176, 1185,	\Vomey	868
929, 932, 947–	1187, 1201–1203,	\Vomey_\dVomey	<u>868</u>
951, 954–956,	1206, 1208, 1210,	W	
968–970, 972,	1223, 1224, 1226,	\Walley	914
975, 978, 981,	1241, 1242, 1259,	\Walley_\dWalley ..	<u>914</u>
993–995, 997–	1261, 1280, 1281,	\Winkey	605
1000, 1012, 1013,	1284, 1289, 1290,	\Winkey_\dWinkey ..	<u>605</u>
1015, 1016, 1019,	1296, 1298, 1300,	\wInnocey	721
1021, 1023, 1025,	1319, 1320, 1322–	\Wintertree	<u>1398</u>
1027, 1029, 1031,	1324, 1327, 1348,	\WorstTree	<u>1371</u>
1033, 1035, 1046,	1350, 1376, 1377	X	
1049–1051, 1053,	\Tongey	\x	1349, 1350
1054, 1057, 1065,	\Tongey_\dTongey ..	\Xey	667
1066, 1068, 1069,	\tree	\Xey_\dXey	<u>667</u>
1072, 1082, 1085–	\Tree@Off@line 96, 99–103		
1088, 1090, 1091,	\Tribar		
1093, 1096–1098,			