

tikzsymbols*

Ben Vitecek
b.vitecek@gmx.at

July 28, 2013

Abstract

Just some symbols created using 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/true/off/false, draft=true/false	2
2.2	draft=absolute	2
2.3	final=true/false	2
2.4	marvosym=true/false	3
2.5	usebox=true/false	3
2.6	prefix, prefix=<prefix>	3
3	Symbols	4
3.1	cooking-symbols 🍷	4
3.2	Emoticons ☺	5
3.2.1	“normal” Emoticons 😊	5
3.2.2	“3D” Emoticons 🍌 🍌	6
3.3	other Symbol(s) 🍌	7
3.4	Trees 🌳	8
3.5	Something to redefine	9
4	Warnings and Errors	9
4.1	Warnings	9
4.2	...and errors	10
5	Nobody is perfect	10

*This document corresponds to tikzsymbols v3.0, dated 2013/07/26.

6	Code	10
6.1	Cookingsymbolcode	17
6.2	Emoticonscode	24
6.3	Other symbols(s)	42
6.4	Trees	50

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-symbols¹? 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 them etc. And here they are.

2 Options

2.1 `tree=on/true/off/false`, `draft=true/false`





These options are only relevant for the commands in the section “Trees” (3.4). The trees look pretty nice (at least I think they do), but have one drawback: L^AT_EX needs extremely long to produce them. So these options come in handy: by setting `tree=off/false` or using `draft=true` or simply `draft` the trees will be replaced by squares drawn by tikz (for examples see section “Trees” 3.4).

One drawback is that these options only change the `\BasicTree` command, but not the others. Also another drawback is that tikz is still used to draw, this means L^AT_EX will be slowed down if you use many symbols.

To solve this problem I made another option: `draft=absolute`, which is – I think – more useful (see 2.2).

2.2 `draft=absolute`

Use this option if you use many symbols!

This option replaces *all* symbols by fast drawn plain vanilla rectangles, which have (mostly) the exactly same proportions as the tikz-symbols. For example, by setting `draft=absolute` `\Smiley` will produce , `\Nursey` , `\BasicTree{red}{red!50!black}{black}{leaf}` , `\Schussel` , etc.

You see, they are *very* plain and *very* vanilla (but L^AT_EX needs no time to produce them).²

2.3 `final=true/false`

This option is the opposite of `draft=true/false`.

¹Well, there are some, but not the one I wanted.

²The old option `draftabsolute` is still useable, but obsolete.

2.4 marvosym=true/false

You can use this special option if you also use package `marvosym`. If you want the `marvosym` Smiley (☺) and Coffeecup (☕) instead of the `tikzsymbols` ones (☺, ☕) you can use option `marvosym` resp. `marvosym=true`. If you use this option, `tikzsymbols` will simply not define its Smiley and Coffeecup.

Note: *Always load `tikzsymbols` after `marvosym`.*

Without option “ <code>marvosym</code> ”: ☺ ☕	With option “ <code>marvosym</code> ”: ☺ ☕
<code>\usepackage{marvosym}</code> <code>\usepackage{tikzsymbols}</code>	<code>\usepackage{marvosym}</code> <code>\usepackage{marvosym}{tikzsymbols}</code>

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

This option is *false* by default.

2.5 usebox=true/false

Since v3.0 the symbols are stored inside a `\savebox`³ and are used with `\usebox`. The advantage is that L^AT_EX doesn’t need to recalculate the symbol again (if you use the *exactly* same symbol, see section 3 for more information). This option is *true* by default.

You can deactivate this storing-inside-saveboxes by setting `usebox=false`.

To activate it you can use `usebox=true` or just `usebox`.

2.6 prefix, prefix=<prefix>

If you use a package which collides with `tikzsymbols`, but want to use symbols of both packages, you can use this option. It adds a `<prefix>` to all symbol commands provided by `tikzsymbols`. All commands will look like `\<prefix>command`, for example: `\<prefix>Smiley`, `\<prefix>drWalley`, `\<prefix>Springtree`, etc.

If you simply use option `prefix`, `<prefix>` will be “`tikzsymbols`”: `\Smiley` will change to `\tikzsymbolsSmiley`, `\drWalley` to `\tikzsymbolsdrWalley`, `\tikzsymbolsSpringtree`, `\tikzsymbolsBasicTree`, etc.

If this prefix is too long for you, you can define your own prefix via `prefix=<prefix>`. `<prefix>` should neither contain any special characters (e.g., ä, ü, ß, etc.) nor empty spaces. For example (using `prefix=T`): `\Smiley` changes to `\TSmiley`, `\Kochtopf` to `\TKochtopf` (and `\pot` to `\Tpot`), etc.

`\tikzsymbolsuse`







If you change the `prefix` often or are not sure if you will change it in future, you may work with `\tikzsymbolsuse{}` to use the symbols without worrying about the prefix. `\tikzsymbolsuse{}` takes one mandatory argument: the command-name of the symbol *without* backslash. Write the optional and mandatory parameters of the symbol after the curly braces.

For example: `\tikzsymbolsuse{Smiley}[2]` ☺

³Inside a `\sbox` to be correctly.

```
\tikzsymbolsuse{BasicTree}[1.2]{black}{red!50!black}{red}{leaf}🌿
\tikzsymbolsuse{Ofen}🔥\tikzsymbolsuse{Fire}[-1.3]🔥
etc.
```

3 Symbols

In this section the symbols are introduced. They  all  change 
automatically  with  the text-size .

Furthermore since v3.0 this package uses a savebox-usebox system. That means the output of a symbol is saved inside a box⁴ using `\sbox` and every time you use the *exactly* same symbol, L^AT_EX just can use the already calculated symbol (via `\usebox`).


What is “the *exactly* same symbol”? Using a symbol with the same optional parameter(s), the same script size and text-color.

For example: ☺ and ☹ and ☺ were only calculated once because they have the same optional parameter, script size and text-color.

☉, ☿ and ♀ would be calculated twice: the first time for the black one, the second time for the red one due to having a different text-color. The third Sadey is the same as the first, so it doesn't have to be recalculated.

\ominus , \ominus , \ominus and \oplus have all to be calculated separately: the first time for the normal, black one, the second time for the blue one (due to text color), the third time for having a different script size than the first one and the fourth time due to having a different script size and a different text-color.

Each symbol is stored in a separate box, but I think using `etex` with 32768 box registers is enough (and I don't think that you are using so many symbols; if you managed somehow to use so many boxes you can deactivate the storing of `tikzsymbols-symbols` inside boxes using `usebox=false` (see 2.5)).

There is a great advantage using this box-system: once calculated, the symbol can be used again without any new calculation⁵ (I know I am repeating myself). This means, you can use for example `\Summertree` many times without having to wait till L^AT_EX finished recalculating all of them (again, only if they are *exactly* the same): . With version <3.0 or option `usebox=false` this would take *extremely* long.

And again I will give you an advice: If you are using trees or many symbols, you should use option `draft=absolute`.

3.1 cooking-symbols

At the following table the cooking-symbols are listed.














⁴To say the truth: I am not sure what exactly happens while storing commands inside `\sbox` and using them via `\usebox...`

⁵And hopefully no disadvantage...

The first column shows the commands (at first the german at second the english ones). In the second the optional parameter(s) are shown. The optional parameter(s) are for both, the german and the english commands the same.

<scale> can be a number between (not exactly) -1400 and (also not exactly) 1400⁶, default is 1.

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

German & English Commands		Optional parameter(s)	Output
\Kochtopf	\pot	[<scale>]	
\Bratpfanne	\fryingpan	[<scale>]	
\Schneebeesen	\eggbeater	[<scale>]	
\Sieb	\sieve	[<scale>]	
\Purierstab	\blender ⁷	[<scale>]	
\Dreizack	\trident	[<scale>]	
\Backblech	\bakingplate	[<scale>]	
\Ofen	\oven	[<scale>]	
\Pfanne	\pan	[<scale>]	
\Herd	\cooker	[<scale>]	
\Saftpresse	\squeezer	[<scale>]	
\Schussel	\bowl	[<scale>]	
\Schaler	\peeler	[<scale>]	

3.2 Emoticons ☺

3.2.1 “normal” Emoticons 🐼

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

<scale> can be a number between (not exactly) -2000 and (not exactly) 2000⁸, default is 1.

<color> can be every defined color. Note: The color names shouldn’t contain special characters like ß, ä, ö, ...

Commands	Optional parameter(s)	Output
\Sadey	[<scale>] [<color>]	☹
\Smiley	[<scale>] [<color>]	☺
\Laughey	[<scale>] [<color>] [<mouth color>]	☺
\Annoey	[<scale>] [<color>]	☹
\Neutrey	[<scale>] [<color>]	☹
\Winkey	[<scale>] [<color>]	☺

⁶Since version 2.2 you can use negative numbers as well (see examples)

⁷I know that “Pürrierstab” should be translated as “immersion blender”, but I’m just using “blender”

⁸Do you even need so large symbols?

Commands	Optional parameter(s)	Output
<code>\oldWinkey</code>	<code>[\scale] [\color]</code>	😏
<code>\Sey</code>	<code>[\scale] [\color]</code>	😞
<code>\Xey</code>	<code>[\scale] [\color]</code>	😏
<code>\Innocey</code>	<code>[\scale] [\color] [\halo color]</code>	😇
<code>\wInnocey</code>	<code>[\scale]</code>	😇
<code>\Cooley</code>	<code>[\scale] [\color]</code>	😏
<code>\Tongey</code>	<code>[\scale] [\color] [\tongue color]</code>	😏
<code>\Nursey</code>	<code>[\scale] [\color] [\cap color] [\cross color]</code>	👩
<code>\Vomey</code>	<code>[\scale] [\color] [\vomit color]</code>	🤮
<code>\Walley</code>	<code>[\scale] [\color] [\wall color]</code>	🏠
<code>\rWalley⁹</code>	<code>[\scale] [\color] [\wall color]</code>	🏠
<code>\Cat</code>	<code>[\scale]</code>	🐱
<code>\Ninja</code>	<code>[\scale] [\color] [\headband color] [\eye color]</code>	🥷
<code>\NiceReapey</code>	<code>[\scale]</code>	👻

Examples: `\Sadey [] [red]` 🚫 `\Cooley [-3] [cyan]` 🤖
`\Vomey [1.5] [green!80!black] [olive]` 🤮
`\Nursey [] [yellow] [blue] [red]` 👩
`\Ninja [1.3] [] [violet] [red]` 🥷
`\colorbox{yellow}{\Winkey \Annoey [-1] \Neutrey}` 😏😏😏
`{\color{blue}\Sey}` 😞

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 number¹⁰ and a large number¹¹, default is 1.











`<color>` can be every defined color (see examples below). Note: The color names shouldn’t contain special characters like ß, ä, ö, ...






Commands	Optional parameter(s)	Output
<code>\dSadey</code>	<code>[\scale] [\color]</code>	🤖
<code>\dSmiley</code>	<code>[\scale] [\color]</code>	😄
<code>\dLaughey</code>	<code>[\scale] [\color] [\mouth color]</code>	😂
<code>\dAnnoey</code>	<code>[\scale] [\color]</code>	😏
<code>\dNeutrey</code>	<code>[\scale] [\color]</code>	😏
<code>\dWinkey</code>	<code>[\scale] [\color]</code>	😏
<code>\olddWinkey</code>	<code>[\scale] [\color]</code>	😏

⁹“r” stands for “random” and means that the cracks in the wall are generated randomly, but it takes some time to generate it.

¹⁰under -500 for sure

¹¹over 500 for sure

Commands	Optional parameter(s)	Output
<code>\dSey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>	
<code>\dXey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>	
<code>\dInnocey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle halo color \rangle]</code>	
<code>\dCooley</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>	
<code>\dTongey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle tongue color \rangle]</code>	
<code>\dNursey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle cap color \rangle] [\langle cross color \rangle]</code>	
<code>\dVomey</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle vomit color \rangle]</code>	
<code>\dWalley</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle wall color \rangle]</code>	
<code>\drWalley</code> ¹²	<code>[\langle scale \rangle] [\langle color \rangle] [\langle wall color \rangle]</code>	
<code>\dNinja</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle headband color \rangle] [\langle eye color \rangle]</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)

`\Strichmaxerl`'s optional parameters 2–5 (`<left arm>` to `<right leg>`) can be a number between -360 and 360¹³. These parameters are the angles between the body and the separate parts of `\Strichmaxerl` (see examples).

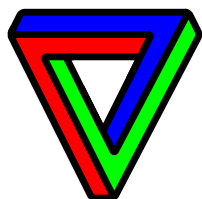
`<scale>` can be a very great and a very small number (but I don't think, that you need so large symbols).

`<color>` can be every defined color. Note: The color names shouldn't contain special characters like ß, ä, ö,

Commands	Optional parameter(s)	Output
<code>\Strichmaxerl</code>	<code>[\langle scale \rangle] [\langle left arm \rangle] [\langle right arm \rangle] [\langle left leg \rangle] [\langle right leg \rangle]</code>	
<code>\Candle</code>	<code>[\langle scale \rangle]</code>	
<code>\Fire</code>	<code>[\langle scale \rangle]</code>	
<code>\Coffeecup</code>	<code>[\langle scale \rangle]</code>	
<code>\Chair</code>	<code>[\langle scale \rangle]</code>	
<code>\Bed</code>	<code>[\langle scale \rangle]</code>	
<code>\Moai</code>	<code>[\langle scale \rangle]</code>	
<code>\Tribar</code>	<code>[\langle scale \rangle] [\langle color 1 \rangle] [\langle color 2 \rangle] [\langle color 3 \rangle]</code>	
<code>\Snowman</code>	<code>[\langle scale \rangle]</code>	

¹²“r” stands for “random” and means that the cracks in the wall are generated randomly, but it takes some time.

¹³Of course the number can be even greater or less, but it doesn't make sense.



`\Tribar[-10][blue][red][green]`



`\Tribar[2.1][blue][blue!50][blue!20]`

`\Strichmaxerl[1][10][30][40][4]%, \Strichmaxerl[1.4][210][310][10][90]%,`
`\Strichmaxerl[2][510][110][190][990]%, \Strichmaxerl[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 ¹⁴, default is 1.


`<color>` can be every defined color (see examples below). Note: The color names shouldn’t contain special characters like ß, ä, ö,


`{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 `leaf`, see examples below).


If you are using those trees, L^AT_EX needs longer to produce the output. So you may use the package option `tree=off`, `draft` or (better) `draft=absolute` (see section 2) to make L^AT_EX faster.


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


`\BasicTree` examples (normal trees):

`\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!20!black}{green!50!black}{green!70!black}{leaf}` 

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

¹⁴if it is larger (or less) it uses too much of L^AT_EX memory and an error message appears.

...and using the same trees with `tree=off/false` or `draft(=true)`:

```
\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!20!black}{green!50!black}{green!70!black}{leaf}
```



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



I think it's better if you define your own tree using `\newcommand` and `\BasicTree`:

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

3.5 Something to redefine

At the end of each symbol `\tikzsymbolsaftersymbolinput` is inserted. By default it is defined to contain `\xspace`:

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

You can redefine this macro. If you don't want `\xspace` just say:

```
\renewcommand{\tikzsymbolsaftersymbolinput}{}
```

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}}
```

4.2 ...and errors

Make sure you load e.g `marvosym` *after* `tikzsymbols` 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` or `prefix`).

all right

5 Nobody is perfect

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

6 Code (do you really need this section?)

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

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

```
1 \NeedsTeXFormat{LaTeX2e}[2011/06/27]
2 \ProvidesPackage{tikzsymbols}
3 [2013/07/26 v3.0 Some symbols created using tikz.]
4 \@ifpackageloaded{tikz}{}{\RequirePackage{tikz}}
5 \@ifpackageloaded{xargs}{}{\RequirePackage{xargs}}
6 \@ifpackageloaded{xcolor}{}{\RequirePackage{xcolor}}
7 \@ifpackageloaded{xkeyval}{}{\RequirePackage{xkeyval}}
8 \@ifpackageloaded{xspace}{}{\RequirePackage{xspace}}
9 \@ifpackageloaded{calc}{}{\RequirePackage{calc}}
```

Furthermore we need to load some libraries from `tikz`:

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

```
\if@tkzssmbles@neg We need \if@tkzssmbles@neg (=negative), well ...if something is negative
\iftikzsymbols@draftabsolute (\chair needs this).
```

`\iftikzsymbols@draftabsolute` is needed for option `draft=absolute`.

```
11 \newif\if@tkzssmbles@neg
12 \newif\iftikzsymbols@draftabsolute
```

```
\Basic@Tree Now we define our \Basic@Tree. We will need it later for our package option.
Furthermore if no option is given this will be the default definition of \Basic@Tree
inside the document.
```

```
13 \newcommand{\Basic@Tree}{\Basic@Tree@on}
```

final If final is false, `\Basic@Tree` will show squares drawn by tikz. If it is true, it will show trees.

```

14 \define@boolkey{tikzsymbols}{final}[true]{%
15   \ifKV@tikzsymbols@final
16     \def\Basic@Tree{\Basic@Tree@on}% final=true => final
17   \else
18     \def\Basic@Tree{\Basic@Tree@off}% = false => draft
19   \fi
20 }
```

draft If option draft or draft=true is set, then squares drawn by tikz are typed instead of trees.

If option draft=absolute is set, we set `\iftikzsymbols@draftabsolute` to true and all symbols are replaced by plain vanilla rectangles drawn by L^AT_EX.

```

21 \define@choicekey*{tikzsymbols}{draft}%
22 [\tikzsymbols@draft@val\tikzsymbols@draft@nr]{false,true,absolute}[true]{%
23   \ifcase\tikzsymbols@draft@nr\relax
24     \def\Basic@Tree{\Basic@Tree@on}% draft=false => final
25   \or
26     \def\Basic@Tree{\Basic@Tree@off}% = true => draft
27   \or
28     \def\Basic@Tree{\Basic@Tree@off}\tikzsymbols@draftabsolutetrue% =absolute
29   \fi
30 }
```

draftabsolute Obsolete Option draftabsolute. You can still use it, but it gives a warning.

```

31 \define@key{tikzsymbols}{draftabsolute}{%
32 \typeout{-----}%
33 \PackageWarningNoLine{tikzsymbols}{Option '\CurrentOption' is obsolete!}
34 \MessageBreak Please use 'draft=absolute' instead!}
35 \typeout{-----}%
36 \tikzsymbols@draftabsolutetrue}
```

marvosym Now I define the boolean option marvosym: you may use it, if you load the package marvosym.

```

37 \define@boolkey{tikzsymbols}{marvosym}[true]{}
```

usebox If it is true, the savebox-usebox-system is used. If it is false, then the system is turned off.

Furthermore it is true by default (`\KV@tikzsymbols@useboxtrue`)

```

38 \define@boolkey{tikzsymbols}{usebox}[true]{}
39 \KV@tikzsymbols@useboxtrue
```

prefix Option prefix changes all commands to `\<prefix>command`. If only prefix is set, `<prefix>` will be “tikzsymbols”, but you can define your own prefix via `prefix=<prefix>`

If this option is not used inside a document, `\cmdKV@tikzsymbols@prefix` wouldn't be defined. So we define it and let it empty.

```

40 \newcommand{\cmdKV@tikzsymbols@prefix}{}
41 \define@cmdkey{tikzsymbols}{prefix}{tikzsymbols}{}

tree Now we declare the name of our option: tree (I could have named it stone, or
wood, etc., but I used “tree”). I am using xkeyval now more than before and tree
can now be set to on/true resp. off/false.
42 \define@choicekey{tikzsymbols}{tree}%
43 [\tikzsymbols@tree@val\tikzsymbols@tree@nr]{true,on,false,off}[on]{%
44   \ifcase\tikzsymbols@tree@nr\relax
45     \def\Basic@Tree{\Basic@Tree@on}% tree=true
46   \or
47     \def\Basic@Tree{\Basic@Tree@on}% tree=on
48   \or
49     \def\Basic@Tree{\Basic@Tree@off}% tree=false
50   \or
51     \def\Basic@Tree{\Basic@Tree@off}% tree=false
52   \fi
53 }

\ProcessOptionsX* We process all options. What is \relax doing?
54 \ProcessOptionsX*<tikzsymbols>\relax

\tikzsymbolsaftersymbolinput Now we define this strange named macro. This macro is inserted after the tikz-code,
and is defined as \xspace. If you don’t want \xspace to be inserted, you can
redefine this command.
55 \newcommand*{\tikzsymbolsaftersymbolinput}{\xspace}

\@leaf@is@leaf We need this command for creating an error-message if the last parameter of
BasicTree is neither “leaf” nor empty.
56 \newcommand*{\@leaf@is@leaf}{leaf}

\tkzsymlsscl The [scale] parameter of the commands is stored inside \tkzsymlsscl.
\set@tkzsymlsscl is a short version of \setlength{\tkzsymlsscl}{#1pt}.
I defined it to write less.
57 \newlength{\tkzsymlsscl}
58 \newcommand*{\set@tkzsymlsscl}[1]{\setlength{\tkzsymlsscl}{#1pt}}

\tikzsymbols@draftboxlength The length and the height of the plain vanilla rectangle are stored inside these
\tikzsymbols@draftboxheight lengths.
59 \newlength{\tikzsymbols@draftboxlength}
60 \newlength{\tikzsymbols@draftboxheight}

\tikzsymbols@draftbox Our plain vanilla rectangle. Using \tikzsymbols@draftboxlength and \tikzsymbols@draftboxheight
to calculate the length and the height of the box.
The box is drawn using \frame and a \vbox, \hbox construction. \vbadness=\maxdimen
is needed because otherwise there would be many overfull v-box errors.
61 \newcommand*{\tikzsymbols@draftbox}[2]{%
62   \setlength{\tikzsymbols@draftboxlength}{#1}%

```

```

63 \setlength{\tikzsymbols@draftboxheight}{#2}%
64 \frame{%
65   {\vbadness=\maxdimen%
66     \frame{\vbox to \tikzsymbols@draftboxheight{%
67       \hbox to \tikzsymbols@draftboxlength{}}}%
68   }%
69 }

\tikzsymbols@draftQbox Short form of \tikzsymbols@draftbox. “Q” means “Quadrat”(square) and just
means that height and length of the box are the same.
70 \newcommand*\tikzsymbols@draftQbox[1]{\tikzsymbols@draftbox{#1}{#1}}

\tikzsymlsPrmtr \tikzsymbols@draftbox doesn't like negative numbers. So to be sure that only
\tksymlsboxPrmtrstore positive numbers are used, \tksymlsboxPrmtrstore changes \tikzsymlsPrmtr
to be positive (if it is negative). I am using \tkzsymlsPrmtr instead of #1 (scaling)
in \tikzsymbols@draftbox
71 \newcommand*\tkzsymlsPrmtr{}
72 \newcommand*\tksymlsboxPrmtrstore[1]{%
73   \edef\tkzsymlsPrmtr{\ifdim\tkzsymlsscl<0pt-\fi#1}%
74 }

\current@tikzsymbols The \sbox and \usebox system I am using now has one drawback: If the symbol
was used inside e.g. \color{red} all other symbols of the same type are red, even
if they are in normal text, or even inside another color.
The same with e.g. \small. If a symbol was used the first time inside e.g.
\small all other symbols of the same type are as small as the first one.
To overcome this problem, the save-boxes name depends of the current color
and the current script size. These things are stored inside \current@tikzsymbols.
75 \newcommand*\current@tikzsymbols{}

\tikzsymbols@ifsaveboxundefined For every symbol we define a box (using \sbox). If a symbol of the same type is
used again, it doesn't have to be recalculated. We simply use the same output as
the first symbol. If the symbol has another input, a new save-box is defined.
For example: \Summertree[1]\Summertree[1]\Summertree[1] is only calcu-
lated once because they are alike: 🌳🌳🌳. \Summertree[1.3] will be saved in an
own save-box because it is different then the others.
76 \newcommand*\tikzsymbols@ifsaveboxundefined[2]{%
77 \edef\current@tikzsymbols{\current@color\fi@size}%
78 \expandafter\ifcsname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname%
79 \relax%
80 \else%
81 \expandafter\expandafter\expandafter\newsavebox%
82 \expandafter\expandafter\expandafter{%
83   \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}%
84 \expandafter\expandafter\expandafter\global%
85 \expandafter\expandafter\expandafter\sbox%
86 \expandafter\expandafter\expandafter{%
87   \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}{#2}%

```

```

88 \fi%
89 }

\tikzsymbolsusebox tikzsymbols version of \usebox.
90 \newcommand*\tikzsymbolsusebox[1]{%
91 \expandafter\expandafter\expandafter\usebox%
92 \expandafter\expandafter\expandafter{%
93 \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}%
94 }

\ifKV@tikzsymbols@usebox = true If option usebox is false, then \tikzsymbols@ifsaveboxundefined and \tikzsymbolsusebox
will be redefined so that they don't store the input inside a save-box.
95 \ifKV@tikzsymbols@usebox
96 \relax% if true
97 \else
98 \renewcommand{\tikzsymbols@ifsaveboxundefined}[2]{\@secondoftwo{#1}{#2}}
99 \renewcommand{\tikzsymbolsusebox}[1]{\@gobble{#1}}
100 \fi

\tkzsymlsDeclareRobustCommand To make use of the prefix option (and to write less), I am using these commands.
If <prefix> is empty, the commands will be "normal": \Smiley will be \Smiley,
etc.
    If <prefix> is not empty the commands will be defined as \<prefix>command
    e.g. \<prefix>Smiley , etc.
    Furthermore a new command is defined. tikzsymbolsuse needs this command
    to specify wherever the input is a symbol of tikzsymbols or not.
101 \newcommand{\tkzsymlsDeclareRobustCommand}[1]{%
102 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
103 \expandafter\DeclareRobustCommand%
104 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
105 }

\tkzsymlsDeclareRobustCommandx Same as before
106 \newcommand{\tkzsymlsDeclareRobustCommandx}[1]{%
107 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
108 \expandafter\DeclareRobustCommandx%
109 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
110 }

\tkzsymlsnewcommand Same as before
111 \newcommand{\tkzsymlsnewcommand}[1]{%
112 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
113 \expandafter\newcommand%
114 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
115 }

\tikzsymbolsuse To be able to don't have to care about the prefix, I made this command. Simply
write the name of the symbols inside without backslash. If the symbol is not
defined, there will be an error message.

```

```

116 \newcommand{\tikzsymbolsuse}[1]{%
117 \ifcsname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname%
118 \relax\else\PackageError{tikzsymbols}{\MessageBreak%
119 Undefined Control sequence: '#1'}{Did you write the name correctly?}\fi%
120 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
121 }

\tikzsymbols@let \eggbeater is the same as \Schneebesen (\let\eggbeater\Schneebesen). To
make the prefix also for the \let commands, I defined my own let-command
122 \newcommand{\tikzsymbols@let}[2]{%
123 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
124 \expandafter\expandafter\expandafter\let%
125 \expandafter\csname\expandafter\cmdKV@tikzsymbols@prefix%
126 \expandafter#1\expandafter\endcsname%
127 \csname\cmdKV@tikzsymbols@prefix#2\endcsname%
128 }

\@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):
129 \def\@Tree@SetUp{\tikzset{%
130 ld/.style={level distance=##1ex},lw/.style={line width=##1ex},%
131 level 1/.style={ld=0.60, trunk, lw=0.1 ,sibling angle=60},%
132 level 2/.style={ld=0.20, trunk!80!leaf a, lw=.073,sibling angle=70},%
133 level 3/.style={ld=0.25, trunk!60!leaf a, lw=.05,sibling angle=70}, %
134 level 4/.style={ld=0.10, trunk!40!leaf a, lw=.025,sibling angle=60},%
135 level 5/.style={ld=0.15, trunk!20!leaf a, lw=.02,sibling angle=60},%
136 level 6/.style={ld=0.08, leaf a, lw=.021,sibling angle=60},%
137 }}%

\Basic@Tree \Basic@Tree@off and \Basic@Tree@on are necessary for option tree, draft and
final.

\Basic@Tree@off \Basic@Tree@off is used when trees are turned off.
138 \DeclareRobustCommand{\Basic@Tree@off}[5][1=1, usedefault]{%
139 \set@tkzsymlsscl{#1}%
140 \pgfmathsetmacro\tikzsymbols@Tree@absolute@scale{#1+0.02ex}%
141 \edef\tkzsymlsPrmtr{\ifdim\tkzsymlsscl<0pt-\fi\tikzsymbols@Tree@absolute@scale}%
142 \ifdim\tkzsymlsscl<0pt \set@tkzsymlsscl{-#1}\fi%
143 \def\leaf@or@not@leaf{#5}%
144 \iftikzsymbols@draftabsolute%
145 \ifx\leaf@or@not@leaf@leaf@is@leaf%
146 \tikzsymbols@draftbox{(1.6772ex+0.4pt)*\real{\tkzsymlsPrmtr}}
147 {(1.42ex-0.2pt+0.4pt)*\real{\tkzsymlsPrmtr}}%
148 \else
149 \tikzsymbols@draftbox{(1.3996ex+0.4pt)*\real{\tkzsymlsPrmtr}}
150 {(1.28ex-0.2pt+0.4pt)*\real{\tkzsymlsPrmtr}}%
151 \fi
152 \else%
153 \begin{tikzpicture}[scale=#1+0.02ex,x=1ex,y=1ex, line width=0.4pt*\tkzsymlsscl]

```

```

154 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
155 \draw[#2] (-0.8386,0+0.2pt) -- (-0.8386,1.42);
156 \draw[#3] (-0.8386,1.42) -- (0.8386,1.42);
157 \draw[#4] (0.8386,1.42) -- (0.8386,0+0.2pt);
158 \draw[#3] (0.8386,0+0.2pt) -- (0,0+0.2pt);
159 \draw[#4] (0,0+0.2pt) -- (-0.8386,0+0.2pt);
160 \else
161 \draw[#2] (-0.6998,0+0.2pt) -- (-0.6998,0.68+0.6);
162 \draw[#3] (-0.6998,0.68+0.6) -- (0.6998,0.68+0.6);
163 \draw[#4] (0.6998,0.68+0.6) -- (0.6998,0+0.2pt);
164 \fi%
165 \end{tikzpicture}%
166 \fi%
167 }}

```

`\Basic@Tree@on` `\Basic@Tree@on` is used when trees are turned on.

```

168 \DeclareRobustCommand{\Basic@Tree@on}[5][1=1, usedefault]{%
169 \iftikzsymbols@draftabsolute%
170 \Basic@Tree@off[#1]{#2}{#3}{#4}{#5}%
171 \else%
172 \set@tkzsymlsscl{#1}%
173 \ifdim\tkzsymlsscl<0pt \set@tkzsymlsscl{-#1}\@tkzssmbles@negtrue\fi%
174 \def\leaf@or@not@leaf{#5}%
175 \@Tree@SetUp%
176 \pgfarrowsdeclare{leaf}{leaf}%
177 {\pgfarrowsleftextend{-.1ex}\pgfarrowsrightextend{-0.05ex}}%
178 {%
179 \pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
180 \pgfpatharc{150}{30}{0.08ex}%
181 \pgfpatharc{-30}{-150}{0.08ex}%
182 \pgfusepathqfill%
183 }%
184 \colorlet{trunk}{#2}%
185 \colorlet{leaf a}{#3}%
186 \colorlet{leaf b}{#4}%
187 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
188 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
189 \draw[opacity=0,scale=#1+0.02ex, line width=0.4pt*\tkzsymlsscl]
190 (-0.8386,0+0.2pt) rectangle
191 (0.8386, 1.42);
192 \else %
193 \draw[opacity=0,scale=#1+0.02ex, line width=0.4pt*\tkzsymlsscl]
194 (-0.6998,0+0.2pt) rectangle (0.6998,0.68+0.6);
195 \fi
196 \pgflowlevel{\pgftransformscale{#1+0.02ex}}{%
197 \coordinate (root) [grow cyclic,rotate=90] child {
198 child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
199 child foreach \c in {0,1,2} { child foreach \d in {0,1} {
200 child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5]}
201 }}} edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]

```



```

202   };}%
203   \end{tikzpicture}%
204   \@tkzssmbles@negfalse%
205   \fi%
206 }%

```

6.1 Cookingsymbolcode

`\Kochtopf = \pot` I am using `\DefineRobustCommand` so that the symbols can be used inside `\section{}`, `\footnote`, `\index{}`, etc. It may would have worked with `\newcommand` too.

You can either use the german commands or the english ones:

```

207 \tkzsymbolsDeclareRobustCommand{Kochtopf}[1][1]{%
208 \tikzsymbols@ifsaveboxundefined{Kochtopf#1}{%
209 \set@tkzsymbolsscl{#1}%
210 \iftikzymbols@draftabsolute%
211 \tkzsymbols@draftbox{2.47ex*\real{\tkzsymbolsPrmtr}}{1.577ex*\real{\tkzsymbolsPrmtr}}%
212 \else%
213 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
214 \begin{tikzpicture}[x=2ex,y=2.2ex, line width=0.07ex*\tkzsymbolsscl,scale=#1]
215 \draw[rounded corners=0.2ex*\tkzsymbolsscl] (0,0.5) -- (0,0) -- (1,0) -- (1,0.5);
216 \draw(0,0.4) arc (90:270:0.1);
217 \draw(1,0.4) arc (90:-90:0.1);
218 \draw (0,0.5) -- (1,0.5) .. controls (1,0.6) and (0,0.6) .. (0,0.5);
219 \draw (0.6,0.585) arc (0:180:0.1);
220 \draw[decorate,
221 decoration={snake,amplitude=.12ex*\tkzsymbolsscl,segment length=0.93ex*\tkzsymbolsscl}]
222 (0,0.35) -- (1,0.35);
223 \draw (0.1,0.25) circle (0.04);
224 \draw (0.3,0.2) circle (0.04);
225 \draw (0.13, 0.125) circle (0.04);
226 \draw (0.6,0.25) circle (0.04);
227 \draw (0.45,0.1) circle (0.04);
228 \draw (0.88,0.2) circle (0.04);
229 \draw (0.7,0.11) circle (0.04);
230 \end{tikzpicture}%
231 \fi%
232 }%
233 }%
234 \tikzsymbolsusebox{Kochtopf#1}%
235 \tikzsymbolsaftersymbolinput%
236 }
237 \tikzsymbols@let{\pot}{Kochtopf}

```

`\Bratpfanne = \fryingpan` If you wonder why I am using `line width=0.07ex*\tkzsymbolsscl` instead of `line width=0.07ex*#1` I will try to explain it.

After being multiplied by a negative number, the line widths would be too thin for the size of the symbol. So it is necessary that the line width is always scaled

with a positive number. Thus I am using `\tkzsymlblsscl` because it is always positive.

```

238 \tkzsymlblsDeclareRobustCommand{Bratpfanne}[1][1]{%
239 \tikzsymbols@ifsaveboxundefined{Bratpfanne#1}{%
240 \set@tkzsymlblsscl{#1}%
241 \iftikzymbols@draftabsolute%
242 \tkzsymlblsbxPrmtrstore{#1}%
243 \tikzsymbols@draftbox{3.5535ex*\real{\tkzsymlblsPrmtr}}{1.4525ex*\real{\tkzsymlblsPrmtr}}}%
244 \else%
245 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
246 \begin{tikzpicture}[x=0.7ex,y=1.4ex, line width=0.07ex*\tkzsymlblsscl, scale=#1,
247 decoration={snake,amplitude=.05ex*\tkzsymlblsscl,segment length=0.408ex*\tkzsymlblsscl}]
248 \draw[rounded corners=0.07ex*\tkzsymlblsscl]
249 (-1,0) -- (1,0) -- (1.5,0.4) -- (-1.5,0.4) -- cycle;
250 \draw[line width=0.037ex*\tkzsymlblsscl, rounded corners=0.023ex*\tkzsymlblsscl]
251 (-1.4,0.3) -- (-3.5,0.3) -- (-3.5,0.25) -- (-1.3,0.25);
252 \draw[line width=0.023ex*\tkzsymlblsscl] (-1.1,0.1) -- (1.1,0.1);
253 \draw[line width=0.035ex*\tkzsymlblsscl, decorate]
254 (-0.3,0.5) -- (-0.3,1);
255 \draw[line width=0.035ex*\tkzsymlblsscl, decorate]
256 (0.3,0.5) -- (0.3,1);
257 \draw[line width=0.035ex*\tkzsymlblsscl, decorate]
258 (-1,0.5) -- (-1,1);
259 \draw[line width=0.035ex*\tkzsymlblsscl, decorate]
260 (1,0.5) -- (1,1);
261 \end{tikzpicture}%
262 \fi%
263 }%
264 \tikzsymbolsusebox{Bratpfanne#1}%
265 \tikzsymbolsaftersymbolinput%
266 }
267 \tikzsymbols@let{fryingpan}{Bratpfanne}

```

`\Schneebesen = \eggbeater` The next one:

```

268 \tkzsymlblsDeclareRobustCommand{Schneebesen}[1][1]{%
269 \tikzsymbols@ifsaveboxundefined{Schneebesen#1}{%
270 \set@tkzsymlblsscl{#1}%
271 \iftikzymbols@draftabsolute%
272 \tkzsymlblsbxPrmtrstore{#1}%
273 \tikzsymbols@draftbox{0.5697ex*\real{\tkzsymlblsPrmtr}}{1.57985ex*\real{\tkzsymlblsPrmtr}}}%
274 \else%
275 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
276 \begin{tikzpicture}[y=2.1ex,x=1.4ex, scale=#1,
277 line width=0.01ex*\tkzsymlblsscl*\real{0.97}]
278 \draw (0,0) .. controls (0.2,0.0) and (0.2,0.2) .. (0,0.4);
279 \draw (0,0) .. controls (-0.2,0.0) and (-0.2,0.2) .. (0,0.4);
280 \draw (0,0) .. controls (-0.1,0.0) and (-0.1,0.2) .. (0,0.4);
281 \draw (0,0) .. controls (0.1,0.0) and (0.1,0.2) .. (0,0.4);
282 \draw (0,0) .. controls (-0.15,0.0) and (-0.15,0.2) .. (0,0.4);

```

```

283 \draw (0,0) .. controls (0.15,0.0) and (0.15,0.2) .. (0,0.4);
284 \draw (0,0) .. controls (-0.05,0.0) and (-0.05,0.2) .. (0,0.4);
285 \draw (0,0) .. controls (0.05,0.0) and (0.05,0.2) .. (0,0.4);
286 \draw (0,0) --(0,0.4);
287 \fill[line width=0.05ex*\tkzsymlblsscl, rounded corners=0.07ex*\tkzsymlblsscl]
288 (-0.05,0.37) -- (0.05,0.37) -- (0.05,0.75) -- (-0.05,0.75) -- cycle;
289 \end{tikzpicture}%
290 \fi%
291 }%
292 \tikzsymbolsusebox{Schneebesen#1}%
293 \tikzsymbolsaftersymbolinput%
294 }
295 \tikzsymbols@let{eggbeater}{Schneebesen}

\Sieb = \sieve Now a long one;

296 \tikzsymbolsDeclareRobustCommand{Sieb}[1][1]{%
297 \tikzsymbols@ifsaveboxundefined{Sieb#1}{%
298 \set@tkzsymlblsscl{#1}%
299 \iftikzymbols@draftabsolute%
300 \tkzsymbols@draftbox{3.478ex*\real{\tkzsymbolsPrmtr}}{1.175ex*\real{\tkzsymbolsPrmtr}}%
301 \else%
302 \ifdim\tkzsymlblsscl<0pt\set@tkzsymlblsscl{-#1}\fi%
303 \begin{tikzpicture}[x=2.8ex, y=2.8ex, line width=0.02ex*\tkzsymlblsscl, scale=#1]
304 \draw[line width=0.09ex*\tkzsymlblsscl] (-0.2,0) -- (1.01,0);
305 \draw (0.2,0) arc (180:360:0.4);
306 \draw (0.25,0) arc (180:360:0.35);
307 \draw (0.3,0) arc (180:360:0.3);
308 \draw (0.35,0) arc (180:360:0.25);
309 \draw (0.4,0) arc (180:360:0.2);
310 \draw (0.45,0) arc (180:360:0.15);
311 \draw (0.5,0) arc (180:360:0.1);
312 \draw (0.55,0) arc (180:360:0.05);
313 \draw (.95,0) -- (0.95,-0.194);
314 \draw (.9,0) -- (0.9,-0.265);
315 \draw (.85,0) -- (0.85,-0.313);
316 \draw (.8,0) -- (0.8,-0.345);
317 \draw (.75,0) -- (0.75,-0.37);
318 \draw (.7,0) -- (0.7,-0.39);
319 \draw (.65,0) -- (0.65,-0.4);
320 \draw (.6,0) -- (0.6,-0.4);
321 \draw (.55,0) -- (0.55,-0.4);
322 \draw (.5,0) -- (0.5,-0.39);
323 \draw (.45,0) -- (0.45,-0.37);
324 \draw (.4,0) -- (0.4,-0.348);
325 \draw (.35,0) -- (0.35,-0.314);
326 \draw (.3,0) -- (0.3,-0.265);
327 \draw (.25,0) -- (0.25,-0.194);
328 \draw (0.2,-0.05) -- (1,-0.05);
329 \draw (0.21,-0.1) -- (0.99,-0.1);
330 \draw (0.21,-0.1) -- (0.99,-0.1);

```

```

331 \draw (0.23,-0.15) -- (0.97,-0.15);
332 \draw (0.255,-0.2) -- (0.945,-0.2);
333 \draw (0.289,-0.25) -- (0.911,-0.25);
334 \draw (0.335,-0.3) -- (0.865,-0.3);
335 \draw (0.406,-0.35) -- (0.794,-0.35);
336 \end{tikzpicture}%
337 \fi%
338 }%
339 \tikzsymbolsusebox{Sieb#1}%
340 \tikzsymbolsaftersymbolinput%
341 }
342 \tikzsymbols@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”:

```

343 \tikzsymbolsDeclareRobustCommand{Purierstab}[1][1]{%
344 \tikzsymbols@ifsaveboxundefined{Purierstab#1}{%
345 \set@tkzsymbolsscl{#1}%
346 \iftikzymbols@draftabsolute%
347 \tkzsymbolsbxPrmtrstore{#1}%
348 \tikzsymbols@draftbox{0.76ex*\real{\tkzsymbolsPrmtr}}{1.575ex*\real{\tkzsymbolsPrmtr}}%
349 \else%
350 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
351 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.07ex*\tkzsymbolsscl,scale=#1]
352 \draw[rounded corners=0.07ex*\tkzsymbolsscl] (0,0) -- (0.3,0) -- (0.15,0.1) --cycle;
353 \fill[rounded corners=0.07ex*\tkzsymbolsscl] (0.15,0.3) -- (0.24,0.4) -- (0.24,0.7) --
354 (0.06,0.7) -- (0.06,0.4) -- cycle;
355 \draw (0.15,0.4) -- (0.15,0.1);
356 \end{tikzpicture}%
357 \fi%
358 }%
359 \tikzsymbolsusebox{Purierstab#1}%
360 \tikzsymbolsaftersymbolinput%
361 }
362 \tikzsymbols@let{blender}{Purierstab}

```

`\Dreizack = \trident` Important cooking-tool for cooking:

```

363 \tikzsymbolsDeclareRobustCommand{Dreizack}[1][1]{%
364 \tikzsymbols@ifsaveboxundefined{Dreizack#1}{%
365 \set@tkzsymbolsscl{#1}%
366 \iftikzymbols@draftabsolute%
367 \tkzsymbolsbxPrmtrstore{#1}%
368 \tikzsymbols@draftbox{0.265ex*\real{\tkzsymbolsPrmtr}}{1.575ex*\real{\tkzsymbolsPrmtr}}%
369 \else%
370 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
371 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.035ex*\tkzsymbolsscl,scale=#1]
372 \fill[rounded corners=0.07ex*(\tkzsymbolsscl-\tkzsymbolsscl/100)]
373 (0,0) -- (0,0.4) -- (0.1,0.4) -- (0.1,0.0) -- cycle;

```

```

374 \draw (0.05,0) -- (0.05,0.7);
375 \draw[rounded corners=0.07ex*(\tkzsymlbsscl-\tkzsymlbsscl/100*\tkzsymlbsscl*2]
376 (0,0.7) -- (0,0.55) -- (0.05,0.55) -- (0.1,0.55) -- (0.1,0.7);
377 \end{tikzpicture}%
378 \fi%
379 }%
380 \tikzsymbolsusebox{Dreizack#1}%
381 \tikzsymbolsaftersymbolinput%
382 }
383 \tikzsymbols@let{trident}{Dreizack}

```

\Backblech = \bakingplate I may have too many strange named commands:

```

384 \tikzsymlsDeclareRobustCommand{Backblech}[1][1]{%
385 \tikzsymbols@ifsaveboxundefined{Backblech#1}{%
386 \set@tkzsymlbsscl{#1}%
387 \iftikzymbols@draftabsolute%
388 \tkzsymlsbxPrmtrstore{#1}%
389 \tikzsymbols@draftbox{2.3155ex*\real{\tkzsymlsPrmtr}}{1.57ex*\real{\tkzsymlsPrmtr}}}%
390 \else%
391 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
392 \begin{tikzpicture}[x=6.53ex,y=5ex, line width=0.07ex*\tkzsymlbsscl,scale=#1]
393 \filldraw[rounded corners=0.09ex*\tkzsymlbsscl] (0,0) rectangle (0.3,0.3);
394 \draw[rounded corners=0.07ex*\tkzsymlbsscl, line width=0.03ex*\tkzsymlbsscl]
395 (0.1,0) -- (-0.025,0) -- (-0.025,0.3) -- (0.1,0.3);
396 \draw[rounded corners=0.07ex*\tkzsymlbsscl, line width=0.03ex*\tkzsymlbsscl]
397 (0.2,0) -- (.325,0) -- (.325,0.3) -- (0.2,0.3);
398 \foreach \@BackblechlochX in {0.007,0.293}
399 \foreach \@BackblechlochY in {0.007,0.293}
400 \fill[white] (\@BackblechlochX,
401 \@BackblechlochY) circle (0.02ex);
402 \end{tikzpicture}%
403 \fi%
404 }%
405 \tikzsymbolsusebox{Backblech#1}%
406 \tikzsymbolsaftersymbolinput%
407 }
408 \tikzsymbols@let{bakingplate}{Backblech}

```

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

```

409 \tikzsymlsDeclareRobustCommand{Ofen}[1][1]{%
410 \tikzsymbols@ifsaveboxundefined{Ofen#1}{%
411 \set@tkzsymlbsscl{#1}%
412 \iftikzymbols@draftabsolute%
413 \tkzsymlsbxPrmtrstore{#1}%
414 \tikzsymbols@draftbox{2.07ex*\real{\tkzsymlsPrmtr}}{1.57ex*\real{\tkzsymlsPrmtr}}}%
415 \else%
416 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
417 \begin{tikzpicture}[x=0.50ex,y=0.5ex, line width=0.07ex*\tkzsymlbsscl,scale=#1]
418 \draw (0,0) rectangle (4,3);
419 \draw (0.25,0.25) rectangle (3.75,2);

```

```

420 \foreach \Ofenschalter in {0.5,1.1,2.9,3.5}
421 \fill (\Ofenschalter,2.5) circle (0.22);
422 \draw (1.5,2.28) rectangle (2.5,2.72);
423 \draw[line width=0.05ex*\tkzsymlsscl] (1,1.75) -- (3,1.75);
424 \end{tikzpicture}%
425 \fi%
426 }%
427 \tikzsymbolsusebox{Ofen#1}%
428 \tikzsymbolsaftersymbolinput%
429 }
430 \tikzsymbols@let{oven}{Ofen}

\Pfanne = \pan A pan ... What did you expect?
431 \tikzsymbolsDeclareRobustCommand{Pfanne}[1][1]{%
432 \tikzsymbols@ifsaveboxundefined{Pfanne#1}{%
433 \set@tkzsymlsscl{#1}%
434 \iftikzymbols@draftabsolute%
435 \tkzsymlsbxPrmtrstore{#1}%
436 \tikzsymbols@draftbox{3.034ex*\real{\tkzsymlsPrmtr}}{0.78ex*\real{\tkzsymlsPrmtr}}%
437 \else%
438 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
439 \begin{tikzpicture}[x=2.3ex,y=2.3ex,line width=0.09ex*\tkzsymlsscl,scale=#1]
440 \draw [rounded corners=0.023ex*\tkzsymlsscl]
441 (0,0) -- (0.9,0) -- (1,0.3) -- (-0.1,0.3) -- cycle;
442 \draw (-0.2,0.22) -- (-0.08,0.22);
443 \draw (0.97,0.22) -- (1.08,0.22);
444 \draw[decorate,decoration={snake,amplitude=.046ex*\tkzsymlsscl,
445 segment length=0.82ex*\tkzsymlsscl},line width=0.05ex*\tkzsymlsscl]
446 (-0.05,0.1) -- (0.95,0.1);
447 \end{tikzpicture}%
448 \fi%
449 }%
450 \tikzsymbolsusebox{Pfanne#1}%
451 \tikzsymbolsaftersymbolinput%
452 }
453 \tikzsymbols@let{pan}{Pfanne}

\Herd = \cooker I hope it's the right translation:
454 \tikzsymbolsDeclareRobustCommand{Herd}[1][1]{%
455 \tikzsymbols@ifsaveboxundefined{Herd#1}{%
456 \set@tkzsymlsscl{#1}%
457 \iftikzymbols@draftabsolute%
458 \tkzsymlsbxPrmtrstore{#1}%
459 \tikzsymbols@draftbox{2.08ex*\real{\tkzsymlsPrmtr}}{1.58ex*\real{\tkzsymlsPrmtr}}%
460 \else%
461 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
462 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.04ex*\tkzsymlsscl,scale=#1]
463 \draw[line width=0.08ex*\tkzsymlsscl] (0,0) rectangle (2,1.5);
464 \draw (0.5,0.45) circle (0.35);
465 \draw (0.5,0.45) circle (0.2);

```

```

466 \draw (1.45,0.45) circle (0.3);
467 \draw (0.5,1.15) circle (0.21);
468 \draw (1.05,0.95) rectangle (1.85,1.35);
469 \draw (1.45,1.15) circle (0.15);
470 \end{tikzpicture}%
471 \fi%
472 }%
473 \tikzsymbolsusebox{Herd#1}%
474 \tikzsymbolsaftersymbolinput%
475 }
476 \tikzsymbols@let{cooker}{Herd}

```

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

```

477 \tikzsymbolsDeclareRobustCommand{Saftpresse}[1][1]{%
478 \tikzsymbols@ifsaveboxundefined{Saftpresse#1}{%
479 \set@tkzsymbolsscl{#1}%
480 \iftikzymbols@draftabsolute%
481 \tkzsymbols@sbxPrmtrstore{#1}%
482 \tikzsymbols@draftbox{1.87ex*\real{\tkzsymbolsPrmtr}}{1.62ex*\real{\tkzsymbolsPrmtr}}}%
483 \else%
484 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
485 \begin{tikzpicture}[x=1.2ex,y=1ex,line width=0.07ex*\tkzsymbolsscl,scale=#1]
486 \draw[rounded corners=0.1ex*\tkzsymbolsscl]
487 (0,0.85) -- (0,0) -- (1.5,0) -- (1.5,0.85) -- cycle;
488 \draw (0,0.7) -- (1.5,0.7);
489 \draw[rounded corners=0.1ex*\tkzsymbolsscl] (0.3,0.7) -- (0.75,1.55) -- (1.2,0.7);
490 \draw[rounded corners=0.1ex*\tkzsymbolsscl] (0.45,0.7) -- (0.75,1.55) -- (1.05,0.7);
491 \draw[rounded corners=0.1ex*\tkzsymbolsscl]
492 (0.65,0.7) -- (0.75,1.55) -- (0.85,0.7);
493 \draw[line width=0.05ex*\tkzsymbolsscl, decorate,
494 decoration={snake,amplitude=.05ex*\tkzsymbolsscl,
495 segment length=0.48ex*\tkzsymbolsscl}] (0,0.3) -- (1.5,0.3);
496 \end{tikzpicture}%
497 \fi%
498 }%
499 \tikzsymbolsusebox{Saftpresse#1}%
500 \tikzsymbolsaftersymbolinput%
501 }
502 \tikzsymbols@let{squeezer}{Saftpresse}

```

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

```

503 \tikzsymbolsDeclareRobustCommand{Schussel}[1][1]{%
504 \tikzsymbols@ifsaveboxundefined{Schussel#1}{%
505 \set@tkzsymbolsscl{#1}%
506 \iftikzymbols@draftabsolute%
507 \tkzsymbols@sbxPrmtrstore{#1}%
508 \tikzsymbols@draftbox{2.32ex*\real{\tkzsymbolsPrmtr}}{1.47ex*\real{\tkzsymbolsPrmtr}}}%
509 \else%
510 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
511 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex*\tkzsymbolsscl, scale=#1]

```

```

512 \draw[rounded corners=0.5ex*\tkzsymlbsscl]
513   (-0.02,1.4) -- (0,1.4) -- (0,0.05) -- (1.5,0.05) -- (1.5,1.4) -- (1.52,1.4);
514 \draw (0.35,0) -- (1.15,0);
515 \draw[opacity=00] (-0.4,0) -- (1.85,0);
516 \end{tikzpicture}%
517 \fi%
518 }%
519 \tikzsymbolsusebox{Schussel#1}%
520 \tikzsymbolsaftersymbolinput%
521 }
522 \tikzsymbols@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:

```

523 \tikzsymbolsDeclareRobustCommand{Schaler}[1][1]{%
524 \tikzsymbols@ifsaveboxundefined{Schaler#1}{%
525 \set@tkzsymlbsscl{#1}%
526 \iftikzymbols@draftabsolute%
527 \tkzsymlsbxPrmtrstore{#1}%
528 \tikzsymbols@draftbox{1.15ex*\real{\tkzsymlbPrmtr}}{1.565ex*\real{\tkzsymlbPrmtr}}%
529 \else%
530 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
531 \begin{tikzpicture}[x=2.7ex,y=2.3ex, line width=0.07ex*\tkzsymlbsscl,scale=#1]
532 \draw[rounded corners=0.07ex*\tkzsymlbsscl]
533   (0,0.4) -- (0,0.1) arc (0:180:-0.1) -- (0.2,0.4)
534   -- (0.3,0.5) -- (0.3,0.65) -- (0.2,0.65) -- (0.2,0.5) -- (0,0.5) -- (0,0.65) --
535   (-0.1,0.65) -- (-0.1,0.5) -- cycle;
536 \draw[line width=0.03ex*\tkzsymlbsscl] (0,0.6) -- (0.2,0.6);
537 \draw[line width=0.03ex*\tkzsymlbsscl] (0,0.58) -- (0.2,0.58);
538 \end{tikzpicture}%
539 \fi%
540 }%
541 \tikzsymbolsusebox{Schaler#1}%
542 \tikzsymbolsaftersymbolinput%
543 }
544 \tikzsymbols@let{peeler}{Schaler}

```

6.2 Emoticonscode

`\Sadey \dSadey` Another 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).

```

545 \tikzsymbolsDeclareRobustCommandx{Sadey}[2][1=1, 2={opacity=0}, usedefault]{%
546 \tikzsymbols@ifsaveboxundefined{Sadey#1#2}{%
547 \set@tkzsymlbsscl{#1}%
548 \iftikzymbols@draftabsolute%
549 \tkzsymlsbxPrmtrstore{#1}%

```



```

550 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbolsPrmtr}}%
551 \else%
552 \ifdim\tkzsymbolsscl<Opt\set\tkzsymbolsscl{-#1}\fi%
553 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
554 \fill[#2, line width=0.1ex*\tkzsymbolsscl] (0,0) circle (0.33);
555 \draw[line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
556 \fill (0.1,0.1) circle (0.05);
557 \fill (-0.1,0.1) circle (0.05);
558 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
559 \end{tikzpicture}%
560 \fi%
561 }%
562 \tikzsymbolsusebox{Sadey#1#2}%
563 \tikzsymbolsaftersymbolinput%
564 }
565 \tkzsymbolsDeclareRobustCommandx{dSadey}[2][1=1,2=yellow,usedefault]{%
566 \tikzsymbols@ifsaveboxundefined{dSadey#1#2}{%
567 \set\tkzsymbolsscl{#1}%
568 \iftikzymbols@draftabsolute%
569 \tkzsymbolsbxPrmtrstore{#1}%
570 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbolsPrmtr}}%
571 \else%
572 \ifdim\tkzsymbolsscl<Opt\set\tkzsymbolsscl{-#1}\fi%
573 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
574 \shade[ball color=#2] (0,0) circle (0.33);
575 \shade[ball color=black] (0.1,0.1) circle (0.05);
576 \shade[ball color=black] (-0.1,0.1) circle (0.05);
577 \draw[black] (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
578 \end{tikzpicture}%
579 \fi%
580 }%
581 \tikzsymbolsusebox{dSadey#1#2}%
582 \tikzsymbolsaftersymbolinput%
583 }

```

\Annoey \dAnnoey An annoyed Smiley -_-

```

584 \tkzsymbolsDeclareRobustCommandx{Annoey}[2][1=1,2={opacity=0},usedefault]{%
585 \tikzsymbols@ifsaveboxundefined{Annoey#1#2}{%
586 \set\tkzsymbolsscl{#1}%
587 \iftikzymbols@draftabsolute%
588 \tkzsymbolsbxPrmtrstore{#1}%
589 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbolsPrmtr}}%
590 \else%
591 \ifdim\tkzsymbolsscl<Opt\set\tkzsymbolsscl{-#1}\fi%
592 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
593 \fill[#2, line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
594 \draw[line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
595 \draw (0.08,0.1) -- (0.22,0.1);
596 \draw (-0.08,0.1) -- (-0.22,0.1);
597 \draw (-0.2,-0.1) -- (0.2,-0.1);

```

```

598 \end{tikzpicture}%
599 \fi%
600 }%
601 \tikzsymbolsusebox{Annoey#1#2}%
602 \tikzsymbolsaftersymbolinput%
603 }
604 \tkzsymlsDeclareRobustCommandx{dAnnoey}[2][1=1,2=yellow,usedefault]{%
605 \tikzsymbols@ifsaveboxundefined{dAnnoey#1#2}{%
606 \set@tkzsymlscl{#1}%
607 \iftikzymbols@draftabsolute%
608 \tkzsymlsboxPrmtrstore{#1}%
609 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymlsPrmtr}}%
610 \else%
611 \ifdim\tkzsymlscl<0pt\set@tkzsymlscl{-#1}\fi%
612 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlscl,scale=#1]
613 \shade[ball color=#2] (0,0) circle (0.33);
614 \draw[black] (0.08,0.1) -- (0.22,0.1);
615 \draw[black] (-0.08,0.1) -- (-0.22,0.1);
616 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
617 \end{tikzpicture}%
618 \fi%
619 }%
620 \tikzsymbolsusebox{dAnnoey#1#2}%
621 \tikzsymbolsaftersymbolinput%
622 }

```

\Smiley \dSmiley A normal Smiley

```

623 \ifKV@tikzsymbols@marvosym\relax\else%
624 \tkzsymlsDeclareRobustCommandx{Smiley}[2][1=1,2={opacity=0} ,usedefault]{%
625 \tikzsymbols@ifsaveboxundefined{Smiley#1#2}{%
626 \set@tkzsymlscl{#1}%
627 \iftikzymbols@draftabsolute%
628 \tkzsymlsboxPrmtrstore{#1}%
629 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}%
630 \else%
631 \ifdim\tkzsymlscl<0pt\set@tkzsymlscl{-#1}\fi%
632 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlscl, scale=#1]
633 \fill[#2] (0,0) circle (0.33);
634 \draw (0,0) circle (0.33);
635 \fill (-0.1,0.1) circle (0.05);
636 \fill (0.1,0.1) circle (0.05);
637 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
638 \end{tikzpicture}%
639 \fi%
640 }%
641 \tikzsymbolsusebox{Smiley#1#2}%
642 \tikzsymbolsaftersymbolinput%
643 }%
644 \fi
645 \tkzsymlsDeclareRobustCommandx{dSmiley}[2][1=1,2=yellow,usedefault]{%

```

```

646 \tikzsymbols@ifsaveboxundefined{dSmiley#1#2}{%
647 \set@tkzsymbolsscl{#1}%
648 \iftikzsymbols@draftabsolute%
649 \tkzsymbolsbxPrmtrstore{#1}%
650 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbolsPrmtr}}}%
651 \else%
652 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
653 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.1ex*\tkzsymbolsscl,scale=#1]
654 \shade[ball color=#2] (0,0) circle (0.33);
655 \shade[ball color=black] (-0.1,0.1) circle (0.05);
656 \shade[ball color=black] (0.1,0.1) circle (0.05);
657 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
658 \end{tikzpicture}%
659 \fi%
660 }%
661 \tikzsymbolsusebox{dSmiley#1#2}%
662 \tikzsymbolsaftersymbolinput%
663 }

```

\Laughey \dLaughey A laughing Smiley

```

664 \tikzsymbolsDeclareRobustCommandx{Laughey}[3][1=1,2={opacity=0},3={opacity=0} ,usedefault]{%
665 \tikzsymbols@ifsaveboxundefined{Laughey#1#2#3}{%
666 \set@tkzsymbolsscl{#1}%
667 \iftikzsymbols@draftabsolute%
668 \tkzsymbolsbxPrmtrstore{#1}%
669 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbolsPrmtr}}}%
670 \else%
671 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
672 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
673 \fill[#2,line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
674 \draw[line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
675 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
676 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
677 \fill[#3,rounded corners=0.1ex*\tkzsymbolsscl, yshift=-0.5]
678 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
679 \draw[rounded corners=0.1ex*\tkzsymbolsscl, yshift=-0.5]
680 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
681 \end{tikzpicture}%
682 \fi%
683 }%
684 \tikzsymbolsusebox{Laughey#1#2#3}%
685 \tikzsymbolsaftersymbolinput%
686 }
687 \tikzsymbolsDeclareRobustCommandx{dLaughey}[3][1=1,2=yellow, 3=red ,usedefault]{%
688 \tikzsymbols@ifsaveboxundefined{dLaughey#1#2#3}{%
689 \set@tkzsymbolsscl{#1}%
690 \iftikzsymbols@draftabsolute%
691 \tkzsymbolsbxPrmtrstore{#1}%
692 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbolsPrmtr}}}%
693 \else%

```

```

694 \ifdim\tkzsymlsscl<Opt\set@tkzsymlsscl{-#1}\fi%
695 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
696 \fill[ball color=#2,line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
697 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
698 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
699 \shade[ball color=#3, rounded corners=0.1ex*\tkzsymlsscl, yshift=-0.3]
700 (-0.25,-0.0) .. controls (-0.13,-0.26) and (0.13,-0.26) .. (0.25,-0.0) -- cycle;
701 \end{tikzpicture}%
702 \fi%
703 }%
704 \tikzsymbolsusebox{dLaughey#1#2#3}%
705 \tikzsymbolsaftersymbolinput%
706 }

```

\Neutrey \dNeutrey neutral Smiley :|

```

707 \tkzsymlsDeclareRobustCommandx{Neutrey}[2][1=1, 2={opacity=0}, usedefault]{%
708 \tikzsymbols@ifsaveboxundefined{Neutrey#1#2}{%
709 \set@tkzsymlsscl{#1}%
710 \iftikzymbols@draftabsolute%
711 \tkzsymlsboxPrmtrstore{#1}%
712 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}}%
713 \else%
714 \ifdim\tkzsymlsscl<Opt\set@tkzsymlsscl{-#1}\fi%
715 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
716 \fill[#2,line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
717 \draw[line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
718 \fill (0.1,0.1) circle (0.05);
719 \fill (-0.1,0.1) circle (0.05);
720 \draw (-0.2,-0.1) -- (0.2,-0.1);
721 \end{tikzpicture}%
722 \fi%
723 }%
724 \tikzsymbolsusebox{Neutrey#1#2}%
725 \tikzsymbolsaftersymbolinput%
726 }
727 \tkzsymlsDeclareRobustCommandx{dNeutrey}[2][1=1,2=yellow,usedefault]{%
728 \tikzsymbols@ifsaveboxundefined{dNeutrey#1#2}{%
729 \set@tkzsymlsscl{#1}%
730 \iftikzymbols@draftabsolute%
731 \tkzsymlsboxPrmtrstore{#1}%
732 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymlsPrmtr}}}%
733 \else%
734 \ifdim\tkzsymlsscl<Opt\set@tkzsymlsscl{-#1}\fi%
735 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
736 \shade[ball color=#2] (0,0) circle (0.33);
737 \shade[ball color=black] (0.1,0.1) circle (0.05);
738 \shade[ball color=black] (-0.1,0.1) circle (0.05);
739 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
740 \end{tikzpicture}%
741 \fi%

```

```

742 }%
743 \tikzsymbolsusebox{dNeutrey#1#2}%
744 \tikzsymbolsaftersymbolinput%
745 }

\Winkey \dWinkey ;)
\oldWinkey \olddWinkey 746 \tkzsymlsDeclareRobustCommandx{Winkey}[2][1=1,2={opacity=0} ,usedefault]{%
747 \tikzsymbols@ifsaveboxundefined{Winkey#1#2}{%
748 \set@tkzsymlscl{#1}%
749 \iftikzymbols@draftabsolute%
750 \tkzsymlsboxPrmtrstore{#1}%
751 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}%
752 \else%
753 \ifdim\tkzsymlscl<Opt\set@tkzsymlscl{-#1}\fi%
754 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlscl,scale=#1]
755 \fill[#2] (0,0) circle (0.33);
756 \draw(0,0) circle (0.33);
757 \draw(0.17,0.1) -- (0.05,0.1);
758 \fill (-0.1,0.1) circle (0.05);
759 \draw (-0.15,-0.15) .. controls (-0.05,-0.2) and (0.15,-0.2) .. (0.19,0);
760 \end{tikzpicture}%
761 \fi%
762 }%
763 \tikzsymbolsusebox{Winkey#1#2}%
764 \tikzsymbolsaftersymbolinput%
765 }

766 \tkzsymlsDeclareRobustCommandx{oldWinkey}[2][1=1,2={opacity=0} ,usedefault]{%
767 \tikzsymbols@ifsaveboxundefined{oldWinkey#1#2}{%
768 \set@tkzsymlscl{#1}%
769 \iftikzymbols@draftabsolute%
770 \tkzsymlsboxPrmtrstore{#1}%
771 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}%
772 \else%
773 \ifdim\tkzsymlscl<Opt\set@tkzsymlscl{-#1}\fi%
774 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlscl,scale=#1]
775 \fill[#2] (0,0) circle (0.33);
776 \draw(0,0) circle (0.33);
777 \draw(0.17,0.1) -- (0.05,0.1);
778 \fill (-0.1,0.1) circle (0.05);
779 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
780 \end{tikzpicture}%
781 \fi%
782 }%
783 \tikzsymbolsusebox{oldWinkey#1#2}%
784 \tikzsymbolsaftersymbolinput%
785 }

786 \tkzsymlsDeclareRobustCommandx{dWinkey}[2][1=1,2=yellow,usedefault]{%
787 \tikzsymbols@ifsaveboxundefined{dWinkey#1#2}{%
788 \set@tkzsymlscl{#1}%
789 \iftikzymbols@draftabsolute%

```

```

790 \tkzsymlsbxPrmtrstore{#1}%
791 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymlsbPrmtr}}%
792 \else%
793 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
794 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlbsscl,scale=#1]
795 \shade[ball color=#2] (0,0) circle (0.33);
796 \draw[black] (0.17,0.1) -- (0.05,0.1);
797 \shade[ball color=black] (-0.1,0.1) circle (0.05);
798 \draw[black] (-0.15,-0.15) .. controls (-0.05,-0.2) and (0.15,-0.2) .. (0.19,0);
799 \end{tikzpicture}%
800 \fi%
801 }%
802 \tikzsymbolsusebox{dWinkey#1#2}%
803 \tikzsymbolsaftersymbolinput%
804 }
805 \tkzsymlsDeclareRobustCommandx{olddWinkey}[2][1=1,2=yellow,usedefault]{%
806 \tikzsymbols@ifsaveboxundefined{olddWinkey#1#2}{%
807 \set@tkzsymlbsscl{#1}%
808 \iftikzymbols@draftabsolute%
809 \tkzsymlsbxPrmtrstore{#1}%
810 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymlsbPrmtr}}%
811 \else%
812 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
813 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlbsscl,scale=#1]
814 \shade[ball color=#2] (0,0) circle (0.33);
815 \draw(0.17,0.1) -- (0.05,0.1);
816 \shade[ball color=black] (-0.1,0.1) circle (0.05);
817 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
818 \end{tikzpicture}%
819 \fi%
820 }%
821 \tikzsymbolsusebox{olddWinkey#1#2}%
822 \tikzsymbolsaftersymbolinput%
823 }

```

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

```

824 \tkzsymlsDeclareRobustCommandx{Sey}[2][1=1,2={opacity=0} ,usedefault]{%
825 \tikzsymbols@ifsaveboxundefined{Sey#1#2}{%
826 \set@tkzsymlbsscl{#1}%
827 \iftikzymbols@draftabsolute%
828 \tkzsymlsbxPrmtrstore{#1}%
829 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsbPrmtr}}%
830 \else%
831 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
832 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
833 \fill[#2, line width=0.12ex*\tkzsymlbsscl] (0,0) circle (0.33);
834 \draw[line width=0.12ex*\tkzsymlbsscl] (0,0) circle (0.33);
835 \fill (0.1,0.1) circle (0.05);
836 \fill (-0.1,0.1) circle (0.05);
837 \draw (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);

```

```

838 \end{tikzpicture}%
839 \fi%
840 }%
841 \tikzsymbolsusebox{Sey#1#2}%
842 \tikzsymbolsaftersymbolinput%
843 }
844 \tkzsymlsDeclareRobustCommandx{dSey}[2][1=1,2=yellow ,usedefault]{%
845 \tikzsymbols@ifsaveboxundefined{dSey#1#2}{%
846 \set@tkzsymlsscl{#1}%
847 \iftikzymbols@draftabsolute%
848 \tkzsymlsboxPrmtrstore{#1}%
849 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymlsPrmtr}}}%
850 \else%
851 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
852 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
853 \shade[ball color=#2] (0,0) circle (0.33);
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] (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
857 \end{tikzpicture}%
858 \fi%
859 }%
860 \tikzsymbolsusebox{dSey#1#2}%
861 \tikzsymbolsaftersymbolinput%
862 }

```

\Xey \dXey I can't think of a better name again.

```

863 \tkzsymlsDeclareRobustCommandx{Xey}[2][1=1, 2={opacity=0}, usedefault]{%
864 \tikzsymbols@ifsaveboxundefined{Xey#1#2}{%
865 \set@tkzsymlsscl{#1}%
866 \iftikzymbols@draftabsolute%
867 \tkzsymlsboxPrmtrstore{#1}%
868 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}}%
869 \else%
870 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
871 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
872 \fill[#2, line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
873 \draw[line width=0.12ex*\tkzsymlsscl] (0,0) circle (0.33);
874 \draw (0.05,0.05) -- ++ (0.1,0.1);
875 \draw (0.15,0.05) -- ++ (-0.1,0.1);
876 \draw (-0.05,0.05) -- ++ (-0.1,0.1);
877 \draw (-0.15,0.05) -- ++ (0.1,0.1);
878 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
879 \end{tikzpicture}%
880 \fi%
881 }%
882 \tikzsymbolsusebox{Xey#1#2}%
883 \tikzsymbolsaftersymbolinput%
884 }
885 \tkzsymlsDeclareRobustCommandx{dXey}[2][1=1, 2={yellow}, usedefault]{%

```

```

886 \tikzsymbols@ifsaveboxundefined{dXey#1#2}{%
887 \set@tkzsymbolsscl{#1}%
888 \iftikzymbols@draftabsolute%
889 \tkzsymbolsbxPrmtrstore{#1}%
890 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbolsPrmtr}}}%
891 \else%
892 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
893 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1]
894 \fill[ball color=#2, line width=0.12ex*\tkzsymbolsscl] (0,0) circle (0.33);
895 \draw (0.05,0.05) -- ++ (0.1,0.1);
896 \draw (0.15,0.05) -- ++ (-0.1,0.1);
897 \draw (-0.05,0.05) -- ++ (-0.1,0.1);
898 \draw (-0.15,0.05) -- ++ (0.1,0.1);
899 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
900 \end{tikzpicture}%
901 \fi%
902 }%
903 \tikzsymbolsusebox{dXey#1#2}%
904 \tikzsymbolsaftersymbolinput%
905 }

```

\Innocey \dInnocey An innocent Smiley

```

906 \tkzsymbolsDeclareRobustCommand{\Innocey}[3][1=1,2={opacity=0},3=yellow ,usedefault]{%
907 \tikzsymbols@ifsaveboxundefined{Innocey#1#2#3}{%
908 \set@tkzsymbolsscl{#1}%
909 \iftikzymbols@draftabsolute%
910 \tkzsymbolsbxPrmtrstore{#1}%
911 \tikzsymbols@draftQbox{1.73ex*\real{\tkzsymbolsPrmtr}}{1.909ex*\real{\tkzsymbolsPrmtr}}}%
912 \else%
913 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
914 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbolsscl,scale=#1]
915 \fill[#2] (0,0) circle (0.33);
916 \draw (0,0) circle (0.33);
917 \fill (-0.1,0.1) circle (0.05);
918 \fill (0.1,0.1) circle (0.05);
919 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
920 \draw[#3, line width=0.095ex*\tkzsymbolsscl] (0.32,0.31) arc (0:360:0.32 and 0.1);
921 \draw[line width=0.05ex*\tkzsymbolsscl] (0.3,0.31) arc (0:360:0.3 and 0.07);
922 \draw[line width=0.05ex*\tkzsymbolsscl] (0.35,0.31) arc (0:360:0.35 and 0.12);
923 \end{tikzpicture}%
924 \fi%
925 }%
926 \tikzsymbolsusebox{Innocey#1#2#3}%
927 \tikzsymbolsaftersymbolinput%
928 }
929 \tkzsymbolsDeclareRobustCommand{\wInnocey}[1][1]{\Innocey[#1][opacity=0][white]}
930 \tkzsymbolsDeclareRobustCommand{\dInnocey}[3][1=1,2=yellow,3=yellow,usedefault]{%
931 \tikzsymbols@ifsaveboxundefined{dInnocey#1#2#3}{%
932 \set@tkzsymbolsscl{#1}%
933 \iftikzymbols@draftabsolute%

```



```

934 \tkzsymlsboxPrmtrstore{#1}%
935 \tikzsymbols@draftbox{1.73ex*\real{\tkzsymlsPrmtr}}{1.849ex*\real{\tkzsymlsPrmtr}}%
936 \else%
937 \ifdim\tkzsymlsscl<0pt\set\tkzsymlsscl{-#1}\fi%
938 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlsscl,scale=#1]
939 \shade[ball color=#2] (0,0) circle (0.33);
940 \shade[ball color=black] (-0.1,0.1) circle (0.05);
941 \shade[ball color=black] (0.1,0.1) circle (0.05);
942 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
943 \draw[color=#3!97!black, line width=0.1ex*\tkzsymlsscl]
944 (0.32,0.31) arc (0:360:0.32 and 0.1);
945 \draw[line width=0.05ex*\tkzsymlsscl] (0.3,0.31) arc (0:360:0.3 and 0.07);
946 \draw[line width=0.05ex*\tkzsymlsscl] (0.35,0.31) arc (0:360:0.35 and 0.12);
947 \end{tikzpicture}%
948 \fi%
949 }%
950 \tikzsymbolsusebox{dInnocey#1#2#3}%
951 \tikzsymbolsaftersymbolinput%
952 }

```

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

```

953 \tkzsymlsDeclareRobustCommandx{Cooley}[2][1=1,2={opacity=0},usedefault]{%
954 \tikzsymbols@ifsaveboxundefined{Cooley#1#2}{%
955 \set\tkzsymlsscl{#1}%
956 \iftikzymbols@draftabsolute%
957 \tkzsymlsboxPrmtrstore{#1}%
958 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}%
959 \else%
960 \ifdim\tkzsymlsscl<0pt\set\tkzsymlsscl{-#1}\fi%
961 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlsscl,scale=#1]
962 \fill[#2] (0,0) circle (0.33);
963 \draw (0,0) circle (0.33);
964 \fill[rounded corners=0.1ex*\tkzsymlsscl]
965 (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
966 \fill[rounded corners=0.1ex*\tkzsymlsscl]
967 (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
968 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
969 \draw (0.02,0.1) -- (-0.02,0.1);
970 \draw (-0.2,0.1) -- (-0.3,0.13);
971 \draw (0.2,0.1) -- (0.3,0.13);
972 \end{tikzpicture}%
973 \fi%
974 }%
975 \tikzsymbolsusebox{Cooley#1#2}%
976 \tikzsymbolsaftersymbolinput%
977 }
978 \tkzsymlsDeclareRobustCommandx{dCooley}[2][1=1,2=yellow,usedefault]{%
979 \tikzsymbols@ifsaveboxundefined{dCooley#1#2}{%
980 \set\tkzsymlsscl{#1}%
981 \iftikzymbols@draftabsolute%

```

```

982 \tkzsymlsboxPrmtrstore{#1}%
983 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymlsPrmtr}}%
984 \else%
985 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
986 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlsscl,scale=#1]
987 \shade[ball color=#2] (0,0) circle (0.33);
988 \draw[black] (0.02,0.1) -- (-0.02,0.1);
989 \draw[black] (-0.2,0.1) -- (-0.295,0.146);
990 \draw[black] (0.2,0.1) -- (0.295,0.146);
991 \shade[ball color=black,rounded corners=0.1ex*\tkzsymlsscl]
992 (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
993 \shade[ball color=black,rounded corners=0.1ex*\tkzsymlsscl]
994 (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
995 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
996 \end{tikzpicture}%
997 \fi%
998 }%
999 \tikzsymbolsusebox{dCooley#1#2}%
1000 \tikzsymbolsaftersymbolinput%
1001 }

```

\Tongey \dTongey :P

```

1002 \tkzsymlsDeclareRobustCommandx{Tongey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
1003 \tikzsymbols@ifsaveboxundefined{Tongey#1#2#3}{%
1004 \set@tkzsymlsscl{#1}%
1005 \iftikzymbols@draftabsolute%
1006 \tkzsymlsboxPrmtrstore{#1}%
1007 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymlsPrmtr}}%
1008 \else%
1009 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
1010 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymlsscl,scale=#1]
1011 \fill[#2] (0,0) circle (0.33);
1012 \draw (0,0) circle (0.33);
1013 \fill (-0.1,0.1) circle (0.05);
1014 \fill (0.1,0.1) circle (0.05);
1015 \fill[#3,line width=0.058ex*\tkzsymlsscl, rounded corners=0.12ex*\tkzsymlsscl]
1016 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1017 \draw[line width=0.07ex*\tkzsymlsscl, yshift=0.21ex]
1018 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1019 \draw[line width=0.058ex*\tkzsymlsscl, rounded corners=0.12ex*\tkzsymlsscl]
1020 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1021 \end{tikzpicture}%
1022 \fi%
1023 }%
1024 \tikzsymbolsusebox{Tongey#1#2#3}%
1025 \tikzsymbolsaftersymbolinput%
1026 }
1027 \tkzsymlsDeclareRobustCommandx{dTongey}[3][1=1,2=yellow,3=red,usedefault]{%
1028 \tikzsymbols@ifsaveboxundefined{dTongey#1#2#3}{%
1029 \set@tkzsymlsscl{#1}%

```

```

1030 \iftikzsymbols@draftabsolute%
1031 \tkzsymbolsbxPrmtrstore{#1}%
1032 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbolsPrmtr}}}%
1033 \else%
1034 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1035 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbolsscl,scale=#1]
1036 \shade[ball color=#2] (0,0) circle (0.33);
1037 \shade[ball color=black] (-0.1,0.1) circle (0.05);
1038 \shade[ball color=black] (0.1,0.1) circle (0.05);
1039 \shade[ball color=#3,line width=0.058ex*\tkzsymbolsscl, rounded corners=0.12ex*\tkzsymbolsscl]
1040 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1041 \draw[black, line width=0.058ex*\tkzsymbolsscl, rounded corners=0.12ex*\tkzsymbolsscl]
1042 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1043 \draw[black, line width=0.07ex*\tkzsymbolsscl, yshift=0.21ex]
1044 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1045 \end{tikzpicture}%
1046 \fi%
1047 }%
1048 \tikzsymbolsusebox{dTongey#1#2#3}%
1049 \tikzsymbolsaftersymbolinput%
1050 }

```

\Nursey \dNursey a Nurse.

```

1051 \tkzsymbolsDeclareRobustCommandx{Nursey}
1052 [4][1=1,2={opacity=0},3={opacity=0},4=black,usedefault]{%
1053 \tikzsymbols@ifsavexundefined{Nursey#1#2#3#4}{%
1054 \set@tkzsymbolsscl{#1}%
1055 \iftikzsymbols@draftabsolute%
1056 \tkzsymbolsbxPrmtrstore{#1}%
1057 \tikzsymbols@draftbox{1.5ex*\real{\tkzsymbolsPrmtr}}{2.19ex*\real{\tkzsymbolsPrmtr}}}%
1058 \else%
1059 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1060 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymbolsscl,scale=#1]
1061 \fill[#3,rounded corners=.023ex*\tkzsymbolsscl]
1062 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
1063 \fill[#2] (0,0) circle (0.3);
1064 \draw (0,0) circle (0.3);
1065 \fill (-0.1,0.1) circle (0.05);
1066 \fill (0.1,0.1) circle (0.05);
1067 \draw[line width=0.09ex*\tkzsymbolsscl, yshift=0.07ex]
1068 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1069 \draw[rounded corners=.023ex*\tkzsymbolsscl]
1070 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
1071 \draw[#4,line width=.046ex*\tkzsymbolsscl] (0,0.35) -- (0,0.5);
1072 \draw[#4,line width=.046ex*\tkzsymbolsscl] (-0.05,0.45) -- (0.05,0.45);
1073 \end{tikzpicture}%
1074 \fi%
1075 }%
1076 \tikzsymbolsusebox{Nursey#1#2#3#4}%
1077 \tikzsymbolsaftersymbolinput%

```

```

1078 }
1079 \tkzsymlsDeclareRobustCommandx{dNursey}[4][1=1,2=yellow,3=white,4=red,usedefault]{%
1080 \tikzsymbols@ifsaveboxundefined{dNursey#1#2#3#4}{%
1081 \set@tkzsymlsscl{#1}%
1082 \iftikzsymbols@draftabsolute%
1083 \tkzsymlsboxPrmtrstore{#1}%
1084 \tikzsymbols@draftbox{1.38ex*\real{\tkzsymlsPrmtr}}{1.98ex*\real{\tkzsymlsPrmtr}}}%
1085 \else%
1086 \ifdim\tkzsymlsscl<Opt\set@tkzsymlsscl{-#1}\fi%
1087 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymlsscl,scale=#1]
1088 \shade[ball color=#2] (0,0) circle (0.3);
1089 \shade[ball color=black] (-0.1,0.1) circle (0.05);
1090 \shade[ball color=black] (0.1,0.1) circle (0.05);
1091 \draw[black, line width=0.09ex*\tkzsymlsscl, yshift=0.07ex]
1092 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1093 \shade[ball color=#3, rounded corners=.023ex*\tkzsymlsscl,yshift=-0.09ex]
1094 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0) arc (0:180:0.3);
1095 \shade[ball color=#4,line width=.046ex*\tkzsymlsscl]
1096 (-0.01,0.31) -- (-0.01,0.46) -- (0.01,0.46) -- (0.01,0.31)--cycle;
1097 \shade[ball color=#4,line width=.046ex*\tkzsymlsscl]
1098 (-0.05,0.4) -- (0.05,0.4) -- (0.05,0.42)--(-0.05,0.42) -- cycle;
1099 \end{tikzpicture}%
1100 \fi%
1101 }%
1102 \tikzsymbolsusebox{dNursey#1#2#3#4}%
1103 \tikzsymbolsaftersymbolinput%
1104 }

```

\Vomey \dVomey *Bläärgh*

```

1105 \tkzsymlsDeclareRobustCommandx{Vomey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
1106 \tikzsymbols@ifsaveboxundefined{Vomey#1#2#3}{%
1107 \set@tkzsymlsscl{#1}%
1108 \iftikzsymbols@draftabsolute%
1109 \tkzsymlsboxPrmtrstore{#1}%
1110 \tikzsymbols@draftbox{3.0335ex*\real{\tkzsymlsPrmtr}}{1.743ex*\real{\tkzsymlsPrmtr}}}%
1111 \else%
1112 \ifdim\tkzsymlsscl<Opt\set@tkzsymlsscl{-#1}\fi%
1113 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymlsscl,scale=#1]
1114 \fill[#2,rounded corners=0.05ex*\tkzsymlsscl] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1115 \draw[rounded corners=0.05ex*\tkzsymlsscl] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1116 \draw[line width=0.05ex*\tkzsymlsscl] (-0.5,0.3) -- (-0.3,0.1);
1117 \fill (-0.45,0.27) arc (100:350:0.1);
1118 \fill[#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1119 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
1120 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1121 \fill[#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) ..
1122 controls (1.7,-1.2) and (1.3,-1.2) .. (1,-1) ..
1123 controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
1124 \draw (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1);
1125 \draw (0,-0.4) .. controls (0.5,-0.5) and (0.8,-0.7) .. (1,-1);

```

```

1126 \draw (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1127     controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2)
1128     and (1,-2) .. (1,-1.5) .. controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1129 \end{tikzpicture}%
1130 \fi%
1131 }%
1132 \tikzsymbolsusebox{Vomey#1#2#3}%
1133 \tikzsymbolsaftersymbolinput%
1134 }
1135 \tkzsymsDeclareRobustCommandx{dVomey}[3][1=1,2=yellow,3={brown!10!olive},usedefault]{%
1136 \tikzsymbols@ifsaveboxundefined{dVomey#1#2#3}{%
1137 \set@tkzsymslsscl{#1}%
1138 \iftikzymbols@draftabsolute%
1139 \tkzsymsbxPrmtrstore{#1}%
1140 \tikzsymbols@draftbox{3.2435ex*\real{\tkzsymsPrmtr}}{1.653ex*\real{\tkzsymsPrmtr}}%
1141 \else%
1142 \ifdim\tkzsymslsscl<0pt\set@tkzsymslsscl{-#1}\fi%
1143 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymslsscl,scale=#1]
1144 \shade[ball color=#2!90!brown,rounded corners=0.03ex*\tkzsymslsscl]
1145     (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1146 \draw[black, line width=0.05ex*\tkzsymslsscl] (-0.5,0.3) -- (-0.3,0.1);
1147 \shade[ball color=black] (-0.45,0.27) arc (100:350:0.1);
1148 \shade[ball color=#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1149     controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
1150     controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1151 \shade[ball color=#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) .. controls
1152     (1.7,-1.2) and (1.3,-1.2) .. (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
1153 \end{tikzpicture}%
1154 \fi%
1155 }%
1156 \tikzsymbolsusebox{dVomey#1#2#3}%
1157 \tikzsymbolsaftersymbolinput%
1158 }

```

\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.

```

1159 \tkzsymsDeclareRobustCommandx{Walley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
1160 \tikzsymbols@ifsaveboxundefined{Walley#1#2#3}{%
1161 \set@tkzsymslsscl{#1}%
1162 \iftikzymbols@draftabsolute%
1163 \tkzsymsbxPrmtrstore{#1}%
1164 \tikzsymbols@draftbox{2.341ex*\real{\tkzsymsPrmtr}}{1.674ex*\real{\tkzsymsPrmtr}}%
1165 \else%
1166 \ifdim\tkzsymslsscl<0pt\set@tkzsymslsscl{-#1}\fi%
1167 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymslsscl,scale=#1,
1168 decoration={random steps,segment length=0.15ex*\tkzsymslsscl, amplitude=0.1ex*\tkzsymslsscl}]
1169 \fill[#2, line width=0.08ex*\tkzsymslsscl] (0,0) circle (0.28);
1170 \draw[line width=0.08ex*\tkzsymslsscl] (0,0) circle (0.28);
1171 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
1172 \draw (0.28,-0.33) rectangle (0.66,0.33);

```

```

1173 \draw[line width=0.06ex*\tkzsymbolsscl]
1174 (0.28,0) ---+(0.05,0.07) ---+(0.03,0.02) ---+
1175 +(0.03,-0.02) ---+(0.03,0.1) ---+(0.03,0.02) -- (0.5,0.25);
1176 \draw[line width=0.06ex*\tkzsymbolsscl]
1177 (0.28,0) ---+(0.06,-0.02) ---+(0.04,0.06) ---+
1178 +(0.0,-0.08) ---+(0.08,0.06) ---+(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);
1179 \draw[line width=0.06ex*\tkzsymbolsscl]
1180 (0.28,0) ---+(0.03,-0.02) ---+(0.03,-0.07) ---+
1181 +(0.03,-0.01) ---+(0.01,-0.07) ---+(0.06,0.01) ---+(0.03,-0.08) --
1182 (0.5,0.-0.25);
1183 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1184 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1185 \end{tikzpicture}%
1186 \fi%
1187 }%
1188 \tikzsymbolsusebox{Walley#1#2#3}%
1189 \tikzsymbolsaftersymbolinput%
1190 }
1191 \tkzsymbolsDeclareRobustCommandx{rWalley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
1192 \tikzsymbols@ifsaveboxundefined{rWalley#1#2#3}{%
1193 \set@tkzsymbolsscl{#1}%
1194 \iftikzymbols@draftabsolute%
1195 \tkzsymbolsbxPrmtrstore{#1}%
1196 \tikzsymbols@draftbox{2.341ex*\real{\tkzsymbolsPrmtr}}{1.674ex*\real{\tkzsymbolsPrmtr}}%
1197 \else%
1198 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1199 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1,
1200 decoration={random steps,segment length=0.15ex*\tkzsymbolsscl, amplitude=0.1ex*\tkzsymbolsscl}]
1201 \fill[#2, line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.28);
1202 \draw[line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.28);
1203 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
1204 \draw (0.28,-0.33) rectangle (0.66,0.33);
1205 \draw[decorate, line width=0.06ex*\tkzsymbolsscl] (0.28,0) -- (0.5,0.25);
1206 \draw[decorate,line width=0.06ex*\tkzsymbolsscl] (0.28,0) -- (0.6,0.0);
1207 \draw[decorate,line width=0.06ex*\tkzsymbolsscl] (0.28,0) -- (0.5,-0.25);
1208 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1209 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1210 \end{tikzpicture}%
1211 \fi%
1212 }%
1213 \tikzsymbolsusebox{rWalley#1#2#3}%
1214 \tikzsymbolsaftersymbolinput%
1215 }
1216 \tkzsymbolsDeclareRobustCommandx{dWalley}[2][1=1, 2=yellow, usedefault]{%
1217 \tikzsymbols@ifsaveboxundefined{dWalley#1#2}{%
1218 \set@tkzsymbolsscl{#1}%
1219 \iftikzymbols@draftabsolute%
1220 \tkzsymbolsbxPrmtrstore{#1}%
1221 \tikzsymbols@draftbox{2.428ex*\real{\tkzsymbolsPrmtr}}{1.6008ex*\real{\tkzsymbolsPrmtr}}%
1222 \else%

```

```

1223 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
1224 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1]
1225 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
1226 \draw[line width=0.06ex*\tkzsymlbsscl]
1227 (0.28,0) ---++(0.05,0.07) ---++(0.03,0.02) ---+
1228 +(0.03,-0.02) ---++(0.03,0.1) ---++(0.03,0.02) -- (0.5,0.25);
1229 \draw[line width=0.06ex*\tkzsymlbsscl]
1230 (0.28,0) ---++(0.06,-0.02) ---++(0.04,0.06) ---+
1231 +(0.0,-0.08) ---++(0.08,0.06) ---++(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);
1232 \draw[line width=0.06ex*\tkzsymlbsscl]
1233 (0.28,0) ---++(0.03,-0.02) ---++(0.03,-0.07) ---+
1234 +(0.03,-0.01) ---++(0.01,-0.07) ---++(0.06,0.01) ---++(0.03,-0.08) -- (0.5,0.-0.25);
1235 \shade[ball color=#2] (-0.01,0) circle (0.31);
1236 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1237 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0,-0.1);
1238 \end{tikzpicture}%
1239 \fi%
1240 }%
1241 \tikzsymbolsusebox{dWalley#1#2}%
1242 \tikzsymbolsaftersymbolinput%
1243 }
1244 \tkzsymlsDeclareRobustCommandx{drWalley}[2][1=1, 2=yellow, usedefault]{%
1245 \tikzsymbols@ifsaveboxundefined{drWalley#1#2}{%
1246 \set@tkzsymlbsscl{#1}%
1247 \iftikzymbols@draftabsolute%
1248 \tkzsymlsbxPrmtrstore{#1}%
1249 \tikzsymbols@draftbox{2.428ex*\real{\tkzsymlsPrmtr}}{1.6008ex*\real{\tkzsymlsPrmtr}}%
1250 \else%
1251 \ifdim\tkzsymlbsscl<0pt\set@tkzsymlbsscl{-#1}\fi%
1252 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlbsscl,scale=#1,
1253 decoration={random steps,segment length=0.15ex*\tkzsymlbsscl, amplitude=0.1ex*\tkzsymlbsscl}]
1254 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
1255 \draw[decorate, line width=0.06ex*\tkzsymlbsscl] (0.298,0) -- (0.5,0.25);
1256 \draw[decorate,line width=0.06ex*\tkzsymlbsscl] (0.298,0) -- (0.6,0.0);
1257 \draw[decorate,line width=0.06ex*\tkzsymlbsscl] (0.298,0) -- (0.5,-0.25);
1258 \shade[ball color=#2, line width=0.08ex*\tkzsymlbsscl] (-0.01,0) circle (0.31);
1259 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1260 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0,-0.1);
1261 \end{tikzpicture}%
1262 \fi%
1263 }%
1264 \tikzsymbolsusebox{drWalley#1#2}%
1265 \tikzsymbolsaftersymbolinput%
1266 }

```

\Cat *Miau*

```

1267 \tkzsymlsDeclareRobustCommandx{Cat}[1][1=1,usedefault]{%
1268 \tikzsymbols@ifsaveboxundefined{Cat#1}{%
1269 \set@tkzsymlbsscl{#1}%
1270 \iftikzymbols@draftabsolute%

```

```

1271 \tkzsymlsbxPrmtrstore{#1}%
1272 \tikzsymbols@draftbox{1.899ex*\real{\tkzsymlsbxPrmtr}}{1.957ex*\real{\tkzsymlsbxPrmtr}}%
1273 \else%
1274 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{#1}\fi%
1275 \begin{tikzpicture}[x=2.33ex,y=2.33ex, line width=0.093ex*\tkzsymlsscl,scale=#1]
1276 \draw (0,0) circle (0.3);
1277 \draw[rounded corners=0.163ex*\tkzsymlsscl] (-0.3,0) -- (-0.35,0.5) -- (0,0.3);
1278 \draw[rounded corners=0.163ex*\tkzsymlsscl] (0,0.3) -- (0.35,0.5) -- (0.3,0);
1279 \fill (-0.15,.15) circle (0.05);
1280 \fill (0.15,.15) circle (0.05);
1281 \draw[rounded corners=0.175ex*\tkzsymlsscl,yshift=-0.12ex]
1282 (0,0) -- (0,-0.1) -- (-0.1,-0.095);
1283 \draw[rounded corners=0.175ex*\tkzsymlsscl,yshift=-0.12ex]
1284 (0,0) -- (0,-0.1) -- (0.1,-0.095);
1285 \draw[rounded corners=.12ex*\tkzsymlsscl,yshift=-.15ex,
1286 line width=0.03em*\real{0.9}*\tkzsymlsscl]
1287 (-0.1,0.1) -- (0,0) -- (0.1,0.1) -- cycle ;
1288 \draw[line width=0.035ex*\tkzsymlsscl]
1289 (-0.1,-0.05)..controls(-0.25,0)and(-0.35,0).. (-0.4,-0.05);
1290 \draw[line width=0.035ex*\tkzsymlsscl](-0.1,-0.05)..
1291 controls(-0.25,-0.01)and(-0.35,-0.09).. (-0.4,-0.14);
1292 \draw[line width=0.035ex*\tkzsymlsscl](-0.1,-0.05)..
1293 controls(-0.25,-0.045)and(-0.35,-0.13).. (-0.4,-0.23);
1294 \draw[line width=0.035ex*\tkzsymlsscl]
1295 (0.1,-0.05)..controls(0.25,0)and(0.35,0).. (0.4,-0.05);
1296 \draw[line width=0.035ex*\tkzsymlsscl]
1297 (0.1,-0.05)..controls(0.25,-0.01)and(0.35,-0.09).. (0.4,-0.14);
1298 \draw[line width=0.035ex*\tkzsymlsscl]
1299 (0.1,-0.05)..controls(0.25,-0.045)and(0.35,-0.13).. (0.4,-0.23);
1300 \end{tikzpicture}%
1301 \fi%
1302 }%
1303 \tikzsymbolsusebox{Cat#1}%
1304 \tikzsymbolsaftersymbolinput%
1305 }

```

\Ninja \dNinja A Ninja.

```

1306 \tkzsymbolsDeclareRobustCommandx{Ninja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1307 \tikzsymbols@ifsaveboxundefined{Ninja#1#2#3#4}{%
1308 \set@tkzsymlsscl{#1}%
1309 \iftikzymbols@draftabsolute%
1310 \tkzsymlsbxPrmtrstore{#1}%
1311 \tikzsymbols@draftbox{2.149ex*\real{\tkzsymlsbxPrmtr}}{1.717ex*\real{\tkzsymlsbxPrmtr}}%
1312 \else%
1313 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{#1}\fi%
1314 \def\Black@is@Black{black}%
1315 \def\Black@or@not@Black{#2}%
1316 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymlsscl,scale=#1,
1317 decoration={random steps,segment length=0.1ex*\tkzsymlsscl, amplitude=0.1ex*\tkzsymlsscl}]
1318 \fill[#2] (0,0) circle (0.33);

```



```

1319 %\draw (-0.2,-0.125) -- ++(0.4,0);
1320 \fill[decoration={random steps,segment length=0.1ex*\tkzsymbolsscl,
1321 amplitude=0.01ex*\tkzsymbolsscl}, decorate,#3]
1322 (-0.33,0) -- (0.33,0) -- (0.23,0.23) -- (-0.23,0.23) -- cycle;
1323 \ifx\Black@or@not@Black\Black@is@Black
1324 \draw[line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.33);\fi
1325 \fill[#3] (0,0.1) -- (-0.33,0) -- (-0.26,0.23);
1326 \fill[#3] (0.3465,0) arc (0:42:0.34 and 0.345) --
1327 (0.2,0.23)-- (0.31,0.0) -- cycle;
1328 \fill[#3] (-0.3465,0) arc (0:-42:-0.34 and -0.345) --
1329 (-0.2,0.23)-- (-0.31,0.0) -- cycle;
1330 \fill[#4] (0.129,0.1425) arc (55:-180:.05);
1331 \fill[#4] (-0.129,0.1425) arc (-55:180:-.05);
1332 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1333 segment length=0.55ex*\tkzsymbolsscl}, #3]
1334 (0.26,0.21) -- (0.5,0.35);
1335 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1336 segment length=0.55ex*\tkzsymbolsscl}, #3]
1337 (0.26,0.21) -- (0.53,0.1);
1338 \ifx\Black@or@not@Black\Black@is@Black
1339 \else\draw[line width=0.08ex*\tkzsymbolsscl] (0,0) circle (0.33);\fi
1340 \end{tikzpicture}%
1341 \fi%
1342 }%
1343 \tikzsymbolsusebox{Ninja#1#2#3#4}%
1344 \tikzsymbolsaftersymbolinput%
1345 }
1346 \tkzsymbolsDeclareRobustCommand{\dNinja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1347 \tikzsymbols@ifsaveboxundefined{\dNinja#1#2#3#4}{%
1348 \set@tkzsymbolsscl{#1}%
1349 \iftikzymbols@draftabsolute%
1350 \tkzsymbolsbxPrmtrstore{#1}%
1351 \tikzsymbols@draftbox{2.1498ex*\real{\tkzsymbolsPrmtr}}{1.7178ex*\real{\tkzsymbolsPrmtr}}%
1352 \else%
1353 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1354 \def\Black@is@Black{black}%
1355 \def\Black@or@not@Black{#2}%
1356 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbolsscl,scale=#1,
1357 decoration={random steps,segment length=0.1ex*\tkzsymbolsscl, amplitude=0.1ex*\tkzsymbolsscl}]
1358 \draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1359 segment length=0.55ex*\tkzsymbolsscl},decorate, #3!50!black]
1360 (0.26,0.21) -- (0.5,0.35);
1361 \draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymbolsscl,
1362 segment length=0.5ex*\tkzsymbolsscl},decorate, #3!50!black]
1363 (0.26,0.21) -- (0.53,0.1);
1364 \shade[ball color=#2] (0,0) circle (0.347);
1365 %\draw (-0.2,-0.125) -- ++(0.4,0);
1366 \fill[decoration={random steps,segment length=0.1ex*\tkzsymbolsscl,
1367 amplitude=0.01ex*\tkzsymbolsscl},ball color=#3]
1368 decorate {(-0.33,0) -- (0.3465,0) }

```

```

1369      {arc (0:42:0.34 and 0.345)}
1370 decorate  {-- (-0.25,0.24)}
1371      { arc (-42:0:-0.375 and -0.345)};
1372 \shade[ball color=#4] (0.129,0.1425) arc (55:-180:.05);
1373 \shade[ball color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1374 \shade[top color=#4!80!black, bottom color=#4] (0.129,0.1425) arc (55:-180:.05);
1375 \shade[top color=#4!80!black, bottom color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1376 \end{tikzpicture}%
1377 \fi%
1378 }%
1379 \tikzsymbolsusebox{dNinja#1#2#3#4}%
1380 \tikzsymbolsaftersymbolinput%
1381 }

```

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

```

1382 \tikzsymlsDeclareRobustCommandx{NiceReapey}[2][1=1,2={black!20!white},usedefault]{%
1383 \tikzsymbols@ifsaveboxundefined{NiceReapey#1#2}{%
1384 \set@tkzsymlsscl{#1}%
1385 \iftikzsymbols@draftabsolute%
1386 \tkzsymlsbxPrmtrstore{#1}%
1387 \tikzsymbols@draftbox{(1.1067em+0.07ex)*\real{\tkzsymlsPrmtr}}
1388      {(0.693em+0.07ex)*\real{\tkzsymlsPrmtr}}%
1389 \else%
1390 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
1391 \begin{tikzpicture}[x=0.11em,y=0.11em, line width=0.07ex*\tkzsymlsscl,scale=#1]
1392 \draw[] (1.7,-1) arc (360:180:1.7 and 2)
1393      arc (260:110:1.5 and 2) .. controls (-1,3.3) and (1,3.3) .. (1.9,2.97)
1394      arc (260:100:-1.3 and -2) -- cycle;
1395 \fill[#2] (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5) -- cycle;
1396 \draw (3,-3) -- (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5);
1397 \draw (0,-1.5) circle (1 and 0.5);
1398 \draw[line width=0.04ex*\tkzsymlsscl] (-0.2,-1) -- (-0.2,-2);
1399 \draw[line width=0.04ex*\tkzsymlsscl] (0.2,-1) -- (0.2,-2);
1400 \draw[line width=0.04ex*\tkzsymlsscl] (0.6,-1) -- (0.6,-2);
1401 \draw[line width=0.04ex*\tkzsymlsscl] (-0.6,-1) -- (-0.6,-2);
1402 \draw[line width=0.04ex*\tkzsymlsscl] (-1,-1.5) -- (1,-1.5);
1403 \fill (1.25,1.25) circle ( 0.5 and 0.75);
1404 \fill (-1.25,1.25) circle ( 0.5 and 0.75);
1405 \end{tikzpicture}%
1406 \fi%
1407 }%
1408 \tikzsymbolsusebox{NiceReapey#1#2}
1409 \tikzsymbolsaftersymbolinput%
1410 }

```

6.3 Other symbols(s)

\tikzsymbols@Strichmaxerl@XCheck This macro is needed for \Strichmaxerl. It's not easy to explain why it does what it does, but ... uhm ... it has something to do with mathematics and why

the plain vanilla rectangle has always the correct proportions.

It is important for having `\Strichmaxerl`'s plain vanilla rectangle the correct size. Well, at first this macro checks if `#1` is greater than 0. If it is, it checks if `#1` is smaller than 0.18 (radius of the Strichmaxerl's head). If it is, it sets `#1` to 0 (it is smaller than the head-radius and is therefore not needed). If it is not, it subtracts 0.18 from `#1` (we only want the length which overhangs the head). Else ...

```

1411 \newcommand{\tikzsymbols@Strichmaxerl@XCheck}[1]{%
1412 \ifdim #1 pt > 0pt%
1413   \ifdim #1 pt < 0.18pt%
1414     \pgfmathsetmacro{#1}{0}%
1415   \else%
1416     \pgfmathsetmacro{#1}{#1-0.18}%
1417   \fi%
1418 \else%
1419   \ifdim #1 pt > -0.18pt%
1420     \pgfmathsetmacro{#1}{0}%
1421   \else%
1422     \pgfmathsetmacro{#1}{#1+0.18}%
1423   \fi%
1424 \fi%
1425 }
```

`\tikzsymbols@Strichmaxerl@ifSmallerZero` Checks if something is smaller than zero (< 0).

```

1426 \newcommand{\tikzsymbols@Strichmaxerl@ifSmallerZero}[1]{%
1427 \ifdim #1 pt < 0pt%
1428   \pgfmathsetmacro{#1}{0}%
1429 \fi%
1430 }
```

`\Strichmaxerl` My first symbol: a Strichmaxerl. And one of the most complex symbols in this package.

```

1431 \tkzsymlsDeclareRobustCommandx{Strichmaxerl}[5][1=1,2=-22,3=22,4=27,5=-27,usedefault]{%
1432 \tikzsymbols@ifsaveboxundefined{Strichmaxerl#1#2#3#4#5}{%
1433 \set@tkzsymlsscl{#1}%
1434 \iftikzsymbols@draftabsolute%
```

Now we have to calculate the length and the height of the separate parts of the `\Strichmaxerl`. At first the lengths (they have all an `x` in the name).

LA for “linker Arm” (*left arm*).
RA for “rechter Arm” (*right arm*).
LB for “linkes Bein” (*left leg*).
RB for “rechtes Bein” (*right leg*).

```

1435 \pgfmathsetmacro{\tikzsymbols@x@LA}{-0.27*cos(#2)}%
1436 \pgfmathsetmacro{\tikzsymbols@x@RA}{0.27*cos(#3)}%
1437 \pgfmathsetmacro{\tikzsymbols@x@LB}{0.34*sin(#4)}%
1438 \pgfmathsetmacro{\tikzsymbols@x@RB}{0.34*sin(#5)}%
```

Now the height (`y`):

LA for “linker Arm” (*left arm*).

RA for “rechter Arm” (*right arm*).

LB for “linkes Bein” (*left leg*).

RB for “rechtes Bein” (*right leg*).

```
1439 \pgfmathsetmacro{\tikzsymbols@y@LA}{0.27*sin(#2)}%
1440 \pgfmathsetmacro{\tikzsymbols@y@RA}{-0.27*sin(#3)}%
1441 \pgfmathsetmacro{\tikzsymbols@y@LB}{0.34*cos(#4)}%
1442 \pgfmathsetmacro{\tikzsymbols@y@RB}{0.34*cos(#5)}%
```

Well then, lets start our calculations. Firstly the length.

We use the `\tikzsymbols@Strichmaxerl@XCheck` to check if ... (see above).

```
1443 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@LA}%
1444 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@RA}%
1445 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@LB}%
1446 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@RB}%
```

We want the greatest and the smallest length for our rectangle. To evaluate them, we define `\tikzsymbols@x@max` and `\tikzsymbols@x@min` using `tikz max()` and `min()`. The 0 is very important: if for example all lengths are negative, the greatest number is 0. (Sorry, I don't want to explain it, it has something to do with math).

```
1447 \pgfmathsetmacro{\tikzsymbols@x@max}
1448     {max(0,\tikzsymbols@x@LA,\tikzsymbols@x@RA,\tikzsymbols@x@LB,\tikzsymbols@x@RB)}%
1449 \pgfmathsetmacro{\tikzsymbols@x@min}
1450     {min(0,\tikzsymbols@x@LA,\tikzsymbols@x@RA,\tikzsymbols@x@LB,\tikzsymbols@x@RB)}%
```

Finished the length. Now we calculate our height. Arms and legs more or less separate.

Arms: First we subtract 0.2 (= adding -0.2) (torso length)

```
1451 \pgfmathsetmacro{\tikzsymbols@y@LA}{\tikzsymbols@y@LA-0.2}%
1452 \pgfmathsetmacro{\tikzsymbols@y@RA}{\tikzsymbols@y@RA-0.2}%
```

Arms and Legs: if they are smaller than 0, make them 0.

```
1453 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@LA}%
1454 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@RA}%
1455 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@LB}%
1456 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@RB}%
```

And find the greatest number.

```
1457 \pgfmathsetmacro{\tikzsymbols@y@max}
1458 {max(\tikzsymbols@y@LA,\tikzsymbols@y@RA,\tikzsymbols@y@LB,\tikzsymbols@y@RB)}%
```

For the box-length we calculate:

$$(0.606ex + 1.35ex(x=1.35ex) * (\text{greatest (positive) length}) - 1.35 * (\text{smallest (negative) length})) * \text{scale} \quad (1)$$

For the height:

$$(1.173ex + 1.35ex * \text{greatest height}) * \text{scale} \quad (2)$$

```
1459 \tkzsymlsboxPrmtrstore{#1}%
1460 \tikzsymbols@draftbox{(0.606ex+1.35ex*\real{\tikzsymbols@x@max}
1461     -1.35ex*\real{\tikzsymbols@x@min})*\real{\tkzsymlsPrmtr}}%
```

```

1462 {(1.173ex+1.35ex*\real{\tikzsymbols@y@max})*\real{\tikzsymlsPrmtr}}%
1463 \else%
1464 \ifdim\tikzsymlbsscl<Opt\set\tikzsymlbsscl{-#1}\fi%
1465 \begin{tikzpicture}[line width=0.12ex*\tikzsymlbsscl, scale=#1, x=1.35ex, y=1.35ex]
1466 \draw[rotate around={#5:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1467 \draw[rotate around={#4:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1468 \draw (.15,.2) -- (.15,.4);
1469 \draw[rotate around={#3:(.15,.4)}] (0.15,0.4) -- (0.42,0.4);
1470 \draw[rotate around={#2:(.15,.4)}] (0.15,0.4) -- (-0.12,0.4);
1471 \draw (.15,.4) -- (.15,.53);
1472 \draw (.15,.8) circle (0.18);
1473 \end{tikzpicture}%
1474 \fi%
1475 }%
1476 \tikzsymbolsusebox{Strichmaxerl#1#2#3#4#5}%
1477 \tikzsymbolsaftersymbolinput%
1478 }
1479 \tikzsymlsDeclareRobustCommand{Person}{%
1480 \PackageWarning{tikzsymbols}{Command '\protect\Person' is obsolete,
1481 \MessageBreak Please use '\protect\Strichmaxerl' instead.\MessageBreak}%
1482 \Strichmaxerl%
1483 }

```

\Candle A burning candle

```

1484 \tikzsymlsDeclareRobustCommand{Candle}[1][1]{%
1485 \tikzsymbols@ifsaveboxundefined{Candle#1}{%
1486 \set\tikzsymlbsscl{#1}%
1487 \iftikzsymbols@draftabsolute%
1488 \tkzsymlsboxPrmtrstore{#1}%
1489 \tikzsymbols@draftbox{0.64ex*\real{\tikzsymlsPrmtr}}{(1.255ex+2.2pt)*\real{\tikzsymlsPrmtr}}%
1490 \else%
1491 \ifdim\tikzsymlbsscl<Opt\set\tikzsymlbsscl{-#1}\fi%
1492 \begin{tikzpicture}[x=1ex, y=1ex, scale=#1, line width=0.07ex*\tikzsymlbsscl]
1493 \draw[rounded corners=0.04ex*\tikzsymlbsscl] (0,0) -- (0.2,0) -- +(0,1) -- (0,1) -- cycle;
1494 \draw[line width=0.05ex*\tikzsymlbsscl] (0.1,1) -- (0.1,1.2);
1495 \draw[xshift=0.95, yshift=2.2, line width=0.04ex*\tikzsymlbsscl]
1496 (-0.1,0.6) .. controls (-0.4,0.8) and (-0.1,1) .. (-0.1,1.2);
1497 \draw [xshift=0.95, yshift=2.2, line width=0.04ex*\tikzsymlbsscl]
1498 (-0.1,0.6) .. controls (0.2,0.8) and (-0.1,1) .. (-0.1,1.2);
1499 \end{tikzpicture}%
1500 \fi%
1501 }%
1502 \tikzsymbolsusebox{Candle#1}%
1503 \tikzsymbolsaftersymbolinput%
1504 }

```

\Fire Just a fire.

```

1505 \tikzsymlsDeclareRobustCommand{Fire}[1][1]{%
1506 \tikzsymbols@ifsaveboxundefined{Fire#1}{%
1507 \set\tikzsymlbsscl{#1}%

```

```

1508 \iftikzsymbols@draftabsolute%
1509 \tkzsymbolsbxPrmtrstore{#1}%
1510 \tikzsymbols@draftbox{1.576ex*\real{\tkzsymbolsPrmtr}}{1.639ex*\real{\tkzsymbolsPrmtr}}%
1511 \else%
1512 \ifdim\tkzsymbolsscl<0pt\set\tkzsymbolsscl{-#1}\fi%
1513 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.07ex*\tkzsymbolsscl,rotate=45, scale=#1]
1514 \fill (-0.05,0) -- (0.05,0) -- (0.05,0.95) -- (-0.05,0.95) -- cycle;
1515 \fill (-0.74,0.7) -- (0.19,0.7) -- (0.19,0.8) -- (-0.74,0.8) -- cycle;
1516 \fill[rotate=-20, xshift=-1.3, yshift=-0.1]
1517 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1518 \fill[rotate=-70, xshift=-3.3, yshift=-2.3]
1519 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1520 \fill[rotate=135, xshift=2.5, yshift=-3.8]
1521 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1522 \draw[rotate=-45, xshift=-2.6, yshift=1.5,line width=0.04ex*\tkzsymbolsscl, x=0.5ex, y=0.5ex]
1523 (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0.05,1.7);
1524 \draw[rotate=-45, xshift=-2.1,yshift=1.5,line width=0.04ex*\tkzsymbolsscl, x=0.5ex, y=0.5ex]
1525 (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.15,1.7);
1526 \draw[rotate=-45, xshift=-2.5] (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0,1.5);
1527 \draw[rotate=-45, xshift=-2] (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.1,1.5);
1528 \end{tikzpicture}%
1529 \fi%
1530 }%
1531 \tikzsymbolsusebox{Fire#1}%
1532 \tikzsymbolsaftersymbolinput%
1533 }

```

\Coffeecup Just a cup of coffee.

```

1534 \ifKV@tikzsymbols@marvosym \else%
1535 \tkzsymbolsDeclareRobustCommand{\Coffeecup}[1][1]{%
1536 \tikzsymbols@ifsaveboxundefined{\Coffeecup#1}{%
1537 \set\tkzsymbolsscl{#1}%
1538 \iftikzsymbols@draftabsolute%
1539 \tkzsymbolsbxPrmtrstore{#1}%
1540 \tikzsymbols@draftbox{1.82ex*\real{\tkzsymbolsPrmtr}}{1.705ex*\real{\tkzsymbolsPrmtr}}%
1541 \else%
1542 \ifdim\tkzsymbolsscl<0pt\set\tkzsymbolsscl{-#1}\fi%
1543 \begin{tikzpicture}[x=0.7ex,y=0.7ex, scale=#1, line width=0.07ex*\tkzsymbolsscl,
1544 decoration={snake,amplitude=.05ex*\tkzsymbolsscl,segment length=0.408ex*\tkzsymbolsscl}]
1545 \draw (0,0) arc (180:270:0.8 and 1) -- ++(0.5,0) arc (270:360:0.8 and 1) -- cycle;
1546 \draw (2.1,-0.15) -- (2.2,-0.15) arc (90:-90:0.3) -- (1.85, -0.75);
1547 \draw[line width=0.05ex*\tkzsymbolsscl, decorate]
1548 (0.4,0.3) -- +(0,1);
1549 \draw[line width=0.05ex*\tkzsymbolsscl, decorate]
1550 (1,0.3) -- +(0,1);
1551 \draw[line width=0.05ex*\tkzsymbolsscl, decorate]
1552 (1.6,0.3) -- +(0,1);
1553 \draw (0,-1.05) -- (2.1,-1.05);
1554 \end{tikzpicture}%
1555 \fi%

```

```

1556 }%
1557 \tikzsymbolsusebox{Coffeecup#1}%
1558 \tikzsymbolsaftersymbolinput%
1559 }%
1560 \fi

\Chair A chair.
1561 \tikzsymbolsDeclareRobustCommand{Chair}[1][1]{%
1562 \tikzsymbols@ifsaveboxundefined{Chair#1}{%
1563 \set@tkzsymbolsscl{#1}%
1564 \iftikzymbols@draftabsolute%
1565 \tkzsymbolsbxPrmtrstore{#1}%
1566 \tikzsymbols@draftbox{0.97ex*\real{\tkzsymbolsPrmtr}}{1.69ex*\real{\tkzsymbolsPrmtr}}%
1567 \else%
1568 \ifdim\tkzsymbolsscl<Opt\set@tkzsymbolsscl{-#1}\@tkzssmbles@negtrue\fi%
1569 \begin{tikzpicture}[x=0.9ex,y=0.9ex, scale=#1, line width=0.07ex*\tkzsymbolsscl]
1570 \draw (0,-0.5) -- (0,0.7) -- (0.5,1) -- (0.5,0.25);
1571 \draw[line width=0.06ex*\tkzsymbolsscl] (0,0.4) -- (0.5,0.7);
1572 \draw (0,0) -- (0.5,0.3) -- (1,0) --(1,-0.5);
1573 \if@tkzssmbles@neg\draw (0.5,0.3) -- +(0,-0.5);\fi%
1574 \draw (0.5,-0.3) -- (0.5,-0.8);
1575 \draw (1,0) -- (0.5,-0.3) -- (0,0);
1576 \end{tikzpicture}%
1577 \fi%
1578 }%
1579 \tikzsymbolsusebox{Chair#1}%
1580 \@tkzssmbles@negfalse%
1581 \tikzsymbolsaftersymbolinput%
1582 }

\Bed A bed.
1583 \tikzsymbolsDeclareRobustCommand{Bed}[1][1]{%
1584 \tikzsymbols@ifsaveboxundefined{Bed#1}{%
1585 \set@tkzsymbolsscl{#1}%
1586 \iftikzymbols@draftabsolute%
1587 \tkzsymbolsbxPrmtrstore{#1}%
1588 \tikzsymbols@draftbox{3.08ex*\real{\tkzsymbolsPrmtr}}{1.68ex*\real{\tkzsymbolsPrmtr}}%
1589 \else%
1590 \ifdim\tkzsymbolsscl<Opt\set@tkzsymbolsscl{-#1}\fi%
1591 \begin{tikzpicture}[x=1ex,y=1ex, scale=#1, line width=0.08ex*\tkzsymbolsscl]
1592 \draw (0,0) -- (0,1.6);
1593 \draw (3,0) -- (3,1.2);
1594 \draw (0,0.5) -- (3,0.5);
1595 \draw (0,0.35) -- (3,0.35);
1596 \draw (0.7,0.5) arc (0:90:0.7);
1597 \draw (0.7,0.5) arc(180:30:1.231 and 0.6);
1598 \end{tikzpicture}%
1599 \fi%
1600 }%
1601 \tikzsymbolsusebox{Bed#1}%

```

```

1602 \tikzsymbolsaftersymbolinput%
1603 }

```

\Tribar Also called Penrose-Triangle

```

1604 \tikzsymbolsDeclareRobustCommandx{Tribar}[4]
1605 [1=1,2={opacity=0},3={opacity=0},4={opacity=0},usedefault]
1606 {%
1607 \tikzsymbols@ifsaveboxundefined{Tribar#1#2#3#4}{%
1608 \set@tkzsymbolsscl{#1}%
1609 \iftikzymbols@draftabsolute%
1610 \tkzsymbolsbxPrmtrstore{#1}%
1611 \tikzsymbols@draftbox{1.7175ex*\real{\tkzsymbolsPrmtr}}{1.685ex*\real{\tkzsymbolsPrmtr}}}%
1612 \else%
1613 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1614 \begin{tikzpicture}[x=0.65ex,y=0.65ex,scale=#1,
1615   rounded corners=0.03ex*\tkzsymbolsscl, line width=0.06ex*\tkzsymbolsscl]
1616 \fill[#2] (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1617   -- (-0.65,-0.6) -- (0.35,1.3) -- +(0.15,-0.3);
1618 \fill[#3] (0,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- (-0.1,-0.3);
1619 \fill[#4] (1,0) -- (0.35,1.3) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (1.3,0);
1620 \draw (0,0) -- (1,0) -- (0.5,1) -- cycle;
1621 \draw (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1622   -- (-0.65,-0.6) -- (0.35,1.3) -- (0.8,.4);
1623 \draw (0.9,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- +(-0.05,-0.1);
1624 \draw (-0.6,-0.6) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (0.4,1.9);
1625 \end{tikzpicture}%
1626 \fi%
1627 }%
1628 \tikzsymbolsusebox{Tribar#1#2#3#4}%
1629 \tikzsymbolsaftersymbolinput%
1630 }

```

\tikzsymbolsMoaitickness You may already thought it: the line width of the \Moai.

```

1631 \newcommand{\tikzsymbolsMoaitickness}{}

```

\Moai From the Easter Island: a Moai.

```

1632 \tikzsymbolsDeclareRobustCommandx{Moai}[1][1=1,usedefault]{%
1633 \tikzsymbols@ifsaveboxundefined{Moai#1}{%
1634 \set@tkzsymbolsscl{#1}%
1635 \tkzsymbolsbxPrmtrstore{#1}%
1636 \ifdim\tkzsymbolsscl<0pt\set@tkzsymbolsscl{-#1}\fi%
1637 \ifdim\tkzsymbolsscl<2pt%
1638 \def\tikzsymbolsMoaitickness{0.05ex}%
1639 \else%
1640 \ifdim\tkzsymbolsscl<5pt%
1641 \def\tikzsymbolsMoaitickness{0.035ex}%
1642 \else%
1643 \def\tikzsymbolsMoaitickness{0.03ex}%
1644 \fi\fi%

```



```

1645 \iftikzsymbols@draftabsolute%
1646 \tikzsymbols@draftbox{(1.001ex+\tikzsymbolsMoaithickness)*\real{\tkzsymlsPrmtr}}
1647 {(1.664ex+\tikzsymbolsMoaithickness)*\real{\tkzsymlsPrmtr}}}%
1648 \else%
1649 \begin{tikzpicture}[x=.13ex, y=.13ex, rounded corners=0.01ex*\tkzsymlsscl, scale=#1,
1650 line width=\tikzsymbolsMoaithickness*\tkzsymlsscl]
1651 \draw (-2.6,-4.25) -- (-2.5,-5.8)
1652 ..controls (-2,-6.8) and (1.5,-6.8) .. (2.2,-5.8) -- (2.4,-3.95);
1653 \draw(-2.5,2.5) .. controls (-2.9,4.6) and (2,5) .. (3.3,2.5) -- (2.9,-3.4)
1654 .. controls (2,-5) and (-4,-5) .. (-3.1,-3) -- cycle;
1655 \draw (-2.5,3) -- (-2,5) .. controls (0,6) and (2,5.8) .. (3.1,4.7) -- (3.3,2.5);
1656 \draw[line width=0.02ex*\tkzsymlsscl]
1657 (-2.2,-1.8) .. controls (-1,-1.3) and (0,-1.7) .. (1,-2);
1658 \draw[line width=0.02ex*\tkzsymlsscl]
1659 (-2.2,-1.8) .. controls (-1,-1) and (0,-1.4) .. (1,-2);
1660 \draw[line width=0.02ex*\tkzsymlsscl]
1661 (-2.2,-1.8) .. controls (-1,-2) and (0,-2) .. (1,-2);
1662 \draw (-0.8,4) .. controls (-0.8,3) and (-0.8,2) .. (-1.6,0.5) -- (-1.8,-0.4)
1663 .. controls (-1,0.2) and (0,0.2) .. (0.6,-0.4) -- (0.7,0.4)
1664 .. controls (0,1) and (0,2) .. (0.8,4);
1665 \draw (-1.8,-0.36) .. controls (-0.5,-0.5) and (0,-0.5) .. (0.6,-0.36);
1666 \draw (3.2,3.5) -- (3.7,3.5) .. controls (3.5,2) and (3.5,2) .. (3.6,-1.5) -- (3,-1.9);
1667 \draw (-2.5,3) .. controls (-2.7,2) and (-3,1) .. (-2.88,-1);
1668 \draw (-2.5,2.8) .. controls (-2,2.5) and (-1,3) .. (-0.8,3.1);
1669 \draw (0.5,3.3) .. controls (1,3) and (1,2.5) .. (3.3,2.4);
1670 \end{tikzpicture}%
1671 \fi%
1672 }%
1673 \tikzsymbolsusebox{Moai#1}%
1674 \tikzsymbolsaftersymbolinput%
1675 }

```

\Snowman A snowman. I think its smile is scary.

```

1676 \tkzsymlsDeclareRobustCommand{Snowman}[1][1]{%
1677 \tikzsymbols@ifsaveboxundefined{Snowman#1}{%
1678 \set@tkzsymlsscl{#1}%
1679 \iftikzsymbols@draftabsolute%
1680 \tkzsymlsboxPrmtrstore{#1}%
1681 \tikzsymbols@draftbox{1.545ex*\real{\tkzsymlsPrmtr}}{1.772ex*\real{\tkzsymlsPrmtr}}}%
1682 \else%
1683 \ifdim\tkzsymlsscl<0pt\set@tkzsymlsscl{-#1}\fi%
1684 \begin{tikzpicture}[x=0.9ex,y=0.9ex,line width=0.07ex*\tkzsymlsscl, scale=#1]
1685 \draw (0,0) circle (0.4 and 0.35);
1686 \draw[line width=0.06ex*\tkzsymlsscl] (0,0.64) circle (0.3 and 0.28);
1687 \draw[line width=0.05ex*\tkzsymlsscl] (0,1.14) circle (0.2 and 0.2);
1688 \draw[rounded corners=0.1ex*\tkzsymlsscl,line width=0.05ex*\tkzsymlsscl,
1689 rotate around={-30:(0,1.14)}]
1690 (-0.2,1.15) -- ++(0,0.35) -- +(0.4,0) -- (0.2,1.14);
1691 \draw[rounded corners=0.07ex*\tkzsymlsscl,line width=0.05ex*\tkzsymlsscl,
1692 rotate around={-30:(0,1.14)}]

```

```

1693      (-0.2,1.19) arc (270:90:0.1);
1694 \fill (0,0.78) circle (0.04);
1695 \fill (0,0.63) circle (0.04);
1696 \fill (0,0.48) circle (0.04);
1697 \fill (0,0.2) circle (0.05);
1698 \fill (0,0) circle (0.05);
1699 \fill (0,-0.2) circle (0.05);
1700 \fill (-0.06,1.18) circle (0.045);
1701 \fill (0.06,1.18) circle (0.045);
1702 \fill (0.1,1.08) circle (0.015);
1703 \fill (-0.1,1.08) circle (0.015);
1704 \fill (0.06,1.055) circle (0.015);
1705 \fill (-0.06,1.055) circle (0.015);
1706 \fill (0.02,1.039) circle (0.015);
1707 \fill (-0.02,1.039) circle (0.015);
1708 \draw (-0.3,0.7) -- (-0.6,0.8);
1709 \draw (-0.6,0.8) -- (-0.75,0.7);
1710 \draw (-0.6,0.8) -- (-0.55,1);
1711 \draw (-0.6,0.8) -- (-0.8,0.9);
1712 \draw[line width=0.06ex*\tkzsymlbsscl] (-0.65,0) -- (-0.65,1);
1713 \foreach\x in {-0.85, -0.75,-0.65,-0.55,-0.45}
1714 \draw[line width=0.05ex*\tkzsymlbsscl] (-0.65,1) -- (\x,1.3);
1715 \draw (0.3,0.7) -- (0.6,0.8);
1716 \draw (0.6,0.8) -- (0.75,0.7);
1717 \draw (0.6,0.8) -- (0.6,1);
1718 \draw (0.6,0.8) -- (0.8,0.9);
1719 \end{tikzpicture}%
1720 \fi%
1721 }%
1722 \tikzsymbolsusebox{Snowman#1}%
1723 \tikzsymbolsaftersymbolinput%
1724 }

```

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 parameter is “leaf”, if not we check if the last parameter is empty, if not: we generate an error message:

```

1725 \tkzsymlsnewcommand{BasicTree}[5][1]{%
1726 \tikzsymbols@ifsaveboxundefined{BasicTree#1#2#3#4#5}{%
1727 \def\leaf@or@not@leaf{#5}%
1728 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
1729 \Basic@Tree[#1]{#2}{#3}{#4}{#5}%
1730 \else%
1731 \ifx\#5\%
1732 \Basic@Tree[#1]{#2}{#3}{#4}{#5}%
1733 \else%
1734 \PackageError{tikzsymbols}{The last\MessageBreak parameter has either to be \MessageBreak

```

```

1735 ‘leaf’ or has to be empty}{See the tikzsymbols documentation. Section "Trees".}%
1736 \fi\fi%
1737 }%
1738 \tikzsymbolsusebox{BasicTree#1#2#3#4#5}%
1739 \tikzsymbolsaftersymbolinput%
1740 }

```

`\WorstTree` It’s not that bad.

```

1741 \tikzsymbolsDeclareRobustCommand{WorstTree}[1][1]{%
1742 \tikzsymbols@ifsaveboxundefined{WorstTree#1}{%
1743 \set@tkzsymbolsscl{#1}%
1744 \iftikzsymbols@draftabsolute%
1745 \tkzsymbols@draftbox{1.64ex*\real{\tkzsymbolsPrmtr}}{1.84ex*\real{\tkzsymbolsPrmtr}}%
1746 \else%
1747 \ifdim\tkzsymbolsscl<Opt\set@tkzsymbolsscl{-#1}\fi%
1748 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.04ex*\tkzsymbolsscl,scale=#1]
1749 \fill[brown] (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1750 controls (0.5,0.7) and (0.5,0.3) .. (1,0);
1751 \draw (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1752 controls (0.5,0.7) and (0.5,0.3) .. (1,0) ;
1753 \fill[green] (0.2,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1754 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1755 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) ..
1756 controls (1.1, 1.6) and (1,1.4) .. (1,1.2) ..
1757 controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.5,0.8);
1758 \draw (0.214,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1759 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1760 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) .. controls (1.1, 1.6) and (1,1.4) ..
1761 (1,1.2) .. controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.486,0.8);
1762 \fill[red] (0,1) circle (0.1);
1763 \fill[red] (0.4,1.2) circle (0.1);
1764 \fill[red] (0.8,1.1) circle (0.1);
1765 \end{tikzpicture}%
1766 \fi%
1767 }%
1768 }%
1769 \tikzsymbolsusebox{WorstTree#1}%
1770 \tikzsymbolsaftersymbolinput%
1771 }

```

`\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 used in

`\Wintertree` `\BasicTree`.

```

1772 \tikzsymbolsDeclareRobustCommandx{Springtree}[1][1=1, usedefault]%
1773 {\tikzsymbolsuse{BasicTree}[#1]{brown!70!black}{green!90!black}{green!80!black}{leaf}}
1774 \tikzsymbolsDeclareRobustCommandx{Summertree}[1][1=1, usedefault]%
1775 {\tikzsymbolsuse{BasicTree}[#1]{brown!50!black}{green!80!black}{red!80!green}{leaf}}
1776 \tikzsymbolsDeclareRobustCommandx{Autumntree}[1][1=1, usedefault]%

```

```

1777 {\tikzsymbolsuse{BasicTree}[#1]{red!30!black}{red!75!black}{orange}{leaf}}
1778 \tikzsymlsDeclareRobustCommand{\Wintertree}[1][1=1, usedefault]%
1779 {\tikzsymbolsuse{BasicTree}[#1]{black!80!}{black!50}{black!25}{}}

```

Error Message If option `marvosym` is active, but the package not loaded, there will be an error message.

```

1780 \AtBeginDocument{
1781 \ifKV@tikzsymbols@marvosym
1782 \@ifpackageloaded{marvosym}{}{%
1783 \PackageError{tikzsymbols}{Use option 'marvosym' only}{\MessageBreak
1784 if you load package 'marvosym'}
1785 {Either load package 'marvosym' or\MessageBreak
1786 delete the tikzsymbols option 'marvosym'}}
1787 \fi
1788 }

```

Well then, happy T_EXing!

PS: I think the Index and Change History is all right now.

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	B	
\@BackblechlochX 398, 400	\b 198	\Cat 1267
\@BackblechlochY 399, 401	\Basic@Tree 13, 16, 18, 24,	\Chair 1561
\@Ofenschalter 420, 421	26, 28, 45, 47, 49,	\cmdKV@tikzsymbols@prefix 40, 102,
\@Tree@SetUp .. 129, 175	51, <u>138</u> , 1729, 1732	104, 107, 109,
\@gobble 99	\Basic@Tree@ff <u>138</u>	112, 114, 117,
\@ifpackageloaded ..	\Basic@Tree@off ...	120, 123, 125, 127
..... 4–9, 1782 18, 26,	\Cofeecup 1534
\@leaf@is@leaf . <u>56</u> ,	28, 49, 51, 138, 170	\Cooley_\dCooley .. <u>953</u>
145, 154, 188, 1728	\Basic@Tree@on . 13,	\coordinate 197
\@secondoftwo 98	16, 24, 45, 47, <u>168</u>	\csname 83, 87,
\@tkzssmbles@negfalse	\BasicTree <u>1725</u>	93, 102, 104, 107,
..... 204, 1580	\Bed <u>1583</u>	109, 112, 114,
\@tkzssmbles@negtrue	\Black@is@Black 1314,	120, 123, 125, 127
..... 173, 1568	1323, 1338, 1354	\current@color 77
\\ 1731	\Black@or@not@Black	\current@tikzsymbols
 1315, <u>75</u> ,
	1323, 1338, 1355	77, 78, 83, 87, 93
A		\CurrentOption 33
\a 198		
\Annoey_\dAnnoey .. <u>584</u>	C	D
\AtBeginDocument . 1780	\c 199	\d 199
\Autumntree <u>1772</u>	\Candle <u>1484</u>	\def 16, 18, 24,

26, 28, 45, 47, 49, 51, 102, 107, 112, 123, 129, 143, 174, 1314, 1315, 1354, 1355, 1638, 1641, 1643, 1727	86, 91, 92, 102, 103, 107, 108, 112, 113, 123–126	1599, 1613, 1626, 1636, 1644, 1671, 1683, 1720, 1736, 1748, 1767, 1787
\define@boolkey 14, 37, 38	F \f@size 77	\filldraw 393
\define@choicekey 21, 42	\fi 19, 29, 52, 73, 88, 100, 119, 141, 142, 151, 164, 166, 173, 195, 205, 214, 232, 245, 262, 275, 290, 303, 337, 350, 357, 370, 378, 391, 403, 416, 425, 438, 448, 461, 471, 484, 497, 510, 517, 530, 539, 552, 560, 572, 579, 591, 599, 611, 618, 631, 639, 644, 652, 659, 671, 682, 694, 702, 714, 722, 734, 741, 753, 761, 773, 781, 793, 800, 812, 819, 831, 839, 851, 858, 870, 880, 892, 901, 913, 924, 937, 948, 960, 973, 985, 997, 1009, 1022, 1034, 1046, 1059, 1074, 1086, 1100, 1112, 1130, 1142, 1154, 1166, 1186, 1198, 1211, 1223, 1239, 1251, 1262, 1274, 1301, 1313, 1324, 1339, 1341, 1353, 1377, 1390, 1406, 1417, 1423, 1424, 1429, 1464, 1474, 1491, 1500, 1512, 1529, 1542, 1555, 1560, 1568, 1573, 1577, 1590,	\final 14 \Fire 1505 \foreach 398, 399, 420, 1713 \frame 64, 66
\draft 21 \draftabsolute 31		G \global 84
E \edef 73, 77, 141 \else 17, 80, 97, 118, 148, 152, 160, 171, 192, 213, 244, 274, 302, 349, 369, 390, 415, 437, 460, 483, 509, 529, 551, 571, 590, 610, 623, 630, 651, 670, 693, 713, 733, 752, 772, 792, 811, 830, 850, 869, 891, 912, 936, 959, 984, 1008, 1033, 1058, 1085, 1111, 1141, 1165, 1197, 1222, 1250, 1273, 1312, 1339, 1352, 1389, 1415, 1418, 1421, 1463, 1490, 1511, 1534, 1541, 1567, 1589, 1612, 1639, 1642, 1648, 1682, 1730, 1733, 1747		H \hbox 67
\endcsname 78, 83, 87, 93, 102, 104, 107, 109, 112, 114, 117, 120, 123, 126, 127		I \if@tkzssmbls@neg 11, 1573 \ifcase 23, 44 \ifcsname 78, 117 \ifdim . 73, 141, 142, 173, 214, 245, 275, 303, 350, 370, 391, 416, 438, 461, 484, 510, 530, 552, 572, 591, 611, 631, 652, 671, 694, 714, 734, 753, 773, 793, 812, 831, 851, 870, 892, 913, 937, 960, 985, 1009, 1034, 1059, 1086, 1112, 1142, 1166, 1198, 1223, 1251, 1274, 1313, 1353, 1390, 1412, 1413, 1419, 1427, 1464, 1491, 1512, 1542, 1568, 1590, 1613, 1636, 1637, 1640, 1683, 1748
\error_message ... 1780		\ifKV@tikzsymbols@final 15
\expandafter 78, 81, 82, 84–		\ifKV@tikzsymbols@marvosym ... 623, 1534, 1781

<code>\ifKV@tikzsymbols@usebox</code>	N	R
..... 95	<code>\NeedsTeXFormat</code> 1	<code>\real</code> 146, 147,
<code>\iftikzsymbols@draftabsolute</code>	<code>\Neutrey\ldNeutrey</code> . 707	149, 150, 212,
..... 11, 144,	<code>\newcommand</code> 13,	243, 273, 277,
169, 210, 241,	40, 55, 56, 58,	301, 348, 368,
271, 299, 346,	61, 70–72, 75, 76,	389, 414, 436,
366, 387, 412,	90, 101, 106, 111,	459, 482, 508,
434, 457, 480,	113, 116, 122,	528, 550, 570,
506, 526, 548,	1411, 1426, 1631	589, 609, 629,
568, 587, 607,	<code>\newif</code> 11, 12	650, 669, 692,
627, 648, 667,	<code>\newlength</code> ... 57, 59, 60	712, 732, 751,
690, 710, 730,	<code>\newsavebox</code> 81	771, 791, 810,
749, 769, 789,	<code>\NiceReapey</code> 1382	829, 849, 868,
808, 827, 847,	<code>\Ninja\ldNinja</code> ... 1306	890, 911, 935,
866, 888, 909,	<code>\Nursey\ldNursey</code> . 1051	958, 983, 1007,
933, 956, 981,		1032, 1057, 1084,
1005, 1030, 1055,	O	1110, 1140, 1164,
1082, 1108, 1138,	<code>\oldWinkey\ldoldWinkey</code>	1196, 1221, 1249,
1162, 1194, 1219, 746	1272, 1286, 1311,
1247, 1270, 1309,	<code>\or</code> ... 25, 27, 46, 48, 50	1351, 1387, 1388,
1349, 1385, 1434,		1460–1462, 1489,
1487, 1508, 1538,	P	1510, 1540, 1566,
1564, 1586, 1609,	<code>\PackageError</code>	1588, 1611, 1646,
1645, 1679, 1744	.. 118, 1734, 1783	1647, 1681, 1746
<code>\ifx</code> 145,	<code>\PackageWarning</code> .. 1480	<code>\relax</code> 23, 44,
154, 188, 1323,	<code>\PackageWarningNoLine</code>	54, 79, 96, 118, 623
1338, 1728, 1731 33	<code>\renewcommand</code> ... 98, 99
<code>\Innocey</code> 929	<code>\Person</code> 1480	<code>\RequirePackage</code> ... 4–9
<code>\Innocey\ldInnocey</code> . 906	<code>\pgfarrowsdeclare</code> .. 176	
	<code>\pgfarrowslefttextend</code>	S
K 177	<code>\Sadey\ldSadey</code> 545
<code>\KV@tikzsymbols@useboxtrue</code>	<code>\pgfarrowsrighttextend</code>	<code>\sbox</code> 85
..... 39 177	<code>\set@tkzsymlsscl</code> ..
	<code>\pgflowlevel</code> 196	.. 58, 139, 142,
L	<code>\pgfmathsetmacro</code> 140,	172, 173, 209,
<code>\Laughey\ldLaughey</code> . 664	1414, 1416, 1420,	214, 240, 245,
<code>\leaf@or@not@leaf</code> ..	1422, 1428, 1435–	270, 275, 298,
. 143, 145, 154,	1442, 1447, 1449,	303, 345, 350,
174, 188, 1727, 1728	1451, 1452, 1457	365, 370, 386,
<code>\leafcolor</code> 200	<code>\pgfpatharc</code> ... 180, 181	391, 411, 416,
<code>\let</code> 124	<code>\pgfpathmoveto</code> 179	433, 438, 456,
	<code>\pgfpoint</code> 179	461, 479, 484,
M	<code>\pgftransformscale</code> . 196	505, 510, 525,
<code>\marvosym</code> 37	<code>\pgfusepathqfill</code> .. 182	530, 547, 552,
<code>\maxdimen</code> 65	<code>\prefix</code> 40	567, 572, 586,
<code>\MessageBreak</code>	<code>\ProcessOptionsX</code> .. 54	591, 606, 611,
. 34, 118, 1481,	<code>\ProcessOptionsX*</code> .. 54	626, 631, 647,
1734, 1783, 1785	<code>\protect</code> 1480, 1481	652, 666, 671,
<code>\Moai</code> 1632	<code>\ProvidesPackage</code> ... 2	689, 694, 709,
		714, 729, 734,

748, 753, 768,	1489, 1510, 1540, 43, 44
773, 788, 793,	1566, 1588, 1611,	\tikzsymbols@tree@val
807, 812, 826,	1646, 1681, 1746 43
831, 846, 851,	\tikzsymbols@draftboxheight	\tikzsymbols@x@LA ..
865, 870, 887, 59, 63, 66 1435,
892, 908, 913,	\tikzsymbols@draftboxlength	1443, 1448, 1450
932, 937, 955, 59, 62, 67	\tikzsymbols@x@LB ..
960, 980, 985,	\tikzsymbols@draftQbox 1437,
1004, 1009, 1029, 70,	1445, 1448, 1450
1034, 1054, 1059,	550, 570, 589,	\tikzsymbols@x@max .
1081, 1086, 1107,	609, 629, 650, 1447, 1460
1112, 1137, 1142,	669, 692, 712,	\tikzsymbols@x@min .
1161, 1166, 1193,	732, 751, 771, 1449, 1461
1198, 1218, 1223,	791, 810, 829,	\tikzsymbols@x@RA ..
1246, 1251, 1269,	849, 868, 890, 1436,
1274, 1308, 1313,	958, 983, 1007, 1032	1444, 1448, 1450
1348, 1353, 1384,	\tikzsymbols@ifsaveboxundefined	\tikzsymbols@x@RB ..
1390, 1433, 1464, 76, 1438,
1486, 1491, 1507,	98, 208, 239, 269,	1446, 1448, 1450
1512, 1537, 1542,	297, 344, 364,	\tikzsymbols@y@LA ..
1563, 1568, 1585,	385, 410, 432, 1439,
1590, 1608, 1613,	455, 478, 504,	1451, 1453, 1458
1634, 1636, 1678,	524, 546, 566,	\tikzsymbols@y@LB ..
1683, 1743, 1748	585, 605, 625,	.. 1441, 1455, 1458
\setlength ... 58, 62, 63	646, 665, 688,	\tikzsymbols@y@max .
\Sey_\dSey 824	708, 728, 747, 1457, 1462
\Smiley_\dSmiley .. 623	767, 787, 806,	\tikzsymbols@y@RA ..
\Snowman 1676	825, 845, 864, 1440,
\Springtree 1772	886, 907, 931,	1452, 1454, 1458
\Strichmaxerl 1431	954, 979, 1003,	\tikzsymbols@y@RB ..
\Summertree 1772	1028, 1053, 1080,	.. 1442, 1456, 1458
	1106, 1136, 1160,	\tikzsymbolsaftersymbolinput
T	1192, 1217, 1245, 55,
\tikzset 129	1268, 1307, 1347,	235, 265, 293,
\tikzsymbols@draft@nr	1383, 1432, 1485,	340, 360, 381,
..... 22, 23	1506, 1536, 1562,	406, 428, 451,
\tikzsymbols@draft@val	1584, 1607, 1633,	474, 500, 520,
..... 22	1677, 1726, 1742	542, 563, 582,
\tikzsymbols@draftbox	\tikzsymbols@let 122,	602, 621, 642,
..... 61,	237, 267, 295,	662, 685, 705,
70, 146, 149, 212,	342, 362, 383,	725, 744, 764,
243, 273, 301,	408, 430, 453,	784, 803, 822,
348, 368, 389,	476, 502, 522, 544	842, 861, 883,
414, 436, 459,	\tikzsymbols@Strichmaxerl@ifSmallOrder	927, 951,
482, 508, 528,	.. 1426, 1453-1456	976, 1000, 1025,
911, 935, 1057,	\tikzsymbols@Strichmaxerl@XCheck	1049, 1077, 1103,
1084, 1110, 1140,	.. 1411, 1443-1446	1133, 1157, 1189,
1164, 1196, 1221,	\tikzsymbols@Tree@absolute@scale	1214, 1242, 1265,
1249, 1272, 1311, 140, 141	1304, 1344, 1380,
1351, 1387, 1460,	\tikzsymbols@tree@nr	1409, 1477, 1503,

1532, 1558, 1581,	1248, 1271, 1310,	1249, 1272, 1311,
1602, 1629, 1674,	1350, 1386, 1459,	1351, 1387, 1388,
1723, 1739, 1770	1488, 1509, 1539,	1461, 1462, 1489,
<code>\tikzsymbolsMoaithickness</code>	1565, 1587, 1610,	1510, 1540, 1566,
..... 1631,	1635, 1680, 1745	1588, 1611, 1646,
1638, 1641, 1643,	<code>\tkzsymbolsDeclareRobustCommand</code>	1647, 1681, 1746
1646, 1647, 1650 101, 207,	<code>\tkzsymbolsscl</code> .. 57,
<code>\tikzsymbolsuse</code> ...	238, 268, 296,	73, 141, 142,
.... 116, 1773,	343, 363, 384,	153, 173, 189,
1775, 1777, 1779	409, 431, 454,	193, 214–216,
<code>\tikzsymbolsusebox</code> .	477, 503, 523,	222, 245–248,
..... 90,	929, 1479, 1484,	250, 252, 253,
99, 234, 264, 292,	1505, 1535, 1561,	255, 257, 259,
339, 359, 380,	1583, 1676, 1741	275, 277, 287,
405, 427, 450,	<code>\tkzsymbolsDeclareRobustCommandx</code>	303–305, 350–
473, 499, 519, 106, 545,	353, 370–372,
541, 562, 581,	565, 584, 604,	375, 391–394,
601, 620, 641,	624, 645, 664,	396, 416, 417,
661, 684, 704,	687, 707, 727,	423, 438–440,
724, 743, 763,	746, 766, 786,	444, 445, 461–
783, 802, 821,	805, 824, 844,	463, 484–486,
841, 860, 882,	863, 885, 906,	489–491, 493–
903, 926, 950,	930, 953, 978,	495, 510–512,
975, 999, 1024,	1002, 1027, 1051,	530–532, 536,
1048, 1076, 1102,	1079, 1105, 1135,	537, 552–555,
1132, 1156, 1188,	1159, 1191, 1216,	572, 573, 591–
1213, 1241, 1264,	1244, 1267, 1306,	594, 611, 612,
1303, 1343, 1379,	1346, 1382, 1431,	631, 632, 652,
1408, 1476, 1502,	1604, 1632, 1772,	653, 671–674,
1531, 1557, 1579,	1774, 1776, 1778	677, 679, 694–
1601, 1628, 1673,	<code>\tkzsymbolsnewcommand</code>	696, 699, 714–
1722, 1738, 1769 111, 1725	717, 734, 735,
<code>\tikzsymbols@draftabsolutet</code>	<code>\tkzsymbolsPrmtr</code> ...	753, 754, 773,
..... 28, 36 71, 141,	774, 793, 794,
<code>\tksymbolsbxPrmtrstore</code>	146, 147, 149,	812, 813, 831–
.. 71, 211, 242,	150, 212, 243,	834, 851, 852,
272, 300, 347,	273, 301, 348,	870–873, 892–
367, 388, 413,	368, 389, 414,	894, 913, 914,
435, 458, 481,	436, 459, 482,	920–922, 937,
507, 527, 549,	508, 528, 550,	938, 943, 945,
569, 588, 608,	570, 589, 609,	946, 960, 961,
628, 649, 668,	629, 650, 669,	964, 966, 985,
691, 711, 731,	692, 712, 732,	986, 991, 993,
750, 770, 790,	751, 771, 791,	1009, 1010, 1015,
809, 828, 848,	810, 829, 849,	1017, 1019, 1034,
867, 889, 910,	868, 890, 911,	1035, 1039, 1041,
934, 957, 982,	935, 958, 983,	1043, 1059–1061,
1006, 1031, 1056,	1007, 1032, 1057,	1067, 1069, 1071,
1083, 1109, 1139,	1084, 1110, 1140,	1072, 1086, 1087,
1163, 1195, 1220,	1164, 1196, 1221,	1091, 1093, 1095,

1097, 1112–	1390, 1391, 1398–	U
1116, 1142–1144,	1402, 1464, 1465,	<code>\usebox</code> 38, 91
1146, 1166–1170,	1491–1495, 1497,	<code>\usetikzlibrary</code> ... 10
1173, 1176, 1179,	1512, 1513, 1522,	
1198–1202, 1205–	1524, 1542–1544,	V
1207, 1223, 1224,	1547, 1549, 1551,	<code>\vbadness</code> 65
1226, 1229, 1232,	1568, 1569, 1571,	<code>\vbox</code> 66
1251–1253, 1255–	1590, 1591, 1613,	<code>\Vomey_\dVomey</code> ... 1105
1258, 1274, 1275,	1615, 1636, 1637,	
1277, 1278, 1281,	1640, 1649, 1650,	W
1283, 1285, 1286,	1656, 1658, 1660,	<code>\Walley_\dWalley</code> . 1159
1288, 1290, 1292,	1683, 1684, 1686–	<code>\Winkey_\dWinkey</code> .. 746
1294, 1296, 1298,	1688, 1691, 1712,	<code>\Wintertree</code> 1772
1313, 1316, 1317,	1714, 1748, 1749	<code>\WorstTree</code> 1741
1320, 1321, 1324,	<code>\Tongey_\dTongey</code> . 1002	
1332, 1333, 1335,	<code>\tree</code> 42	X
1336, 1339, 1353,	<code>\Tribar</code> 1604	<code>\x</code> 1713, 1714
1356–1359, 1361,	<code>\typeout</code> 32, 35	<code>\Xey_\dXey</code> 863
1362, 1366, 1367,		

Change History

v1.0	General: Initial version 1	didn't improve the source-description 1
v1.05	General: Deleted a “t” in the BasicTree-code, shortened the trunk from the tree a bit, renamed some codes, made an index 1	v1.7 General: New symbols, etc. 1
v1.6	General: on/off. 1	v2.0 General: Fixed Bugs, improved BasicTree, new option “marvosym”, new symbol 1
	Renamed “tikzsymbolsaftersymbolinput” to “tikzsymbolsaftersymbolinput” 1	v2.2 General: Now you can use negative scaling. Include <code>\@ifpackageloaded</code> . Did something else, I can't remember ... 1
	Now “Person” can be used in sections, etc. 1	v2.5 General: New option: draftabsolute, changed the documentation a bit 1
	Now an error message is generated if the last parameter of “BasicTree” is neither “leaf” nor empty. 1	v3.0 General: Changed output of “absolute” option 1
v1.61	General: Made an invisible box in BasicTree. 1	Changed symbol code 1
v1.65	General: Improved BasicTree; New symbols “Schaler/peeler”, Laughey, Walley, Ninja; but	Changed the documentation ... 1
		Replaced <code>\let</code> by <code>\tikzsymbols@let</code> 1
		Using <code>\changes</code> correctly (hopefully) 1

marvosym: using xkeaval's boolkey .	11	\tikzsymbols@draftboxlength:	
\Autumntree: Replaced \BasicTree		New length	12
by \tikzsymbolsuse{BasicTree}		\tikzsymbols@draftQbox: New	
.	51	macro. Short form of	
\Basic@Tree: has now the default		\tikzsymbols@draftbox . . .	13
definition \Basic@Tree@on . . .	10	\tikzsymbols@ifsaveboxundefined:	
\Basic@Tree@off: Draft rectangle		New macro. Checks if save-box	
same size as normal tree.		is undefined, if true: does noth-	
Changed parts of the code . . .	15	ing; if false: defined new box. .	13
\Basic@Tree@on: Same here . . .	16	\tikzsymbols@let: New macro.	
\cmdKV@tikzsymbols@prefix: New		\let with prefix	15
macro.	11	\tikzsymbols@Strichmaxerl@ifSmallerZero:	
\Cofeecup: Changed angle of arc a		New macro. Is needed for	
bit. Changed a length.	46	\Strichmaxerl. Checks if some-	
\current@tikzsymbols: New		thing is smaller than 0	43
macro. Current script size and		\tikzsymbols@Strichmaxerl@XCheck:	
text color is stored inside . . .	13	New macro. Is needed for	
\iftikzymbols@draftabsolute:		\Strichmaxerl	42
New \newif	10	\tikzsymbolsuse: New macro. Us-	
final: using xkeaval's boolkey . . .	11	ing symbols without carrying	
using xkeaval's choicekey	11	about the prefix	14
\oldWinkey_\olddWinkey: Old		\tikzsymbolsusebox: New macro.	
Winkey	29	Prints the savebox	14
draft: using xkeaval's choicekey.		\tkzsymlsboxPrmtrstore: New	
Furthermore new option abso-		macro. Makes \tkzsymlsPrmtr	
lute is available	11	positive (if it is negative)	13
draftabsolute: is obsolete	11	\tkzsymlsDeclareRobustCommand:	
tree: Using xkeyval's choicekey. Can		New macro. Is needed for option	
be set: on/true, off/false	12	prefix	14
usebox: New option	11	\tkzsymlsDeclareRobustCommandx:	
\Springtree: Replaced \BasicTree		New macro. Is needed for option	
by \tikzsymbolsuse{BasicTree}		prefix	14
.	51	\tkzsymlsnewcommand: New	
\Strichmaxerl: Renamed \Person		macro. Is needed for option	
to \Strichmaxerl and added		prefix	14
great parts of code	43	\tkzsymlsPrmtr: New macro.	
\Summertree: Replaced \BasicTree		Needed for the plain vanilla	
by \tikzsymbolsuse{BasicTree}		rectangles	13
.	51	\Winkey_\dWinkey: Changed the	
\tikzsymbols@draftbox: New		smile. Old Winkey available as	
macro. Draw a plain vanilla		\oldWinkey and \olddWinkey . . .	29
rectangle if draft is absolute . .	12	\Wintertree: Replaced \BasicTree	
\tikzsymbols@draftboxheight:		by \tikzsymbolsuse{BasicTree}	
New length	12	51