

tikzsymbols*

Ben Vitecek
b.vitecek@gmx.at

October 2, 2015

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= <i>⟨prefix⟩</i>	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 v3.0g, dated 2015/10/01.

6	Code	10
6.1	Cookingsymbolcode	17
6.2	Emoticonscode	24
6.3	Other symbols(s)	42
6.4	Trees	49

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. You may use `draft=absolute`, which is – I think – more useful.

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 tikzsymbols. 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 again⁴ (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.

³Inside a `\sbox` to be correctly.

⁴I think.

For example: `\tikzsymbolsuse{Smiley}[2]` ☺
`\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 box⁵ 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 , $\omin�$, $\omin�$ and $\omin�$ 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 calculation⁶ (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.
















⁵To say the truth: I am (still) not sure what exactly happens while storing commands inside `\sbox` and using them via `\usebox`...

⁶And 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.

$\langle scale \rangle$ can be a number between (not exactly) -1400 and (also not exactly) 1400⁷, 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
<code>\Kochtopf</code>	<code>\pot</code>	$[\langle scale \rangle]$	
<code>\Bratpfanne</code>	<code>\fryingpan</code>	$[\langle scale \rangle]$	
<code>\Schneebeesen</code>	<code>\eggbeater</code>	$[\langle scale \rangle]$	
<code>\Sieb</code>	<code>\sieve</code>	$[\langle scale \rangle]$	
<code>\Purierstab</code>	<code>\blender</code> ⁸	$[\langle scale \rangle]$	
<code>\Dreizack</code>	<code>\trident</code>	$[\langle scale \rangle]$	
<code>\Backblech</code>	<code>\bakingplate</code>	$[\langle scale \rangle]$	
<code>\Ofen</code>	<code>\oven</code>	$[\langle scale \rangle]$	
<code>\Pfanne</code>	<code>\pan</code>	$[\langle scale \rangle]$	
<code>\Herd</code>	<code>\cooker</code>	$[\langle scale \rangle]$	
<code>\Saftpresse</code>	<code>\squeezer</code>	$[\langle scale \rangle]$	
<code>\Schussel</code>	<code>\bowl</code>	$[\langle scale \rangle]$	
<code>\Schaler</code>	<code>\peeler</code>	$[\langle scale \rangle]$	
<code>\Reibe</code>	<code>\grater</code>	$[\langle scale \rangle]$	
<code>\Flasche</code>	<code>\bottle</code>	$[\langle scale \rangle]$	

3.2 Emoticons ☺

3.2.1 “normal” Emoticons 🐼

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

$\langle scale \rangle$ can be a number between (not exactly) -2000 and (not exactly) 2000⁹, default is 1.

$\langle color \rangle$ can be every defined color. Note: The color names shouldn't contain special characters like ß, ä, ö, ...

Commands	Optional parameter(s)	Output
<code>\Smiley</code>	$[\langle scale \rangle] [\langle color \rangle]$	☺
<code>\Sadey</code>	$[\langle scale \rangle] [\langle color \rangle]$	☹
<code>\Laughey</code>	$[\langle scale \rangle] [\langle color \rangle] [\langle mouth color \rangle]$	😄
<code>\Annoey</code>	$[\langle scale \rangle] [\langle color \rangle]$	😡

⁷Since version 2.2 you can use negative numbers as well (see examples)

⁸I know that “Pürierstab” should be translated as “immersion blender”, but I'm just using “blender”

⁹Do you even need so large symbols?

Commands	Optional parameter(s)	Output
<code>\Neutrey</code>	<code>[\scale][\color]</code>	☹
<code>\Winkey</code>	<code>[\scale][\color]</code>	☺
<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>\rWalley¹⁰</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 number¹¹ and a large number¹², 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>	☹

¹⁰“r” stands for “random” and means that the cracks in the wall are generated randomly, but it takes some time to generate them.

¹¹under -500 for sure

¹²over 500 for sure

Commands	Optional parameter(s)	Output
<code>\dWinkey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>	
<code>\olddWinkey</code>	<code>[\langle scale \rangle] [\langle color \rangle]</code>	
<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> ¹³	<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 (`\langle left arm \rangle` to `\langle right leg \rangle`) can be a number between -360 and 360¹⁴. These parameters are the angles between the body and the separate parts of `\Strichmaxerl` (see examples).

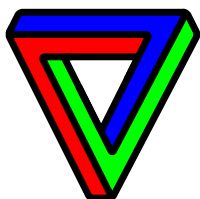
`\langle scale \rangle` can be a very great and a very small number (but I don't think, that you need so large symbols).

`\langle color \rangle` 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>	

¹³“r” stands for “random” and means that the cracks in the wall are generated randomly, but it takes some time.

¹⁴Of 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 ¹⁵, 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}}` 

¹⁵if 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{\enquote{3D} Emoticons \texorpdfstring{\dSmiley}{dSmiley}}
```

or

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

or

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

4.2 ...and errors

Make sure you load `marvosym` *before* `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`).

5 Nobody is perfect

If you find a bug, please send me a mail involving a *minimal example* which shows the bug and a short description. Please mention the `tikzsymbols` in the header, “gmx” has a habit of putting mails in the spam-folder and it helps me to recognize those mails faster. This can also be the reason why I may need some time to answer the mail.

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 [2015/10/01 v3.0g 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}}
```

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

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

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

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

```
10 \newif\if@tkzssmbles@neg
11 \newif\iftikzymbols@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.
```

```
12 \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.

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

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.

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

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

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

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

```
36 \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`)

```
37 \define@boolkey{tikzsymbols}{usebox}[true]{}
38 \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.

```

39 \newcommand{\cmdKV@tikzsymbols@prefix}{}
40 \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.
41 \define@choicekey{tikzsymbols}{tree}%
42 [\tikzsymbols@tree@val\tikzsymbols@tree@nr]{true,on,false,off}[on]{%
43   \ifcase\tikzsymbols@tree@nr\relax
44     \def\Basic@Tree{\Basic@Tree@on}% tree=true
45   \or
46     \def\Basic@Tree{\Basic@Tree@on}% tree=on
47   \or
48     \def\Basic@Tree{\Basic@Tree@off}% tree=false
49   \or
50     \def\Basic@Tree{\Basic@Tree@off}% tree=false
51   \fi
52 }

\ProcessOptionsX* We process all options. What is \relax doing?
53 \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.
54 \newcommand*{\tikzsymbolsaftersymbolinput}{\xspace}

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

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

\tikzsymbols@draftboxlength The length and the height of the plain vanilla rectangle are stored inside these
\tikzsymbols@draftboxheight lengths.
58 \newlength{\tikzsymbols@draftboxlength}
59 \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.
60 \newcommand*{\tikzsymbols@draftbox}[2]{%
61   \pgfmathsetlength{\tikzsymbols@draftboxlength}{#1}%

```

```

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

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

\tikzsymls@Prmtr \tikzsymbols@draftbox doesn't like negative numbers. So to be sure that only
\tksymls@bx@Prmtrstore positive numbers are used, \tkzsymls@bx@Prmtrstore changes \tkzsymls@Prmtr
to be positive (if it is negative). I am using \tkzsymls@Prmtr instead of #1 (scal-
ing) in \tikzsymbols@draftbox
70 \newcommand*\tkzsymls@Prmtr{}
71 \newcommand*\tkzsymls@bx@Prmtrstore[1]{%
72   \edef\tkzsymls@Prmtr{\ifdim\tkzsymls@scl<0pt-\fi#1}%
73 }

\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.
74 \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.
75 \newcommand*{\tikzsymbols@ifsaveboxundefined}[2]{%
76   \edef\current@tikzsymbols{\current@color\f@size}%
77   \expandafter\ifcsname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname%
78   \relax%
79   \else%
80   \expandafter\expandafter\expandafter\newsavebox%
81   \expandafter\expandafter\expandafter{%
82     \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}%
83   \expandafter\expandafter\expandafter\global%
84   \expandafter\expandafter\expandafter\sbox%
85   \expandafter\expandafter\expandafter{%
86     \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}{#2}%

```

```

87 \fi%
88 }

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

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

\tikzsymbols@Declare@Robust@Command 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.
100 \newcommand{\tkzsymls@Declare@Robust@Command}[1]{%
101 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
102 \expandafter\DeclareRobustCommand%
103 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
104 }

\tikzsymls@Declare@Robust@Commandx Same as before
105 \newcommand{\tkzsymls@Declare@Robust@Commandx}[1]{%
106 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
107 \expandafter\DeclareRobustCommandx%
108 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
109 }

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

\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.

```

```

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

\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
121 \newcommand{\tikzsymbols@let}[2]{%
122   \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
123     \expandafter\expandafter\expandafter\let%
124     \expandafter\csname\expandafter\cmdKV@tikzsymbols@prefix%
125     \expandafter#1\expandafter\endcsname%
126     \csname\cmdKV@tikzsymbols@prefix#2\endcsname%
127 }

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

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

```

```

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

```

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

```

167 \DeclareRobustCommandx{\Basic@Tree@on}[5][1=1, usedefault]{%
168   \iftikzymbols@draftabsolute%
169     \Basic@Tree@off[#1]{#2}{#3}{#4}{#5}%
170   \else%
171     \set@tkzsymlsscl{#1}%
172     \ifdim\tkzsymls@scl<0pt \set@tkzsymlsscl{-#1}\@tkzssmbles@negtrue\fi%
173     \def\tikzsymbols@leaf@or@not@leaf{#5}%
174     \@Tree@SetUp%
175     \colorlet{trunk}{#2}%
176     \colorlet{leaf a}{#3}%
177     \colorlet{leaf b}{#4}%
178     \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
179       \pgfarrowsdeclare{leaf}{leaf}%
180       {\pgfarrowslefttextend{-0.1ex}\pgfarrowsrighttextend{-0.05ex}}%
181       {%
182         \pgfpathmoveto{\pgfpoint{-0.01ex}{0ex}}%
183         \pgfpatharc{150}{30}{0.08ex}%
184         \pgfpatharc{-30}{-150}{0.08ex}%
185         \pgfusepathqfill%
186       }%
187       \ifx\tikzsymbols@leaf@or@not@leaf\tikzsymbols@leaf@is@leaf%
188       \draw[transparent,scale=#1+0.02ex, line width=0.4pt*\tkzsymls@scl]
189         (-0.8386,0+0.2pt) rectangle
190         (0.8386, 1.42);
191       \else %
192       \draw[transparent,scale=#1+0.02ex, line width=0.4pt*\tkzsymls@scl]
193         (-0.6998,0+0.2pt) rectangle (0.6998,0.68+0.6);
194       \fi
195       \pgflevel{\pgftransformscale{#1+0.02ex}}{%
196         \coordinate (root) [grow cyclic,rotate=90] child {
197           child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
198             child foreach \c in {0,1,2} { child foreach \d in {0,1} {
199               child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafco
200                 }}}} edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]

```



```

201     };}%
202 \end{tikzpicture}%
203 \@tkzssmbles@negfalse%
204 \fi%
205 }}

```

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:

```

206 \tkzsymbols@Declare@Robust@Command{Kochtopf}[1][1]{%
207 \tikzsymbols@ifsaveboxundefined{Kochtopf#1}{%
208 \set@tkzsymbolsscl{#1}%
209 \iftikzymbols@draftabsolute%
210 \tkzsymbols@bx@Prmtrstore{#1}%
211 \tikzsymbols@draftbox{2.47ex*\tkzsymbols@Prmtr}{1.577ex*\tkzsymbols@Prmtr}%
212 \else%
213 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
214 \begin{tikzpicture}[x=2ex,y=2.2ex, line width=0.07ex*\tkzsymbols@scl,scale=#1]
215   \draw[rounded corners=0.2ex*\tkzsymbols@scl] (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*\tkzsymbols@scl,segment length=0.93ex*\tkzsymbols@scl}]
222     (0,0.35) -- (1,0.35);
223   \draw (0.45,0.1) circle (0.04);
224   \draw (0.7,0.11) circle (0.04);
225   \draw (0.13, 0.125) circle (0.04);
226   \draw (0.3,0.2) circle (0.04);
227   \draw (0.88,0.2) circle (0.04);
228   \draw (0.1,0.25) circle (0.04);
229   \draw (0.6,0.25) circle (0.04);
230 \end{tikzpicture}%
231 \fi%
232 }%
233 \tikzsymbols@use@box{Kochtopf#1}%
234 \tikzsymbols@ftersymbolinput%
235 }
236 \tikzsymbols@let{pot}{Kochtopf}

```

\Bratpfanne,\fryingpan If you wonder why I am using line width=0.07ex*\tkzsymbols@scl 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 `\tkzsymls@scl` because it is always positive.

```

237 \tkzsymls@Declare@Robust@Command{Bratpfanne}[1][1]{%
238   \tikzsymbols@ifsaveboxundefined{Bratpfanne#1}{%
239     \set@tkzsymls@scl{#1}%
240     \iftikzymbols@draftabsolute%
241       \tkzsymls@bx@Prmtrstore{#1}%
242       \tikzsymbols@draftbox{3.5535ex*\tkzsymls@Prmtr}{1.4525ex*\tkzsymls@Prmtr}%
243     \else%
244       \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
245       \begin{tikzpicture}[x=0.7ex,y=1.4ex, line width=0.07ex*\tkzsymls@scl, scale=#1,
246         decoration={snake,amplitude=.05ex*\tkzsymls@scl,
247         segment length=0.408ex*\tkzsymls@scl}]
248         \draw[rounded corners=0.07ex*\tkzsymls@scl]
249           (-1,0) -- (1,0) -- (1.5,0.4) -- (-1.5,0.4) -- cycle;
250         \draw[line width=0.037ex*\tkzsymls@scl, rounded corners=0.023ex*\tkzsymls@scl]
251           (-1.4,0.3) -- (-3.5,0.3) -- (-3.5,0.25) -- (-1.3,0.25);
252         \draw[line width=0.023ex*\tkzsymls@scl] (-1.1,0.1) -- (1.1,0.1);
253         \foreach \tikzsymbols@x in { -0.3, 0.3, -1 , 1 }
254           \draw[line width=0.035ex*\tkzsymls@scl, decorate] (\tikzsymbols@x,0.5) -- (\tikzsymbol
255           \end{tikzpicture}%
256       \fi%
257   }%
258 \tikzsymbols@use@box{Bratpfanne#1}%
259 \tikzsymbols@after@symbol@input%
260 }
261 \tikzsymbols@let{fryingpan}{Bratpfanne}

```

`\Schneebesen,\eggbeater` The next one:

```

262 \tkzsymls@Declare@Robust@Command{Schneebesen}[1][1]{%
263   \tikzsymbols@ifsaveboxundefined{Schneebesen#1}{%
264     \set@tkzsymls@scl{#1}%
265     \iftikzymbols@draftabsolute%
266       \tkzsymls@bx@Prmtrstore{#1}%
267       \tikzsymbols@draftbox{0.5697ex*\tkzsymls@Prmtr}{1.57985ex*\tkzsymls@Prmtr}%
268     \else%
269       \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
270       \begin{tikzpicture}[y=2.1ex,x=1.4ex, scale=#1,
271         line width=0.01ex*\tkzsymls@scl*0.97]
272         \foreach \x in { -0.2 , -0.15, -0.1, -0.05, 0, 0.05, 0.1, 0.15, 0.2 }
273           \draw (0,0) .. controls (\x,0.0) and (\x,0.2) .. (0,0.4);
274         \fill[line width=0.05ex*\tkzsymls@scl, rounded corners=0.07ex*\tkzsymls@scl]
275           (-0.05,0.37) -- (0.05,0.37) -- (0.05,0.75) -- (-0.05,0.75) -- cycle;
276       \end{tikzpicture}%
277     \fi%
278   }%
279 \tikzsymbols@use@box{Schneebesen#1}%
280 \tikzsymbols@after@symbol@input%
281 }

```

```

282 \tikzsymbols@let{eggbeater}{Schneebesen}

\Sieb,\sieve Now a long one;
283 \tikzsymbols@Declare@Robust@Command{Sieb}[1][1]{%
284 \tikzsymbols@ifsaveboxundefined{Sieb#1}{%
285 \set@tkzsymbolsscl{#1}%
286 \iftikzymbols@draftabsolute%
287 \tkzsymbols@bx@Prmtrstore{#1}%
288 \tikzsymbols@draftbox{3.478ex*\tkzsymbols@Prmtr}{1.175ex*\tkzsymbols@Prmtr}%
289 \else%
290 \ifdim\tikzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
291 \begin{tikzpicture}[x=2.8ex, y=2.8ex, line width=0.02ex*\tkzsymbols@scl, scale=#1]
292 \draw[line width=0.09ex*\tkzsymbols@scl] (-0.2,0) -- (1.01,0);
293 \foreach \pos in { 0.2,0.25,...,0.551 }
294 \draw (\pos,0) arc [start angle=180, end angle=360, radius=0.6-\pos];
295 \foreach \x in { 0.95,0.9,...,0.249 }
296 \draw (\x,0) -- (\x,{ -0.4*sin(acos(( \x - 0.6 ) /0.4))});
297 \foreach \y in { -0.05,-0.1,...,-0.351 }
298 \pgfmathsetmacro{\x}{0.4*cos( asin( \y /0.4 ) )}
299 \draw ({0.6 - \x },\y) -- ({ 0.6 +\x},\y);
300 \end{tikzpicture}%
301 \fi%
302 }%
303 \tikzsymbols@use@box{Sieb#1}%
304 \tikzsymbols@ftersymbolinput%
305 \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 “Pürrierstab” would be
“immersion blender”, but I am just using “blender”:
306 \tikzsymbols@Declare@Robust@Command{Purierstab}[1][1]{%
307 \tikzsymbols@ifsaveboxundefined{Purierstab#1}{%
308 \set@tkzsymbolsscl{#1}%
309 \iftikzymbols@draftabsolute%
310 \tkzsymbols@bx@Prmtrstore{#1}%
311 \tikzsymbols@draftbox{0.76ex*\tkzsymbols@Prmtr}{1.575ex*\tkzsymbols@Prmtr}%
312 \else%
313 \ifdim\tikzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
314 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.07ex*\tkzsymbols@scl, scale=#1]
315 \draw[rounded corners=0.07ex*\tkzsymbols@scl] (0,0) -- (0.3,0) -- (0.15,0.1) --cycle;
316 \fill[rounded corners=0.07ex*\tkzsymbols@scl] (0.15,0.3) -- (0.24,0.4) -- (0.24,0.7) --
317 (0.06,0.7) -- (0.06,0.4) -- cycle;
318 \draw (0.15,0.4) -- (0.15,0.1);
319 \end{tikzpicture}%
320 \fi%
321 }%
322 \tikzsymbols@use@box{Purierstab#1}%
323 \tikzsymbols@ftersymbolinput%
324 }
325 \tikzsymbols@let{blender}{Purierstab}

```

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

```

326 \tkzsymbols@Declare@Robust@Command{Dreizack}[1][1]{%
327 \tikzsymbols@ifsaveboxundefined{Dreizack#1}{%
328 \set@tkzsymbolsscl{#1}%
329 \iftikzymbols@draftabsolute%
330 \tkzsymbols@bx@Prmtrstore{#1}%
331 \tikzsymbols@draftbox{0.265ex*\tkzsymbols@Prmtr}{1.575ex*\tkzsymbols@Prmtr}%
332 \else%
333 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
334 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.035ex*\tkzsymbols@scl,scale=#1]
335 \fill[rounded corners=0.07ex*(\tkzsymbols@scl-\tkzsymbols@scl/100)]
336 (0,0) -- (0,0.4) -- (0.1,0.4) -- (0.1,0.0) -- cycle;
337 \draw (0.05,0) -- (0.05,0.7);
338 \draw[rounded corners=0.07ex*(\tkzsymbols@scl-\tkzsymbols@scl/100*\tkzsymbols@scl*2]
339 (0,0.7) -- (0,0.55) -- (0.05,0.55) -- (0.1,0.55) -- (0.1,0.7);
340 \end{tikzpicture}%
341 \fi%
342 }%
343 \tikzsymbols@use@box{Dreizack#1}%
344 \tikzsymbolsaftersymbolinput%
345 }
346 \tikzsymbols@let{trident}{Dreizack}

```

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

```

347 \tkzsymbols@Declare@Robust@Command{Backblech}[1][1]{%
348 \tikzsymbols@ifsaveboxundefined{Backblech#1}{%
349 \set@tkzsymbolsscl{#1}%
350 \iftikzymbols@draftabsolute%
351 \tkzsymbols@bx@Prmtrstore{#1}%
352 \tikzsymbols@draftbox{2.3155ex*\tkzsymbols@Prmtr}{1.57ex*\tkzsymbols@Prmtr}%
353 \else%
354 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
355 \begin{tikzpicture}[x=6.53ex,y=5ex, line width=0.07ex*\tkzsymbols@scl,scale=#1]
356 \filldraw[rounded corners=0.09ex*\tkzsymbols@scl] (0,0) rectangle (0.3,0.3);
357 \foreach \xI/\xII in { 0.1/-0.025 , 0.2/0.325 }
358 \draw[rounded corners=0.07ex*\tkzsymbols@scl, line width=0.03ex*\tkzsymbols@scl]
359 (\xI,0) -- (\xII,0) -- (\xII,0.3) -- (\xI,0.3);
360 \foreach \@BackblechlochX in {0.007,0.293}
361 \foreach \@BackblechlochY in {0.007,0.293}
362 \fill[white] (\@BackblechlochX, \@BackblechlochY) circle (0.02ex);
363 \end{tikzpicture}%
364 \fi%
365 }%
366 \tikzsymbols@use@box{Backblech#1}%
367 \tikzsymbolsaftersymbolinput%
368 }
369 \tikzsymbols@let{bakingplate}{Backblech}

```

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

```

370 \tkzsymbols@Declare@Robust@Command{Ofen}[1][1]{%
371 \tikzsymbols@ifsaveboxundefined{Ofen#1}{%
372 \set@tkzsymbolsscl{#1}%
373 \iftikzymbols@draftabsolute%
374 \tkzsymbols@bx@Prmtrstore{#1}%
375 \tikzsymbols@draftbox{2.07ex*\tkzsymbols@Prmtr}{1.57ex*\tkzsymbols@Prmtr}%
376 \else%
377 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
378 \begin{tikzpicture}[x=0.50ex,y=0.5ex, line width=0.07ex*\tkzsymbols@scl,scale=#1]
379 \draw (0,0) rectangle (4,3);
380 \draw (0.25,0.25) rectangle (3.75,2);
381 \foreach \Ofenschalter in {0.5,1.1,2.9,3.5}
382 \fill (\Ofenschalter,2.5) circle (0.22);
383 \draw (1.5,2.28) rectangle (2.5,2.72);
384 \draw[line width=0.05ex*\tkzsymbols@scl] (1,1.75) -- (3,1.75);
385 \end{tikzpicture}%
386 \fi%
387 }%
388 \tikzsymbols@use@box{Ofen#1}%
389 \tikzsymbols@aftersymbolinput%
390 }
391 \tikzsymbols@let{oven}{Ofen}

```

\Pfanne,\pan A pan ... What did you expect?

```

392 \tkzsymbols@Declare@Robust@Command{Pfanne}[1][1]{%
393 \tikzsymbols@ifsaveboxundefined{Pfanne#1}{%
394 \set@tkzsymbolsscl{#1}%
395 \iftikzymbols@draftabsolute%
396 \tkzsymbols@bx@Prmtrstore{#1}%
397 \tikzsymbols@draftbox{3.034ex*\tkzsymbols@Prmtr}{0.78ex*\tkzsymbols@Prmtr}%
398 \else%
399 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
400 \begin{tikzpicture}[x=2.3ex,y=2.3ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
401 \draw [rounded corners=0.023ex*\tkzsymbols@scl]
402 (0,0) -- (0.9,0) -- (1,0.3) -- (-0.1,0.3) -- cycle;
403 \draw (-0.2,0.22) -- (-0.08,0.22);
404 \draw (0.97,0.22) -- (1.08,0.22);
405 \draw[decorate,decoration={snake,amplitude=.046ex*\tkzsymbols@scl,
406 segment length=0.82ex*\tkzsymbols@scl},line width=0.05ex*\tkzsymbols@scl]
407 (-0.05,0.1) -- (0.95,0.1);
408 \end{tikzpicture}%
409 \fi%
410 }%
411 \tikzsymbols@use@box{Pfanne#1}%
412 \tikzsymbols@aftersymbolinput%
413 }
414 \tikzsymbols@let{pan}{Pfanne}

```

\Herd,\cooker I hope it's the right translation:

```

415 \tkzsymbols@Declare@Robust@Command{Herd}[1][1]{%
416 \tikzsymbols@ifsaveboxundefined{Herd#1}{%
417 \set@tkzsymbolsscl{#1}%
418 \iftikzymbols@draftabsolute%
419 \tkzsymbols@bx@Prmtrstore{#1}%
420 \tikzsymbols@draftbox{2.08ex*\tkzsymbols@Prmtr}{1.58ex*\tkzsymbols@Prmtr}%
421 \else%
422 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
423 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.04ex*\tkzsymbols@scl,scale=#1]
424 \draw[line width=0.08ex*\tkzsymbols@scl] (0,0) rectangle (2,1.5);
425 \foreach \y/\radius in { 0.45/0.35 , 0.45/0.2 , 1.15/0.21 }
426 \draw (0.5,\y) circle (\radius);
427 \draw (1.45,1.15) circle (0.15);
428 \draw (1.45,0.45) circle (0.3);
429 \draw (1.05,0.95) rectangle (1.85,1.35);
430 \end{tikzpicture}%
431 \fi%
432 }%
433 \tikzsymbols@use@box{Herd#1}%
434 \tikzsymbols@aftersymbolinput%
435 }
436 \tikzsymbols@let{cooker}{Herd}

```

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

```

437 \tkzsymbols@Declare@Robust@Command{Saftpresse}[1][1]{%
438 \tikzsymbols@ifsaveboxundefined{Saftpresse#1}{%
439 \set@tkzsymbolsscl{#1}%
440 \iftikzymbols@draftabsolute%
441 \tkzsymbols@bx@Prmtrstore{#1}%
442 \tikzsymbols@draftbox{1.87ex*\tkzsymbols@Prmtr}{1.62ex*\tkzsymbols@Prmtr}%
443 \else%
444 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
445 \begin{tikzpicture}[x=1.2ex,y=1ex,line width=0.07ex*\tkzsymbols@scl,scale=#1]
446 \draw[rounded corners=0.1ex*\tkzsymbols@scl]
447 (0,0) rectangle (1.5,0.85) -- cycle;
448 \draw (0,0.7) -- (1.5,0.7);
449 \foreach \xi/\xii in { 0.3/1.2 , 0.45/1.05 , 0.65/0.85 }
450 \draw[rounded corners=0.1ex*\tkzsymbols@scl]
451 (\xi,0.7) -- (0.75,1.55) -- (\xii,0.7);
452 \draw[line width=0.05ex*\tkzsymbols@scl, decorate,
453 decoration={snake,amplitude=0.05ex*\tkzsymbols@scl,
454 segment length=0.48ex*\tkzsymbols@scl}] (0,0.3) -- (1.5,0.3);
455 \end{tikzpicture}%
456 \fi%
457 }%
458 \tikzsymbols@use@box{Saftpresse#1}%
459 \tikzsymbols@aftersymbolinput%
460 }
461 \tikzsymbols@let{squeezer}{Saftpresse}

```

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

```

462 \tkzsymls@Declare@Robust@Command{Schussel}[1][1]{%
463 \tikzsymbols@ifsaveboxundefined{Schussel#1}{%
464 \set@tkzsymls@{#1}%
465 \iftikzsymbols@draftabsolute%
466 \tkzsymls@bx@Prmtrstore{#1}%
467 \tikzsymbols@draftbox{2.32ex*\tkzsymls@Prmtr}{1.47ex*\tkzsymls@Prmtr}%
468 \else%
469 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@{#1}\fi%
470 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex*\tkzsymls@scl, scale=#1]
471 \draw[rounded corners=0.5ex*\tkzsymls@scl]
472   (-0.02,1.4) -- (0,1.4) -- (0,0.05) -- (1.5,0.05) -- (1.5,1.4) -- (1.52,1.4);
473 \draw (0.35,0) -- (1.15,0);
474 \draw[transparent] (-0.4,0) -- (1.85,0);
475 \end{tikzpicture}%
476 \fi%
477 }%
478 \tikzsymbols@use@box{Schussel#1}%
479 \tikzsymbols@after@symbolinput%
480 }
481 \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:

```

482 \tkzsymls@Declare@Robust@Command{Schaler}[1][1]{%
483 \tikzsymbols@ifsaveboxundefined{Schaler#1}{%
484 \set@tkzsymls@{#1}%
485 \iftikzsymbols@draftabsolute%
486 \tkzsymls@bx@Prmtrstore{#1}%
487 \tikzsymbols@draftbox{1.15ex*\tkzsymls@Prmtr}{1.565ex*\tkzsymls@Prmtr}%
488 \else%
489 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@{#1}\fi%
490 \begin{tikzpicture}[x=2.7ex,y=2.3ex, line width=0.07ex*\tkzsymls@scl,scale=#1]
491 \draw[rounded corners=0.07ex*\tkzsymls@scl]
492   (0,0.4) -- (0,0.1) arc (0:180:-0.1) -- (0.2,0.4)
493   -- (0.3,0.5) -- (0.3,0.65) -- (0.2,0.65) -- (0.2,0.5) -- (0,0.5) -- (0,0.65) --
494   (-0.1,0.65) -- (-0.1,0.5) -- cycle;
495 \draw[line width=0.03ex*\tkzsymls@scl] (0,0.58) rectangle (0.2,0.6);
496 \end{tikzpicture}%
497 \fi%
498 }%
499 \tikzsymbols@use@box{Schaler#1}%
500 \tikzsymbols@after@symbolinput%
501 }
502 \tikzsymbols@let{peeler}{Schaler}

```

Text hinschreiben, \grater Text hinschreiben

```

503 \tkzsymls@Declare@Robust@Command{Reibe}[1][1]{%
504 \tikzsymbols@ifsaveboxundefined{Reibe#1}{%

```

```

505 \set@tkzsymlsscl{#1}%
506 \iftikzsymbols@draftabsolute%
507 \tkzsymls@bx@Prmtrstore{#1}%
508 \tikzsymbols@draftbox{1.08ex*\tkzsymls@Prmtr}{1.58ex*\tkzsymls@Prmtr}%
509 \else%
510 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
511 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.08ex*\tkzsymls@scl, scale=#1]%
512 \draw (0,0) rectangle (1,1.2);
513 \draw[rounded corners=0.04ex] (0.05,1.2) rectangle (0.95,1.5);
514 \foreach\x in {0.2,0.4,0.6,0.8}
515 \foreach\y in {0.2,0.4, 0.6, 0.8, 1}
516 \fill (\x,\y) circle (0.05ex);
517 \end{tikzpicture}%
518 \fi%
519 }%
520 \tikzsymbols@use@box{Reibe#1}%
521 \tikzsymbolsaftersymbolinput%
522 }
523 \tikzsymbols@let{grater}{Reibe}

\Flasche,\bottle Text hinschreiben
524 \tkzsymls@Declare@Robust@Command{Flasche}[1][1]{%
525 \tikzsymbols@ifsaveboxundefined{Flasche#1}{%
526 \set@tkzsymlsscl{#1}%
527 \iftikzsymbols@draftabsolute%
528 \tkzsymls@bx@Prmtrstore{#1}%
529 \tikzsymbols@draftbox{0.78ex*\tkzsymls@Prmtr}{1.58ex*\tkzsymls@Prmtr}%
530 \else%
531 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
532 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.08ex*\tkzsymls@scl,
533 rounded corners=0.08ex*\tkzsymls@scl, scale=#1]
534 \draw (0, 1.5) -- (0,1.2) -- (-0.15,0.8) -- (-0.15,0) ---- (0.6,0) ---- (0,0.8) ---- (-0.15,0.4
535 %\draw (-0.15,0.8) -- (0.45,0.8);
536 %\draw (-0.15,0.3) -- (0.45,0.3);
537 \draw[transparent] (-0.2,0) ---- (0.7,0);
538 \end{tikzpicture}\fi%
539 }%
540 \tikzsymbols@use@box{Flasche#1}%
541 \tikzsymbolsaftersymbolinput%
542 }
543 \tikzsymbols@let{bottle}{Flasche}

```

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 `none`, it’s useful for `\colorbox{yellow}{\Sadey}` which leads to ☹ instead of 😞 (with default=white).


```

544 \tkzsymls@Declare@Robust@Commandx{Sadey}[2][1=1, 2={none}, usedefault]{%
545 \tikzsymbols@ifsaveboxundefined{Sadey#1#2}{%
546 \set@tkzsymlsscl{#1}%
547 \iftikzymbols@draftabsolute%
548 \tkzsymls@bx@Prmtrstore{#1}%
549 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
550 \else%
551 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
552 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
553 \filldraw[fill=#2, line width=0.1ex*\tkzsymls@scl] (0,0) circle (0.33);
554 \fill (0.1,0.1) circle (0.05);
555 \fill (-0.1,0.1) circle (0.05);
556 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
557 \end{tikzpicture}%
558 \fi%
559 }%
560 \tikzsymbols@use@box{Sadey#1#2}%
561 \tikzsymbolsaftersymbolinput%
562 }
563 \tkzsymls@Declare@Robust@Commandx{dSadey}[2][1=1,2=yellow,usedefault]{%
564 \tikzsymbols@ifsaveboxundefined{dSadey#1#2}{%
565 \set@tkzsymlsscl{#1}%
566 \iftikzymbols@draftabsolute%
567 \tkzsymls@bx@Prmtrstore{#1}%
568 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
569 \else%
570 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
571 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
572 \shade[ball color=#2] (0,0) circle (0.33);
573 \shade[ball color=black] (0.1,0.1) circle (0.05);
574 \shade[ball color=black] (-0.1,0.1) circle (0.05);
575 \draw[black] (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
576 \end{tikzpicture}%
577 \fi%
578 }%
579 \tikzsymbols@use@box{dSadey#1#2}%
580 \tikzsymbolsaftersymbolinput%
581 }

```

\Annoey \dAnnoey An annoyed Smiley --

```

582 \tkzsymls@Declare@Robust@Commandx{Annoey}[2][1=1,2={none},usedefault]{%
583 \tikzsymbols@ifsaveboxundefined{Annoey#1#2}{%
584 \set@tkzsymlsscl{#1}%
585 \iftikzymbols@draftabsolute%
586 \tkzsymls@bx@Prmtrstore{#1}%
587 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
588 \else%
589 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
590 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
591 \filldraw[fill=#2, line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);

```

```

592 \draw (0.08,0.1) -- (0.22,0.1);
593 \draw (-0.08,0.1) -- (-0.22,0.1);
594 \draw (-0.2,-0.1) -- (0.2,-0.1);
595 \end{tikzpicture}%
596 \fi%
597 }%
598 \tikzsymbols@use@box{Annoey#1#2}%
599 \tikzsymbolsaftersymbolinput%
600 }
601 \tkzsymls@Declare@Robust@Commandx{dAnnoey}[2][1=1,2=yellow,usedefault]{%
602 \tikzsymbols@ifsaveboxundefined{dAnnoey#1#2}{%
603 \set@tkzsymls@{#1}%
604 \iftikzsymbols@draftabsolute%
605 \tkzsymls@bx@Prmtrstore{#1}%
606 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
607 \else%
608 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@{#1}\fi%
609 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
610 \shade[ball color=#2] (0,0) circle (0.33);
611 \draw[black] (0.08,0.1) -- (0.22,0.1);
612 \draw[black] (-0.08,0.1) -- (-0.22,0.1);
613 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
614 \end{tikzpicture}%
615 \fi%
616 }%
617 \tikzsymbols@use@box{dAnnoey#1#2}%
618 \tikzsymbolsaftersymbolinput%
619 }

```

\Smiley \dSmiley A normal Smiley

```

620 \ifKV@tikzsymbols@marvosym\relax\else%
621 \tkzsymls@Declare@Robust@Commandx{Smiley}[2][1=1,2={none},usedefault]{%
622 \tikzsymbols@ifsaveboxundefined{Smiley#1#2}{%
623 \set@tkzsymls@{#1}%
624 \iftikzsymbols@draftabsolute%
625 \tkzsymls@bx@Prmtrstore{#1}%
626 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
627 \else%
628 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@{#1}\fi%
629 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl, scale=#1]
630 \filldraw[fill=#2] (0,0) circle (0.33);
631 \fill (-0.1,0.1) circle (0.05);
632 \fill (0.1,0.1) circle (0.05);
633 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
634 \end{tikzpicture}%
635 \fi%
636 }%
637 \tikzsymbols@use@box{Smiley#1#2}%
638 \tikzsymbolsaftersymbolinput%
639 }%

```

```

640 \fi
641 \tkzsymls@Declare@Robust@Commandx{dSmiley}[2][1=1,2=yellow,usedefault]{%
642 \tikzsymbols@ifsaveboxundefined{dSmiley#1#2}{%
643 \set@tkzsymlsscl{#1}%
644 \iftikzsymbols@draftabsolute%
645 \tkzsymls@bx@Prmtrstore{#1}%
646 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
647 \else%
648 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
649 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.1ex*\tkzsymls@scl,scale=#1]
650 \shade[ball color=#2] (0,0) circle (0.33);
651 \shade[ball color=black] (-0.1,0.1) circle (0.05);
652 \shade[ball color=black] (0.1,0.1) circle (0.05);
653 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
654 \end{tikzpicture}%
655 \fi%
656 }%
657 \tikzsymbols@use@box{dSmiley#1#2}%
658 \tikzsymbolsaftersymbolinput%
659 }

```

\Laughey \dLaughey A laughing Smiley

```

660 \tkzsymls@Declare@Robust@Commandx{Laughey}[3][1=1,2={none},3={none} ,usedefault]{%
661 \tikzsymbols@ifsaveboxundefined{Laughey#1#2#3}{%
662 \set@tkzsymlsscl{#1}%
663 \iftikzsymbols@draftabsolute%
664 \tkzsymls@bx@Prmtrstore{#1}%
665 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
666 \else%
667 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
668 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
669 \filldraw[fill=#2,line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);
670 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
671 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
672 \filldraw[fill=#3,rounded corners=0.1ex*\tkzsymls@scl, yshift=-0.5]
673 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
674 \end{tikzpicture}%
675 \fi%
676 }%
677 \tikzsymbols@use@box{Laughey#1#2#3}%
678 \tikzsymbolsaftersymbolinput%
679 }
680 \tkzsymls@Declare@Robust@Commandx{dLaughey}[3][1=1,2=yellow, 3=red ,usedefault]{%
681 \tikzsymbols@ifsaveboxundefined{dLaughey#1#2#3}{%
682 \set@tkzsymlsscl{#1}%
683 \iftikzsymbols@draftabsolute%
684 \tkzsymls@bx@Prmtrstore{#1}%
685 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
686 \else%
687 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%

```

```

688 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
689 \fill[ball color=#2,line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);
690 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
691 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
692 \shade[ball color=#3, rounded corners=0.1ex*\tkzsymls@scl, yshift=-0.3]
693 (-0.25,-0.0) .. controls (-0.13,-0.26) and (0.13,-0.26) .. (0.25,-0.0) -- cycle;
694 \end{tikzpicture}%
695 \fi%
696 }%
697 \tikzsymbols@use@box{dLaughey#1#2#3}%
698 \tikzsymbolsaftersymbolinput%
699 }

```

\Neutrey \dNeutrey neutral Smiley : |

```

700 \tikzsymbols@Declare@Robust@Commandx{Neutrey}[2][1=1, 2={none}, usedefault]{%
701 \tikzsymbols@ifsaveboxundefined{Neutrey#1#2}{%
702 \set@tkzsymlsscl{#1}%
703 \iftikzymbols@draftabsolute%
704 \tkzsymls@bx@Prmtrstore{#1}%
705 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
706 \else%
707 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
708 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
709 \filldraw[fill=#2,line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);
710 \fill (0.1,0.1) circle (0.05);
711 \fill (-0.1,0.1) circle (0.05);
712 \draw (-0.2,-0.1) -- (0.2,-0.1);
713 \end{tikzpicture}%
714 \fi%
715 }%
716 \tikzsymbols@use@box{Neutrey#1#2}%
717 \tikzsymbolsaftersymbolinput%
718 }
719 \tikzsymbols@Declare@Robust@Commandx{dNeutrey}[2][1=1,2=yellow,usedefault]{%
720 \tikzsymbols@ifsaveboxundefined{dNeutrey#1#2}{%
721 \set@tkzsymlsscl{#1}%
722 \iftikzymbols@draftabsolute%
723 \tkzsymls@bx@Prmtrstore{#1}%
724 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
725 \else%
726 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
727 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
728 \shade[ball color=#2] (0,0) circle (0.33);
729 \shade[ball color=black] (0.1,0.1) circle (0.05);
730 \shade[ball color=black] (-0.1,0.1) circle (0.05);
731 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
732 \end{tikzpicture}%
733 \fi%
734 }%
735 \tikzsymbols@use@box{dNeutrey#1#2}%

```

```

736 \tikzsymbolsaftersymbolinput%
737 }

\Winkey \dWinkey ;)
\oldWinkey \olddWinkey
738 \tkzsymbols@Declare@Robust@Commandx{Winkey}[2][1=1,2={none} ,usedefault]{%
739 \tikzsymbols@ifsaveboxundefined{Winkey#1#2}{%
740 \set@tkzsymbolsscl{#1}%
741 \iftikzymbols@draftabsolute%
742 \tkzsymbols@bx@Prmtrstore{#1}%
743 \tikzsymbols@draftQbox{1.704ex*\tkzsymbols@Prmtr}%
744 \else%
745 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
746 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
747 \filldraw[fill=#2] (0,0) circle (0.33);
748 \draw(0.17,0.1) -- (0.05,0.1);
749 \fill (-0.1,0.1) circle (0.05);
750 \draw (-0.15,-0.15) .. controls (-0.05,-0.2) and (0.15,-0.2) .. (0.19,0);
751 \end{tikzpicture}%
752 \fi%
753 }%
754 \tikzsymbols@use@box{Winkey#1#2}%
755 \tikzsymbolsaftersymbolinput%
756 }

757 \tkzsymbols@Declare@Robust@Commandx{oldWinkey}[2][1=1,2={none} ,usedefault]{%
758 \tikzsymbols@ifsaveboxundefined{oldWinkey#1#2}{%
759 \set@tkzsymbolsscl{#1}%
760 \iftikzymbols@draftabsolute%
761 \tkzsymbols@bx@Prmtrstore{#1}%
762 \tikzsymbols@draftQbox{1.704ex*\tkzsymbols@Prmtr}%
763 \else%
764 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
765 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
766 \filldraw[fill=#2] (0,0) circle (0.33);
767 \draw(0.17,0.1) -- (0.05,0.1);
768 \fill (-0.1,0.1) circle (0.05);
769 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
770 \end{tikzpicture}%
771 \fi%
772 }%
773 \tikzsymbols@use@box{oldWinkey#1#2}%
774 \tikzsymbolsaftersymbolinput%
775 }

776 \tkzsymbols@Declare@Robust@Commandx{dWinkey}[2][1=1,2=yellow,usedefault]{%
777 \tikzsymbols@ifsaveboxundefined{dWinkey#1#2}{%
778 \set@tkzsymbolsscl{#1}%
779 \iftikzymbols@draftabsolute%
780 \tkzsymbols@bx@Prmtrstore{#1}%
781 \tikzsymbols@draftQbox{1.584ex*\tkzsymbols@Prmtr}%
782 \else%
783 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%

```

```

784 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
785 \shade[ball color=#2] (0,0) circle (0.33);
786 \draw[black] (0.17,0.1) -- (0.05,0.1);
787 \shade[ball color=black] (-0.1,0.1) circle (0.05);
788 \draw[black] (-0.15,-0.15) .. controls (-0.05,-0.2) and (0.15,-0.2) .. (0.19,0);
789 \end{tikzpicture}%
790 \fi%
791 }%
792 \tikzsymbols@use@box{dWinkey#1#2}%
793 \tikzsymbolsaftersymbolinput%
794 }
795 \tkzsymls@Declare@Robust@Commandx{olddWinkey}[2][1=1,2=yellow,usedefault]{%
796 \tikzsymbols@ifsaveboxundefined{olddWinkey#1#2}{%
797 \set@tkzsymls@scl{#1}%
798 \iftikzsymbols@draftabsolute%
799 \tkzsymls@bx@Prmtrstore{#1}%
800 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
801 \else%
802 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
803 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
804 \shade[ball color=#2] (0,0) circle (0.33);
805 \draw(0.17,0.1) -- (0.05,0.1);
806 \shade[ball color=black] (-0.1,0.1) circle (0.05);
807 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
808 \end{tikzpicture}%
809 \fi%
810 }%
811 \tikzsymbols@use@box{olddWinkey#1#2}%
812 \tikzsymbolsaftersymbolinput%
813 }

```

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

```

814 \tkzsymls@Declare@Robust@Commandx{Sey}[2][1=1,2={none} ,usedefault]{%
815 \tikzsymbols@ifsaveboxundefined{Sey#1#2}{%
816 \set@tkzsymls@scl{#1}%
817 \iftikzsymbols@draftabsolute%
818 \tkzsymls@bx@Prmtrstore{#1}%
819 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
820 \else%
821 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
822 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
823 \filldraw[fill=#2, line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);
824 \fill (0.1,0.1) circle (0.05);
825 \fill (-0.1,0.1) circle (0.05);
826 \draw (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
827 \end{tikzpicture}%
828 \fi%
829 }%
830 \tikzsymbols@use@box{Sey#1#2}%
831 \tikzsymbolsaftersymbolinput%

```

```

832 }
833 \tkzsymls@Declare@Robust@Commandx{dSey}[2][1=1,2=yellow ,usedefault]{%
834 \tikzsymbols@ifsaveboxundefined{dSey#1#2}{%
835 \set@tkzsymlsscl{#1}%
836 \iftikzymbols@draftabsolute%
837 \tkzsymls@bx@Prmtrstore{#1}%
838 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
839 \else%
840 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
841 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
842 \shade[ball color=#2] (0,0) circle (0.33);
843 \shade[ball color=black] (0.1,0.1) circle (0.05);
844 \shade[ball color=black] (-0.1,0.1) circle (0.05);
845 \draw[black] (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
846 \end{tikzpicture}%
847 \fi%
848 }%
849 \tikzsymbols@use@box{dSey#1#2}%
850 \tikzsymbolsaftersymbolinput%
851 }

```

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

```

852 \tkzsymls@Declare@Robust@Commandx{Xey}[2][1=1, 2={none}, usedefault]{%
853 \tikzsymbols@ifsaveboxundefined{Xey#1#2}{%
854 \set@tkzsymlsscl{#1}%
855 \iftikzymbols@draftabsolute%
856 \tkzsymls@bx@Prmtrstore{#1}%
857 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
858 \else%
859 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
860 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
861 \filldraw[fill=#2, line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);
862 \foreach \xi in { 0.05 , -0.15 }
863 \draw (\xi,0.05) -- ++ (0.1,0.1) (-\xi,0.05) -- ++ (-0.1,0.1);
864 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
865 \end{tikzpicture}%
866 \fi%
867 }%
868 \tikzsymbols@use@box{Xey#1#2}%
869 \tikzsymbolsaftersymbolinput%
870 }
871 \tkzsymls@Declare@Robust@Commandx{dXey}[2][1=1, 2={yellow}, usedefault]{%
872 \tikzsymbols@ifsaveboxundefined{dXey#1#2}{%
873 \set@tkzsymlsscl{#1}%
874 \iftikzymbols@draftabsolute%
875 \tkzsymls@bx@Prmtrstore{#1}%
876 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
877 \else%
878 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
879 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]

```

```

880 \fill[ball color=#2, line width=0.12ex*\tkzsymls@scl] (0,0) circle (0.33);
881 \foreach \xi in { 0.05 , -0.15 }
882     \draw (\xi,0.05) -- ++ (0.1,0.1) (-\xi,0.05) -- ++ (-0.1,0.1);
883 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
884 \end{tikzpicture}%
885 \fi%
886 }%
887 \tikzsymbols@use@box{dXey#1#2}%
888 \tikzsymbolsaftersymbolinput%
889 }

```

\Innocey \dInnocey An innocent Smiley

```

890 \tkzsymls@Declare@Robust@Commandx{Innocey}[3][1=1,2={none},3=yellow ,usedefault]{%
891 \tikzsymbols@ifsaveboxundefined{Innocey#1#2#3}{%
892 \set@tkzsymls@scl{#1}%
893 \iftikzsymls@draftabsolute%
894 \tkzsymls@bx@Prmtrstore{#1}%
895 \tikzsymbols@draftbox{1.73ex*\tkzsymls@Prmtr}{1.909ex*\tkzsymls@Prmtr}%
896 \else%
897 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
898 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
899 \filldraw[fill=#2] (0,0) circle (0.33);
900 \fill (-0.1,0.1) circle (0.05);
901 \fill (0.1,0.1) circle (0.05);
902 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
903 \draw[#3, line width=0.095ex*\tkzsymls@scl] (0.32,0.31) arc (0:360:0.32 and 0.1);
904 \draw[line width=0.05ex*\tkzsymls@scl] (0.3,0.31) arc (0:360:0.3 and 0.07);
905 \draw[line width=0.05ex*\tkzsymls@scl] (0.35,0.31) arc (0:360:0.35 and 0.12);
906 \end{tikzpicture}%
907 \fi%
908 }%
909 \tikzsymbols@use@box{Innocey#1#2#3}%
910 \tikzsymbolsaftersymbolinput%
911 }
912 \tkzsymls@Declare@Robust@Command{wInnocey}[1][1]{\Innocey[#1][none][white]}
913 \tkzsymls@Declare@Robust@Commandx{dInnocey}[3][1=1,2=yellow,3=yellow,usedefault]{%
914 \tikzsymbols@ifsaveboxundefined{dInnocey#1#2#3}{%
915 \set@tkzsymls@scl{#1}%
916 \iftikzsymls@draftabsolute%
917 \tkzsymls@bx@Prmtrstore{#1}%
918 \tikzsymbols@draftbox{1.73ex*\tkzsymls@Prmtr}{1.849ex*\tkzsymls@Prmtr}%
919 \else%
920 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
921 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
922 \shade[ball color=#2] (0,0) circle (0.33);
923 \shade[ball color=black] (-0.1,0.1) circle (0.05);
924 \shade[ball color=black] (0.1,0.1) circle (0.05);
925 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
926 \draw[color=#3!97!black, line width=0.1ex*\tkzsymls@scl]
927     (0.32,0.31) arc (0:360:0.32 and 0.1);

```



```

928 \draw[line width=0.05ex*\tkzsymls@scl] (0.3,0.31) arc (0:360:0.3 and 0.07);
929 \draw[line width=0.05ex*\tkzsymls@scl] (0.35,0.31) arc (0:360:0.35 and 0.12);
930 \end{tikzpicture}%
931 \fi%
932 }%
933 \tikzsymbols@use@box@dInnocey#1#2#3}%
934 \tikzsymbolsaftersymbolinput%
935 }

```

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

```

936 \tikzsymbols@Declare@Robust@Commandx{Cooley}[2][1=1,2={none} ,usedefault]{%
937 \tikzsymbols@ifsaveboxundefined{Cooley#1#2}{%
938 \set@tkzsymls@scl{#1}%
939 \iftikzymbols@draftabsolute%
940 \tkzsymls@bx@Prmtrstore{#1}%
941 \tikzsymbols@draftQbox{1.704ex*\tkzsymls@Prmtr}%
942 \else%
943 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
944 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
945 \filldraw[fill=#2] (0,0) circle (0.33);
946 \foreach \xi/\xii in { 0.24/0.01 , -0.24/-0.01 }
947 \fill[rounded corners=0.1ex*\tkzsymls@scl]
948 (\xi,0.15) -- (\xii,0.15) -- (\xii,0) -- (\xi,0) -- cycle;
949 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
950 \draw (0.02,0.1) -- (-0.02,0.1);
951 \draw (-0.2,0.1) -- (-0.3,0.13);
952 \draw (0.2,0.1) -- (0.3,0.13);
953 \end{tikzpicture}%
954 \fi%
955 }%
956 \tikzsymbols@use@box{Cooley#1#2}%
957 \tikzsymbolsaftersymbolinput%
958 }
959 \tikzsymbols@Declare@Robust@Commandx{dCooley}[2][1=1,2=yellow,usedefault]{%
960 \tikzsymbols@ifsaveboxundefined{dCooley#1#2}{%
961 \set@tkzsymls@scl{#1}%
962 \iftikzymbols@draftabsolute%
963 \tkzsymls@bx@Prmtrstore{#1}%
964 \tikzsymbols@draftQbox{1.584ex*\tkzsymls@Prmtr}%
965 \else%
966 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
967 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
968 \shade[ball color=#2] (0,0) circle (0.33);
969 \draw[black] (0.02,0.1) -- (-0.02,0.1);
970 \draw[black] (-0.2,0.1) -- (-0.295,0.146);
971 \draw[black] (0.2,0.1) -- (0.295,0.146);
972 \foreach \xi/\xii in { 0.24/0.01 , -0.24/-0.01 }
973 \shade[ball color=black,rounded corners=0.1ex*\tkzsymls@scl]
974 (\xi,0.15) -- (\xii,0.15) -- (\xii,0) -- (\xi,0) -- cycle;
975 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);

```

```

976 \end{tikzpicture}%
977 \fi%
978 }%
979 \tikzsymbols@use@box{dCooley#1#2}%
980 \tikzsymbolsaftersymbolinput%
981 }

\Tongey \dTongey :P
982 \tikzsymbols@Declare@Robust@Commandx{Tongey}[3][1=1,2={none},3={none} ,usedefault]{%
983 \tikzsymbols@ifsaveboxundefined{Tongey#1#2#3}{%
984 \set@tkzsymbolsscl{#1}%
985 \iftikzymbols@draftabsolute%
986 \tkzsymbols@bx@Prmtrstore{#1}%
987 \tikzsymbols@draftQbox{1.704ex*\tkzsymbols@Prmtr}%
988 \else%
989 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
990 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
991 \filldraw[fill=#2] (0,0) circle (0.33);
992 \fill (-0.1,0.1) circle (0.05);
993 \fill (0.1,0.1) circle (0.05);
994 \filldraw[fill=#3,line width=0.058ex*\tkzsymbols@scl, rounded corners=0.12ex*\tkzsymbols@scl]
995 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
996 \draw[line width=0.07ex*\tkzsymbols@scl, yshift=0.21ex]
997 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
998 \end{tikzpicture}%
999 \fi%
1000 }%
1001 \tikzsymbols@use@box{Tongey#1#2#3}%
1002 \tikzsymbolsaftersymbolinput%
1003 }
1004 \tikzsymbols@Declare@Robust@Commandx{dTongey}[3][1=1,2=yellow,3=red,usedefault]{%
1005 \tikzsymbols@ifsaveboxundefined{dTongey#1#2#3}{%
1006 \set@tkzsymbolsscl{#1}%
1007 \iftikzymbols@draftabsolute%
1008 \tkzsymbols@bx@Prmtrstore{#1}%
1009 \tikzsymbols@draftQbox{1.584ex*\tkzsymbols@Prmtr}%
1010 \else%
1011 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1012 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
1013 \shade[ball color=#2] (0,0) circle (0.33);
1014 \shade[ball color=black] (-0.1,0.1) circle (0.05);
1015 \shade[ball color=black] (0.1,0.1) circle (0.05);
1016 \shadedraw[ball color=#3, draw=black,line width=0.058ex*\tkzsymbols@scl, rounded corners=0.12ex*
1017 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1018 \draw[black, line width=0.07ex*\tkzsymbols@scl, yshift=0.21ex]
1019 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1020 \end{tikzpicture}%
1021 \fi%
1022 }%
1023 \tikzsymbols@use@box{dTongey#1#2#3}%

```

```

1024 \tikzsymbolsaftersymbolinput%
1025 }

\nursey \dnursey a Nurse.

1026 \tkzsymbols@Declare@Robust@Commandx{Nursey}
1027 [4] [1=1,2={none},3={none},4={},usedefault]{%
1028 \tikzsymbols@ifsaveboxundefined{Nursey#1#2#3#4}{%
1029 \set@tkzsymbolsscl{#1}%
1030 \iftikzymbols@draftabsolute%
1031 \tkzsymbols@bx@Prmtrstore{#1}%
1032 \tikzsymbols@draftbox{1.5ex*\tkzsymbols@Prmtr}{2.19ex*\tkzsymbols@Prmtr}%
1033 \else%
1034 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1035 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
1036 \fill[fill=#3,rounded corners=.023ex*\tkzsymbols@scl]
1037 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
1038 \filldraw[fill=#2] (0,0) circle (0.3);
1039 \fill (-0.1,0.1) circle (0.05);
1040 \fill (0.1,0.1) circle (0.05);
1041 \draw[line width=0.09ex*\tkzsymbols@scl, yshift=0.07ex]
1042 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1043 \draw[rounded corners=.023ex*\tkzsymbols@scl]
1044 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
1045 \draw[#4,line width=.046ex*\tkzsymbols@scl] (0,0.35) -- (0,0.5) (-0.05,0.45) -- (0.05,0.45) ;
1046 \end{tikzpicture}%
1047 \fi%
1048 }%
1049 \tikzsymbols@use@box{Nursey#1#2#3#4}%
1050 \tikzsymbolsaftersymbolinput%
1051 }

1052 \tkzsymbols@Declare@Robust@Commandx{dnursey}[4] [1=1,2=yellow,3=white,4=red,usedefault]{%
1053 \tikzsymbols@ifsaveboxundefined{dnursey#1#2#3#4}{%
1054 \set@tkzsymbolsscl{#1}%
1055 \iftikzymbols@draftabsolute%
1056 \tkzsymbols@bx@Prmtrstore{#1}%
1057 \tikzsymbols@draftbox{1.38ex*\tkzsymbols@Prmtr}{1.98ex*\tkzsymbols@Prmtr}%
1058 \else%
1059 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1060 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
1061 \shade[ball color=#2] (0,0) circle (0.3);
1062 \shade[ball color=black] (-0.1,0.1) circle (0.05);
1063 \shade[ball color=black] (0.1,0.1) circle (0.05);
1064 \draw[black, line width=0.09ex*\tkzsymbols@scl, yshift=0.07ex]
1065 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1066 \shade[ball color=#3, rounded corners=.023ex*\tkzsymbols@scl,yshift=-0.09ex]
1067 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0) arc (0:180:0.3);
1068 \shade[ball color=#4,line width=.046ex*\tkzsymbols@scl]
1069 (-0.01,0.31) -- (-0.01,0.46) -- (0.01,0.46) -- (0.01,0.31)--cycle;
1070 \shade[ball color=#4,line width=.046ex*\tkzsymbols@scl]
1071 (-0.05,0.4) -- (0.05,0.4) -- (0.05,0.42) -- (-0.05,0.42) -- cycle;

```

```

1072 \end{tikzpicture}%
1073 \fi%
1074 }%
1075 \tikzsymbols@use@box{dNurse#1#2#3#4}%
1076 \tikzsymbolsaftersymbolinput%
1077 }

\Vomey \dVomey *Bläärg*

1078 \tikzsymbols@Declare@Robust@Commandx{Vomey}[3][1=1,2={none},3={opacity=0},usedefault]{%
1079 \tikzsymbols@ifsaveboxundefined{Vomey#1#2#3}{%
1080 \set@tkzsymbolsscl{#1}%
1081 \iftikzymbols@draftabsolute%
1082 \tkzsymbols@bx@Prmtrstore{#1}%
1083 \tikzsymbols@draftbox{3.0335ex*\tkzsymbols@Prmtr}{1.743ex*\tkzsymbols@Prmtr}%
1084 \else%
1085 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1086 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
1087 \filldraw[fill=#2,rounded corners=0.05ex*\tkzsymbols@scl] (0,0) arc (15:330:1) -- (-0.6,-0.3)
1088 \draw[line width=0.05ex*\tkzsymbols@scl] (-0.5,0.3) -- (-0.3,0.1);
1089 \fill (-0.45,0.27) arc (100:350:0.1);
1090 \fill[#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1091 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
1092 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1093 \fill[#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) ..
1094 controls (1.7,-1.2) and (1.3,-1.2) .. (1,-1) ..
1095 controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
1096 %\draw (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1);
1097 %\draw (0,-0.4) .. controls (0.5,-0.5) and (0.8,-0.7) .. (1,-1);
1098 \draw[rounded corners=0.1ex*\tkzsymbols@scl] (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,
1099 \draw (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1100 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2)
1101 and (1,-2) .. (1,-1.5) .. controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1102 \end{tikzpicture}%
1103 \fi%
1104 }%
1105 \tikzsymbols@use@box{Vomey#1#2#3}%
1106 \tikzsymbolsaftersymbolinput%
1107 }

1108 \tikzsymbols@Declare@Robust@Commandx{dVomey}[3][1=1,2=yellow,3={brown!10!olive},usedefault]{%
1109 \tikzsymbols@ifsaveboxundefined{dVomey#1#2#3}{%
1110 \set@tkzsymbolsscl{#1}%
1111 \iftikzymbols@draftabsolute%
1112 \tkzsymbols@bx@Prmtrstore{#1}%
1113 \tikzsymbols@draftbox{3.2435ex*\tkzsymbols@Prmtr}{1.653ex*\tkzsymbols@Prmtr}%
1114 \else%
1115 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1116 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
1117 \shade[ball color=#2!90!brown,rounded corners=0.03ex*\tkzsymbols@scl]
1118 (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1119 \draw[black, line width=0.05ex*\tkzsymbols@scl] (-0.5,0.3) -- (-0.3,0.1);

```

```

1120 \shade[ball color=black] (-0.45,0.27) arc (100:350:0.1);
1121 \shade[ball color=#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1122 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
1123 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1124 \shade[ball color=#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) .. controls
1125 (1.7,-1.2) and (1.3,-1.2) .. (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
1126 \end{tikzpicture}%
1127 \fi%
1128 }%
1129 \tikzsymbols@use@box{dVomey#1#2#3}%
1130 \tikzsymbolsaftersymbolinput%
1131 }

```

y \dWalley, \rWalley, \drWalley 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.

```

1132 \tikzsymbols@Declare@Robust@Commandx{Walley}[3][1=1, 2={none},3={none}, usedefault]{%
1133 \tikzsymbols@ifsaveboxundefined{Walley#1#2#3}{%
1134 \set@tkzsymbolsscl{#1}%
1135 \iftikzymbols@draftabsolute%
1136 \tkzsymbols@bx@Prmtrstore{#1}%
1137 \tikzsymbols@draftbox{2.341ex*\tkzsymbols@Prmtr}{1.674ex*\tkzsymbols@Prmtr}%
1138 \else%
1139 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1140 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1,
1141 decoration={random steps,segment length=0.15ex*\tkzsymbols@scl, amplitude=0.1ex*\tkzsymbols@scl}]
1142 \filldraw[fill=#2, line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.28);
1143 \filldraw[fill=#3] (0.28,-0.33) rectangle (0.66,0.33);
1144 \draw[line width=0.06ex*\tkzsymbols@scl] (0.28,0) ---++(0.05,0.07) ---++(0.03,0.02) ---+
1145 +(0.03,-0.02) ---++(0.03,0.1) ---++(0.03,0.02) -- (0.5,0.25);
1146 \draw[line width=0.06ex*\tkzsymbols@scl]
1147 (0.28,0) ---++(0.06,-0.02) ---++(0.04,0.06) ---+
1148 +(0.0,-0.08) ---++(0.08,0.06) ---++(0.03,-0.02) ---++(0.08,0.02) -- (0.6,0.0);
1149 \draw[line width=0.06ex*\tkzsymbols@scl]
1150 (0.28,0) ---++(0.03,-0.02) ---++(0.03,-0.07) ---+
1151 +(0.03,-0.01) ---++(0.01,-0.07) ---++(0.06,0.01) ---++(0.03,-0.08) --
1152 (0.5,0.-0.25);
1153 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1154 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1155 \end{tikzpicture}%
1156 \fi%
1157 }%
1158 \tikzsymbols@use@box{Walley#1#2#3}%
1159 \tikzsymbolsaftersymbolinput%
1160 }
1161 \tikzsymbols@Declare@Robust@Commandx{rWalley}[3][1=1, 2={none},3={none}, usedefault]{%
1162 \tikzsymbols@ifsaveboxundefined{rWalley#1#2#3}{%
1163 \set@tkzsymbolsscl{#1}%
1164 \iftikzymbols@draftabsolute%
1165 \tkzsymbols@bx@Prmtrstore{#1}%
1166 \tikzsymbols@draftbox{2.341ex*\tkzsymbols@Prmtr}{1.674ex*\tkzsymbols@Prmtr}%

```

```

1167 \else%
1168 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
1169 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1,
1170 decoration={random steps,segment length=0.15ex*\tkzsymls@scl, amplitude=0.1ex*\tkzsymls@scl}]
1171 \filldraw[fill=#2, line width=0.08ex*\tkzsymls@scl] (0,0) circle (0.28);
1172 \filldraw[fill=#3] (0.28,-0.33) rectangle (0.66,0.33);
1173 \foreach \x/\y in { 0.5/0.25 , 0.6/0 , 0.5/-0.25 }
1174 \draw[decorate, line width=0.06ex*\tkzsymls@scl] (0.28,0) -- (\x,\y);
1175 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1176 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0,-0.1);
1177 \end{tikzpicture}%
1178 \fi%
1179 }%
1180 \tikzsymbols@use@box{rWalley#1#2#3}%
1181 \tikzsymbolsaftersymbolinput%
1182 }
1183 \tkzsymls@Declare@Robust@Commandx{dWalley}[2][1=1, 2=yellow, usedefault]{%
1184 \tikzsymbols@ifsaveboxundefined{dWalley#1#2}{%
1185 \set@tkzsymlsscl{#1}%
1186 \iftikzymbols@draftabsolute%
1187 \tkzsymls@bx@Prmtrstore{#1}%
1188 \tikzsymbols@draftbox{2.4288ex*\tkzsymls@Prmtr}{1.6008ex*\tkzsymls@Prmtr}%
1189 \else%
1190 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
1191 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
1192 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
1193 \draw[line width=0.06ex*\tkzsymls@scl]
1194 (0.28,0) ---++(0.05,0.07) ---++(0.03,0.02) ---+
1195 +(0.03,-0.02) ---++(0.03,0.1) ---++(0.03,0.02) -- (0.5,0.25);
1196 \draw[line width=0.06ex*\tkzsymls@scl]
1197 (0.28,0) ---++(0.06,-0.02) ---++(0.04,0.06) ---+
1198 +(0.0,-0.08) ---++(0.08,0.06) ---++(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);
1199 \draw[line width=0.06ex*\tkzsymls@scl]
1200 (0.28,0) ---++(0.03,-0.02) ---++(0.03,-0.07) ---+
1201 +(0.03,-0.01) ---++(0.01,-0.07) ---++(0.06,0.01) ---++(0.03,-0.08) -- (0.5,0.-0.25);
1202 \shade[ball color=#2] (-0.01,0) circle (0.31);
1203 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1204 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0,-0.1);
1205 \end{tikzpicture}%
1206 \fi%
1207 }%
1208 \tikzsymbols@use@box{dWalley#1#2}%
1209 \tikzsymbolsaftersymbolinput%
1210 }
1211 \tkzsymls@Declare@Robust@Commandx{drWalley}[2][1=1, 2=yellow, usedefault]{%
1212 \tikzsymbols@ifsaveboxundefined{drWalley#1#2}{%
1213 \set@tkzsymlsscl{#1}%
1214 \iftikzymbols@draftabsolute%
1215 \tkzsymls@bx@Prmtrstore{#1}%
1216 \tikzsymbols@draftbox{2.4288ex*\tkzsymls@Prmtr}{1.6008ex*\tkzsymls@Prmtr}%

```

```

1217 \else%
1218 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
1219 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1,
1220 decoration={random steps,segment length=0.15ex*\tkzsymls@scl, amplitude=0.1ex*\tkzsymls@scl}]
1221 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
1222 \foreach \x/\y in { 0.5/0.25 , 0.6/0 , 0.5/-0.25 }
1223 \draw[decorate, line width=0.06ex*\tkzsymls@scl] (0.298,0) -- (\x,\y);
1224 \shade[ball color=#2] (-0.01,0) circle (0.31);
1225 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1226 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1227 \end{tikzpicture}%
1228 \fi%
1229 }%
1230 \tikzsymbols@use@box{drWalley#1#2}%
1231 \tikzsymbolsaftersymbolinput%
1232 }

\Cat *Miau*

1233 \tikzsymbols@Declare@Robust@Commandx{Cat}[1][1=1,usedefault]{%
1234 \tikzsymbols@ifsaveboxundefined{Cat#1}{%
1235 \set@tkzsymlsscl{#1}%
1236 \iftikzymbols@draftabsolute%
1237 \tkzsymls@bx@Prmtrstore{#1}%
1238 \tikzsymbols@draftbox{1.899ex*\tkzsymls@Prmtr}{1.957ex*\tkzsymls@Prmtr}%
1239 \else%
1240 \ifdim\tkzsymls@scl<0pt\set@tkzsymlsscl{-#1}\fi%
1241 \begin{tikzpicture}[x=2.33ex,y=2.33ex, line width=0.093ex*\tkzsymls@scl,scale=#1]
1242 \draw (0,0) circle (0.3);
1243 \foreach \xi/\xii in { 0.3/0.35 }
1244 {
1245 \draw[rounded corners=0.163ex*\tkzsymls@scl]
1246 (-\xi,0) -- (-\xii,0.5) -- (0,\xi) (0,\xi) -- (\xii,0.5) -- (\xi,0);
1247 }
1248 \fill (-0.15,.15) circle (0.05);
1249 \fill (0.15,.15) circle (0.05);
1250 \foreach \x in { 0.1 , -0.1 }
1251 \draw[rounded corners=0.175ex*\tkzsymls@scl,yshift=-0.12ex]
1252 (0,0) -- (0,-0.1) -- (\x,-0.095);
1253 \draw[rounded corners=.12ex*\tkzsymls@scl,yshift=-.15ex,
1254 line width=0.03em*0.9*\tkzsymls@scl]
1255 (-0.1,0.1) -- (0,0) -- (0.1,0.1) -- cycle ;
1256 \foreach \a/\b/\c/\d in
1257 { 0.1/0.25/0.35/0.4 , -0.1/-0.25/-0.35/-0.4 }
1258 \foreach \yi/\yii/\yiii in
1259 { 0/0/-0.05 , -0.01/-0.09/-0.14 , -0.045/-0.13/-0.23 }
1260 {
1261 \draw[line width=0.035ex*\tkzsymls@scl]
1262 (\a,-0.05)..controls(\b,\yi)and(\c,\yii).. (\d,\yiii);
1263 }
1264 \end{tikzpicture}%

```

```

1265 \fi%
1266 }%
1267 \tikzsymbols@use@box{Cat#1}%
1268 \tikzsymbolsaftersymbolinput%
1269 }

\Ninja \dNinja A Ninja.

1270 \tikzsymbols@Declare@Robust@Commandx{Ninja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1271 \tikzsymbols@ifsaveboxundefined{Ninja#1#2#3#4}{%
1272 \set@tkzsymbolsscl{#1}%
1273 \iftikzymbols@draftabsolute%
1274 \tkzsymbols@bx@Prmtrstore{#1}%
1275 \tikzsymbols@draftbox{2.149ex*\tkzsymbols@Prmtr}{1.717ex*\tkzsymbols@Prmtr}%
1276 \else%
1277 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1278 \def\Black@is@Black{black}%
1279 \def\Black@or@not@Black{#2}%
1280 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1,
1281 decoration={random steps,segment length=0.1ex*\tkzsymbols@scl, amplitude=0.1ex*\tkzsymbols@scl}]
1282 \fill[#2] (0,0) circle (0.33);
1283 \fill[decoration={random steps,segment length=0.1ex*\tkzsymbols@scl,
1284 amplitude=0.01ex*\tkzsymbols@scl}, decorate,#3]
1285 (-0.33,0) -- (0.33,0) -- (0.23,0.23) -- (-0.23,0.23) -- cycle;
1286 \ifx\Black@or@not@Black\Black@is@Black
1287 \draw[line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.33);\fi
1288 \fill[#3] (0,0.1) -- (-0.33,0) -- (-0.26,0.23);
1289 \fill[#3] (0.3465,0) arc (0:42:0.34 and 0.345) -- (0.2,0.23)-- (0.31,0.0) -- cycle;
1290 \fill[#3] (-0.3465,0) arc (0:-42:-0.34 and -0.345) -- (-0.2,0.23)-- (-0.31,0.0) -- cycle;
1291 \fill[#4] (0.129,0.1425) arc (55:-180:.05);
1292 \fill[#4] (-0.129,0.1425) arc (-55:180:-.05);
1293 \foreach \x in { (0.5,0.35) , (0.53,0.1) }
1294 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymbols@scl,
1295 segment length=0.55ex*\tkzsymbols@scl}, #3]
1296 (0.26,0.21) -- \x;
1297 \ifx\Black@or@not@Black\Black@is@Black
1298 \else\draw[line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.33);\fi
1299 \end{tikzpicture}%
1300 \fi%
1301 }%
1302 \tikzsymbols@use@box{Ninja#1#2#3#4}%
1303 \tikzsymbolsaftersymbolinput%
1304 }

1305 \tikzsymbols@Declare@Robust@Commandx{dNinja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1306 \tikzsymbols@ifsaveboxundefined{dNinja#1#2#3#4}{%
1307 \set@tkzsymbolsscl{#1}%
1308 \iftikzymbols@draftabsolute%
1309 \tkzsymbols@bx@Prmtrstore{#1}%
1310 \tikzsymbols@draftbox{2.1498ex*\tkzsymbols@Prmtr}{1.7178ex*\tkzsymbols@Prmtr}%
1311 \else%
1312 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%

```



```

1313 \def\Black@is@Black{black}%
1314 \def\Black@or@not@Black{#2}%
1315 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1,
1316 decoration={random steps,segment length=0.1ex*\tkzsymbols@scl, amplitude=0.1ex*\tkzsymbols@scl}]
1317 \foreach \length/\coord in { 0.55/{(0.5,0.35)} , 0.5/{(0.53,0.1)} }
1318 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymbols@scl,
1319 segment length=\length ex*\tkzsymbols@scl},decorate, #3!50!black]
1320 \coord -- (0.26,0.21);
1321 \shade[ball color=#2] (0,0) circle (0.347);
1322 \fill[decoration={random steps,segment length=0.1ex*\tkzsymbols@scl,
1323 amplitude=0.01ex*\tkzsymbols@scl},ball color=#3]
1324 decorate {(-0.33,0) -- (0.3465,0) }
1325 {arc (0:42:0.34 and 0.345)}
1326 decorate {-- (-0.25,0.24)}
1327 { arc (-42:0:-0.375 and -0.345)};
1328 \shade[ball color=#4] (0.129,0.1425) arc (55:-180:.05);
1329 \shade[ball color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1330 \shade[top color=#4!80!black, bottom color=#4] (0.129,0.1425) arc (55:-180:.05);
1331 \shade[top color=#4!80!black, bottom color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1332 \end{tikzpicture}%
1333 \fi%
1334 }%
1335 \tikzsymbols@use@box{dNinja#1#2#3#4}%
1336 \tikzsymbolsaftersymbolinput%
1337 }

```

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

```

1338 \tkzsymbols@Declare@Robust@Commandx{NiceReapey}[2][1=1,2={black!20!white},usedefault]{%
1339 \tikzsymbols@ifsaveboxundefined{NiceReapey#1#2}{%
1340 \set@tkzsymbolsscl{#1}%
1341 \iftikzymbols@draftabsolute%
1342 \tkzsymbols@bx@Prmtrstore{#1}%
1343 \tikzsymbols@draftbox{(1.1067em+0.07ex)*\tkzsymbols@Prmtr}
1344 {(0.693em+0.07ex)*\tkzsymbols@Prmtr}%
1345 \else%
1346 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1347 \begin{tikzpicture}[x=0.11em,y=0.11em, line width=0.07ex*\tkzsymbols@scl,scale=#1]
1348 \draw[] (1.7,-1) arc (360:180:1.7 and 2)
1349 arc (260:110:1.5 and 2) .. controls (-1,3.3) and (1,3.3) .. (1.9,2.97)
1350 arc (260:100:-1.3 and -2) -- cycle;
1351 \filldraw[fill=#2] (3,-3) -- (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5) -- cycle;
1352 \draw (0,-1.5) circle (1 and 0.5);
1353 \foreach \x in { 0.2 , 0.6 }
1354 \draw[line width=0.04ex*\tkzsymbols@scl]
1355 (\x,-1) -- (\x,-2) (-\x,-1) -- (-\x,-2) ;
1356 \draw[line width=0.04ex*\tkzsymbols@scl] (-1,-1.5) -- (1,-1.5);
1357 \fill (1.25,1.25) circle ( 0.5 and 0.75);
1358 \fill (-1.25,1.25) circle ( 0.5 and 0.75);
1359 \end{tikzpicture}%
1360 \fi%

```

```

1361 }%
1362 \tikzsymbols@use@box{NiceReapey#1#2}
1363 \tikzsymbolsaftersymbolinput%
1364 }

```

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 ...

```

1365 \newcommand{\tikzsymbols@Strichmaxerl@XCheck}[1]{%
1366 \ifdim #1 pt > 0pt%
1367   \ifdim #1 pt < 0.18pt%
1368     \pgfmathsetmacro{#1}{0}%
1369   \else%
1370     \pgfmathsetmacro{#1}{#1-0.18}%
1371   \fi%
1372 \else%
1373   \ifdim #1 pt >-0.18pt%
1374     \pgfmathsetmacro{#1}{0}%
1375   \else%
1376     \pgfmathsetmacro{#1}{#1+0.18}%
1377   \fi%
1378 \fi%
1379 }

```

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

```

1380 \newcommand{\tikzsymbols@Strichmaxerl@ifSmallerZero}[1]{%
1381 \ifdim #1 pt < 0pt%
1382   \pgfmathsetmacro{#1}{0}%
1383 \fi%
1384 }

```

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

```

1385 \tkzsymls@Declare@Robust@Commandx{Strichmaxerl}[5][1=1,2=-22,3=22,4=27,5=-27,usedefault]{%
1386 \tikzsymbols@ifsaveboxundefined{Strichmaxerl#1#2#3#4#5}{%
1387 \set@tkzsymlsscl{#1}%
1388 \iftikzymbols@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*).

```

1389 \pgfmathsetmacro{\tikzsymbols@x@LA}{-0.27*cos(#2)}%
1390 \pgfmathsetmacro{\tikzsymbols@x@RA}{0.27*cos(#3)}%
1391 \pgfmathsetmacro{\tikzsymbols@x@LB}{0.34*sin(#4)}%
1392 \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*).

```

1393 \pgfmathsetmacro{\tikzsymbols@y@LA}{0.27*sin(#2)}%
1394 \pgfmathsetmacro{\tikzsymbols@y@RA}{-0.27*sin(#3)}%
1395 \pgfmathsetmacro{\tikzsymbols@y@LB}{0.34*cos(#4)}%
1396 \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).

```

1397 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@LA}%
1398 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@RA}%
1399 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@LB}%
1400 \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).

```

1401 \pgfmathsetmacro{\tikzsymbols@x@max}
1402     {max(0,\tikzsymbols@x@LA,\tikzsymbols@x@RA,\tikzsymbols@x@LB,\tikzsymbols@x@RB)}%
1403 \pgfmathsetmacro{\tikzsymbols@x@min}
1404     {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)

```

1405 \pgfmathsetmacro{\tikzsymbols@y@LA}{\tikzsymbols@y@LA-0.2}%
1406 \pgfmathsetmacro{\tikzsymbols@y@RA}{\tikzsymbols@y@RA-0.2}%

```

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

```

1407 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@LA}%
1408 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@RA}%
1409 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@LB}%
1410 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@RB}%

```

And find the greatest number.

```

1411 \pgfmathsetmacro{\tikzsymbols@y@max}
1412 {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:

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

```

1413 \tkzsymls@bx@Prmtrstore{#1}%
1414 \tikzsymbols@draftbox{(0.606ex+1.35ex*\tikzsymbols@x@max
1415   -1.35ex*\tikzsymbols@x@min)*\tkzsymls@Prmtr}%
1416 {(1.173ex+1.35ex*\tikzsymbols@y@max)*\tkzsymls@Prmtr}%
1417 \else%
1418 \ifdim\tkzsymls@scl<0pt\set\tkzsymls@scl{-#1}\fi%
1419 \begin{tikzpicture}[line width=0.12ex*\tkzsymls@scl, scale=#1, x=1.35ex, y=1.35ex]
1420 \draw[rotate around={#5:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1421 \draw[rotate around={#4:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1422 \draw (.15,.2) -- (.15,.4);
1423 \draw[rotate around={#3:(.15,.4)}] (0.15,0.4) -- (0.42,0.4);
1424 \draw[rotate around={#2:(.15,.4)}] (0.15,0.4) -- (-0.12,0.4);
1425 \draw (.15,.4) -- (.15,.53);
1426 \draw (.15,.8) circle (0.18);
1427 \end{tikzpicture}%
1428 \fi%
1429 }%
1430 \tikzsymbols@use@box{Strichmaxerl#1#2#3#4#5}%
1431 \tikzsymbolsaftersymbolinput%
1432 }
1433 \tkzsymls@Declare@Robust@Command{Person}{%
1434 \PackageWarning{tikzsymbols}{Command '\protect\Person' is obsolete,
1435 \MessageBreak Please use '\protect\Strichmaxerl' instead.\MessageBreak}%
1436 \Strichmaxerl%
1437 }

```

\Candle A burning candle

```

1438 \tkzsymls@Declare@Robust@Command{Candle}[1][1]{%
1439 \tikzsymbols@ifsaveboxundefined{Candle#1}{%
1440 \set\tkzsymls@scl{#1}%
1441 \iftikzsymls@draftabsolute%
1442 \tkzsymls@bx@Prmtrstore{#1}%
1443 \tikzsymbols@draftbox{0.64ex*\tkzsymls@Prmtr}{(1.255ex+2.2pt)*\tkzsymls@Prmtr}%
1444 \else%
1445 \ifdim\tkzsymls@scl<0pt\set\tkzsymls@scl{-#1}\fi%
1446 \begin{tikzpicture}[x=1ex, y=1ex, scale=#1, line width=0.07ex*\tkzsymls@scl]
1447 \draw[rounded corners=0.04ex*\tkzsymls@scl] (0,0) -- (0.2,0) -- +(0,1) -- (0,1) -- cycle;
1448 \draw[line width=0.05ex*\tkzsymls@scl] (0.1,1) -- (0.1,1.2);
1449 \foreach \x in { -0.4 , 0.2 }
1450 \draw[xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymls@scl]
1451 (-0.1,0.6) .. controls (\x,0.8) and (-0.1,1) .. (-0.1,1.2);
1452 \end{tikzpicture}%

```

```

1453 \fi%
1454 }%
1455 \tikzsymbols@use@box{Candle#1}%
1456 \tikzsymbolsaftersymbolinput%
1457 }

\Fire Just a fire.

1458 \tikzsymbols@Declare@Robust@Command{Fire}[1][1]{%
1459 \tikzsymbols@ifsaveboxundefined{Fire#1}{%
1460 \set@tkzsymbolsscl{#1}%
1461 \iftikzymbols@draftabsolute%
1462 \tkzsymbols@bx@Prmtrstore{#1}%
1463 \tikzsymbols@draftbox{1.576ex*\tkzsymbols@Prmtr}{1.639ex*\tkzsymbols@Prmtr}%
1464 \else%
1465 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1466 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.07ex*\tkzsymbols@scl,rotate=45, scale=#1]
1467 \fill (-0.05,0) -- (0.05,0) -- (0.05,0.95) -- (-0.05,0.95) -- cycle;
1468 \fill (-0.74,0.7) -- (0.19,0.7) -- (0.19,0.8) -- (-0.74,0.8) -- cycle;
1469 \fill[rotate=-20, xshift=-1.3, yshift=-0.1]
1470 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1471 \fill[rotate=-70, xshift=-3.3, yshift=-2.3]
1472 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1473 \fill[rotate=135, xshift=2.5, yshift=-3.8]
1474 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1475 \draw[rotate=-45, xshift=-2.6, yshift=1.5,line width=0.04ex*\tkzsymbols@scl, x=0.5ex, y=0.5ex]
1476 (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0.05,1.7);
1477 \draw[rotate=-45, xshift=-2.1,yshift=1.5,line width=0.04ex*\tkzsymbols@scl, x=0.5ex, y=0.5ex]
1478 (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.15,1.7);
1479 \draw[rotate=-45, xshift=-2.5] (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0,1.5);
1480 \draw[rotate=-45, xshift=-2] (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.1,1.5);
1481 \end{tikzpicture}%
1482 \fi%
1483 }%
1484 \tikzsymbols@use@box{Fire#1}%
1485 \tikzsymbolsaftersymbolinput%
1486 }

\Coffeecup Just a cup of coffee.

1487 \ifKV\tikzsymbols@marvosym \else%
1488 \tikzsymbols@Declare@Robust@Command{Coffeecup}[1][1]{%
1489 \tikzsymbols@ifsaveboxundefined{Coffeecup#1}{%
1490 \set@tkzsymbolsscl{#1}%
1491 \iftikzymbols@draftabsolute%
1492 \tkzsymbols@bx@Prmtrstore{#1}%
1493 \tikzsymbols@draftbox{1.82ex*\tkzsymbols@Prmtr}{1.705ex*\tkzsymbols@Prmtr}%
1494 \else%
1495 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1496 \begin{tikzpicture}[x=0.7ex,y=0.7ex, scale=#1, line width=0.07ex*\tkzsymbols@scl,
1497 decoration={snake,amplitude=.05ex*\tkzsymbols@scl,segment length=0.408ex*\tkzsymbols@scl}]
1498 \draw (0,0) arc (180:270:0.8 and 1) -- ++(0.5,0) arc (270:360:0.8 and 1) -- cycle;

```

```

1499 \draw (2.1,-0.15) -- (2.2,-0.15) arc (90:-90:0.3) -- (1.85, -0.75);
1500 \foreach \x in {0.4,1,1.6}
1501 \draw[line width=0.05ex*\tkzsymls@scl, decorate]
1502   (\x,0.3) -- +(0,1);
1503 \draw (0,-1.05) -- (2.1,-1.05);
1504 \end{tikzpicture}%
1505 \fi%
1506 }%
1507 \tikzsymbols@use@box{Coffeecup#1}%
1508 \tikzsymbolsaftersymbolinput%
1509 }%

```

\Chair A chair.

```

1510 \tikzsymbols@Declare@Robust@Command{Chair}[1][1]{%
1511 \tikzsymbols@ifsaveboxundefined{Chair#1}{%
1512 \set@tkzsymls@scl{#1}%
1513 \iftikzsymbols@draftabsolute%
1514 \tkzsymls@bx@Prmtrstore{#1}%
1515 \tikzsymbols@draftbox{0.97ex*\tkzsymls@Prmtr}{1.69ex*\tkzsymls@Prmtr}%
1516 \else%
1517 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\@tkzssmbls@negtrue\fi%
1518 \begin{tikzpicture}[x=0.9ex,y=0.9ex, scale=#1, line width=0.07ex*\tkzsymls@scl]
1519 \draw (0,-0.5) -- (0,0.7) -- (0.5,1) -- (0.5,0.25);
1520 \draw[line width=0.06ex*\tkzsymls@scl] (0,0.4) -- (0.5,0.7);
1521 \draw (0,0) -- (0.5,0.3) -- (1,0) --(1,-0.5);
1522 \if@tkzssmbls@neg\draw (0.5,0.3) -- +(0,-0.5);\fi%
1523 \draw (0.5,-0.3) -- (0.5,-0.8);
1524 \draw (1,0) -- (0.5,-0.3) -- (0,0);
1525 \end{tikzpicture}%
1526 \fi%
1527 }%
1528 \tikzsymbols@use@box{Chair#1}%
1529 \@tkzssmbls@negfalse%
1530 \tikzsymbolsaftersymbolinput%
1531 }

```

\Bed A bed.

```

1532 \tikzsymbols@Declare@Robust@Command{Bed}[1][1]{%
1533 \tikzsymbols@ifsaveboxundefined{Bed#1}{%
1534 \set@tkzsymls@scl{#1}%
1535 \iftikzsymbols@draftabsolute%
1536 \tkzsymls@bx@Prmtrstore{#1}%
1537 \tikzsymbols@draftbox{3.08ex*\tkzsymls@Prmtr}{1.68ex*\tkzsymls@Prmtr}%
1538 \else%
1539 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1540 \begin{tikzpicture}[x=1ex,y=1ex, scale=#1, line width=0.08ex*\tkzsymls@scl]
1541 \draw (0,0) -- (0,1.6);
1542 \draw (3,0) -- (3,1.2);
1543 \draw (0,0.5) -- (3,0.5);
1544 \draw (0,0.35) -- (3,0.35);

```

```

1545 \draw (0.7,0.5) arc (0:90:0.7);
1546 \draw (0.7,0.5) arc(180:30:1.231 and 0.6);
1547 \end{tikzpicture}%
1548 \fi%
1549 }%
1550 \tikzsymbols@use@box{Bed#1}%
1551 \tikzsymbolsaftersymbolinput%
1552 }

```

\Tribar Also called Penrose-Triangle.

```

1553 \tikzsymbols@Declare@Robust@Commandx{Tribar}[4]
1554 [1=1,2={opacity=0},3={opacity=0},4={opacity=0},usedefault]
1555 {%
1556 \tikzsymbols@ifsaveboxundefined{Tribar#1#2#3#4}{%
1557 \set@tkzsymbolsscl{#1}%
1558 \iftikzymbols@draftabsolute%
1559 \tkzsymbols@bx@Prmtrstore{#1}%
1560 \tikzsymbols@draftbox{1.7175ex*\tkzsymbols@Prmtr}{1.685ex*\tkzsymbols@Prmtr}%
1561 \else%
1562 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{#1}\fi%
1563 \begin{tikzpicture}[x=0.65ex,y=0.65ex,scale=#1,
1564   rounded corners=0.03ex*\tkzsymbols@scl, line width=0.06ex*\tkzsymbols@scl]
1565 \fill[#2] (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1566   -- (-0.65,-0.6) -- (0.35,1.3) -- +(0.15,-0.3);
1567 \fill[#3] (0,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- (-0.1,-0.3);
1568 \fill[#4] (1,0) -- (0.35,1.3) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (1.3,0);
1569 \draw (0,0) -- (1,0) -- (0.5,1) -- cycle;
1570 \draw (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1571   -- (-0.65,-0.6) -- (0.35,1.3) -- (0.8,.4);
1572 \draw (0.9,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- +(-0.05,-0.1);
1573 \draw (-0.6,-0.6) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (0.4,1.9);
1574 \end{tikzpicture}%
1575 \fi%
1576 }%
1577 \tikzsymbols@use@box{Tribar#1#2#3#4}%
1578 \tikzsymbolsaftersymbolinput%
1579 }

```

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

```

1580 \newcommand{\tikzsymbolsMoaitthickness}{}

```

\Moai From the Easter Island: a Moai.

```

1581 \tikzsymbols@Declare@Robust@Commandx{Moai}[1][1=1,usedefault]{%
1582 \tikzsymbols@ifsaveboxundefined{Moai#1}{%
1583 \set@tkzsymbolsscl{#1}%
1584 \tkzsymbols@bx@Prmtrstore{#1}%
1585 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{#1}\fi%
1586 \ifdim\tkzsymbols@scl<2pt%
1587 \def\tikzsymbolsMoaitthickness{0.05ex}%

```

```

1588 \else%
1589 \ifdim \tkzsymls@scl<5pt%
1590 \def\tikzsymbolsMoaithickness{0.035ex}%
1591 \else%
1592 \def\tikzsymbolsMoaithickness{0.03ex}%
1593 \fi\fi%
1594 \iftikzymbols@draftabsolute%
1595 \tikzsymbols@draftbox{(1.001ex+\tikzsymbolsMoaithickness)*\tkzsymls@Prmtr}
1596 {(1.664ex+\tikzsymbolsMoaithickness)*\tkzsymls@Prmtr}%
1597 \else%
1598 \begin{tikzpicture}[x=.13ex, y=.13ex, rounded corners=0.01ex*\tkzsymls@scl, scale=#1,
1599 line width=\tikzsymbolsMoaithickness*\tkzsymls@scl]
1600 \draw (-2.6,-4.25) -- (-2.5,-5.8)
1601 ..controls (-2,-6.8) and (1.5,-6.8) .. (2.2,-5.8) -- (2.4,-3.95);
1602 \draw(-2.5,2.5) .. controls (-2.9,4.6) and (2,5) .. (3.3,2.5) -- (2.9,-3.4)
1603 .. controls (2,-5) and (-4,-5) .. (-3.1,-3) -- cycle;
1604 \draw (-2.5,3) -- (-2,5) .. controls (0,6) and (2,5.8) .. (3.1,4.7) -- (3.3,2.5);
1605 \draw[line width=0.02ex*\tkzsymls@scl]
1606 (-2.2,-1.8) .. controls (-1,-1.3) and (0,-1.7) .. (1,-2);
1607 \draw[line width=0.02ex*\tkzsymls@scl]
1608 (-2.2,-1.8) .. controls (-1,-1) and (0,-1.4) .. (1,-2);
1609 \draw[line width=0.02ex*\tkzsymls@scl]
1610 (-2.2,-1.8) .. controls (-1,-2) and (0,-2) .. (1,-2);
1611 \draw (-0.8,4) .. controls (-0.8,3) and (-0.8,2) .. (-1.6,0.5) -- (-1.8,-0.4)
1612 .. controls (-1,0.2) and (0,0.2) .. (0.6,-0.4) -- (0.7,0.4)
1613 .. controls (0,1) and (0,2) .. (0.8,4);
1614 \draw (-1.8,-0.36) .. controls (-0.5,-0.5) and (0,-0.5) .. (0.6,-0.36);
1615 \draw (3.2,3.5) -- (3.7,3.5) .. controls (3.5,2) and (3.5,2) .. (3.6,-1.5) -- (3,-1.9);
1616 \draw (-2.5,3) .. controls (-2.7,2) and (-3,1) .. (-2.88,-1);
1617 \draw (-2.5,2.8) .. controls (-2,2.5) and (-1,3) .. (-0.8,3.1);
1618 \draw (0.5,3.3) .. controls (1,3) and (1,2.5) .. (3.3,2.4);
1619 \end{tikzpicture}%
1620 \fi%
1621 }%
1622 \tikzsymbols@use@box{Moi#1}%
1623 \tikzsymbolsaftersymbolinput%
1624 }

```

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

```

1625 \tkzsymls@Declare@Robust@Command{Snowman}[1][1]{%
1626 \tikzsymbols@ifsaveboxundefined{Snowman#1}{%
1627 \set@tkzsymls@scl{#1}%
1628 \iftikzymbols@draftabsolute%
1629 \tkzsymls@bx@Prmtrstore{#1}%
1630 \tikzsymbols@draftbox{1.545ex*\tkzsymls@Prmtr}{1.772ex*\tkzsymls@Prmtr}%
1631 \else%
1632 \ifdim \tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1633 \begin{tikzpicture}[x=0.9ex,y=0.9ex,line width=0.07ex*\tkzsymls@scl, scale=#1]
1634 \draw (0,0) circle (0.4 and 0.35);
1635 \draw[line width=0.06ex*\tkzsymls@scl] (0,0.64) circle (0.3 and 0.28);

```



```

1636 \draw[line width=0.05ex*\tkzsymsbls@sc1] (0,1.14) circle (0.2 and 0.2);
1637 \draw[rounded corners=0.1ex*\tkzsymsbls@sc1,line width=0.05ex*\tkzsymsbls@sc1,
1638   rotate around={-30:(0,1.14)}]
1639   (-0.2,1.15) -- ++(0,0.35) -- +(0.4,0) -- (0.2,1.14);
1640 \draw[rounded corners=0.07ex*\tkzsymsbls@sc1,line width=0.05ex*\tkzsymsbls@sc1,
1641   rotate around={-30:(0,1.14)}]
1642   (-0.2,1.19) arc (270:90:0.1);
1643 \foreach \y in { 0.78 , 0.63 , 0.48 }
1644   \fill (0,\y) circle (0.04);
1645 \foreach \y in { 0.2 , 0 , -0.2 }
1646   \fill (0,\y) circle (0.05);
1647 \fill (-0.06,1.18) circle (0.045);
1648 \fill (0.06,1.18) circle (0.045);
1649 \foreach \x/\y in { 0.1/1.08 , 0.06/1.055 , 0.02/1.039}
1650   \fill (\x,\y) circle (0.015) (-\x,\y) circle (0.015) ;
1651 \draw (-0.3,0.7) -- (-0.6,0.8);
1652 \draw (0.3,0.7) -- (0.6,0.8);
1653 \draw[line width=0.06ex*\tkzsymsbls@sc1] (-0.65,0) -- (-0.65,1);
1654 \foreach \x in {-0.85, -0.75,-0.65,-0.55,-0.45}
1655 \draw[line width=0.05ex*\tkzsymsbls@sc1] (-0.65,1) -- (\x,1.3);
1656 \foreach \y/\x/\z in { 0.7/0.75 , 0.9/0.8 , 1/0.6/0.55 }
1657   \draw (0.6,0.8) -- (\x,\y) (-0.6,0.8) -- (-\z,\y) ;
1658 \end{tikzpicture}%
1659 \fi%
1660 }%
1661 \tikzsymbols@use@box{Snowman#1}%
1662 \tikzsymbolsaftersymbolinput%
1663 }

```

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:

```

1664 \tikzsymbolsnewcommand{BasicTree}[5][1]{%
1665 \tikzsymbols@ifsaveboxundefined{BasicTree#1#2#3#4#5}{%
1666 \def\tikzsymbols@leaf@or@not@leaf{#5}%
1667 \ifx\tikzsymbols@leaf@or@not@leaf\tikzsymbols@leaf@is@leaf%
1668 \Basic@Tree{#1}{#2}{#3}{#4}{#5}%
1669 \else%
1670 \ifx\#5\%
1671 \Basic@Tree{#1}{#2}{#3}{#4}{#5}%
1672 \else%
1673 \PackageError{tikzsymbols}{The last\MessageBreak parameter has either to be \MessageBreak
1674 'leaf' or has to be empty}{See the tikzsymbols documentation. Section 'Trees'.}%
1675 \fi\fi%
1676 }%
1677 \tikzsymbols@use@box{BasicTree#1#2#3#4#5}%

```

```

1678 \tikzsymbolsaftersymbolinput%
1679 }

\WorstTree It's not that bad.

1680 \tkzsymls@Declare@Robust@Command{WorstTree}[1][1]{%
1681 \tikzsymbols@ifsaveboxundefined{WorstTree#1}{%
1682 \set@tkzsymls@sc1{#1}%
1683 \iftikzsymbols@draftabsolute%
1684 \tkzsymls@bx@Prmtrstore{#1}%
1685 \tikzsymbols@draftbox{1.64ex*\tkzsymls@Prmtr}{1.84ex*\tkzsymls@Prmtr}%
1686 \else%
1687 \ifdim\tkzsymls@sc1<0pt\set@tkzsymls@sc1{-#1}\fi%
1688 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.04ex*\tkzsymls@sc1,scale=#1]
1689 \fill[brown] (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1690 controls (0.5,0.7) and (0.5,0.3) .. (1,0);
1691 \draw (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1692 controls (0.5,0.7) and (0.5,0.3) .. (1,0) ;
1693 \fill[green] (0.2,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1694 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1695 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) ..
1696 controls (1.1, 1.6) and (1,1.4) .. (1,1.2) ..
1697 controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.5,0.8);
1698 \draw (0.214,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1699 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1700 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) .. controls (1.1, 1.6) and (1,1.4) ..
1701 (1,1.2) .. controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.486,0.8);
1702 \fill[red] (0,1) circle (0.1);
1703 \fill[red] (0.4,1.2) circle (0.1);
1704 \fill[red] (0.8,1.1) circle (0.1);
1705 \end{tikzpicture}%
1706 \fi%
1707 }%
1708 \tikzsymbols@use@box{WorstTree#1}%
1709 \tikzsymbolsaftersymbolinput%
1710 }

\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.

1711 \tkzsymls@Declare@Robust@Commandx{Springtree}[1][1=1, usedefault]%
1712 {\tikzsymbolsuse{BasicTree}[#1]{brown!70!black}{green!90!black}{green!80!black}{leaf}}
1713 \tkzsymls@Declare@Robust@Commandx{Summertree}[1][1=1, usedefault]%
1714 {\tikzsymbolsuse{BasicTree}[#1]{brown!50!black}{green!80!black}{red!80!green}{leaf}}
1715 \tkzsymls@Declare@Robust@Commandx{Autumntree}[1][1=1, usedefault]%
1716 {\tikzsymbolsuse{BasicTree}[#1]{red!30!black}{red!75!black}{orange}{leaf}}
1717 \tkzsymls@Declare@Robust@Commandx{Wintertree}[1][1=1, usedefault]%
1718 {\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.

```

1719 \AtBeginDocument{%
1720 \ifKV@tikzsymbols@marvosym
1721 \@ifpackageloaded{marvosym}{\}%
1722 \PackageError{tikzsymbols}{Use option 'marvosym' only}{\MessageBreak
1723 if you load package 'marvosym'}
1724 {Either load package 'marvosym' or\MessageBreak
1725 delete the tikzsymbols option 'marvosym'}}
1726 \fi
1727 }

```

Well then, happy T_EXing!

Change History

v1.0	General: Initial version 1	v2.0	General: Fixed Bugs, improved BasicTree, new option “marvosym”, new symbol 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	v2.2	General: Now you can use negative scaling. Include <code>\@ifpackageloaded</code> . Did something else, I can’t remember . . . 1
v1.6	General: on/off. 1	v2.5	General: New option: <code>draftabsolute</code> , changed the documentation a bit 1
	Renamed “ <code>tikzsymbolsaftersymbolinput</code> ” to “ <code>tikzsymbolsaftersymbolinput</code> ” 1	v3.0	General: Changed output of “absolute” option 1
	Now “Person” can be used in sections, etc. 1		Changed symbol code 1
	Now an error message is generated if the last parameter of “BasicTree” is neither “leaf” nor empty. 1		Changed the documentation . . . 1
v1.61	General: Made an invisible box in BasicTree. 1		Replaced <code>\let</code> by <code>\tikzsymbols@let</code> 1
v1.65	General: Improved BasicTree; New symbols “Schaler/peeler”, Laughy, Walley, Ninja; but didn’t improve the source-description 1		Using <code>\changes</code> correctly (hopefully) 1
v1.7	General: New symbols, etc. 1		<code>marvosym</code> : using <code>xkeaval</code> ’s boolkey . 11
			<code>\Autumntree</code> : Replaced <code>\BasicTree</code> by <code>\tikzsymbolsuse{BasicTree}</code> 50
			<code>\Basic@Tree</code> : has now the default definition <code>\Basic@Tree@on</code> . . . 10
			<code>\Basic@Tree@ff</code> : Draft rectangle same size as normal tree. Changed parts of the code . . . 15

\Basic@Tree@on: Same here	16	\tikzsymbols@Strichmaxerl@XCheck:	
\cmdKV@tikzsymbols@prefix: New		New macro. Is needed for	
macro.	11	\Strichmaxerl	42
\Cofeecup: Changed angle of arc a		\tikzsymbols@use@box: New	
bit. Changed a length.	45	macro. Prints the savebox . . .	14
\current@tikzsymbols: New		\tikzsymbolsuse: New macro. Us-	
macro. Current script size and		ing symbols without carrying	
text color is stored inside	13	about the prefix	14
\iftikzsymbols@draftabsolute:		\tkzsymls@bx@Prmtrstore:	
New \newif	10	New macro. Makes	
final: using xkeaval's boolkey . . .	11	\tkzsymlsPrmtr positive (if	
using xkeaval's choicekey	11	it is negative)	13
\oldWinkey_\olddWinkey: Old		\tkzsymls@Declare@Robust@Command:	
Winkey	29	New macro. Is needed for option	
draft: using xkeaval's choicekey.		prefix	14
Furthermore new option abso-		\tkzsymls@Declare@Robust@Commandx:	
lute is available	11	New macro. Is needed for option	
draftabsolute: is obsolete	11	prefix	14
tree: Using xkeyval's choicekey. Can		\tkzsymls@Prmtr: New macro.	
be set: on/true, off/false	12	Needed for the plain vanilla	
usebox: New option	11	rectangles	13
\Springtree: Replaced \BasicTree		\tkzsymlsnewcommand: New	
by \tikzsymbolsuse{BasicTree}		macro. Is needed for option	
.	50	prefix	14
\Strichmaxerl: Renamed \Person		\Winkey_\dWinkey: Changed the	
to \Strichmaxerl and added		smile. Old Winkey available as	
great parts of code	42	\oldWinkey and \olddWinkey . . .	29
\Summertree: Replaced \BasicTree		\Wintertree: Replaced \BasicTree	
by \tikzsymbolsuse{BasicTree}		by \tikzsymbolsuse{BasicTree}	
.	50	50
\tikzsymbols@draftbox: New		v3.0a	
macro. Draw a plain vanilla		\Flasche,\bottle: new command:	
rectangle if draft is absolute . .	12	\Flasche	24
\tikzsymbols@draftboxheight:		\Reibe,\grater: new command:	
New length	12	\Reibe	23
\tikzsymbols@draftboxlength:		v3.0b	
New length	12	General: Deleted non ASCII charac-	
\tikzsymbols@draftQbox: New		ters in der .sty file.	1
macro. Short form of		\Basic@Tree@on: Transferred	
\tikzsymbols@draftbox	13	\pgfarrowsdeclare inside the	
\tikzsymbols@ifsaveboxundefined:		tikzpicture environment. It	
New macro. Checks if save-box		seems this command produces	
is undefined, if true: does noth-		an unwanted space.	16
ing; if false: defined new box. . .	13	v3.0c	
\tikzsymbols@let: New macro.		\tkzsymls@bx@Prmtrstore: Added	
\let with prefix	15	a @ sign.	13
\tikzsymbols@Strichmaxerl@ifSmallerZero:		v3.0d	
New macro. Is needed for		General: \tkzsymlsscl to	
\Strichmaxerl. Checks if some-		\tkzsymls@scl	1
thing is smaller than 0	42		

v3.0e	Added \foreach to the code. . .	41
General: \tkzsymsDeclareRobustCommand	\Ninja_\dNinja: Added \foreach	
to \tkzsymsDeclare@Robust@Command	to the code.	40
..... 1	\Nursey_\dNursey: Added	
v3.0f	\filldraw to the code.	35
General: \tkzsymsbxPrmtrstore	Using option none	35
to \tkzsyms@bx@Prmtrstore . 1	\oldWinkey_\olddWinkey: Added	
v3.0g	\filldraw to the code.	29
General: Deleted package calc, us-	Using option none	29
ing now \pgfmathsetlength in-	\Sadey_\dSadey: Added \filldraw	
stead of \setlength.	to the code.	24
10	Added \foreach to the code. .	24
\Annoey_\dAnnoey: Added	Using option none	24
\filldraw to the code.	\Saftpresse,\squeezer: Added	
25	\foreach to the code.	22
Using option none	\Schneebesen,\eggbeater: Added	
25	\foreach to the code.	18
\Backblech,\bakingplate: Added	\Sey_\dSey: Added \filldraw to	
\filldraw to the code.	the code.	30
20	Using option none	30
Added \foreach to the code. .	\Sieb,\sieve: Added \foreach to	
20	the code.	19
\Bratpfanne,\fryingpan: Added	\Smiley_\dSmiley: Added	
\foreach to the code.	\filldraw to the code.	26
17	Using option none	26
\Candle: Added \foreach to the	\Snowman: Added \foreach to the	
code.	code.	48
44	\tikzsymbols@leaf@is@leaf: Re-	
\Cat: Added \foreach to the code.	named \@leaf@ist@leaf	12
39	\Tongey_\dTongey: Added	
\Cofeecup: Added \foreach to the	\filldraw to the code.	34
code.	Using option none	34
45	\Vomey_\dVomey: Added \filldraw	
\Cooley_\dCooley: Added	to the code.	36
\filldraw to the code.	Using option none	36
33	\Walley_\dWalley,\rWalley,\drWalley:	
Added \foreach to the code. .	Added \filldraw to the code.	37
33	Added \foreach to the code. .	37
Using option none	Using option none	37
33	\Xey_\dXey: Added \filldraw to	
\Flasche,\bottle: Using option	the code.	31
transparent.	Added \foreach to the code. .	31
24	Using option none	31
\Herd,\cooker: Added \foreach to		
the code.		
21		
\Innocey_\dInnocey: Added		
\filldraw to the code.		
32		
Using option none		
32		
\Laughy_\dLaughy: Added		
\filldraw to the code.		
27		
Using option none		
27		
\Neutrey_\dNeutrey: Added		
\foreach to the code.		
28		
Using option none		
28		
\NiceReapey: Added \filldraw to		
the code.		
41		

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	C	\endcsname 77,
\@BackblechlochX 360, 362	\c 198, 1256, 1262	82, 86, 92, 101,
\@BackblechlochY 361, 362	\Candle <u>1438</u>	103, 106, 108,
\@Ofenschalter 381, 382	\Cat <u>1233</u>	111, 113, 116,
\@Tree@SetUp . . <u>128</u> , 174	\Chair <u>1510</u>	119, 122, 125, 126
\@gobble 98	\cmdKV@tikzsymbols@prefix	\ErrorMessage . . . <u>1719</u>
\@ifpackageloaded <u>39</u> , 101,	\expandafter
. 4–8, 1721	103, 106, 108,	. 77, 80, 81, 83–
\@secondoftwo 97	111, 113, 116,	85, 90, 91, 101,
\@tkzssmbles@negfalse	119, 122, 124, 126	102, 106, 107,
. 203, 1529	\Cofeecup <u>1487</u>	111, 112, 122–125
\@tkzssmbles@negtrue	\Cooley\@dCooley . . <u>936</u>	F
. 172, 1517	\coord 1317, 1320	\f@size 76
\@ 1670	\coordinate 196	\final <u>13</u>
A	\csname 82, 86,	\Fire <u>1458</u>
\a 197, 1256, 1262	92, 101, 103, 106,	\Flasche,\bottle . . <u>524</u>
\Annoey\@dAnnoey . . <u>582</u>	108, 111, 113,	\foreach 253, 272, 293,
\AtBeginDocument . 1719	119, 122, 124, 126	295, 297, 357,
\Autummtree <u>1711</u>	\current@color . . . 76	360, 361, 381,
B	\current@tikzsymbols	425, 449, 514,
\b 197, 1256, 1262 <u>74</u> ,	515, 862, 881,
\Backblech,\bakingplate	76, 77, 82, 86, 92	946, 972, 1173,
. <u>347</u>	\CurrentOption . . . 32	1222, 1243, 1250,
\Basic@Tree	D	1256, 1258, 1293,
. <u>12</u> , 15, 17, 23,	\d 198, 1256, 1262	1317, 1353, 1449,
25, 27, 44, 46, 48,	\def 15, 17, 23,	1500, 1643, 1645,
50, <u>137</u> , 1668, 1671	25, 27, 44, 46, 48,	1649, 1654, 1656
\Basic@Tree@ff <u>137</u>	50, 101, 106, 111,	\frame 63, 65
\Basic@Tree@off . . .	122, 128, 142,	G
. 17, 25,	173, 1278, 1279,	\global 83
27, 48, 50, 137, 169	1313, 1314, 1587,	H
\Basic@Tree@on . 12,	1590, 1592, 1666	\hbox 66
15, 23, 44, 46, <u>167</u>	\define@boolkey . . .	\Herd,\cooker <u>415</u>
\BasicTree <u>1664</u> 13, 36, 37	I
\Bed <u>1532</u>	\define@choicekey 20, 41	\if@tkzssmbles@neg . .
\Black@is@Black 1278,	\define@cmdkey . . . 40 <u>10</u> , 1522
1286, 1297, 1313	\define@key 30	\ifcase 22, 43
\Black@or@not@Black	\draft <u>20</u>	\ifcsname 77, 116
. 1279,	\draftabsolute . . . 30	\ifKV@tikzsymbols@final
1286, 1297, 1314	\Dreizack,\trident . <u>326</u> 14
\Bratpfanne,\fryingpan	E	
. <u>237</u>	\edef 72, 76, 140	

878, 892, 897,	\tikzsymbols@draftboxheight	.. 1365, 1397–1400
915, 920, 938, 58, 62, 65	\tikzsymbols@Tree@absolute@scale
943, 961, 966,	\tikzsymbols@draftboxlength 139, 140
984, 989, 1006, 58, 61, 66	\tikzsymbols@tree@nr
1011, 1029, 1034,	\tikzsymbols@draft@box 42, 43
1054, 1059, 1080, 69,	\tikzsymbols@tree@val
1085, 1110, 1115,	549, 568, 587, 42
1134, 1139, 1163,	606, 626, 646,	\tikzsymbols@use@box
1168, 1185, 1190,	665, 685, 705,	89, 98, 233, 258,
1213, 1218, 1235,	724, 743, 762,	279, 303, 322,
1240, 1272, 1277,	781, 800, 819,	343, 366, 388,
1307, 1312, 1340,	838, 857, 876,	411, 433, 458,
1346, 1387, 1418,	941, 964, 987, 1009	478, 499, 520,
1440, 1445, 1460,	\tikzsymbols@ifsaveboxundefined	540, 560, 579,
1465, 1490, 1495,	75, 97, 207, 238,	598, 617, 637,
1512, 1517, 1534,	263, 284, 307,	657, 677, 697,
1539, 1557, 1562,	327, 348, 371,	716, 735, 754,
1583, 1585, 1627,	393, 416, 438,	773, 792, 811,
1632, 1682, 1687	463, 483, 504,	830, 849, 868,
\Sey_\dSey	525, 545, 564,	887, 909, 933,
\shadedraw	583, 602, 622,	956, 979, 1001,
\Sieb,\sieve	642, 661, 681,	1023, 1049, 1075,
\Smiley_\dSmiley ..	701, 720, 739,	1105, 1129, 1158,
\Snowman	758, 777, 796,	1180, 1208, 1230,
\Springtree	815, 834, 853,	1267, 1302, 1335,
\Strichmaxerl	872, 891, 914,	1362, 1430, 1455,
\Summertree	937, 960, 983,	1484, 1507, 1528,
	1005, 1028, 1053,	1550, 1577, 1622,
	1079, 1109, 1133,	1661, 1677, 1708
	1162, 1184, 1212,	\tikzsymbols@x 253, 254
\tikzset	1234, 1271, 1306,	\tikzsymbols@x@LA ..
\tikzsymbols@draft@nr	1339, 1386, 1439, 1389,
..... 21, 22	1459, 1489, 1511,	1397, 1402, 1404
\tikzsymbols@draft@val	1533, 1556, 1582,	\tikzsymbols@x@LB ..
..... 21	1626, 1665, 1681 1391,
\tikzsymbols@draftbox	\tikzsymbols@leaf@is@leaf	1399, 1402, 1404
60, 69, 145, 148, 55,	\tikzsymbols@x@max ..
211, 242, 267,	144, 153, 187, 1667 1401, 1414
288, 311, 331,	\tikzsymbols@leaf@or@not@leaf	\tikzsymbols@x@min ..
352, 375, 397,	. 142, 144, 153, 1403, 1415
420, 442, 467,	173, 187, 1666, 1667	\tikzsymbols@x@RA ..
487, 508, 529,	\tikzsymbols@let 1390,
895, 918, 1032,	. 121, 236, 261,	1398, 1402, 1404
1057, 1083, 1113,	282, 305, 325,	\tikzsymbols@x@RB ..
1137, 1166, 1188,	346, 369, 391, 1392,
1216, 1238, 1275,	414, 436, 461,	1400, 1402, 1404
1310, 1343, 1414,	481, 502, 523, 543	\tikzsymbols@ifSmiley@LA ..
1443, 1463, 1493,	\tikzsymbols@Strichmaxerl@ifSmiley@LA 1393,
1515, 1537, 1560,	.. 1380, 1407–1410	\tikzsymbols@Strichmaxerl@XCheck1405, 1407, 1412
1595, 1630, 1685		

<code>\tikzsymbols@y@LB ..</code>	742, 761, 780,	800, 819, 838,
.. 1395, 1409, 1412	799, 818, 837,	857, 876, 895,
<code>\tikzsymbols@y@max ..</code>	856, 875, 894,	918, 941, 964,
..... 1411, 1416	917, 940, 963,	987, 1009, 1032,
<code>\tikzsymbols@y@RA ..</code>	986, 1008, 1031,	1057, 1083, 1113,
..... 1394,	1056, 1082, 1112,	1137, 1166, 1188,
1406, 1408, 1412	1136, 1165, 1187,	1216, 1238, 1275,
<code>\tikzsymbols@y@RB ..</code>	1215, 1237, 1274,	1310, 1343, 1344,
.. 1396, 1410, 1412	1309, 1342, 1413,	1415, 1416, 1443,
<code>\tikzsymbolsaftersymbolinput</code>	1442, 1462, 1492,	1463, 1493, 1515,
.. <u>54</u> , 234, 259,	1514, 1536, 1559,	1537, 1560, 1595,
280, 304, 323,	1584, 1629, 1684	1596, 1630, 1685
344, 367, 389,	<code>\tkzsymls@Declare@Robust@Command</code>	<code>\tkzsymls@Declare@Robust@Command</code>
412, 434, 459, <u>100</u> ,	72, 140, 141, 152,
479, 500, 521,	206, 237, 262,	172, 188, 192,
541, 561, 580,	283, 306, 326,	213–215, 221,
599, 618, 638,	347, 370, 392,	244–248, 250,
658, 678, 698,	415, 437, 462,	252, 254, 269,
717, 736, 755,	482, 503, 524,	271, 274, 290–
774, 793, 812,	912, 1433, 1438,	292, 313–316,
831, 850, 869,	1458, 1488, 1510,	333–335, 338,
888, 910, 934,	1532, 1625, 1680	354–356, 358,
957, 980, 1002,	<code>\tkzsymls@Declare@Robust@Command</code>	377, 378, 384,
1024, 1050, 1076, <u>105</u> , 544,	399–401, 405,
1106, 1130, 1159,	563, 582, 601,	406, 422–424,
1181, 1209, 1231,	621, 641, 660,	444–446, 450,
1268, 1303, 1336,	680, 700, 719,	452–454, 469–
1363, 1431, 1456,	738, 757, 776,	471, 489–491,
1485, 1508, 1530,	795, 814, 833,	495, 510, 511,
1551, 1578, 1623,	852, 871, 890,	531–533, 551–
1662, 1678, 1709	913, 936, 959,	553, 570, 571,
<code>\tikzsymbolsMoaitthickness</code>	982, 1004, 1026,	589–591, 608,
..... <u>1580</u> ,	1052, 1078, 1108,	609, 628, 629,
1587, 1590, 1592,	1132, 1161, 1183,	648, 649, 667–
1595, 1596, 1599	1211, 1233, 1270,	669, 672, 687–
<code>\tikzsymbolsuse ...</code>	1305, 1338, 1385,	689, 692, 707–
.... <u>115</u> , 1712,	1553, 1581, 1711,	709, 726, 727,
1714, 1716, 1718	1713, 1715, 1717	745, 746, 764,
<code>\tikzymbols@draftabsolutet</code>	<code>\tkzsymls@Prmtr</code>	<u>70</u> ,
..... 27, 35	140, 145, 146,	765, 783, 784,
<code>\tkzsymls@bx@Prmtrstore</code>	148, 149, 211,	802, 803, 821–
..... <u>70</u> , 210,	242, 267, 288,	823, 840, 841,
241, 266, 287,	311, 331, 352,	859–861, 878–
310, 330, 351,	375, 397, 420,	880, 897, 898,
374, 396, 419,	442, 467, 487,	903–905, 920,
441, 466, 486,	508, 529, 549,	921, 926, 928,
507, 528, 548,	568, 587, 606,	929, 943, 944,
567, 586, 605,	626, 646, 665,	947, 966, 967,
625, 645, 664,	685, 705, 724,	973, 989, 990,
684, 704, 723,	743, 762, 781,	994, 996, 1011,
		1012, 1016, 1018,

1034–1036, 1041,	1605, 1607, 1609,	299, 514, 516,
1043, 1045, 1059,	1632, 1633, 1635–	1173, 1174, 1222,
1060, 1064, 1066,	1637, 1640, 1653,	1223, 1250, 1252,
1068, 1070, 1085–	1655, 1687, 1688	1293, 1296, 1353,
1088, 1098, 1115–	<code>\tkzsymsblsnewcommand</code>	1355, 1449, 1451,
1117, 1119, 1139– <u>110</u> , 1664	1500, 1502, 1649,
1142, 1144, 1146,	<code>\Tongey_\dTongey</code> .. <u>982</u>	1650, 1654–1657
1149, 1168–1171,	<code>\tree</code> <u>41</u>	<code>\Xkey_\dXkey</code> <u>852</u>
1174, 1190, 1191,	<code>\Tribar</code> <u>1553</u>	<code>\xI</code> 357, 359
1193, 1196, 1199,	<code>\typeout</code> 31, 34	<code>\xi</code> 449, 451,
1218–1220, 1223,		862, 863, 881,
1240, 1241, 1245,	U	882, 946, 948,
1251, 1253, 1254,	<code>\usebox</code> <u>37</u> , 90	972, 974, 1243, 1246
1261, 1277, 1280,	<code>\usetikzlibrary</code> 9	<code>\xII</code> 357, 359
1281, 1283, 1284,	V	<code>\xii</code> 449, 451, 946, 948,
1287, 1294, 1295,	<code>\vbadness</code> 64	972, 974, 1243, 1246
1298, 1312, 1315,	<code>\vbox</code> 65	Y
1316, 1318, 1319,	<code>\Vomey_\dVomey</code> ... <u>1078</u>	<code>\y</code> 297–299, 425, 426,
1322, 1323, 1346,		515, 516, 1173,
1347, 1354, 1356,	W	1174, 1222, 1223,
1418, 1419, 1445–	<code>\Walley_\dWalley, \rWalley, \drWalley</code>	1643–1646, 1649,
1448, 1450, 1465, <u>1132</u>	1650, 1656, 1657
1466, 1475, 1477,	<code>\Winkey_\dWinkey</code> .. <u>738</u>	<code>\yi</code> 1258, 1262
1495–1497, 1501,	<code>\Wintertree</code> <u>1711</u>	<code>\yii</code> 1258, 1262
1517, 1518, 1520,	<code>\WorstTree</code> <u>1680</u>	<code>\yiii</code> 1258, 1262
1539, 1540, 1562,		
1564, 1585, 1586,	X	
1589, 1598, 1599,	<code>\x</code> 272, 273,	Z
	295, 296, 298,	<code>\z</code> 1656, 1657