

# tikzsymbols\*

Ben Vitecek  
[b.vitecek@gmx.at](mailto:b.vitecek@gmx.at)

October 31, 2014

## Abstract

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

## Contents

<b>1</b>	<b>Short Introduction</b>	<b>2</b>
<b>2</b>	<b>Options</b>	<b>2</b>
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
<b>3</b>	<b>Symbols</b>	<b>4</b>
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
<b>4</b>	<b>Warnings and Errors</b>	<b>9</b>
4.1	Warnings . . . . .	9
4.2	...and errors . . . . .	10
<b>5</b>	<b>Nobody is perfect</b>	<b>10</b>

---

\*This document corresponds to tikzsymbols v3.0f, dated 2014/10/31.

<b>6</b>	<b>Code</b>	<b>10</b>
6.1	Cookingsymbolcode	17
6.2	Emoticonscode	25
6.3	Other symbols(s)	43
6.4	Trees	51

## 1 Short Introduction

There are about two emoticons available in L<sup>A</sup>T<sub>E</sub>X: Smiley and Frowny. But why aren't there more? Or why did nobody make cooking-symbols<sup>1</sup>? 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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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  $\square$ , `\Nursey`  $\square$ , `\BasicTree{red}{red!50!black}{black}{leaf}`  $\square$ , `\Schussel`  $\square$ , etc.

You see, they are *very* plain and *very* vanilla (but L<sup>A</sup>T<sub>E</sub>X needs no time to produce them).<sup>2</sup>

### 2.3 final=true/false

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

<sup>1</sup>Well, there are some, but not the one I wanted.

<sup>2</sup>The 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<sup>A</sup>T<sub>E</sub>X 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`<sup>3</sup> and are used with `\usebox`. The advantage is that L<sup>A</sup>T<sub>E</sub>X doesn’t need to recalculate the symbol again<sup>4</sup> (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.

<sup>3</sup>Inside a `\sbox` to be correctly.

<sup>4</sup>I 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  $x$ .

Furthermore since v3.0 this package uses a savebox-usebox system. That means the output of a symbol is saved inside a box<sup>5</sup> using `\sbox` and every time you use the *exactly* same symbol, L<sup>A</sup>T<sub>E</sub>X 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.

☺, ☺, ☺ and ☺ 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 calculation<sup>6</sup> (I know I am repeating myself). This means, you can use for example `\Summertree` many times without having to wait till L<sup>A</sup>T<sub>E</sub>X 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.
















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

<sup>6</sup>And 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) 1400<sup>7</sup>, default is 1.

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

German & English Commands	Optional parameter(s)	Output
\Kochtopf    \pot	[ $\langle scale \rangle$ ]	
\Bratpfanne    \fryingpan	[ $\langle scale \rangle$ ]	
\Schneebeesen    \eggbeater	[ $\langle scale \rangle$ ]	
\Sieb    \sieve	[ $\langle scale \rangle$ ]	
\Purierstab    \blender <sup>8</sup>	[ $\langle scale \rangle$ ]	
\Dreizack    \trident	[ $\langle scale \rangle$ ]	
\Backblech    \bakingplate	[ $\langle scale \rangle$ ]	
\Ofen    \oven	[ $\langle scale \rangle$ ]	
\Pfanne    \pan	[ $\langle scale \rangle$ ]	
\Herd    \cooker	[ $\langle scale \rangle$ ]	
\Saftpresse    \squeezer	[ $\langle scale \rangle$ ]	
\Schussel    \bowl	[ $\langle scale \rangle$ ]	
\Schaler    \peeler	[ $\langle scale \rangle$ ]	
\Reibe    \grater	[ $\langle scale \rangle$ ]	
\Flasche    \bottle	[ $\langle scale \rangle$ ]	

## 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) 2000<sup>9</sup>, 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
\Sadey	[ $\langle scale \rangle$ ] [ $\langle color \rangle$ ]	☹
\Smiley	[ $\langle scale \rangle$ ] [ $\langle color \rangle$ ]	☺
\Laughey	[ $\langle scale \rangle$ ] [ $\langle color \rangle$ ] [ $\langle mouth color \rangle$ ]	☺
\Annoey	[ $\langle scale \rangle$ ] [ $\langle color \rangle$ ]	☹

<sup>7</sup>Since version 2.2 you can use negative numbers as well (see examples)

<sup>8</sup>I know that “Purierstab” should be translated as “immersion blender”, but I'm just using “blender”

<sup>9</sup>Do 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>\rWalley<sup>10</sup></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 number<sup>11</sup> and a large number<sup>12</sup>, 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>	☹

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

<sup>11</sup>under -500 for sure

<sup>12</sup>over 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> <sup>13</sup>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle wall color \rangle]</code>	
<code>\dNinja</code>	<code>[\langle scale \rangle] [\langle color \rangle] [\langle headband color \rangle] [\langle eye color \rangle]</code>	

Examples: `\dSadey [] [red]` `\dCooley [-3] [cyan]`

`\dVomey [1.5] [green!70!black] [olive]`

`\dNursey [] [yellow] [blue] [red]`

`\dNinja [1.3] [] [violet] [red]`

### 3.3 other Symbol(s)

`\Strichmaxerl`'s optional parameters 2–5 (*left arm* to *right leg*) can be a number between -360 and 360<sup>14</sup>. These parameters are the angles between the body and the separate parts of `\Strichmaxerl` (see examples).

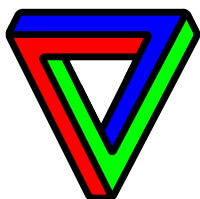
`\langle scale \rangle` can be a very great and a very small number (but I don't think, that you need so large symbols).

`\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>	

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

<sup>14</sup>Of 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)  $900$ <sup>15</sup>, default is 1.


`<color>` can be every defined color (see examples below). Note: The color names shouldn’t contain special characters like  $\beta$ ,  $\ddot{a}$ ,  $\ddot{o}$ , ...


`{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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X faster.


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

<sup>15</sup>if it is larger (or less) it uses too much of L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X generates an error-message because `\Smiley` has already been defined.

If you load `marvosym` *before* `tikzsymbols`, `tikzsymbols` will overwrite `marvosym`'s `Smiley` and no error-message is generated (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 description would be nice.

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

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

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

```
1 \NeedsTeXFormat{LaTeX2e}[2011/06/27]
2 \ProvidesPackage{tikzsymbols}
3 [2014/10/31 v3.0f Some symbols created using tikz.]
4 \@ifpackageloaded{tikz}{\RequirePackage{tikz}}
5 \@ifpackageloaded{xargs}{\RequirePackage{xargs}}
6 \@ifpackageloaded{xcolor}{\RequirePackage{xcolor}}
7 \@ifpackageloaded{xkeyval}{\RequirePackage{xkeyval}}
8 \@ifpackageloaded{xspace}{\RequirePackage{xspace}}
9 \@ifpackageloaded{calc}{\RequirePackage{calc}}
```

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

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

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

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

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

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

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

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

```

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

```

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

If option `draft=absolute` is set, we set `\iftikzsymbols@draftabsolute` to `true` and *all* symbols are replaced by plain vanilla rectangles drawn by L<sup>A</sup>T<sub>E</sub>X.

```

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

```

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

```

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

```

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

```

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

```

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

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

```

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

```

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

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

```

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

```

`tree` Now we declare the name of our option: `tree` (I could have named it `stone`, or `wood`, etc., but I used “`tree`”). I am using `xkeyval` now more than before and `tree` can now be set to `on/true` resp. `off/false`.

```
42 \define@choicekey{tikzsymbols}{tree}%
43 [\tikzsymbols@tree@val\tikzsymbols@tree@nr]{true,on,false,off}[on]{%
44 \ifcase\tikzsymbols@tree@nr\relax
45 \def\Basic@Tree{\Basic@Tree@on}% tree=true
46 \or
47 \def\Basic@Tree{\Basic@Tree@on}% tree=on
48 \or
49 \def\Basic@Tree{\Basic@Tree@off}% tree=false
50 \or
51 \def\Basic@Tree{\Basic@Tree@off}% tree=false
52 \fi
53 }
```

`\ProcessOptionsX*` We process all options. What is `\relax` doing?

```
54 \ProcessOptionsX*<tikzsymbols>\relax
```

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

```
55 \newcommand*{\tikzsymbolsaftersymbolinput}{\xspace}
```

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

```
56 \newcommand*{\@leaf@is@leaf}{leaf}
```

`\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.

```
57 \newlength{\tkzsymls@scl}
58 \newcommand*{\set@tkzsymls@scl}[1]{\setlength{\tkzsymls@scl}{#1pt}}
```

`\tikzsymbols@draftboxlength`  
`\tikzsymbols@draftboxheight` The length and the height of the plain vanilla rectangle are stored inside these lengths.

```
59 \newlength{\tikzsymbols@draftboxlength}
60 \newlength{\tikzsymbols@draftboxheight}
```

`\tikzsymbols@draftbox` Our plain vanilla rectangle. Using `\tikzsymbols@draftboxlength` and `\tikzsymbols@draftboxheight` to calculate the length and the height of the box.

The box is drawn using `\frame` and a `\vbox`, `\hbox` construction. `\vbadness=\maxdimen` is needed because otherwise there would be many overfull v-box errors.

```
61 \newcommand*{\tikzsymbols@draftbox}[2]{%
62 \setlength{\tikzsymbols@draftboxlength}{#1}%
63 \setlength{\tikzsymbols@draftboxheight}{#2}%
64 \frame{%
65 {\vbadness=\maxdimen%
```

```

66   \frame{\vbox to \tikzsymbols@draftboxheight{%
67     \hbox to \tikzsymbols@draftboxlength{}}}%
68   }%
69 }

```

`\tikzsymbols@draftQbox` Short form of `\tikzsymbols@draftbox`. “Q” means “Quadrat” (*square*) and just means that height and length of the box are the same.

```
70 \newcommand*\tikzsymbols@draftQbox[1]{\tikzsymbols@draftbox{#1}{#1}}
```

`\tkzsymls@Prmtr` `\tikzsymbols@draftbox` doesn’t like negative numbers. So to be sure that only 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` (scaling) in `\tikzsymbols@draftbox`

```

71 \newcommand*\tkzsymls@Prmtr{}
72 \newcommand*\tkzsymls@bx@Prmtrstore[1]{%
73   \edef\tkzsymls@Prmtr{\ifdim\tkzsymls@scl<0pt-\fi#1}%
74 }

```

`\current@tikzsymbols` The `\sbox` and `\usebox` system I am using now has one drawback: If the symbol was used inside e.g. `\color{red}` all other symbols of the same type are red, even if they are in normal text, or even inside another color.

The same with e.g. `\small`. If a symbol was used the first time inside e.g. `\small` all other symbols of the same type are as small as the first one.

To overcome this problem, the save-boxes name depends of the current color and the current script size. These things are stored inside `\current@tikzsymbols`.

```
75 \newcommand*\current@tikzsymbols{}
```

`\tikzsymbols@ifsaveboxundefined` For every symbol we define a box (using `\sbox`). If a symbol of the same type is used again, it doesn’t have to be recalculated. We simply use the same output as the first symbol. If the symbol has another input, a new save-box is defined.

For example: `\Summertree[1]\Summertree[1]\Summertree[1]` is only calculated once because they are alike: 🌳🌳🌳. `\Summertree[1.3]` will be saved in an own save-box because it is different then the others.

```

76 \newcommand*\tikzsymbols@ifsaveboxundefined[2]{%
77   \edef\current@tikzsymbols{\current@color\fi@size}%
78   \expandafter\ifcsname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname%
79   \relax%
80   \else%
81   \expandafter\expandafter\expandafter\newsavebox%
82   \expandafter\expandafter\expandafter{%
83     \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}%
84   \expandafter\expandafter\expandafter\global%
85   \expandafter\expandafter\expandafter\sbox%
86   \expandafter\expandafter\expandafter{%
87     \csname\current@tikzsymbols tikzsymbols@savebox@#1\endcsname}{#2}%
88   \fi%
89 }

```

```

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

\ifKV@tikzsymbols@usebox = true If option usebox is false, then \tikzsymbols@ifsaveboxundefined and \tikzsymbols@use@box
will be redefined so that they don't store the input inside a save-box.
95 \ifKV@tikzsymbols@usebox
96 \relax% if true
97 \else
98 \renewcommand{\tikzsymbols@ifsaveboxundefined}[2]{\@secondoftwo{#1}{#2}}%
99 \renewcommand{\tikzsymbols@use@box}[1]{\@gobble{#1}}%
100 \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.
101 \newcommand{\tkzsymls@Declare@Robust@Command}[1]{%
102 \expandafter\def\csname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname{%
103 \expandafter\DeclareRobustCommand%
104 \csname\cmdKV@tikzsymbols@prefix#1\endcsname%
105 }

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

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

\tikzsymbolsuse To be able to don't have to care about the prefix, I made this command. Simply
write the name of the symbols inside without backslash. If the symbol is not
defined, there will be an error message.
116 \newcommand{\tikzsymbolsuse}[1]{%
117 \ifcsname\cmdKV@tikzsymbols@prefix#1tikzsymbolsuse\endcsname%

```

```

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

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

\@Tree@SetUp First we define our \@Tree@SetUp (how the trees will look like) (I used the code
from the tikz manual and changed it a little bit):
129 \def\@Tree@SetUp{\tikzset{%
130   ld/.style={level distance=##1ex},lw/.style={line width=##1ex},%
131   level 1/.style={ld=0.60, trunk, lw=0.1 ,sibling angle=60},%
132   level 2/.style={ld=0.20, trunk!80!leaf a, lw=.073,sibling angle=70},%
133   level 3/.style={ld=0.25, trunk!60!leaf a, lw=.05,sibling angle=70}, %
134   level 4/.style={ld=0.10, trunk!40!leaf a, lw=.025,sibling angle=60},%
135   level 5/.style={ld=0.15, trunk!20!leaf a, lw=.02,sibling angle=60},%
136   level 6/.style={ld=0.08, leaf a, lw=.021,sibling angle=60},%
137 }}%

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

\Basic@Tree@off \Basic@Tree@off is used when trees are turned off.
138 \DeclareRobustCommand{\Basic@Tree@off}[5][1=1, usedefault]{%
139   \set@tkzsymlsscl{#1}%
140   \pgfmathsetmacro\tikzsymbols@Tree@absolute@scale{#1+0.02ex}%
141   \edef\tkzsymls@Prmtr{\ifdim\tkzsymls@scl<0pt-\fi\tikzsymbols@Tree@absolute@scale}%
142   \ifdim\tkzsymls@scl<0pt \set@tkzsymlsscl{-#1}\fi%
143   \def\leaf@or@not@leaf{#5}%
144   \iftikzymbols@draftabsolute%
145     \ifx\leaf@or@not@leaf\@leaf@is@leaf%
146       \tikzsymbols@draftbox{(1.6772ex+0.4pt)*\real{\tkzsymls@Prmtr}}
147       {(1.42ex-0.2pt+0.4pt)*\real{\tkzsymls@Prmtr}}%
148     \else
149       \tikzsymbols@draftbox{(1.3996ex+0.4pt)*\real{\tkzsymls@Prmtr}}
150       {(1.28ex-0.2pt+0.4pt)*\real{\tkzsymls@Prmtr}}%
151     \fi
152   \else%
153     \begin{tikzpicture}[scale=#1+0.02ex,x=1ex,y=1ex, line width=0.4pt*\tkzsymls@scl]
154       \ifx\leaf@or@not@leaf\@leaf@is@leaf%
155         \draw[#2] (-0.8386,0+0.2pt) -- (-0.8386,1.42);

```

```

156         \draw[#3] (-0.8386,1.42) -- (0.8386,1.42);
157         \draw[#4] (0.8386,1.42) -- (0.8386,0+0.2pt);
158         \draw[#3] (0.8386,0+0.2pt) -- (0,0+0.2pt);
159         \draw[#4] (0,0+0.2pt) -- (-0.8386,0+0.2pt);
160     \else
161         \draw[#2] (-0.6998,0+0.2pt) -- (-0.6998,0.68+0.6);
162         \draw[#3] (-0.6998,0.68+0.6) -- (0.6998,0.68+0.6);
163         \draw[#4] (0.6998,0.68+0.6) -- (0.6998,0+0.2pt);
164     \fi%
165 \end{tikzpicture}%
166 \fi%
167 }}

\Basic@Tree@on \Basic@Tree@on is used when trees are turned on.
168 \DeclareRobustCommand{\Basic@Tree@on}[5][1=1, usedefault]{%
169 \iftikzsymbols@draftabsolute%
170 \Basic@Tree@off[#1]{#2}{#3}{#4}{#5}%
171 \else%
172 \set@tkzsymlbsscl{#1}%
173 \ifdim\tkzsymlb@scl<0pt \set@tkzsymlbsscl{-#1}\@tkzssmb@s@negtrue\fi%
174 \def\leaf@or@not@leaf{#5}%
175 \@Tree@SetUp%
176 \colorlet{trunk}{#2}%
177 \colorlet{leaf a}{#3}%
178 \colorlet{leaf b}{#4}%
179 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
180 \pgfarrowdeclare{leaf}{leaf}%
181 {\pgfarrowlefttextend{-0.1ex}\pgfarrowrighttextend{-0.05ex}}%
182 %
183 \pgfpathmoveto{\pgfpoint{-0.01ex}{0ex}}%
184 \pgfpatharc{150}{30}{0.08ex}%
185 \pgfpatharc{-30}{-150}{0.08ex}%
186 \pgfusepathqfill%
187 %
188 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
189 \draw[opacity=0,scale=#1+0.02ex, line width=0.4pt*\tkzsymlb@scl]
190 (-0.8386,0+0.2pt) rectangle
191 (0.8386, 1.42);
192 \else %
193 \draw[opacity=0,scale=#1+0.02ex, line width=0.4pt*\tkzsymlb@scl]
194 (-0.6998,0+0.2pt) rectangle (0.6998,0.68+0.6);
195 \fi
196 \pgflevel{\pgftransformscale{#1+0.02ex}}{%
197 \coordinate (root) [grow cyclic,rotate=90] child {
198 child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
199 child foreach \c in {0,1,2} { child foreach \d in {0,1} {
200 child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafco
201 }}}} edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
202 };}%
203 \end{tikzpicture}%

```



```

204 \@tkzssmbles@negfalse%
205 \fi%
206 }}

```

## 6.1 Cookingsymbolcode

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

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

```

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

```

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

```

\Schneebesen = \eggbeater The next one:

```

268 \tkzsymbols@Declare@Robust@Command{Schneebesen}[1][1]{%
269 \tikzsymbols@ifsaveboxundefined{Schneebesen#1}{%
270 \set@tkzsymbolsscl{#1}%
271 \iftikzymbols@draftabsolute%
272 \tkzsymbols@bx@Prmtrstore{#1}%
273 \tikzsymbols@draftbox{0.5697ex*\real{\tkzsymbols@Prmtr}}{1.57985ex*\real{\tkzsymbols@Prmtr}}%
274 \else%
275 \ifdim\tkzsymbols@scl<Opt\set@tkzsymbolsscl{-#1}\fi%
276 \begin{tikzpicture}[y=2.1ex,x=1.4ex, scale=#1,
277 line width=0.01ex*\tkzsymbols@scl*\real{0.97}]
278 \draw (0,0) .. controls (0.2,0.0) and (0.2,0.2) .. (0,0.4);
279 \draw (0,0) .. controls (-0.2,0.0) and (-0.2,0.2) .. (0,0.4);
280 \draw (0,0) .. controls (-0.1,0.0) and (-0.1,0.2) .. (0,0.4);
281 \draw (0,0) .. controls (0.1,0.0) and (0.1,0.2) .. (0,0.4);
282 \draw (0,0) .. controls (-0.15,0.0) and (-0.15,0.2) .. (0,0.4);
283 \draw (0,0) .. controls (0.15,0.0) and (0.15,0.2) .. (0,0.4);
284 \draw (0,0) .. controls (-0.05,0.0) and (-0.05,0.2) .. (0,0.4);
285 \draw (0,0) .. controls (0.05,0.0) and (0.05,0.2) .. (0,0.4);

```

```

286 \draw (0,0) --(0,0.4);
287 \fill[line width=0.05ex*\tkzsymls@scl, rounded corners=0.07ex*\tkzsymls@scl]
288   (-0.05,0.37) -- (0.05,0.37) -- (0.05,0.75) -- (-0.05,0.75) -- cycle;
289 \end{tikzpicture}%
290 \fi%
291 }%
292 \tikzsymbols@use@box{Schneebesen#1}%
293 \tikzsymbolsaftersymbolinput%
294 }
295 \tikzsymbols@let{eggbeater}{Schneebesen}

\Sieb = \sieve Now a long one;
296 \tkzsymls@Declare@Robust@Command{Sieb}[1][1]{%
297 \tikzsymbols@ifsaveboxundefined{Sieb#1}{%
298 \set@tkzsymls@scl{#1}%
299 \iftikzsymbols@draftabsolute%
300 \tkzsymls@bx@Prmtrstore{#1}%
301 \tikzsymbols@draftbox{3.478ex*\real{\tkzsymls@Prmtr}}{1.175ex*\real{\tkzsymls@Prmtr}}}%
302 \else%
303 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
304 \begin{tikzpicture}[x=2.8ex, y=2.8ex, line width=0.02ex*\tkzsymls@scl, scale=#1]
305 \draw[line width=0.09ex*\tkzsymls@scl] (-0.2,0) -- (1.01,0);
306 \draw (0.2,0) arc (180:360:0.4);
307 \draw(0.25,0) arc (180:360:0.35);
308 \draw (0.3,0) arc (180:360:0.3);
309 \draw (0.35,0) arc (180:360:0.25);
310 \draw (0.4,0) arc (180:360:0.2);
311 \draw (0.45,0) arc (180:360:0.15);
312 \draw (0.5,0) arc (180:360:0.1);
313 \draw (0.55,0) arc (180:360:0.05);
314 \draw (.95,0) -- (0.95,-0.194);
315 \draw (.9,0) -- (0.9,-0.265);
316 \draw (.85,0) -- (0.85,-0.313);
317 \draw (.8,0) -- (0.8,-0.345);
318 \draw (.75,0) -- (0.75,-0.37);
319 \draw (.7,0) -- (0.7,-0.39);
320 \draw (.65,0) -- (0.65,-0.4);
321 \draw (.6,0) -- (0.6,-0.4);
322 \draw (.55,0) -- (0.55,-0.4);
323 \draw (.5,0) -- (0.5,-0.39);
324 \draw (.45,0) -- (0.45,-0.37);
325 \draw (.4,0) -- (0.4,-0.348);
326 \draw (.35,0) -- (0.35,-0.314);
327 \draw (.3,0) -- (0.3,-0.265);
328 \draw (.25,0) -- (0.25,-0.194);
329 \draw (0.2,-0.05) -- (1,-0.05);
330 \draw (0.21,-0.1) -- (0.99,-0.1);
331 \draw (0.23,-0.15) -- (0.97,-0.15);
332 \draw (0.255,-0.2) -- (0.945,-0.2);
333 \draw (0.289,-0.25) -- (0.911,-0.25);

```

```

334 \draw (0.335,-0.3) -- (0.865,-0.3);
335 \draw (0.406,-0.35) -- (0.794,-0.35);
336 \end{tikzpicture}%
337 \fi%
338 }%
339 \tikzsymbols@use@box{Sieb#1}%
340 \tikzsymbolsaftersymbolinput%
341 }
342 \tikzsymbols@let{sieve}{Sieb}

```

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

```

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

```

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

```

363 \tikzsymbols@Declare@Robust@Command{Dreizack}[1][1]{%
364 \tikzsymbols@ifsaveboxundefined{Dreizack#1}{%
365 \set@tkzsymbolsscl{#1}%
366 \iftikzymbols@draftabsolute%
367 \tkzsymbols@bx@Prmtrstore{#1}%
368 \tikzsymbols@draftbox{0.265ex*\real{\tkzsymbols@Prmtr}}{1.575ex*\real{\tkzsymbols@Prmtr}}%
369 \else%
370 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
371 \begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.035ex*\tkzsymbols@scl,scale=#1]
372 \fill[rounded corners=0.07ex*(\tkzsymbols@scl-\tkzsymbols@scl/100)]
373 (0,0) -- (0,0.4) -- (0.1,0.4) -- (0.1,0.0) -- cycle;
374 \draw (0.05,0) -- (0.05,0.7);
375 \draw[rounded corners=0.07ex*(\tkzsymbols@scl-\tkzsymbols@scl/100*\tkzsymbols@scl*2]
376 (0,0.7) -- (0,0.55) -- (0.05,0.55) -- (0.1,0.55) -- (0.1,0.7);

```

```

377 \end{tikzpicture}%
378 \fi%
379 }%
380 \tikzsymbols@use@box{Dreizack#1}%
381 \tikzsymbolsaftersymbolinput%
382 }
383 \tikzsymbols@let{trident}{Dreizack}

```

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

```

384 \tikzsymbols@Declare@Robust@Command{Backblech}[1][1]{%
385 \tikzsymbols@ifsaveboxundefined{Backblech#1}{%
386 \set@tkzsymbolsscl{#1}%
387 \iftikzymbols@draftabsolute%
388 \tkzsymbols@bx@Prmtrstore{#1}%
389 \tikzsymbols@draftbox{2.315ex*\real{\tkzsymbols@Prmtr}}{1.57ex*\real{\tkzsymbols@Prmtr}}%
390 \else%
391 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
392 \begin{tikzpicture}[x=6.53ex,y=5ex, line width=0.07ex*\tkzsymbols@scl,scale=#1]
393 \filldraw[rounded corners=0.09ex*\tkzsymbols@scl] (0,0) rectangle (0.3,0.3);
394 \draw[rounded corners=0.07ex*\tkzsymbols@scl, line width=0.03ex*\tkzsymbols@scl]
395 (0.1,0) -- (-0.025,0) -- (-0.025,0.3) -- (0.1,0.3);
396 \draw[rounded corners=0.07ex*\tkzsymbols@scl, line width=0.03ex*\tkzsymbols@scl]
397 (0.2,0) -- (.325,0) -- (.325,0.3) -- (0.2,0.3);
398 \foreach \@BackblechlochX in {0.007,0.293}
399 \foreach \@BackblechlochY in {0.007,0.293}
400 \fill[white] (\@BackblechlochX,
401 \@BackblechlochY) circle (0.02ex);
402 \end{tikzpicture}%
403 \fi%
404 }%
405 \tikzsymbols@use@box{Backblech#1}%
406 \tikzsymbolsaftersymbolinput%
407 }
408 \tikzsymbols@let{bakingplate}{Backblech}

```

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

```

409 \tikzsymbols@Declare@Robust@Command{Ofen}[1][1]{%
410 \tikzsymbols@ifsaveboxundefined{Ofen#1}{%
411 \set@tkzsymbolsscl{#1}%
412 \iftikzymbols@draftabsolute%
413 \tkzsymbols@bx@Prmtrstore{#1}%
414 \tikzsymbols@draftbox{2.07ex*\real{\tkzsymbols@Prmtr}}{1.57ex*\real{\tkzsymbols@Prmtr}}%
415 \else%
416 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
417 \begin{tikzpicture}[x=0.50ex,y=0.5ex, line width=0.07ex*\tkzsymbols@scl,scale=#1]
418 \draw (0,0) rectangle (4,3);
419 \draw (0.25,0.25) rectangle (3.75,2);
420 \foreach \@Ofenschalter in {0.5,1.1,2.9,3.5}
421 \fill (\@Ofenschalter,2.5) circle (0.22);
422 \draw (1.5,2.28) rectangle (2.5,2.72);

```

```

423 \draw[line width=0.05ex*\tkzsymls@scl] (1,1.75) -- (3,1.75);
424 \end{tikzpicture}%
425 \fi%
426 }%
427 \tikzsymbols@use@box{Ofen#1}%
428 \tikzsymbolsaftersymbolinput%
429 }
430 \tikzsymbols@let{oven}{Ofen}

```

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

```

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

```

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

```

454 \tikzsymbols@Declare@Robust@Command{Herd}[1][1]{%
455 \tikzsymbols@ifsaveboxundefined{Herd#1}{%
456 \set@tkzsymls@scl{#1}%
457 \iftikzymbols@draftabsolute%
458 \tkzsymls@bx@Prmtrstore{#1}%
459 \tikzsymbols@draftbox{2.08ex*\real{\tkzsymls@Prmtr}}{1.58ex*\real{\tkzsymls@Prmtr}}}%
460 \else%
461 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
462 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.04ex*\tkzsymls@scl,scale=#1]
463 \draw[line width=0.08ex*\tkzsymls@scl] (0,0) rectangle (2,1.5);
464 \draw (0.5,0.45) circle (0.35);
465 \draw (0.5,0.45) circle (0.2);
466 \draw (1.45,0.45) circle (0.3);
467 \draw (0.5,1.15) circle (0.21);
468 \draw (1.05,0.95) rectangle (1.85,1.35);

```

```

469 \draw (1.45,1.15) circle (0.15);
470 \end{tikzpicture}%
471 \fi%
472 }%
473 \tikzsymbols@use@box{Herd#1}%
474 \tikzsymbolsaftersymbolinput%
475 }
476 \tikzsymbols@let{cooker}{Herd}

```

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

```

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

```

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

```

503 \tikzsymbols@Declare@Robust@Command{Schussel}[1][1]{%
504 \tikzsymbols@ifsaveboxundefined{Schussel#1}{%
505 \set@tkzsymbolsscl{#1}%
506 \iftikzymbols@draftabsolute%
507 \tkzsymbols@bx@Prmtrstore{#1}%
508 \tikzsymbols@draftbox{2.32ex*\real{\tkzsymbols@Prmtr}}{1.47ex*\real{\tkzsymbols@Prmtr}}%
509 \else%
510 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
511 \begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex*\tkzsymbols@scl, scale=#1]
512 \draw[rounded corners=0.5ex*\tkzsymbols@scl]
513 (-0.02,1.4) -- (0,1.4) -- (0,0.05) -- (1.5,0.05) -- (1.5,1.4) -- (1.52,1.4);
514 \draw (0.35,0) -- (1.15,0);

```

```

515 \draw[opacity=00] (-0.4,0) -- (1.85,0);
516 \end{tikzpicture}%
517 \fi%
518 }%
519 \tikzsymbols@use@box{Schussel#1}%
520 \tikzsymbolsaftersymbolinput%
521 }
522 \tikzsymbols@let{bowl}{Schussel}

```

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

```

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

```

Text hinschreiben = \grater Text hinschreiben

```

545 \tkzsymbols@Declare@Robust@Command{Reibe}[1][1]{%
546 \tikzsymbols@ifsaveboxundefined{Reibe#1}{%
547 \set@tkzsymbolsscl{#1}%
548 \iftikzymbols@draftabsolute%
549 \tkzsymbols@bx@Prmtrstore{#1}%
550 \tikzsymbols@draftbox{1.08ex*\real{\tkzsymbols@Prmtr}}{1.58ex*\real{\tkzsymbols@Prmtr}}%
551 \else%
552 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
553 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.08ex*\tkzsymbols@scl, scale=#1]%
554 \draw (0,0) rectangle (1,1.2);
555 \draw[rounded corners=0.04ex] (0.05,1.2) rectangle (0.95,1.5);
556 \foreach\x in {0.2,0.4,0.6,0.8}
557 \foreach\y in {0.2,0.4, 0.6, 0.8, 1}
558 \fill (\x,\y) circle (0.05ex);
559 \end{tikzpicture}%

```



```

560 \fi%
561 }%
562 \tikzsymbols@use@box{Reibe#1}%
563 \tikzsymbolsaftersymbolinput%
564 }
565 \tikzsymbols@let{grater}{Reibe}

```



`\Flasche = \bottle` Text hinschreiben

```

566 \tikzsymbols@Declare@Robust@Command{Flasche}[1][1]{%
567 \tikzsymbols@ifsaveboxundefined{Flasche#1}{%
568 \set@tkzsymbolsscl{#1}%
569 \iftikzymbols@draftabsolute%
570 \tkzsymbols@bx@Prmtrstore{#1}%
571 \tikzsymbols@draftbox{0.78ex*\real{\tkzsymbols@Prmtr}}{1.58ex*\real{\tkzsymbols@Prmtr}}%
572 \else%
573 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
574 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.08ex, rounded corners=0.08ex*\tkzsymbols@scl, scal
575 \draw (0, 1.5) -- (0,1.2) -- (-0.15,0.8) -- (-0.15,0) ---+ (0.6,0) ---+ (0,0.8) ---+ (-0.15,0.4
576 %\draw (-0.15,0.8) -- (0.45,0.8);
577 %\draw (-0.15,0.3) -- (0.45,0.3);
578 \draw[opacity=00] (-0.2,0) ---+ (0.7,0);
579 \end{tikzpicture}\fi%
580 }%
581 \tikzsymbols@use@box{Flasche#1}%
582 \tikzsymbolsaftersymbolinput%
583 }
584 \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 `opacity=0`, it’s useful for `\colorbox{yellow}{\Sadey}` which leads to  instead of  (with default=white).

```

585 \tikzsymbols@Declare@Robust@Commandx{Sadey}[2][1=1, 2={opacity=0}, usedefault]{%
586 \tikzsymbols@ifsaveboxundefined{Sadey#1#2}{%
587 \set@tkzsymbolsscl{#1}%
588 \iftikzymbols@draftabsolute%
589 \tkzsymbols@bx@Prmtrstore{#1}%
590 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
591 \else%
592 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
593 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
594 \fill[#2, line width=0.1ex*\tkzsymbols@scl] (0,0) circle (0.33);
595 \draw[line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
596 \fill (0.1,0.1) circle (0.05);
597 \fill (-0.1,0.1) circle (0.05);
598 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);

```

```

599 \end{tikzpicture}%
600 \fi%
601 }%
602 \tikzsymbols@use@box{Sadey#1#2}%
603 \tikzsymbolsaftersymbolinput%
604 }
605 \tkzsymls@Declare@Robust@Commandx{dSadey}[2][1=1,2=yellow,usedefault]{%
606 \tikzsymbols@ifsaveboxundefined{dSadey#1#2}{%
607 \set@tkzsymls@sc1{#1}%
608 \iftikzymbols@draftabsolute%
609 \tkzsymls@bx@Prmtrstore{#1}%
610 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymls@Prmtr}}%
611 \else%
612 \ifdim\tkzsymls@sc1<0pt\set@tkzsymls@sc1{-#1}\fi%
613 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@sc1,scale=#1]
614 \shade[ball color=#2] (0,0) circle (0.33);
615 \shade[ball color=black] (0.1,0.1) circle (0.05);
616 \shade[ball color=black] (-0.1,0.1) circle (0.05);
617 \draw[black] (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
618 \end{tikzpicture}%
619 \fi%
620 }%
621 \tikzsymbols@use@box{dSadey#1#2}%
622 \tikzsymbolsaftersymbolinput%
623 }

```

\Annoey \dAnnoey An annoyed Smiley --

```

624 \tkzsymls@Declare@Robust@Commandx{Annoey}[2][1=1,2={opacity=0},usedefault]{%
625 \tikzsymbols@ifsaveboxundefined{Annoey#1#2}{%
626 \set@tkzsymls@sc1{#1}%
627 \iftikzymbols@draftabsolute%
628 \tkzsymls@bx@Prmtrstore{#1}%
629 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymls@Prmtr}}%
630 \else%
631 \ifdim\tkzsymls@sc1<0pt\set@tkzsymls@sc1{-#1}\fi%
632 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@sc1,scale=#1]
633 \fill[#2, line width=0.12ex*\tkzsymls@sc1] (0,0) circle (0.33);
634 \draw[line width=0.12ex*\tkzsymls@sc1] (0,0) circle (0.33);
635 \draw (0.08,0.1) -- (0.22,0.1);
636 \draw (-0.08,0.1) -- (-0.22,0.1);
637 \draw (-0.2,-0.1) -- (0.2,-0.1);
638 \end{tikzpicture}%
639 \fi%
640 }%
641 \tikzsymbols@use@box{Annoey#1#2}%
642 \tikzsymbolsaftersymbolinput%
643 }
644 \tkzsymls@Declare@Robust@Commandx{dAnnoey}[2][1=1,2=yellow,usedefault]{%
645 \tikzsymbols@ifsaveboxundefined{dAnnoey#1#2}{%
646 \set@tkzsymls@sc1{#1}%

```

```

647 \iftikzsymbols@draftabsolute%
648 \tkzsymbols@bx@Prmtrstore{#1}%
649 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
650 \else%
651 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
652 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
653 \shade[ball color=#2] (0,0) circle (0.33);
654 \draw[black] (0.08,0.1) -- (0.22,0.1);
655 \draw[black] (-0.08,0.1) -- (-0.22,0.1);
656 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
657 \end{tikzpicture}%
658 \fi%
659 }%
660 \tikzsymbols@use@box@dAnnoey#1#2}%
661 \tikzsymbolsaftersymbolinput%
662 }

```

\Smiley \dSmiley A normal Smiley

```

663 \ifKV@tikzsymbols@marvosym\relax\else%
664 \tkzsymbols@Declare@Robust@Commandx{Smiley}[2][1=1,2={opacity=0} ,usedefault]{%
665 \tikzsymbols@ifsaveboxundefined{Smiley#1#2}{%
666 \set@tkzsymbolsscl{#1}%
667 \iftikzsymbols@draftabsolute%
668 \tkzsymbols@bx@Prmtrstore{#1}%
669 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
670 \else%
671 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
672 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl, scale=#1]
673 \fill[#2] (0,0) circle (0.33);
674 \draw (0,0) circle (0.33);
675 \fill (-0.1,0.1) circle (0.05);
676 \fill (0.1,0.1) circle (0.05);
677 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
678 \end{tikzpicture}%
679 \fi%
680 }%
681 \tikzsymbols@use@box{Smiley#1#2}%
682 \tikzsymbolsaftersymbolinput%
683 }%
684 \fi
685 \tkzsymbols@Declare@Robust@Commandx{dSmiley}[2][1=1,2=yellow,usedefault]{%
686 \tikzsymbols@ifsaveboxundefined{dSmiley#1#2}{%
687 \set@tkzsymbolsscl{#1}%
688 \iftikzsymbols@draftabsolute%
689 \tkzsymbols@bx@Prmtrstore{#1}%
690 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
691 \else%
692 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
693 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.1ex*\tkzsymbols@scl,scale=#1]
694 \shade[ball color=#2] (0,0) circle (0.33);

```

```

695 \shade[ball color=black] (-0.1,0.1) circle (0.05);
696 \shade[ball color=black] (0.1,0.1) circle (0.05);
697 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
698 \end{tikzpicture}%
699 \fi%
700 }%
701 \tikzsymbols@use@box@dSmiley#1#2}%
702 \tikzsymbolsaftersymbolinput%
703 }

```

\Laughey \dLaughey A laughing Smiley

```

704 \tikzsymbols@Declare@Robust@Commandx{Laughey}[3][1=1,2={opacity=0},3={opacity=0} ,usedefault]{%
705 \tikzsymbols@ifsaveboxundefined{Laughey#1#2#3}{%
706 \set@tkzsymbolsscl{#1}%
707 \iftikzymbols@draftabsolute%
708 \tkzsymbols@bx@Prmtrstore{#1}%
709 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}}%
710 \else%
711 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
712 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
713 \fill[#2,line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
714 \draw[line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
715 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
716 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
717 \fill[#3,rounded corners=0.1ex*\tkzsymbols@scl, yshift=-0.5]
718 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
719 \draw[rounded corners=0.1ex*\tkzsymbols@scl, yshift=-0.5]
720 (-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
721 \end{tikzpicture}%
722 \fi%
723 }%
724 \tikzsymbols@use@box{Laughey#1#2#3}%
725 \tikzsymbolsaftersymbolinput%
726 }
727 \tikzsymbols@Declare@Robust@Commandx{dLaughey}[3][1=1,2=yellow, 3=red ,usedefault]{%
728 \tikzsymbols@ifsaveboxundefined{dLaughey#1#2#3}{%
729 \set@tkzsymbolsscl{#1}%
730 \iftikzymbols@draftabsolute%
731 \tkzsymbols@bx@Prmtrstore{#1}%
732 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}}%
733 \else%
734 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
735 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
736 \fill[ball color=#2,line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
737 \draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
738 \draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
739 \shade[ball color=#3, rounded corners=0.1ex*\tkzsymbols@scl, yshift=-0.3]
740 (-0.25,-0.0) .. controls (-0.13,-0.26) and (0.13,-0.26) .. (0.25,-0.0) -- cycle;
741 \end{tikzpicture}%
742 \fi%

```

```

743 }%
744 \tikzsymbols@use@box{dLaughey#1#2#3}%
745 \tikzsymbolsaftersymbolinput%
746 }

```

\Neutrey \dNeutrey neutral Smiley :|

```

747 \tikzsymbols@Declare@Robust@Commandx{Neutrey}[2][1=1, 2={opacity=0}, usedefault]{%
748 \tikzsymbols@ifsaveboxundefined{Neutrey#1#2}{%
749 \set@tkzsymbolsscl{#1}%
750 \iftikzymbols@draftabsolute%
751 \tkzsymbols@bx@Prmtrstore{#1}%
752 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
753 \else%
754 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
755 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
756 \fill[#2,line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
757 \draw[line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
758 \fill (0.1,0.1) circle (0.05);
759 \fill (-0.1,0.1) circle (0.05);
760 \draw (-0.2,-0.1) -- (0.2,-0.1);
761 \end{tikzpicture}%
762 \fi%
763 }%
764 \tikzsymbols@use@box{Neutrey#1#2}%
765 \tikzsymbolsaftersymbolinput%
766 }
767 \tikzsymbols@Declare@Robust@Commandx{dNeutrey}[2][1=1,2=yellow,usedefault]{%
768 \tikzsymbols@ifsaveboxundefined{dNeutrey#1#2}{%
769 \set@tkzsymbolsscl{#1}%
770 \iftikzymbols@draftabsolute%
771 \tkzsymbols@bx@Prmtrstore{#1}%
772 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
773 \else%
774 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
775 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
776 \shade[ball color=#2] (0,0) circle (0.33);
777 \shade[ball color=black] (0.1,0.1) circle (0.05);
778 \shade[ball color=black] (-0.1,0.1) circle (0.05);
779 \draw[black] (-0.2,-0.1) -- (0.2,-0.1);
780 \end{tikzpicture}%
781 \fi%
782 }%
783 \tikzsymbols@use@box{dNeutrey#1#2}%
784 \tikzsymbolsaftersymbolinput%
785 }

```

\Winkey \dWinkey ;)

```

\doldWinkey \olddWinkey 786 \tikzsymbols@Declare@Robust@Commandx{Winkey}[2][1=1,2={opacity=0} ,usedefault]{%
787 \tikzsymbols@ifsaveboxundefined{Winkey#1#2}{%
788 \set@tkzsymbolsscl{#1}%

```

```

789 \iftikzsymbols@draftabsolute%
790 \tkzsymbols@bx@Prmtrstore{#1}%
791 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
792 \else%
793 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
794 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
795 \fill[#2] (0,0) circle (0.33);
796 \draw(0,0) circle (0.33);
797 \draw(0.17,0.1) -- (0.05,0.1);
798 \fill (-0.1,0.1) circle (0.05);
799 \draw (-0.15,-0.15) .. controls (-0.05,-0.2) and (0.15,-0.2) .. (0.19,0);
800 \end{tikzpicture}%
801 \fi%
802 }%
803 \tikzsymbols@use@box{Winkey#1#2}%
804 \tikzsymbolsaftersymbolinput%
805 }
806 \tkzsymbols@Declare@Robust@Commandx{oldWinkey}[2][1=1,2={opacity=0} ,usedefault]{%
807 \tikzsymbols@ifsaveboxundefined{oldWinkey#1#2}{%
808 \set@tkzsymbolsscl{#1}%
809 \iftikzsymbols@draftabsolute%
810 \tkzsymbols@bx@Prmtrstore{#1}%
811 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
812 \else%
813 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
814 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
815 \fill[#2] (0,0) circle (0.33);
816 \draw(0,0) circle (0.33);
817 \draw(0.17,0.1) -- (0.05,0.1);
818 \fill (-0.1,0.1) circle (0.05);
819 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
820 \end{tikzpicture}%
821 \fi%
822 }%
823 \tikzsymbols@use@box{oldWinkey#1#2}%
824 \tikzsymbolsaftersymbolinput%
825 }
826 \tkzsymbols@Declare@Robust@Commandx{dWinkey}[2][1=1,2=yellow,usedefault]{%
827 \tikzsymbols@ifsaveboxundefined{dWinkey#1#2}{%
828 \set@tkzsymbolsscl{#1}%
829 \iftikzsymbols@draftabsolute%
830 \tkzsymbols@bx@Prmtrstore{#1}%
831 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
832 \else%
833 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
834 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
835 \shade[ball color=#2] (0,0) circle (0.33);
836 \draw[black] (0.17,0.1) -- (0.05,0.1);
837 \shade[ball color=black] (-0.1,0.1) circle (0.05);
838 \draw[black] (-0.15,-0.15) .. controls (-0.05,-0.2) and (0.15,-0.2) .. (0.19,0);

```

```

839 \end{tikzpicture}%
840 \fi%
841 }%
842 \tikzsymbols@use@box{dWinkey#1#2}%
843 \tikzsymbolsaftersymbolinput%
844 }
845 \tkzsymls@Declare@Robust@Commandx{olddWinkey}[2][1=1,2=yellow,usedefault]{%
846 \tikzsymbols@ifsaveboxundefined{olddWinkey#1#2}{%
847 \set@tkzsymls@sc1{#1}%
848 \iftikzymbols@draftabsolute%
849 \tkzsymls@bx@Prmtrstore{#1}%
850 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymls@Prmtr}}%
851 \else%
852 \ifdim\tkzsymls@sc1<0pt\set@tkzsymls@sc1{-#1}\fi%
853 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@sc1,scale=#1]
854 \shade[ball color=#2] (0,0) circle (0.33);
855 \draw(0.17,0.1) -- (0.05,0.1);
856 \shade[ball color=black] (-0.1,0.1) circle (0.05);
857 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
858 \end{tikzpicture}%
859 \fi%
860 }%
861 \tikzsymbols@use@box{olddWinkey#1#2}%
862 \tikzsymbolsaftersymbolinput%
863 }

```

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

```

864 \tkzsymls@Declare@Robust@Commandx{Sey}[2][1=1,2={opacity=0} ,usedefault]{%
865 \tikzsymbols@ifsaveboxundefined{Sey#1#2}{%
866 \set@tkzsymls@sc1{#1}%
867 \iftikzymbols@draftabsolute%
868 \tkzsymls@bx@Prmtrstore{#1}%
869 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymls@Prmtr}}%
870 \else%
871 \ifdim\tkzsymls@sc1<0pt\set@tkzsymls@sc1{-#1}\fi%
872 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@sc1,scale=#1]
873 \fill[#2, line width=0.12ex*\tkzsymls@sc1] (0,0) circle (0.33);
874 \draw[line width=0.12ex*\tkzsymls@sc1] (0,0) circle (0.33);
875 \fill (0.1,0.1) circle (0.05);
876 \fill (-0.1,0.1) circle (0.05);
877 \draw (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
878 \end{tikzpicture}%
879 \fi%
880 }%
881 \tikzsymbols@use@box{Sey#1#2}%
882 \tikzsymbolsaftersymbolinput%
883 }
884 \tkzsymls@Declare@Robust@Commandx{dSey}[2][1=1,2=yellow ,usedefault]{%
885 \tikzsymbols@ifsaveboxundefined{dSey#1#2}{%
886 \set@tkzsymls@sc1{#1}%

```

```

887 \iftikzsymbols@draftabsolute%
888 \tkzsymbols@bx@Prmtrstore{#1}%
889 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
890 \else%
891 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
892 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
893 \shade[ball color=#2] (0,0) circle (0.33);
894 \shade[ball color=black] (0.1,0.1) circle (0.05);
895 \shade[ball color=black] (-0.1,0.1) circle (0.05);
896 \draw[black] (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
897 \end{tikzpicture}%
898 \fi%
899 }%
900 \tikzsymbols@use@box{dSey#1#2}%
901 \tikzsymbolsaftersymbolinput%
902 }

```

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

```

903 \tkzsymbols@Declare@Robust@Commandx{Xey}[2][1=1, 2={opacity=0}, usedefault]{%
904 \tikzsymbols@ifsaveboxundefined{Xey#1#2}{%
905 \set@tkzsymbolsscl{#1}%
906 \iftikzsymbols@draftabsolute%
907 \tkzsymbols@bx@Prmtrstore{#1}%
908 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
909 \else%
910 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
911 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
912 \fill[#2, line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
913 \draw[line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);
914 \draw (0.05,0.05) -- ++ (0.1,0.1);
915 \draw (0.15,0.05) -- ++ (-0.1,0.1);
916 \draw (-0.05,0.05) -- ++ (-0.1,0.1);
917 \draw (-0.15,0.05) -- ++ (0.1,0.1);
918 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
919 \end{tikzpicture}%
920 \fi%
921 }%
922 \tikzsymbols@use@box{Xey#1#2}%
923 \tikzsymbolsaftersymbolinput%
924 }
925 \tkzsymbols@Declare@Robust@Commandx{dXey}[2][1=1, 2={yellow}, usedefault]{%
926 \tikzsymbols@ifsaveboxundefined{dXey#1#2}{%
927 \set@tkzsymbolsscl{#1}%
928 \iftikzsymbols@draftabsolute%
929 \tkzsymbols@bx@Prmtrstore{#1}%
930 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
931 \else%
932 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
933 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
934 \fill[ball color=#2, line width=0.12ex*\tkzsymbols@scl] (0,0) circle (0.33);

```



```

935 \draw (0.05,0.05) -- ++ (0.1,0.1);
936 \draw (0.15,0.05) -- ++ (-0.1,0.1);
937 \draw (-0.05,0.05) -- ++ (-0.1,0.1);
938 \draw (-0.15,0.05) -- ++ (0.1,0.1);
939 \draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
940 \end{tikzpicture}%
941 \fi%
942 }%
943 \tikzsymbols@use@box{dXkey#1#2}%
944 \tikzsymbolsaftersymbolinput%
945 }

```

\Innocey \dInnocey An innocent Smiley

```

946 \tikzsymbols@Declare@Robust@Commandx{Innocey}[3][1=1,2={opacity=0},3=yellow ,usedefault]{%
947 \tikzsymbols@ifsaveboxundefined{Innocey#1#2#3}{%
948 \set@tkzsymbolsscl{#1}%
949 \iftikzymbols@draftabsolute%
950 \tkzsymbols@bx@Prmtrstore{#1}%
951 \tikzsymbols@draftbox{1.73ex*\real{\tkzsymbols@Prmtr}}{1.909ex*\real{\tkzsymbols@Prmtr}}%
952 \else%
953 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
954 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
955 \fill[#2] (0,0) circle (0.33);
956 \draw (0,0) circle (0.33);
957 \fill (-0.1,0.1) circle (0.05);
958 \fill (0.1,0.1) circle (0.05);
959 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
960 \draw[#3, line width=0.095ex*\tkzsymbols@scl] (0.32,0.31) arc (0:360:0.32 and 0.1);
961 \draw[line width=0.05ex*\tkzsymbols@scl] (0.3,0.31) arc (0:360:0.3 and 0.07);
962 \draw[line width=0.05ex*\tkzsymbols@scl] (0.35,0.31) arc (0:360:0.35 and 0.12);
963 \end{tikzpicture}%
964 \fi%
965 }%
966 \tikzsymbols@use@box{Innocey#1#2#3}%
967 \tikzsymbolsaftersymbolinput%
968 }
969 \tikzsymbols@Declare@Robust@Command{wInnocey}[1][1]{\Innocey[#1][opacity=0][white]}
970 \tikzsymbols@Declare@Robust@Commandx{dInnocey}[3][1=1,2=yellow,3=yellow,usedefault]{%
971 \tikzsymbols@ifsaveboxundefined{dInnocey#1#2#3}{%
972 \set@tkzsymbolsscl{#1}%
973 \iftikzymbols@draftabsolute%
974 \tkzsymbols@bx@Prmtrstore{#1}%
975 \tikzsymbols@draftbox{1.73ex*\real{\tkzsymbols@Prmtr}}{1.849ex*\real{\tkzsymbols@Prmtr}}%
976 \else%
977 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
978 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
979 \shade[ball color=#2] (0,0) circle (0.33);
980 \shade[ball color=black] (-0.1,0.1) circle (0.05);
981 \shade[ball color=black] (0.1,0.1) circle (0.05);
982 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);

```

```

983 \draw[color=#3!97!black, line width=0.1ex*\tkzsymbols@scl]
984   (0.32,0.31) arc (0:360:0.32 and 0.1);
985 \draw[line width=0.05ex*\tkzsymbols@scl] (0.3,0.31) arc (0:360:0.3 and 0.07);
986 \draw[line width=0.05ex*\tkzsymbols@scl] (0.35,0.31) arc (0:360:0.35 and 0.12);
987 \end{tikzpicture}%
988 \fi%
989 }%
990 \tikzsymbols@use@box@dInnocey#1#2#3}%
991 \tikzsymbolsaftersymbolinput%
992 }

```

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

```

993 \tikzsymbols@Declare@Robust@Commandx{Cooley}[2][1=1,2={opacity=0} ,usedefault]{%
994 \tikzsymbols@ifsaveboxundefined{Cooley#1#2}{%
995 \set@tkzsymbolsscl{#1}%
996 \iftikzymbols@draftabsolute%
997 \tkzsymbols@bx@Prmtrstore{#1}%
998 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymbols@Prmtr}}%
999 \else%
1000 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1001 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
1002 \fill[#2] (0,0) circle (0.33);
1003 \draw (0,0) circle (0.33);
1004 \fill[rounded corners=0.1ex*\tkzsymbols@scl]
1005   (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
1006 \fill[rounded corners=0.1ex*\tkzsymbols@scl]
1007   (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
1008 \draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1009 \draw (0.02,0.1) -- (-0.02,0.1);
1010 \draw (-0.2,0.1) -- (-0.3,0.13);
1011 \draw (0.2,0.1) -- (0.3,0.13);
1012 \end{tikzpicture}%
1013 \fi%
1014 }%
1015 \tikzsymbols@use@box{Cooley#1#2}%
1016 \tikzsymbolsaftersymbolinput%
1017 }
1018 \tikzsymbols@Declare@Robust@Commandx{dCooley}[2][1=1,2=yellow,usedefault]{%
1019 \tikzsymbols@ifsaveboxundefined{dCooley#1#2}{%
1020 \set@tkzsymbolsscl{#1}%
1021 \iftikzymbols@draftabsolute%
1022 \tkzsymbols@bx@Prmtrstore{#1}%
1023 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymbols@Prmtr}}%
1024 \else%
1025 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1026 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymbols@scl,scale=#1]
1027 \shade[ball color=#2] (0,0) circle (0.33);
1028 \draw[black] (0.02,0.1) -- (-0.02,0.1);
1029 \draw[black] (-0.2,0.1) -- (-0.295,0.146);
1030 \draw[black] (0.2,0.1) -- (0.295,0.146);

```

```

1031 \shade[ball color=black,rounded corners=0.1ex*\tkzsymls@scl]
1032   (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
1033 \shade[ball color=black,rounded corners=0.1ex*\tkzsymls@scl]
1034   (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
1035 \draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1036 \end{tikzpicture}%
1037 \fi%
1038 }%
1039 \tikzsymbols@use@box@dCooley#1#2}%
1040 \tikzsymbolsaftersymbolinput%
1041 }

\Tongey \dTongey :P
1042 \tkzsymbols@Declare@Robust@Commandx{Tongey}[3][1=1,2={opacity=0},3={opacity=0} ,usedefault]{%
1043 \tikzsymbols@ifsaveboxundefined{Tongey#1#2#3}{%
1044 \set@tkzsymls@scl{#1}%
1045 \iftikzymbols@draftabsolute%
1046 \tkzsymls@bx@Prmtrstore{#1}%
1047 \tikzsymbols@draftQbox{1.704ex*\real{\tkzsymls@Prmtr}}}%
1048 \else%
1049 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1050 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
1051 \fill[#2] (0,0) circle (0.33);
1052 \draw (0,0) circle (0.33);
1053 \fill (-0.1,0.1) circle (0.05);
1054 \fill (0.1,0.1) circle (0.05);
1055 \fill[#3,line width=0.058ex*\tkzsymls@scl, rounded corners=0.12ex*\tkzsymls@scl]
1056   (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1057 \draw[line width=0.07ex*\tkzsymls@scl, yshift=0.21ex]
1058   (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1059 \draw[line width=0.058ex*\tkzsymls@scl, rounded corners=0.12ex*\tkzsymls@scl]
1060   (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1061 \end{tikzpicture}%
1062 \fi%
1063 }%
1064 \tikzsymbols@use@box{Tongey#1#2#3}%
1065 \tikzsymbolsaftersymbolinput%
1066 }

1067 \tkzsymbols@Declare@Robust@Commandx{dTongey}[3][1=1,2=yellow,3=red,usedefault]{%
1068 \tikzsymbols@ifsaveboxundefined{dTongey#1#2#3}{%
1069 \set@tkzsymls@scl{#1}%
1070 \iftikzymbols@draftabsolute%
1071 \tkzsymls@bx@Prmtrstore{#1}%
1072 \tikzsymbols@draftQbox{1.584ex*\real{\tkzsymls@Prmtr}}}%
1073 \else%
1074 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1075 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
1076 \shade[ball color=#2] (0,0) circle (0.33);
1077 \shade[ball color=black] (-0.1,0.1) circle (0.05);
1078 \shade[ball color=black] (0.1,0.1) circle (0.05);

```

```

1079 \shade[ball color=#3,line width=0.058ex*\tkzsymls@scl, rounded corners=0.12ex*\tkzsymls@scl]
1080 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1081 \draw[black, line width=0.058ex*\tkzsymls@scl, rounded corners=0.12ex*\tkzsymls@scl]
1082 (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
1083 \draw[black, line width=0.07ex*\tkzsymls@scl, yshift=0.21ex]
1084 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1085 \end{tikzpicture}%
1086 \fi%
1087 }%
1088 \tikzsymbols@use@box{dTongey#1#2#3}%
1089 \tikzsymbolsaftersymbolinput%
1090 }

```

\Nursey \dNursey a Nurse.

```

1091 \tkzsymbols@Declare@Robust@Commandx{Nursey}
1092 [4] [1=1,2={opacity=0},3={opacity=0},4=black,usedefault]{%
1093 \tikzsymbols@ifsaveboxundefined{Nursey#1#2#3#4}{%
1094 \set@tkzsymls@scl{#1}%
1095 \iftikzymbols@draftabsolute%
1096 \tkzsymls@bx@Prmtrstore{#1}%
1097 \tikzsymbols@draftbox{1.5ex*\real{\tkzsymls@Prmtr}}{2.19ex*\real{\tkzsymls@Prmtr}}%
1098 \else%
1099 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1100 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
1101 \fill[#3,rounded corners=.023ex*\tkzsymls@scl]
1102 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
1103 \fill[#2] (0,0) circle (0.3);
1104 \draw (0,0) circle (0.3);
1105 \fill (-0.1,0.1) circle (0.05);
1106 \fill (0.1,0.1) circle (0.05);
1107 \draw[line width=0.09ex*\tkzsymls@scl, yshift=0.07ex]
1108 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1109 \draw[rounded corners=.023ex*\tkzsymls@scl]
1110 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
1111 \draw[#4,line width=.046ex*\tkzsymls@scl] (0,0.35) -- (0,0.5);
1112 \draw[#4,line width=.046ex*\tkzsymls@scl] (-0.05,0.45) -- (0.05,0.45);
1113 \end{tikzpicture}%
1114 \fi%
1115 }%
1116 \tikzsymbols@use@box{Nursey#1#2#3#4}%
1117 \tikzsymbolsaftersymbolinput%
1118 }
1119 \tkzsymbols@Declare@Robust@Commandx{dNursey}[4] [1=1,2=yellow,3=white,4=red,usedefault]{%
1120 \tikzsymbols@ifsaveboxundefined{dNursey#1#2#3#4}{%
1121 \set@tkzsymls@scl{#1}%
1122 \iftikzymbols@draftabsolute%
1123 \tkzsymls@bx@Prmtrstore{#1}%
1124 \tikzsymbols@draftbox{1.38ex*\real{\tkzsymls@Prmtr}}{1.98ex*\real{\tkzsymls@Prmtr}}%
1125 \else%
1126 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%

```

```

1127 \begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymls@scl,scale=#1]
1128 \shade[ball color=#2] (0,0) circle (0.3);
1129 \shade[ball color=black] (-0.1,0.1) circle (0.05);
1130 \shade[ball color=black] (0.1,0.1) circle (0.05);
1131 \draw[black, line width=0.09ex*\tkzsymls@scl, yshift=0.07ex]
1132 (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
1133 \shade[ball color=#3, rounded corners=.023ex*\tkzsymls@scl,yshift=-0.09ex]
1134 (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0) arc (0:180:0.3);
1135 \shade[ball color=#4,line width=.046ex*\tkzsymls@scl]
1136 (-0.01,0.31) -- (-0.01,0.46) -- (0.01,0.46) -- (0.01,0.31)--cycle;
1137 \shade[ball color=#4,line width=.046ex*\tkzsymls@scl]
1138 (-0.05,0.4) -- (0.05,0.4) -- (0.05,0.42)--(-0.05,0.42) -- cycle;
1139 \end{tikzpicture}%
1140 \fi%
1141 }%
1142 \tikzsymbols@use@box{dNursey#1#2#3#4}%
1143 \tikzsymbolsaftersymbolinput%
1144 }

\Vomey \dVomey *Bläärgh*

1145 \tikzsymbols@Declare@Robust@Commandx{Vomey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
1146 \tikzsymbols@ifsaveboxundefined{Vomey#1#2#3}{%
1147 \set@tkzsymls@scl{#1}%
1148 \iftikzymbols@draftabsolute%
1149 \tkzsymls@bx@Prmtrstore{#1}%
1150 \tikzsymbols@draftbox{3.0335ex*\real{\tkzsymls@Prmtr}}{1.743ex*\real{\tkzsymls@Prmtr}}%
1151 \else%
1152 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1153 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymls@scl,scale=#1]
1154 \fill[#2,rounded corners=0.05ex*\tkzsymls@scl] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1155 \draw[rounded corners=0.05ex*\tkzsymls@scl] (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1156 \draw[line width=0.05ex*\tkzsymls@scl] (-0.5,0.3) -- (-0.3,0.1);
1157 \fill (-0.45,0.27) arc (100:350:0.1);
1158 \fill[#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1159 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
1160 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1161 \fill[#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) ..
1162 controls (1.7,-1.2) and (1.3,-1.2) .. (1,-1) ..
1163 controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
1164 \draw (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1);
1165 \draw (0,-0.4) .. controls (0.5,-0.5) and (0.8,-0.7) .. (1,-1);
1166 \draw (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1167 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2)
1168 and (1,-2) .. (1,-1.5) .. controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1169 \end{tikzpicture}%
1170 \fi%
1171 }%
1172 \tikzsymbols@use@box{Vomey#1#2#3}%
1173 \tikzsymbolsaftersymbolinput%
1174 }

```

```

1175 \tkzsymbols@Declare@Robust@Commandx{dVomey}[3][1=1,2=yellow,3={brown!10!olive},usedefault]{%
1176 \tikzsymbols@ifsaveboxundefined{dVomey#1#2#3}{%
1177 \set@tkzsymbolsscl{#1}%
1178 \iftikzymbols@draftabsolute%
1179 \tkzsymbols@bx@Prmtrstore{#1}%
1180 \tikzsymbols@draftbox{3.2435ex*\real{\tkzsymbols@Prmtr}}{1.653ex*\real{\tkzsymbols@Prmtr}}%
1181 \else%
1182 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1183 \begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
1184 \shade[ball color=#2!90!brown,rounded corners=0.03ex*\tkzsymbols@scl]
1185 (0,0) arc (15:330:1) -- (-0.6,-0.3) -- cycle;
1186 \draw[black, line width=0.05ex*\tkzsymbols@scl] (-0.5,0.3) -- (-0.3,0.1);
1187 \shade[ball color=black] (-0.45,0.27) arc (100:350:0.1);
1188 \shade[ball color=#3] (1.8,-0.5) .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
1189 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
1190 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
1191 \shade[ball color=#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) .. controls
1192 (1.7,-1.2) and (1.3,-1.2) .. (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
1193 \end{tikzpicture}%
1194 \fi%
1195 }%
1196 \tikzsymbols@use@box{dVomey#1#2#3}%
1197 \tikzsymbolsaftersymbolinput%
1198 }

```

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

```

1199 \tkzsymbols@Declare@Robust@Commandx{Walley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
1200 \tikzsymbols@ifsaveboxundefined{Walley#1#2#3}{%
1201 \set@tkzsymbolsscl{#1}%
1202 \iftikzymbols@draftabsolute%
1203 \tkzsymbols@bx@Prmtrstore{#1}%
1204 \tikzsymbols@draftbox{2.341ex*\real{\tkzsymbols@Prmtr}}{1.674ex*\real{\tkzsymbols@Prmtr}}%
1205 \else%
1206 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1207 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1,
1208 decoration={random steps,segment length=0.15ex*\tkzsymbols@scl, amplitude=0.1ex*\tkzsymbols@scl}]
1209 \fill[#2, line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.28);
1210 \draw[line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.28);
1211 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
1212 \draw (0.28,-0.33) rectangle (0.66,0.33);
1213 \draw[line width=0.06ex*\tkzsymbols@scl]
1214 (0.28,0) ---+(0.05,0.07) ---+(0.03,0.02) ---
1215 +(0.03,-0.02) ---+(0.03,0.1) ---+(0.03,0.02) -- (0.5,0.25);
1216 \draw[line width=0.06ex*\tkzsymbols@scl]
1217 (0.28,0) ---+(0.06,-0.02) ---+(0.04,0.06) ---
1218 +(0.0,-0.08) ---+(0.08,0.06) ---+(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);
1219 \draw[line width=0.06ex*\tkzsymbols@scl]
1220 (0.28,0) ---+(0.03,-0.02) ---+(0.03,-0.07) ---
1221 +(0.03,-0.01) ---+(0.01,-0.07) ---+(0.06,0.01) ---+(0.03,-0.08) --

```

```

1222 (0.5,0.-0.25);
1223 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1224 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1225 \end{tikzpicture}%
1226 \fi%
1227 }%
1228 \tikzsymbols@use@box{Walley#1#2#3}%
1229 \tikzsymbolsaftersymbolinput%
1230 }
1231 \tikzsymbols@Declare@Robust@Commandx{rWalley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
1232 \tikzsymbols@ifsaveboxundefined{rWalley#1#2#3}{%
1233 \set@tkzsymbolsscl{#1}%
1234 \iftikzymbols@draftabsolute%
1235 \tkzsymbols@bx@Prmtrstore{#1}%
1236 \tikzsymbols@draftbox{2.341ex*\real{\tkzsymbols@Prmtr}}{1.674ex*\real{\tkzsymbols@Prmtr}}%
1237 \else%
1238 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1239 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1,
1240 decoration={random steps,segment length=0.15ex*\tkzsymbols@scl, amplitude=0.1ex*\tkzsymbols@scl}]
1241 \fill[#2, line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.28);
1242 \draw[line width=0.08ex*\tkzsymbols@scl] (0,0) circle (0.28);
1243 \fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
1244 \draw (0.28,-0.33) rectangle (0.66,0.33);
1245 \draw[decorate, line width=0.06ex*\tkzsymbols@scl] (0.28,0) -- (0.5,0.25);
1246 \draw[decorate,line width=0.06ex*\tkzsymbols@scl] (0.28,0) -- (0.6,0.0);
1247 \draw[decorate,line width=0.06ex*\tkzsymbols@scl] (0.28,0) -- (0.5,-0.25);
1248 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1249 \draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
1250 \end{tikzpicture}%
1251 \fi%
1252 }%
1253 \tikzsymbols@use@box{rWalley#1#2#3}%
1254 \tikzsymbolsaftersymbolinput%
1255 }
1256 \tikzsymbols@Declare@Robust@Commandx{dWalley}[2][1=1, 2=yellow, usedefault]{%
1257 \tikzsymbols@ifsaveboxundefined{dWalley#1#2}{%
1258 \set@tkzsymbolsscl{#1}%
1259 \iftikzymbols@draftabsolute%
1260 \tkzsymbols@bx@Prmtrstore{#1}%
1261 \tikzsymbols@draftbox{2.4288ex*\real{\tkzsymbols@Prmtr}}{1.6008ex*\real{\tkzsymbols@Prmtr}}%
1262 \else%
1263 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1264 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymbols@scl,scale=#1]
1265 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
1266 \draw[line width=0.06ex*\tkzsymbols@scl]
1267 (0.28,0) ---+(0.05,0.07) ---+(0.03,0.02) ---+
1268 +(0.03,-0.02) ---+(0.03,0.1) ---+(0.03,0.02) -- (0.5,0.25);
1269 \draw[line width=0.06ex*\tkzsymbols@scl]
1270 (0.28,0) ---+(0.06,-0.02) ---+(0.04,0.06) ---+
1271 +(0.0,-0.08) ---+(0.08,0.06) ---+(0.03,-0.02) ---+(0.08,0.02) -- (0.6,0.0);

```

```

1272 \draw[line width=0.06ex*\tkzsymls@scl]
1273 (0.28,0) ---+(0.03,-0.02) ---+(0.03,-0.07) ---+
1274 +(0.03,-0.01) ---+(0.01,-0.07) ---+(0.06,0.01) ---+(0.03,-0.08) -- (0.5,0.-0.25);
1275 \shade[ball color=#2] (-0.01,0) circle (0.31);
1276 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1277 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0,-0.1);
1278 \end{tikzpicture}%
1279 \fi%
1280 }%
1281 \tikzsymbols@use@box{dWalley#1#2}%
1282 \tikzsymbolsaftersymbolinput%
1283 }
1284 \tkzsymls@Declare@Robust@Commandx{drWalley}[2][1=1, 2=yellow, usedefault]{%
1285 \tikzsymbols@ifsaveboxundefined{drWalley#1#2}{%
1286 \set@tkzsymls@scl{#1}%
1287 \iftikzsymbols@draftabsolute%
1288 \tkzsymls@bx@Prmtrstore{#1}%
1289 \tikzsymbols@draftbox{2.4288ex*\real{\tkzsymls@Prmtr}}{1.6008ex*\real{\tkzsymls@Prmtr}}%
1290 \else%
1291 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1292 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1,
1293 decoration={random steps,segment length=0.15ex*\tkzsymls@scl, amplitude=0.1ex*\tkzsymls@scl}]
1294 \shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
1295 \draw[decorate, line width=0.06ex*\tkzsymls@scl] (0.298,0) -- (0.5,0.25);
1296 \draw[decorate,line width=0.06ex*\tkzsymls@scl] (0.298,0) -- (0.6,0.0);
1297 \draw[decorate,line width=0.06ex*\tkzsymls@scl] (0.298,0) -- (0.5,-0.25);
1298 \shade[ball color=#2, line width=0.08ex*\tkzsymls@scl] (-0.01,0) circle (0.31);
1299 \draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
1300 \draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0,-0.1);
1301 \end{tikzpicture}%
1302 \fi%
1303 }%
1304 \tikzsymbols@use@box{drWalley#1#2}%
1305 \tikzsymbolsaftersymbolinput%
1306 }

```

\Cat \*Miau\*

```

1307 \tkzsymls@Declare@Robust@Commandx{Cat}[1][1=1,usedefault]{%
1308 \tikzsymbols@ifsaveboxundefined{Cat#1}{%
1309 \set@tkzsymls@scl{#1}%
1310 \iftikzsymbols@draftabsolute%
1311 \tkzsymls@bx@Prmtrstore{#1}%
1312 \tikzsymbols@draftbox{1.899ex*\real{\tkzsymls@Prmtr}}{1.957ex*\real{\tkzsymls@Prmtr}}%
1313 \else%
1314 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1315 \begin{tikzpicture}[x=2.33ex,y=2.33ex, line width=0.093ex*\tkzsymls@scl,scale=#1]
1316 \draw (0,0) circle (0.3);
1317 \draw[rounded corners=0.163ex*\tkzsymls@scl] (-0.3,0) -- (-0.35,0.5) -- (0,0.3);
1318 \draw[rounded corners=0.163ex*\tkzsymls@scl] (0,0.3) -- (0.35,0.5) -- (0.3,0);
1319 \fill (-0.15,.15) circle (0.05);

```



```

1320 \fill (0.15,.15) circle (0.05);
1321 \draw[rounded corners=0.175ex*\tkzsymls@scl,yshift=-0.12ex]
1322   (0,0) -- (0,-0.1) -- (-0.1,-0.095);
1323 \draw[rounded corners=0.175ex*\tkzsymls@scl,yshift=-0.12ex]
1324   (0,0) -- (0,-0.1) -- (0.1,-0.095);
1325 \draw[rounded corners=.12ex*\tkzsymls@scl,yshift=-.15ex,
1326   line width=0.03em*\real{0.9}*\tkzsymls@scl]
1327 (-0.1,0.1) -- (0,0) -- (0.1,0.1) -- cycle ;
1328 \draw[line width=0.035ex*\tkzsymls@scl]
1329   (-0.1,-0.05)..controls(-0.25,0)and(-0.35,0).. (-0.4,-0.05);
1330 \draw[line width=0.035ex*\tkzsymls@scl](-0.1,-0.05)..
1331   controls(-0.25,-0.01)and(-0.35,-0.09).. (-0.4,-0.14);
1332 \draw[line width=0.035ex*\tkzsymls@scl](-0.1,-0.05)..
1333   controls(-0.25,-0.045)and(-0.35,-0.13).. (-0.4,-0.23);
1334 \draw[line width=0.035ex*\tkzsymls@scl]
1335   (0.1,-0.05)..controls(0.25,0)and(0.35,0).. (0.4,-0.05);
1336 \draw[line width=0.035ex*\tkzsymls@scl]
1337   (0.1,-0.05)..controls(0.25,-0.01)and(0.35,-0.09).. (0.4,-0.14);
1338 \draw[line width=0.035ex*\tkzsymls@scl]
1339   (0.1,-0.05)..controls(0.25,-0.045)and(0.35,-0.13).. (0.4,-0.23);
1340 \end{tikzpicture}%
1341 \fi%
1342 }%
1343 \tikzsymbols@use@box{Cat#1}%
1344 \tikzsymbolsaftersymbolinput%
1345 }

```

\Ninja \dNinja A Ninja.

```

1346 \tkzsymbols@Declare@Robust@Commandx{Ninja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
1347 \tikzsymbols@ifsaveboxundefined{Ninja#1#2#3#4}{%
1348 \set@tkzsymls@#1}%
1349 \iftikzsymbols@draftabsolute%
1350 \tkzsymbols@bx@Prmtrstore{#1}%
1351 \tikzsymbols@draftbox{2.149ex*\real{\tkzsymls@Prmtr}}{1.717ex*\real{\tkzsymls@Prmtr}}%
1352 \else%
1353 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@#1\fi%
1354 \def\Black@is@Black{black}%
1355 \def\Black@or@not@Black{#2}%
1356 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1,
1357 decoration={random steps,segment length=0.1ex*\tkzsymls@scl, amplitude=0.1ex*\tkzsymls@scl}]
1358 \fill[#2] (0,0) circle (0.33);
1359 \draw (-0.2,-0.125) -- ++(0.4,0);
1360 \fill[decoration={random steps,segment length=0.1ex*\tkzsymls@scl,
1361   amplitude=0.01ex*\tkzsymls@scl}, decorate,#3]
1362 (-0.33,0) -- (0.33,0) -- (0.23,0.23) -- (-0.23,0.23) -- cycle;
1363 \ifx\Black@or@not@Black\Black@is@Black
1364   \draw[line width=0.08ex*\tkzsymls@scl] (0,0) circle (0.33);\fi
1365 \fill[#3] (0,0.1) -- (-0.33,0) -- (-0.26,0.23);
1366 \fill[#3] (0.3465,0) arc (0:42:0.34 and 0.345) --
1367 (0.2,0.23)-- (0.31,0.0) -- cycle;

```

```

1368 \fill[#3] (-0.3465,0) arc (0:-42:-0.34 and -0.345) --
1369 (-0.2,0.23)-- (-0.31,0.0) -- cycle;
1370 \fill[#4] (0.129,0.1425) arc (55:-180:.05);
1371 \fill[#4] (-0.129,0.1425) arc (-55:180:-.05);
1372 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymls@scl,
1373 segment length=0.55ex*\tkzsymls@scl}, #3]
1374 (0.26,0.21) -- (0.5,0.35);
1375 \draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymls@scl,
1376 segment length=0.55ex*\tkzsymls@scl}, #3]
1377 (0.26,0.21) -- (0.53,0.1);
1378 \ifx\Black@or@not@Black\Black@is@Black
1379 \else\draw[line width=0.08ex*\tkzsymls@scl] (0,0) circle (0.33);\fi
1380 \end{tikzpicture}%
1381 \fi%
1382 }%
1383 \tikzsymbols@use@box{Ninja#1#2#3#4}%
1384 \tikzsymbolsaftersymbolinput%
1385 }
1386 \tkzsymls@Declare@Robust@Commandx{dNinja}[4][1=, 2=black, 3=red, 4=white, usedefault]{%
1387 \tikzsymbols@ifsaveboxundefined{dNinja#1#2#3#4}{%
1388 \set@tkzsymls@scl{#1}%
1389 \iftikzsymbols@draftabsolute%
1390 \tkzsymls@bx@Prmtrstore{#1}%
1391 \tikzsymbols@draftbox{2.1498ex*\real{\tkzsymls@Prmtr}}{1.7178ex*\real{\tkzsymls@Prmtr}}%
1392 \else%
1393 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1394 \def\Black@is@Black{black}%
1395 \def\Black@or@not@Black{#2}%
1396 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymls@scl,scale=#1,
1397 decoration={random steps,segment length=0.1ex*\tkzsymls@scl, amplitude=0.1ex*\tkzsymls@scl}]
1398 \draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymls@scl,
1399 segment length=0.55ex*\tkzsymls@scl},decorate, #3!50!black]
1400 (0.26,0.21) -- (0.5,0.35);
1401 \draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymls@scl,
1402 segment length=0.5ex*\tkzsymls@scl},decorate, #3!50!black]
1403 (0.26,0.21) -- (0.53,0.1);
1404 \shade[ball color=#2] (0,0) circle (0.347);
1405 \draw (-0.2,-0.125) -- ++(0.4,0);
1406 \fill[decoration={random steps,segment length=0.1ex*\tkzsymls@scl,
1407 amplitude=0.01ex*\tkzsymls@scl},ball color=#3]
1408 decorate {(-0.33,0) -- (0.3465,0) }
1409 {arc (0:42:0.34 and 0.345)}
1410 decorate {-- (-0.25,0.24)}
1411 { arc (-42:0:-0.375 and -0.345)};
1412 \shade[ball color=#4] (0.129,0.1425) arc (55:-180:.05);
1413 \shade[ball color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1414 \shade[top color=#4!80!black, bottom color=#4] (0.129,0.1425) arc (55:-180:.05);
1415 \shade[top color=#4!80!black, bottom color=#4] (-0.129,0.1425) arc (-55:180:-.05);
1416 \end{tikzpicture}%
1417 \fi%

```

```

1418 }%
1419 \tikzsymbols@use@box{dNinja#1#2#3#4}%
1420 \tikzsymbolsaftersymbolinput%
1421 }

```

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

```

1422 \tikzsymbols@Declare@Robust@Commandx{NiceReapey}[2][1=1,2={black!20!white},usedefault]{%
1423 \tikzsymbols@ifsaveboxundefined{NiceReapey#1#2}{%
1424 \set@tkzsymbolsscl{#1}%
1425 \iftikzymbols@draftabsolute%
1426 \tkzsymbols@bx@Prmtrstore{#1}%
1427 \tikzsymbols@draftbox{(1.1067em+0.07ex)*\real{\tkzsymbols@Prmtr}}
1428 \quad (0.693em+0.07ex)*\real{\tkzsymbols@Prmtr}}%
1429 \else%
1430 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1431 \begin{tikzpicture}[x=0.11em,y=0.11em, line width=0.07ex*\tkzsymbols@scl,scale=#1]
1432 \draw[] (1.7,-1) arc (360:180:1.7 and 2)
1433 \quad arc (260:110:1.5 and 2) .. controls (-1,3.3) and (1,3.3) .. (1.9,2.97)
1434 \quad arc (260:100:-1.3 and -2) -- cycle;
1435 \fill[#2] (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5) -- cycle;
1436 \draw (3,-3) -- (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5);
1437 \draw (0,-1.5) circle (1 and 0.5);
1438 \draw[line width=0.04ex*\tkzsymbols@scl] (-0.2,-1) -- (-0.2,-2);
1439 \draw[line width=0.04ex*\tkzsymbols@scl] (0.2,-1) -- (0.2,-2);
1440 \draw[line width=0.04ex*\tkzsymbols@scl] (0.6,-1) -- (0.6,-2);
1441 \draw[line width=0.04ex*\tkzsymbols@scl] (-0.6,-1) -- (-0.6,-2);
1442 \draw[line width=0.04ex*\tkzsymbols@scl] (-1,-1.5) -- (1,-1.5);
1443 \fill (1.25,1.25) circle (0.5 and 0.75);
1444 \fill (-1.25,1.25) circle (0.5 and 0.75);
1445 \end{tikzpicture}%
1446 \fi%
1447 }%
1448 \tikzsymbols@use@box{NiceReapey#1#2}
1449 \tikzsymbolsaftersymbolinput%
1450 }

```

### 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 ...

```

1451 \newcommand{\tikzsymbols@Strichmaxerl@XCheck}[1]{%
1452 \ifdim #1 pt > 0pt%
1453 \quad \ifdim #1 pt < 0.18pt%

```

```

1454     \pgfmathsetmacro{#1}{0}%
1455     \else%
1456     \pgfmathsetmacro{#1}{#1-0.18}%
1457     \fi%
1458 \else%
1459     \ifdim #1 pt >-0.18pt%
1460     \pgfmathsetmacro{#1}{0}%
1461     \else%
1462     \pgfmathsetmacro{#1}{#1+0.18}%
1463     \fi%
1464 \fi%
1465 }

```

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

```

1466 \newcommand{\tikzsymbols@Strichmaxerl@ifSmallerZero}[1]{%
1467 \ifdim #1 pt < 0pt%
1468     \pgfmathsetmacro{#1}{0}%
1469 \fi%
1470 }

```

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

```

1471 \tkzsymls@Declare@Robust@Commandx{Strichmaxerl}[5][1=1,2=-22,3=22,4=27,5=-27,usedefault]{%
1472 \tikzsymbols@ifsaveboxundefined{Strichmaxerl#1#2#3#4#5}{%
1473 \set@tkzsymls@#1}%
1474 \iftikzsymbols@draftabsolute%

```

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

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

```

1475 \pgfmathsetmacro{\tikzsymbols@x@LA}{-0.27*cos(#2)}%
1476 \pgfmathsetmacro{\tikzsymbols@x@RA}{0.27*cos(#3)}%
1477 \pgfmathsetmacro{\tikzsymbols@x@LB}{0.34*sin(#4)}%
1478 \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*).

```

1479 \pgfmathsetmacro{\tikzsymbols@y@LA}{0.27*sin(#2)}%
1480 \pgfmathsetmacro{\tikzsymbols@y@RA}{-0.27*sin(#3)}%
1481 \pgfmathsetmacro{\tikzsymbols@y@LB}{0.34*cos(#4)}%
1482 \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).

```
1483 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@LA}%
1484 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@RA}%
1485 \tikzsymbols@Strichmaxerl@XCheck{\tikzsymbols@x@LB}%
1486 \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).

```
1487 \pgfmathsetmacro{\tikzsymbols@x@max}
1488     {\max(0,\tikzsymbols@x@LA,\tikzsymbols@x@RA,\tikzsymbols@x@LB,\tikzsymbols@x@RB)}%
1489 \pgfmathsetmacro{\tikzsymbols@x@min}
1490     {\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)

```
1491 \pgfmathsetmacro{\tikzsymbols@y@LA}{\tikzsymbols@y@LA-0.2}%
1492 \pgfmathsetmacro{\tikzsymbols@y@RA}{\tikzsymbols@y@RA-0.2}%
```

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

```
1493 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@LA}%
1494 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@RA}%
1495 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@LB}%
1496 \tikzsymbols@Strichmaxerl@ifSmallerZero{\tikzsymbols@y@RB}%
```

And find the greatest number.

```
1497 \pgfmathsetmacro{\tikzsymbols@y@max}
1498 {\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:

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

```
1499 \tkzsymls@bx@Prmtrstore{#1}%
1500 \tikzsymbols@draftbox{(0.606ex+1.35ex*\real{\tikzsymbols@x@max}
1501     -1.35ex*\real{\tikzsymbols@x@min})*\real{\tkzsymls@Prmtr}}%
1502 {(1.173ex+1.35ex*\real{\tikzsymbols@y@max})*\real{\tkzsymls@Prmtr}}%
1503 \else%
1504 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1505 \begin{tikzpicture}[line width=0.12ex*\tkzsymls@scl, scale=#1, x=1.35ex, y=1.35ex]
1506 \draw[rotate around={#5:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1507 \draw[rotate around={#4:(0.15,0.2)}] (0.15,0.2) -- (0.15,-0.14);
1508 \draw (.15,.2) -- (.15,.4);
1509 \draw[rotate around={#3:(.15,.4)}] (0.15,0.4) -- (0.42,0.4);
```

```

1510 \draw[rotate around={#2:(.15,.4)}] (0.15,0.4) -- (-0.12,0.4);
1511 \draw (.15, .4) -- (.15, .53);
1512 \draw (.15,.8) circle (0.18);
1513 \end{tikzpicture}%
1514 \fi%
1515 }%
1516 \tikzsymbols@use@box{Strichmaxerl#1#2#3#4#5}%
1517 \tikzsymbolsaftersymbolinput%
1518 }
1519 \tkzsymbols@Declare@Robust@Command{Person}{%
1520 \PackageWarning{tikzsymbols}{Command '\protect\Person' is obsolete,
1521 \MessageBreak Please use '\protect\Strichmaxerl' instead.\MessageBreak}%
1522 \Strichmaxerl%
1523 }

```

`\Candle` A burning candle

```

1524 \tkzsymbols@Declare@Robust@Command{Candle}[1][1]{%
1525 \tikzsymbols@ifsaveboxundefined{Candle#1}{%
1526 \set@tkzsymbolsscl{#1}%
1527 \iftikzymbols@draftabsolute%
1528 \tkzsymbols@bx@Prmtrstore{#1}%
1529 \tikzsymbols@draftbox{0.64ex*\real{\tkzsymbols@Prmtr}}{(1.255ex+2.2pt)*\real{\tkzsymbols@Prmtr}}%
1530 \else%
1531 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1532 \begin{tikzpicture}[x=1ex, y=1ex, scale=#1, line width=0.07ex*\tkzsymbols@scl]
1533 \draw[rounded corners=0.04ex*\tkzsymbols@scl] (0,0) -- (0.2,0) -- +(0,1) -- (0,1) -- cycle;
1534 \draw[line width=0.05ex*\tkzsymbols@scl] (0.1,1) -- (0.1,1.2);
1535 \draw[xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymbols@scl]
1536 (-0.1,0.6) .. controls (-0.4,0.8) and (-0.1,1) .. (-0.1,1.2);
1537 \draw [xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymbols@scl]
1538 (-0.1,0.6) .. controls (0.2,0.8) and (-0.1,1) .. (-0.1,1.2);
1539 \end{tikzpicture}%
1540 \fi%
1541 }%
1542 \tikzsymbols@use@box{Candle#1}%
1543 \tikzsymbolsaftersymbolinput%
1544 }

```

`\Fire` Just a fire.

```

1545 \tkzsymbols@Declare@Robust@Command{Fire}[1][1]{%
1546 \tikzsymbols@ifsaveboxundefined{Fire#1}{%
1547 \set@tkzsymbolsscl{#1}%
1548 \iftikzymbols@draftabsolute%
1549 \tkzsymbols@bx@Prmtrstore{#1}%
1550 \tikzsymbols@draftbox{1.576ex*\real{\tkzsymbols@Prmtr}}{1.639ex*\real{\tkzsymbols@Prmtr}}%
1551 \else%
1552 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1553 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.07ex*\tkzsymbols@scl,rotate=45, scale=#1]
1554 \fill (-0.05,0) -- (0.05,0) -- (0.05,0.95) -- (-0.05,0.95) -- cycle;
1555 \fill (-0.74,0.7) -- (0.19,0.7) -- (0.19,0.8) -- (-0.74,0.8) -- cycle;

```

```

1556 \fill[rotate=-20, xshift=-1.3, yshift=-0.1]
1557 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1558 \fill[rotate=-70, xshift=-3.3, yshift=-2.3]
1559 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1560 \fill[rotate=135, xshift=2.5, yshift=-3.8]
1561 (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
1562 \draw[rotate=-45, xshift=-2.6, yshift=1.5, line width=0.04ex*\tkzsymbols@scl, x=0.5ex, y=0.5ex]
1563 (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0.05,1.7);
1564 \draw[rotate=-45, xshift=-2.1, yshift=1.5, line width=0.04ex*\tkzsymbols@scl, x=0.5ex, y=0.5ex]
1565 (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.15,1.7);
1566 \draw[rotate=-45, xshift=-2.5] (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) .. (0,1.5);
1567 \draw[rotate=-45, xshift=-2] (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) .. (-0.1,1.5);
1568 \end{tikzpicture}%
1569 \fi%
1570 }%
1571 \tikzsymbols@use@box{Fire#1}%
1572 \tikzsymbolsaftersymbolinput%
1573 }

```

`\Coffeecup` Just a cup of coffee.

```

1574 \ifKV@tikzsymbols@marvosym \else%
1575 \tikzsymbols@Declare@Robust@Command{Coffeecup}[1][1]{%
1576 \tikzsymbols@ifsaveboxundefined{Coffeecup#1}{%
1577 \set@tkzsymbolsscl{#1}%
1578 \iftikzymbols@draftabsolute%
1579 \tkzsymbols@bx@Prmtrstore{#1}%
1580 \tikzsymbols@draftbox{1.82ex*\real{\tkzsymbols@Prmtr}}{1.705ex*\real{\tkzsymbols@Prmtr}}%
1581 \else%
1582 \ifdim\tkzsymbols@scl<Opt\set@tkzsymbolsscl{#1}\fi%
1583 \begin{tikzpicture}[x=0.7ex,y=0.7ex, scale=#1, line width=0.07ex*\tkzsymbols@scl,
1584 decoration={snake,amplitude=.05ex*\tkzsymbols@scl,segment length=0.408ex*\tkzsymbols@scl}]
1585 \draw (0,0) arc (180:270:0.8 and 1) -- ++(0.5,0) arc (270:360:0.8 and 1) -- cycle;
1586 \draw (2.1,-0.15) -- (2.2,-0.15) arc (90:-90:0.3) -- (1.85, -0.75);
1587 \draw[line width=0.05ex*\tkzsymbols@scl, decorate]
1588 (0.4,0.3) -- +(0,1);
1589 \draw[line width=0.05ex*\tkzsymbols@scl, decorate]
1590 (1,0.3) -- +(0,1);
1591 \draw[line width=0.05ex*\tkzsymbols@scl, decorate]
1592 (1.6,0.3) -- +(0,1);
1593 \draw (0,-1.05) -- (2.1,-1.05);
1594 \end{tikzpicture}%
1595 \fi%
1596 }%
1597 \tikzsymbols@use@box{Coffeecup#1}%
1598 \tikzsymbolsaftersymbolinput%
1599 }%
1600 \fi

```

`\Chair` A chair.

```

1601 \tkzsymbols@Declare@Robust@Command{Chair}[1][1]{%
1602 \tikzsymbols@ifsaveboxundefined{Chair#1}{%
1603 \set@tkzsymbolsscl{#1}%
1604 \iftikzymbols@draftabsolute%
1605 \tkzsymbols@bx@Prmtrstore{#1}%
1606 \tikzsymbols@draftbox{0.97ex*\real{\tkzsymbols@Prmtr}}{1.69ex*\real{\tkzsymbols@Prmtr}}%
1607 \else%
1608 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\@tkzssmbles@negtrue\fi%
1609 \begin{tikzpicture}[x=0.9ex,y=0.9ex, scale=#1, line width=0.07ex*\tkzsymbols@scl]
1610 \draw (0,-0.5) -- (0,0.7) -- (0.5,1) -- (0.5,0.25);
1611 \draw[line width=0.06ex*\tkzsymbols@scl] (0,0.4) -- (0.5,0.7);
1612 \draw (0,0) -- (0.5,0.3) -- (1,0) --(1,-0.5);
1613 \if@tkzssmbles@neg\draw (0.5,0.3) -- +(0,-0.5);\fi%
1614 \draw (0.5,-0.3) -- (0.5,-0.8);
1615 \draw (1,0) -- (0.5,-0.3) -- (0,0);
1616 \end{tikzpicture}%
1617 \fi%
1618 }%
1619 \tikzsymbols@use@box{Chair#1}%
1620 \@tkzssmbles@negfalse%
1621 \tikzsymbolsaftersymbolinput%
1622 }

```

\Bed A bed.

```

1623 \tkzsymbols@Declare@Robust@Command{Bed}[1][1]{%
1624 \tikzsymbols@ifsaveboxundefined{Bed#1}{%
1625 \set@tkzsymbolsscl{#1}%
1626 \iftikzymbols@draftabsolute%
1627 \tkzsymbols@bx@Prmtrstore{#1}%
1628 \tikzsymbols@draftbox{3.08ex*\real{\tkzsymbols@Prmtr}}{1.68ex*\real{\tkzsymbols@Prmtr}}%
1629 \else%
1630 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1631 \begin{tikzpicture}[x=1ex,y=1ex, scale=#1, line width=0.08ex*\tkzsymbols@scl]
1632 \draw (0,0) -- (0,1.6);
1633 \draw (3,0) -- (3,1.2);
1634 \draw (0,0.5) -- (3,0.5);
1635 \draw (0,0.35) -- (3,0.35);
1636 \draw (0.7,0.5) arc (0:90:0.7);
1637 \draw (0.7,0.5) arc(180:30:1.231 and 0.6);
1638 \end{tikzpicture}%
1639 \fi%
1640 }%
1641 \tikzsymbols@use@box{Bed#1}%
1642 \tikzsymbolsaftersymbolinput%
1643 }

```

\Tribar Also called Penrose-Triangle

```

1644 \tkzsymbols@Declare@Robust@Commandx{Tribar}[4]
1645 [1=1,2={opacity=0},3={opacity=0},4={opacity=0},usedefault]
1646 {%

```



```

1647 \tikzsymbols@ifsaveboxundefined{Tribar#1#2#3#4}{%
1648 \set@tkzsymbolsscl{#1}%
1649 \iftikzymbols@draftabsolute%
1650 \tkzsymbols@bx@Prmtrstore{#1}%
1651 \tikzsymbols@draftbox{1.7175ex*\real{\tkzsymbols@Prmtr}}{1.685ex*\real{\tkzsymbols@Prmtr}}%
1652 \else%
1653 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1654 \begin{tikzpicture}[x=0.65ex,y=0.65ex,scale=#1,
1655   rounded corners=0.03ex*\tkzsymbols@scl, line width=0.06ex*\tkzsymbols@scl]
1656 \fill[#2] (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1657   -- (-0.65,-0.6) -- (0.35,1.3) -- +(0.15,-0.3);
1658 \fill[#3] (0,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- (-0.1,-0.3);
1659 \fill[#4] (1,0) -- (0.35,1.3) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (1.3,0);
1660 \draw (0,0) -- (1,0) -- (0.5,1) -- cycle;
1661 \draw (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
1662   -- (-0.65,-0.6) -- (0.35,1.3) -- (0.8,.4);
1663 \draw (0.9,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- +(-0.05,-0.1);
1664 \draw (-0.6,-0.6) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (0.4,1.9);
1665 \end{tikzpicture}%
1666 \fi%
1667 }%
1668 \tikzsymbols@use@box{Tribar#1#2#3#4}%
1669 \tikzsymbolsaftersymbolinput%
1670 }

```

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

```
1671 \newcommand{\tikzsymbolsMoaitickness}{}
```

\Moai From the Easter Island: a Moai.

```

1672 \tkzsymbols@Declare@Robust@Commandx{Moai}[1][1=1,usedefault]{%
1673 \tikzsymbols@ifsaveboxundefined{Moai#1}{%
1674 \set@tkzsymbolsscl{#1}%
1675 \tkzsymbols@bx@Prmtrstore{#1}%
1676 \ifdim\tkzsymbols@scl<0pt\set@tkzsymbolsscl{-#1}\fi%
1677 \ifdim\tkzsymbols@scl<2pt%
1678 \def\tikzsymbolsMoaitickness{0.05ex}%
1679 \else%
1680 \ifdim\tkzsymbols@scl<5pt%
1681 \def\tikzsymbolsMoaitickness{0.035ex}%
1682 \else%
1683 \def\tikzsymbolsMoaitickness{0.03ex}%
1684 \fi\fi%
1685 \iftikzymbols@draftabsolute%
1686 \tikzsymbols@draftbox{(1.001ex+\tikzsymbolsMoaitickness)*\real{\tkzsymbols@Prmtr}}
1687   {(1.664ex+\tikzsymbolsMoaitickness)*\real{\tkzsymbols@Prmtr}}%
1688 \else%
1689 \begin{tikzpicture}[x=.13ex, y=.13ex, rounded corners=0.01ex*\tkzsymbols@scl, scale=#1,
1690   line width=\tikzsymbolsMoaitickness*\tkzsymbols@scl]
1691 \draw (-2.6,-4.25) -- (-2.5,-5.8)
1692 ..controls (-2,-6.8) and (1.5,-6.8) .. (2.2,-5.8) -- (2.4,-3.95);

```

```

1693 \draw(-2.5,2.5) .. controls (-2.9,4.6) and (2,5) .. (3.3,2.5) -- (2.9,-3.4)
1694 .. controls (2,-5) and (-4,-5) .. (-3.1,-3) -- cycle;
1695 \draw (-2.5,3) -- (-2,5) .. controls (0,6) and (2,5.8) .. (3.1,4.7) -- (3.3,2.5);
1696 \draw[line width=0.02ex*\tkzsymls@scl]
1697   (-2.2,-1.8) .. controls (-1,-1.3) and (0,-1.7) .. (1,-2);
1698 \draw[line width=0.02ex*\tkzsymls@scl]
1699   (-2.2,-1.8) .. controls (-1,-1) and (0,-1.4) .. (1,-2);
1700 \draw[line width=0.02ex*\tkzsymls@scl]
1701   (-2.2,-1.8) .. controls (-1,-2) and (0,-2) .. (1,-2);
1702 \draw (-0.8,4) .. controls (-0.8,3) and (-0.8,2) .. (-1.6,0.5) -- (-1.8,-0.4)
1703   .. controls (-1,0.2) and (0,0.2) .. (0.6,-0.4) -- (0.7,0.4)
1704   .. controls (0,1) and (0,2) .. (0.8,4);
1705 \draw (-1.8,-0.36) .. controls (-0.5,-0.5) and (0,-0.5) .. (0.6,-0.36);
1706 \draw (3.2,3.5) -- (3.7,3.5) .. controls (3.5,2) and (3.5,2) .. (3.6,-1.5) -- (3,-1.9);
1707 \draw (-2.5,3) .. controls (-2.7,2) and (-3,1) .. (-2.88,-1);
1708 \draw (-2.5,2.8) .. controls (-2,2.5) and (-1,3) .. (-0.8,3.1);
1709 \draw (0.5,3.3) .. controls (1,3) and (1,2.5) .. (3.3,2.4);
1710 \end{tikzpicture}%
1711 \fi%
1712 }%
1713 \tikzsymbols@use@box{Moai#1}%
1714 \tikzsymbolsaftersymbolinput%
1715 }

```

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

```

1716 \tikzsymbols@Declare@Robust@Command{Snowman}[1][1]{%
1717 \tikzsymbols@ifsaveboxundefined{Snowman#1}{%
1718 \set@tkzsymls@scl{#1}%
1719 \iftikzymbols@draftabsolute%
1720 \tkzsymls@bx@Prmtrstore{#1}%
1721 \tikzsymbols@draftbox{1.545ex*\real{\tkzsymls@Prmtr}}{1.772ex*\real{\tkzsymls@Prmtr}}%
1722 \else%
1723 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1724 \begin{tikzpicture}[x=0.9ex,y=0.9ex,line width=0.07ex*\tkzsymls@scl, scale=#1]
1725 \draw (0,0) circle (0.4 and 0.35);
1726 \draw[line width=0.06ex*\tkzsymls@scl] (0,0.64) circle (0.3 and 0.28);
1727 \draw[line width=0.05ex*\tkzsymls@scl] (0,1.14) circle (0.2 and 0.2);
1728 \draw[rounded corners=0.1ex*\tkzsymls@scl,line width=0.05ex*\tkzsymls@scl,
1729   rotate around={-30:(0,1.14)}]
1730   (-0.2,1.15) -- ++(0,0.35) -- +(0.4,0) -- (0.2,1.14);
1731 \draw[rounded corners=0.07ex*\tkzsymls@scl,line width=0.05ex*\tkzsymls@scl,
1732   rotate around={-30:(0,1.14)}]
1733   (-0.2,1.19) arc (270:90:0.1);
1734 \fill (0,0.78) circle (0.04);
1735 \fill (0,0.63) circle (0.04);
1736 \fill (0,0.48) circle (0.04);
1737 \fill (0,0.2) circle (0.05);
1738 \fill (0,0) circle (0.05);
1739 \fill (0,-0.2) circle (0.05);
1740 \fill (-0.06,1.18) circle (0.045);

```

```

1741 \fill (0.06,1.18) circle (0.045);
1742 \fill (0.1,1.08) circle (0.015);
1743 \fill (-0.1,1.08) circle (0.015);
1744 \fill (0.06,1.055) circle (0.015);
1745 \fill (-0.06,1.055) circle (0.015);
1746 \fill (0.02,1.039) circle (0.015);
1747 \fill (-0.02,1.039) circle (0.015);
1748 \draw (-0.3,0.7) -- (-0.6,0.8);
1749 \draw (-0.6,0.8) -- (-0.75,0.7);
1750 \draw (-0.6,0.8) -- (-0.55,1);
1751 \draw (-0.6,0.8) -- (-0.8,0.9);
1752 \draw[line width=0.06ex*\tkzsymbols@sc1] (-0.65,0) -- (-0.65,1);
1753 \foreach\x in {-0.85, -0.75,-0.65,-0.55,-0.45}
1754 \draw[line width=0.05ex*\tkzsymbols@sc1] (-0.65,1) -- (\x,1.3);
1755 \draw (0.3,0.7) -- (0.6,0.8);
1756 \draw (0.6,0.8) -- (0.75,0.7);
1757 \draw (0.6,0.8) -- (0.6,1);
1758 \draw (0.6,0.8) -- (0.8,0.9);
1759 \end{tikzpicture}%
1760 \fi%
1761 }%
1762 \tikzsymbols@use@box{Snowman#1}%
1763 \tikzsymbolsaftersymbolinput%
1764 }

```

## 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:

```

1765 \tikzsymbolsnewcommand{BasicTree}[5][1]{%
1766 \tikzsymbols@ifsaveboxundefined{BasicTree#1#2#3#4#5}{%
1767 \def\leaf@or@not@leaf{#5}%
1768 \ifx\leaf@or@not@leaf\@leaf@is@leaf%
1769 \Basic@Tree[#1]{#2}{#3}{#4}{#5}%
1770 \else%
1771 \ifx\#5\%
1772 \Basic@Tree[#1]{#2}{#3}{#4}{#5}%
1773 \else%
1774 \PackageError{tikzsymbols}{The last\MessageBreak parameter has either to be \MessageBreak
1775 'leaf' or has to be empty}{See the tikzsymbols documentation. Section 'Trees'.}%
1776 \fi\fi%
1777 }%
1778 \tikzsymbols@use@box{BasicTree#1#2#3#4#5}%
1779 \tikzsymbolsaftersymbolinput%
1780 }

```

`\WorstTree` It's not that bad.

```

1781 \tkzsymls@Declare@Robust@Command{WorstTree}[1][1]{%
1782 \tikzsymbols@ifsaveboxundefined{WorstTree#1}{%
1783 \set@tkzsymls@scl{#1}%
1784 \iftikzymbols@draftabsolute%
1785 \tkzsymls@bx@Prmtrstore{#1}%
1786 \tikzsymbols@draftbox{1.64ex*\real{\tkzsymls@Prmtr}}{1.84ex*\real{\tkzsymls@Prmtr}}%
1787 \else%
1788 \ifdim\tkzsymls@scl<0pt\set@tkzsymls@scl{-#1}\fi%
1789 \begin{tikzpicture}[x=1ex,y=1ex, line width=0.04ex*\tkzsymls@scl,scale=#1]
1790 \fill[brown] (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1791 controls (0.5,0.7) and (0.5,0.3) .. (1,0);
1792 \draw (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
1793 controls (0.5,0.7) and (0.5,0.3) .. (1,0) ;
1794 \fill[green] (0.2,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1795 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1796 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) ..
1797 controls (1.1, 1.6) and (1,1.4) .. (1,1.2) ..
1798 controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.5,0.8);
1799 \draw (0.214,0.8) -- (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) .. (-0.3,1.2) ..
1800 controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
1801 controls (0.3,1.8) and (0.6,1.6) .. (0.7,1.5) .. controls (1.1, 1.6) and (1,1.4) ..
1802 (1,1.2) .. controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.486,0.8);
1803 \fill[red] (0,1) circle (0.1);
1804 \fill[red] (0.4,1.2) circle (0.1);
1805 \fill[red] (0.8,1.1) circle (0.1);
1806 \end{tikzpicture}%
1807 \fi%
1808 }%
1809 \tikzsymbols@use@box{WorstTree#1}%
1810 \tikzsymbolsaftersymbolinput%
1811 }

```

`\Springtree` Some predefined Trees.

`\Summertree` “Hey that look like the trees in the ...” – “Yes, Yes, I know!”.

`\Autumtree` We don’t need `\tikzsymbolsaftersymbolinput` because it is already used in

`\Wintertree` `\BasicTree`.

```

1812 \tkzsymls@Declare@Robust@Commandx{Springtree}[1][1=1, usedefault]%
1813 {\tikzsymbolsuse{BasicTree}[#1]{brown!70!black}{green!90!black}{green!80!black}{leaf}}
1814 \tkzsymls@Declare@Robust@Commandx{Summertree}[1][1=1, usedefault]%
1815 {\tikzsymbolsuse{BasicTree}[#1]{brown!50!black}{green!80!black}{red!80!green}{leaf}}
1816 \tkzsymls@Declare@Robust@Commandx{Autumtree}[1][1=1, usedefault]%
1817 {\tikzsymbolsuse{BasicTree}[#1]{red!30!black}{red!75!black}{orange}{leaf}}
1818 \tkzsymls@Declare@Robust@Commandx{Wintertree}[1][1=1, usedefault]%
1819 {\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.

```

1820 \AtBeginDocument{%
1821 \ifKV@tikzsymbols@marvosym

```

```

1822 \@ifpackageloaded{marvosym}{\}%
1823 \PackageError{tikzsymbols}{Use option 'marvosym' only\MessageBreak
1824 if you load package 'marvosym'}
1825 {Either load package 'marvosym' or\MessageBreak
1826 delete the tikzsymbols option 'marvosym'}}
1827 \fi
1828 }

```

## Change History

v1.0		\@ifpackageloaded. Did something else, I can't remember . . .	1
General: Initial version . . . . .			1
v1.05		v2.5	
General: Deleted a “t” in the BasicTree-code, shortened the trunk from the tree a bit, renamed some codes, made an index . . . . .		General: New option: draftabsolute, changed the documentation a bit . . . . .	1
v1.6		v3.0	
General: on/off. . . . .		General: Changed output of “absolute” option . . . . .	1
Renamed “tikzsymbolsaftersymbolinput” to “tikzsymbolsaftersymbolinput” . . . . .		Changed symbol code . . . . .	1
Now “Person” can be used in sections, etc. . . . .		Changed the documentation . . .	1
Now an error message is generated if the last parameter of “BasicTree” is neither “leaf” nor empty. . . . .		Replaced \let by \tikzsymbols@let . . . . .	1
v1.61		Using \changes correctly (hopefully) . . . . .	1
General: Made an invisible box in BasicTree. . . . .		marvosym: using xkeaval's boolkey .	11
v1.65		\Autumntree: Replaced \BasicTree by \tikzsymbolsuse{BasicTree} . . . . .	52
General: Improved BasicTree; New symbols “Schaler/peeler”, Laughey, Walley, Ninja; but didn't improve the source-description . . . . .		\Basic@Tree: has now the default definition \Basic@Tree@on . . .	10
v1.7		\Basic@Tree@ff: Draft rectangle same size as normal tree. Changed parts of the code . . .	15
General: New symbols, etc. . . . .		\Basic@Tree@on: Same here . . . .	16
v2.0		\cmdKV@tikzsymbols@prefix: New macro. . . . .	11
General: Fixed Bugs, improved BasicTree, new option “marvosym”, new symbol . . . . .		\Cofeecup: Changed angle of arc a bit. Changed a length. . . . .	47
v2.2		\current@tikzsymbols: New macro. Current script size and text color is stored inside . . . .	13
General: Now you can use negative scaling. Include		\iftikzsymbols@draftabsolute: New \newif . . . . .	10
		final: using xkeaval's boolkey . .	10
		using xkeaval's choicekey . . . .	10
		\oldWinkey <sub>□</sub> \olddWinkey: Old Winkey . . . . .	29

<code>draft</code> : using <code>xkeaval</code> 's choicekey. Furthermore new option absolute is available . . . . .	11		
<code>draftabsolute</code> : is obsolete . . . . .	11		
<code>tree</code> : Using <code>xkeyval</code> 's choicekey. Can be set: on/true, off/false . . . .	12		
<code>usebox</code> : New option . . . . .	11		
<code>\Springtree</code> : Replaced <code>\BasicTree</code> by <code>\tikzsymbolsuse{BasicTree}</code> . . . . .	52		
<code>\Strichmaxerl</code> : Renamed <code>\Person</code> to <code>\Strichmaxerl</code> and added great parts of code . . . . .	44		
<code>\Summertree</code> : Replaced <code>\BasicTree</code> by <code>\tikzsymbolsuse{BasicTree}</code> . . . . .	52		
<code>\tikzsymbols@draftbox</code> : New macro. Draw a plain vanilla rectangle if draft is absolute . .	12		
<code>\tikzsymbols@draftboxheight</code> : New length . . . . .	12		
<code>\tikzsymbols@draftboxlength</code> : New length . . . . .	12		
<code>\tikzsymbols@draftQbox</code> : New macro. Short form of <code>\tikzsymbols@draftbox</code> . . . .	13		
<code>\tikzsymbols@ifsaveboxundefined</code> : New macro. Checks if save-box is undefined, if true: does nothing; if false: defined new box. .	13		
<code>\tikzsymbols@let</code> : New macro. <code>\let</code> with prefix . . . . .	15		
<code>\tikzsymbols@Strichmaxerl@ifSmallerZero</code> : New macro. Is needed for <code>\Strichmaxerl</code> . Checks if something is smaller than 0 . . . . .	44		
<code>\tikzsymbols@Strichmaxerl@XCheck</code> : New macro. Is needed for <code>\Strichmaxerl</code> . . . . .	43		
<code>\tikzsymbols@use@box</code> : New macro. Prints the savebox . . .	14		
<code>\tikzsymbolsuse</code> : New macro. Using symbols without carrying about the prefix . . . . .	14		
<code>\tkzsymbols@bx@Prmtrstore</code> : New macro. Makes <code>\tkzsymbolsPrmtr</code> positive (if it is negative) . . . . .	13		
<code>\tkzsymbols@Declare@Robust@Command</code> : New macro. Is needed for option prefix . . . . .	14		
<code>\tkzsymbols@Declare@Robust@Commandx</code> : New macro. Is needed for option prefix . . . . .	14		
<code>\tkzsymbols@Prmtr</code> : New macro. Needed for the plain vanilla rectangles . . . . .	13		
<code>\tkzsymbolsnewcommand</code> : New macro. Is needed for option prefix . . . . .	14		
<code>\Winkey_\dWinkey</code> : Changed the smile. Old Winkey available as <code>\oldWinkey</code> and <code>\olddWinkey</code> . . . . .	29		
<code>\Wintertree</code> : Replaced <code>\BasicTree</code> by <code>\tikzsymbolsuse{BasicTree}</code> . . . . .	52		
		v3.0b	
		<code>\Basic@Tree@on</code> : Transferred <code>\pgfarrowsdeclare</code> inside the <code>tikzpicture</code> environment. It seems this command produces an unwanted space. . . . .	16
		v3.0c	
		<code>\tkzsymbols@bx@Prmtrstore</code> : Added a @ sign. . . . .	13
		v3.0d	
		General: <code>\tkzsymbolsscl</code> to <code>\tkzsymbols@scl</code> . . . . .	1
		v3.0e	
		General: <code>\tkzsymbolsDeclareRobustCommand</code> to <code>\tkzsymbols@Declare@Robust@Command</code> . . . . .	1
		v3.0f	
		General: <code>\tkzsymbolsbxPrmtrstore</code> to <code>\tkzsymbols@bx@Prmtrstore</code> . .	1

# Index

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

	<b>Symbols</b>		<b>C</b>		152, 160, 171, 192, 213, 244, 274, 302, 349, 369, 390, 415, 437, 460, 483, 509, 529, 551, 572, 591, 611, 630, 650, 663, 670, 691, 710, 733, 753, 773, 792, 812, 832, 851, 870, 890, 909, 931, 952, 976, 999, 1024, 1048, 1073, 1098, 1125, 1151, 1181, 1205, 1237, 1262, 1290, 1313, 1352, 1379, 1392, 1429, 1455, 1458, 1461, 1503, 1530, 1551, 1574, 1581, 1607, 1629, 1652, 1679, 1682, 1688, 1722, 1770, 1773, 1787
	<code>\@BackblechlochX</code> 398, 400		<code>\c</code> ..... 199		
	<code>\@BackblechlochY</code> 399, 401		<code>\Candle</code> ..... <u>1524</u>		
	<code>\@Ofenschalter</code> 420, 421		<code>\Cat</code> ..... <u>1307</u>		
	<code>\@Tree@SetUp</code> .. <u>129</u> , 175		<code>\Chair</code> ..... <u>1601</u>		
	<code>\@gobble</code> ..... 99		<code>\cmdKV@tikzsymbols@prefix</code> ..... <u>40</u> , 102, 104, 107, 109, 112, 114, 117, 120, 123, 125, 127		
	<code>\@ifpackageloaded</code> .. ..... 4–9, 1822		<code>\Cofeecup</code> ..... <u>1574</u>		
	<code>\@leaf@is@leaf</code> . <u>56</u> , 145, 154, 188, 1768		<code>\Cooley_\dCooley</code> .. <u>993</u>		
	<code>\@secondoftwo</code> ..... 98		<code>\coordinate</code> ..... 197		
	<code>\@tkzssmbles@negfalse</code> ..... 204, 1620		<code>\csname</code> ..... 83, 87, 93, 102, 104, 107, 109, 112, 114, 120, 123, 125, 127		
	<code>\@tkzssmbles@negtrue</code> ..... 173, 1608		<code>\current@color</code> .... 77		
	<code>\@</code> ..... 1771		<code>\current@tikzsymbols</code> ..... <u>75</u> , 77, 78, 83, 87, 93		
	<b>A</b>		<code>\CurrentOption</code> .... 33		
	<code>\a</code> ..... 198				
	<code>\Annoey_\dAnnoey</code> .. <u>624</u>		<b>D</b>		
	<code>\AtBeginDocument</code> . 1820		<code>\d</code> ..... 199		
	<code>\Autumntree</code> ..... <u>1812</u>		<code>\def</code> ..... 16, 18, 24, 26, 28, 45, 47, 49, 51, 102, 107, 112, 123, 129, 143, 174, 1354, 1355, 1394, 1395, 1678, 1681, 1683, 1767		<code>\endcsname</code> ..... 78, 83, 87, 93, 102, 104, 107, 109, 112, 114, 117, 120, 123, 126, 127
	<b>B</b>		<code>\define@boolkey</code> ... ..... 14, 37, 38		<code>\Error_Message</code> ... <u>1820</u>
	<code>\b</code> ..... 198		<code>\define@choicekey</code> 21, 42		<code>\expandafter</code> ..... . 78, 81, 82, 84– 86, 91, 92, 102, 103, 107, 108, 112, 113, 123–126
	<code>\Basic@Tree</code> ..... . <u>13</u> , 16, 18, 24, 26, 28, 45, 47, 49, 51, <u>138</u> , 1769, 1772		<code>\define@cmdkey</code> ..... 41		
	<code>\Basic@Tree@ff</code> .... <u>138</u>		<code>\define@key</code> ..... 31		
	<code>\Basic@Tree@off</code> ... ..... 18, 26, 28, 49, 51, 138, 170		<code>\draft</code> ..... <u>21</u>		
	<code>\Basic@Tree@on</code> . 13, 16, 24, 45, 47, <u>168</u>		<code>\draftabsolute</code> .... <u>31</u>		
	<code>\BasicTree</code> ..... <u>1765</u>				<b>F</b>
	<code>\Bed</code> ..... <u>1623</u>				<code>\f@size</code> ..... 77
	<code>\Black@is@Black</code> 1354, 1363, 1378, 1394		<b>E</b>		<code>\fi</code> .. 19, 29, 52, 73, 88, 100, 119, 141, 142, 151, 164, 166, 173, 195, 205, 214, 232,
	<code>\Black@or@not@Black</code> ..... 1355, 1363, 1378, 1395		<code>\edef</code> ..... 73, 77, 141		
			<code>\else</code> ..... 17, 80, 97, 118, 148,		

245, 262, 275,			
290, 303, 337,			
350, 357, 370,			
378, 391, 403,			
416, 425, 438,			
448, 461, 471,			
484, 497, 510,			
517, 530, 539,			
552, 560, 573,			
579, 592, 600,			
612, 619, 631,			
639, 651, 658,			
671, 679, 684,			
692, 699, 711,			
722, 734, 742,			
754, 762, 774,			
781, 793, 801,			
813, 821, 833,			
840, 852, 859,			
871, 879, 891,			
898, 910, 920,			
932, 941, 953,			
964, 977, 988,			
1000, 1013, 1025,			
1037, 1049, 1062,			
1074, 1086, 1099,			
1114, 1126, 1140,			
1152, 1170, 1182,			
1194, 1206, 1226,			
1238, 1251, 1263,			
1279, 1291, 1302,			
1314, 1341, 1353,			
1364, 1379, 1381,			
1393, 1417, 1430,			
1446, 1457, 1463,			
1464, 1469, 1504,			
1514, 1531, 1540,			
1552, 1569, 1582,			
1595, 1600, 1608,			
1613, 1617, 1630,			
1639, 1653, 1666,			
1676, 1684, 1711,			
1723, 1760, 1776,			
1788, 1807, 1827			
<code>\filldraw</code> . . . . .	393		
<code>\final</code> . . . . .	<u>14</u>		
<code>\Fire</code> . . . . .	<u>1545</u>		
<code>\foreach</code> ..	398, 399,		
	420, 556, 557, 1753		
<code>\frame</code> . . . . .	64, 66		
		<b>G</b>	
		<code>\global</code> . . . . .	84
		<b>H</b>	
		<code>\hbox</code> . . . . .	67
		<b>I</b>	
		<code>\if@tkzssmbles@neg</code> ..	
		. . . . .	<u>11</u> , 1613
		<code>\ifcase</code> . . . . .	23, 44
		<code>\ifcsname</code> . . . . .	78, 117
		<code>\ifdim</code> . . . . .	73, 141,
			142, 173, 214,
			245, 275, 303,
			350, 370, 391,
			416, 438, 461,
			484, 510, 530,
			552, 573, 592,
			612, 631, 651,
			671, 692, 711,
			734, 754, 774,
			793, 813, 833,
			852, 871, 891,
			910, 932, 953,
			977, 1000, 1025,
			1049, 1074, 1099,
			1126, 1152, 1182,
			1206, 1238, 1263,
			1291, 1314, 1353,
			1393, 1430, 1452,
			1453, 1459, 1467,
			1504, 1531, 1552,
			1582, 1608, 1630,
			1653, 1676, 1677,
			1680, 1723, 1788
		<code>\ifKV@tikzsymbols@final</code>	
		. . . . .	15
		<code>\ifKV@tikzsymbols@marvosym</code>	
		. . . . .	663, 1574, 1821
		<code>\ifKV@tikzsymbols@usebox</code>	
		. . . . .	95
		<code>\iftikzymbols@draftabsolute</code>	
		. . . . .	<u>11</u> ,
			144, 169, 210,
			241, 271, 299,
			346, 366, 387,
			412, 434, 457,
			480, 506, 526,
			548, 569, 588,
			608, 627, 647,
			667, 688, 707,
			730, 750, 770,
			789, 809, 829,
			848, 867, 887,
			906, 928, 949,
			973, 996, 1021,
			1045, 1070, 1095,
			1122, 1148, 1178,
			1202, 1234, 1259,
			1287, 1310, 1349,
			1389, 1425, 1474,
			1527, 1548, 1578,
			1604, 1626, 1649,
			1685, 1719, 1784
		<code>\ifx</code> . . . . .	145,
			154, 188, 1363,
			1378, 1768, 1771
		<code>\Innocey</code> . . . . .	969
		<code>\Innocey\_\dInnocey</code> .	<u>946</u>
		<b>K</b>	
		<code>\KV@tikzsymbols@useboxtrue</code>	
		. . . . .	39
		<b>L</b>	
		<code>\Laughey\_\dLaughey</code> .	<u>704</u>
		<code>\leaf@or@not@leaf</code> ..	
		. . . . .	143, 145, 154,
			174, 188, 1767, 1768
		<code>\leafcolor</code> . . . . .	200
		<code>\let</code> . . . . .	124
		<b>M</b>	
		<code>\marvosym</code> . . . . .	<u>37</u>
		<code>\maxdimen</code> . . . . .	65
		<code>\MessageBreak</code> . . . . .	
		. . . . .	34, 118, 1521,
		<code>\ifKV@tikzsymbols@marvosym</code>	
		. . . . .	1774, 1823, 1825
		<code>\Moai</code> . . . . .	<u>1672</u>
		<b>N</b>	
		<code>\NeedsTeXFormat</code> . . . . .	1
		<code>\Neutrey\_\dNeutrey</code> .	<u>747</u>
		<code>\newcommand</code> . . . . .	13,
			40, 55, 56, 58,
			61, 70–72, 75, 76,
			90, 101, 106, 111,
			113, 116, 122,
			1451, 1466, 1671
		<code>\newif</code> . . . . .	11, 12





<code>\tikzsymbols@draftQbox</code>	380, 405, 427,	381, 406, 428,
..... <u>70</u> , 590,	450, 473, 499,	451, 474, 500,
610, 629, 649,	519, 541, 562,	520, 542, 563,
669, 690, 709,	581, 602, 621,	582, 603, 622,
732, 752, 772,	641, 660, 681,	642, 661, 682,
791, 811, 831,	701, 724, 744,	702, 725, 745,
850, 869, 889,	764, 783, 803,	765, 784, 804,
908, 930, 998,	823, 842, 861,	824, 843, 862,
1023, 1047, 1072	881, 900, 922,	882, 901, 923,
<code>\tikzsymbols@ifsaveboxundefined</code>	943, 966, 990,	944, 967, 991,
<u>76</u> , 98, 208, 239,	1015, 1039, 1064,	1016, 1040, 1065,
269, 297, 344,	1088, 1116, 1142,	1089, 1117, 1143,
364, 385, 410,	1172, 1196, 1228,	1173, 1197, 1229,
432, 455, 478,	1253, 1281, 1304,	1254, 1282, 1305,
504, 524, 546,	1343, 1383, 1419,	1344, 1384, 1420,
567, 586, 606,	1448, 1516, 1542,	1449, 1517, 1543,
625, 645, 665,	1571, 1597, 1619,	1572, 1598, 1621,
686, 705, 728,	1641, 1668, 1713,	1642, 1669, 1714,
748, 768, 787,	1762, 1778, 1809	1763, 1779, 1810
807, 827, 846,	<code>\tikzsymbols@x@LA</code> ..	<code>\tikzsymbols@Moaitthickness</code>
865, 885, 904,	..... 1475,	..... <u>1671</u> ,
926, 947, 971,	1483, 1488, 1490	1678, 1681, 1683,
994, 1019, 1043,	<code>\tikzsymbols@x@LB</code> ..	1686, 1687, 1690
1068, 1093, 1120,	..... 1477,	<code>\tikzsymbols@suse</code> ...
1146, 1176, 1200,	1485, 1488, 1490	.... <u>116</u> , 1813,
1232, 1257, 1285,	<code>\tikzsymbols@x@max</code> .	1815, 1817, 1819
1308, 1347, 1387,	..... 1487, 1500	<code>\tikzsymbols@draftabsolutetrue</code>
1423, 1472, 1525,	<code>\tikzsymbols@x@min</code> .	..... 28, 36
1546, 1576, 1602,	..... 1489, 1501	<code>\tikzsymbols@bx@Prmtrstore</code>
1624, 1647, 1673,	<code>\tikzsymbols@x@RA</code> ..	..... <u>71</u> , 211,
1717, 1766, 1782	..... 1476,	242, 272, 300,
<code>\tikzsymbols@let</code> ..	1484, 1488, 1490	347, 367, 388,
. <u>122</u> , 237, 267,	<code>\tikzsymbols@x@RB</code> ..	413, 435, 458,
295, 342, 362,	..... 1478,	481, 507, 527,
383, 408, 430,	1486, 1488, 1490	549, 570, 589,
453, 476, 502,	<code>\tikzsymbols@y@LA</code> ..	609, 628, 648,
522, 544, 565, 584	..... 1479,	668, 689, 708,
<code>\tikzsymbols@Strichmaxerl@ifSmall@Zero</code>	1493, 1498	731, 751, 771,
.. <u>1466</u> , 1493–1496	<code>\tikzsymbols@y@LB</code> ..	790, 810, 830,
<code>\tikzsymbols@Strichmaxerl@XCheck</code> ..	1481, 1495, 1498	849, 868, 888,
.. <u>1451</u> , 1483–1486	<code>\tikzsymbols@y@max</code> .	907, 929, 950,
<code>\tikzsymbols@Tree@absolute@scale</code> ..	1497, 1502	974, 997, 1022,
..... 140, 141	<code>\tikzsymbols@y@RA</code> ..	1046, 1071, 1096,
<code>\tikzsymbols@tree@nr</code>	..... 1480,	1123, 1149, 1179,
..... 43, 44	1492, 1494, 1498	1203, 1235, 1260,
<code>\tikzsymbols@tree@val</code>	<code>\tikzsymbols@y@RB</code> ..	1288, 1311, 1350,
..... 43	.. 1482, 1496, 1498	1390, 1426, 1499,
<code>\tikzsymbols@use@box</code>	<code>\tikzsymbols@aftersymbolinput</code>	1528, 1549, 1579,
<u>90</u> , 99, 234, 264,	.. <u>55</u> , 235, 265,	1605, 1627, 1650,
292, 339, 359,	293, 340, 360,	1675, 1720, 1785

<code>\tkzsybls@Declare@Robust@Ctkzsybls@sc1</code> .	57,	1247, 1263, 1264,
.....	<u>101</u> ,	1266, 1269, 1272,
207, 238, 268,	73, 141, 142,	1291–1293, 1295–
296, 343, 363,	153, 173, 189,	1298, 1314, 1315,
384, 409, 431,	193, 214–216,	1317, 1318, 1321,
454, 477, 503,	222, 245–248,	1323, 1325, 1326,
523, 545, 566,	250, 252, 253,	1328, 1330, 1332,
969, 1519, 1524,	255, 257, 259,	1334, 1336, 1338,
1545, 1575, 1601,	275, 277, 287,	1353, 1356, 1357,
1623, 1716, 1781	303–305, 350–	1360, 1361, 1364,
<code>\tkzsybls@Declare@Robust@Command</code>	353, 370–372,	1372, 1373, 1375,
.....	<u>106</u> , 585,	1376, 1379, 1393,
605, 624, 644,	375, 391–394,	1396–1399, 1401,
664, 685, 704,	396, 416, 417,	1402, 1406, 1407,
727, 747, 767,	423, 438–440,	1430, 1431, 1438–
786, 806, 826,	444, 445, 461–	1442, 1504, 1505,
845, 864, 884,	463, 484–486,	1531–1535, 1537,
903, 925, 946,	489–491, 493–	1552, 1553, 1562,
970, 993, 1018,	495, 510–512,	1564, 1582–1584,
1042, 1067, 1091,	530–532, 536,	1587, 1589, 1591,
1119, 1145, 1175,	537, 552, 553,	1608, 1609, 1611,
1199, 1231, 1256,	573, 574, 592–	1630, 1631, 1653,
1284, 1307, 1346,	595, 612, 613,	1655, 1676, 1677,
1386, 1422, 1471,	631–634, 651,	1680, 1689, 1690,
1644, 1672, 1812,	652, 671, 672,	1696, 1698, 1700,
1814, 1816, 1818	692, 693, 711–	1723, 1724, 1726–
<code>\tkzsybls@Prmtr</code> <u>71</u> ,	714, 717, 719,	1728, 1731, 1752,
141, 146, 147,	734–736, 739,	1754, 1788, 1789
149, 150, 212,	754–757, 774,	<code>\tkzsyblsnewcommand</code>
243, 273, 301,	775, 793, 794,	.....
348, 368, 389,	813, 814, 833,	<u>111</u> , 1765
414, 436, 459,	834, 852, 853,	<code>\Tongey</code> <code>\d</code> Tongey . <u>1042</u>
482, 508, 528,	871–874, 891,	<code>\tree</code> ..... <u>42</u>
550, 571, 590,	892, 910–913,	<code>\Tribar</code> ..... <u>1644</u>
610, 629, 649,	932–934, 953,	<code>\typeout</code> ..... 32, 35
669, 690, 709,	954, 960–962,	
732, 752, 772,	977, 978, 983,	<b>U</b>
791, 811, 831,	985, 986, 1000,	<code>\usebox</code> ..... <u>38</u> , 91
850, 869, 889,	1001, 1004, 1006,	<code>\usetikzlibrary</code> ... 10
908, 930, 951,	1025, 1026, 1031,	
975, 998, 1023,	1033, 1049, 1050,	<b>V</b>
1047, 1072, 1097,	1055, 1057, 1059,	<code>\vbadness</code> ..... 65
1124, 1150, 1180,	1074, 1075, 1079,	<code>\vbox</code> ..... 66
1204, 1236, 1261,	1081, 1083, 1099–	<code>\Vomey</code> <code>\d</code> Vomey ... <u>1145</u>
1289, 1312, 1351,	1101, 1107, 1109,	
1391, 1427, 1428,	1111, 1112, 1126,	<b>W</b>
1501, 1502, 1529,	1127, 1131, 1133,	<code>\Walley</code> <code>\d</code> Walley . <u>1199</u>
1550, 1580, 1606,	1135, 1137, 1152–	<code>\Winkey</code> <code>\d</code> Winkey .. <u>786</u>
1628, 1651, 1686,	1156, 1182–1184,	<code>\Wintertree</code> ..... <u>1812</u>
1687, 1721, 1786	1186, 1206–1210,	<code>\WorstTree</code> ..... <u>1781</u>
	1213, 1216, 1219,	
	1238–1242, 1245–	

**X**                       $\backslash$ key $\square$ \dkey . . . . . 903                      **Y**

$\backslash$ x . . . . . 556, 558, 1753, 1754                       $\backslash$ y . . . . . 557, 558

Well then, happy T<sub>E</sub>Xing!