tikzsymbols*

Ben Vitecek
b.vitecek@gmx.at

March 24, 2013

Abstract

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

Contents

*This document corresponds to tikzsymbols v2.2, dated 2013/03/23.

1 Short Introduction

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

2 Options

2.1 For Trees

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

Options to produce normal trees: □	Options for "spacefillers": □
`\usepackage{tikzsymbols}`	
`\usepackage{tree=on}{tikzsymbols}`	`\usepackage{tree=off}{tikzsymbols}`
`\usepackage{final}{tikzsymbols}`	`\usepackage{draft}{tikzsymbols}`
`\documentclass[final]{class}` `\usepackage{tikzsymbols}`	`\documentclass[draft]{class}` `\usepackage{tikzsymbols}`

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

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

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

2.2 If you load the package "marvosym"

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

Without option "marvosym" ☺ ☕	With option "marvosym" ☺ ☕
`\usepackage{marvosym}` `\usepackage{tikzsymbols}`	`\usepackage{marvosym}` `\usepackage{marvosym}{tikzsymbols}`

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

3 Symbols

In this section the symbols are introduced. They all change automaticly with the textsize.

3.1 cooking-symbols

At the following table the cooking-symbols are listed.

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

<scale> can be a number between (not exactly) -1400 and (also not exactly) 1400[1] , default is 1. The optional parameter(s) are for both, the german and the english commands the same.

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

German & English Commands		Optional parameter(s)	Output (defaultsize)
`\Kochtopf`	`\pot`	[⟨*scale*⟩]	
`\Bratpfanne`	`\fryingpan`	[⟨*scale*⟩]	
`\Schneebesen`	`\eggbeater`	[⟨*scale*⟩]	
`\Sieb`	`\sieve`	[⟨*scale*⟩]	
`\Purierstab`	`\blender`[2]	[⟨*scale*⟩]	
`\Dreizack`	`\trident`	[⟨*scale*⟩]	
`\Backblech`	`\bakingplate`	[⟨*scale*⟩]	
`\Ofen`	`\oven`	[⟨*scale*⟩]	
`\Pfanne`	`\pan`	[⟨*scale*⟩]	
`\Herd`	`\cooker`	[⟨*scale*⟩]	
`\Saftpresse`	`\squeezer`	[⟨*scale*⟩]	
`\Schussel`	`\bowl`	[⟨*scale*⟩]	
`\Schaler`	`\peeler`	[⟨*scale*⟩]	

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

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

3.2 Emoticons ☺

3.2.1 "normal" Emoticons

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

<scale> can be a number between (not exactly) -2000 and (not exactly) 2000[3], default is 1.

<color> can be every defined color.

Commands	Optional parameter(s)	Output (default)
`\Sadey`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Smiley`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Laughey`	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*mouth color*⟩`]`	
`\Annoey`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Neutrey`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Winkey`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Sey`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Xey`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Innocey`	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*halo color*⟩`]`	
`\wInnocey`	`[`⟨*scale*⟩`]`	
`\Cooley`	`[`⟨*scale*⟩`][`⟨*color*⟩`]`	
`\Tongey`	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*tongue color*⟩`]`	
`\Nursey`[4]	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*cap color*⟩`][`⟨*cross color*⟩`]`	
`\Vomey`	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*vomit color*⟩`]`	
`\Walley`	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*wall color*⟩`]`	
`\Cat`	`[`⟨*scale*⟩`]`	
`\Ninja`	`[`⟨*scale*⟩`][`⟨*color*⟩`][`⟨*headband color*⟩`][`⟨*eye color*⟩`]`	
`\NiceReapey`	`[`⟨*scale*⟩`]`	

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 paramter(s), the third shows the default-output.

[3] Do you even need so large symbols?

[4] The cross has nothing to do with religion meanings.

<scale> can be a number between a small number[5] and a large number[6], default is 1. <color> can be every defined color (see examples below).

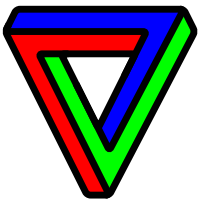
Commands	Optional parameter(s)	Output (default)
`\dSadey`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dSmiley`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dLaughey`	[⟨*scale*⟩] [⟨*color*⟩] [⟨*mouth color*⟩]	
`\dAnnoey`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dNeutrey`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dWinkey`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dSey`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dXey`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dInnocey`	[⟨*scale*⟩] [⟨*color*⟩] [⟨*halo color*⟩]	
`\dCooley`	[⟨*scale*⟩] [⟨*color*⟩]	
`\dTongey`	[⟨*scale*⟩] [⟨*color*⟩] [⟨*tongue color*⟩]	
`\dNursey`[4]	[⟨*scale*⟩] [⟨*color*⟩] [⟨*cap color*⟩] [⟨*cross color*⟩]	
`\dVomey`	[⟨*scale*⟩] [⟨*color*⟩] [⟨*vomit color*⟩]	
`\dWalley`	[⟨*scale*⟩] [⟨*color*⟩] [⟨*wall color*⟩]	
`\dNinja`	[⟨*scale*⟩] [⟨*color*⟩] [⟨*headband color*⟩] [⟨*eye color*⟩]	

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

3.3 other Symbol(s)

Commands	Optional parameter(s)	Output (default)
`\Person`	[⟨*scale*⟩] [⟨*left arm*⟩] [⟨*right arm*⟩] [⟨*left leg*⟩] [⟨*right arm*⟩]	
`\Candle`	[⟨*scale*⟩]	
`\Fire`	[⟨*scale*⟩]	
`\Coffeecup`	[⟨*scale*⟩]	
`\Chair`	[⟨*scale*⟩]	
`\Bed`	[⟨*scale*⟩]	
`\Moai`	[⟨*scale*⟩]	
`\Tribar`	[⟨*scale*⟩] [⟨*color 1*⟩] [⟨*color 2*⟩] [⟨*color 3*⟩]	
`\Snowman`	[⟨*scale*⟩]	

[5] under 500 for sure
[6] over 500 for sure



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

3.4 Trees

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

<scale> can be a number between −128 and 128[7], default is 1.

<color> can be every defined color.

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

If you are using those trees, LaTeX needs longer to produce the pdf. So you may use the package option `tree=off` or `draft` (see section 2) to make LaTeX faster.

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

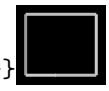
`\BasicTree` examples:

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

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

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

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



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

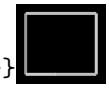
```
\colorbox{green}{\BasicTree{red}{orange}{yellow}{leaf}}
```

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

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

[7] if it is larger (or less) it uses too much of LaTeX memory and an error message appears.

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



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

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

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

3.5 Something to redefine

At the end of the code I am using the command:
`\newcommand{\tikzsymbolsaftersymbolinput}{\xspace}`

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

Well that's it.

4 Warnings and Errors

4.1 Warnings

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

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

You can avoid those messages by putting the symbol into this command:
`\texorpdfstring{\Smiley}{Smiley}`
For example you may use something like that:
`\subsubsection{``3D'' Emoticons \texorpdfstring{\dSmiley}{dSmiley}}`
or
`\subsection{Emoticons \texorpdfstring{\Smiley}{Smiley}}`
or
`\subsubsection{``normal'' Emoticons \texorpdfstring{\Cat}{Cat}}`

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

4.2 and errors

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

If you load “marvosym” *after* “tikzsymbols”, LaTeX generates an errormessage because “Smiley” has already been defined.

If you load “marvosym” *before* “tikzsymbols”, “tikzsymbols” will overwrite marvosym’s Smiley and no errormessage is generated (if you like the “Simley” from marvosym more, use the tikzsymbols option `marvosym`).

5 Nobody is perfect

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

6 Code (do you really need this section?)

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

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

```
\NeedsTeXFormat{LaTeX2e}[2011/06/27]
\ProvidesPackage{tikzsymbols}
[2013/03/23 v2.2 Some symbols  created using tikz.]
\@ifpackageloaded{tikz}{}{\RequirePackage{tikz}}
\@ifpackageloaded{xargs}{}{\RequirePackage{xargs}}
\@ifpackageloaded{xcolor}{}{\RequirePackage{xcolor}}
\@ifpackageloaded{xspace}{}{\RequirePackage{xspace}}
\@ifpackageloaded{xkeyval}{}{\RequirePackage{xkeyval}}
```

Furthermore we need to load some libraries from tikz:

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

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

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

`\@leaf@is@leaf` We need this command for creating an errormessage if the last paramter of BasicTree is neither “leaf” nor empty.

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

`\tkzsymblsscl` .

```
\newlength{\tkzsymblsscl}
\def\set@tkzsymblsscl#1{\setlength{\tkzsymblsscl}{#1pt}}
```

`\if@draft` `\if@final` `\if@tkzssmbls@neg` `\if@tikzsymbols@marvosym` We need draft and final for some package warnings. “marvosym” for the option “marvosym” and “neg” (negative) if something is negativ.

```
\newif\if@tikzsymbols@draft
\newif\if@tikzsymbols@final
\newif\if@tkzssmbls@neg
\newif\if@tikzsymbols@marvosym
```

`marvosym` I define the option "marvosym": you should use it, if you load the package "marvosym"

```
\DeclareOptionX{marvosym}{\@tikzsymbols@marvosymtrue}
```

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

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

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

```
\DeclareRobustCommandx{\Basic@Tree}[5][1=1, usedefault]{{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt \@tkzssmbls@negtrue\fi%
\def\leaf@or@not@leaf{#5}%
\@Tree@SetUp%
\pgfarrowsdeclare{leaf}{leaf}%
{\pgfarrowsleftextend{-.1ex} \pgfarrowsrightextend{-0.05ex}}%
{%
\pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
\pgfpatharc{150}{30}{0.08ex}% dicke
\pgfpatharc{-30}{-150}{0.08ex}%
\pgfusepathqfill%
}%
\colorlet{trunk}{#2}%
\colorlet{leaf a}{#3}%
\colorlet{leaf b}{#4}%
\begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
\ifx\leaf@or@not@leaf\@leaf@is@leaf%
   \if@tkzssmbls@neg%
   \draw[opacity=0,scale=#1+0.1*#1] (-0.82+0.1*#1/100,0) rectangle
   (0.82-0.1*#1/100, 1.4-0.01*#1/100);
   \else
   \draw[opacity=0,scale=#1+0.1*#1] (-0.82-0.1*#1/100,0) rectangle
   (0.82+0.1*#1/100, 1.4+0.01*#1/100);
   \fi%
\else
 \draw[opacity=0,scale=#1+0.1*#1] (-0.75,-0.01) rectangle (0.75,1.3);
\fi
\pgflowlevel{\pgftransformscale{#1+0.02ex}}{%
\coordinate (root) [grow cyclic,rotate=90] child {
child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
child foreach \c in {0,1,2} { child foreach \d in {0,1} {
```

```
child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5] }
}}} } edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
};}%
\end{tikzpicture}%
}}
```

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

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

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

```
\DeclareOptionX{final}{\@tikzsymbols@draftfalse\@tikzsymbols@finaltrue
\def\Basic@Tree{\Basic@Tree@on}}
```

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

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

```
\DeclareOptionX{tree}{%
  \def\Basic@Tree{\csname Basic@Tree@#1\endcsname}%
  \if@tikzsymbols@draft%
\PackageWarningNoLine{tikzsymbols}{You can use  class option \MessageBreak
draft with package option tree=on/off;
\MessageBreak but I think it would be better if you
\MessageBreak delete tree=on/off}\fi%
\if@tikzsymbols@final%
\PackageWarningNoLine{tikzsymbols}{You can use  class option \MessageBreak
final with package option tree=on/off;
\MessageBreak but I think it would be better if you
\MessageBreak delete tree=on/off}\fi%
  }
```

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

```
\DeclareRobustCommandx{\Basic@Tree@off}[5][1=1, usedefault]{{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt \set@tkzsymblsscl{-#1}\fi%
\def\leaf@or@not@leaf{#5}%
\def\Tree@Off@line{0.095*#1/100}%
\begin{tikzpicture}[scale=#1+0.01*#1,x=1.35ex,y=1.3ex, line width=0.07ex*\tkzsymblsscl]
\ifx\leaf@or@not@leaf\@leaf@is@leaf%
\draw[#2] (0-\Tree@Off@line,0) -- (0-\Tree@Off@line,1.08+0.05*#1/100);
\draw[#3] (0-\Tree@Off@line,1.08+0.05*#1/100) -- (1.2+\Tree@Off@line,1.08+0.05*#1/100);
\draw[#4] (1.2+\Tree@Off@line,1.08+0.05*#1/100) -- (1.2+\Tree@Off@line,0);
\draw[#3] (1.2+\Tree@Off@line,0) -- (0.5,0);
```

```
\draw[#4] (0.5+0.4*#1/100,0) -- (0-\Tree@Off@line,0);
\else
\draw[#2] (0,0) -- (0,1);
\draw[#3] (0,1) -- (1.15,1);
\draw[#4] (1.15,1) -- +(0,-1);
\fi%
\end{tikzpicture}%
}}
```

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

```
\DeclareRobustCommandx{\Basic@Tree@on}[5][1=1, usedefault]{{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt \@tkzssmbls@negtrue\fi%
\def\leaf@or@not@leaf{#5}%
\@Tree@SetUp%
\pgfarrowsdeclare{leaf}{leaf}%
{\pgfarrowsleftextend{-.1ex} \pgfarrowsrightextend{-0.05ex}}%
{%
\pgfpathmoveto{\pgfpoint{-.01ex}{0ex}}%
\pgfpatharc{150}{30}{0.08ex}% dicke
\pgfpatharc{-30}{-150}{0.08ex}%
\pgfusepathqfill%
}%
\colorlet{trunk}{#2}%
\colorlet{leaf a}{#3}%
\colorlet{leaf b}{#4}%
\begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex]%
\ifx\leaf@or@not@leaf\@leaf@is@leaf%
   \if@tkzssmbls@neg%
   \draw[opacity=0,scale=#1+0.1*#1] (-0.82+0.1*#1/100,0) rectangle
   (0.82-0.1*#1/100, 1.4-0.01*#1/100);
   \else
   \draw[opacity=0,scale=#1+0.1*#1] (-0.82-0.1*#1/100,0) rectangle
   (0.82+0.1*#1/100, 1.4+0.01*#1/100);
   \fi%
\else
 \draw[opacity=0,scale=#1+0.1*#1] (-0.75,-0.01) rectangle (0.75,1.3);
\fi
\pgflowlevel{\pgftransformscale{#1+0.02ex}}{%
\coordinate (root) [grow cyclic,rotate=90] child {
child [line cap=round] foreach \a in {0,1, 2} { child foreach \b in {0,1} {
child foreach \c in {0,1,2} { child foreach \d in {0,1} {
child foreach \leafcolor in {leaf a,leaf b} { edge from parent [color=\leafcolor,-#5] }
}}} } edge from parent [shorten >=-0.05ex,serif cm-,line cap=butt]
};}%
\end{tikzpicture}%
}}%
```

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

```
\ProcessOptionsX*\relax
```

6.1 Cookingsymbolcode

`\Kochtopf = \pot`

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

```
\DeclareRobustCommand{\Kochtopf}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2ex,y=2.2ex, line width=0.07ex*\tkzsymblsscl,scale=#1]
\draw[rounded corners=0.2ex*\tkzsymblsscl] (0,0.5) -- (0,0) -- (1,0) -- (1,0.5);
\draw(0,0.4) arc (90:270:0.1);
\draw(1,0.4) arc (90:-90:0.1);
\draw (0,0.5) -- (1,0.5) .. controls (1,0.6) and (0,0.6) .. (0,0.5);
\draw (0.6,0.585) arc (0:180:0.1);
\draw[decorate,
decoration={snake,amplitude=.12ex*\tkzsymblsscl,segment length=0.93ex*\tkzsymblsscl}]
(0,0.35) -- (1,0.35);
\draw (0.1,0.25) circle (0.04);
\draw (0.3,0.2) circle (0.04);
\draw (0.13, 0.125) circle (0.04);
\draw (0.6,0.25) circle (0.04);
\draw (0.45,0.1) circle (0.04);
\draw (0.88,0.2) circle (0.04);
\draw (0.7,0.11) circle (0.04);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\pot\Kochtopf
```

`\Bratpfanne = \fryingpan`

You may wonder why I am writing something like: `amplitude=.12ex*#1`. Well it's hard to explain in english, but I try my best: After being scaled the symbols would not look so good without `*#1`. The lines would be too thin, the corners not rounded enough, etc. To prevent too thin lines due to scaling I am multiplying the line width and the corners etc. so that they look the same, no matter how you scale it.

```
\DeclareRobustCommand{\Bratpfanne}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.7ex,y=1.4ex, line width=0.07ex*\tkzsymblsscl, scale=#1,
decoration={snake,amplitude=.05ex*\tkzsymblsscl,segment length=0.408ex*\tkzsymblsscl}]
\draw[rounded corners=0.07ex*\tkzsymblsscl]
(-1,0) -- (1,0) -- (1.5,0.4) -- (-1.5,0.4) -- cycle;
\draw[ line width=0.037ex*\tkzsymblsscl, rounded corners=0.023ex*\tkzsymblsscl]
(-1.4,0.3) -- (-3.5,0.3) -- (-3.5,0.25) -- (-1.3,0.25);
\draw[line width=0.023ex*\tkzsymblsscl] (-1.1,0.1) -- (1.1,0.1);
\draw[line width=0.035ex*\tkzsymblsscl, decorate]
(-0.3,0.5) -- (-0.3,1);
\draw[line width=0.035ex*\tkzsymblsscl, decorate]
   (0.3,0.5) -- (0.3,1);
\draw[line width=0.035ex*\tkzsymblsscl, decorate]
(-1,0.5) -- (-1,1);
```

```
\draw[line width=0.035ex*\tkzsymblsscl, decorate]
   (1,0.5) -- (1,1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\fryingpan\Bratpfanne
```

`\Schneebesen = \eggbeater` The next one:

```
\DeclareRobustCommand{\Schneebesen}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[y=2.1ex,x=1.4ex, scale=#1]
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (0.2,0.0) and  (0.2,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (-0.2,0.0) and  (-0.2,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (-0.1,0.0) and  (-0.1,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (0.1,0.0) and  (0.1,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (-0.15,0.0) and  (-0.15,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (0.15,0.0) and  (0.15,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (-0.05,0.0) and  (-0.05,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) .. controls (0.05,0.0) and  (0.05,0.2) ..  (0,0.4);
\draw[line width=0.01ex*(\tkzsymblsscl-\tkzsymblsscl/100*3)]
   (0,0) --(0,0.4);
\fill[line width=0.05ex*\tkzsymblsscl, rounded corners=0.07ex*\tkzsymblsscl]
   (-0.05,0.37) -- (0.05,0.37) -- (0.05,0.75) -- (-0.05,0.75) -- cycle;
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\eggbeater\Schneebesen
```

`\Sieb = \sieve` Now a long one;

```
\DeclareRobustCommand{\Sieb}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.8ex, y=2.8ex,line width=0.02ex*\tkzsymblsscl ,scale=#1]
\draw[line width=0.09ex*\tkzsymblsscl] (-0.2,0) -- (1.01,0);
\draw (0.2,0) arc (180:360:0.4);
\draw(0.25,0) arc (180:360:0.35);
\draw (0.3,0) arc (180:360:0.3);
\draw (0.35,0) arc (180:360:0.25);
\draw (0.4,0) arc (180:360:0.2);
\draw (0.45,0) arc (180:360:0.15);
\draw (0.5,0) arc (180:360:0.1);
\draw (0.55,0) arc (180:360:0.05);
\draw (.95,0) -- (0.95,-0.194);
```

```
\draw (.9,0) -- (0.9,-0.265);
\draw (.85,0) -- (0.85,-0.313);
\draw (.8,0) -- (0.8,-0.345);
\draw (.75,0) -- (0.75,-0.37);
\draw (.7,0) -- (0.7,-0.39);
\draw (.65,0) -- (0.65,-0.4);
\draw (.6,0) -- (0.6,-0.4);
\draw (.55,0) -- (0.55,-0.4);
\draw (.5,0) -- (0.5,-0.39);
\draw (.45,0) -- (0.45,-0.37);
\draw (.4,0) -- (0.4,-0.348);
\draw (.35,0) -- (0.35,-0.314);
\draw (.3,0) -- (0.3,-0.265);
\draw (.25,0) -- (0.25,-0.194);
\draw (0.2,-0.05) -- (1,-0.05);
\draw (0.21,-0.1) -- (0.99,-0.1);
\draw (0.23,-0.15) -- (0.97,-0.15);
\draw (0.255,-0.2) -- (0.945,-0.2);
\draw (0.289,-0.25) -- (0.911,-0.25);
\draw (0.335,-0.3) -- (0.865,-0.3);
\draw (0.406,-0.35) -- (0.794,-0.35);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\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":

```
\DeclareRobustCommand{\Purierstab}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.07ex*\tkzsymblsscl,scale=#1]
\draw[rounded corners=0.07ex*\tkzsymblsscl] (0,0) -- (0.3,0) --  (0.15,0.1) --cycle;
\fill[rounded corners=0.07ex*\tkzsymblsscl] (0.15,0.3) -- (0.24,0.4) -- (0.24,0.7) --
   (0.06,0.7) -- (0.06,0.4) -- cycle;
\draw (0.15,0.4) -- (0.15,0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\blender\Purierstab
```

`\Dreizack = \trident` Important for cooking:

```
\DeclareRobustCommand{\Dreizack}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.3ex, y=2.2ex, line width=0.035ex*\tkzsymblsscl,scale=#1]
\fill[rounded corners=0.07ex*(\tkzsymblsscl-\tkzsymblsscl/100)]
(0,0) -- (0,0.4)  -- (0.1,0.4) -- (0.1,0.0) -- cycle;
\draw (0.05,0) -- (0.05,0.7);
\draw[rounded corners=0.07ex*(\tkzsymblsscl-\tkzsymblsscl/100*\tkzsymblsscl*2]
```

```
(0,0.7) -- (0,0.55)  -- (0.05,0.55) -- (0.1,0.55) --  (0.1,0.7);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\trident\Dreizack
```

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

```
\DeclareRobustCommand{\Backblech}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=6.53ex,y=5.ex, line width=0.07ex*\tkzsymblsscl,scale=#1]
\filldraw[rounded corners=0.09ex*\tkzsymblsscl] (0,0) rectangle (0.3,0.3);
\draw[rounded corners=0.07ex*\tkzsymblsscl, line width=0.03ex*\tkzsymblsscl]
   (0.1,0) -- (-0.025,0) -- (-0.025,0.3) -- (0.1,0.3);
\draw[rounded corners=0.07ex*\tkzsymblsscl, line width=0.03ex*\tkzsymblsscl]
   (0.2,0) -- (.325,0) -- (.325,0.3) -- (0.2,0.3);
\foreach \@BackblechlochX in {0.007,0.293}
\foreach \@BackblechlochY in {0.007,0.293}
\fill[white] (\@BackblechlochX,
\@BackblechlochY) circle (0.02ex);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\bakingplate\Backblech
```

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

```
\DeclareRobustCommand{\Ofen}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.50ex,y=.5ex, line width=0.07ex*\tkzsymblsscl,scale=#1]
\draw (0,0) rectangle (4,3);
\draw (0.25,0.25) rectangle (3.75,2);
\foreach \@Ofenschalter in {0.5,1.1,2.9,3.5}
\fill (\@Ofenschalter,2.5) circle (0.22);
\draw (1.5,2.28) rectangle  (2.5,2.72);
\draw[line width=0.05ex*\tkzsymblsscl] (1,1.75) -- (3,1.75);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\oven\Ofen
```

`\Pfanne = \pan` I can't think of a better word ...:

```
\DeclareRobustCommand{\Pfanne}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.3ex,y=2.3ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\draw [rounded corners=0.023ex*\tkzsymblsscl]
   (0,0) -- (0.9,0) -- (1,0.3) -- (-0.1,0.3) -- cycle;
\draw (-0.2,0.22) -- (-0.08,0.22);
\draw (0.97,0.22) -- (1.08,0.22);
\draw[decorate,decoration={snake,amplitude=.046ex*\tkzsymblsscl,
   segment length=0.82ex*\tkzsymblsscl},line width=0.05ex*\tkzsymblsscl]
```

```
   (-0.05,0.1) -- (0.95,0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\pan\Pfanne
```

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

```
\DeclareRobustCommand{\Herd}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1ex,y=1ex,line width=0.04ex*\tkzsymblsscl,scale=#1]
\draw[line width=0.08ex*\tkzsymblsscl] (0,0) rectangle (2,1.5);
\draw (0.5,0.45) circle (0.35);
\draw (0.5,0.45) circle (0.2);
\draw (1.45,0.45) circle (0.3);
\draw (0.5,1.15) circle (0.21);
\draw (1.05,0.95) rectangle (1.85,1.35);
\draw (1.45,1.15) circle (0.15);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\cooker\Herd
```

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

```
\DeclareRobustCommand{\Saftpresse}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1.2ex,y=1ex,line width=0.07ex*\tkzsymblsscl,scale=#1]
\draw[rounded corners=0.1ex*\tkzsymblsscl]
   (0,0.85) -- (0,0) -- (1.5,0) -- (1.5,0.85) -- cycle;
\draw (0,0.7) -- (1.5,0.7);
\draw[rounded corners=0.1ex*\tkzsymblsscl] (0.3,0.7) -- (0.75,1.55) -- (1.2,0.7);
\draw[rounded corners=0.1ex*\tkzsymblsscl] (0.45,0.7) -- (0.75,1.55) -- (1.05,0.7);
\draw[rounded corners=0.1ex*\tkzsymblsscl]
   (0.65,0.7) -- (0.75,1.55) -- (0.85,0.7);
\draw[line width=0.05ex*\tkzsymblsscl, decorate,
   decoration={snake,amplitude=.05ex*\tkzsymblsscl,
   segment length=0.48ex*\tkzsymblsscl}]  (0,0.3) -- (1.5,0.3);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\squeezer\Saftpresse
```

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

```
\DeclareRobustCommand{\Schussel}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1ex,y=1ex,line width=0.07ex*\tkzsymblsscl, scale=#1]
\draw[rounded corners=0.5ex*\tkzsymblsscl]
   (-0.02,1.4) -- (0,1.4) -- (0,0.05) -- (1.5,0.05) -- (1.5,1.4) -- (1.52,1.4);
\draw (0.35,0) -- (1.15,0);
\end{tikzpicture}
```

```
\tikzsymbolsaftersymbolinput%
}
\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:

```
\DeclareRobustCommand{\Schaler}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.7ex,y=2.3ex, line width=0.07ex*\tkzsymblsscl,scale=#1]
\draw[rounded corners=0.07ex*\tkzsymblsscl]
   (0,0.4) -- (0,0.1) arc (0:180:-0.1) -- (0.2,0.4)
   -- (0.3,0.5) -- (0.3,0.65) -- (0.2,0.65) -- (0.2,0.5) -- (0,0.5) -- (0,0.65) --
   (-0.1,0.65) -- (-0.1,0.5)  -- cycle;
\draw[line width=0.03ex*\tkzsymblsscl] (0,0.6) -- (0.2,0.6);
\draw[line width=0.03ex*\tkzsymblsscl] (0,0.58) -- (0.2,0.58);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\let\peeler\Schaler
```

6.2 Emoticonscode

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

```
\DeclareRobustCommandx{\Sadey}[2][1=1, 2={opacity=0}, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2, line width=0.1ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw[line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\fill (0.1,0.1) circle (0.05);
\fill (-0.1,0.1) circle (0.05);
\draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dSadey}[2][1=1,2=yellow,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\draw[black] (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Annoey \dAnnoey` An annoyed Smiley -_-

```
\DeclareRobustCommandx{\Annoey}[2][1=1,2={opacity=0},usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2, line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw[line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw (0.08,0.1) -- (0.22,0.1);
\draw (-0.08,0.1) -- (-0.22,0.1);
\draw (-0.2,-0.1) -- (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dAnnoey}[2][1=1,2=yellow,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\draw[black] (0.08,0.1) -- (0.22,0.1);
\draw[black] (-0.08,0.1) -- (-0.22,0.1);
\draw[black] (-0.2,-0.1) -- (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Smiley \dSmiley` A normal Smiley

```
\if@tikzsymbols@marvosym\relax\else%
 \DeclareRobustCommandx{\Smiley}[2][1=1,2={opacity=0} ,usedefault]{%
 \set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\fill[#2] (0,0) circle (0.33);
\draw (0,0) circle (0.33);
\fill (-0.1,0.1) circle (0.05);
\fill (0.1,0.1) circle (0.05);
\draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}%
\fi
\DeclareRobustCommandx{\dSmiley}[3][1=1,2=yellow,3=yellow,usedefault]{%
 \set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
 \begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.1ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Laughey \dLaughey` A laughing Smiley

```
\DeclareRobustCommandx{\Laughey}[3][1=1,2={opacity=0},3={opacity=0} ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2,line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw[line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
\draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
\fill[#3,rounded corners=0.1ex*\tkzsymblsscl, yshift=-0.5]
(-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
\draw[rounded corners=0.1ex*\tkzsymblsscl, yshift=-0.5]
(-0.22,-0.0) .. controls (-0.13,-0.23) and (0.13,-0.23) .. (0.22,-0.0) -- cycle;
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dLaughey}[3][1=1,2=yellow, 3=red ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[ball color=#2,line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw (-0.09,0.06) .. controls (-0.11,0.16) and (-0.17,0.16) .. +(-0.1,0);
\draw (0.09,0.06) .. controls (0.11,0.16) and (0.17,0.16) .. +(0.1,0);
\shade[ball color=#3, rounded corners=0.1ex*\tkzsymblsscl, yshift=-0.3]
 (-0.25,-0.0) .. controls (-0.13,-0.26) and (0.13,-0.26) .. (0.25,-0.0) -- cycle;
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Neutrey \dNeutrey` neutral Smiley :|

```
\DeclareRobustCommandx{\Neutrey}[2][1=1, 2={opacity=0}, usedefault]{%
 \set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2,line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw[line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\fill (0.1,0.1) circle (0.05);
\fill (-0.1,0.1) circle (0.05);
\draw (-0.2,-0.1) -- (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dNeutrey}[2][1=1,2=yellow,usedefault]{%
 \set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\draw[black] (-0.2,-0.1) -- (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Winkey \dWinkey` ;)

```
\DeclareRobustCommandx{\Winkey}[2][1=1,2={opacity=0} ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\fill[#2] (0,0) circle (0.33);
\draw(0,0) circle (0.33);
\draw(0.17,0.1) -- (0.05,0.1);
\fill (-0.1,0.1) circle (0.05);
\draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dWinkey}[2][1=1,2=yellow,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\draw(0.17,0.1) -- (0.05,0.1);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.15,-0.2) .. (0.2,0);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

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

```
\DeclareRobustCommandx{\Sey}[2][1=1,2={opacity=0} ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2, line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw[line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\fill (0.1,0.1) circle (0.05);
\fill (-0.1,0.1) circle (0.05);
\draw (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dSey}[2][1=1,2=yellow ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\draw[black] (-0.2,-0.08) .. controls (-0.0,-0.2) and (0.0,0) .. (0.2,-0.12);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

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

```
\DeclareRobustCommandx{\Xey}[2][1=1, 2={opacity=0}, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2, line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
```

```
\draw[line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw (0.05,0.05) -- ++ (0.1,0.1);
\draw (0.15,0.05) -- ++ (-0.1,0.1);
\draw (-0.05,0.05) -- ++ (-0.1,0.1);
\draw (-0.15,0.05) -- ++ (0.1,0.1);
\draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dXey}[2][1=1, 2={yellow}, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[ball color=#2, line width=0.12ex*\tkzsymblsscl] (0,0) circle (0.33);
\draw (0.05,0.05) -- ++ (0.1,0.1);
\draw (0.15,0.05) -- ++ (-0.1,0.1);
\draw (-0.05,0.05) -- ++ (-0.1,0.1);
\draw (-0.15,0.05) -- ++ (0.1,0.1);
\draw (-0.2,-0.15) .. controls (-0.1,-0.06) and (0.1,-0.06) .. (0.2,-0.15);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Innocey` `\dInnocey` An innocent Smiley

```
\DeclareRobustCommandx{\Innocey}[3][1=1,2={opacity=0},3=yellow ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\fill[#2] (0,0) circle (0.33);
\draw (0,0) circle (0.33);
\fill (-0.1,0.1) circle (0.05);
\fill (0.1,0.1) circle (0.05);
\draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\draw[#3, line width=0.095ex*\tkzsymblsscl] (0.32,0.31) arc (0:360:0.32 and 0.1);
\draw[line width=0.05ex*\tkzsymblsscl] (0.3,0.31) arc (0:360:0.3 and 0.07);
\draw[line width=0.05ex*\tkzsymblsscl] (0.35,0.31) arc (0:360:0.35 and 0.12);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommand{\wInnocey}[1][1]{\Innocey[#1][opacity=0][white]}
\DeclareRobustCommandx{\dInnocey}[3][1=1,2=yellow,3=yellow,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\draw[color=#3!97!black, line width=0.09ex*\tkzsymblsscl]
   (0.32,0.31) arc (0:360:0.32 and 0.1);
\draw[line width=0.05ex*\tkzsymblsscl] (0.3,0.31) arc (0:360:0.3 and 0.07);
\draw[line width=0.05ex*\tkzsymblsscl] (0.35,0.31) arc (0:360:0.35 and 0.12);
\end{tikzpicture}%
```

```
\tikzsymbolsaftersymbolinput%
}
```

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

```
\DeclareRobustCommandx{\Cooley}[2][1=1,2={opacity=0} ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\fill[#2] (0,0) circle (0.33);
\draw (0,0) circle (0.33);
\fill[rounded corners=0.1ex*\tkzsymblsscl]
   (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
\fill[rounded corners=0.1ex*\tkzsymblsscl]
   (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
\draw (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\draw (0.02,0.1) -- (-0.02,0.1);
\draw (-0.2,0.1) -- (-0.3,0.13);
\draw (0.2,0.1) -- (0.3,0.13);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dCooley}[2][1=1,2=yellow,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\draw[black] (0.02,0.1) -- (-0.02,0.1);
\draw[black] (-0.2,0.1) -- (-0.295,0.146);
\draw[black] (0.2,0.1) -- (0.295,0.146);
\shade[ball color=black,rounded corners=0.1ex*\tkzsymblsscl]
   (0.24,0.15) -- (0.01,0.15) -- (0.01,0) -- (0.24,0) -- cycle;
\shade[ball color=black,rounded corners=0.1ex*\tkzsymblsscl]
   (-0.24,0.15) -- (-0.01,0.15) -- (-0.01,0) -- (-0.24,0) -- cycle;
\draw[black] (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Tongey` `\dTongey` :P

```
\DeclareRobustCommandx{\Tongey}[3][1=1,2={opacity=0},3={opacity=0} ,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\fill[#2] (0,0) circle (0.33);
\draw (0,0) circle (0.33);
\fill (-0.1,0.1) circle (0.05);
\fill (0.1,0.1) circle (0.05);
\fill[#3,line width=0.058ex*\tkzsymblsscl, rounded corners=0.12ex*\tkzsymblsscl]
   (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
\draw[line width=0.07ex*\tkzsymblsscl, yshift=0.21ex]
   (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\draw[line width=0.058ex*\tkzsymblsscl, rounded corners=0.12ex*\tkzsymblsscl]
   (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
```

```
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dTongey}[3][1=1,2=yellow,3=red,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.33);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\shade[ball color=#3,line width=0.058ex*\tkzsymblsscl, rounded corners=0.12ex*\tkzsymblsscl]
   (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
\draw[black, line width=0.058ex*\tkzsymblsscl, rounded corners=0.12ex*\tkzsymblsscl]
   (0,-0.09) -- (0.05,-0.2) -- (0.16,-0.23) -- (0.2,-0.15) -- (0.19,-0.03);
\draw[black, line width=0.07ex*\tkzsymblsscl, yshift=0.21ex]
   (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

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

```
\DeclareRobustCommandx{\Nursey}[4][1=1,2={opacity=0},3={opacity=0},4=black,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\fill[#3,rounded corners=.023ex*\tkzsymblsscl]
   (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
\fill[#2] (0,0) circle (0.3);
\draw (0,0) circle (0.3);
\fill (-0.1,0.1) circle (0.05);
\fill (0.1,0.1) circle (0.05);
\draw[line width=0.09ex*\tkzsymblsscl, yshift=0.07ex]
   (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\draw[rounded corners=.023ex*\tkzsymblsscl]
   (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0);
\draw[#4,line width=.046ex*\tkzsymblsscl] (0,0.35) -- (0,0.5);
\draw[#4,line width=.046ex*\tkzsymblsscl] (-0.05,0.45) -- (0.05,0.45);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dNursey}[4][1=1,2=yellow,3=white,4=red,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.3ex, y=2.3ex, line width=0.12ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2] (0,0) circle (0.3);
\shade[ball color=black] (-0.1,0.1) circle (0.05);
\shade[ball color=black] (0.1,0.1) circle (0.05);
\draw[black, line width=0.09ex*\tkzsymblsscl, yshift=0.07ex]
   (-0.2,-0.1) .. controls (-0.1,-0.2) and (0.1,-0.2) .. (0.2,-0.1);
\shade[ball color=#3, rounded corners=.023ex*\tkzsymblsscl,yshift=-0.09ex]
   (-0.3,0) -- (-0.3,0.3) -- (0,0.6) -- (0.3,0.3) -- (0.3,0) arc (0:180:0.3);
\shade[ball color=#4,line width=.046ex*\tkzsymblsscl]
   (-0.01,0.31) -- (-0.01,0.46) -- (0.01,0.46) -- (0.01,0.31)--cycle;
```

```
\shade[ball color=#4,line width=.046ex*\tkzsymblsscl]
   (-0.05,0.4) -- (0.05,0.4) -- (0.05,0.42)--(-0.05,0.42) -- cycle;
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Vomey \dVomey` *Bläärgh*

```
\DeclareRobustCommandx{\Vomey}[3][1=1,2={opacity=0},3={opacity=0},usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\fill[#2,rounded corners=0.05ex*\tkzsymblsscl]  (0,0)  arc (15:330:1) -- (-0.6,-0.3) -- cycle;
\draw[rounded corners=0.05ex*\tkzsymblsscl]  (0,0)  arc (15:330:1) -- (-0.6,-0.3) -- cycle;
\draw[line width=0.05ex*\tkzsymblsscl] (-0.5,0.3) -- (-0.3,0.1);
\fill (-0.45,0.27) arc (100:350:0.1);
\fill[#3] (1.8,-0.5)  .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
   controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
   controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
\fill[#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) ..
   controls (1.7,-1.2) and (1.3,-1.2) .. (1,-1) ..
   controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
\draw (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1);
\draw (0,-0.4) .. controls (0.5,-0.5) and (0.8,-0.7) .. (1,-1);
\draw (1.8,-0.5)  .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
   controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2)
   and (1,-2) .. (1,-1.5) .. controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dVomey}[3][1=1,2=yellow,3={brown!10!olive},usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.58ex,y=0.58ex, line width=0.09ex*\tkzsymblsscl,scale=#1]
\shade[ball color=#2!90!brown,rounded corners=0.03ex*\tkzsymblsscl]
   (0,0)  arc (15:330:1) -- (-0.6,-0.3) -- cycle;
\draw[black, line width=0.05ex*\tkzsymblsscl] (-0.5,0.3) -- (-0.3,0.1);
\shade[ball color=black] (-0.45,0.27) arc (100:350:0.1);
\shade[ball color=#3] (1.8,-0.5)  .. controls (2.5,-0.3) and (2.8,-0.7) .. (2.5,-1) ..
 controls (3,-1) and (3,-1.7) .. (2,-1.5) .. controls (1.7,-2) and (1,-2) .. (1,-1.5) ..
 controls (0.5,-1.9) and (0.3,-1) .. (0.7,-0.9);
\shade[ball color=#3] (0,-0.4) .. controls (1,0) and (2,-1) .. (2,-1) .. controls
   (1.7,-1.2) and (1.3,-1.2) .. (1,-1) .. controls (0.8,-0.7) and (0.5,-0.5) .. (0,-0.4);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\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.

```
\DeclareRobustCommandx{\Walley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1,
decoration={random steps,segment length=0.15ex*\tkzsymblsscl, amplitude=0.1ex*\tkzsymblsscl}]
```

```
\fill[#2, line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.28);
\draw[line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.28);
\fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
\draw (0.28,-0.33) rectangle (0.66,0.33);
\draw[line width=0.06ex*\tkzsymblsscl]
(0.28,0) --++(0.05,0.07) --++(0.03,0.02) --+
+(0.03,-0.02) --++(0.03,0.1) --++(0.03,0.02) -- (0.5,0.25);
\draw[line width=0.06ex*\tkzsymblsscl]
(0.28,0) --++(0.06,-0.02) --++(0.04,0.06) --+
+(0.0,-0.08) --++(0.08,0.06) --++(0.03,-0.02) --+(0.08,0.02) -- (0.6,0.0);
\draw[line width=0.06ex*\tkzsymblsscl]
(0.28,0) --++(0.03,-0.02)  --++(0.03,-0.07)  --+
+(0.03,-0.01)  --++(0.01,-0.07)  --++(0.06,0.01)  --++(0.03,-0.08)   --
(0.5,0.-0.25);
\draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
\draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\rWalley}[3][1=1, 2={opacity=0},3={opacity=0}, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1,
decoration={random steps,segment length=0.15ex*\tkzsymblsscl, amplitude=0.1ex*\tkzsymblsscl}]
\fill[#2, line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.28);
\draw[line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.28);
\fill[#3] (0.28,-0.33) rectangle (0.66,0.33);
\draw (0.28,-0.33) rectangle (0.66,0.33);
\draw[decorate, line width=0.06ex*\tkzsymblsscl] (0.28,0) -- (0.5,0.25);
\draw[decorate,line width=0.06ex*\tkzsymblsscl] (0.28,0) -- (0.6,0.0);
\draw[decorate,line width=0.06ex*\tkzsymblsscl] (0.28,0) -- (0.5,-0.25);
\draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
\draw[rotate=-20] (0.27,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dWalley}[2][1=1, 2=yellow, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1,
decoration={random steps,segment length=0.15ex*\tkzsymblsscl, amplitude=0.1ex*\tkzsymblsscl}]
\shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
\draw[line width=0.06ex*\tkzsymblsscl]
(0.28,0) --++(0.05,0.07) --++(0.03,0.02) --+
+(0.03,-0.02) --++(0.03,0.1) --++(0.03,0.02) -- (0.5,0.25);
\draw[line width=0.06ex*\tkzsymblsscl]
(0.28,0) --++(0.06,-0.02) --++(0.04,0.06) --+
+(0.0,-0.08) --++(0.08,0.06) --++(0.03,-0.02) --+(0.08,0.02) -- (0.6,0.0);
\draw[line width=0.06ex*\tkzsymblsscl]
(0.28,0) --++(0.03,-0.02)  --++(0.03,-0.07)  --+
+(0.03,-0.01)  --++(0.01,-0.07)  --++(0.06,0.01)  --++(0.03,-0.08)   -- (0.5,0.-0.25);
\shade[ball color=#2, line width=0.08ex*\tkzsymblsscl] (-0.01,0) circle (0.31);
```

```
\draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
\draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\drWalley}[2][1=1, 2=yellow, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1,
decoration={random steps,segment length=0.15ex*\tkzsymblsscl, amplitude=0.1ex*\tkzsymblsscl}]
\shade[ball color=orange!80!black] (0.298,-0.33) rectangle (0.692,0.337);
\draw[decorate, line width=0.06ex*\tkzsymblsscl] (0.298,0) -- (0.5,0.25);
\draw[decorate,line width=0.06ex*\tkzsymblsscl] (0.298,0) -- (0.6,0.0);
\draw[decorate,line width=0.06ex*\tkzsymblsscl] (0.298,0) -- (0.5,-0.25);
\shade[ball color=#2, line width=0.08ex*\tkzsymblsscl] (-0.01,0) circle (0.31);
\draw[rotate=-20] (0.12,0.1) -- (0.2,0.05);
\draw[rotate=-20] (0.283,-0.1) .. controls (0.2,-0.072) and (0.1,-0.06) .. (0.,-0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Cat` *Miau*

```
\DeclareRobustCommandx{\Cat}[1][1=1,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=2.33ex,y=2.33ex, line width=0.093ex*\tkzsymblsscl,scale=#1]
\draw (0,0) circle (0.3);
\draw[rounded corners=0.163ex*\tkzsymblsscl] (-0.3,0) -- (-0.35,0.5) -- (0,0.3);
\draw[rounded corners=0.163ex*\tkzsymblsscl] (0,0.3) -- (0.35,0.5) -- (0.3,0);
\fill (-0.15,.15) circle (0.05);
\fill (0.15,.15) circle (0.05);
\draw[rounded corners=0.175ex*\tkzsymblsscl,yshift=-0.12ex]
   (0,0) -- (0,-0.1) -- (-0.1,-0.095);
\draw[rounded corners=0.175ex*\tkzsymblsscl,yshift=-0.12ex]
   (0,0) -- (0,-0.1) -- (0.1,-0.095);
\draw[rounded corners=.12ex*\tkzsymblsscl,yshift=-.15ex, line width=0.03em*(#1-.#1)]
(-0.1,0.1) -- (0,0) -- (0.1,0.1) -- cycle ;
\draw[line width=0.035ex*\tkzsymblsscl]
   (-0.1,-0.05)..controls(-0.25,0)and(-0.35,0).. (-0.4,-0.05);
\draw[line width=0.035ex*\tkzsymblsscl](-0.1,-0.05)..
   controls(-0.25,-0.01)and(-0.35,-0.09).. (-0.4,-0.14);
\draw[line width=0.035ex*\tkzsymblsscl](-0.1,-0.05)..
   controls(-0.25,-0.045)and(-0.35,-0.13).. (-0.4,-0.23);
\draw[line width=0.035ex*\tkzsymblsscl]
   (0.1,-0.05)..controls(0.25,0)and(0.35,0).. (0.4,-0.05);
\draw[line width=0.035ex*\tkzsymblsscl]
   (0.1,-0.05)..controls(0.25,-0.01)and(0.35,-0.09).. (0.4,-0.14);
\draw[line width=0.035ex*\tkzsymblsscl]
   (0.1,-0.05)..controls(0.25,-0.045)and(0.35,-0.13).. (0.4,-0.23);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

\Ninja \dNinja A Ninja.

```
\DeclareRobustCommandx{\Ninja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\def\Black@is@Black{black}%
\def\Black@or@not@Black{#2}%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1,
decoration={random steps,segment length=0.1ex*\tkzsymblsscl, amplitude=0.1ex*\tkzsymblsscl}]
\fill[#2, line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.33);
%\draw (-0.2,-0.125) -- ++(0.4,0);
\fill[decoration={random steps,segment length=0.1ex*\tkzsymblsscl,
  amplitude=0.01ex*\tkzsymblsscl}, decorate,#3]
(-0.33,0) -- (0.33,0)  -- (0.23,0.23) -- (-0.23,0.23) -- cycle;
\ifx\Black@or@not@Black\Black@is@Black
   \draw[line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.33);\fi
\fill[#3] (0,0.1) -- (-0.33,0) -- (-0.26,0.23);
\fill[#3] (0.3465,0) arc (0:42:0.34 and 0.345)  --
  (0.2,0.23)-- (0.31,0.0) -- cycle;
\fill[#3] (-0.3465,0) arc (0:-42:-0.34 and -0.345)  --
   (-0.2,0.23)-- (-0.31,0.0) -- cycle;
\fill[#4] (0.129,0.1425) arc (55:-180:.05);
\fill[#4] (-0.129,0.1425) arc (-55:180:-.05);
\draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymblsscl,
 segment length=0.55ex*\tkzsymblsscl}, #3]
  (0.26,0.21) -- (0.5,0.35);
\draw[decorate,decoration={snake,amplitude=.1ex*\tkzsymblsscl,
 segment length=0.55ex*\tkzsymblsscl}, #3]
  (0.26,0.21) -- (0.53,0.1);
\ifx\Black@or@not@Black\Black@is@Black
\else\draw[line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.33);\fi
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
\DeclareRobustCommandx{\dNinja}[4][1=1, 2=black, 3=red, 4=white, usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\def\Black@is@Black{black}%
\def\Black@or@not@Black{#2}%
\begin{tikzpicture}[x=2.4ex, y=2.4ex, line width=0.09ex*\tkzsymblsscl,scale=#1,
decoration={random steps,segment length=0.1ex*\tkzsymblsscl, amplitude=0.1ex*\tkzsymblsscl}]
\draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymblsscl,
segment length=0.55ex*\tkzsymblsscl},decorate, #3!50!black]
   (0.26,0.21) -- (0.5,0.35);
\draw[ decorate,decoration={snake,amplitude=.1ex*\tkzsymblsscl,
segment length=0.5ex*\tkzsymblsscl},decorate, #3!50!black]
   (0.26,0.21) -- (0.53,0.1);
\shade[ball color=#2, line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.347);
%\draw (-0.2,-0.125) -- ++(0.4,0);
\ifx\Black@or@not@Black\Black@is@Black
   \draw[line width=0.08ex*\tkzsymblsscl] (0,0) circle (0.33);\fi
\fill[decoration={random steps,segment length=0.1ex*\tkzsymblsscl,
```

```
 amplitude=0.01ex*\tkzsymblsscl},ball color=#3]
decorate  {(-0.33,0) -- (0.3465,0)  }
        {arc (0:42:0.34 and 0.345)}
decorate  {-- (-0.25,0.24)}
     { arc (-42:0:-0.375 and -0.345)};
\shade[ball color=#4] (0.129,0.1425) arc (55:-180:.05);
\shade[ball color=#4] (-0.129,0.1425) arc (-55:180:-.05);
\shade[top color=#4!80!black, bottom color=#4] (0.129,0.1425) arc (55:-180:.05);
\shade[top color=#4!80!black, bottom color=#4] (-0.129,0.1425) arc (-55:180:-.05);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

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

```
\DeclareRobustCommandx{\NiceReapey}[1][1=1,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.11em,y=0.11em, line width=0.07ex*\tkzsymblsscl,scale=#1]
\draw (1.7,-1) arc (360:180:1.7 and 2)
   arc (260:110:1.5 and 2) .. controls (-1,3.3) and (1,3.3) .. (1.9,2.97)
   arc (260:100:-1.3 and -2) -- cycle;
\fill[black!20!white] (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5) -- cycle;
\draw (3,-3) -- (3,3) .. controls (5,3) and (6,2) .. (7,1.5) -- (3,1.5);
\draw (0,-1.5) circle (1 and 0.5);
\draw[line width=0.04ex*\tkzsymblsscl] (-0.2,-1) -- (-0.2,-2);
\draw[line width=0.04ex*\tkzsymblsscl] (0.2,-1) -- (0.2,-2);
\draw[line width=0.04ex*\tkzsymblsscl] (0.6,-1) -- (0.6,-2);
\draw[line width=0.04ex*\tkzsymblsscl] (-0.6,-1) -- (-0.6,-2);
\draw[line width=0.04ex*\tkzsymblsscl] (-1,-1.5) -- (1,-1.5);
\fill (1.25,1.25) circle ( 0.5 and 0.75);
\fill (-1.25,1.25) circle ( 0.5 and 0.75);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

6.3 Other symbols(s)

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

```
\DeclareRobustCommandx{\Person}[5][1=1,2=-22,3=22,4=27,5=-27,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
 \begin{tikzpicture}[line width=0.12ex*\tkzsymblsscl, scale=#1, x=1.35ex, y=1.35ex]
 \draw[rotate around={#5:(0.15,0.2)}]  (0.15,0.2) -- (0.15,-0.14);
 \draw[rotate around={#4:(0.15,0.2)}]  (0.15,0.2) -- (0.15,-0.14);
 \draw (.15,.2) -- (.15,.4);
 \draw[rotate around={#3:(.15,.4)}] (.15,.4) -- (.42,.4);
 \draw[rotate around={#2:(.15,.4)}] (.15,.4) -- (-0.12,.4);
 \draw (.15, .4) -- (.15, .53);
 \draw (.15,.8) circle (0.18);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
```

```
}
```

`\Candle` A burning candle

```
\DeclareRobustCommand{\Candle}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1ex, y=1ex, scale=#1, line width=0.07ex*\tkzsymblsscl]
\draw[rounded corners=0.04ex*\tkzsymblsscl] (0,0) -- (0.2,0) -- +(0,1) -- (0,1) -- cycle;
\draw[line width=0.05ex*\tkzsymblsscl] (0.1,1) -- (0.1,1.2);
\draw[xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymblsscl]
   (-0.1,0.6) .. controls (-0.4,0.8) and (-0.1,1) ..  (-0.1,1.2);
\draw [xshift=0.95, yshift=2.2, line width=0.04ex*\tkzsymblsscl]
   (-0.1,0.6) .. controls (0.2,0.8) and (-0.1,1) ..  (-0.1,1.2);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Fire` Just a fire.

```
\DeclareRobustCommand{\Fire}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1ex,y=1ex,  line width=0.07ex*\tkzsymblsscl,rotate=45, scale=#1]
\fill (-0.05,0) -- (0.05,0) -- (0.05,0.95) -- (-0.05,0.95) -- cycle;
\fill (-0.74,0.7) -- (0.19,0.7) -- (0.19,0.8) -- (-0.74,0.8) -- cycle;
\fill[rotate=-20, xshift=-1.3, yshift=-0.1]
  (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
\fill[rotate=-70, xshift=-3.3, yshift=-2.3]
  (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
\fill[rotate=135, xshift=2.5, yshift=-3.8]
  (-0.05,0.07) -- (0.05,0.07) -- (0.05,0.9) -- (-0.05,0.9) -- cycle;
\draw[rotate=-45, xshift=-2.6, yshift=1.5,line width=0.04ex*\tkzsymblsscl, x=0.5ex, y=0.5ex]
(-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) ..  (0.05,1.7);
\draw[rotate=-45, xshift=-2.1,yshift=1.5,line width=0.04ex*\tkzsymblsscl, x=0.5ex, y=0.5ex]
(-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) ..  (-0.15,1.7);
\draw[rotate=-45, xshift=-2.5] (-0.1,0.29) .. controls (-0.7,0.6) and (0,1.2) ..  (0,1.5);
\draw[rotate=-45, xshift=-2] (-0.1,0.29) .. controls (0.7,0.6) and (-0.1,1.2) ..  (-0.1,1.5);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Cofeecup` Just a cup of coffee.

```
\if@tikzsymbols@marvosym\relax\else%
\DeclareRobustCommand{\Coffeecup}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.7ex,y=0.7ex, scale=#1, line width=0.07ex*\tkzsymblsscl,
decoration={snake,amplitude=.05ex*\tkzsymblsscl,segment length=0.408ex*\tkzsymblsscl}]
\draw (0,0) arc (180:269:0.8 and 1) -- ++(0.5,0) arc (269:360:0.8 and 1) -- cycle;
\draw (2.1,-0.15) -- (2.2,-0.15) arc (90:-90:0.3) -- (1.8, -0.75);
\draw[line width=0.05ex*\tkzsymblsscl, decorate]
   (0.4,0.3) -- +(0,1);
\draw[line width=0.05ex*#1, decorate]
```

```
   (1,0.3) -- +(0,1);
\draw[line width=0.05ex*#1, decorate]
   (1.6,0.3) -- +(0,1);
\draw (0,-1.05) -- (2.1,-1.05);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}%
\fi
```

`\Chair` A chair.

```
\DeclareRobustCommand{\Chair}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1} \@tkzssmbls@negtrue\fi%
\begin{tikzpicture}[x=0.9ex,y=0.9ex, scale=#1, line width=0.07ex*\tkzsymblsscl]
\draw (0,-0.5) -- (0,0.7) -- (0.5,1) -- (0.5,0.25);
\draw[line width=0.06ex*\tkzsymblsscl] (0,0.4) -- (0.5,0.7);
\draw (0,0) -- (0.5,0.3) -- (1,0) --(1,-0.5);
\if@tkzssmbls@neg\draw (0.5,0.3) -- +(0,-0.5);\fi
\draw (0.5,-0.3) -- (0.5,-0.8);
\draw (1,0) -- (0.5,-0.3) -- (0,0);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Bed` A chair.

```
\DeclareRobustCommand{\Bed}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1ex,y=1ex, scale=#1, line width=0.08ex*\tkzsymblsscl]
\draw (0,0) -- (0,1.6);
\draw (3,0) -- (3,1.2);
\draw (0,0.5) -- (3,0.5);
\draw (0,0.35) -- (3,0.35);
\draw (0.7,0.5) arc (0:90:0.7);
\draw (0.7,0.5) arc(180:30:1.231 and 0.6);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Tribar` Also called Penrose-Triangle

```
\DeclareRobustCommandx{\Tribar}[4][1=1,2={opacity=0},3={opacity=0},4={opacity=0},usedefault]
{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.65ex,y=0.65ex,scale=#1,
   rounded corners=0.03ex*\tkzsymblsscl, line width=0.06ex*\tkzsymblsscl]
\fill[#2] (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
   -- (-0.65,-0.6) -- (0.35,1.3) -- +(0.15,-0.3);
\fill[#3] (0,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- (-0.1,-0.3);
\fill[#4]  (1,0) -- (0.35,1.3) --  (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (1.3,0);
\draw (0,0) -- (1,0) -- (0.5,1) -- cycle;
\draw (0.15,0.3) -- (-0.15,-0.3) -- (1.75,-0.3) -- ++ (-0.15,-0.3)
```

```
   -- (-0.65,-0.6) -- (0.35,1.3) -- (0.8,.4);
\draw (0.9,0) -- (1.3,0) -- (0.35,1.9) -- (0.65,1.9) -- (1.75,-0.3) -- +(-.05,-0.1);
\draw (-0.6,-0.6) -- (-0.65,-0.6) -- ++ (-0.15,0.3) -- (0.35,1.9) -- (0.4,1.9);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Moai` From the Easter Island: a Moai.

```
\DeclareRobustCommandx{\Moai}[1][1=1,usedefault]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\ifdim \tkzsymblsscl<2pt%
\def\tikzsymbolsMoaithickness{0.05ex}%
\else%
\ifdim \tkzsymblsscl<5pt%
\def\tikzsymbolsMoaithickness{0.035ex}%
\else%
\def\tikzsymbolsMoaithickness{0.03ex}%
\fi\fi%
\begin{tikzpicture}[x=.13ex, y=.13ex, rounded corners=0.01ex*\tkzsymblsscl, scale=#1,
    line width=\tikzsymbolsMoaithickness*\tkzsymblsscl]
\draw  (-2.6,-4.25) --  (-2.5,-5.8)
..controls (-2,-6.8) and (1.5,-6.8) ..  (2.2,-5.8) -- (2.4,-3.95);
\draw(-2.5,2.5) .. controls (-2.9,4.6) and (2,5) .. (3.3,2.5) -- (2.9,-3.4)
.. controls (2,-5) and (-4,-5) .. (-3.1,-3) -- cycle;
\draw (-2.5,3) -- (-2,5) .. controls (0,6) and (2,5.8) .. (3.1,4.7) -- (3.3,2.5);
\draw[line width=0.02ex*\tkzsymblsscl]
   (-2.2,-1.8) .. controls (-1,-1.3) and (0,-1.7) .. (1,-2);
\draw[line width=0.02ex*\tkzsymblsscl]
   (-2.2,-1.8) .. controls (-1,-1) and (0,-1.4) .. (1,-2);
\draw[line width=0.02ex*\tkzsymblsscl]
   (-2.2,-1.8) .. controls (-1,-2) and (0,-2) .. (1,-2);
\draw (-0.8,4) .. controls (-0.8,3) and (-0.8,2) ..  (-1.6,0.5) -- (-1.8,-0.4)
   .. controls (-1,0.2) and (0,0.2) ..  (0.6,-0.4) -- (0.7,0.4)
   .. controls (0,1) and (0,2) .. (0.8,4);
\draw (-1.8,-0.4) .. controls (-0.5,-0.5) and (0,-0.5) .. (0.6,-0.4);
\draw (3.2,3.5) -- (3.7,3.5) .. controls (3.5,2) and (3.5,2) .. (3.6,-1.5) -- (3,-1.9);
\draw (-2.5,3) .. controls (-2.7,2) and (-3,1) .. (-2.88,-1);
\draw (-2.5,2.8) .. controls (-2,2.5) and (-1,3) .. (-0.8,3.1);
\draw (0.5,3.3) .. controls (1,3) and (1,2.5) .. (3.3,2.4);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

`\Snowman` A snowman. I think his smile is scary.

```
\DeclareRobustCommand{\Snowman}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=0.9ex,y=0.9ex,line width=0.07ex*\tkzsymblsscl, scale=#1]
\draw (0,0) circle (0.4 and 0.35);
\draw[line width=0.06ex*\tkzsymblsscl] (0,0.64) circle (0.3 and 0.28);
\draw[line width=0.05ex*\tkzsymblsscl] (0,1.14) circle (0.2 and 0.2);
```

```
\draw[rounded corners=0.1ex*\tkzsymblsscl,line width=0.05ex*\tkzsymblsscl,
   rotate around={-30:(0,1.14)}]
   (-0.2,1.15) -- ++(0,0.35) -- +(0.4,0) -- (0.2,1.14);
\draw[rounded corners=0.07ex*\tkzsymblsscl,line width=0.05ex*\tkzsymblsscl,
    rotate around={-30:(0,1.14)}]
    (-0.2,1.19) arc (270:90:0.1);
\fill (0,0.78) circle (0.04);
\fill (0,0.63) circle (0.04);
\fill (0,0.48) circle (0.04);
\fill (0,0.2) circle (0.05);
\fill (0,0) circle (0.05);
\fill (0,-0.2) circle (0.05);
\fill (-0.06,1.18) circle (0.045);
\fill (0.06,1.18) circle (0.045);
\fill (0.1,1.08) circle (0.015);
\fill (-0.1,1.08) circle (0.015);
\fill (0.06,1.055) circle (0.015);
\fill (-0.06,1.055) circle (0.015);
\fill (0.02,1.039) circle (0.015);
\fill (-0.02,1.039) circle (0.015);
\draw (-0.3,0.7) -- (-0.6,0.8);
\draw (-0.6,0.8) -- (-0.75,0.7);
\draw (-0.6,0.8) -- (-0.55,1);
\draw (-0.6,0.8) -- (-0.8,0.9);
\draw[line width=0.06ex*\tkzsymblsscl] (-0.65,0) -- (-0.65,1);
\foreach\x in {-0.85, -0.75,-0.65,-0.55,-0.45}
\draw[line width=0.05ex*\tkzsymblsscl] (-0.65,1) -- (\x,1.3);
\draw (0.3,0.7) -- (0.6,0.8);
\draw (0.6,0.8) -- (0.75,0.7);
\draw (0.6,0.8) -- (0.6,1);
\draw (0.6,0.8) -- (0.8,0.9);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

6.4 Trees

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

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

```
\newcommand\BasicTree[5][1]{%
\def\leaf@or@not@leaf{#5}%
\ifx\leaf@or@not@leaf\@leaf@is@leaf%
\Basic@Tree[#1]{#2}{#3}{#4}{#5}\tikzsymbolsaftersymbolinput%
\else%
\ifx\\#5\\%
\Basic@Tree[#1]{#2}{#3}{#4}{#5}\tikzsymbolsaftersymbolinput%
\else%
```

```
\PackageError{tikzsymbols}{The last\MessageBreak parameter has either to be \MessageBreak
‘leaf‘ or has to be empty}{See the tikzsymbols documentation. Section ‘‘Trees’’.}%
\fi\fi%
}
```

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

```
\DeclareRobustCommand{\WorstTree}[1][1]{%
\set@tkzsymblsscl{#1}\ifdim\tkzsymblsscl<0pt\set@tkzsymblsscl{-#1}\fi%
\begin{tikzpicture}[x=1ex,y=1ex, line width=0.04ex*\tkzsymblsscl,scale=#1]
\fill[brown] (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
   controls (0.5,0.7) and (0.5,0.3) .. (1,0);
\draw (-0.3,0) .. controls (0.2,0.3) and (0.2,0.7) .. (0.2,1) -- (0.5,1) ..
    controls (0.5,0.7) and (0.5,0.3) .. (1,0) ;
\fill[green] (0.2,0.8) --  (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) ..  (-0.3,1.2) ..
    controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
    controls (0.3,1.8) and (0.6,1.6) ..  (0.7,1.5) ..
    controls (1.1, 1.6) and (1,1.4) ..  (1,1.2) ..
    controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.5,0.8);
\draw (0.214,0.8) --  (0,0.8) .. controls (-0.4,0.7) and (-0.4,1) ..  (-0.3,1.2) ..
   controls (-0.3, 1.6) and (-0.1,1.6) .. (0.1,1.5) ..
   controls (0.3,1.8) and (0.6,1.6) ..  (0.7,1.5) .. controls (1.1, 1.6) and (1,1.4) ..
   (1,1.2) .. controls (1.2,1) and (1.2,0.7) .. (0.8,0.8) -- (0.486,0.8);
\fill[red] (0,1) circle (0.1);
\fill[red] (0.4,1.2) circle (0.1);
\fill[red] (0.8,1.1) circle (0.1);
\end{tikzpicture}%
\tikzsymbolsaftersymbolinput%
}
```

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

```
\DeclareRobustCommandx{\Springtree}[1][1=1, usedefault]%
  {\BasicTree[#1]{brown!70!black}{green!90!black}{green!80!black}{leaf}}
\DeclareRobustCommandx{\Summertree}[1][1=1, usedefault]%
  {\BasicTree[#1]{brown!50!black}{green!80!black}{red!80!green}{leaf}}
\DeclareRobustCommandx{\Autumntree}[1][1=1, usedefault]%
  {\BasicTree[#1]{red!30!black}{red!75!black}{orange}{leaf}}
\DeclareRobustCommandx{\Wintertree}[1][1=1, usedefault]%
  {\BasicTree[#1]{black!80!}{black!50}{black!25}{}}

\AtBeginDocument{
\if@tikzsymbols@marvosym
\@ifpackageloaded{marvosym}{}{%
\PackageError{tikzsymbols}{Use option 'marvosym' only\MessageBreak
if you load package 'marvosym'}
{Either load package 'marvosym' or\MessageBreak
delete the tikzsymbols option 'marvosym'}}
\fi
```

```
1113 }
```

Well that's it. Happy TEXing!

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