The `tikzfill` package

Manual for version 1.0.1 (2023/08/08)

Thomas F. Sturm

Cover code

```
% \documentclass[a4paper]{article}
% \usepackage{tikzfill}
% \usetikzlibrary{fadings,shadings}
% \usepackage[skins,breakable]{tcolorbox}
% \usepackage{ninecolors}
% \begin{document}
\begin{tcolorbox}[spread,blankest,phantom={\thispagestyle{empty}}]
\begin{tikzfadingfrompicture}[name=titlepicture]
  \path
    [
      pattern hexagon cycle = { size=28mm, rings=5 },
      pattern color         = white,
    ]
    (-\tcbtextwidth/2,-\tcbtextheight/2) rectangle
    (\tcbtextwidth/2,\tcbtextheight/2);
\end{tikzfadingfrompicture}%
%
\begin{tikzpicture}
  \fill
    [
      upper left = blue5,    upper right = cyan5,
      lower left = magenta5, lower right = blue5,
    ]
   (-\tcbtextwidth/2,-\tcbtextheight/2) rectangle
   (\tcbtextwidth/2,\tcbtextheight/2);
\shade
    [
      path fading = titlepicture,
      fit fading  = false,
      upper left = blue6,    upper right = cyan6,
      lower left = magenta6, lower right = blue6,
    ]
    (-\tcbtextwidth/2,-\tcbtextheight/2) rectangle
    (\tcbtextwidth/2,\tcbtextheight/2);
\node[white,font=\Huge\bfseries] (title) at (0,\tcbtextheight/4)
     {The \texttt{tikzfill} package};
\node[white,font=\Large\bfseries,below=8mm] (title) at (title.south)
     {Manual for version \version\ (\datum)};
\node[white,font=\large\bfseries,below=8mm] (title) at (title.south)
     {Thomas F.~Sturm};
\end{tikzpicture}
\end{tcolorbox}
% \end{document}
```

The `tikzfill` package

Manual for version 1.0.1 (2023/08/08)

Thomas F. Sturm[1]

https://www.ctan.org/pkg/tikzfill

https://github.com/T-F-S/tikzfill

Abstract

`tikzfill` is a collection of Ti*k*Z libraries which add further options to fill Ti*k*Z paths with images and patterns. The libraries comprise fillings with images from files and from Ti*k*Z pictures. Also, patterns of hexagons and of rhombi are provided.

[1]Prof. Dr. Dr. Thomas F. Sturm, Institut für Mathematik und Informatik, University of the Bundeswehr Munich, D-85577 Neubiberg, Germany; email: thomas.sturm@unibw.de

Contents

1 Short Introduction **5**

2 Image and Picture Fill Library **6**
2.1 Fill Plain . . . 6
2.2 Fill Stretch . . . 7
2.3 Fill Overzoom . . . 8
2.4 Fill Zoom . . . 9
2.5 Fill Shrink . . . 10
2.6 Fill Tile . . . 11
2.7 Filling Options . . . 12

3 Hexagon Pattern Library **14**
3.1 Hexagon . . . 14
3.2 Hexagon Grid . . . 18
3.3 Hexagon Cycle . . . 20

4 Rhombus Pattern Library **23**
4.1 Rhombus . . . 23

Index **27**

1 Short Introduction

Ti*k*Z is a very advanced and comprehensive graphics package for LaTeX. The package `tikzfill` comprises a collection of libraries for Ti*k*Z which add further options to fill Ti*k*Z paths with images and patterns.

For LaTeX, the provided libraries can be loaded using the preferred Ti*k*Z mechanism by

```
\usetikzlibrary{fill.***} % LaTeX (primary choice) and plain TeX
```

where *** is to be replaced by the actual library name found on the following pages.

Alternatively, the libraries can be loaded using LaTeX style files

```
\usepackage{tikzfill.***} % LaTeX (secondary choice)
```

If you want to load all Ti*k*Z libraries of this package, you can use the following LaTeX style file

```
\usepackage{tikzfill} % load all libraries
```

2 Image and Picture Fill Library

TikZ Library `fill.image`

```
\usetikzlibrary{fill.image} % LaTeX (primary choice) and plain TeX
\usetikzlibrary[fill.image] % ConTeXt
\usepackage{tikzfill.image} % LaTeX (secondary choice)
```

This library defines options to fill graphs with images or arbitray pictures.

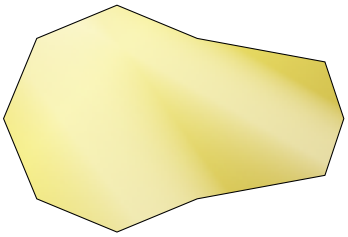
Until `tcolorbox` version 5.1.1 (2022/06/24), the code of this library was part of `tcolorbox`. Now, on suggestion of muzimuzhi, it is a separate library usable without `tcolorbox`. Also, the code is completely rewritten with `expl3`.

2.1 Fill Plain

`/tikz/fill plain image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is put in the center of the path, but it is not resized to fit into the path area.

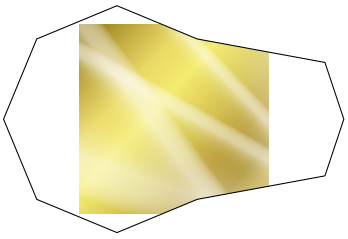
```
\begin{tikzpicture}
\path[draw,fill plain image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill plain image*=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is put in the center of the path, but it is not resized to fit into the path area. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command.

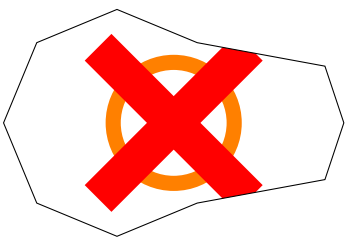
```
\begin{tikzpicture}
\path[draw,
    fill plain image*={width=2.5cm}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill plain picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is put in the center of the path, but it is not resized to fit into the path area. Note that this is almost identical to the standard `path picture` option.

```
\begin{tikzpicture}
\path[draw,fill plain picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

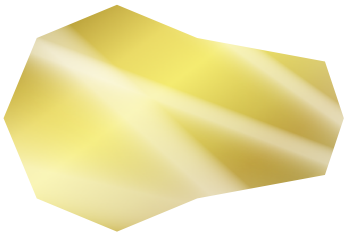


2.2 Fill Stretch

`/tikz/fill stretch image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is stretched to fill the path area.

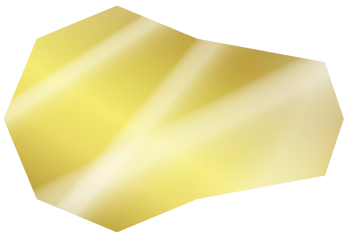
```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill stretch image*=`{⟨*graphics options*⟩}{⟨*file name*⟩} (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is stretched to fill the path area.

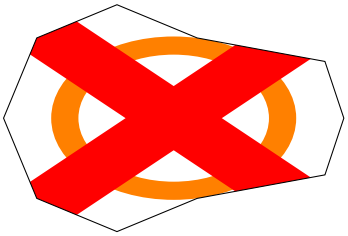
```
\begin{tikzpicture}
\path[fill stretch image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill stretch picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is stretched to fill the path area.

```
\begin{tikzpicture}
\path[draw,fill stretch picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

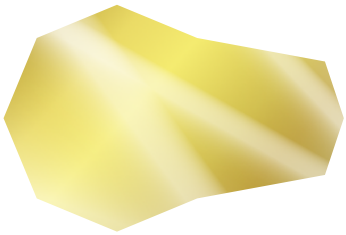


2.3 Fill Overzoom

`/tikz/fill overzoom image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that the path area fills the image.

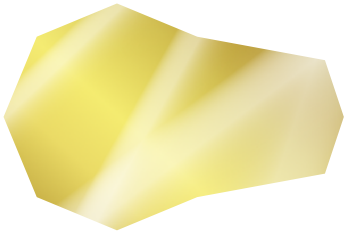
```
\begin{tikzpicture}
\path[fill overzoom image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill overzoom image*={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that the path area fills the image.

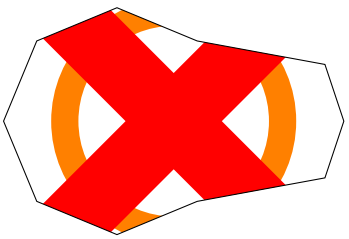
```
\begin{tikzpicture}
\path[fill overzoom image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill overzoom picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that the path area fills the image.

```
\begin{tikzpicture}
\path[draw,fill overzoom picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

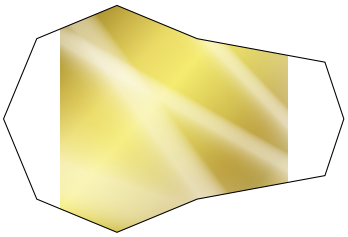


2.4 Fill Zoom

`/tikz/fill zoom image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

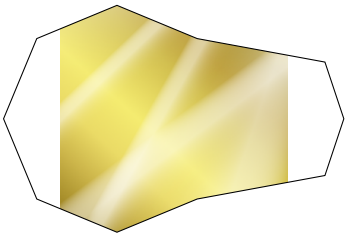
```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill zoom image*={`⟨*graphics options*⟩`}{`⟨*file name*⟩`}` (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

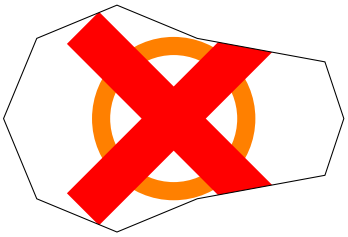
```
\begin{tikzpicture}
\path[draw,fill zoom image*=
  {angle=90,origin=c}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill zoom picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that it fits inside the path area. Typically, some parts of the path area will stay unfilled.

```
\begin{tikzpicture}
\path[draw,fill zoom picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

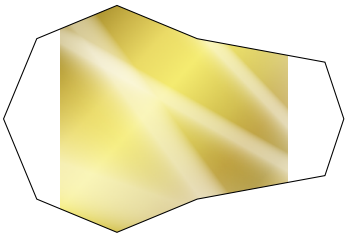


2.5 Fill Shrink

`/tikz/fill shrink image=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The image is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

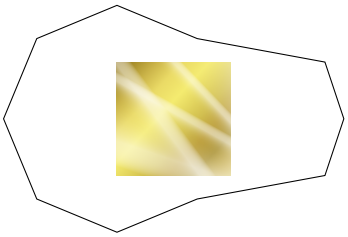
```
\begin{tikzpicture}
\path[draw,fill shrink image=goldshade.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill shrink image*=`⟨*file name*⟩ (no default, initially unset)

Fills the current path with an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command. The image is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

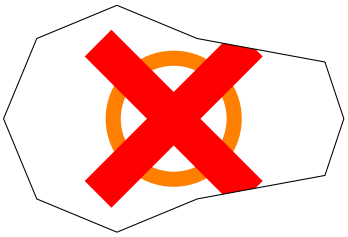
```
\begin{tikzpicture}
\path[draw,
    fill shrink image*={width=1.5cm}{goldshade.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill shrink picture=`⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with the given ⟨*graphical code*⟩. The result is zoomed such that it fits inside the path area, but it never gets enlarged. Typically, some parts of the path area will stay unfilled.

```
\begin{tikzpicture}
\path[draw,fill shrink picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



2.6 Fill Tile

`/tikz/fill tile image`=⟨*file name*⟩ (no default, initially unset)

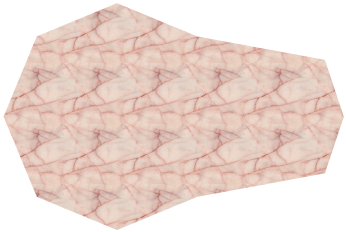
Fills the current path with a tile pattern using an external image referenced by ⟨*file name*⟩.

```
\begin{tikzpicture}
\path[fill tile image=pink_marble.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

`/tikz/fill tile image*`={⟨*graphics options*⟩}{⟨*file name*⟩} (no default, initially unset)

Fills the current path with a tile pattern using an external image referenced by ⟨*file name*⟩. The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command.

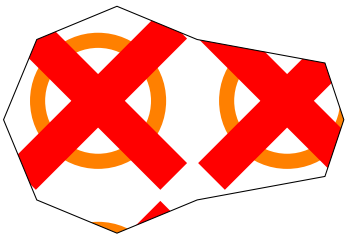
```
\begin{tikzpicture}
\path[fill tile image*={width=8mm}{pink_marble.png}]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill tile picture`=⟨*graphical code*⟩ (no default, initially unset)

Fills the current path with a tile pattern using the given ⟨*graphical code*⟩.

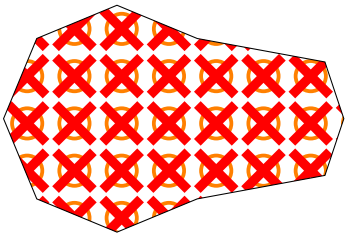
```
\begin{tikzpicture}
\path[draw,fill tile picture={%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill tile picture*`={⟨*fraction*⟩}{⟨*graphical code*⟩} (no default, initially unset)

Fills the current path with a tile pattern using the given ⟨*graphical code*⟩. The graphic is resized by ⟨*fraction*⟩.

```
\begin{tikzpicture}
\path[draw,fill tile picture*={0.25}{%
  \draw[red!50!yellow,line width=2mm]
    (0,0) circle (8mm);
  \draw[red,line width=5mm] (-1,-1) -- (1,1);
  \draw[red,line width=5mm] (-1,1) -- (1,-1);
  }]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



2.7 Filling Options

`/tikz/fill image opacity`=⟨*fraction*⟩ (no default, initially `1.0`)

Sets the fill opacity for the image or picture fill options to the given ⟨*fraction*⟩.

```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png] (0,0) circle (8mm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.75]
  (2,0) circle (8mm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.5]
  (4,0) circle (8mm);
\path[fill=red,fill stretch image=goldshade.png,fill image opacity=0.25]
  (6,0) circle (8mm);
\path[fill=red] (8,0) circle (8mm);
\end{tikzpicture}
```

`/tikz/fill image scale`=⟨*fraction*⟩ (no default, initially `1.0`)

Stretches, zooms, overzooms or shrinks the image or picture to the given ⟨*fraction*⟩ of the width and height of the current path.

```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
  (0,0) rectangle +(2,2);

\path[draw,fill zoom image=goldshade.png,fill image scale=0.75]
  (3,0) rectangle +(2,2);

\path[draw,fill zoom image=goldshade.png,fill image scale=1.5]
  (6,0) rectangle +(2,2);
\end{tikzpicture}
```

`/tikz/fill image options`=⟨*graphics options*⟩ (no default, initially empty)

The ⟨*graphics options*⟩ are given to the underlying `\includegraphics` command for the image fill options. This can be just together with `/tikz/fill stretch image` → P. 7, `/tikz/fill overzoom image` → P. 8, `/tikz/fill zoom image` → P. 9, and `/tikz/fill tile image` → P. 11.

```
\begin{tikzpicture}
\path[fill image options={width=8mm},
  fill tile image=pink_marble.png]
  (2.75,-0.75) -- (3,0) -- (2.75,0.75)
  \foreach \w in {45,90,...,315}
    { -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```

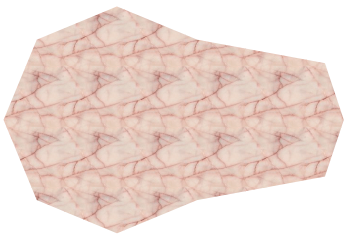
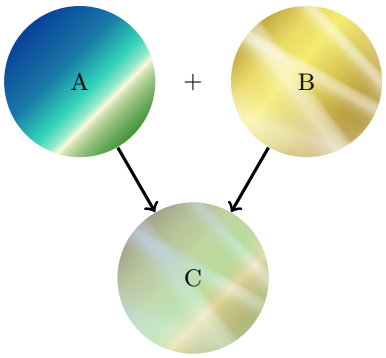


Image blending example

```
\begin{tikzpicture}[every node/.style=
  {circle,minimum width=2cm}]
\node[fill stretch image=blueshade.png]
  (A) at (120:3cm) {A};
\node[fill stretch image=goldshade.png]
  (B) at (60:3cm) {B};
\node[
  preaction={fill stretch image=blueshade.png},
  fill stretch image=goldshade.png,
  fill image opacity=0.5] (C) {C};
\path (A) -- node{$+$} (B);
\draw[->,very thick] (A)--(C);
\draw[->,very thick] (B)--(C);
\end{tikzpicture}
```



3 Hexagon Pattern Library

TikZ Library `fill.hexagon`

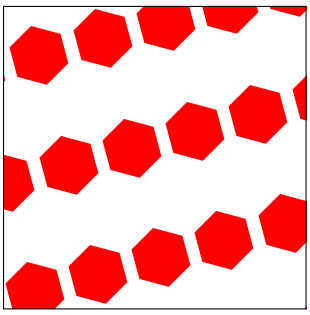
```
\usetikzlibrary{fill.hexagon} % LaTeX (primary choice) and plain TeX
\usetikzlibrary[fill.hexagon] % ConTeXt
\usepackage{tikzfill.hexagon} % LaTeX (secondary choice)
```

Based on `patterns.meta`, this library defines new hexagon patterns to fill graphs.

3.1 Hexagon

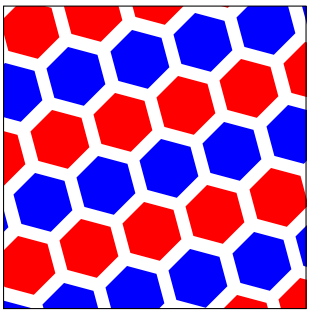
The `hexagon` pattern draws hexagons which may be filled or outlined. A single pattern is one of two different *bands*, called band 0 and band 1.

```
\begin{tikzpicture}
\draw[
  pattern = { hexagon
    [
      size = 5mm, angle = 15, line width = 1mm
    ]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



Both bands together build a uniform combined pattern.

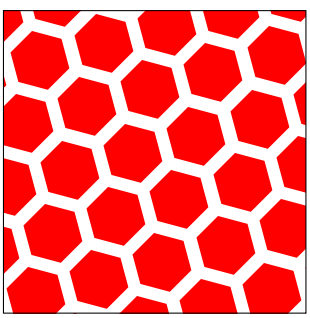
```
\begin{tikzpicture}
\draw[
  preaction = {
    pattern = { hexagon
      [
        size = 5mm, angle = 15, line width = 1mm, band = 1
      ]},
    pattern color=blue },
  pattern = { hexagon
    [
      size = 5mm, angle = 15, line width = 1mm, band = 0
    ]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/tikz/pattern hexagon={`⟨*pattern keys*⟩`}` (style, no default)

Convenience shortcut for setting the combined pattern (in one color).

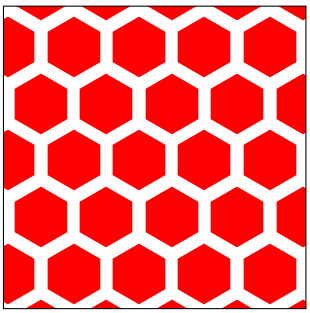
```
\begin{tikzpicture}
\draw[
  pattern hexagon =
    {
      size = 5mm, angle = 15, line width = 1mm
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/size`=⟨*size*⟩ (no default, initially `8mm`)

The given ⟨*size*⟩ denotes the length of an edge of one hexagonical tile where the (possibly smaller) hexagon is located in.

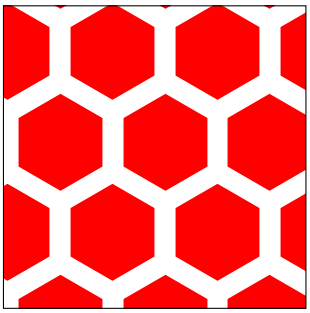
```
\begin{tikzpicture}
\draw[
  pattern hexagon =
    {
      size = 5mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/fill` (no value, initially set)

Sets the hexagons to be filled. `fill` and `draw` are mutually exclusionary.

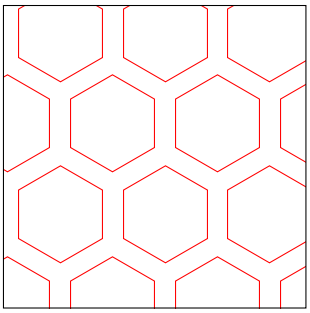
```
\begin{tikzpicture}
\draw[
  pattern hexagon =
    {
      fill,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/draw` (no value, initially unset)

Sets the hexagons to be outlined. `fill` and `draw` are mutually exclusionary.

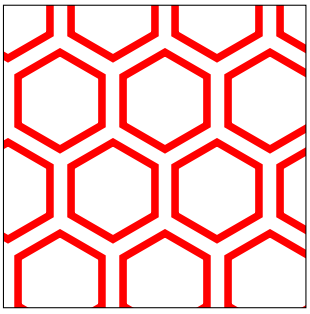
```
\begin{tikzpicture}
\draw[
  pattern hexagon =
    {
      draw,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/line width`=⟨*length*⟩ (no default, initially `0.4pt`)

Sets the ⟨*length*⟩ value of the line width. This is only relevant, if the hexagons are not filled.

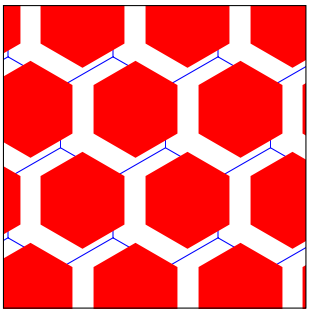
```
\begin{tikzpicture}
\draw[
  pattern hexagon =
    {
      draw, line width = 1mm,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/xshift=`⟨*xshift*⟩ (no default, initially `0pt`)
`/pgf/pattern keys/yshift=`⟨*yshift*⟩ (no default, initially `0pt`)

The pattern is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩.
Note that for **hexagon** is valid, that a pattern is shifted first and rotated afterwards.

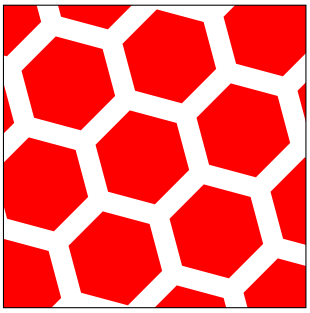
```
\begin{tikzpicture}
\draw[
  preaction={pattern hexagon grid, pattern color=blue},
  pattern hexagon =
    {
      xshift=3mm, yshift=1mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/angle=`⟨*angle*⟩ (no default, initially `0`)

The pattern is rotated by the given ⟨*angle*⟩.
Note that for **hexagon** is valid, that a pattern is shifted first and rotated afterwards.

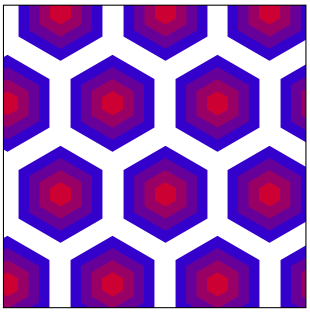
```
\begin{tikzpicture}
\draw[
  pattern hexagon =
    {
      angle = 15,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/pos=`⟨*value*⟩ (no default, initially `0.8`)

Sets the edge position with a ⟨*value*⟩ between 0 and 1, where 0 is the center and 1 the outer rim of the hexagonical tile. 1 is a less efficient way to either fill the whole graph or to draw a **hexagon grid**.

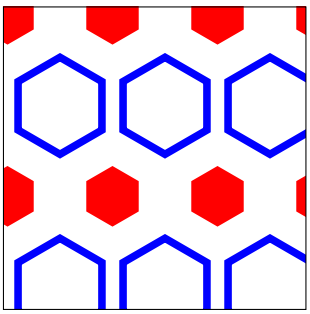
```
\begin{tikzpicture}
\draw[
  preaction={ pattern hexagon={pos=0.8},
    pattern color=blue!80!red },
  preaction={ pattern hexagon={pos=0.6},
    pattern color=blue!60!red },
  preaction={ pattern hexagon={pos=0.4},
    pattern color=blue!40!red },
  pattern hexagon={pos=0.2},
    pattern color=blue!20!red,
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/band=`⟨*number*⟩ (no default, initially `0`)

⟨*number*⟩ can take 0 or 1 and denotes one of two different bands of the pattern.

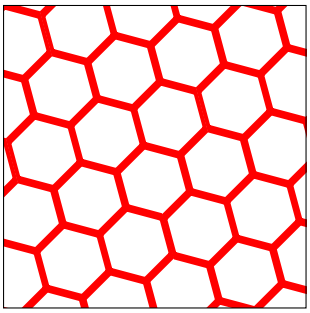
```
\begin{tikzpicture}
\draw[
  preaction = { pattern={hexagon[band=1,draw,
      line width=1mm]},
    pattern color=blue },
  pattern={hexagon[band=0,pos=0.5]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



3.2 Hexagon Grid

The **hexagon grid** pattern draws a grid made of hexagons. It is a single pattern und more efficient than **hexagon** with settings `draw,pos=1`.

```
\begin{tikzpicture}
\draw[
  pattern = { hexagon grid
    [
      size = 5mm, angle = 15, line width = 1mm
    ]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```

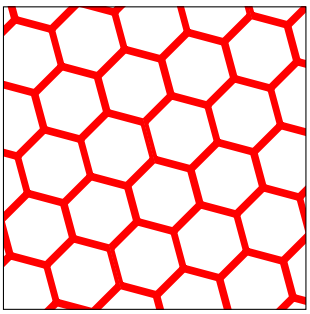


/tikz/pattern hexagon grid={⟨*pattern keys*⟩} (style, no default)

Convenience shortcut for setting the pattern to **hexagon grid**:

```
  pattern = { hexagon grid [ ... ] }
```

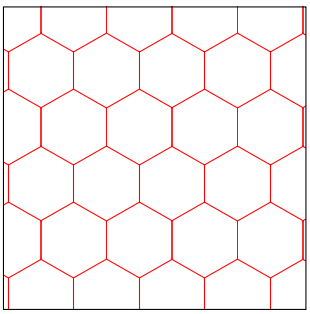
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
    {
      size = 5mm, angle = 15, line width = 1mm
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



/pgf/pattern keys/size=⟨*size*⟩ (no default, initially `8mm`)

The given ⟨*size*⟩ denotes the length of an edge of one hexagon.

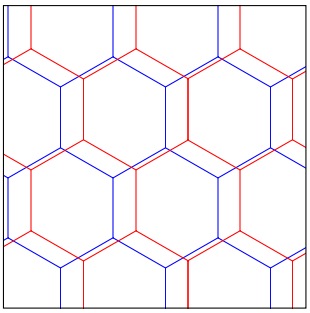
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
    {
      size = 5mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/xshift=`⟨*xshift*⟩ (no default, initially `0pt`)
`/pgf/pattern keys/yshift=`⟨*yshift*⟩ (no default, initially `0pt`)

The pattern is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩.
Note that for `hexagon grid` is valid, that a pattern is shifted first and rotated afterwards.

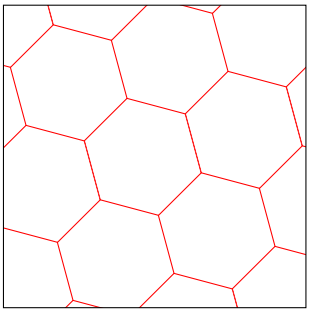
```
\begin{tikzpicture}
\draw[
  preaction={pattern={hexagon grid}, pattern color=blue},
  pattern hexagon grid =
    {
      xshift=3mm, yshift=1mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/angle=`⟨*angle*⟩ (no default, initially `0`)

The pattern is rotated by the given ⟨*angle*⟩.
Note that for `hexagon grid` is valid, that a pattern is shifted first and rotated afterwards.

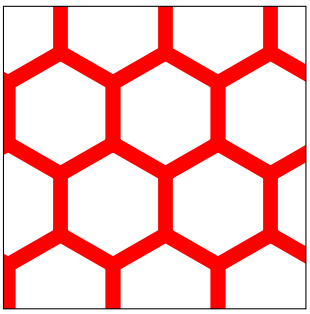
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
    {
      angle = 15,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/line width=`⟨*length*⟩ (no default, initially `0.4pt`)

Sets the ⟨*length*⟩ value of the line width.

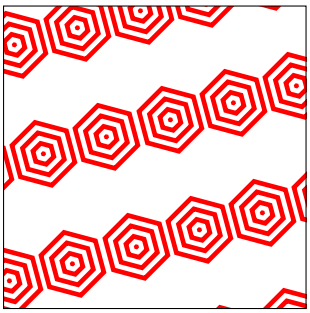
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
    {
      line width = 2mm,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



3.3 Hexagon Cycle

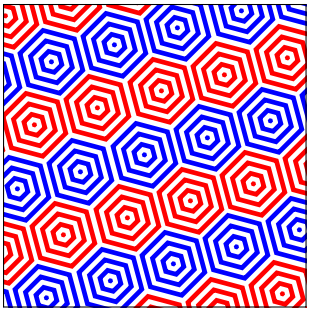
The **hexagon cycle** pattern draws several hexagon rings in a cyclic manor. A single pattern is one of two different *bands*, called band 0 and band 1.

```
\begin{tikzpicture}
\draw[
  pattern = { hexagon cycle
    [
      size = 5mm, angle = 15
    ]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



Both bands together build a uniform combined pattern.

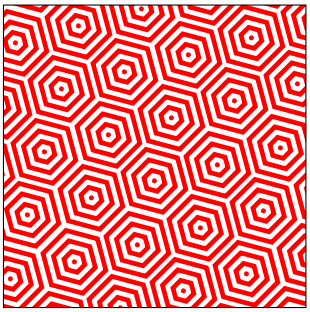
```
\begin{tikzpicture}
\draw[
  preaction = {
    pattern = { hexagon cycle
      [
        size = 5mm, angle = 15, band = 1
      ]},
    pattern color=blue },
  pattern = { hexagon cycle
    [
      size = 5mm, angle = 15, band = 0
    ]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



/tikz/pattern hexagon cycle={⟨*pattern keys*⟩} (style, no default)

Convenience shortcut for setting the combined pattern (in one color).

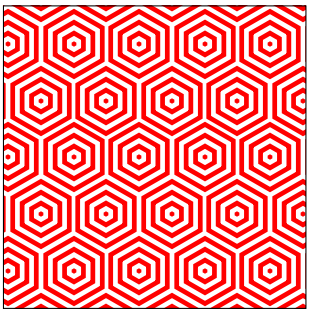
```
\begin{tikzpicture}
\draw[
  pattern hexagon cycle =
    {
      size = 5mm, angle = 15
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/size`=⟨*size*⟩ (no default, initially `8mm`)

The given ⟨*size*⟩ denotes the length of an edge of one hexagonical tile where the (smaller) hexagons are located in.

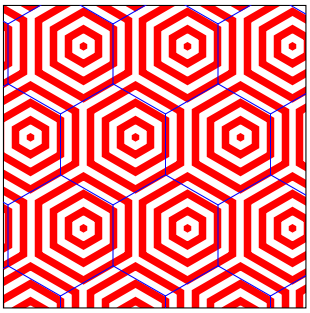
```
\begin{tikzpicture}
\draw[
  pattern hexagon cycle =
    {
      size = 5mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/xshift`=⟨*xshift*⟩ (no default, initially `0pt`)
`/pgf/pattern keys/yshift`=⟨*yshift*⟩ (no default, initially `0pt`)

The pattern is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩.
Note that for `hexagon cycle` is valid, that a pattern is shifted first and rotated afterwards.

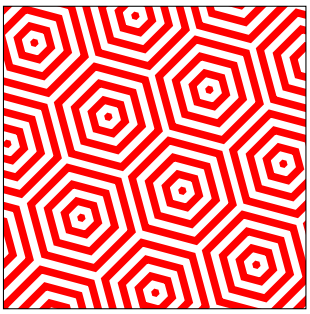
```
\begin{tikzpicture}
\draw[
  postaction={pattern={hexagon grid}, pattern color=blue},
  pattern hexagon cycle =
    {
      xshift=3mm, yshift=1mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/angle`=⟨*angle*⟩ (no default, initially `0`)

The pattern is rotated by the given ⟨*angle*⟩.
Note that for `hexagon cycle` is valid, that a pattern is shifted first and rotated afterwards.

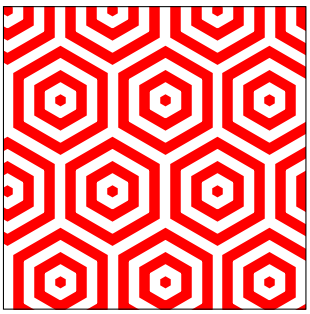
```
\begin{tikzpicture}
\draw[
  pattern hexagon cycle =
    {
      angle = 15,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



/pgf/pattern keys/rings=⟨*number*⟩ (no default, initially 3)

Sets the ⟨*number*⟩ of rings as $0, 1, 2, 3, \ldots$

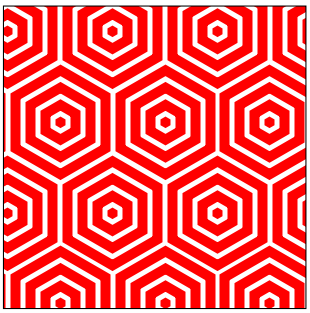
```
\begin{tikzpicture}
\draw[
  pattern hexagon cycle =
    {
      rings = 2,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



/pgf/pattern keys/gap=⟨*value*⟩ (no default, initially 1)

Sets the gap between two rings as ⟨*value*⟩ times the line width of a ring. ⟨*value*⟩ has to be greater or equal 0.01.

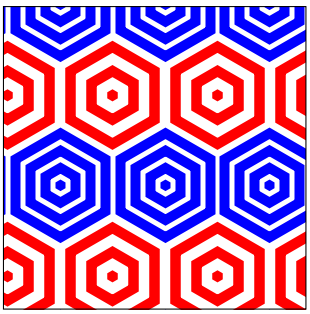
```
\begin{tikzpicture}
\draw[
  pattern hexagon cycle =
    {
      gap = 0.5,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



/pgf/pattern keys/band=⟨*number*⟩ (no default, initially 0)

⟨*number*⟩ can take 0 or 1 and denotes one of two different bands of the pattern.

```
\begin{tikzpicture}
\draw[
  preaction = { pattern={hexagon cycle[
    band=1, gap=0.5 ]}, pattern color=blue },
  pattern={hexagon cycle[band=0,rings=2]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



4 Rhombus Pattern Library

TikZ Library `fill.rhombus`

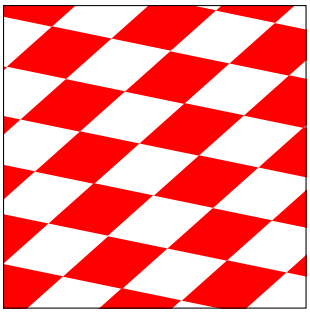
```
\usetikzlibrary{fill.rhombus} % LaTeX (primary choice) and plain TeX
\usetikzlibrary[fill.rhombus] % ConTeXt
\usepackage{tikzfill.rhombus} % LaTeX (secondary choice)
```

Based on `patterns.meta`, this library defines new rhombus patterns to fill graphs.

4.1 Rhombus

The **rhombus** pattern draws rhombi or diamonds. The rhombi may be filled or outlined and can be arranged in different *bands*, called band 0, band 1, and band 2.

```
\begin{tikzpicture}
\draw[
  pattern = { rhombus
    [
      size = 8mm, angle = 15
    ]},
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```

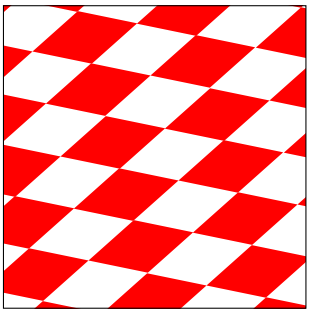


`/tikz/pattern rhombus={`⟨*pattern keys*⟩`}` (style, no default)

Convenience shortcut for setting the pattern to **rhombus**:

```
pattern = { rhombus [ ... ] }
```

```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      size = 8mm, angle = 15
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/size=`⟨*size*⟩ (no default, initially `10mm`)

The given ⟨*size*⟩ denotes the length of an edge of one rhombical tile where the (possibly smaller) rhombus is located in.

```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      size = 5mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```

`/pgf/pattern keys/fill` (no value, initially set)

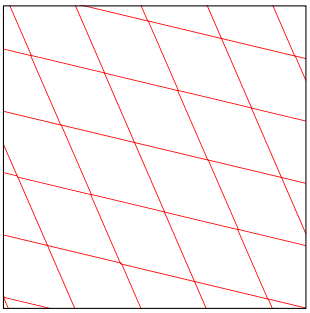
Sets the rhombi to be filled. `fill` and `draw` are mutually exclusionary.

```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      fill,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```

`/pgf/pattern keys/draw` (no value, initially unset)

Sets the rhombi to be outlined. `fill` and `draw` are mutually exclusionary.

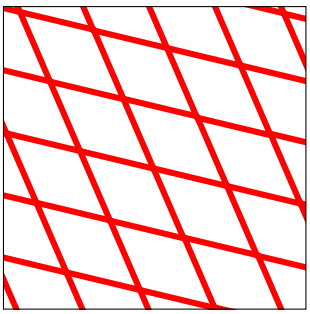
```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      draw,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/line width=`⟨*length*⟩ (no default, initially `0.4pt`)

Sets the ⟨*length*⟩ value of the line width. This is only relevant, if the rhombi are not filled.

```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      line width = 1mm, draw
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```

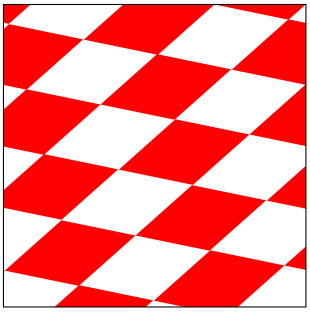


`/pgf/pattern keys/angle=`⟨*angle*⟩ (no default, initially `-40`)

The pattern is rotated by the given ⟨*angle*⟩.

Note that for `rhombus` is valid, that a pattern is rotated first and shifted afterwards.

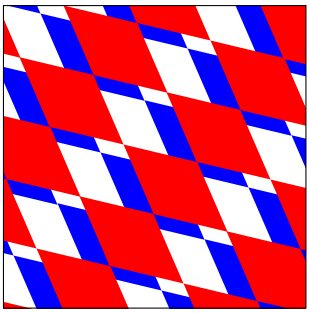
```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      angle = 15,
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/xshift=`⟨*xshift*⟩ (no default, initially `0pt`)
`/pgf/pattern keys/yshift=`⟨*yshift*⟩ (no default, initially `0pt`)

The pattern is shifted by ⟨*xshift*⟩ and ⟨*yshift*⟩.
Note that for **rhombus** is valid, that a pattern is rotated first and shifted afterwards.

```
\begin{tikzpicture}
\draw[
  preaction={pattern rhombus, pattern color=blue},
  pattern rhombus =
    {
      xshift=3mm, yshift=1mm,
    },
  pattern color=red
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/ratio=`⟨*value*⟩ (no default, initially `2`)

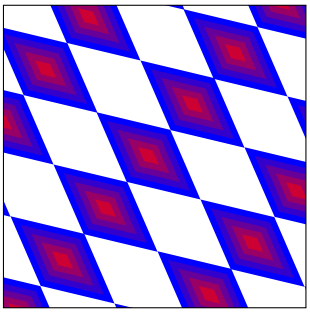
Sets the ⟨*value*⟩ of the ratio between the longer diagonal and the shorter diagonal. Therefore, ⟨*value*⟩ ≥ 1.

```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      ratio = 4
    },
  pattern color=red
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```

`/pgf/pattern keys/pos=`⟨*value*⟩ (no default, initially `1`)

Sets the edge position with a ⟨*value*⟩ between 0 and 1, where 0 is the center and 1 the outer rim of the rhombical tile.

```
\begin{tikzpicture}
\draw[
  preaction={ pattern rhombus={pos=1},
    pattern color=blue },
  preaction={ pattern rhombus={pos=0.8},
    pattern color=blue!80!red },
  preaction={ pattern rhombus={pos=0.6},
    pattern color=blue!60!red },
  preaction={ pattern rhombus={pos=0.4},
    pattern color=blue!40!red },
  pattern rhombus={pos=0.2},
    pattern color=blue!20!red,
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```

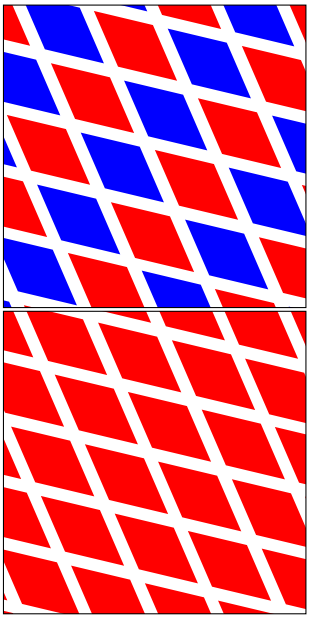


`/pgf/pattern keys/band`=⟨*number*⟩ (no default, initially 0)

⟨*number*⟩ can take 0, 1, or 2. Here, 0 and 1 denote one of two different bands of the pattern, while 2 denotes the comination of both.

```
\begin{tikzpicture}
\draw[
  preaction = {
    pattern rhombus = {
      pos = 0.8, band = 0 },
    pattern color=red },
  pattern rhombus = {
      pos = 0.8, band = 1
    },
  pattern color=blue
  ] (0,0) rectangle (4,4);
\end{tikzpicture}

\begin{tikzpicture}
\draw[
  pattern rhombus = {
      pos = 0.8, band = 2
    },
  pattern color=red
  ] (0,0) rectangle (4,4);
\end{tikzpicture}
```



Index

angle key, 16, 19, 21, 24

band key, 17, 22, 26

draw key, 15, 24

fill key, 15, 24
fill image opacity key, 12
fill image options key, 12
fill image scale key, 12
fill overzoom image key, 8
fill overzoom image* key, 8
fill overzoom picture key, 8
fill plain image key, 6
fill plain image* key, 6
fill plain picture key, 6
fill shrink image key, 10
fill shrink image* key, 10
fill shrink picture key, 10
fill stretch image key, 7
fill stretch image* key, 7
fill stretch picture key, 7
fill tile image key, 11
fill tile image* key, 11
fill tile picture key, 11
fill tile picture* key, 11
fill zoom image key, 9
fill zoom image* key, 9
fill zoom picture key, 9

gap key, 22

hexagon value, 14
hexagon cycle value, 20
hexagon grid value, 18

Keys
- /pgf/pattern keys/
  - angle, 16, 19, 21, 24
  - band, 17, 22, 26
  - draw, 15, 24
  - fill, 15, 24
  - gap, 22
  - line width, 15, 19, 24
  - pos, 16, 25
  - ratio, 25
  - rings, 22
  - size, 15, 18, 21, 23
  - xshift, 16, 19, 21, 25
  - yshift, 16, 19, 21, 25
- /tikz/
  - fill image opacity, 12
  - fill image options, 12
  - fill image scale, 12
  - fill overzoom image, 8
  - fill overzoom image*, 8
  - fill overzoom picture, 8
  - fill plain image, 6
  - fill plain image*, 6
  - fill plain picture, 6
  - fill shrink image, 10
  - fill shrink image*, 10
  - fill shrink picture, 10
  - fill stretch image, 7
  - fill stretch image*, 7
  - fill stretch picture, 7
  - fill tile image, 11
  - fill tile image*, 11
  - fill tile picture, 11
  - fill tile picture*, 11
  - fill zoom image, 9
  - fill zoom image*, 9
  - fill zoom picture, 9
  - pattern hexagon, 14
  - pattern hexagon cycle, 20
  - pattern hexagon grid, 18
  - pattern rhombus, 23

line width key, 15, 19, 24

pattern hexagon key, 14
pattern hexagon cycle key, 20
pattern hexagon grid key, 18
pattern rhombus key, 23
pos key, 16, 25

ratio key, 25
rhombus value, 23
rings key, 22

size key, 15, 18, 21, 23

Values
- hexagon, 14
- hexagon cycle, 20
- hexagon grid, 18
- rhombus, 23

xshift key, 16, 19, 21, 25

yshift key, 16, 19, 21, 25