



The tikzfill package

Manual for version 1.2.0 (2026/04/23)

Thomas F. Sturm

Cover code

```
% \documentclass[a4paper]{article}
% \usepackage{tikzfill}
% \usepackage[skins,breakable]{tcolorbox}
% \begin{document}
\begin{tcolorbox}[
  spread,
  blankest,
  center upper,
  phantom      = \thispagestyle{empty},
  top          = 65mm,
  fontupper    = \bfseries,
  frame engine = path,
  frame style  = {
    fill = none,
    preaction = {
      fill gea Weihnichl = {
        scale fixed = 1,
        xstep       = 21mm,
        ystep equidistance,
        style 1 = {},
        style 2 = {fill=black!5},
        style 3 = {fill=black!10},
        draw-if-row-col =
          {11/5,11/6,12/4,12/5,12/6,12/7,12/8,13/3,13/4,13/5,13/6,13/7,13/8},
        freeze-row,
      },
    },
  },
  fill gea Edenland = {
    scale fixed = 1,
    xstep       = 21mm,
    ystep equidistance,
    init-row< = \TFRowColorLet{mycolor}{red}{blue},
    prime color 1 = mycolor,
    draw-if-not-row-col =
      {11/5,11/6,12/4,12/5,12/6,12/7,12/8,13/3,13/4,13/5,13/6,13/7,13/8},
    freeze-row,
  },
},
]
\Huge The \texttt{tikzfill} package\!\! [10mm]
\Large Manual for version \version\ (\datum)\!\! [10mm]
\large Thomas F.-Sturm
\end{tcolorbox}
% \end{document}
```

The `tikzfill` package

Manual for version 1.2.0 (2026/04/23)

Thomas F. Sturm¹

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

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

Abstract

The `tikzfill` package is a collection of `TikZ` libraries that provide additional options for filling `TikZ` paths with images and patterns. The libraries support fillings based on external image files as well as `TikZ` pictures. They also include patterns of hexagons and rhombi.

In addition, the package offers fillings composed of similar, though not necessarily identical, small pictures arranged on a grid, providing greater flexibility than usual `TikZ` patterns.

¹Prof. 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
5	Geometric Array Library	27
5.1	Base Settings	27
5.2	Custom Creation of Geometric Arrays	34
5.3	Auxiliary Macros	41
5.4	Templates	42
	Index	59

1 Short Introduction

TikZ is a very advanced and comprehensive graphics package for L^AT_EX. The package `tikzfill` comprises a collection of libraries for TikZ which add further options to fill TikZ paths with images and patterns.

For L^AT_EX, the provided libraries can be loaded using the preferred TikZ 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 L^AT_EX style files

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

If you want to load all TikZ libraries of this package, you can use the following L^AT_EX 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 arbitrary pictures.

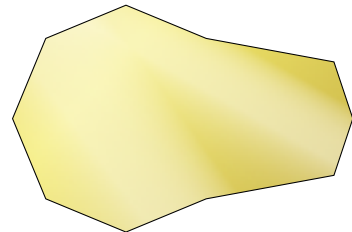
Until `tcolorbox` version 5.1.1 (2022/06/24), the code of this library was part of `tcolorbox`. Now, on suggestion of `muzimuzhi`, it is a separate library usable without `tcolorbox`. Also, the code is completely rewritten with `expl3`.

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.

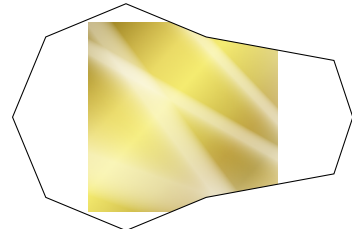
```
\begin{tikzpicture}
\path[draw,fill plain image=goldshade.png]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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.

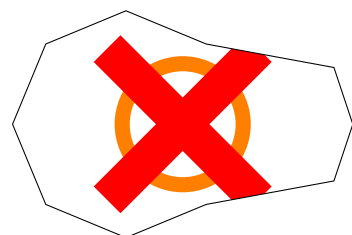
```
\begin{tikzpicture}
\path[draw,
fill plain image*={width=2.5cm}{goldshade.png}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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}
```

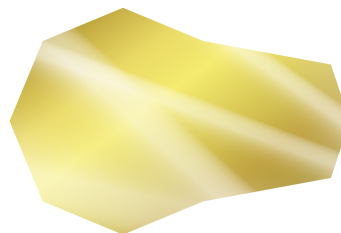


2.2 Fill Stretch

`/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.

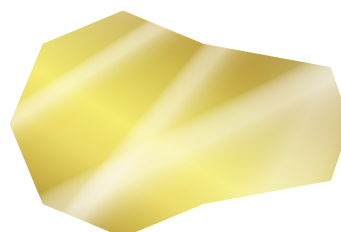
```
\begin{tikzpicture}
\path[fill stretch image=goldshade.png]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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.

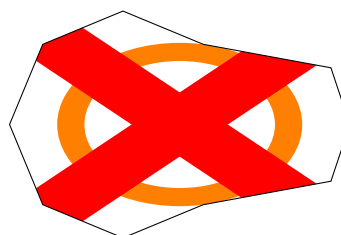
```
\begin{tikzpicture}
\path[fill stretch image*=
{angle=90,origin=c}{goldshade.png}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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}
```

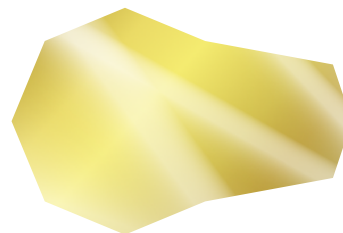


2.3 Fill Overzoom

`/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.

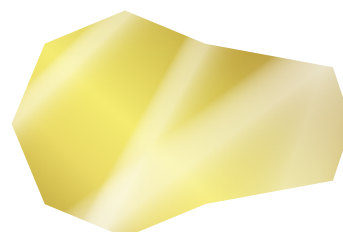
```
\begin{tikzpicture}
\path[fill overzoom image=goldshade.png]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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.

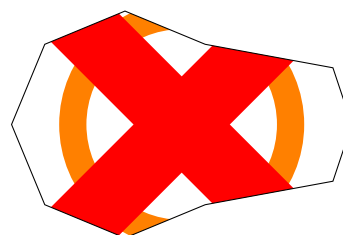
```
\begin{tikzpicture}
\path[fill overzoom image*=
{angle=90,origin=c}{goldshade.png}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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}
```

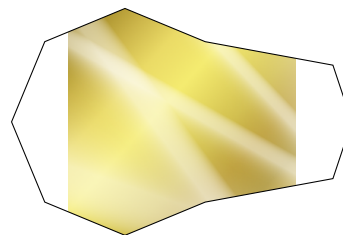


2.4 Fill Zoom

`/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.

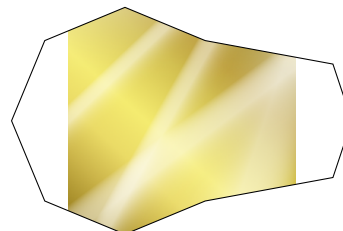
```
\begin{tikzpicture}
\path[draw,fill zoom image=goldshade.png]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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.

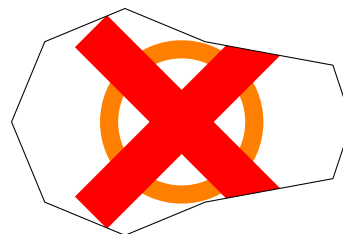
```
\begin{tikzpicture}
\path[draw,fill zoom image*=
{angle=90,origin=c}{goldshade.png}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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}
```

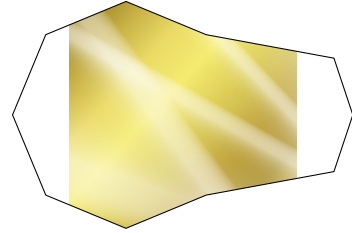


2.5 Fill Shrink

`/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.

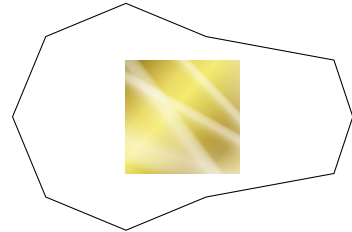
```
\begin{tikzpicture}
\path[draw,fill shrink image=goldshade.png]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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.

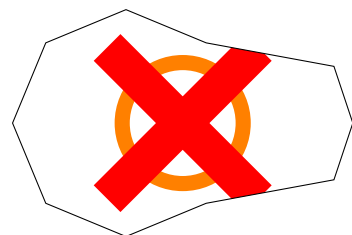
```
\begin{tikzpicture}
\path[draw,
fill shrink image*={width=1.5cm}{goldshade.png}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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>`.

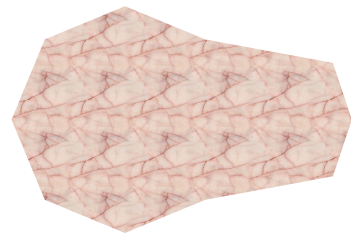
```
\begin{tikzpicture}
\path[fill tile image=pink_marble.png]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/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.

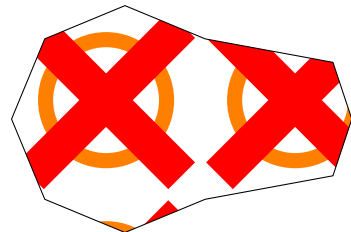
```
\begin{tikzpicture}
\path[fill tile image*={width=8mm}{pink_marble.png}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}
```



`/tikz/fill tile picture=<graphical code>` (no default, initially unset)

Fills the current path with a tile pattern using the given `<graphical code>`.

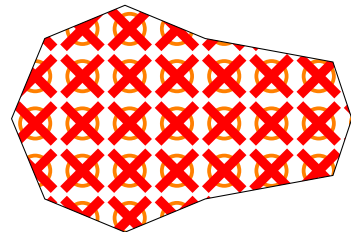
```
\begin{tikzpicture}
\path[draw,fill tile picture={%
\draw[red!50!yellow,line width=2mm]
(0,0) circle (8mm);
\draw[red,line width=5mm] (-1,-1) -- (1,1);
\draw[red,line width=5mm] (-1,1) -- (1,-1);
}]
(2.75,-0.75) -- (3,0) -- (2.75,0.75)
\foreach \w in {45,90,...,315}
{ -- (\w:1.5cm) } -- cycle;
\end{tikzpicture}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



`/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= $\langle fraction \rangle$` (no default, initially 1.0)

Sets the fill opacity for the image or picture fill options to the given $\langle fraction \rangle$.

```
\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= $\langle fraction \rangle$` (no default, initially 1.0)

Stretches, zooms, overzooms or shrinks the image or picture to the given $\langle fraction \rangle$ 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= $\langle graphics options \rangle$` (no default, initially empty)

The $\langle graphics options \rangle$ 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}
```

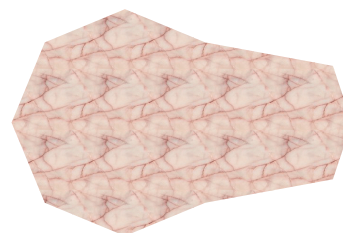
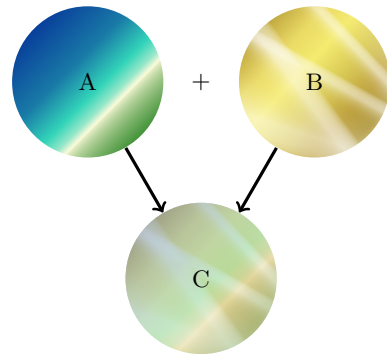


Image blending example

```
\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

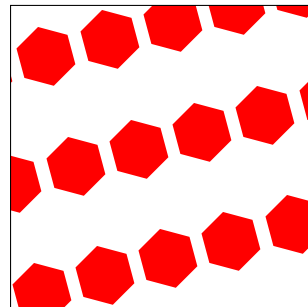
```
\usetikzlibrary{fill.hexagon} % LaTeX (primary choice) and plain TeX
\usetikzlibrary[fill.hexagon] % ConTeXt
\usepackage{tikzfill.hexagon} % LaTeX (secondary choice)
```

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

3.1 Hexagon

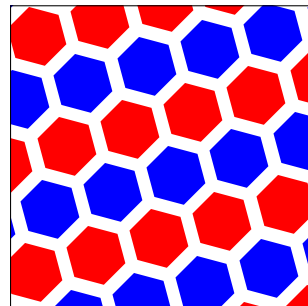
The `hexagon` pattern draws hexagons which may be filled or outlined. A single pattern is one of two different *bands*, called band 0 and band 1.

```
\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.

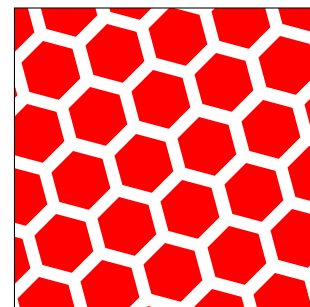
```
\begin{tikzpicture}
\draw[
  preaction = {
    pattern = { hexagon
      [
        size = 5mm, angle = 15, line width = 1mm, band = 1
      ]},
    pattern color=blue },
  pattern = { hexagon
    [
      size = 5mm, angle = 15, line width = 1mm, band = 0
    ]},
  pattern color=red
]
(0,0) rectangle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



`/tikz/pattern hexagon={\langle pattern keys \rangle}` (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}
```

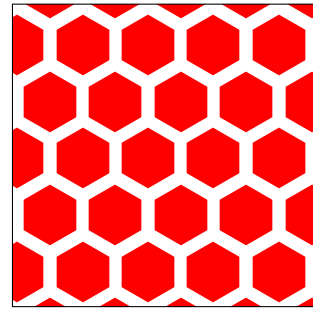


`/pgf/pattern keys/size=<size>`

(no default, initially 8mm)

The given $\langle size \rangle$ denotes the length of an edge of one hexagonal 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}
```

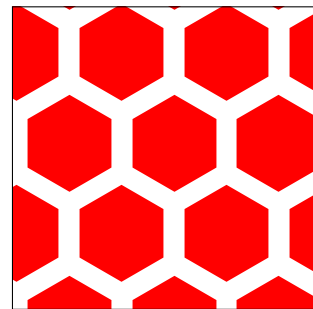


`/pgf/pattern keys/fill`

(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}
```

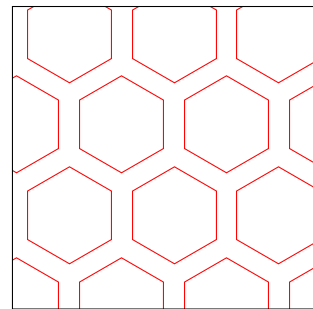


`/pgf/pattern keys/draw`

(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}
```

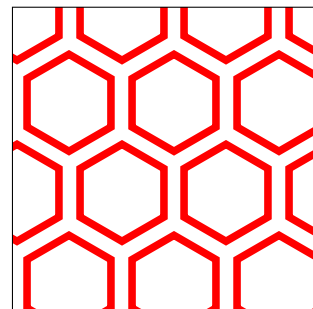


`/pgf/pattern keys/line width=<length>`

(no default, initially 0.4pt)

Sets the $\langle length \rangle$ 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}
```

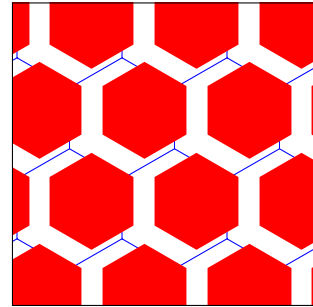


`/pgf/pattern keys/xshift= $\langle xshift \rangle$` (no default, initially 0pt)
`/pgf/pattern keys/yshift= $\langle yshift \rangle$` (no default, initially 0pt)

The pattern is shifted by $\langle xshift \rangle$ and $\langle yshift \rangle$.

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}
```

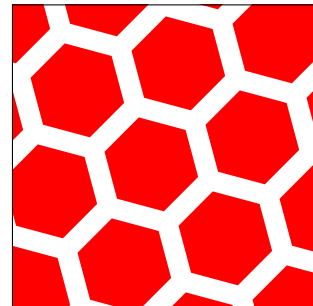


`/pgf/pattern keys/angle= $\langle angle \rangle$` (no default, initially 0)

The pattern is rotated by the given $\langle angle \rangle$.

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

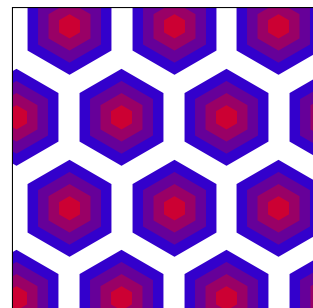
```
\begin{tikzpicture}
\draw[
pattern hexagon =
{
angle = 15,
},
pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/pos= $\langle value \rangle$` (no default, initially 0.8)

Sets the edge position with a $\langle value \rangle$ between 0 and 1, where 0 is the center and 1 the outer rim of the hexagonal 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}
```

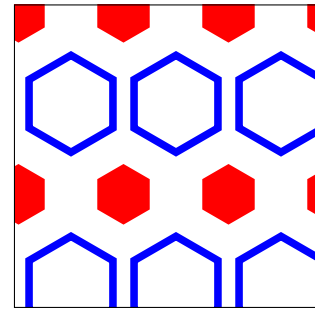


`/pgf/pattern keys/band=<number>`

(no default, initially 0)

`<number>` can take 0 or 1 and denotes one of two different bands of the pattern.

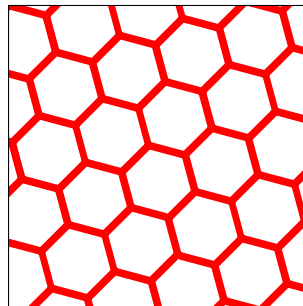
```
\begin{tikzpicture}
\draw[
  preaction = { pattern={hexagon[band=1,draw,
    line width=1mm]},
    pattern color=blue },
  pattern={hexagon[band=0,pos=0.5]},
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



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}
```

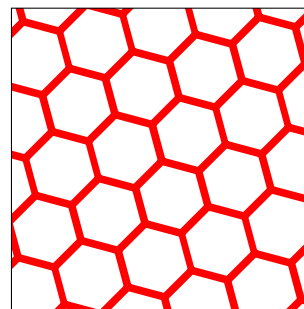


`/tikz/pattern hexagon grid={\langle pattern keys \rangle}` (style, no default)

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

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

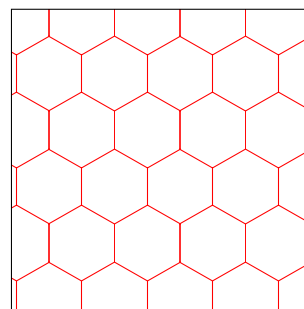
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
  {
    size = 5mm, angle = 15, line width = 1mm
  },
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/size=\langle size \rangle` (no default, initially 8mm)

The given `\langle size \rangle` 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}
```

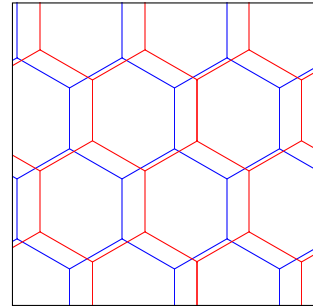


`/pgf/pattern keys/xshift= $\langle xshift \rangle$` (no default, initially 0pt)
`/pgf/pattern keys/yshift= $\langle yshift \rangle$` (no default, initially 0pt)

The pattern is shifted by $\langle xshift \rangle$ and $\langle yshift \rangle$.

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}
```

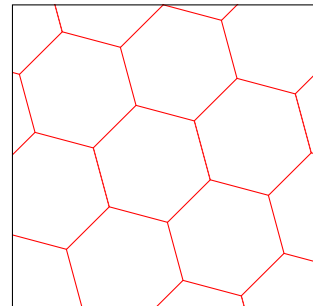


`/pgf/pattern keys/angle= $\langle angle \rangle$` (no default, initially 0)

The pattern is rotated by the given $\langle angle \rangle$.

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

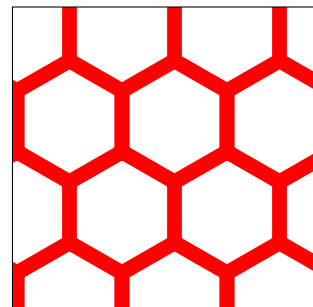
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
  {
    angle = 15,
  },
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/line width= $\langle length \rangle$` (no default, initially 0.4pt)

Sets the $\langle length \rangle$ value of the line width.

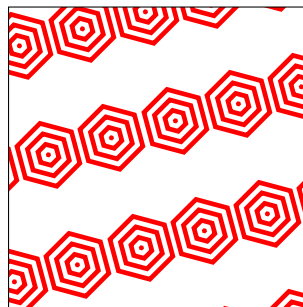
```
\begin{tikzpicture}
\draw[
  pattern hexagon grid =
  {
    line width = 2mm,
  },
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



3.3 Hexagon Cycle

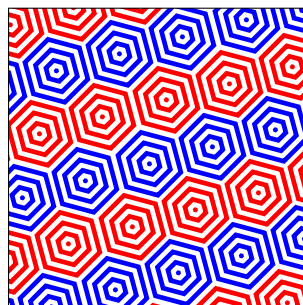
The `hexagon cycle` pattern draws several hexagon rings in a cyclic manor. A single pattern is one of two different *bands*, called band 0 and band 1.

```
\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.

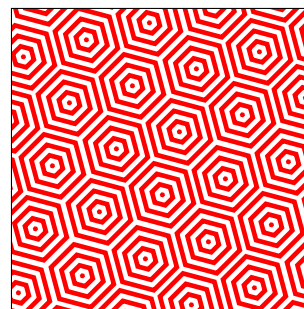
```
\begin{tikzpicture}
\draw[
  preaction = {
    pattern = { hexagon cycle
      [
        size = 5mm, angle = 15, band = 1
      ]},
    pattern color=blue },
  pattern = { hexagon cycle
    [
      size = 5mm, angle = 15, band = 0
    ]},
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}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



`/tikz/pattern hexagon cycle={\langle pattern keys \rangle}` (style, no default)

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

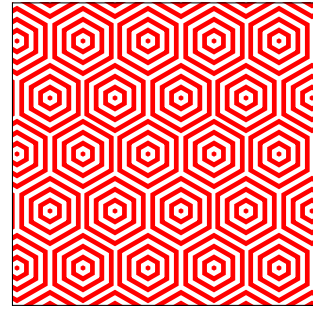
```
\begin{tikzpicture}
\draw[
  pattern hexagon cycle =
    {
      size = 5mm, angle = 15
    },
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/size=<size>` (no default, initially 8mm)

The given $\langle size \rangle$ denotes the length of an edge of one hexagonal 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}
```



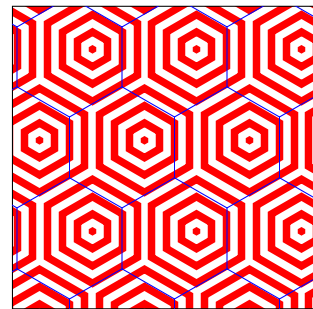
`/pgf/pattern keys/xshift=<xshift>` (no default, initially 0pt)

`/pgf/pattern keys/yshift=<yshift>` (no default, initially 0pt)

The pattern is shifted by $\langle xshift \rangle$ and $\langle yshift \rangle$.

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}
```

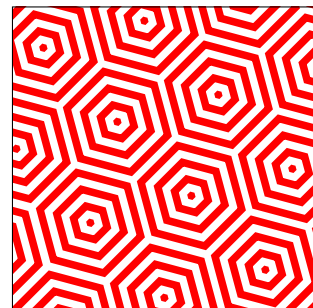


`/pgf/pattern keys/angle=<angle>` (no default, initially 0)

The pattern is rotated by the given $\langle angle \rangle$.

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}
```

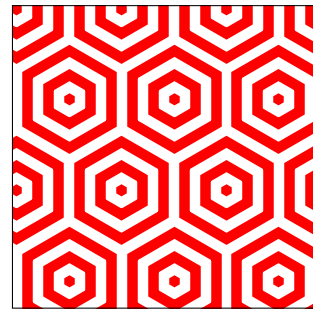


`/pgf/pattern keys/rings= $\langle number \rangle$`

(no default, initially 3)

Sets the $\langle number \rangle$ of rings as 0, 1, 2, 3, ...

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

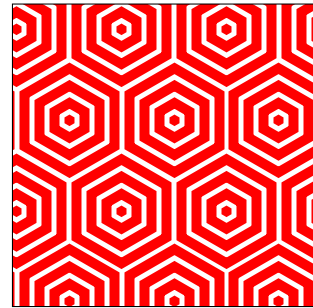


`/pgf/pattern keys/gap= $\langle value \rangle$`

(no default, initially 1)

Sets the gap between two rings as $\langle value \rangle$ times the line width of a ring. $\langle value \rangle$ 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}
```

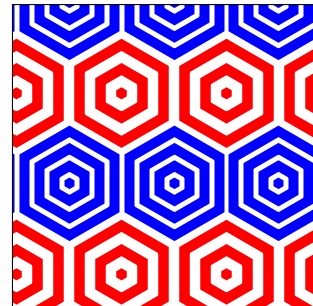


`/pgf/pattern keys/band= $\langle number \rangle$`

(no default, initially 0)

$\langle number \rangle$ 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

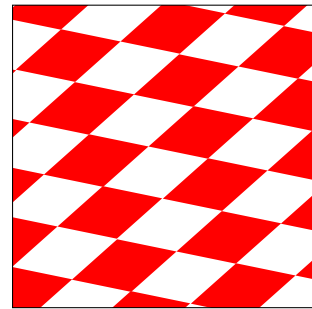
```
\usetikzlibrary{fill.rhombus} % LaTeX (primary choice) and plain TeX
\usetikzlibrary[fill.rhombus] % ConTeXt
\usepackage{tikzfill.rhombus} % LaTeX (secondary choice)
```

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}
```

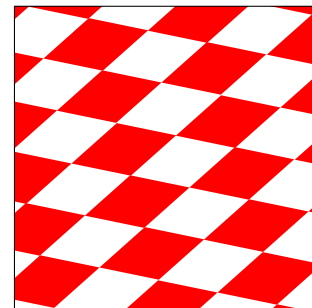


`/tikz/pattern rhombus={\langle pattern keys \rangle}` (style, no default)

Convenience shortcut for setting the pattern to `rhombus`:

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

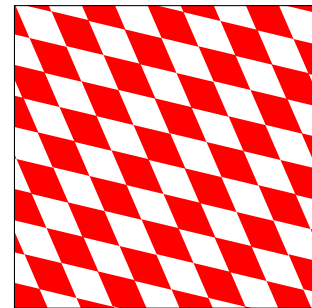
```
\begin{tikzpicture}
\draw[
  pattern rhombus =
    {
      size = 8mm, angle = 15
    },
  pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/size=\langle size \rangle` (no default, initially 10mm)

The given `\langle size \rangle` 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}
```

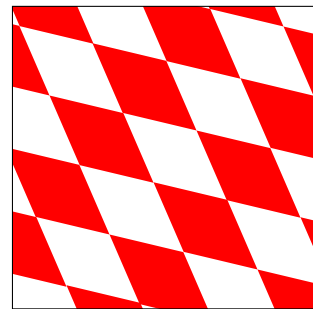


/pgf/pattern keys/fill

(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}
```

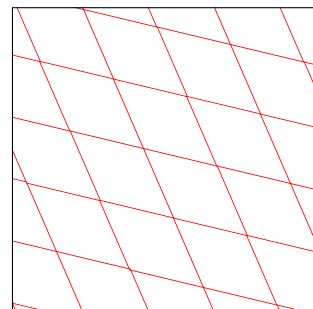


/pgf/pattern keys/draw

(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}
```

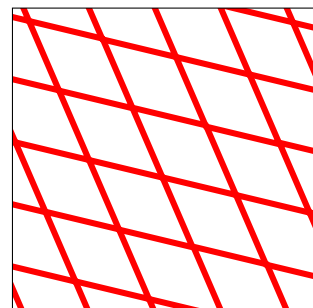


/pgf/pattern keys/line width=<length>

(no default, initially 0.4pt)

Sets the $\langle length \rangle$ 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}
```



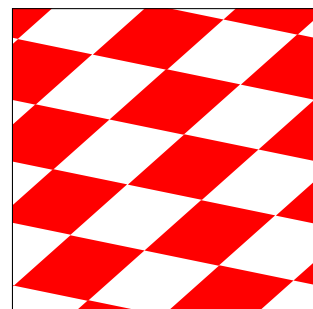
/pgf/pattern keys/angle=<angle>

(no default, initially -40)

The pattern is rotated by the given $\langle angle \rangle$.

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}
```

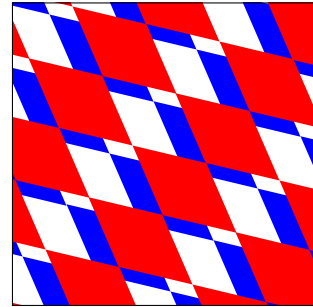


`/pgf/pattern keys/xshift= $\langle xshift \rangle$` (no default, initially 0pt)
`/pgf/pattern keys/yshift= $\langle yshift \rangle$` (no default, initially 0pt)

The pattern is shifted by $\langle xshift \rangle$ and $\langle yshift \rangle$.

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

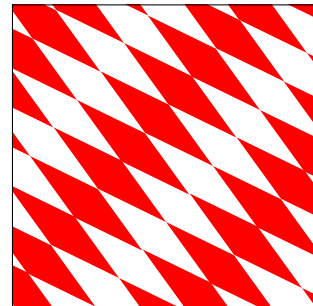
```
\begin{tikzpicture}
\draw[
preaction={pattern rhombus, pattern color=blue},
pattern rhombus =
{
xshift=3mm, yshift=1mm,
},
pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/ratio= $\langle value \rangle$` (no default, initially 2)

Sets the $\langle value \rangle$ of the ratio between the longer diagonal and the shorter diagonal. Therefore, $\langle value \rangle \geq 1$.

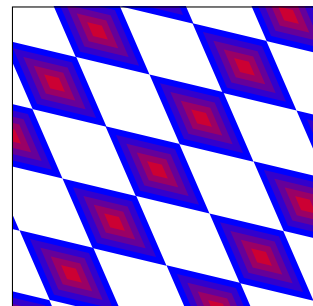
```
\begin{tikzpicture}
\draw[
pattern rhombus =
{
ratio = 4
},
pattern color=red
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/pgf/pattern keys/pos= $\langle value \rangle$` (no default, initially 1)

Sets the edge position with a $\langle value \rangle$ 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}
```



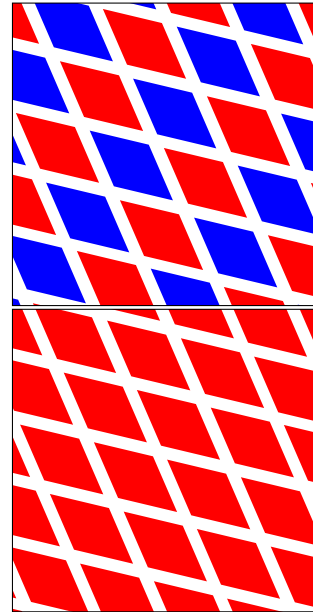
`/pgf/pattern keys/band=<number>`

(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 combination 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}
```



5 Geometric Array Library

TikZ Library fill.geomarray

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

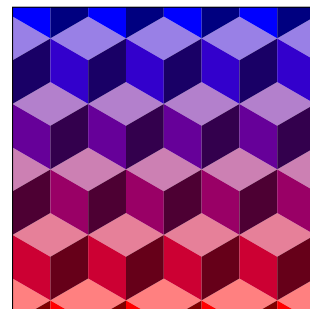
The fillings provided by this library consist of similar, though not necessarily identical, small pictures arranged on a grid. These pictures are conceptually similar to TikZ pic objects and are referred to as “items” in the following. Unlike TikZ tiling patterns, which are fixed in form, these pictures *may* vary. Note that this flexibility comes at the cost of increased compilation time.

Since TikZ already uses terms such as grid and pattern, we refer to such a pattern-like grid of pic-like objects as a “geometric array” of “items”.

Each geometric array consists of rows and columns of items arranged in either a rectangular or triangular grid. Built-in support is provided for scaling items from row to row, enabling fading-like effects.

The most common use case is to employ a predefined template (see Section 5.4) and adjust its geometry and colors.

```
\begin{tikzpicture}
\draw[ fill gea Edenland = {
  scale fixed = 1,
  step equi = 10mm,
  init-row< = \IFRowColorLet{mycolor}{red}{blue},
  prime color 1 = mycolor,
  freeze-row,
}]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



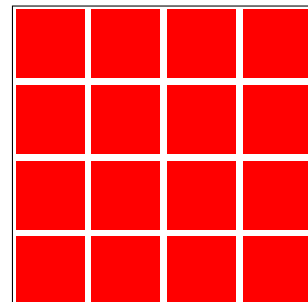
5.1 Base Settings

`/tikz/fill geomarray={\options}` (initially unset) N 2026-03-18

Applies a TikZ path picture with geometric array settings customized by various *options* to the current path. Without additional settings, this produces a default filling. Templates from Section 5.4 apply `/tikz/fill geomarray`.

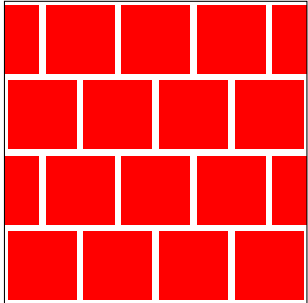
Note that the following keys are reset by this key: `/tikz/geomarray/init`^{P.34}, `/tikz/geomarray/init-row`^{P.35}, `/tikz/geomarray/init-item`^{P.35}, and `/tikz/geomarray/draw-item`^{P.35}.

```
\begin{tikzpicture}
\draw[ fill geomarray ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



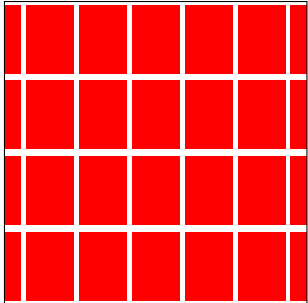
Geometric arrays can be either rectangular or triangular grids. Triangular grids shift every second row by half of a horizontal step.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { triangular },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



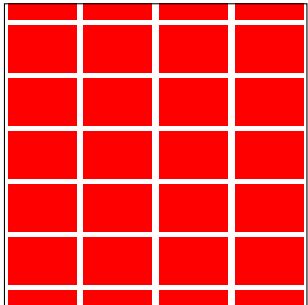
Sets the horizontal step distance between two columns to $\langle length \rangle$. This value can be used read-only as `\l_tikzfill_gea_xstep_dim` for item drawing.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { xstep = 7mm },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



Sets the vertical step distance between two rows to $\langle length \rangle$. This value can be used read-only as `\l_tikzfill_gea_ystep_dim` for item drawing.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { ystep = 7mm },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



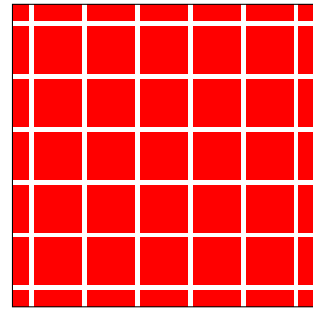
/tikz/geomarray/step={⟨length⟩}

(style)

N 2026-03-20

Sets both `/tikz/geomarray/xstep`^{→P.28} and `/tikz/geomarray/ystep`^{→P.28} to `⟨length⟩`.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { step = 7mm },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



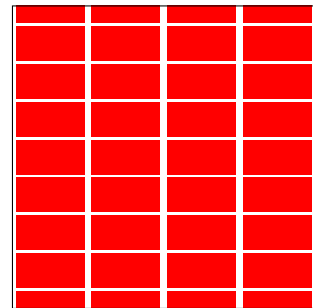
/tikz/geomarray/step half={⟨length⟩}

(style)

N 2026-03-23

Sets `/tikz/geomarray/xstep`^{→P.28} to `⟨length⟩` and `/tikz/geomarray/ystep`^{→P.28} to half of `⟨length⟩`.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { step half = 10mm },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



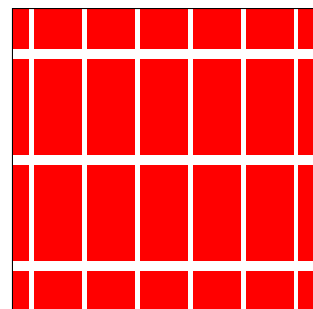
/tikz/geomarray/step double={⟨length⟩}

(style)

N 2026-03-23

Sets `/tikz/geomarray/xstep`^{→P.28} to `⟨length⟩` and `/tikz/geomarray/ystep`^{→P.28} to double of `⟨length⟩`.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { step double = 7mm },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/tikz/geomarray/ystep equidistance` (style, no value) [N 2026-03-18](#)
`/tikz/geomarray/ystep equidistance*` (style, no value) [N 2026-03-25](#)

For rectangular geometric arrays, this sets `/tikz/geomarray/ystep`^{→P.28} to `/tikz/geomarray/xstep`^{→P.28}. For triangular geometric arrays, `/tikz/geomarray/ystep`^{→P.28} is set such that the centers of the items become equidistant. Note that `/tikz/geomarray/triangular`^{→P.28} has to be set beforehand. `/tikz/geomarray/ystep equidistance*` switches rectangular and triangular computation.

```

\begin{tikzpicture}
\draw[
  fill gea Furth = {
    triangular,
    xstep = 7mm,
    ystep equidistance,
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}

```



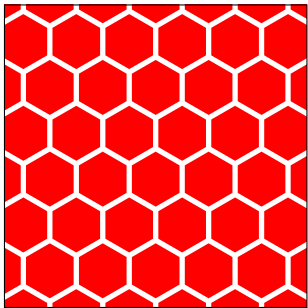
`/tikz/geomarray/step equi={⟨length⟩}` (style) [N 2026-03-20](#)
`/tikz/geomarray/step equi*={⟨length⟩}` (style) [N 2026-03-25](#)

Sets `/tikz/geomarray/xstep`^{→P.28} to `⟨length⟩` and applies `/tikz/geomarray/ystep equidistance` or `/tikz/geomarray/ystep equidistance*`, respectively.

```

\begin{tikzpicture}
\draw[
  fill gea Bruckberg = { step equi = 7mm },
]
  (0,0) rectangle (4,4);
\end{tikzpicture}

```



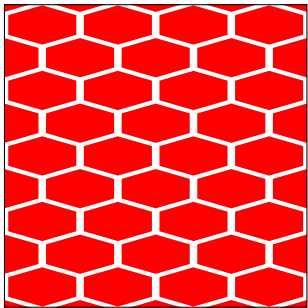
`/tikz/geomarray/step equi half={⟨length⟩}` (style) [N 2026-03-23](#)
`/tikz/geomarray/step equi* half={⟨length⟩}` (style) [N 2026-03-23](#)

Sets `/tikz/geomarray/xstep`^{→P.28} to `⟨length⟩` and applies `/tikz/geomarray/ystep equidistance` or `/tikz/geomarray/ystep equidistance*`, respectively, but halve ystep.

```

\begin{tikzpicture}
\draw[
  fill gea Bruckberg = { step equi half = 10mm },
]
  (0,0) rectangle (4,4);
\end{tikzpicture}

```



`/tikz/geomarray/step equi double={\langle length \rangle}`

(style)

[N 2026-03-23](#)

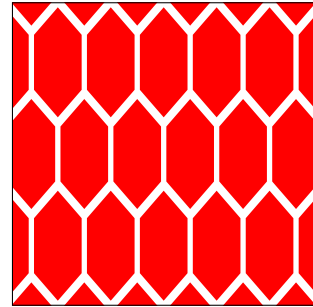
`/tikz/geomarray/step equi* double={\langle length \rangle}`

(style)

[N 2026-03-23](#)

Sets `/tikz/geomarray/xstep`^{→P.28} to `\langle length \rangle` and applies `/tikz/geomarray/ystep`^{→P.30} `equidistance`^{→P.30} or `/tikz/geomarray/ystep`^{→P.30} `equidistance*`^{→P.30}, respectively, but `double` `ystep`.

```
\begin{tikzpicture}
\draw[
  fill gea Bruckberg = { step equi double = 7mm },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



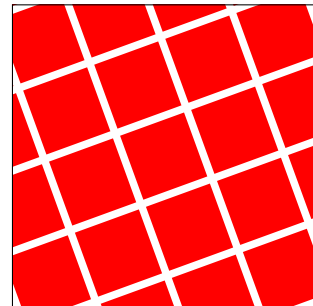
`/tikz/geomarray/angle={\langle angle \rangle}`

(initially 0)

[N 2026-03-18](#)

Rotates the whole geometric array by `\langle angle \rangle`.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { angle = 20 },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/tikz/geomarray/scale={⟨data points⟩}`

(initially 0/0.9,1/0.9)

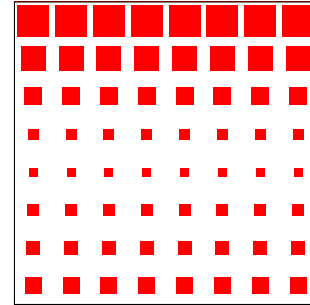
N 2026-03-18

Defines a comma-separated list of interpolation $\langle data\ points \rangle$ of type $t/scale$. The t values must be strictly monotonically increasing from 0 to 1 (including both endpoints). They represent relative row positions.

The $scale$ values are piecewise linearly interpolated and specify the actual scale of an item in a given row, where 1 typically denotes full scale (though not necessarily).

During item drawing, `\l_tikzfill_gea_row_frac_fp` represents the current relative row position, while `\l_tikzfill_gea_scale_fp` represents the interpolated scale value according to the specified interpolation $\langle data\ points \rangle$.

```
\begin{tikzpicture}
\draw[
  fill geomarray = {
    step equi = 5mm,
    scale = {0/0.5,0.5/0.2,1/1},
  },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



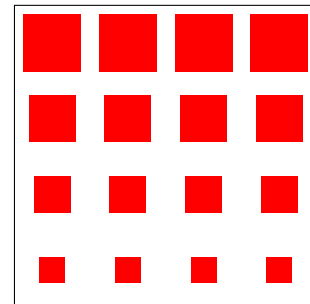
`/tikz/geomarray/scale linear={⟨scale0⟩}{⟨scale1⟩}`

(initially {0.9}{0.9})

N 2026-03-23

This is a shortcut for setting `/tikz/geomarray/scale={0/⟨scale0⟩,1/⟨scale1⟩}`, i.e. the scale changes linearly from $\langle scale0 \rangle$ to $\langle scale1 \rangle$ from the first to the last row of the geometric array.

```
\begin{tikzpicture}
\draw[
  fill geomarray = {
    scale linear = {0.2}{0.9},
  },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



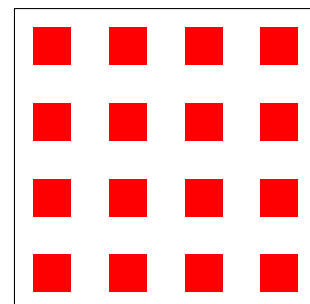
`/tikz/geomarray/scale fixed={⟨scale⟩}`

(initially 0.9)

N 2026-03-18

This is a shortcut for setting `/tikz/geomarray/scale={0/⟨scale⟩,1/⟨scale⟩}`, i.e. a fixed $\langle scale \rangle$ is used for all rows of the geometric array.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { scale fixed = 0.5 },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



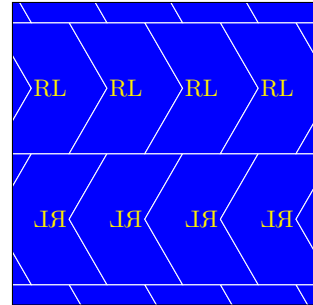
`/tikz/geomarray/xflip=true|false` (default true, initially false)
`/tikz/geomarray/xflip even=true|false` (default true, initially false)
`/tikz/geomarray/xflip odd=true|false` (default true, initially false)

For drawing an item in the geometric array, the x -value of all coordinates is multiplied by -1 , resulting in a reflection across the y -axis. This can be applied to all items or restricted to items in even or odd rows.

```

\begin{tikzpicture}
\draw[
  fill gea Hohenthann = {
    scale fixed = 1,
    step equi* double = 10mm,
    style 1 = {draw=white, fill=blue},
    text<={\color{yellow}RL},
    xflip odd,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}

```



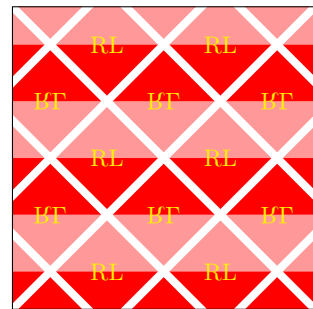
`/tikz/geomarray/yflip=true|false` (default true, initially false)
`/tikz/geomarray/yflip even=true|false` (default true, initially false)
`/tikz/geomarray/yflip odd=true|false` (default true, initially false)

For drawing an item in the geometric array, the y -value of all coordinates is multiplied by -1 , resulting in a reflection across the x -axis. This can be applied to all items or restricted to items in even or odd rows.

```

\begin{tikzpicture}
\draw[
  fill gea Pfettrach = {
    step half = 15mm,
    style 1 = { fill = red!40 },
    style 2 = { fill = red },
    text<={\color{yellow}RL},
    yflip even,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}

```



5.2 Custom Creation of Geometric Arrays

The following pseudocode illustrates how a geometric array is constructed, including four hook positions for inserting custom code.

Pseudo code for geometric array creation

```

init                % (1) initialize before looping
                   % Usable values:
                   % \l_tikzfill_gea_xstep_dim  \l_tikzfill_gea_ystep_dim
                   % \l_tikzfill_gea_cols_in    \l_tikzfill_gea_rows_int

\loop_over_rows
{
  init-row          % (2) initialize at begin of a row
                   % Usable values:
                   % \l_tikzfill_gea_row_tl
                   % \l_tikzfill_gea_row_frac_fp \l_tikzfill_gea_scale_fp

  \loop_over_columns
  {
    init-item       % (3) initialize before drawing an item
                   % Usable values:
                   % \l_tikzfill_gea_col_tl

    draw-item       % (4) item drawing code
  }
}

```

Code can be inserted at these four positions using the following options:

- The most important position is the last one, addressed by `/tikz/geomarray/draw-item`^{→P.35}. Here, the actual TikZ drawing code for the item must be provided.
- The first three positions, addressed by `/tikz/geomarray/init`, `/tikz/geomarray/init-row`^{→P.35}, and `/tikz/geomarray/init-item`^{→P.35}, may contain optional initialization code to prepare for subsequent drawing. No TikZ drawing code should be placed here; instead, these positions are intended, for example, for calculations affecting the entire array or a single row.
- In addition to providing fully custom code, a practical and convenient approach is to apply a template (see Section 5.4) and adjust only selected settings at one of the four positions.

```

/tikz/geomarray/init={⟨code⟩} (initially empty)
/tikz/geomarray/init<={⟨code⟩}
/tikz/geomarray/init>={⟨code⟩}

```

N 2026-03-18

This code position is executed before the loop over rows begins. The following read-only values are available: the grid step sizes `\l_tikzfill_gea_xstep_dim` and `\l_tikzfill_gea_ystep_dim`, as well as the number of columns and rows given by `\l_tikzfill_gea_cols_int` and `\l_tikzfill_gea_rows_int` (the actual counts are one greater, since indexing starts at 0).

- `/tikz/geomarray/init` replaces the contents by `⟨code⟩`.
- `/tikz/geomarray/init<` appends `⟨code⟩`.
- `/tikz/geomarray/init>` prepends `⟨code⟩`.

```

/tikz/geomarray/init-row={⟨code⟩} (initially empty)
/tikz/geomarray/init-row<={⟨code⟩}
/tikz/geomarray/init-row>={⟨code⟩}

```

N 2026-03-18

This code position is inside the main loop over rows, at the beginning of row `\l_tikzfill_gea_row_tl`. Here, `\l_tikzfill_gea_row_frac_fp` denotes the relative row position as a fractional value between 0 and 1. The macro `\l_tikzfill_gea_scale_fp` represents the current scale value for the row, as defined by `/tikz/geomarray/scale`^{P.32}.

- `/tikz/geomarray/init-row` replaces the contents by `⟨code⟩`.
- `/tikz/geomarray/init-row<` appends `⟨code⟩`.
- `/tikz/geomarray/init-row>` prepends `⟨code⟩`.

```

/tikz/geomarray/init-item={⟨code⟩} (initially empty)
/tikz/geomarray/init-item<={⟨code⟩}
/tikz/geomarray/init-item>={⟨code⟩}

```

N 2026-03-18

This code position is inside the inner loop over columns, at the beginning of column `\l_tikzfill_gea_col_tl`. Here, final calculations or settings may be applied before the current item is drawn.

- `/tikz/geomarray/init-item` replaces the contents by `⟨code⟩`.
- `/tikz/geomarray/init-item<` appends `⟨code⟩`.
- `/tikz/geomarray/init-item>` prepends `⟨code⟩`.

```

/tikz/geomarray/draw-item={⟨code⟩} (initially special)
/tikz/geomarray/draw-item<={⟨code⟩}
/tikz/geomarray/draw-item>={⟨code⟩}

```

N 2026-03-18

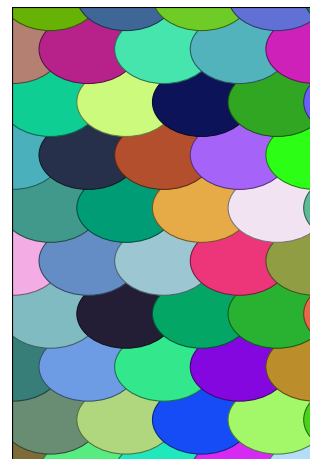
This code position contains the TikZ drawing code for the current item. All previously mentioned read-only values are available for use in drawing. The center of the item corresponds to the TikZ position (0,0). The style settings `/tikz/fill-gea-style-1` to `/tikz/fill-gea-style-9` may be used as drawing options.

- `/tikz/geomarray/draw-item` replaces the contents by `⟨code⟩`.
- `/tikz/geomarray/draw-item<` appends `⟨code⟩`.
- `/tikz/geomarray/draw-item>` prepends `⟨code⟩`.

```

\ExplSyntaxOn
\begin{tikzpicture}
\draw[ fill-geomarray = {
  scale~fixed = 1.3,
  triangular,
  ystep = 7mm,
  init = \sys_gset_rand_seed:n {1701},
  init-item = { \TFDefineRandomColor{mycolor} },
  draw-item = {
    \path[fill=mycolor,draw=mycolor!50!black]
    circle [
      x~radius=\fp_to_dim:n {\l_tikzfill_gea_xstep_dim / 2
        * \l_tikzfill_gea_scale_fp },
      y~radius=\fp_to_dim:n {\l_tikzfill_gea_ystep_dim / 2
        * \l_tikzfill_gea_scale_fp },
    ];
  },
}]
(0,0) rectangle (4,6);
\end{tikzpicture}
\ExplSyntaxOff

```



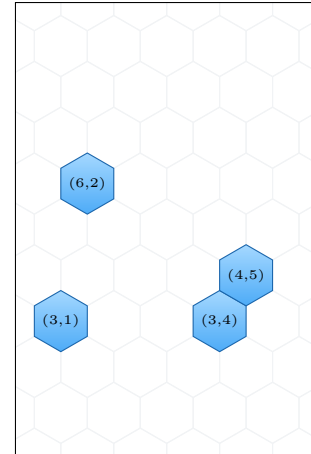
`/tikz/geomarray/draw-if-row-col={⟨coordinate list⟩}`

(initially unset)

N 2026-04-17

Restricts item drawing to those items whose coordinates are contained in the specified comma-separated *⟨coordinate list⟩* of type *⟨row⟩/⟨col⟩*. The keys `/tikz/geomarray/init`^{P.34} and `/tikz/geomarray/init-item`^{P.35} are used to append the corresponding code.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  preaction = {
    fill gea Bruckberg = {
      scale fixed = 1, step equi = 7mm,
      style 1      = { draw = oc-gray-1 },
      freeze-row,
    },
  },
  fill gea Bruckberg = {
    scale fixed = 1, step equi = 7mm,
    style 1 = {
      draw      = oc-blue-9,
      top color = oc-blue-2,
      bottom color = oc-blue-4 },
    debug text,
    draw-if-row-col = { 4/5, 3/1, 3/4, 6/2 },
  ]
(0,0) rectangle (4,6);
\end{tikzpicture}
```



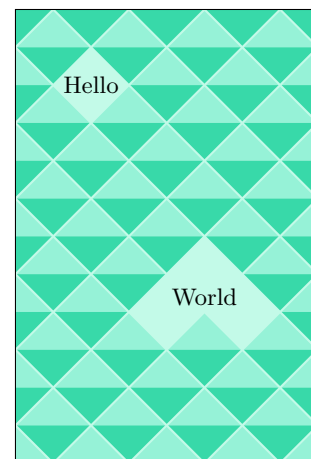
`/tikz/geomarray/draw-if-not-row-col={⟨coordinate list⟩}`

(initially unset)

N 2026-04-20

Restricts item drawing to those items whose coordinates are not contained in the specified comma-separated *⟨coordinate list⟩* of type *⟨row⟩/⟨col⟩*. The keys `/tikz/geomarray/init`^{P.34} and `/tikz/geomarray/init-item`^{P.35} are used to append the corresponding code.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  preaction = { fill = oc-teal-1 },
  fill gea Pfettrach = {
    scale fixed = 0.96, step half = 10mm,
    style 1 = { fill = oc-teal-2 },
    style 2 = { fill = oc-teal-4 },
    draw-if-not-row-col = { 4/3, 4/4, 5/3, 10/2 },
    freeze-row,
  }
]
(0,0) rectangle (4,6);
\node at (1,5) {Hello};
\node at (2.5,2.2) {World};
\end{tikzpicture}
```



<code>/tikz/geomarray/draw-if-row={⟨row list⟩}</code>	(initially unset)	<hr/> N 2026-04-21
<code>/tikz/geomarray/draw-if-not-row={⟨row list⟩}</code>	(initially unset)	
<code>/tikz/geomarray/draw-if-even-row</code>	(initially unset)	
<code>/tikz/geomarray/draw-if-odd-row</code>	(initially unset)	

Restricts item drawing to those items whose rows are

- contained in the specified comma-separated $\langle row list \rangle$,
- not contained in the specified comma-separated $\langle row list \rangle$,
- even,
- odd, respectively.

The keys `/tikz/geomarray/init→P.34` and `/tikz/geomarray/init-row→P.35` are used to append the corresponding code.

<code>/tikz/geomarray/draw-if-col={⟨column list⟩}</code>	(initially unset)	<hr/> N 2026-04-21
<code>/tikz/geomarray/draw-if-not-col={⟨column list⟩}</code>	(initially unset)	
<code>/tikz/geomarray/draw-if-even-col</code>	(initially unset)	
<code>/tikz/geomarray/draw-if-odd-col</code>	(initially unset)	

Restricts item drawing to those items whose columns are

- contained in the specified comma-separated $\langle column list \rangle$,
- not contained in the specified comma-separated $\langle column list \rangle$,
- even,
- odd, respectively.

The keys `/tikz/geomarray/init→P.34` and `/tikz/geomarray/init-item→P.35` are used to append the corresponding code.

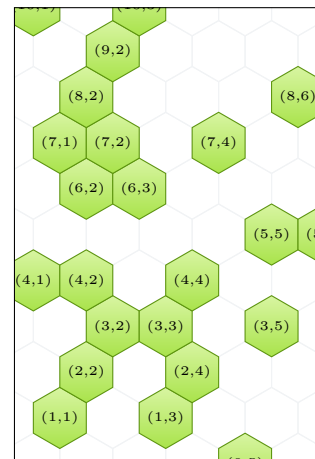
<code>/tikz/geomarray/draw-if-random=⟨probability⟩</code>	(initially unset)	<hr/> N 2026-04-17
---	-------------------	---------------------------

Restricts item drawing to random items with given $\langle probability \rangle \in [0,1]$. The keys `/tikz/geomarray/init→P.34` and `/tikz/geomarray/init-item→P.35` are used to append the corresponding code.

```

% \usepackage{opencolor}
\begin{tikzpicture}
\ExplSyntaxOn
\sys_gset_rand_seed:n {1701}
\ExplSyntaxOff
\draw[
  preaction = {
    fill gea Bruckberg = {
      scale fixed = 1, step equi = 7mm,
      style 1 = { draw = oc-gray-1 },
      freeze-row,
    },
  },
  fill gea Bruckberg = {
    scale fixed = 1, step equi = 7mm,
    style 1 = {
      draw = oc-lime-9,
      top color = oc-lime-2,
      bottom color = oc-lime-4 },
    debug text,
    draw-if-random = 0.33,
  }
]
(0,0) rectangle (4,6);
\end{tikzpicture}

```



```

/tikz/geomarray/node={\options}\{text}
/tikz/geomarray/node<={\options}\{text}
/tikz/geomarray/node>={\options}\{text}

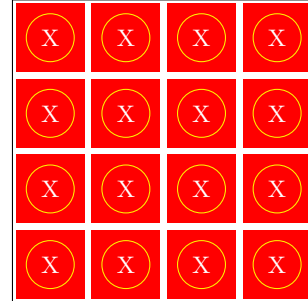
```

This is a shortcut for drawing a TikZ node applying `/tikz/geomarray/draw-item`^{→P.35}, `/tikz/geomarray/draw-item<`^{→P.35}, or `/tikz/geomarray/draw-item>`^{→P.35}, respectively. The node contents are set by `<text>` and any TikZ `<options>` for the node can be applied.

```

\begin{tikzpicture}
\draw[
  fill geomarray =
  {
    node< = {white,draw=yellow,circle}{X}
  },
]
(0,0) rectangle (4,4);
\end{tikzpicture}

```



```

/tikz/geomarray/text={text}
/tikz/geomarray/text<={text}
/tikz/geomarray/text>={text}

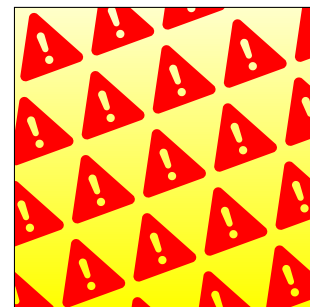
```

This is a shortcut for setting `<text>` with `\pgftext` applying `/tikz/geomarray/draw-item`^{→P.35}, `/tikz/geomarray/draw-item<`^{→P.35}, or `/tikz/geomarray/draw-item>`^{→P.35}, respectively. In contrast to `/tikz/geomarray/node`, there are no drawing options, but `/tikz/geomarray/text` is much faster.

```

% \usepackage{fontawesome7}
\begin{tikzpicture}
\draw[
  preaction = {top color=yellow!20,bottom color=yellow},
  fill geomarray =
  {
    triangular,
    text = {\Huge\color{red}\faTriangleExclamation},
    angle = 20,
  },
]
(0,0) rectangle (4,4);
\end{tikzpicture}

```



```

/tikz/geomarray/style 1={\langle path style \rangle} (initially fill=red)
/tikz/geomarray/style 2={\langle path style \rangle} (initially fill=green)
/tikz/geomarray/style 3={\langle path style \rangle} (initially fill=blue)
/tikz/geomarray/style 4={\langle path style \rangle} (initially fill=cyan)
/tikz/geomarray/style 5={\langle path style \rangle} (initially fill=magenta)
/tikz/geomarray/style 6={\langle path style \rangle} (initially fill=yellow)
/tikz/geomarray/style 7={\langle path style \rangle} (initially fill=lightgray)
/tikz/geomarray/style 8={\langle path style \rangle} (initially fill=orange)
/tikz/geomarray/style 9={\langle path style \rangle} (initially fill=purple)
/tikz/geomarray/style 10={\langle path style \rangle} (initially fill=lime)
/tikz/geomarray/style 11={\langle path style \rangle} (initially fill=olive)
/tikz/geomarray/style 12={\langle path style \rangle} (initially fill=pink)
/tikz/geomarray/style 13={\langle path style \rangle} (initially fill=teal)
/tikz/geomarray/style 14={\langle path style \rangle} (initially fill=violet)
/tikz/geomarray/style base={\langle path style \rangle} (initially empty)

```

These convenience options generate styles named

```

/tikz/fill-gea-style-1, /tikz/fill-gea-style-2,
/tikz/fill-gea-style-3, /tikz/fill-gea-style-4,
/tikz/fill-gea-style-5, /tikz/fill-gea-style-6,
/tikz/fill-gea-style-7, /tikz/fill-gea-style-8,
/tikz/fill-gea-style-9, /tikz/fill-gea-style-10,
/tikz/fill-gea-style-11, /tikz/fill-gea-style-12,
/tikz/fill-gea-style-13, /tikz/fill-gea-style-14,
and /tikz/fill-gea-style-base.

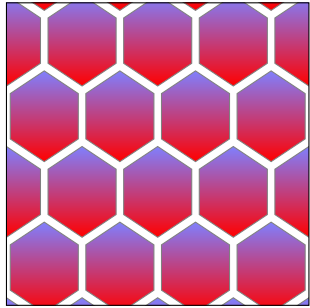
```

Styles of this kind are used within templates (see Section 5.4) for drawing. The style `/tikz/fill-gea-style-1` is also used in the default configuration. Modifying these styles is the simplest way to customize the appearance.

```

\begin{tikzpicture}
\draw[fill=gea Bruckberg =
{
style 1 =
{
draw = gray,
top color = blue!50,
bottom color = red
},
}]
(0,0) rectangle (4,4);
\end{tikzpicture}

```



```

/tikz/geomarray/prime color 1=\langle color \rangle (initially red)
/tikz/geomarray/prime color 2=\langle color \rangle (initially blue)

```

These convenience options assign the specified `\langle color \rangle` to the L3 token lists `\l_tikzfill_gea_prime_col_i_tl` and `\l_tikzfill_gea_prime_col_ii_tl`. The `\langle color \rangle` must be a valid `xcolor` color expression.

Some templates (e.g. `/tikz/fill gea Marktstauden`^{→ P. 56}) use these settings for drawing.

```

/tikz/geomarray/ratio=\langle fp expr \rangle (initially 1)

```

This convenience option assigns the evaluated floating point expression `\langle fp expr \rangle` to the L3 token list `\l_tikzfill_gea_ratio_tl`.

Some templates (e.g. `/tikz/fill gea Niederhatzkofen`^{→ P. 55}) use this setting for drawing.

`/tikz/geomarray/debug text=<color>` (default black, initially black) N 2026-03-18
`/tikz/geomarray/debug node={<options>}` (default black, initially black) N 2026-04-14

Prints the row and column numbers in the specified `<color>` or with the specified `<options>`. Note that this option appends drawing code; therefore, it should be specified after other drawing settings. It is intended for debugging or for identifying a specific item.

During item drawing, the following values are available (read-only):

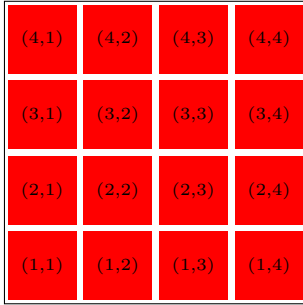
- `\l_tikzfill_gea_rows_int`: total number of rows minus one.
- `\l_tikzfill_gea_cols_int`: total number of columns minus one.
- `\l_tikzfill_gea_row_tl`: current row index, with $0 \leq \l_tikzfill_gea_row_tl \leq \l_tikzfill_gea_rows_int$.
- `\l_tikzfill_gea_col_tl`: current column index, with $0 \leq \l_tikzfill_gea_col_tl \leq \l_tikzfill_gea_cols_int$.

`/tikz/geomarray/debug text` displays the current pair consisting of `\l_tikzfill_gea_row_tl` and `\l_tikzfill_gea_col_tl`.

Note that at least the first and last row or column may lie partially or entirely outside the clipping area.

`/tikz/geomarray/debug text` is based on `/tikz/geomarray/text`^{→P.38}, while `/tikz/geomarray/debug node` is based on `/tikz/geomarray/node`^{→P.38}.

```
\begin{tikzpicture}
\draw[
  fill geomarray = { debug text },
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/tikz/geomarray/freeze-row` (experimental) N 2026-03-18

This is an experimental compile-time optimization for geometric arrays in which each row contains uniform items. For each row, an item is drawn only once into a box (“frozen”), and this box is then copied to fill the row.

Depending on the complexity of the graphical code, this can reduce compilation time. However, note the restriction that all items within a row must be uniform. Also note that applying `/tikz/geomarray/freeze-row` may shift the geometric array if the center of an item’s enclosing box is not identical to its origin.

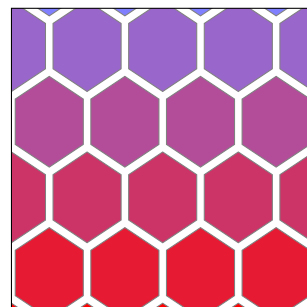
5.3 Auxiliary Macros

\TFRowColorLet{*<name>*}{*<color1>*}{*<color2>*}

N 2026-03-18

Creates a new color with given *<name>* which mixes the colors *<color1>* and *<color2>*. On the first row, *<color1>* is used, on the last row *<color2>*, and a linear mixture in between.

```
\begin{tikzpicture}
\draw[
  fill gea Bruckberg = {
    init-row< =
      \TFRowColorLet{mycolor}{red}{blue!50!white},
    style 1 = {draw=gray, fill=mycolor},
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```

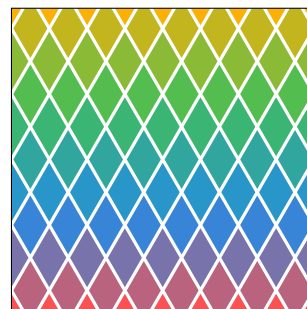


\TFRowColorLetSeq{*<name>*}{*<comma-list>*}

N 2026-03-18

Creates a new color with given *<name>* which mixes the colors given by the *<comma-list>* from the first row (first color) to the last row (last color) and a linear mixture in between.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Essenbach = {
    init-row< = \TFRowColorLetSeq{mycolor}
      {oc-red-6,oc-blue-6,oc-green-6,oc-yellow-6},
    style 1 = {fill=mycolor},
    step equi = 5mm,
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```

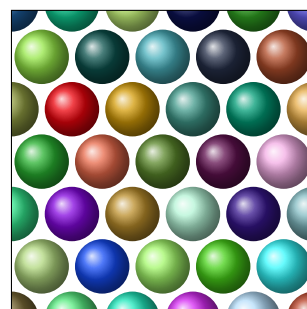


\TFDefineRandomColor{*<name>*}

N 2026-03-18

Defines a random color with given *<name>*. Random numbers are taken from the L3 programming layer.

```
\ExplSyntaxOn
\sys_gset_rand_seed:n {1701}
\ExplSyntaxOff
\begin{tikzpicture}
\draw[
  fill gea Furth = {
    triangular,
    step equi = 8mm,
    init-item< = \TFDefineRandomColor{mycolor},
    style 1 = {ball color=mycolor},
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



5.4 Templates

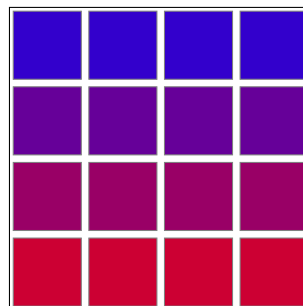
`/tikz/fill gea Altdorf={\langle options \rangle}`

(initially unset)

N 2026-03-18

The basic item shape of this template is a rectangle. `/tikz/geomarray/style 1→P.39` is used for graph drawing.

```
\begin{tikzpicture}
\draw[
  fill gea Altdorf = {
    init-row< =
      \TFRowColorLet{mycolor}{red}{blue},
    style 1 =
      {draw=gray, fill=mycolor},
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



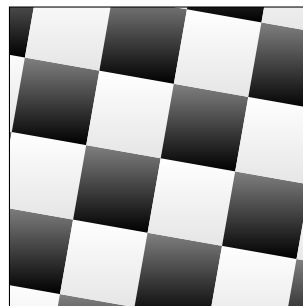
`/tikz/fill gea Loeschenbrand={\langle options \rangle}`

(initially unset)

N 2026-03-20

This is a variation of `/tikz/fill gea Altdorf` where `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used in a checkerboard pattern.

```
\begin{tikzpicture}
\draw[
  fill gea Loeschenbrand = {
    scale fixed = 1,
    style 1 =
      {bottom color=black, top color=black!50},
    style 2 =
      {bottom color=black!10, top color=white},
    angle = -10,
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



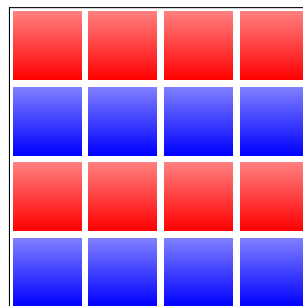
`/tikz/fill gea Ganslberg={\langle options \rangle}`

(initially unset)

N 2026-03-18

This is a variation of `/tikz/fill gea Altdorf` in which `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for odd and even rows, respectively.

```
\begin{tikzpicture}
\draw[
  fill gea Ganslberg = {
    style 1 =
      {bottom color=red, top color=red!50},
    style 2 =
      {bottom color=blue, top color=blue!50},
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



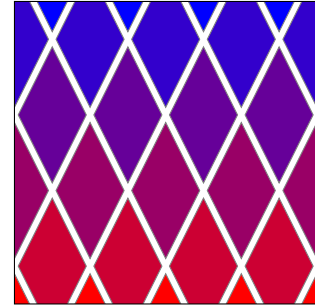
`/tikz/fill gea Essenbach={\langle options \rangle}`

(initially unset)

N 2026-03-18

The basic item shape of this template is a rhombus. `/tikz/geomarray/style 1→P.39` is used for graph drawing.

```
\begin{tikzpicture}
\draw[
  fill gea Essenbach = {
    init-row< =
      \TFRowColorLet{mycolor}{red}{blue},
    style 1 =
      {draw=gray, fill=mycolor},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



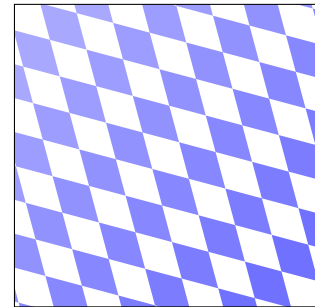
`/tikz/fill gea Mirskofen={\langle options \rangle}`

(initially unset)

N 2026-03-18

This is a variation of `/tikz/fill gea Essenbach` in which `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for odd and even rows, respectively.

```
\begin{tikzpicture}
\draw[
  fill gea Mirskofen = {
    scale fixed = 1,
    step equi = 5mm,
    init-row< =
      \TFRowColorLet{mycolor}{blue!60}{blue!30},
    style 1 = {fill=mycolor},
    style 2 = {},
    angle = 45,
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



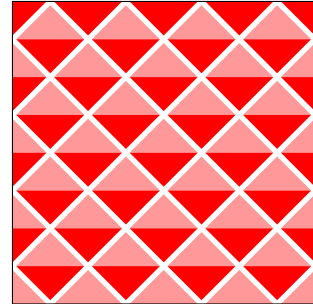
`/tikz/fill gea Pfettrach={\langle options \rangle}`

(initially unset)

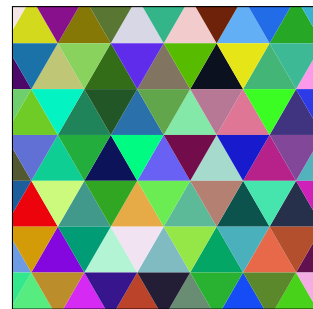
N 2026-03-18

The basic item shape of this template is a rhombus divided into two parts. `/tikz/geomarray/style 1P.39` and `/tikz/geomarray/style 2P.39` are used for the two parts.

```
\begin{tikzpicture}
\draw[
  fill gea Pfettrach = {
    step half = 10mm,
    style 1 = { fill = red!40 },
    style 2 = { fill = red },
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



```
\ExplSyntaxOn
\sys_gset_rand_seed:n {1701}
\ExplSyntaxOff
\begin{tikzpicture}
\draw[
  fill gea Pfettrach = {
    scale fixed = 1,
    step equi = 7mm,
    init-item< = \TFDefineRandomColor{col1}%
      \TFDefineRandomColor{col2},
    style 1 = { fill = col1 },
    style 2 = { fill = col2 },
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



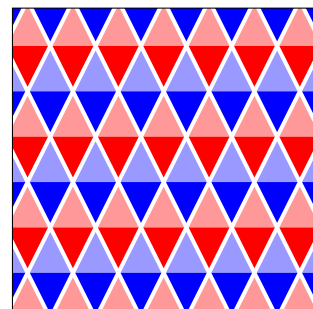
`/tikz/fill gea Kolmhub={\langle options \rangle}`

(initially unset)

N 2026-03-18

This is a variation of `/tikz/fill gea Pfettrach` in which `/tikz/geomarray/style 1P.39`, `/tikz/geomarray/style 2P.39`, `/tikz/geomarray/style 3P.39`, and `/tikz/geomarray/style 4P.39`, are used for odd and even rows, respectively.

```
\begin{tikzpicture}
\draw[
  fill gea Kolmhub = {
    step = 6mm,
    style 1 = { fill = red!40 },
    style 2 = { fill = red },
    style 3 = { fill = blue!40 },
    style 4 = { fill = blue },
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```

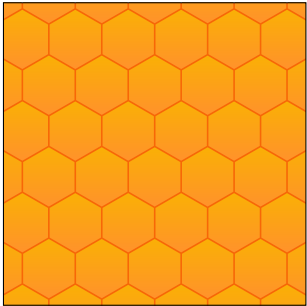


The basic item shape of this template is a hexagon. `/tikz/geomarray/style 1`^{P.39} is used for graph drawing.

```

% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  preaction = {fill=oc-orange-7},
  fill gea Bruckberg = {
    scale fixed = 0.97,
    step equi = 7mm,
    style 1 = {
      top color = oc-yellow-6,
      bottom color = oc-orange-5 },
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}

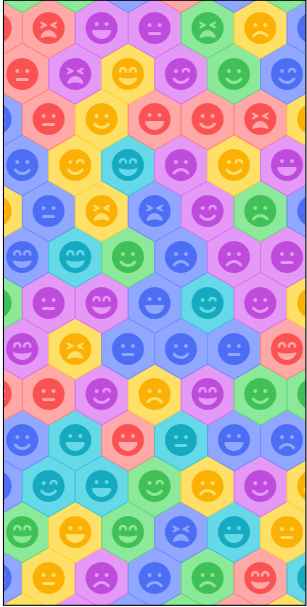
```



```

% \usepackage{opencolor}
% \usepackage{fontawesome7}
\ExplSyntaxOn
\begin{tikzpicture}
\draw[
  preaction = {fill=oc-orange-7},
  fill~gea~Bruckberg = {
    scale~fixed = 1,
    step~equi = 7mm,
    init< = \sys_gset_rand_seed:n {101},
    init~item< = \tl_set:Ne \l_tmpa_tl
      {oc-\clist_rand_item:n {red,grape,indigo,cyan,
        green,yellow} },
    style~1 = {draw=\l_tmpa_tl-4,fill=\l_tmpa_tl-3},
    text< = {
      \color{\l_tmpa_tl-6}\large
      \faIcon{face-\clist_rand_item:n {smile,frown,
        grin-beam,meh,tired,laugh,smile-wink}}
    },
  ]
  (0,0) rectangle (4,8);
\end{tikzpicture}
\ExplSyntaxOff

```



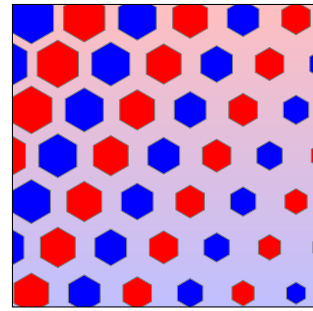
`/tikz/fill gea Bachhorn={\langle options \rangle}`

(initially unset)

N 2026-03-18

This is a variation of `/tikz/fill gea Bruckberg`^{P.45} in which `/tikz/geomarray/style 1`^{P.39} and `/tikz/geomarray/style 2`^{P.39} are used for odd and even rows, respectively.

```
\begin{tikzpicture}
\draw[
  preaction = {top color=red!25,bottom color=blue!25},
  fill gea Bachhorn = {
    scale = {0/0.3,1/0.9},
    step equi = 7mm,
    style 1 = { draw=gray, fill=red },
    style 2 = { draw=gray, fill=blue },
    angle = 60,
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



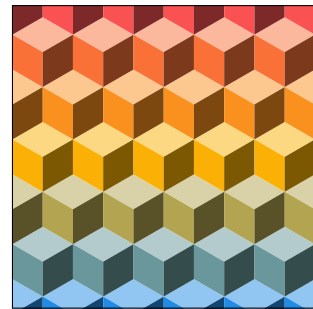
`/tikz/fill gea Weihmichl={\langle options \rangle}`

(initially unset)

N 2026-03-18

The basic item shape of this template is a hexagon divided into three parts. `/tikz/geomarray/style 1`^{P.39}, `/tikz/geomarray/style 2`^{P.39}, and `/tikz/geomarray/style 3`^{P.39}, are used for the three parts.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Weihmichl = {
    scale fixed = 1,
    step equi = 8mm,
    init-row< = \TFRowColorLetSeq{mycolor}{
      oc-blue-6,oc-yellow-6,oc-red-6},
    style 1 = {fill=mycolor!50!white},
    style 2 = {fill=mycolor},
    style 3 = {fill=mycolor!50!black},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



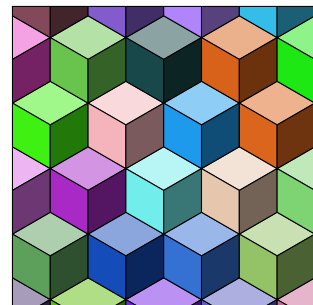
`/tikz/fill gea Edenland={\langle options \rangle}`

(initially unset)

N 2026-04-20

This is a variation of `/tikz/fill gea Weihmichl` in which a prime color, `/tikz/geomarray/prime color 1`^{P.39}, defines the individual colors of the three parts. The key `/tikz/geomarray/style base`^{P.39} specifies a style applied to all parts.

```
\ExplSyntaxOn
\sys_gset_rand_seed:n {12345}
\ExplSyntaxOff
\begin{tikzpicture}
\draw[ fill gea Edenland = {
  scale fixed = 1,
  step equi = 10mm,
  ystep equidistance,
  style base = draw,
  init-item< = \TFDefineRandomColor{mycolor},
  prime color 1 = mycolor,
}]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



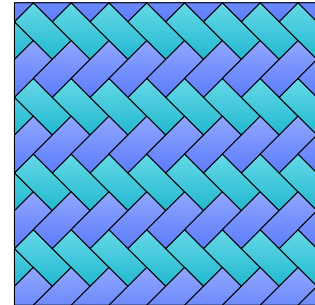
`/tikz/fill gea Baumgarten={\langle options \rangle}`

(initially unset)

N 2026-04-21

This template consists of two quadrilaterals arranged in an L-shape. `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for the two parts.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Baumgarten = {
    scale fixed = 1,
    step double = 5mm,
    style 1 = { draw, top color=oc-cyan-3,
               bottom color=oc-cyan-5 },
    style 2 = { draw, top color=oc-indigo-3,
               bottom color=oc-indigo-5 },
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



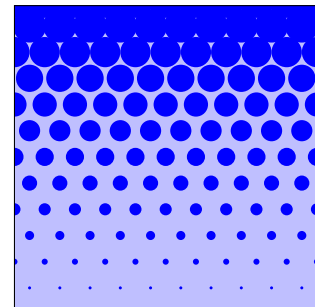
`/tikz/fill gea Furth={\langle options \rangle}`

(initially unset)

N 2026-03-18

The basic item shape of this template is a circle. `/tikz/geomarray/style 1→P.39` is used for graph drawing.

```
\begin{tikzpicture}
\draw[
  preaction = {fill=blue!25},
  fill gea Furth = {
    triangular,
    step equi = 4mm,
    scale = {0/0,1/1.2},
    style 1 = { fill=blue },
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



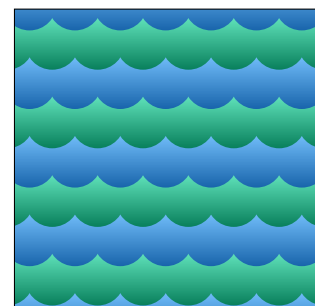
`/tikz/fill gea Schatzhofen={\langle options \rangle}`

(initially unset)

N 2026-03-18

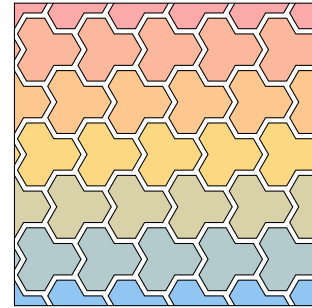
This is a variation of `/tikz/fill gea Furth` in which `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for odd and even rows, respectively.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Schatzhofen = {
    triangular,
    scale fixed=1.2,
    step equi = 6mm,
    style 1 = { top color=oc-blue-3,
               bottom color=oc-blue-9},
    style 2 = { top color=oc-teal-3,
               bottom color=oc-teal-9},
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



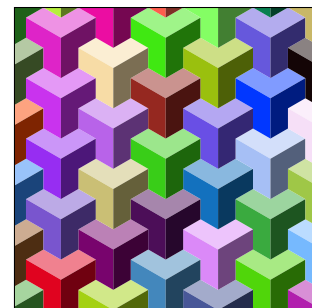
The basic item shape of this template is a tripod. `/tikz/geomarray/style 1→P.39` is used for graph drawing.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Ergolding = {
    step equi = 8mm,
    init-row< = \TFRowColorLetSeq{mycolor}{
      oc-blue-6,oc-yellow-6,oc-red-6},
    style 1 = {draw,fill=mycolor!50!white},
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



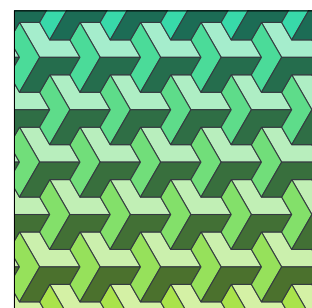
The basic item shape of this template is a tripod divided into three parts. `/tikz/geomarray/style 1→P.39`, `/tikz/geomarray/style 2→P.39`, and `/tikz/geomarray/style 3→P.39`, are used for the three parts.

```
\ExplSyntaxOn
\sys_gset_rand_seed:n {42}
\ExplSyntaxOff
\begin{tikzpicture}
\draw[
  fill gea Hascherkeller = {
    scale fixed = 1,
    step equi = 8mm,
    init-item< = \TFDefineRandomColor{mycolor},
    style 1 = {fill=mycolor!50!white},
    style 2 = {fill=mycolor},
    style 3 = {fill=mycolor!50!black},
    angle=30,
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



This is a variation of `/tikz/fill gea Hascherkeller` in which a prime color, `/tikz/geomarray/prime color 1→P.39`, defines the individual colors of the three parts. The key `/tikz/geomarray/style base→P.39` specifies a style applied to all parts.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Klosterholz = {
    scale fixed = 1,
    step equi = 8mm,
    init-row< = \TFRowColorLet{mycolor}
      {oc-lime-4}{oc-teal-4},
    prime color 1 = mycolor,
    style base = {draw=oc-gray-8,line join=round},
    freeze-row,
  }]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



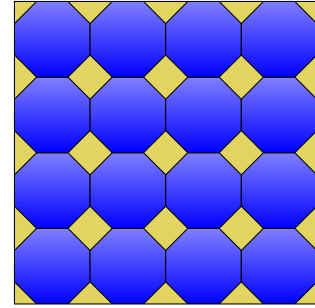
`/tikz/fill gea Unterglaim={\langle options \rangle}`

(initially unset)

N 2026-03-24

The basic item shape of this template is an octagon. `/tikz/geomarray/style 1→P.39` is used for graph drawing.

```
\begin{tikzpicture}
\draw[
  preaction = {fill=yellow!70!gray},
  fill gea Unterglaim = {
    scale fixed = 1,
    style 1 =
      {draw, top color=blue!50,bottom color=blue},
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



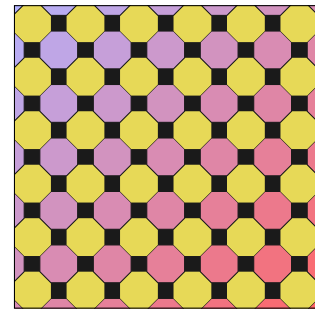
`/tikz/fill gea Grandsberg={\langle options \rangle}`

(initially unset)

N 2026-03-20

This is a variation of `/tikz/fill gea Unterglaim` where `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used in a checkerboard pattern.

```
\begin{tikzpicture}
\draw[
  preaction = {fill=black!90},
  fill gea Grandsberg = {
    scale fixed = 1,
    step = 5mm,
    init-row < =
      \TFRowColorLet{mycolor}{red!60}{blue!30},
    style 1 = {draw=black!90,fill=mycolor},
    style 2 = {fill=yellow!75!gray},
    angle = 45,
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



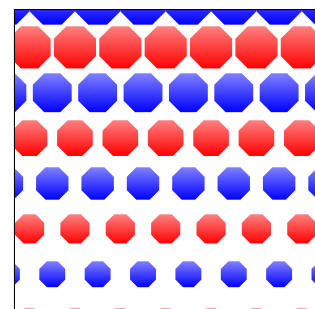
`/tikz/fill gea Kottingrohr={\langle options \rangle}`

(initially unset)

N 2026-03-18

This is a variation of `/tikz/fill gea Unterglaim` in which `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for odd and even rows, respectively.

```
\begin{tikzpicture}
\draw[
  fill gea Kottingrohr = {
    triangular,
    scale linear = {0.5}{1},
    step = 6mm,
    style 1 = {bottom color=red, top color=red!50},
    style 2 = {bottom color=blue, top color=blue!50},
  ]
(0,0) rectangle (4,4);
\end{tikzpicture}
```

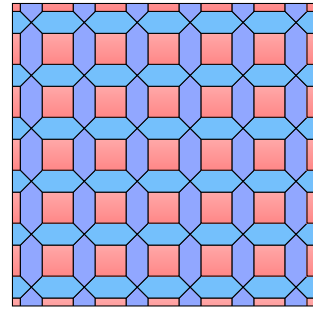


This template consists of a rectangle with two attached hexagons. An optical illusion creates the impression of an octagon.

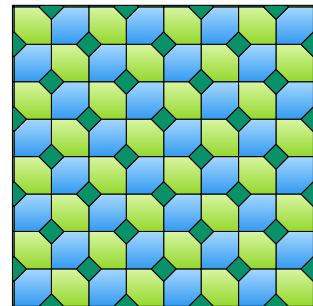
`/tikz/geomarray/style 1→P.39`, `/tikz/geomarray/style 2→P.39`, and `/tikz/geomarray/style 3→P.39` are used for the three parts.

The ratio between the inner rectangle and the (theoretical) base rectangle is controlled by `/tikz/geomarray/ratio→P.39`, which defaults to $2 - \sqrt{2}$. For equal values of `/tikz/geomarray/xstep→P.28` and `/tikz/geomarray/ystep→P.28`, this default produces the impression of a regular octagon.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Ostergaden = {
    scale fixed = 1,
    step equi = 7mm,
    style 1 = { draw, top color=oc-red-3,
               bottom color=oc-red-4 },
    style 2 = { draw, fill=oc-indigo-3 },
    style 3 = { draw, fill=oc-blue-3 },
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Ostergaden = {
    ratio = 1/3,
    scale fixed = 1,
    step equi = 7mm,
    style 1 = { draw, fill=oc-teal-8 },
    style 2 = { draw, top color=oc-lime-2,
               bottom color=oc-lime-5 },
    style 3 = { draw, top color=oc-blue-2,
               bottom color=oc-blue-5 },
    angle = 45,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



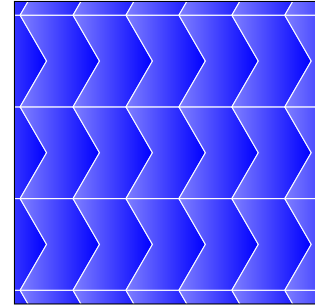
`/tikz/fill gea Hohenthann={\langle options \rangle}`

(initially unset)

N 2026-03-25

The basic item shape of this template is an arrow. `/tikz/geomarray/style 1→P.39` is used for graph drawing.

```
\begin{tikzpicture}
\draw[
  fill gea Hohenthann = {
    scale fixed = 1,
    step equi* double = 7mm,
    style 1 = {draw=white,
      left color=blue!50,right color=blue},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



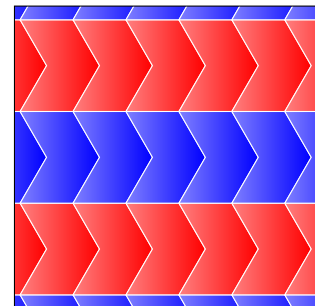
`/tikz/fill gea Bibelsbach={\langle options \rangle}`

(initially unset)

N 2026-03-25

This is a variation of `/tikz/fill gea Hohenthann` in which `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for odd and even rows, respectively.

```
\begin{tikzpicture}
\draw[
  fill gea Bibelsbach = {
    scale fixed = 1,
    step equi* double = 7mm,
    style 1 = {draw=white,
      left color=blue!50,right color=blue},
    style 2 = {draw=white,
      left color=red!50,right color=red},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



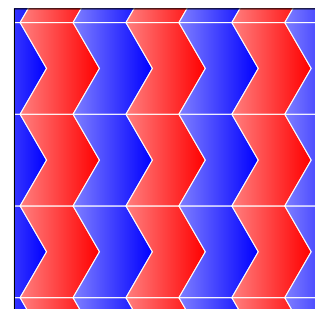
`/tikz/fill gea Roseneck={\langle options \rangle}`

(initially unset)

N 2026-03-25

This is a variation of `/tikz/fill gea Hohenthann` in which `/tikz/geomarray/style 1→P.39` and `/tikz/geomarray/style 2→P.39` are used for odd and even columns, respectively.

```
\begin{tikzpicture}
\draw[
  fill gea Roseneck = {
    scale fixed = 1,
    step equi* double = 7mm,
    style 1 = {draw=white,
      left color=blue!50,right color=blue},
    style 2 = {draw=white,
      left color=red!50,right color=red},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



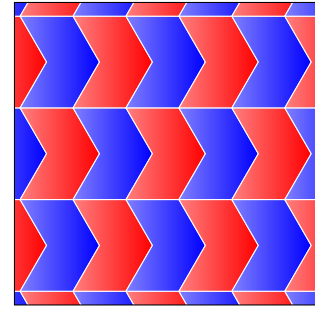
`/tikz/fill gea Altenkofen={\langle options \rangle}`

(initially unset)

N 2026-03-25

This is a variation of `/tikz/fill gea Hohenthann`^{→P.51} in which `/tikz/geomarray/style 1`^{→P.39} and `/tikz/geomarray/style 2`^{→P.39} are used in a checkerboard pattern.

```
\begin{tikzpicture}
\draw[
  fill gea Altenkofen = {
    scale fixed = 1,
    step equi* double = 7mm,
    style 1 = {draw=white,
      left color=blue!50,right color=blue},
    style 2 = {draw=white,
      left color=red!50,right color=red},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



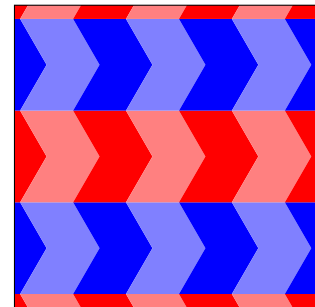
`/tikz/fill gea Irlmuehle={\langle options \rangle}`

(initially unset)

N 2026-03-25

This is a variation of `/tikz/fill gea Hohenthann`^{→P.51} in which `/tikz/geomarray/style 1`^{→P.39}, `/tikz/geomarray/style 2`^{→P.39}, `/tikz/geomarray/style 3`^{→P.39}, and `/tikz/geomarray/style 4`^{→P.39} are used.

```
\begin{tikzpicture}
\draw[
  fill gea Irlmuehle = {
    scale fixed = 1,
    step equi* double = 7mm,
    style 1 = {fill=red},
    style 2 = {fill=red!50},
    style 3 = {fill=blue},
    style 4 = {fill=blue!50},
  }]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



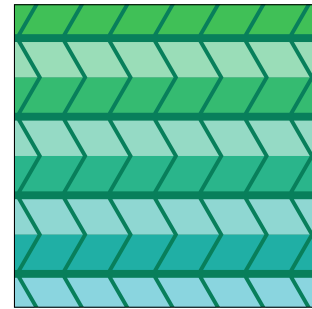
`/tikz/fill gea Gammelsdorf={\langle options \rangle}`

(initially unset)

N 2026-03-25

The basic item shape of this template is an arrow divided into two parts. `/tikz/geomarray/style 1P.39` and `/tikz/geomarray/style 2P.39` are used for the two parts.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  preaction = {fill=oc-teal-9},
  fill gea Gammelsdorf = {
    step equi* double = 6mm,
    init-row< =
      \TFRowColorLet{mycolor}{oc-cyan-6}{oc-green-6},
    style 1 = {fill=mycolor},
    style 2 = {fill=mycolor!50},
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



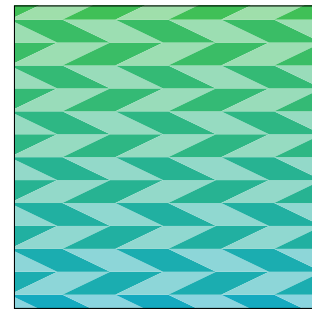
`/tikz/fill gea Traich={\langle options \rangle}`

(initially unset)

N 2026-03-25

This is a variation of `/tikz/fill gea Gammelsdorf` in which `/tikz/geomarray/style 1P.39` and `/tikz/geomarray/style 2P.39` are used in a checkerboard pattern.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Traich = {
    scale fixed = 1,
    step equi* = 7mm,
    init-row< =
      \TFRowColorLet{mycolor}{oc-cyan-6}{oc-green-6},
    style 1 = {fill=mycolor},
    style 2 = {fill=mycolor!50},
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



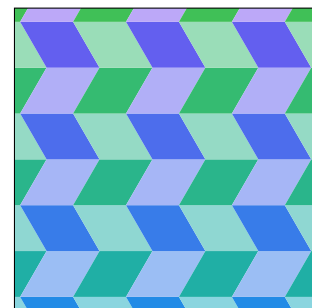
`/tikz/fill gea Winbuerg={\langle options \rangle}`

(initially unset)

N 2026-03-25

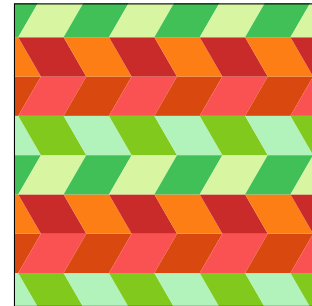
This is a variation of `/tikz/fill gea Gammelsdorf` in which `/tikz/geomarray/style 1P.39`, `/tikz/geomarray/style 2P.39`, `/tikz/geomarray/style 3P.39`, and `/tikz/geomarray/style 4P.39` are used.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Winbuerg = {
    scale fixed = 1,
    step equi* double = 7mm,
    init-row< =
      \TFRowColorLet{mycolA}{oc-cyan-6}{oc-green-6}%
      \TFRowColorLet{mycolB}{oc-blue-6}{oc-violet-6},
    style 1 = {fill=mycolA},
    style 2 = {fill=mycolA!50},
    style 3 = {fill=mycolB!50},
    style 4 = {fill=mycolB},
  ]
  (0,0) rectangle (4,4);
\end{tikzpicture}
```



This is a variation of `/tikz/fill gea Gammelsdorf`^{P.53} in which `/tikz/geomarray/style 1`^{P.39} up to `/tikz/geomarray/style 8`^{P.39} are used.

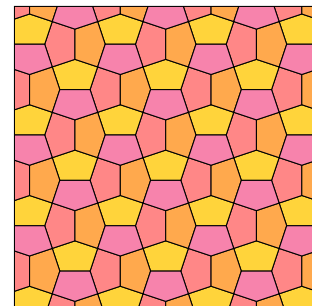
```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Kreuzholzen = {
    scale fixed = 1,
    step equi* double = 6mm,
    style 1 = {fill=oc-green-6},
    style 2 = {fill=oc-green-2},
    style 3 = {fill=oc-lime-2},
    style 4 = {fill=oc-lime-6},
    style 5 = {fill=oc-orange-9},
    style 6 = {fill=oc-orange-6},
    style 7 = {fill=oc-red-6},
    style 8 = {fill=oc-red-9},
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



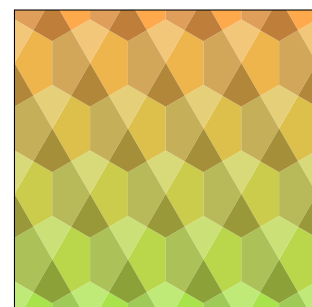
The basic item shape of this template is a hexagon divided into four pentagons (or two pentagons and two tetragons). `/tikz/geomarray/style 1`^{P.39}, `/tikz/geomarray/style 2`^{P.39}, `/tikz/geomarray/style 3`^{P.39}, and `/tikz/geomarray/style 4`^{P.39} are used for the four parts.

The optimal congruence of the parts is found with `/tikz/geomarray/step equi* half`^{P.30}, but other relations between `xstep` and `ystep` also produce pleasing results.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Grafenhaun = {
    step equi* half = 12mm,
    scale fixed = 1,
    style 1 = {draw,fill=oc-red-4},
    style 2 = {draw,fill=oc-yellow-4},
    style 3 = {draw,fill=oc-orange-4},
    style 4 = {draw,fill=oc-pink-4},
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Grafenhaun = {
    step equi = 10mm,
    scale fixed = 1,
    init-row< =
      \TFRowColorLet{mycolor}{oc-lime-4}{oc-orange-4},
    style 1 = {fill=mycolor},
    style 2 = {fill=mycolor!75!white},
    style 3 = {fill=mycolor!75!gray},
    style 4 = {fill=mycolor!75!black},
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



`/tikz/fill gea Niederhatzkofen={\langle options \rangle}`

(initially unset)

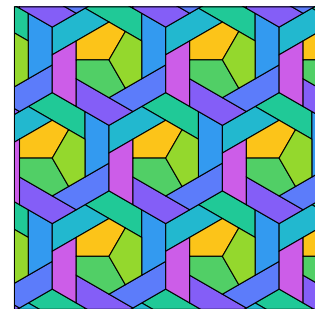
N 2026-04-17

The basic item shape of this template is a hexagon divided into nine parts, three of which form an inner hexagon surrounded by the remaining six parts. The styles `/tikz/geomarray/style 1`^{P.39} to `/tikz/geomarray/style 9`^{P.39} are used for these nine parts.

The ratio between the inner and outer hexagon is controlled by `/tikz/geomarray/ratio`^{P.39}, which defaults to $\frac{1}{\sqrt{3}}$. Notable ratio values include:

- $\frac{1}{\sqrt{3}}$: all nine parts have equal area.
- $\frac{1}{3}$: intersections are equally spaced.
- $\frac{1}{2}$: triangular structures become visible.
- $\frac{2}{3}$: three hexagonal structures become visible.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Niederhatzkofen = {
    scale fixed = 1,
    step equi = 15mm,
    style 1 = {draw, fill=oc-teal-5},
    style 2 = {draw, fill=oc-cyan-5},
    style 3 = {draw, fill=oc-grape-5},
    style 4 = {draw, fill=oc-violet-5},
    style 5 = {draw, fill=oc-indigo-5},
    style 6 = {draw, fill=oc-blue-5},
    style 7 = {draw, fill=oc-yellow-5},
    style 8 = {draw, fill=oc-green-5},
    style 9 = {draw, fill=oc-lime-5},
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



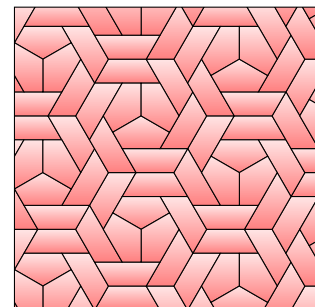
`/tikz/fill gea Eggerach={\langle options \rangle}`

(initially unset)

N 2026-04-23

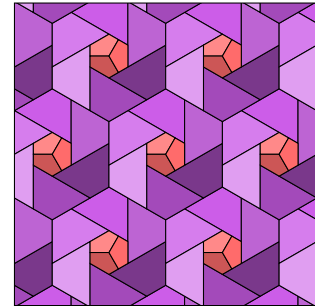
This is a variation of `/tikz/fill gea Niederhatzkofen` in which `/tikz/geomarray/style base`^{P.39} specifies a style applied to all nine parts.

```
\begin{tikzpicture}
\draw[
  fill gea Eggerach = {
    scale fixed = 1,
    step equi = 15mm,
    xflip even,
    style base = {draw, top color=red!10,
      bottom color=red!50},
    angle = 30,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```

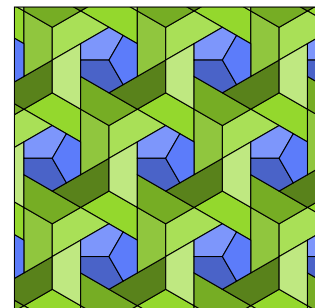


This is a variation of `/tikz/fill gea Niederhatzkofen`^{→P.55} in which two prime colors, `/tikz/geomarray/prime color 1`^{→P.39} and `/tikz/geomarray/prime color 2`^{→P.39}, define the individual colors of the six outer and three inner parts. The key `/tikz/geomarray/style base`^{→P.39} specifies a style applied to all nine parts.

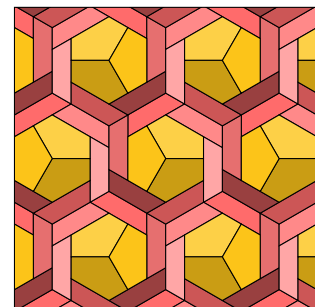
```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Marktstauden = {
    scale fixed = 1,
    ratio = 1/3,
    step equi = 15mm,
    style base = { draw },
    prime color 1 = oc-grape-5,
    prime color 2 = oc-red-5,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Marktstauden = {
    scale fixed = 1,
    ratio = 0.5,
    step equi = 15mm,
    style base = { draw },
    prime color 1 = oc-lime-5,
    prime color 2 = oc-indigo-5,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Marktstauden = {
    scale fixed = 1,
    ratio = 2/3,
    xflip even,
    step equi = 15mm,
    style base = { draw },
    prime color 1 = oc-red-5,
    prime color 2 = oc-yellow-5,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}
```



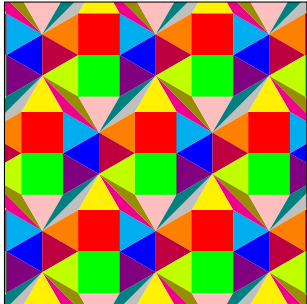
The basic item shape of this template is a hexagon divided into fourteen parts. An optical illusion creates the impression of a decagon. The styles `/tikz/geomarray/style 1`^{P.39} to `/tikz/geomarray/style 14`^{P.39} are used for these fourteen parts.

The option `/tikz/geomarray/triangular`^{P.28} is applied by default. The key `/tikz/geomarray/step equi`^{P.30} produces squares and several equilateral triangles. When the specialized key `/tikz/geomarray/step equi special` is used, the six vertices of the hexagon lie on a circle.

```

\begin{tikzpicture}
\draw[
  fill gea Alpersdorf = {
    scale fixed = 1,
    step equi = 15mm,
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}

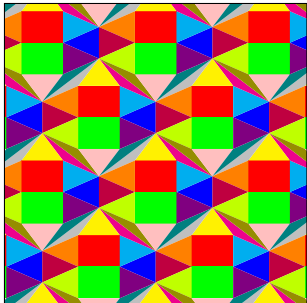
```



```

\begin{tikzpicture}
\draw[
  fill gea Alpersdorf = {
    scale fixed = 1,
    step equi special = 15mm,
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}

```

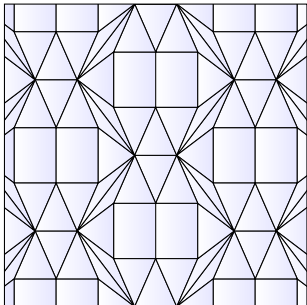


This is a variation of `/tikz/fill gea Alpersdorf` in which `/tikz/geomarray/style base`^{P.39} specifies a style applied to all fourteen parts.

```

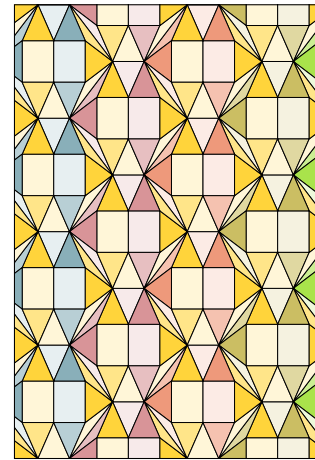
\begin{tikzpicture}
\draw[
  fill gea Schwarzberg = {
    scale fixed = 1,
    step equi special = 20mm,
    style base = {draw, top color=white,
      bottom color=blue!10},
    angle = 90,
    freeze-row,
  }
]
(0,0) rectangle (4,4);
\end{tikzpicture}

```



This is a variation of `/tikz/fill gea Alpersdorf`^{→P.57} in which two prime colors, `/tikz/geomarray/prime color 1`^{→P.39} and `/tikz/geomarray/prime color 2`^{→P.39}, define the individual colors of the fourteen parts. The key `/tikz/geomarray/style base`^{→P.39} specifies a style applied to all parts.

```
% \usepackage{opencolor}
\begin{tikzpicture}
\draw[
  fill gea Scheckenhofen = {
    scale fixed = 1,
    step equi special = 15mm,
    init-row< = \TFRowColorLetSeq{mycolor}
      {oc-lime-4,oc-red-4,oc-cyan-4},
    prime color 1 = mycolor,
    prime color 2 = oc-yellow-4,
    style base = { draw=black, line width=0.2pt },
    angle = 90,
    freeze-row,
  }
]
(0,0) rectangle (4,6);
\end{tikzpicture}
```



Index

`angle` key, 16, 19, 21, 24, 31

`band` key, 17, 22, 26

Commands

`\l_tikzfill_gea_col_tl`, 35, 40
`\l_tikzfill_gea_cols_int`, 34, 40
`\l_tikzfill_gea_prime_col_i_tl`, 39
`\l_tikzfill_gea_prime_col_ii_tl`, 39
`\l_tikzfill_gea_ratio_tl`, 39
`\l_tikzfill_gea_row_frac_fp`, 32, 35
`\l_tikzfill_gea_row_tl`, 35, 40
`\l_tikzfill_gea_rows_int`, 34, 40
`\l_tikzfill_gea_scale_fp`, 32, 35
`\l_tikzfill_gea_xstep_dim`, 28, 34
`\l_tikzfill_gea_ystep_dim`, 28, 34
`\TFDefineRandomColor`, 41
`\TFRowColorLet`, 41
`\TFRowColorLetSeq`, 41

`debug node` key, 40

`debug text` key, 40

`draw` key, 15, 24

`draw-if-col` key, 37

`draw-if-even-col` key, 37

`draw-if-even-row` key, 37

`draw-if-not-col` key, 37

`draw-if-not-row` key, 37

`draw-if-not-row-col` key, 36

`draw-if-odd-col` key, 37

`draw-if-odd-row` key, 37

`draw-if-random` key, 37

`draw-if-row` key, 37

`draw-if-row-col` key, 36

`draw-item` key, 35

`draw-item<` key, 35

`draw-item>` key, 35

`fill` key, 15, 24

`fill gea Alpersdorf` key, 57

`fill gea Altdorf` key, 42

`fill gea Altenkofen` key, 52

`fill gea Bachhorn` key, 46

`fill gea Baumgarten` key, 47

`fill gea Bibelsbach` key, 51

`fill gea Bruckberg` key, 45

`fill gea Edenland` key, 46

`fill gea Eggerach` key, 55

`fill gea Ergolding` key, 48

`fill gea Essenbach` key, 43

`fill gea Furth` key, 47

`fill gea Gammelsdorf` key, 53

`fill gea Ganslberg` key, 42

`fill gea Grafenhaun` key, 54

`fill gea Grandsberg` key, 49

`fill gea Hascherkeller` key, 48

`fill gea Hohenthann` key, 51

`fill gea Irlmuehle` key, 52

`fill gea Klosterholz` key, 48

`fill gea Kolmhub` key, 44

`fill gea Kottlingrohr` key, 49

`fill gea Kreuzholzen` key, 54

`fill gea Loeschenbrand` key, 42

`fill gea Marktstauden` key, 56

`fill gea Mirskofen` key, 43

`fill gea Niederhatzkofen` key, 55

`fill gea Ostergaden` key, 50

`fill gea Pfettrach` key, 44

`fill gea Roseneck` key, 51

`fill gea Schatzhofen` key, 47

`fill gea Scheckenhofen` key, 58

`fill gea Schwarzberg` key, 57

`fill gea Traich` key, 53

`fill gea Unterglaim` key, 49

`fill gea Weihmichl` key, 46

`fill gea Winbuerg` key, 53

`fill geomarray` key, 27

`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

`fill-gea-style-1` key, 35, 39

`fill-gea-style-10` key, 39

`fill-gea-style-11` key, 39

`fill-gea-style-12` key, 39

`fill-gea-style-13` key, 39

`fill-gea-style-14` key, 39

`fill-gea-style-2` key, 39

`fill-gea-style-3` key, 39

`fill-gea-style-4` key, 39

`fill-gea-style-5` key, 39

`fill-gea-style-6` key, 39

`fill-gea-style-7` key, 39

`fill-gea-style-8` key, 39

`fill-gea-style-9` key, 35, 39

`fill-gea-style-base` key, 39

`freeze-row` key, 40

`gap` key, 22

`hexagon` value, 14

`hexagon cycle` value, 20

`hexagon grid` value, 18

`init` key, 34

`init-item` key, 35

`init-item<` key, 35

`init-item>` key, 35

`init-row` key, 35

`init-row<` key, 35

`init-row>` key, 35

`init<` key, 34

`init>` key, 34

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 gea Alpersdorf`, 57

`fill gea Altdorf`, 42

`fill gea Altenkofen`, 52

`fill gea Bachhorn`, 46

`fill gea Baumgarten`, 47

`fill gea Bibelsbach`, 51

`fill gea Bruckberg`, 45

`fill gea Edenland`, 46

`fill gea Eggerach`, 55

`fill gea Ergolding`, 48

`fill gea Essenbach`, 43

`fill gea Furth`, 47

`fill gea Gammelsdorf`, 53

`fill gea Ganslberg`, 42

`fill gea Grafenhaun`, 54

`fill gea Grandsberg`, 49

`fill gea Hascherkeller`, 48

`fill gea Hohenthann`, 51

`fill gea Irlmuehle`, 52

`fill gea Klosterholz`, 48

`fill gea Kolmhub`, 44

`fill gea Kottingrohr`, 49

`fill gea Kreuzholzen`, 54

`fill gea Loeschenbrand`, 42

`fill gea Marktstauden`, 56

`fill gea Mirskofen`, 43

`fill gea Niederhatzkofen`, 55

`fill gea Ostergaden`, 50

`fill gea Pfettrach`, 44

`fill gea Roseneck`, 51

`fill gea Schatzhofen`, 47

`fill gea Scheckenhofen`, 58

`fill gea Schwarzberg`, 57

`fill gea Traich`, 53

`fill gea Unterglaim`, 49

`fill gea Weihmichl`, 46

`fill gea Winbuerg`, 53

`fill geomarray`, 27

`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

`fill-gea-style-1`, 35, 39

`fill-gea-style-10`, 39

`fill-gea-style-11`, 39

`fill-gea-style-12`, 39

`fill-gea-style-13`, 39

`fill-gea-style-14`, 39

`fill-gea-style-2`, 39

`fill-gea-style-3`, 39

`fill-gea-style-4`, 39

`fill-gea-style-5`, 39

`fill-gea-style-6`, 39

`fill-gea-style-7`, 39

`fill-gea-style-8`, 39

`fill-gea-style-9`, 35, 39

`fill-gea-style-base`, 39

`pattern hexagon`, 14

`pattern hexagon cycle`, 20

`pattern hexagon grid`, 18

`pattern rhombus`, 23

`/tikz/geomarray/`

`angle`, 31

`debug node`, 40

`debug text`, 40

`draw-if-col`, 37

`draw-if-even-col`, 37

draw-if-even-row, 37
 draw-if-not-col, 37
 draw-if-not-row, 37
 draw-if-not-row-col, 36
 draw-if-odd-col, 37
 draw-if-odd-row, 37
 draw-if-random, 37
 draw-if-row, 37
 draw-if-row-col, 36
 draw-item, 35
 draw-item<, 35
 draw-item>, 35
 freeze-row, 40
 init, 34
 init-item, 35
 init-item<, 35
 init-item>, 35
 init-row, 35
 init-row<, 35
 init-row>, 35
 init<, 34
 init>, 34
 node, 38
 node<, 38
 node>, 38
 prime color 1, 39
 prime color 2, 39
 ratio, 39
 scale, 32
 scale fixed, 32
 scale linear, 32
 step, 29
 step double, 29
 step equi, 30
 step equi double, 31
 step equi half, 30
 step equi special, 57
 step equi*, 30
 step equi* double, 31
 step equi* half, 30
 step half, 29
 style 1, 39
 style 10, 39
 style 11, 39
 style 12, 39
 style 13, 39
 style 14, 39
 style 2, 39
 style 3, 39
 style 4, 39
 style 5, 39
 style 6, 39
 style 7, 39
 style 8, 39
 style 9, 39
 style base, 39
 text, 38
 text<, 38
 text>, 38
 triangular, 28
 xflip, 33
 xflip even, 33
 xflip odd, 33
 xstep, 28
 yflip, 33
 yflip even, 33
 yflip odd, 33
 ystep, 28
 ystep equidistance, 30
 ystep equidistance*, 30
 \l_tikzfill_gea_col_tl, 35, 40
 \l_tikzfill_gea_cols_int, 34, 40
 \l_tikzfill_gea_prime_col_i_tl, 39
 \l_tikzfill_gea_prime_col_ii_tl, 39
 \l_tikzfill_gea_ratio_tl, 39
 \l_tikzfill_gea_row_frac_fp, 32, 35
 \l_tikzfill_gea_row_tl, 35, 40
 \l_tikzfill_gea_rows_int, 34, 40
 \l_tikzfill_gea_scale_fp, 32, 35
 \l_tikzfill_gea_xstep_dim, 28, 34
 \l_tikzfill_gea_ystep_dim, 28, 34
 line width key, 15, 19, 24
 node key, 38
 node< key, 38
 node> key, 38
 pattern hexagon key, 14
 pattern hexagon cycle key, 20
 pattern hexagon grid key, 18
 pattern rhombus key, 23
 pos key, 16, 25
 prime color 1 key, 39
 prime color 2 key, 39
 ratio key, 25, 39
 rhombus value, 23
 rings key, 22
 scale key, 32
 scale fixed key, 32
 scale linear key, 32
 size key, 15, 18, 21, 23
 step key, 29
 step double key, 29
 step equi key, 30
 step equi double key, 31
 step equi half key, 30
 step equi special key, 57
 step equi* key, 30
 step equi* double key, 31
 step equi* half key, 30
 step half key, 29
 style 1 key, 39
 style 10 key, 39
 style 11 key, 39
 style 12 key, 39
 style 13 key, 39

`style 14` key, 39
`style 2` key, 39
`style 3` key, 39
`style 4` key, 39
`style 5` key, 39
`style 6` key, 39
`style 7` key, 39
`style 8` key, 39
`style 9` key, 39
`style base` key, 39

`text` key, 38
`text<` key, 38
`text>` key, 38
`\TFDefineRandomColor`, 41
`\TFRowColorLet`, 41
`\TFRowColorLetSeq`, 41
`triangular` key, 28

Values

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

`xflip` key, 33
`xflip even` key, 33
`xflip odd` key, 33
`xshift` key, 16, 19, 21, 25
`xstep` key, 28

`yflip` key, 33
`yflip even` key, 33
`yflip odd` key, 33
`yshift` key, 16, 19, 21, 25
`ystep` key, 28
`ystep equidistance` key, 30
`ystep equidistance*` key, 30