The Ornaments package [1]

[1] Inspired by F.Fradin (psvectorian)

Alain Matthes

March 1, 2012

(Version beta 0.1)

This document describes the LaTeX package ***pgfornament***. It also provides examples and comments on the package's use. Firstly, I would like to thank Till Tantau for the beautiful LaTeX package, namely **TikZ**. I am grateful to Vincent Le Moign for allowing me to distribute the ornaments [2] in the format **PGF/TikZ**. I would like to thank also Enrico Gregorio for some great ideas used in this package.

[2] http://www.vectorian.net/ (free sample)

Contents

How to install the package 5
How to use the package 5
The main macro 6
Number argument 6
Argument and options 6
Examples of the use of options 7
Examples of symmetry 8
Option **ydelta** 8
Option **color** 8
Style **pgfornamentstyle** 9
Advanced options from **TikZ** 9
What is a (pgf)ornament? 10
Placing a vector ornament on a page 11
On each page with the package eso-pic 11
On one page with the picture environment 11
With **TikZ**, *the options* **remember picture** *and* **overlay** 12
Placing a vector ornament in the flow 13
Directly 13
In the flow with **TikZ** 13
Ornament inside a node 14
One ornament between two nodes 15
How to use **to [ornament= ...]** 15
How to use the option **at** 16
How to use the option **options** 16

Ornaments : Vector Symbols 17

Symbols part 1 17

Symbols part 2 18

Ornaments : Vector Corners 19

Ornaments : Vector Lines 20

Application : Creating a frame 21

Application : Frame around a text 22

Application : text inside a frame 23

Application : other way to get a pentagon 24

Package **tikzrput** 25

Example with **\rput** 25

Ornament with **\rput** 25

Examples from psvectorian 26

Large Title – e01 26

Cover with frame – e02 26

Little Title – e03 27

Advanced usage 28

Look at the code 28

How to use the code differently 29

Define a symbol with Inskape 29

From .eps or .mps file 30

Index 31

List of Figures

1 Minimal code for vectorian ornaments 5
2 Result of the minimal code 5
3 Vectorian ornament n° 1 6
4 Vectorian ornament n° 2 6
5 am ornament n° 1 6
6 Vertical symmetry 8
7 Horizontal symmetry 8
8 Central symmetry 8
9 How to use tkznameydelta 8
10 How to use tkznamecolor 8
11 How to use the style **pgfornamentstyle** 9
12 How to add **TikZ**' styles 9
13 Minimal code to get an ornament 10
14 Assembling of ornaments version 2 13
15 Assembling of ornaments version 1 13
16 Style with node 14
17 Node in the flow 14
18 One ornament between two nodes 15
19 Two ornaments between two nodes 15
20 A pentagon 15
21 at 16
22 How options 16
23 Creating a frame 21
24 A poem 22
25 Text inside a frame with a tikzpicture's environment 23
26 A pentagon 24
27 Example with **\rput** 25
28 Placement with rput 25
29 ... 25
30 Example named e01 in psvectorian 26
31 Example named e02; See the 25 for a tikz's version 27
32 Example named e03 27
33 Usage of another family 29
34 Symbol from Inskape 30
35 Symbol from .eps file 30

List of Tables

1 List of options for the pgfornament macro. 6
2 List of symbols (part 1). 17
3 A list of symbols (part 2). 18
4 A list of corners 19
5 A list of lines. 20

How to install the package

With TeXLive, if you need to install it by yourself, a TDS compliant zip archive is provided (pgfornament.zip). Just download that file, and unpack it in your TDS directory (/texmf for Unix-like systems).

- pgfornament must to be in /texmf/tex/latex
- pgflibraryvectorian.code.tex must to be in /texmf/tex/latex
- pgflibraryam.code.tex must to be in /texmf/tex/latex
- the folder vectorian must to be in /texmf/tex/generic
- the folder am must to be in /texmf/tex/generic

With MiKTeX, copy folder pgfornament into C:\texmf\tex\latex, then run MiKTeX Options . In the File name database section, click on Refresh now.

How to use the package

You only need to add

```
\usepackage{ornament}
```

or

```
\usepackage[object=vectorian]{ornament}
```

in your preamble. The pgfornament package loads TikZ.

Without any options, ornament package uses the vectorian symbols. If you want to use other symbols, you give the name of the list of symbols like this :

```
\usepackage[object=am]{ornament}.
```

I create am to show you how to create new symbols and how to use it (see the section ??). You can see in the margin, the minimum code to get a vector ornament.

```
\documentclass{scrartcl}
\usepackage{pgfornament}
\begin{document}
\pgfornament[width = 2cm,
             color = red]{1}
\end{document}
```

Figure 1: Minimal code for vectorian ornaments

Figure 2: Result of the minimal code

The main macro

The macro `\pgfornament` draws the object linked to the given number, with the vectorian family this number is between 1 and 89. This macro can be used alone, or inside a picture . It's defined by an environment ***tikzpicture*** placed at the current point.

The objects displayed depend of the option used when `\pgfornament` is called. The specifications of the `\pgfornament` command is:

`\pgfornament[`⟨*options*⟩`]{`*number*`}`

The result is picture defined by a ***tikzpicture*** environment.

Number argument

The number designs an object of a list by a rank. With you get the figure 3

```
\usepackage{ornament}
...
\pgfornament[width=2cm]{1}
```

with

```
\usepackage{ornament}
...
\pgfornament[width=2cm]{2}
```

and with

```
\usepackage[object=am]{ornament}
 ...
\pgfornament[width=4cm]{1}
```

you get the figure

Figure 3: Vectorian ornament n° 1

you get

Figure 4: Vectorian ornament n° 2

you get the figure

Figure 5: am ornament n° 1

Argument and options

The macro has five options. You have four possibilities for the last option `symmetry`. The next table describes these options.

name	default	definition
`scale`	1	ratio of height to width is unchanged
`width`	{}	set the width, ratio unchanged
`height`	{}	set the height, ratio unchanged
`color`	black	color of the ornament
`ydelta`	0 pt	value to adjust vertically the ornament
`symmetry=v`	none	vertical symmetry
`symmetry=h`	none	horizontal symmetry
`symmetry=c`	none	central symmetry
`symmetry=none`	none	no symmetry by default

Table 1: List of options for the pgfornament macro.

Examples of the use of options

1. Option `scale`

```
\pgfornament[scale=0.25]{77}
```

2. Option `width`

```
\pgfornament[width=5cm]{77}
```

3. Option `height`

```
\pgfornament[height=2cm]{77}
```

4. Option `color`

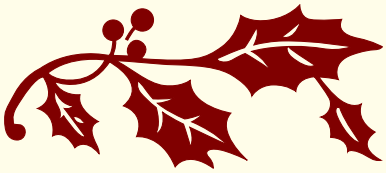
```
\pgfornament[width=5cm,color=green!20!black]{77}
```

5. Option `symmetry=h`

```
\pgfornament[width=5cm,symmetry=h]{77}
```

6. Option `symmetry=v`

```
\pgfornament[width=5cm,symmetry=v]{77}
```



7. Option `symmetry=c`

```
\pgfornament[width=4cm,symmetry=c]{77}
```

Examples of symmetry

1. Symmetry vertical axis

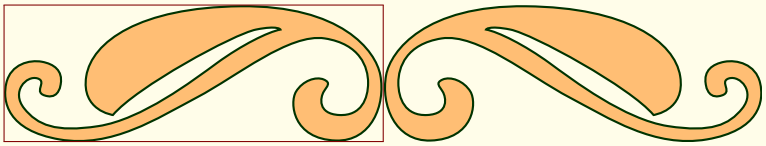


Figure 6: Vertical symmetry

2. Symmetry horizontal axis

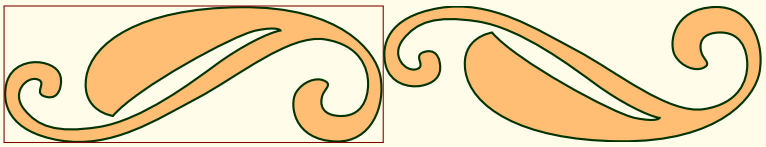
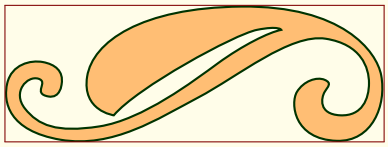


Figure 7: Horizontal symmetry

3. Symmetry with respect to the origin



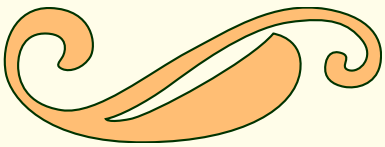


Figure 8: Central symmetry

Option `ydelta`

```
\pgfornament[color=MidnightBlue,width=2cm,ydelta=-10pt]{25}%
\pgfornament[color=PineGreen,width=2cm]{25}%
\pgfornament[color=Periwinkle,width=2cm,ydelta=+10pt]{25}%
```

Figure 9: How to use tkznameydelta

Option `color`

```
\pgfornament[color=MidnightBlue,width=2cm]{24}%
```

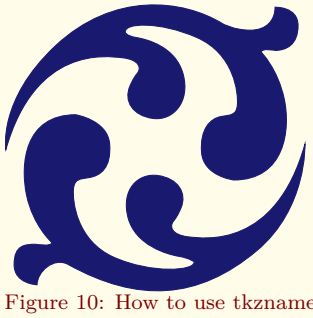


Figure 10: How to use tkznamecolor

Style `pgfornamentstyle`

This style can modify some options like the color and also how to fill the symbol when it's possible.

```
\begin{tikzpicture}
\tikzset{pgfornamentstyle/.style={
         fill=SpringGreen,
         fill opacity=.5,
         line width=1pt}}%
\pgfornament[color=OliveGreen,scale=2,anchor=south]{24}%
\end{tikzpicture}
```

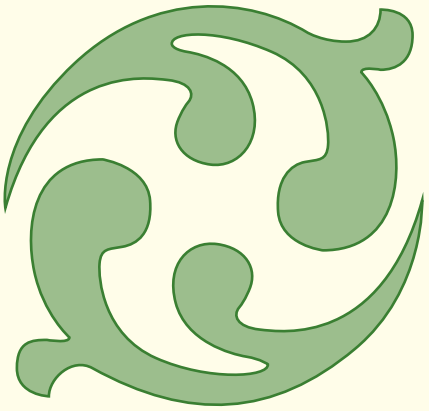


Figure 11: How to use the style `pgfornamentstyle`

Advanced options from `TikZ`

```
\begin{tikzpicture}
 \tikzset{pgfornamentstyle/.style={draw=Periwinkle,
             fill=SpringGreen}}%
 \node[draw=Periwinkle,circle,anchor=center,
      inner sep=0pt,fill=GreenYellow] at (0,0){%
 \pgfornament[anchor=center,scale=2]{24}};
\end{tikzpicture}
```

Figure 12: How to add `TikZ`' styles

What is a (pgf)ornament?

When you write in your document `\pgfornament{1}`, you get the first ornament of a family (by default `vectorian`'s family). This ornament is a vector object defined by an environment *`tikzpicture`*.

```
\begin{tikzpicture}[%
   baseline={([yshift=\pgfornamentydelta]%
             current bounding box.\pgfornamentanchor)},
   pgfornamentstyle]
   \pgftransformscale{\pgfornamentscale}%
   \pgf@@ornament{#2}%
\end{tikzpicture}%
```

You can modify the aspect of the picture if you change `\pgfornamentscale`, or `pgfornamentstyle`. With `\pgfornamentydelta`, or `\pgfornamentanchor` you can move the picture but this depends on the different environments. The next code gives the picture 13. I chose this method so that the use is as simple as possible.

```
\documentclass{scrartcl}
\usepackage{pgfornament}
\begin{document}
\pgfornament{1}
\end{document}
```

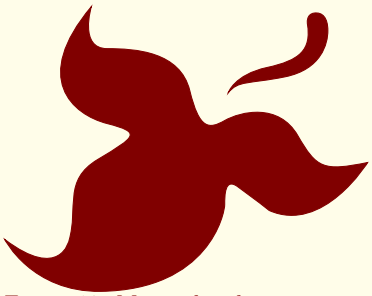
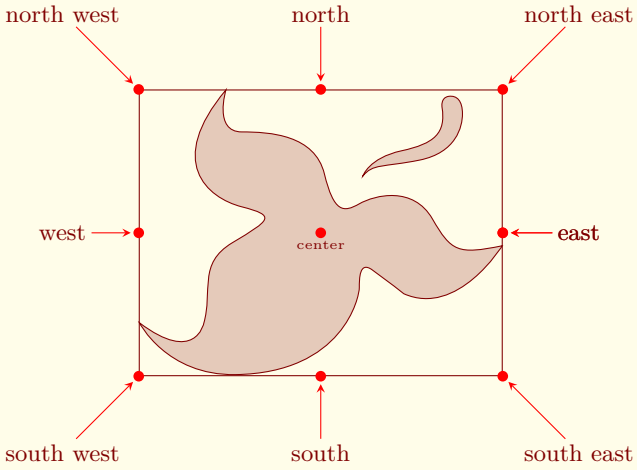


Figure 13: Minimal code to get an ornament

[3] You can find the dimensions of this shape in the file pgflibraryvectorian.code.tex. The name of this file depends of the name of the vector family By default actually it's `vectorian`.

The ornament is placed in a rectangle[3].



On the last figure, I represent all the anchors that you can use. Now you will see how to place this picture on a page, in the flow of text or inside a complex picture.

Placing a vector ornament on a page

On each page with the package `eso-pic`

You may have noticed the existence of an ornament placed at each corner of the pages. The next code explains how to do this. The only part of the code linked to `pgfornament` is to use the macro `\pgfornament`. To put the object at the right place on the page, we need to consider its width.

Perhaps you saw the ornaments in each corner of each page I used the package `eso-pic` and the next code. The macro `\put` places the ornament at a point but you need to change correctly the anchor.

```
\usepackage{eso-pic}
\makeatletter
\AddToShipoutPicture{%
\begingroup
\setlength{\@tempdima}{2mm}%
\setlength{\@tempdimb}{\paperwidth-\@tempdima-2cm}%
\setlength{\@tempdimc}{\paperheight-\@tempdima}%
\put(\LenToUnit{\@tempdima},\LenToUnit{\@tempdimc}){%
 \pgfornament[anchor=north west,width=2cm]{63}}
\put(\LenToUnit{\@tempdima},\LenToUnit{\@tempdima}){%
  \pgfornament[anchor=south west,width=2cm,symmetry=h]{63}}
\put(\LenToUnit{\@tempdimb},\LenToUnit{\@tempdimc}){%
  \pgfornament[anchor=north east,width=2cm,symmetry=v]{63}}
\put(\LenToUnit{\@tempdimb},\LenToUnit{\@tempdima}){%
  \pgfornament[anchor=south east,width=2cm,symmetry=c]{63}}
\endgroup
}
\makeatother
```

On one page with the picture environment

The next code is used to delimit the text area on the page defined by the tufte class. [4]

[4] `\strippt` is defined by `\let\strippt\strip@pt`

```
\newcommand{\eachpageornament}{%
\unitlength=1pt
\begin{picture}(0,0)%
\put(0,0){\pgfornament[width=1cm]{41}};%
\put(\strippt\textwidth,0){%
     \pgfornament[width=1cm,symmetry=v]{41}};%
\put(0,-\strippt\textheight){%
      \pgfornament[width=1cm,symmetry=h]{41}};%
\put(\strippt\textwidth,-\strippt\textheight){%
      \pgfornament[width=1cm,symmetry=c]{41}};%
\end{picture}}%

\eachpageornament
```

With **TikZ**, *the options* **remember picture** *and* **overlay**

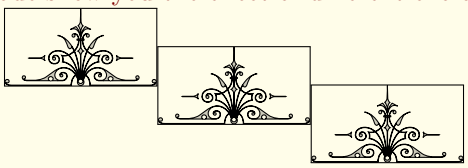
You can without **eso-pic** but with **TikZ** get the same result on one page with the next macro. **remember picture** is obligatory, this option tells **TikZ** that it should attempt to remember the position of the current picture on the page, you need to compile twice if you use such code. The option **overlay** switches the computation of the bounding box so the pictures are not in the flow of the text and they don't modify the layout.

```
\newcommand{\eachpageornament}{%
\begin{tikzpicture}[remember picture, overlay]
  \node[anchor=north west] at (current page.north west){%
                    \pgfornament[width=2cm]{63}};
  \node[anchor=north east] at (current page.north east){%
                    \pgfornament[width=2cm,symmetry=v]{63}};
  \node[anchor=south west] at (current page.south west){%
                   \pgfornament[width=2cm,symmetry=h]{63}};
  \node[anchor=south east] at (current page.south east){%
                    \pgfornament[width=2cm,symmetry=c]{63}};
 \end{tikzpicture}
 }
```

Placing a vector ornament in the flow

Directly

The next code show you the effect of different choice of anchor.

baseline  baseline

```
{ \color{black}baseline \pgfsetfillopacity{0.2}%
  \fbox{\pgfornament[anchor=south,width=2cm]{69}}%
  \fbox{\pgfornament[width=2cm]{69}}%
  \fbox{\pgfornament[anchor=north,width=2cm]{69}}%
  \pgfsetfillopacity{1} baseline }
```

Perhaps you are interesting by the code to modify the subsection?

```
\subsection{\protect\pgfornament[anchor=south,width=1cm]{78}\
    Directly \
    \protect\pgfornament[anchor=south,width=1cm,symmetry=v]{78}}
```

In the flow with `TikZ`

Generally, the best way is to place the ornament inside a node and the node inside an environment *tikzpicture*. You can need to specify the position of the node inside the `tikzpicture` and you can add an anchor to place exactly the ornament like you want.

```
\begin{tikzpicture}
   \foreach \a in {0,45,...,315}
     \node[anchor=west,rotate=\a,inner sep=0pt,xshift=12pt] {%
     \pgfornament[width=1cm]{88}};
\end{tikzpicture}
```

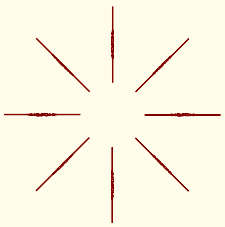


Figure 14: Assembling of ornaments version 2

```
\begin{tikzpicture}
   \foreach \a in {0,45,...,315}
     \node[anchor=west,rotate=\a,inner sep=0pt] {%
     \pgfornament[width=1cm]{88}};
\end{tikzpicture}
```

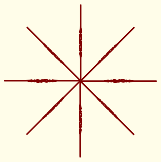
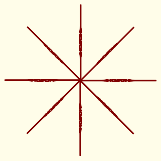


Figure 15: Assembling of ornaments version 1

Remark : It's difficult to get the same result with `\put` *and* `\rotatebox` *but it's easy with the* `rotating` *package.*

```
 \foreach \a in {0,45,...,315}{%
    \turnbox{\a}{\pgfornament[width=1cm]{88}}}%
```



Ornament inside a node

This method is very useful and flexible because it's possible to use the options and styles with the command `\node`. You can modify the style `pgfornamentstyle` [5].

[5] I you want to rest the style you can use `\resetpgfornamentstyle`

```
\tikzset{pgfornamentstyle/.style={%
  draw=green!20!black,inner sep=0pt,fill=orange,
  fill opacity=.5,scale=2,ultra thick}}%
  \tikz\node {\fbox{\pgfornament{3}}};
```

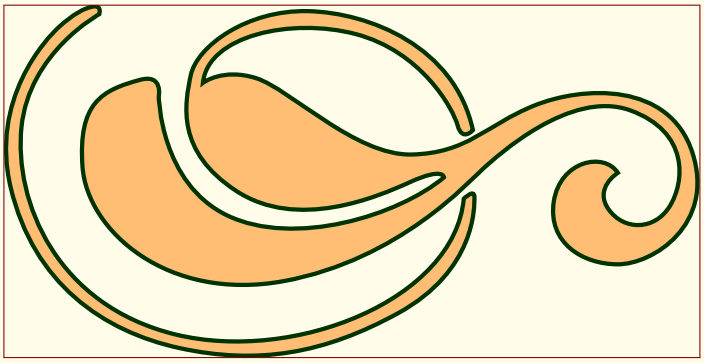


Figure 16: Style with node

☞ If we use a tikzpicture inside the flow then it's very useful to know how to place the picture. The important part of the code is :

```
\tikz[baseline=(current bounding box.south)]
```

☞ Don't forget to use **`inner sep =0pt`** because you can get undesirable space around the object.

```
baseline\tikz[baseline]
\node[inner sep=0pt]{\fbox{\pgfornament[width=2cm]{3}}};
baseline
\tikz[baseline=(current bounding box.south)]
\node[inner sep=0pt]{\fbox{\pgfornament[width=2cm]{3}}};
baseline
\tikz[baseline=(current bounding box.north)]
\node[inner sep=0pt]{\fbox{\pgfornament[width=2cm]{3}}};
baseline
```

Figure 17: Node in the flow

baseline baseline baseline baseline

One ornament between two nodes

I created an option for the **`to`** command. You only need to call an ornament with `ornament=number`.

```
\draw (A) to [object = ⟨number⟩] (B) ;
```

How to use `to [ornament= ...]`

This code shows how to place an ornament between to node. The width of the ornament is automatically calculate.

```
\begin{tikzpicture}
\node (A) at (0,0) {};
\node (B) at (5,2) {};
\draw [help lines,color=Maroon!60]  (0,0) grid (5,2);
\draw [fill=Maroon!30]  (A) circle (2pt) (B) circle (2pt);
\draw [orange] (A)  to [ornament=88]   (B);
\end{tikzpicture}
```

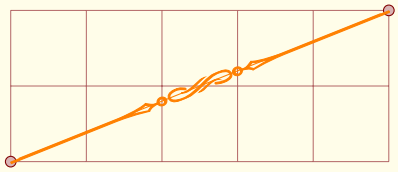


Figure 18: One ornament between two nodes

The next code shows how to place two ornaments between two nodes.

```
\begin{tikzpicture}
\node (A) at (0,0) {};
\node (B) at (5,2) {};
\draw [help lines,color=Maroon!60]  (0,0) grid (5,2);
\draw [fill=Maroon!30]  (A) circle (2pt) (B) circle (2pt);
\path (A)-(B) coordinate[pos=.5] (c1);
\draw [orange] (A)  to [ornament=84]
               (c1) to [ornament=84] (B);
\end{tikzpicture}
```

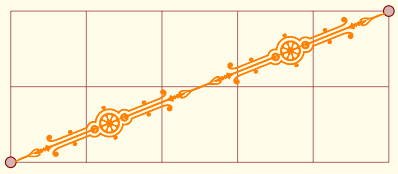


Figure 19: Two ornaments between two nodes

Example with a pentagon

```
\begin{tikzpicture}[every node={anchor=center,
                                inner sep=0pt}]
\node[regular polygon, regular polygon sides=5,
rotate=36,minimum size=6cm,inner sep=0pt](s)  {};
\path (s.side 1) to [ornament=83] (s.side 2)
                 to [ornament=83] (s.side 3)
                 to [ornament=83] (s.side 4)
                 to [ornament=83] (s.side 5)
                 to [ornament=83] (s.side 1);
\end{tikzpicture}
```

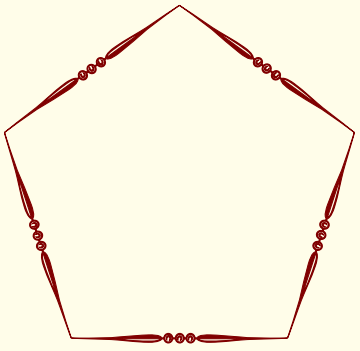


Figure 20: A pentagon

How to use the option **at**

It's possible to move the ornament on the line AB. You only need to write **at = number** where number is a percent like **pos**.

```
\begin{tikzpicture}
\node (A) at (0,0) {};
\node (B) at (4,0) {};
\draw [help lines,color=Maroon!60]  (0,-1) grid (4,1);
\path  (A.center)  to [ornament=84,at=0]   (B.center);
\path  (A.center)  to [ornament=84,at=1]   (B.center);
\end{tikzpicture}
```

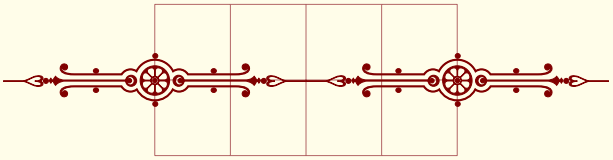


Figure 21: at

How to use the option ***options***

If an ornament is misplaced we can move it up or down. Look at the code to see how to use **options**.

```
\begin{tikzpicture}
\node (A) at (0,0) {};
\node (B) at (5,2) {};
\draw [help lines,color=Maroon!40]  (0,0) grid (5,2);
\draw [fill=Maroon!20]  (A) circle (2pt) (B) circle (2pt);
\path  (A.center) to [ornament=84,
       options/.append style={yshift=1pt}] (B.center);
\end{tikzpicture}
```

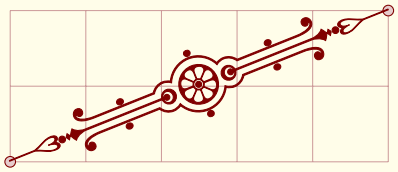


Figure 22: How options

Ornaments : Vector Symbols

Here a liste of the first thirty elements

Symbols part 1

1)		2)	
3)		4)	
5)		6)	
7)		8)	
9)		10)	
11)		12)	
13)		14)	
15)		16)	
17)		18)	
19)		20)	
21)		22)	
23)		24)	
25)		26)	
27)		28)	
29)		30)	

Table 2: List of symbols (part 1).

Symbols part 2

The next list is about symbols of decoration. The design is more sophisticated. Be careful indices range from sixty-five to seventy-nine.

65)

66)

67)

68)

69)

70)

71)

72)

73)

74)

75)

76)

77)

78)

79)

Table 3: A list of symbols (part 2).

Ornaments : Vector Corners

The next list of ornaments concerns objects to place in the corners of a figure. Half of them is not useful because it is obtained by symmetry of the other.

31)		32)	
33)		34)	
35)		36)	
37)		38)	
39)		40)	
41)		42)	
61)		62)	
63)		64)	

Table 4: A list of corners

Ornaments : Vector Lines

The last list concerns symbols used to make a line.

Table 5: A list of lines.

80)

81)

82)

83)

84)

85)

86)

87)

88)

89)

Application : Creating a frame

Remark : Corners are the same dimensions (widht = height)

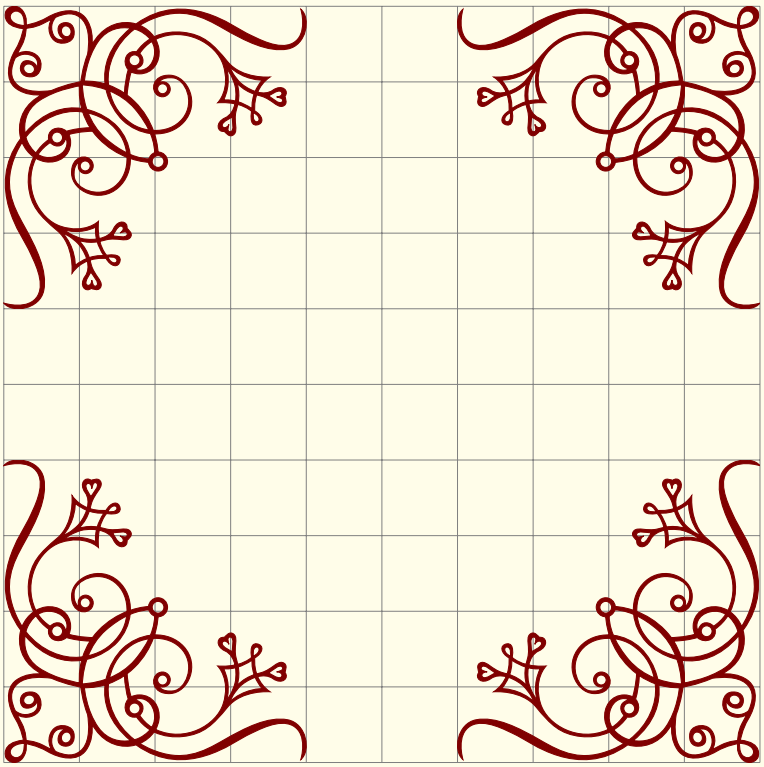


Figure 23: Creating a frame

```
\begin{tikzpicture}[color=Maroon,
                    every node/.style={inner sep=0pt}]
 \draw[help lines] (-6,-6) grid (6,6);
 \node[minimum size=12cm](vecbox){};
 \node[anchor=north west] at (vecbox.north west)
      {\pgfornament[width=5cm]{61}};
 \node[anchor=north east] at (vecbox.north east)
      {\pgfornament[width=5cm,symmetry=v]{61}};
 \node[anchor=south west] at (vecbox.south west)
      {\pgfornament[width=5cm,symmetry=h]{61}};
 \node[anchor=south east] at (vecbox.south east)
      {\pgfornament[width=5cm,symmetry=c]{61}};
\end{tikzpicture}
```

Application : Frame around a text

I chose a poem to illustrate this theme.

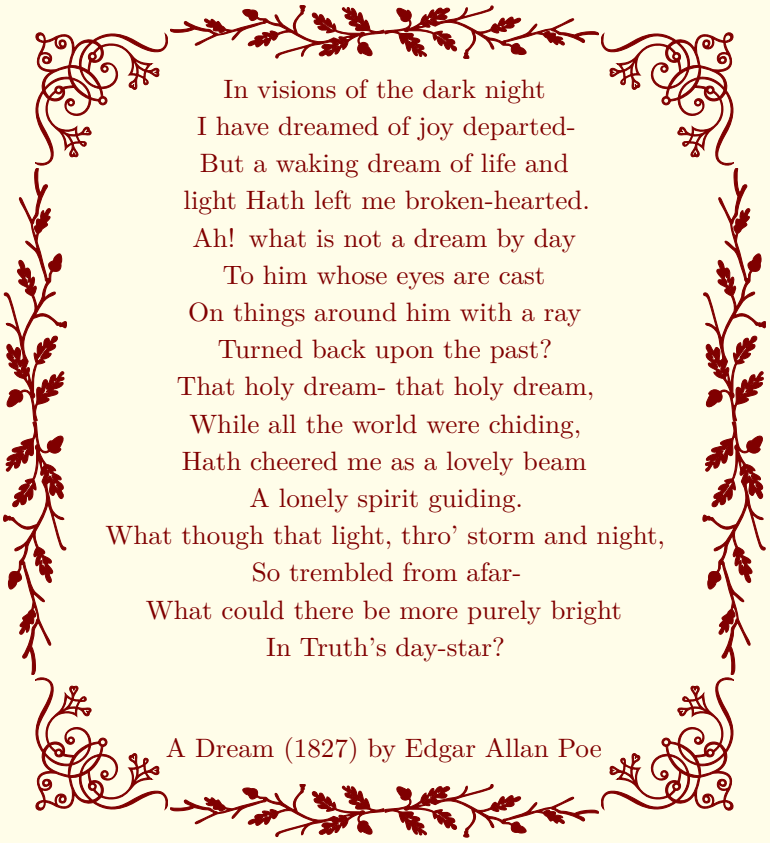


Figure 24: A poem

The poem is placed in a node named `Text`. Then we can place the corners relatively to four anchors of the node `Text`. Finally with the macros `\gfornamenthline` and `\gfornamentvline` it's possible to finish the frame.

```
\begin{tikzpicture}[every node/.style={inner sep=0pt}]
\node[text width=8cm,align=center](Text){%
    In visions of the dark night ...} ;
\node[shift={(-1cm,1cm)},anchor=north west](CNW)
at (Text.north west) {\pgfornament[width=1.75cm]{61}};
\node[shift={(1cm,1cm)},anchor=north east](CNE)
at (Text.north east) {\pgfornament[width=1.75cm,symmetry=v]{61}};
\node[shift={(-1cm,-1cm)},anchor=south west](CSW)
at (Text.south west) {\pgfornament[width=1.75cm,symmetry=h]{61}};
\node[shift={(1cm,-1cm)},anchor=south east](CSE)
at (Text.south east) {\pgfornament[width=1.75cm,symmetry=c]{61}};
\pgfornamenthline{CNW}{CNE}{north}{87}
\pgfornamenthline{CSW}{CSE}{south}{87}
\pgfornamentvline{CNW}{CSW}{west}{87}
\pgfornamentvline{CNE}{CSE}{east}{87}
\end{tikzpicture}
```

Application : text inside a frame

Firstly we build the frame with the help of nodes and the we place the text in a node relatively to others nodes.

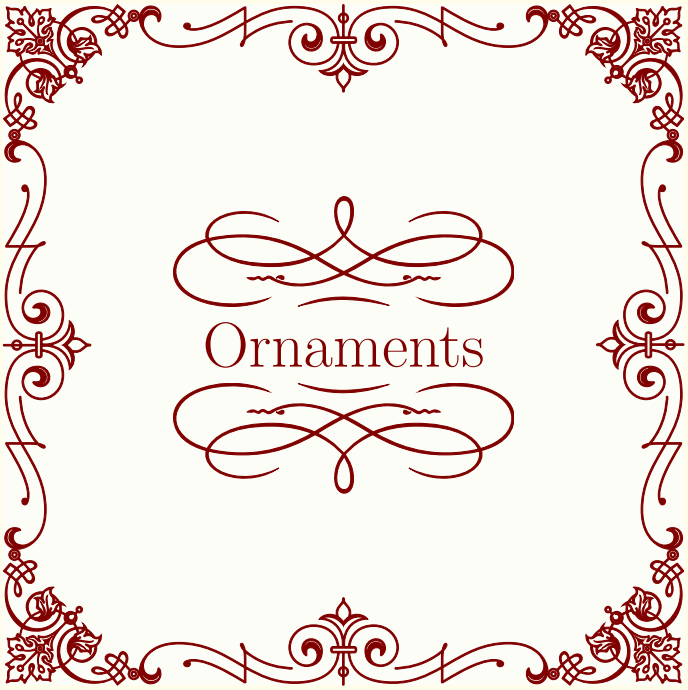


Figure 25: Text inside a frame with a tikzpicture's environment

```
\newcommand{\framesize}{9 cm}
\begin{tikzpicture}[color=Maroon,
                    transform shape,
                    every node/.style={inner sep=0pt}]
\node[minimum size=\framesize,fill=Beige!10](vecbox){};
\node[anchor=north west] at (vecbox.north west){%
      \pgfornament[width=0.2*\framesize]{63}};
\node[anchor=north east] at (vecbox.north east){%
      \pgfornament[width=0.2*\framesize,symmetry=v]{63}};
\node[anchor=south west] at (vecbox.south west){%
      \pgfornament[width=0.2*\framesize,symmetry=h]{63}};
\node[anchor=south east] at (vecbox.south east){%
      \pgfornament[width=0.2*\framesize,symmetry=c]{63}};
\node[anchor=north] at (vecbox.north){%
      \pgfornament[width=0.6*\framesize,symmetry=h]{46}};
\node[anchor=south] at (vecbox.south){%
      \pgfornament[width=0.6*\framesize]{46}};
\node[anchor=north,rotate=90]  at (vecbox.west){%
      \pgfornament[width=0.6*\framesize,symmetry=h]{46}};
\node[anchor=north,rotate=-90] at (vecbox.east){%
      \pgfornament[width=0.6*\framesize,symmetry=h]{46}};
\node[inner sep=6pt] (text) at (vecbox.center){\Huge Ornaments};
\node[anchor=north] at (text.south){%
      \pgfornament[width=0.5*\framesize]{75}};
\node[anchor=south] at (text.north){%
      \pgfornament[width=0.5*\framesize,symmetry=h]{75}};
\end{tikzpicture}
```

Application : other way to get a pentagon

We can place ornaments manually but the last method can also be used . [6]

[6] `\getornamentlength` is ...

```
\begin{tikzpicture}[every node={anchor=center,inner sep=0pt}]
    \node[regular polygon,
          regular polygon sides=5,
          minimum size=5cm,
          inner sep=0pt](s)  {};
    \getornamentlength{s}{corner 1}{s}{corner 2}
    \node[rotate=216] at (s.side 1)
         {\pgfornament[width=\ornamentlen]{88}};
    \node[rotate=288] at (s.side 2)
         {\pgfornament[width=\ornamentlen]{88}};
    \node[rotate=0]   at (s.side 3)
         {\pgfornament[width=\ornamentlen]{88}};
    \node[rotate=72]  at (s.side 4)
         {\pgfornament[width=\ornamentlen]{88}};
    \node[rotate=144] at (s.side 5)
         {\pgfornament[width=\ornamentlen]{88}};
\end{tikzpicture}
```

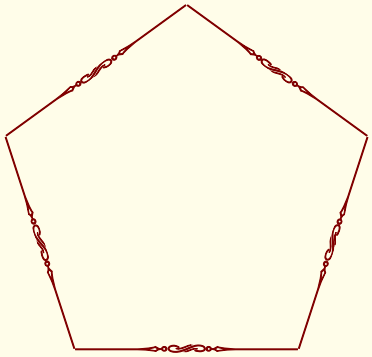


Figure 26: A pentagon

Package ***tikzrput***

Pstricks Users are accustomed to placing objects with **\rput**, so I created a package **tikzrput** with only one macro **\rput**. This macro is used as that of Pstricks with the same argument and options. The display of an object at the point (x,y) is realized with **\rput** of *pstricks* like this :

\rput[⟨*refpoint*⟩]{*angle*}(x,y){**\pgfornament**[⟨*options*⟩]{*number*}}

Example with **\rput**

```
\foreach \a in {0,4,...,356}{%
  \rput(\a;2){$\bullet$}%
  }
    \rput[B](0;0){Circle}%
```

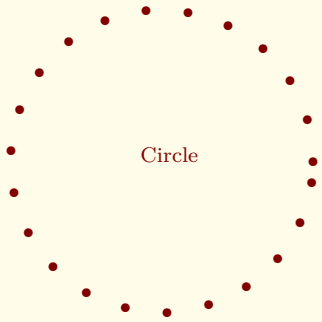


Figure 27: Example with **\rput**

Ornament with **\rput**

```
\begin{picture}(5,4)
 \rput(2,1){\pgfornament[width=2cm]{1}}
 \rput(4,2){\pgfornament[width=2cm]{2}}
\end{picture}
```

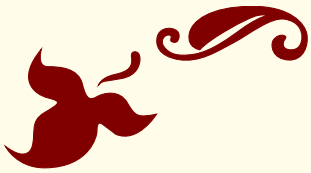


Figure 28: Placement with rput

Pour rappel,

```
\begin{tikzpicture}
   \draw[help lines] (0,0) grid (6,4);
   \draw [use as bounding box] (0,0) rectangle (6,4);
   \node[inner sep=0pt,] at (2,1){%
         \pgfornament[width=2cm,color=CadetBlue]{3}};
   \node[anchor=south,inner sep=0pt] at (4,2){%
         \pgfornament[color=CadetBlue,width=2cm]{3}};
\end{tikzpicture}
\caption{Placement with  nodes}
```

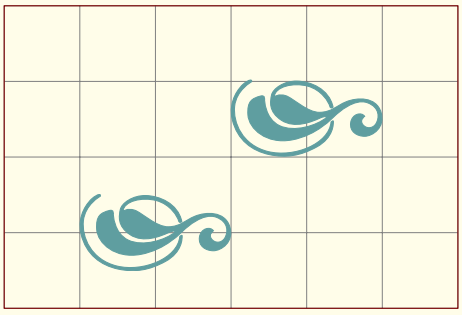


Figure 29: ...

Examples from psvectorian

Large Title – e01

This example is given here :
http://melusine.eu.org/syracuse/pstricks/vectorian/e01.tex . I use the macro `rput` from my package tikzrput to get the figure with the same code. I only replace `\psvectorian` by `\pgfornament`.

Figure 30: Example named e01 in psvectorian

```
\rput[r](-3pt,3pt){\pgfornament[scale=.35]{72}}
\large{Motifs d'ornements}%
\rput[l](3pt,3pt){\pgfornament[scale=.35]{73}}\\
\rput(0,0){\pgfornament[scale=.5]{85}}
```

Cover with frame – e02

This example is given here
http://melusine.eu.org/syracuse/pstricks/vectorian/e02.tex
I need `tikzpicture` and `\draw` to replace `pspicture` and `\psframe`.

```
\begin{tikzpicture}[color=blue]
 \draw[use as bounding box,thin] (-5,-5) rectangle (5,5);
 \node {\rput[tl](-3,5){\pgfornament[width=6cm]{71}}
 \rput[bl](-3,-5){\pgfornament[width=6cm„symmetry=h]{71}}
 %coins
 \rput[tl](-5,5){\pgfornament[width=2cm]{63}}
 \rput[tr](5,5){\pgfornament[width=2cm„symmetry=v]{63}}
 \rput[bl](-5,-5){\pgfornament[width=2cm„symmetry=h]{63}}
 \rput[br](5,-5){\pgfornament[width=2cm„symmetry=c]{63}}
 % côtés
 \rput[bl]{-90}(-5,3){\pgfornament[width=6cm]{46}}
 \rput[bl]{90}(5,-3){\pgfornament[width=6cm]{46}}
 %texte+soulignement+chapeau
 \rput(0,0){\Huge Ornaments}
 \rput[t](0,-0.5){\pgfornament[width=5cm]{75}}
 \rput[b](0,0.5){\pgfornament[width=5cm]{69}}
 % oiseaux
 \rput[tr]{-30}(-1,2.5){\pgfornament[width=2cm]{57}}
 \rput[tl]{30}(1,2.5){\pgfornament[width=2cm,symmetry=v]{57}}};
 \end{tikzpicture}
```

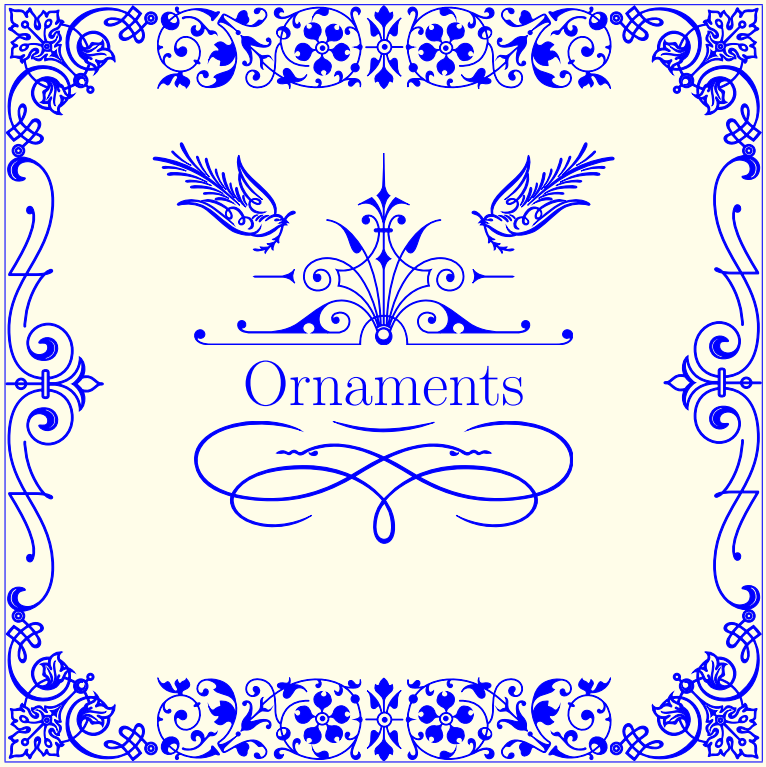


Figure 31: Example named e02; See the 25 for a tikz's version

Little Title – e03

This example is given here

http://melusine.eu.org/syracuse/pstricks/vectorian/e03.tex

I corrected a little problem with blank space around the text.

```
\begin{center}
 \rput[r](-2pt,6pt){\pgfornament[,height=1cm]{21}}
 {\Large Texte}%
 \rput[l](2pt,6pt){\pgfornament[height=1cm]{23}}
\end{center}
```

Figure 32: Example named e03

Advanced usage

Look at the code

The package first define the name of the family of ornament **\vectorian** by default it's **vectorian**.

```
\begin{tikzpicture}[%
   baseline={([yshift=\pgfornamentydelta]%
   current bounding box.\pgfornamentanchor)},pgfornamentstyle]
   \pgftransformscale{\pgfornamentscale}%
   \pgf@@ornament{#2}%
\end{tikzpicture}%
```

Options for placement are **yshift=\pgfornamentydelta** and \pgfornamentanchor . Options for aspect are pgfornamentstyle and \pgfornamentscale . The object is called by \pgf@@ornament. This macro define locally other macros used for creating the symbols and it loads the symbol with \@@input \OrnamentsFamily#1.pgf.. The symbol with the rank #1 in the family \OrnamentsFamily is loaded.

```
\def\pgf@@ornament#1{%
\begingroup
\def\i{\pgfusepath{clip}}%
\let\o\pgfpathclose
\let\s\pgfusepathqfillstroke
\def\p ##1##2{\pgfqpoint{##1bp}{##2bp}}%
\def\m ##1 ##2 {\pgfpathmoveto{\p{##1}{##2}}}%
\def\l ##1 ##2 {\pgfpathlineto{\p{##1}{##2}}}%
\def\r ##1 ##2 ##3 ##4 {\pgfpathrectangle{\p{##1}{##2}}{%
                        \p{##3}{##4}}}%
\def\c ##1 ##2 ##3 ##4 ##5 ##6 {%
\pgfpathcurveto{\p{##1}{##2}}{\p{##3}{##4}}{\p{##5}{##6}}}%
\@@input \OrnamentsFamily#1.pgf%
\endgroup}%
```

A symbol : the next code is used to define the first object of the family **am**. For example I created two very simple vector ornaments am1.pgf and am2.pgf . The family **am** is actually composed by two elements.

The real definition of an object uses a lot of bytes, with the mechanism[7] described above, I can save the object like this :

[7] I received an useful help from *Enrico Gregorio*

```
\m 0 0
\c 50 0 150 0 200 16
\c 250 0 350 0 400 0
\l 400 1
\c 350 0 250 0 200 22
\c 150 0 50 0 0 1
\l 0 0
\s
\endinput
```

How to use the code differently

For example you can create a new macro to call an object of another family and you can modifiy the object.

```
\makeatletter
\newcommand{\callornament}[1]{%
\begingroup
\def\i{\pgfusepath{clip}}%
\let\o\pgfpathclose
\let\s\pgfusepathqfillstroke
\def\p ##1##2{\pgfqpoint{##1bp}{##2bp}}%
\def\m ##1 ##2 {\pgfpathmoveto{\p{##1}{##2}}}%
\def\l ##1 ##2 {\pgfpathlineto{\p{##1}{##2}}}%
\def\r ##1 ##2 ##3 ##4 {\pgfpathrectangle{\p{##1}{##2}}{%
                       \p{##3}{##4}}}%
\def\c ##1 ##2 ##3 ##4 ##5 ##6 {%
\pgfpathcurveto{\p{##1}{##2}}{\p{##3}{##4}}{\p{##5}{##6}}}%
\@@input #1\relax
\m 0 0 \l 400 0 \o\s
\endgroup}
\makeatother
```

```
\tikz[scale=.5] \callornament{am1.pgf}  ;
```

Figure 33: Usage of another family

Define a symbol with Inskape

You can create a symbol with **Inskape**, then you save the symbol with the format **LaTeX with Pstricks**.

```
%LaTeX with PSTricks extensions
%%Creator: inkscape 0.48.2
%%Please note this file requires PSTricks extensions
\psset{xunit=.5pt,yunit=.5pt,runit=.5pt}
\begin{pspicture}(744.09448242,1052.36218262)
{
\newrgbcolor{curcolor}{0 0 0}
\pscustom[linewidth=1,linecolor=curcolor]
{
\newpath
\moveto(231.428,665.714)
\curveto(235.869,658.981)(224.543,656.406)(220.238,658.333)
\curveto(208.570,663.555)(209.816,679.616)(216.666,688.095)
\curveto(228.919,703.261)(252.107,700.575)(265.000,687.857)
\curveto(283.919,669.192)(279.643,638.050)(260.952,620.952)
\curveto(236.039,598.163)(196.704,604.097)(175.476,628.809)
\curveto(148.762,659.906)(156.386,707.535)(187.142,732.857)
\curveto(224.393,763.525)(280.367,754.197)(309.761,717.380)
\curveto(344.402,673.993)(333.361,609.645)(290.476,576.190)
\curveto(240.963,537.565)(168.220,550.325)(130.714,599.285)
\curveto(88.097,654.917)(102.579,736.068)(157.619,777.619)
\curveto(219.364,824.233)(308.932,808.026)(354.523,746.904)
\curveto(405.139,679.048)(387.205,581.057)(319.999,531.428)
\curveto(294.222,512.3928)(262.917,501.397)(230.928,499.848)
}
}
\end{pspicture}
```

You modify the code like this : [8]

```
\begingroup
\def\i{\pgfusepath{clip}}%
\def\k{\pgfusepath{stroke}}%
\let\o\pgfpathclose
\let\s\pgfusepathqfillstroke
\def\p #1#2{\pgfqpoint{#1bp}{#2bp}}%
\def\m #1 #2 {\pgfpathmoveto{\p{#1}{#2}}}%
\def\r #1 #2 #3 #4 {\pgfpathrectangle{\p{#1}{#2}}{%
                    \p{#3}{#4}}}%
\def\l #1 #2 {\pgfpathlineto{\p{#1}{#2}}}%
\def\c #1 #2 #3 #4 #5 #6 {%
\pgfpathcurveto{\p{#1}{#2}}{\p{#3}{#4}}{\p{#5}{#6}}}%
\begin{tikzpicture}
\pgftransformscale{.4}
\m 231.428 665.714
\c 235.869 658.981 224.543 656.406 220.238 658.333
\c 208.570 663.555 209.816 679.616 216.666 688.095
\c 228.919 703.261 252.107 700.575 265.000 687.857
\c 283.919 669.192 279.643 638.050 260.952 620.952
\c 236.039 598.163 196.704 604.097 175.476 628.809
\c 148.762 659.906 156.386 707.535 187.142 732.857
\c 224.393 763.525 280.367 754.197 309.761 717.380
\c 344.402 673.993 333.361 609.645 290.476 576.190
\c 240.963 537.565 168.220 550.325 130.714 599.285
\c 88.097 654.917 102.579 736.068 157.619 777.619
\c 219.364 824.233 308.932 808.026 354.523 746.904
\c 405.139 679.048 387.205 581.057 319.999 531.428
\c 294.222 512.392 262.917 501.397 230.928 499.848
\k
\end{tikzpicture}
\endgroup
```

[8] You can also modify all the coordinates if you don't want to use **\pgftransformscale**

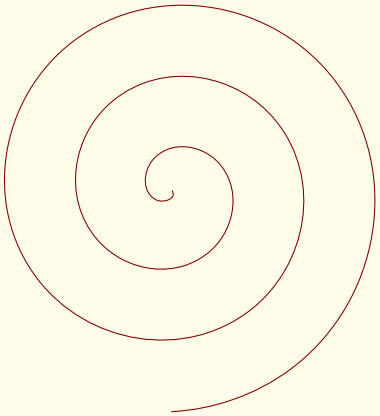


Figure 34: Symbol from Inskape

From .eps or .mps file

Another symbol : [9].

```
\begin{tikzpicture}
\pgftransformscale{.4}
\m 71.43 238.86
\l 310.29 238.86
\l 310.29 332.57
\l 428.57 214.29
\l 310.29 96.00
\l 310.29 189.71
\l 71.43  189.71
\l 71.43  238.86
\s
\m 453.14 381.71
\l 500.00 381.71
\l 500.00 46.86
\l 453.14 46.86
\l 453.14 381.71
\s
\end{tikzpicture}
```

[9] You can create a new family name **symb** and you save the new code in a file **symb1.pgf**. It's the first vector object of the new family

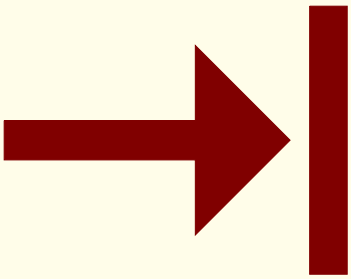


Figure 35: Symbol from .eps file

Index

TikZ package, 9, 12

\AddToShipoutPicture, 11
am family, 5
anchor, 21
\anchor, 11
anchors, 10
at, 16

baseline, 14

color, 23
color option, 6–8
current bounding box, 14
current page, 12

distributions
  MiKTeX, 5
  TeXLive, 5
\draw, 26

environments
  pspicture, 26
  tikzpicture, 6, 10, 13, 26
eso-pic package, 11, 12
every node , 23

families
  am, 5
  vectorian, 5, 10
\foreach, 25

\gfornamenthline, 22
\gfornamentvline, 22

height option, 6, 7
help lines, 25

inner sep, 14
Inskape, 29

\LenToUnit, 11

MiKTeX distribution, 5
minimum size, 21

\node, 14

options, 16
options(pgfornament)
  color, 6–8
  height, 6, 7
  pgfornamentstyle, 14
  scale, 6, 7
  symmetry=c, 6, 7
  symmetry=h, 6, 7
  symmetry=none, 6
  symmetry=v, 6, 7
  width, 6, 7
  ydelta, 6, 8
ornament
  lines, 19, 20
  symbols1, 17
  symbols2, 18
\OrnamentsFamily, 28
overlay, 12

packages
  TikZ, 9, 12
  eso-pic, 11, 12
  pgfornament, 1, 5, 11
  Pstricks, 29
  rotating, 13
  tikzrput, 25
pgfornament
  options, 6
\pgfornament, 6, 11, 26
pgfornament package, 1, 5, 11
\pgfornament[⟨*options*⟩]{*number*}, 6, 25
\pgfornamentanchor, 10, 28
\pgfornamentscale, 10, 28
pgfornamentstyle option, 14
pgfornamentstyle style(TikZ), 9, 10, 28
\pgfornamentydelta, 10, 28
\pgfpathclose, 29
\pgfpathcurveto, 29
\pgfpathlineto, 29
\pgfpathmoveto, 29
\pgfqpoint, 29
\pgfusepath, 29
\psframe, 26
pspicture environment, 26
Pstricks package, 29
\psvectorian, 26
\put, 11, 13

regular polygon, 24
\resetpgfornamentstyle, 14
rotate, 24
\rotatebox, 13
rotating package, 13
\rput, 25
\rput, 25

scale option, 6, 7
\setlength, 27
south, 25
\strippt, 11
styles(TikZ)

pgfornamentstyle, 9, 10, 28
symmetry=c option, 6, 7
symmetry=h option, 6, 7
symmetry=none option, 6
symmetry=v option, 6, 7

TeXLive distribution, 5
tikzpicture environment, 6, 10, 13, 26
tikzrput package, 25
to, 15
transform shape, 23

\unitlength, 27
\use as bounding box, 25

vectorian family, 5, 10

width option, 6, 7

ydelta option, 6, 8