

The Ornaments package

Alain Matthes

April 8, 2020



 (Version 1.1)

This document describes the L^AT_EX package *pgfornament* and presents the syntax and parameters of the macro "pgfornament". It also provides examples and comments on the package's use.

Firstly, I would like to thank Till Tantau for the beautiful L^AT_EX package, namely **TikZ**.

I am grateful to Vincent Le Moign for allowing us to distribute the ornaments ¹ in the format Pstricks and PGF/**TikZ**.

I also thank P. Fradin who first created a package on ornaments in relation to PStricks, which gave me the idea to do the same thing in relation to **TikZ**.

I would like to thank also Enrico Gregorio for some great ideas used in this package. You will find at the end of this document the 196 symbols provided with the package.

With this new version comes a new family of ornaments. Chennan Zhang drew the motifs using a CAD application, re-drew them in TikZ, and granted permission for these to be turned into a library (*pgfornament-han*) suitable for use with the *pgfornament* package by LianTze Lim. It is now possible to use directly the library for Chinese traditional motifs and patterns.

Next to the document you are reading, you will find documentation on the package *tikzrput*.





¹<http://www.vectorian.net/> (free sample)



Contents

How to install the package	4
How to use the package	5
The main macro	6
Number argument	6
Argument and options	7
Examples of the use of options	7
Style <i>pgfornamentstyle</i>	9
Advanced options from TikZ	9



What is a (pgf)ornament?	9
Placing a vector ornament on a page	10
On each page with the package <code>eso-pic</code>	10
On one page with the <code>picture</code> environment	11
With <code>TikZ[remember picture,overlay]</code>	11
Placing a vector ornament in the flow	11
 Directly 	11
In the flow with <code>TikZ</code>	12
Ornament inside a node	12
One ornament between two nodes	13
How to use <code>to [ornament= ...]</code>	13
How to use the option <code>ornament/at</code>	14
How to use the option <code>options</code>	14
How to make a line of ornaments	14
With the <code>chains</code> library	14
With the macro <code>\pgfornamentline</code>	15
Place ornaments with <code>chains</code> on a circle	15
Vectorian Library	15
Ornaments : Vector Symbols	15
Symbols part 1	15
Symbols part 2	17
Ornaments : Vector Corners	18
Ornaments : Vector Lines	19
Ornaments : Animals	19
Ornaments : Hands	21
Ornaments : Humans	22
Ornaments : Objects	22
Chinese traditional motifs	24
Corner symbols for connecting simple lines	24
Corner symbols for connecting double lines	24
Corner symbols	25
Single line, double line, straight line	25
Other symbols	26
Frame around a page	28





Application: Placing corners	29
Application: Create a frame for the page	29
Application: Frame around a text	29
Application: Text inside a frame	30
Application: Other way to get a pentagon	31
Package <i>tikzrput</i>	32
Example with <code>\rput</code>	32
Ornament with <code>\rput</code>	32
Examples from <i>pgfornament</i>	32
Example 1 from LianTze Lim	32
Example 2 from LianTze Lim	33
Example 3 (based on an example of LianTze Lim)	34
Example 4 (based on an example of LianTze Lim)	35
Examples from <i>psvectorian</i>	36
Large Title – e01	36
Cover with frame – e02	36
Little Title – e03	37
Advanced usage	37
Look at the code	37
How to use the code differently	38
Define a symbol with Inskape	39
From .eps or .mps file	40
Problem	40
Index	40



List of Figures

1	Result of the minimal code	5
2	Result of the minimal code	5
3	Vectorian ornament n° 1	6
4	Vectorian ornament n° 2	6
5	Chinese ornament n° 57	7
6	am ornament n° 1	7
7	Option scale	7
8	Option width	7





9	Option height	8
10	Option color	8
11	How to use color	8
12	Option opacity	8
13	Example for symmetry	8
14	Horizontal symmetry	8
15	Vertical symmetry	8
16	Central symmetry	8
17	How to use ydelta	8
18	How to use the style pgfornamentstyle	9
19	How to add TikZ ' styles	9
20	Minimal code to get an ornament	9
21	Assembling of ornaments version 2	12
22	Assembling of ornaments version 1	12
23	Style with node	12
24	Node in the flow	13
25	One ornament between two nodes	13
26	Two ornaments between two nodes	13
27	A pentagon	14
28	option at	14
29	How options	14
30	Line with chains library	14
31	A line with ornaments	15
32	A circle with ornaments	15
33	Placing corners	29
34	A poem	30
35	Text inside a frame with a tikzpicture 's environment	31
36	A pentagon	32
37	Example with \rput	32
38	Placement with rput	32
39	Example 1 LianTze Lim	32
40	Example 2 LianTze Lim	33
41	Example named e01 in psvectorian	36
42	Example named e02 in psvectorian	36
43	Example named e03	37
44	Usage of another family	39
45	Symbol from Inskape	40
46	Symbol from .eps file	40



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`
- `pgflibraryhan.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 `han` must to be in `/texmf/tex/generic`
- the folder `am` must to be in `/texmf/tex/generic`

With MiKTeX, copy the 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{pgfornament}
or
\usepackage[object=vectorian]{pgfornament}
```

in your preamble. The `pgfornament` package loads **TikZ**.

Without any options, `pgfornament` 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=pgfhan]{pgfornament}.
```

”`pgfhan`” is the family for Chinese traditional motifs and patterns.

I create `am` to show you how to create new symbols and how to use it (see the section 6). You can see below, the minimum code to get a vector ornament.

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

If you want to work with the Han library

```
\documentclass{scrartcl}
\usepackage[object=pgfhan]{pgfornament}
\begin{document}
\pgfornament[width = 2cm,
               color = SeaGreen]{78}
\end{document}
```

How to use different families of ornaments?

You have two possibilities: the macro `\newpgfornamentfamily` or an environment `newfamily`



Figure 1: Result of the minimal code



Figure 2: Result of the minimal code



For example:



with the code:

```
\newpgfornamentfamily{pgfhan}
\pgfornament[width = 2cm, color = SeaGreen]{59}
\newpgfornamentfamily{vectorian}
\pgfornament[width = 2cm, color = SeaGreen]{59}
```

Now with the environment. At the end, you will find the previous ornament library.



with the code:

```
\begin{newfamily}[pgfhan]
\begin{tikzpicture}
\node{\pgfornament[color=Dandelion,width=2cm]{1}};
\end{tikzpicture}
\end{newfamily}

\begin{tikzpicture}
\node{\pgfornament[color=MidnightBlue,width=2cm]{1}};
\end{tikzpicture}
```

The main macro

The macro `\pgfornament` draws the object linked to the given number, with the vectorian family this number is between 1 and now 196. 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 a picture defined by a *tikzpicture* environment.

Number argument

The number designs an object of a list by a rank.

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



Figure 3: Vectorian ornament nř 1



Figure 4: Vectorian ornament nř 2





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

```
\usepackage[object=pgfhan]{pgfornament}  
...  
\pgfornament[color=Mahogany,width=2cm]{57}
```

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



Figure 5: Chinese ornament nr 57



Figure 6: am ornament nr 1

Argument and options

The macro has six 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
opacity	1	nb inf 1, opacity 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}
```



Figure 7: Option scale

2. Option width

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



Figure 8: Option width

3. Option height



```
\pgfornament[height=1cm]{77}
```



Figure 9: Option height

4. Option color

```
\pgfornament[height=1cm,color=green!20!black]{77}
```



Figure 10: Option color

```
\pgfornament[color=MidnightBlue,width=3cm]{24}%
```



Figure 11: How to use color

5. Option opacity

```
\pgfornament[height=1cm,color=green!20!black,opacity=0.2]{77}
```



Figure 12: Option opacity

6. Option symmetry=h (Symmetry horizontal axis)



Figure 13: Example for symmetry

7. Option symmetry=v (Symmetry vertical axis)



Figure 14: Horizontal symmetry

8. Option symmetry=c (Symmetry with respect to the origin)

9. Option ydelta

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

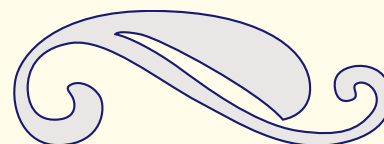


Figure 15: Vertical symmetry



Figure 16: Central symmetry

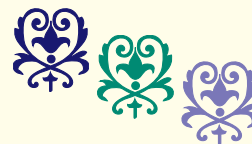


Figure 17: How to use ydelta



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=1.25,anchor=south]{24}%
\end{tikzpicture}
```



Figure 18: How to use the style `pgfornamentstyle`

Advanced options from TikZ

```
\begin{tikzpicture}
  \tikzset{pgfornamentstyle/.style={draw=Periwinkle,
    fill=SpringGreen}}%
  \node[draw=Periwinkle,draw opacity=.5,fill=SpringGreen,
    inner sep=0pt,fill=GreenYellow] at (0,0){%
    \pgfornament[anchor=center]{24}};
\end{tikzpicture}
```

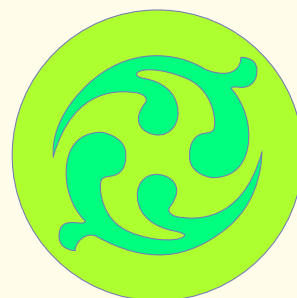


Figure 19: How to add TikZ' styles

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 20. I chose this method so that the use is as simple as possible.

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

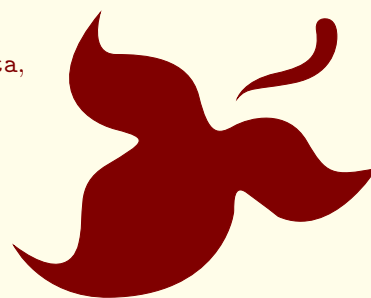
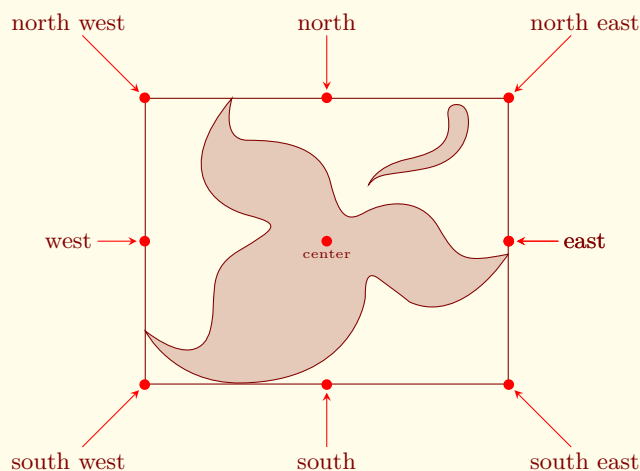


Figure 20: Minimal code to get an ornament

The ornament is placed in a rectangle².

² 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`.





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.³

³ `\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[remember picture,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.





```
{ \color{black}baseline \pgfsetfillopacity{0.2}%
\fbbox{\pgfornament[anchor=south,width=2cm]{69}}%
\fbbox{\pgfornament[width=2cm]{69}}%
\fbbox{\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}
\draw[anchor=west,rotate=\a,inner sep=0pt,xshift=12pt] {
\pgfornament[width=1cm]{88}};
\end{tikzpicture}
```

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


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` ⁴.

```
\tikzset{pgfornamentstyle/.style={%
draw=green!20!black,inner sep=0pt,fill=orange,
fill opacity=.5,scale=1.25,ultra thick}}%
\tikz\node {\fbbox{\pgfornament{3}}};
```

 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 :

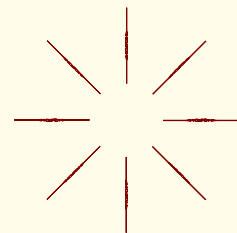


Figure 21: Assembling of ornaments version 2

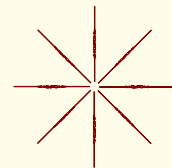
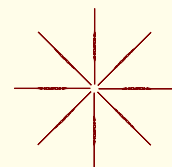


Figure 22: Assembling of ornaments version 1



⁴ I you want to reset the style you can use `\resetpgfornamentstyle`

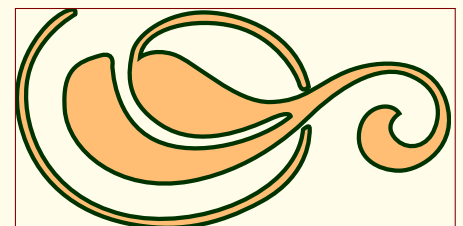


Figure 23: Style with node



```
\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 24: Node in the flow

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 [ornament = <number>] (B) ;
```

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

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

```
\begin{tikzpicture}
\node (A) at (0,0) {};
\node (B) at (4,2) {};
\draw [help lines,color=Maroon!60] (0,0) grid (4,2);
\draw [fill=Maroon!30] (A) circle (2pt) (B) circle (2pt);
\draw [orange] (A) to [ornament=88] (B);
\end{tikzpicture}
```

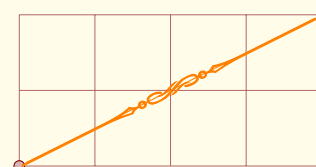


Figure 25: One ornament between two nodes

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}
```

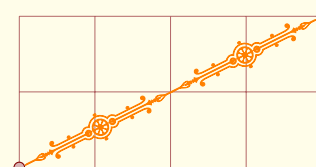


Figure 26: Two ornaments between two nodes

Example with a pentagon

```
\begin{tikzpicture}[every node={anchor=center,
                                inner sep=0pt}]
\node[regular polygon, regular polygon sides=5,
rotate=36,minimum size=5cm,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}
```

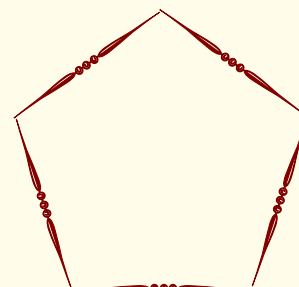


Figure 27: A pentagon

How to use the option **ornament/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=red!60] (0,-1) grid (4,1);
\path (A.center) to [ornament=67,ornament/at=0,
                    options/.append style={scale=.25}] (B.center);
\path (A.center) to [ornament=67,ornament/at=1,
                    options/.append style={scale=.25}] (B.center);
\end{tikzpicture}
```

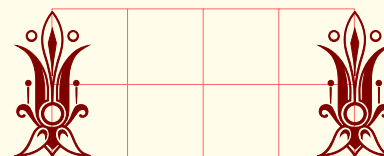


Figure 28: option at

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 (4,2) {};
\draw [help lines,color=Maroon!40] (0,0) grid (4,2);
\draw [fill=Maroon!20] (A) circle (2pt) (B) circle (2pt);
\path (A.center) to [ornament=84,
                    options/.append style={yshift=10pt}] (B.center);
\end{tikzpicture}
```

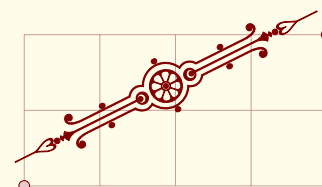


Figure 29: How options

How to make a line of ornaments

With the chains library

```
\begin{tikzpicture}
\node[draw,circle,
      minimum size=4pt,inner sep=0] (A) at (0,0){};
\coordinate (B) at (8,0);
\begin{chain,node distance=0,inner sep=0}
\node[anchor=west] [on chain] at (A){\pgfornament[width=1cm]{70}};
\node [on chain] {\pgfornament[width=1cm]{70}};
\node [on chain] {\pgfornament[width=1cm]{70}};
\end{chain}
```



Figure 30: Line with chains library



```
\node [on chain] {\pgfornament[width=1cm]{70}};}
\end{tikzpicture}
```

With the macro `\pgfornamentline`

Autopsy of this macro, you need 4 mandatory arguments: first of all two points between which the line is placed, the number of ornaments to create the line and of course the number of the ornament. An optional argument allows you to set options.

```
\begin{tikzpicture}[bullet/.style={%
    circle,draw,fill=black!30,inner sep=2pt}]
\draw [help lines,color=black!60] (0,0) grid (5,2);
\node[bullet] (A) at (0,0) {};
\node[bullet] (B) at (6,4) {};
\pgfornamentline[color=red]{A}{B}{4}{88}
\end{tikzpicture}
```

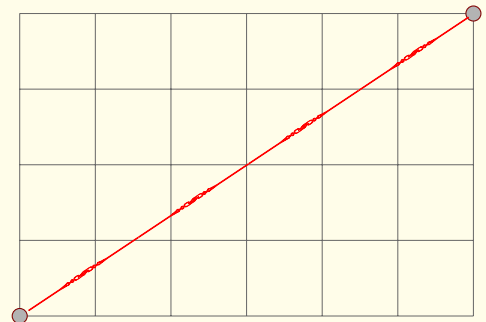


Figure 31: A line with ornaments

Place ornaments with chains on a circle

```
\begin{tikzpicture}[start chain=circle placed %
    {at=(\tikzchaincount*30:2)}]
\foreach \i in {1,...,12} \node [on chain]%
    {\pgfornament[width=1cm]{4}};
\end{tikzpicture}
```

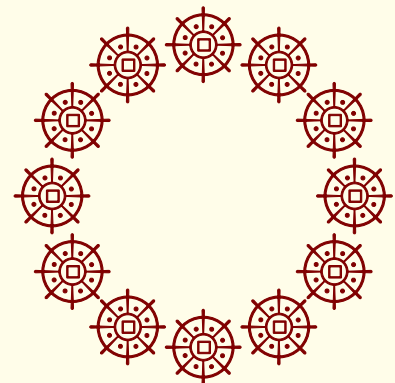


Figure 32: A circle with ornaments

Vectorian Library

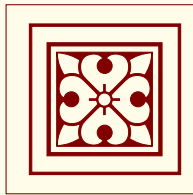
Ornaments : Vector Symbols

Here a list of the first thirty elements

Symbols part 1

	1 X: 136 Y: 107		2 X: 134 Y: 48
	3 X: 130 Y: 65		4 X: 133 Y: 133
	5 X: 129 Y: 146		6 X: 134 Y: 148





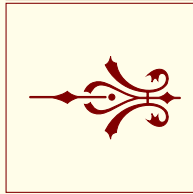
7

X: 136
Y: 135



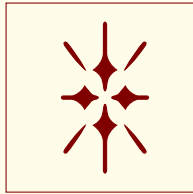
9

X: 79
Y: 105



11

X: 123
Y: 67



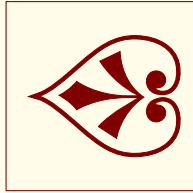
13

X: 136
Y: 236



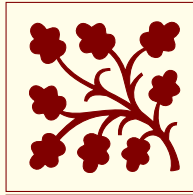
15

X: 103
Y: 52



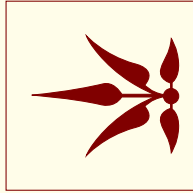
17

X: 74
Y: 59



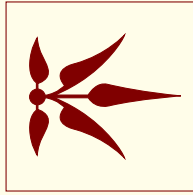
19

X: 81
Y: 81



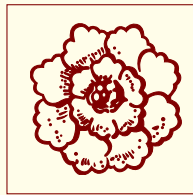
21

X: 70
Y: 58



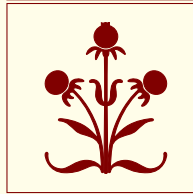
23

X: 68
Y: 55



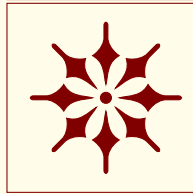
8

X: 134
Y: 134



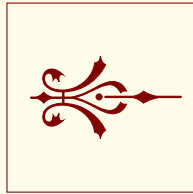
10

X: 80
Y: 99



12

X: 136
Y: 136



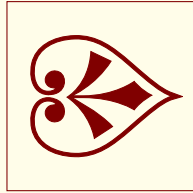
14

X: 123
Y: 67



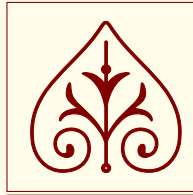
16

X: 103
Y: 52



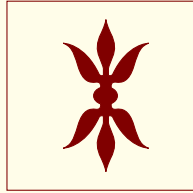
18

X: 74
Y: 59



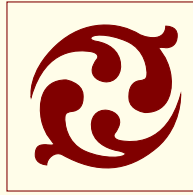
20

X: 81
Y: 81



22

X: 34
Y: 61

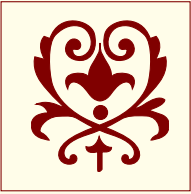
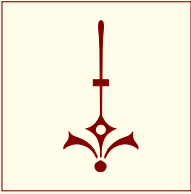

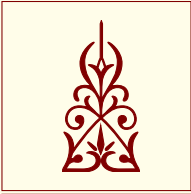

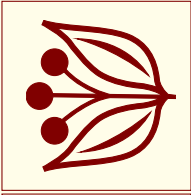

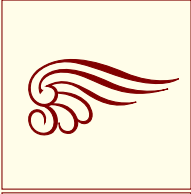




24

X: 79
Y: 76




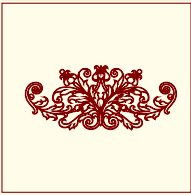
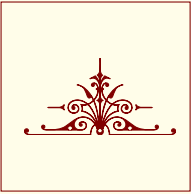





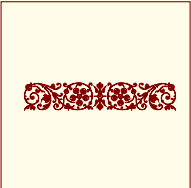
	25 X: 80 Y: 88		26 X: 59 Y: 120
	27 X: 101 Y: 98		28 X: 52 Y: 102
	29 X: 65 Y: 65		30 X: 63 Y: 64
	97 X: 29 Y: 14		98 X: 29 Y: 14
	140 X: 15 Y: 15		141 X: 15 Y: 15

Symbols part 2

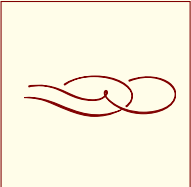
Tsubhe next list is about symbols of decoration. The design is more sophisticated. Be careful indices range from sixty-five to seventy-nine.

	65 X: 132 Y: 166		66 X: 177 Y: 175
	67 X: 80 Y: 155		68 X: 361 Y: 154
	69 X: 448 Y: 227		70 X: 226 Y: 79





71
X: 443
Y: 81



73
X: 216
Y: 58



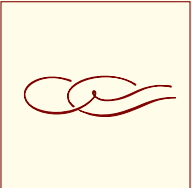
75
X: 373
Y: 120



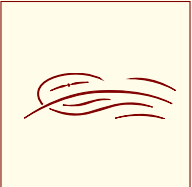
77
X: 207
Y: 89



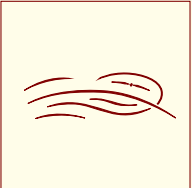
79
X: 249
Y: 122



72
X: 216
Y: 58



74
X: 308
Y: 93



76
X: 308
Y: 93



78
X: 207
Y: 132

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
X: 48
Y: 48



32
X: 48
Y: 48



33
X: 85
Y: 85



34
X: 85
Y: 85



35
X: 97
Y: 97



36
X: 97
Y: 97





	37 X: 105 Y: 104		38 X: 105 Y: 104
	39 X: 112 Y: 112		40 X: 112 Y: 112
	41 X: 111 Y: 113		42 X: 111 Y: 113
	61 X: 205 Y: 205		62 X: 205 Y: 205
	63 X: 212 Y: 212		64 X: 212 Y: 212
	131 X: 45 Y: 44		132 X: 45 Y: 44
	194 X: 90 Y: 90		195 X: 90 Y: 90

Ornaments : Vector Lines

The next list concerns symbols used to make a line.

	80 X: 454 Y: 31		89 X: 454 Y: 10
--	------------------------------	--	------------------------------

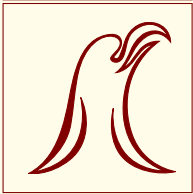
Ornaments : Animals

The next list concerns symbols with animals.





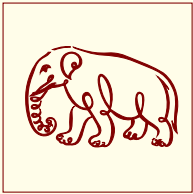
90
X: 30
Y: 30



100
X: 29
Y: 30



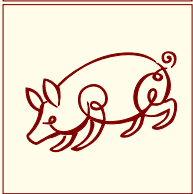
104
X: 44
Y: 29



107
X: 44
Y: 29



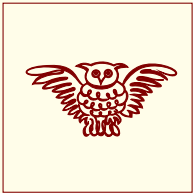
109
X: 72
Y: 29



111
X: 43
Y: 28



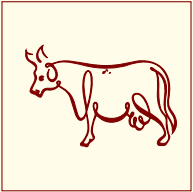
113
X: 57
Y: 28



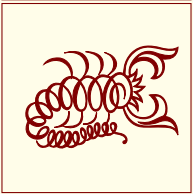
123
X: 59
Y: 29



158
X: 58
Y: 29



91
X: 44
Y: 29



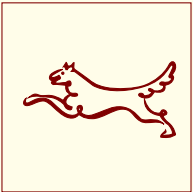
102
X: 43
Y: 29



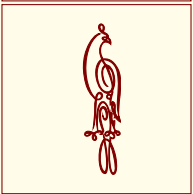
106
X: 44
Y: 30



108
X: 59
Y: 29



110
X: 29
Y: 15



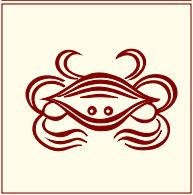
112
X: 15
Y: 59



122
X: 44
Y: 29

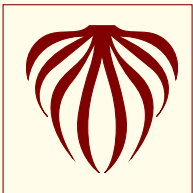


124
X: 58
Y: 59



159
X: 43
Y: 29





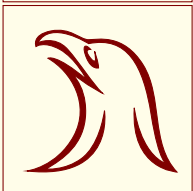
133

X: 15
Y: 15



134

X: 15
Y: 15



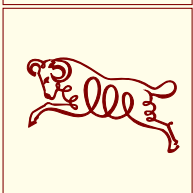
135

X: 15
Y: 14



136

X: 72
Y: 44



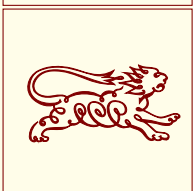
156

X: 45
Y: 28



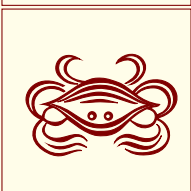
157

X: 44
Y: 29



158

X: 58
Y: 29



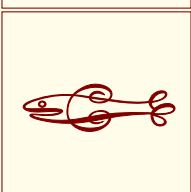
159

X: 43
Y: 29



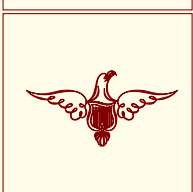
190

X: 44
Y: 29



193

X: 44
Y: 14

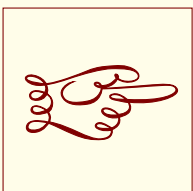


137

X: 149
Y: 74

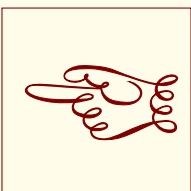
Ornaments : Hands

Remark : Ornaments 154 and 155 are identic but their sizes are smaller.



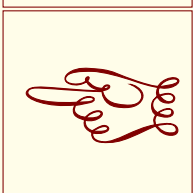
152

X: 57
Y: 28



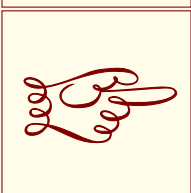
153

X: 57
Y: 28



154

X: 34
Y: 17



155


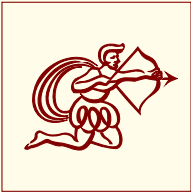
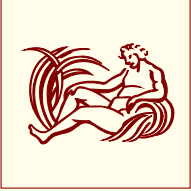
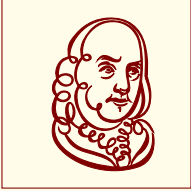
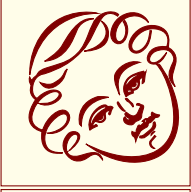


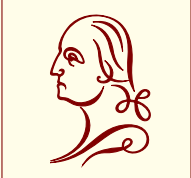
X: 34
Y: 17




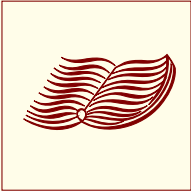
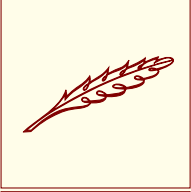
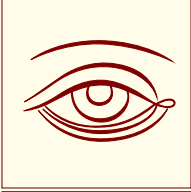

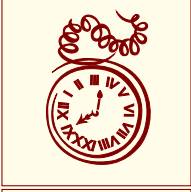




Ornaments : Humans

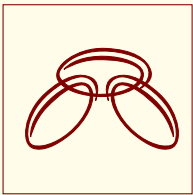
Remark : Ornaments 143, 144 and 145, 146 are identic but their sizes are different.

	95 X: 43 Y: 29		103 X: 42 Y: 30
	105 X: 45 Y: 30		125 X: 30 Y: 44
	143 X: 30 Y: 30		144 X: 29 Y: 30
	160 X: 43 Y: 30		164 X: 29 Y: 44

Ornaments : Objects

	92 X: 43 Y: 29		93 X: 59 Y: 29
	94 X: 59 Y: 29		95 X: 43 Y: 29
	114 X: 44 Y: 29		126 X: 29 Y: 43
	147 X: 31 Y: 29		148 X: 30 Y: 44





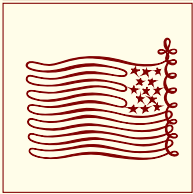
151

X: 43
Y: 29



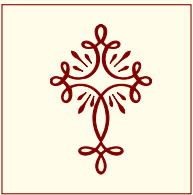
173

X: 29
Y: 44



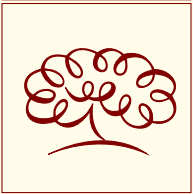
191

X: 73
Y: 58



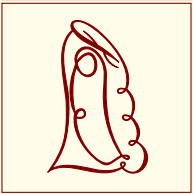
162

X: 30
Y: 44



184

X: 42
Y: 29



192

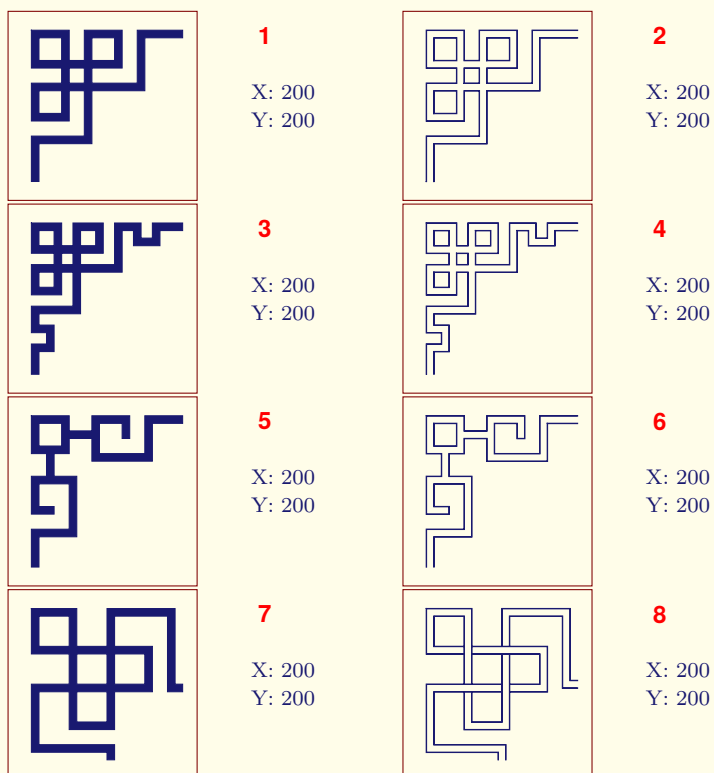
X: 29
Y: 44



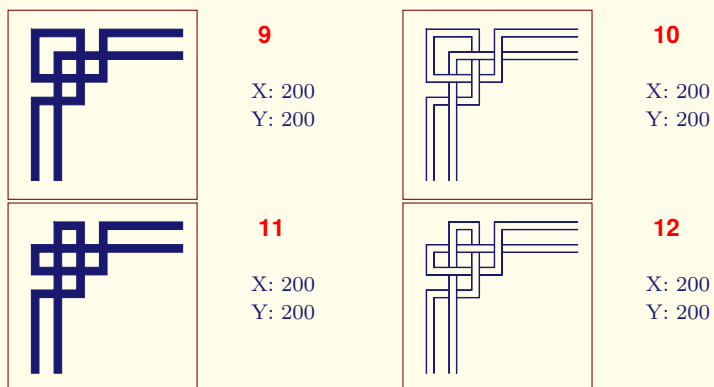
Chinese traditional motifs

This library of Chinese motifs is the work of two people: LianTze Lim and Chennan Zhang. They've been trying to provide some of the traditional patterns of the Han style using the existing mechanism of `pgfornament`. All patterns were finalized by Chennan Zhang in CAD, drawn by TikZ, and converted by LianTze Lim into macro package code suitable for the `pgfornament` mechanism. This package is called `pgfornament-han`. Now I've incorporated the patterns directly...

Corner symbols for connecting simple lines



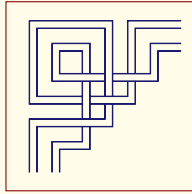
Corner symbols for connecting double lines





13

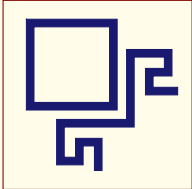
X: 200
Y: 200



14

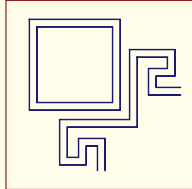
X: 200
Y: 200

Corner symbols



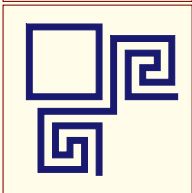
19

X: 200
Y: 200



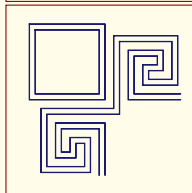
20

X: 200
Y: 200



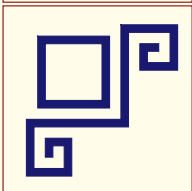
21

X: 260
Y: 260



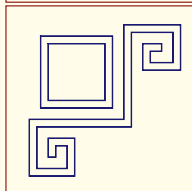
22

X: 260
Y: 260



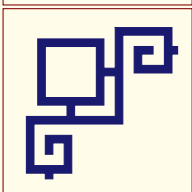
23

X: 200
Y: 200



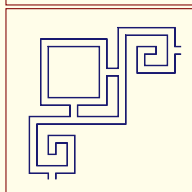
24

X: 200
Y: 200



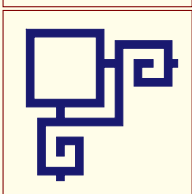
25

X: 200
Y: 200



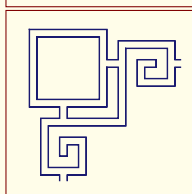
26

X: 200
Y: 200



27

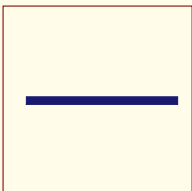
X: 200
Y: 200



28

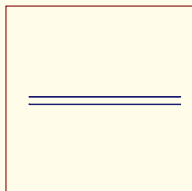
X: 200
Y: 200

Single line, double line, straight line



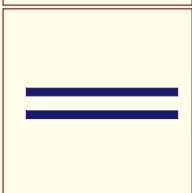
29

X: 200
Y: 10



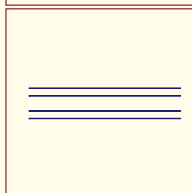
30

X: 200
Y: 10



31

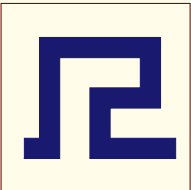
X: 200
Y: 40



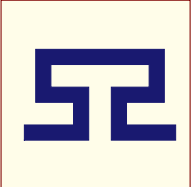
32

X: 200
Y: 40

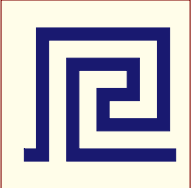
Other symbols



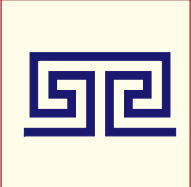
33
X: 75
Y: 60



35
X: 100
Y: 60



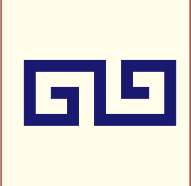
37
X: 125
Y: 110



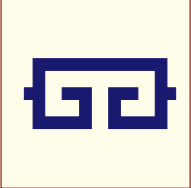
39
X: 200
Y: 110



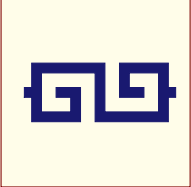
41
X: 115
Y: 60



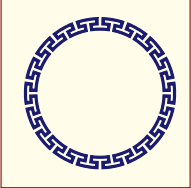
43
X: 140
Y: 60



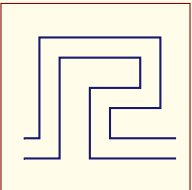
45
X: 130
Y: 60



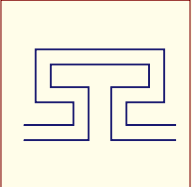
47
X: 155
Y: 60



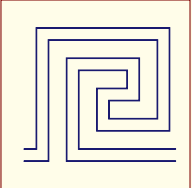
49
X: 360
Y: 360



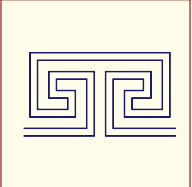
34
X: 75
Y: 60



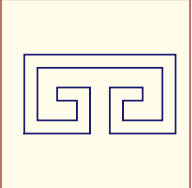
36
X: 100
Y: 60



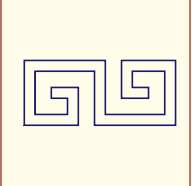
38
X: 125
Y: 110



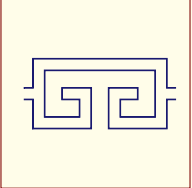
40
X: 200
Y: 110



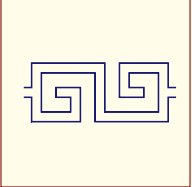
42
X: 115
Y: 60



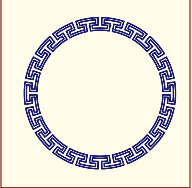
44
X: 140
Y: 60



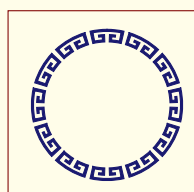
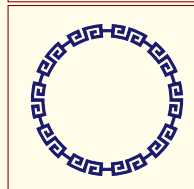
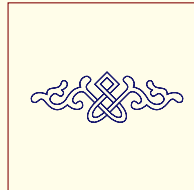
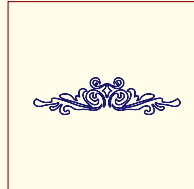
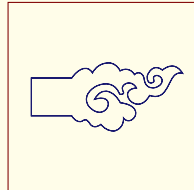
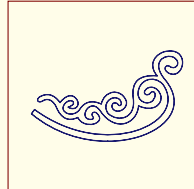
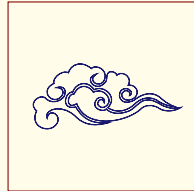
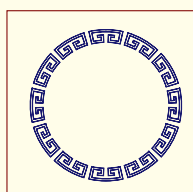
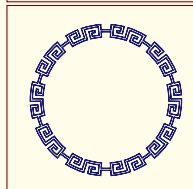
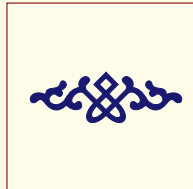
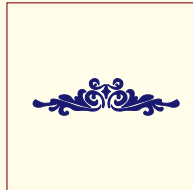
46
X: 130
Y: 60

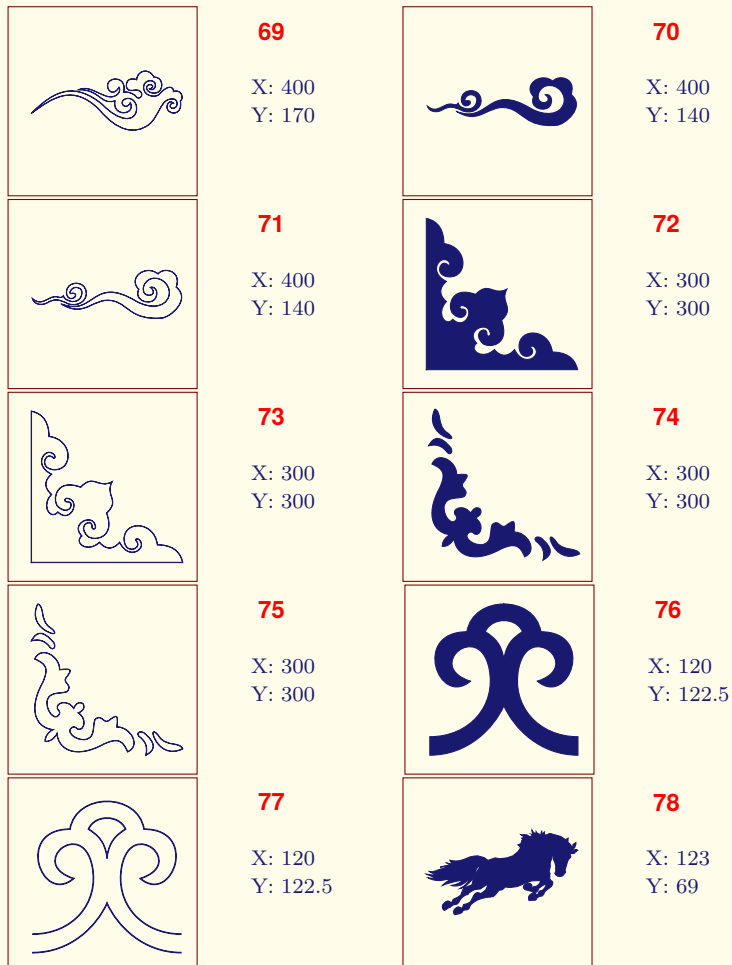


48
X: 155
Y: 60



50
X: 360
Y: 360

**51**X: 360
Y: 360**53**X: 360
Y: 360**55**X: 200
Y: 200**57**X: 200
Y: 200**59**X: 810
Y: 270**61**X: 450
Y: 120**63**X: 400
Y: 180**65**X: 350
Y: 210**67**X: 400
Y: 180**52**X: 360
Y: 360**54**X: 360
Y: 360**56**X: 200
Y: 200**58**X: 810
Y: 270**60**X: 450
Y: 120**62**X: 400
Y: 180**64**X: 350
Y: 210**66**X: 400
Y: 180**68**X: 400
Y: 170



Frame around a page

Here the code to the frame around the page

```
\AddToShipoutPicture{%
\begingroup
\setlength{\@tempdima}{2mm}%
\setlength{\@tempdimb}{\paperwidth-\@tempdima-1cm}%
\setlength{\@tempdimc}{\paperheight-\@tempdima}%
\put(\LenToUnit{\@tempdima},\LenToUnit{\@tempdimc}){%
  \pgfornament[color=Maroon,anchor=north west,width=1cm]{39}}
\put(\LenToUnit{\@tempdima},\LenToUnit{\@tempdima}){%
  \pgfornament[color=Maroon,anchor=south west,width=1cm,symmetry=h]{39}}
\put(\LenToUnit{\@tempdimb},\LenToUnit{\@tempdimc}){%
  \pgfornament[color=Maroon,anchor=north east,width=1cm,symmetry=v]{39}}
\put(\LenToUnit{\@tempdimb},\LenToUnit{\@tempdima}){%
  \pgfornament[color=Maroon,anchor=south east,width=1cm,symmetry=c]{39}}
\endgroup
}
```

`\let\strippt\strip@pt`

Application: Placing corners

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

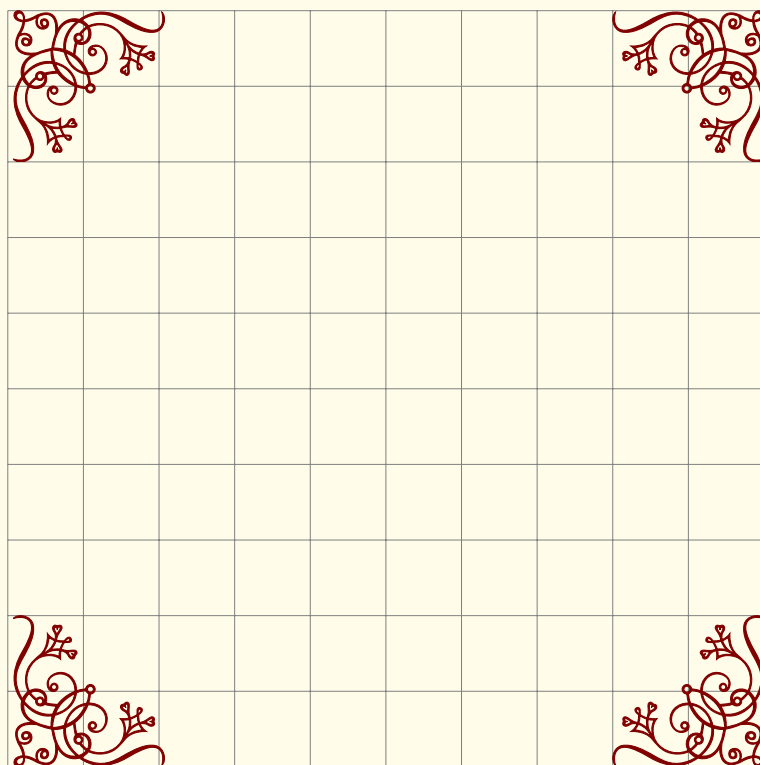


Figure 33: Placing corners

```
\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: Create a frame for the page

Application: Frame around a text

I chose a poem to illustrate this theme.

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

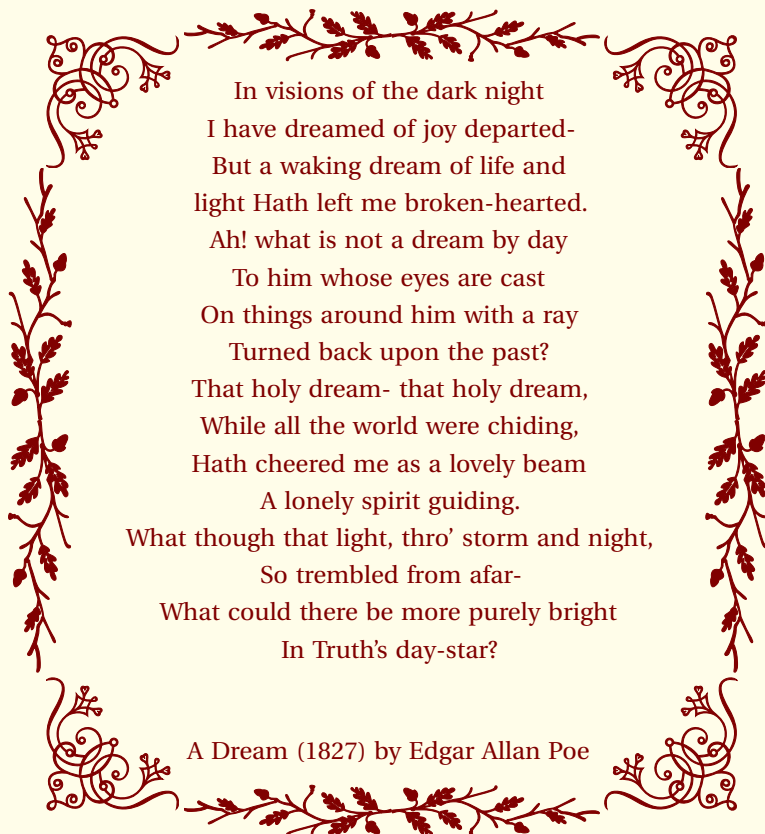


Figure 34: A poem

macros `\pgfornamenthline` and `\pgfornamentvline` 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 then we place the text in a node relatively to others nodes.

```
\newcommand{\framesize}{8 cm}
\begin{tikzpicture}[color=Maroon,
  transform shape,
```

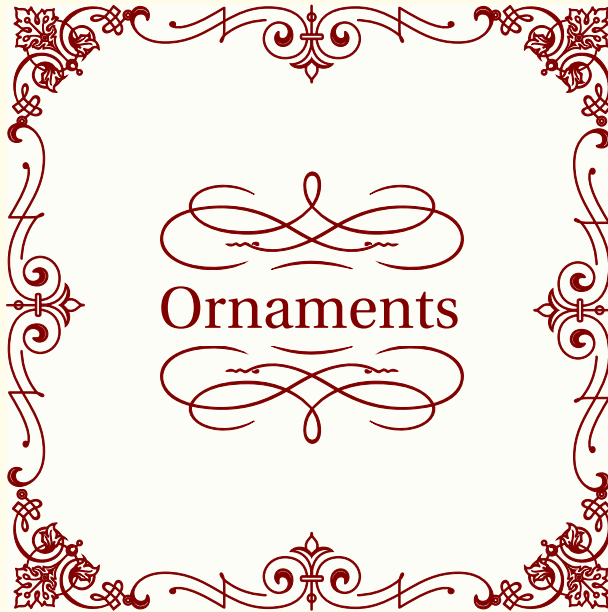


Figure 35: Text inside a frame with a `tikzpicture`'s environment

```

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 .⁵

⁵ `\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}

```

```

\begin{tikzpicture}
\draw[red] (0,0) -- (1,1) -- (2,0) -- (1,-1) -- (0,0);
\foreach \side in {1,...,5}
\foreach \angle in {0,72,...,216}
\foreach \width in {0.5cm,1cm,...,2cm}
\pgfornament[width=\width]{\side}
\end{tikzpicture}

```

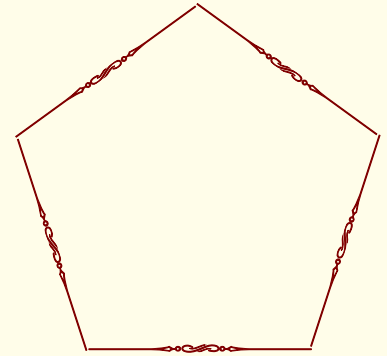


Figure 36: A pentagon

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. Next to the document you are reading, you will find documentation on this package. The display of an object at the point (x,y) is realized with `\rput` of pstricks like this :

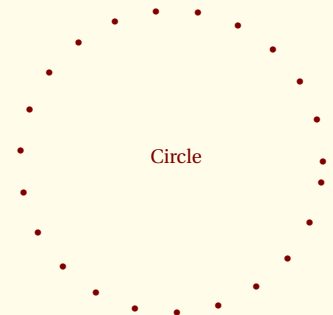
```
\rput[(refpoint)]{angle}(x,y){\pgfornament[(options)]{number}}
```

Example with `\rput`

```

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

```

Figure 37: 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 38: Placement with `rput`

Examples from `pgfornament-han`

Example 1 from LianTze Lim

<https://github.com/liantze/pgfornament-han>

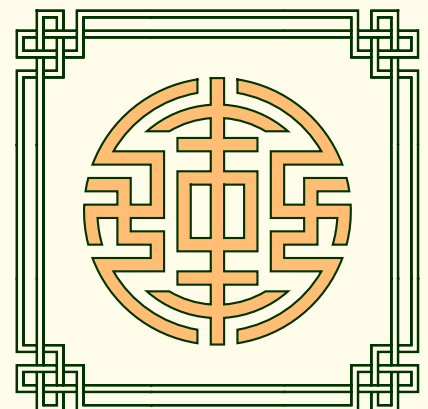


Figure 39: Example 1 LianTze Lim


```

\begin{tikzpicture}
\tikzset{every node/.append style={inner sep=0pt,color=MidnightBlue!50}}
\tikzset{pgfornamentstyle/.style={draw=green!20!black,
    fill=orange,fill opacity=.5,thick}}%
\node (nw) {\pgfornament[scale=0.25]{12}};
\node[right=50bp of nw] (ne){\pgfornament[scale=0.25,symmetry=v]{12}};
\node[below=50bp of nw] (sw){\pgfornament[scale=0.25,symmetry=h]{12}};
\node[below=50bp of ne] (se){\pgfornament[scale=0.25,symmetry=c]{12}};
\node[anchor=north west] at (nw.north east)%
    {\pgfornament[scale=0.25]{32}};
\node[anchor=south west] at (sw.south east)%
    {\pgfornament[scale=0.25]{32}};
\node[anchor=south west,rotate=-90] at (nw.south west)
    {\pgfornament[scale=0.25]{32}};
\node[anchor=south east,rotate=90] at (ne.south east)
    {\pgfornament[scale=0.25]{32}};
\node[anchor=center,shift={(25bp,-25bp)}] at (nw.south east)
    {\pgfornament[scale=0.5]{57}};
\end{tikzpicture}

```

Example 2 from LianTze Lim

```

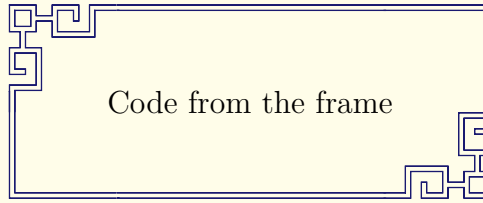
\begin{tikzpicture}
\tikzset{every node/.append style={color=Goldenrod,inner sep=0pt}}
\node (nw) {\pgfornament[scale=0.25]{23}};
\node[right=53bp of nw] (ne){\pgfornament[scale=0.4,symmetry=v]{23}};
\node[anchor=north west,xshift=8bp] at (nw.north east)
    {\pgfornament[scale=0.25]{41}};
\node[anchor=north east,xshift=-8bp] at (ne.north west)
    {\pgfornament[scale=0.25,symmetry=v]{41}};
\end{tikzpicture}

```



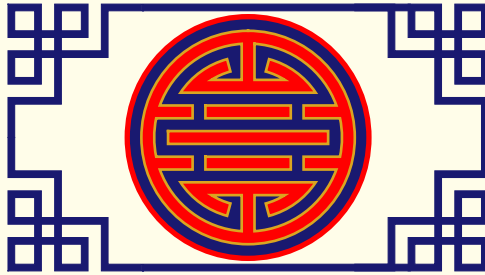
Figure 40: Example 2 LianTze Lim

Example 3 (based on an example of LianTze Lim)



```
\newpgfornamentfamily{pgfhan}
\newbox{\fortyseven}
\savebox{\fortyseven}{\pgfornament[scale=0.20,color=MidnightBlue]{47}}
\tikzset{every node/.append style={inner sep=0pt}}
\AddToShipoutPictureBG{%
\begin{tikzpicture}[overlay,remember picture,color=MidnightBlue]
\node[anchor=north west,shift={(0.7,-0.85)}] at (current page.north west)
(nw) {\pgfornament[scale=0.2]{25}};
\node[anchor=north east,shift={(-0.7,-0.85)}] at (current page.north east)
(ne) {\pgfornament[scale=0.2,symmetry=v]{25}};
\node[anchor=south west,shift={(0.7,0.85)}] at (current page.south west)
(sw) {\pgfornament[scale=0.2,symmetry=h]{25}};
\node[anchor=south east,shift={(-0.7,0.85)}] at (current page.south east)
(se) {\pgfornament[scale=0.2,symmetry=c]{25}};
\begin{scope}[start chain,node distance=-3pt]
\node[anchor=north west,on chain] at (nw.north east)
{\usebox{\fortyseven}};
\foreach \i in {1,...,14} {\node[on chain]{\usebox{\fortyseven}};}
\end{scope}
\begin{scope}[start chain,node distance=-3pt]
\node[anchor=south west,on chain] at (sw.south east)
{\usebox{\fortyseven}};
\foreach \i in {1,...,6} \node[on chain]{\usebox{\fortyseven}};
\end{scope}
\begin{scope}[start chain=going left,node distance=-3pt]
\node[anchor=south east,on chain,xshift={3pt}] at (se.south west)
{\usebox{\fortyseven}};
\foreach \i in {1,...,6} \node[on chain]
{\usebox{\fortyseven}};
\end{scope}
\foreach \i in {0,...,22}
\node[anchor=south west,rotate=-90,
shift={($\i*(31bp,0)$)}] at (nw.south west)
{\usebox{\fortyseven}};
\foreach \i in {0,...,22}
\node[anchor=south east,rotate=90,shift={($\i*(-31bp,0)$)}] at
([yshift={+3pt}]ne.south east){\usebox{\fortyseven}};
\node[yshift=32pt] at (current page.south){\pgfornament[scale=0.1]{51}};
\node[yshift=32pt,text=black] at (current page.south){\large\thepage};
\end{tikzpicture}
}
```

Example 4 (based on an example of LianTze Lim)



```
\begin{newfamily}[pgfhan]
\begin{center}
\begin{tikzpicture}
\tikzset{every node/.append style={%
    inner sep=0pt,
    color= MidnightBlue}}
\node[minimum width=180bp,minimum height=100bp] (chframe){};
\node[anchor=north west] (nw) at (chframe.north west)
    {\pgfornament[scale=0.25]{1}};
\node[anchor=north east] at (chframe.north east)
    {\pgfornament[symmetry=v,scale=0.25]{1}};
\node[anchor=south west] (sw) at (chframe.south west)
    {\pgfornament[symmetry=h,scale=0.25]{1}};
\node[anchor=south east] at (chframe.south east)
    {\pgfornament[symmetry=c,scale=0.25]{1}};
\node[anchor=south west,xscale=2] at (sw.south east)
    {\pgfornament[scale=0.25]{29}};
\node[anchor=north west,xscale=2] at (nw.north east)
    {\pgfornament[scale=0.25]{29}};
% circle
\begin{scope}
\tikzset{pgfornamentstyle/.style={draw=Goldenrod,
    fill=Red,
    line width=1pt}}
\node[fill=MidnightBlue,circle,draw=Red,
    line width=2pt,inner sep=-8pt]
    at (chframe.center) {\pgfornament[scale=0.40]{56}};
\end{scope}
\end{tikzpicture}
\end{center}
\end{newfamily}
```

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 41: 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`.



Figure 42: Example named e02 in *psvectorian*

```

\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}}
\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}}
\rput[bl]{-90}(-5,3){\pgfornament[width=6cm]{46}}
\rput[bl]{90}(5,-3){\pgfornament[width=6cm]{46}}
\rput(0,0){\Huge Ornaments}
\rput[t](0,-0.5){\pgfornament[width=5cm]{75}}
\rput[b](0,0.5){\pgfornament[width=5cm]{69}}
\rput[tr]{-30}(-1,2.5){\pgfornament[width=2cm]{57}}
\rput[tl]{30}(1,2.5){\pgfornament[width=2cm,symmetry=v]{57}}};
\end{tikzpicture}

```

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.

```

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

```



Figure 43: 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`⁶ and `am2.pgf`. Actually the family **am** is only composed by two elements.

The real definition of an object uses a lot of bytes, with the mechanism⁷ described above, I can save the object like this :

⁶ The next code defines this ornament

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 modify 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 44: Usage of another family

Define a symbol with Inksape

You can create a symbol with **Inksape**, 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{\pgfpoint{#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
```

⁸ You can also modify all the co-ordinates if you don't want to use `\pgftransformscale`

```

\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

```

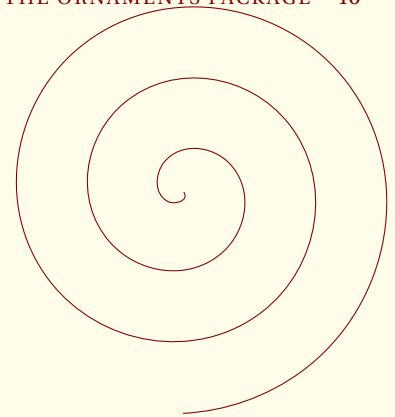


Figure 45: Symbol from Inksape

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 **symp** and you save the new code in a file **symp1.pgff**. It's the first vector object of the new family

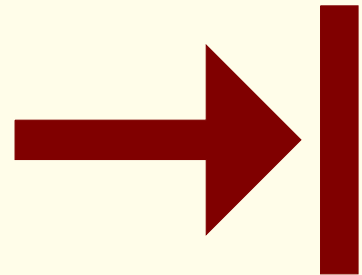


Figure 46: Symbol from .eps file

Problem

If you got an error like "Package tikz Error: + or - expected.", perhaps there is a conflict with the babel package. It's possible to resolve this type of conflict with `\shorthandoff{!}` just before your `tikzpicture`. You can also write in your preamble

```

\tikzset{every picture/.prefix style={%
  execute at begin picture=\shorthandoff{!}}}

```

and finally you can use `\usetikzlibrary{babel}` only with pgf 3.0

In french, you can get an error with `!`, `,` and `;`. Babel makes these characters activ

If you got a problem with the option `at` replace `at` by `ornament/at`.

Index

- TikZ** package, 9, 11
- `\AddToShipoutPicture`, 10
- am family, 5
- anchor, 29
- `\anchor`, 10
- anchors, 10
- baseline, 13
- chains library, 14, 15
- color, 31
- color option, 7, 8
- current bounding box, 13
- current page, 11
- distributions
 - MiKTeX, 5
 - TeXLive, 4
- `\draw`, 36
- environments
 - newfamily, 5
 - pspicture, 36
 - tikzpicture, 6, 9, 12, 36
- eso-pic package, 10, 11
- every node, 31
- families
 - am, 5
 - pgfhan, 5
 - vectorian, 5, 9
- `\foreach`, 32
- height option, 7
- inner sep, 13
- Inskape, 39
- `\LenToUnit`, 10
- library
 - chains, 14, 15
- MiKTeX distribution, 5
- minimum size, 29
- newfamily environment, 5
- `\newpgfornamentfamily`, 5
- `\node`, 12
- opacity option, 7, 8
- options, 14
- options(pgforment)
 - color, 7, 8
 - height, 7
 - opacity, 7, 8
 - pgformentstyle, 12
 - scale, 7
 - symmetry=c, 7, 8
 - symmetry=h, 7, 8
 - symmetry=none, 7
 - symmetry=v, 7, 8
 - width, 7
 - ydelta, 7, 8
- ornament
 - Animals, 19
 - corners1, 18
 - corners2, 19
 - lines, 19
 - symbols1a, 15
 - symbols2, 17
- ornament/at, 14
- `\OrnamentsFamily`, 37
- overlay, 11
- packages
 - TikZ**, 9, 11
 - eso-pic, 10, 11
 - pgforment, 1, 5, 10
 - pgforment-han, 1
 - Pstricks, 39
 - rotating, 12
 - tikzrput, 1, 32
- pgfhan family, 5
- pgforment
 - options, 7
- `\pgforment`, 6, 10, 36
- pgforment package, 1, 5, 10
- pgforment-han package, 1
- `\pgforment[(options)]{number}`, 6, 32
- `\pgformentanchor`, 9, 37
- `\pgformenthline`, 30
- `\pgformentline`, 2, 15
- `\pgformentscale`, 9, 37
- pgformentstyle option, 12
- pgformentstyle style(**TikZ**), 9, 37
- `\pgformentvline`, 30
- `\pgformentydelta`, 9, 37
- `\pgfpathclose`, 38
- `\pgfpathcurveto`, 38
- `\pgfpathlineto`, 38
- `\pgfpathmoveto`, 38
- `\pgfqpoint`, 38
- `\pgfusepath`, 38
- `\psframe`, 36
- pspicture environment, 36
- Pstricks package, 39
- `\psvectorian`, 36
- `\put`, 10, 12

- regular polygon, 32
- `\resetpgfornamentstyle`, 12
- rotate, 32
- `\rotatebox`, 12
- rotating package, 12
- rput, 32
- `\rput`, 32
- `\rput`, 32
- scale option, 7
- shorthandoff, 40
- `\strippt`, 11
- `styles(TikZ)`
 - `pgfornamentstyle`, 9, 37
- `symmetry=c` option, 7, 8
- `symmetry=h` option, 7, 8
- `symmetry=none` option, 7
- `symmetry=v` option, 7, 8
- TeXLive distribution, 4
- `tikzpicture` environment, 6, 9, 12, 36
- `tikzrput` package, 1, 32
- to, 13
- transform shape, 31
- vectorian family, 5, 9
- width option, 7
- `ydelta` option, 7, 8