

pgfornament——一个给你花样装饰的宏包

这次给大家介绍一个宏包——pgfornament，可以给文档添加非常漂亮的装饰效果，尤其是在封面制作，幻灯片演示方面应该有很好的应用。林莲枝老师在此包的基础上扩展编写了中文版本的 pgfornament-han 包，加入了很多中国的元素，也值得大家参考，但是这里的话我们只介绍 pgfornament 包的使用。

要使用此宏包,您只需要在导言区加入`\usepackage{pgfornament}`或者`\usepackage[object=vectorian]{pgfornament}`，此包会自动加载 TikZ 包。`\pgfornament`命令画出一个指定的数字对应的矢量图，这个命令可以单独用，也可以放在一个`picture`环境中，事实上此命令就是由当前位置的一个`tikzpicture`环境所定义的。

`\pgfornament`的完整命令如下

```
\pgfornament[<options>]{number}
```

这里的数字参数范围是 1-196，也就是本包中自定义的 196 个矢量图，大家可以查看 pgfornament 的源文档说明，命令行输入 `texdoc pgfornament` 即可，第 17 页开始列出了所有的矢量图。

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

您可以得到图1



图 1: 矢量图 1

```
\usepackage{ornament}
...
\pgfornament[width=2cm]{4}
```

您可以得到图2



图 2: 矢量图 2

此命令含有 6 个可选参数: `scale,width,height,color,opacity,ydelta,symmetry`，其中最后一个几何变换参数 `symmetry` 有四种选择。

1. scale 选项

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

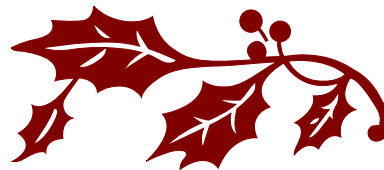


参数	默认值	定义
<code>scale</code>	1	图片的宽度和高度不变
<code>width</code>	{}	设置宽度，宽高比不变
<code>height</code>	{}	设置高度，宽高比不变
<code>color</code>	black	图片颜色
<code>opacity</code>	1	图片透明度
<code>ydelta</code>	0 pt	图片的垂直平移
<code>symmetry=v</code>	none	垂直轴对称
<code>symmetry=h</code>	none	水平轴对称
<code>symmetry=c</code>	none	中心对称
<code>symmetry=none</code>	none	默认不作变换

表 1: pgfornament 包的可选项

2. width 选项

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



3. height 选项

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



4. color 选项

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



5. opacity 选项

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



6. symmetry=h 选项

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



7. symmetry=v 选项

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



8. symmetry=c 选项

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

对称选项的例子

1. 垂直轴对称

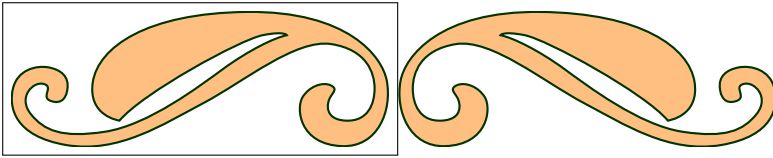


图 3: Vertical symmetry

2. 水平轴对称

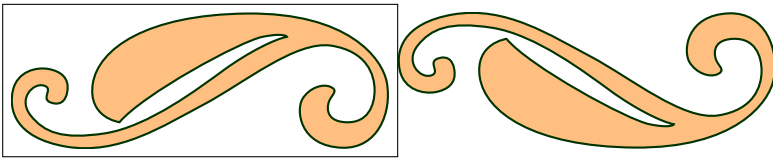


图 4: Horizontal symmetry

3. 原点中心对称

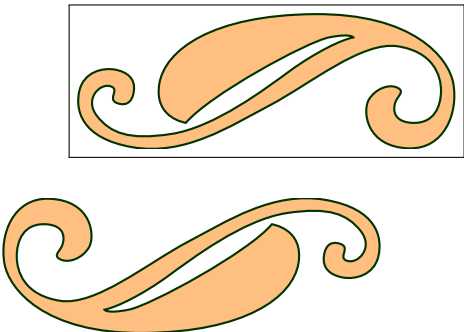
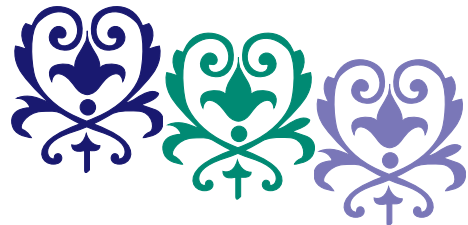


图 5: Central symmetry

ydelta 选项

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



除此之外，还可以借助`eso-pic`包给文档的每一页的四个角落增加一个对称的装饰，正如本文档一样，导言区加入如下代码即可。

```

\usepackage{eso-pic}
\makeatletter
\AddToShipoutPicture{%
\beginingroup
\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

```

这里的数字 63 可以更换为 31,33,335,37,39,41,61 以获得其他对称的装饰。

由于给每一页增加装饰,这会使得编译时间增加很多。所以如果只想给当前页增加装饰,可以定义如下命令来完成:

```

\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

```

当然,这个效果借助于 TikZ 包的 `current page` 选项也可以实现

```

\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}
}

```

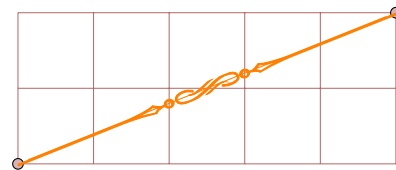
此包的选项还可以嵌入到tikzpicture环境的 to 算子选项中，语法如下：

```
\draw (A) to [ornament=<number>] (B) ;
```

```

\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}

```

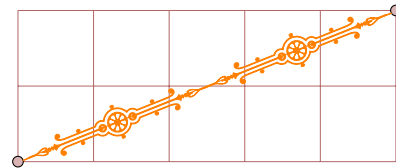


我们还可以用两个装饰来连接两个点

```

\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}

```

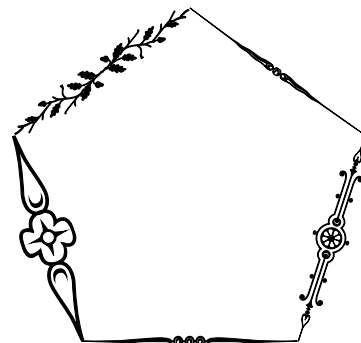


如果加载了 TikZ 的 shapes.geometric 库，我们还可以做出如下花边多边形（实际的多边形边数可以自行选择，比如六边形，然后设置 rotate=30）

```

\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=81] (s.side 2)
to [ornament=83] (s.side 3)
to [ornament=84] (s.side 4)
to [ornament=86] (s.side 5)
to [ornament=87] (s.side 1);
\end{tikzpicture}

```



最后呢，我们引用源文档中的一个漂亮的装饰环境来结束本文，因为代码略长，我就不在这里引用了，放在此链接中 <https://paste.ubuntu.com/p/NpxF2h8q9S/>。

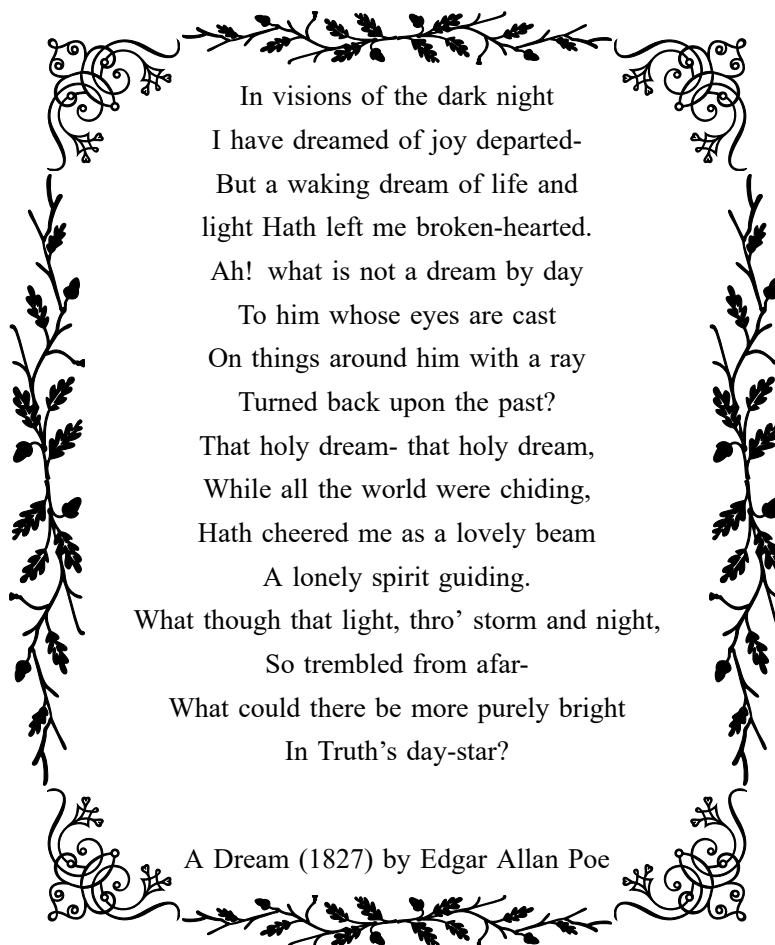


图 6: A poem