

## 1 圆

通过本节的命令中，可以定义并绘制圆。为此，需要知道圆心以及半径或圆上的点。常用的方法是给定圆心绘制过指定的点的圆，这是默认方法，否则则需要给出圆的半径 $R$ 。另外，还有一些特殊的圆，例如三角形的外接圆等。

- `\tkzDefCircle`命令根据指定的圆心和半径(单位: cm)定义一个圆, `\tkzGetPoint`和`\tkzGetLength`命令得到圆心和半径;
- `\tkzDrawCircle`命令用于绘制圆;
- `\tkzFillCircle`命令用于在不绘制圆的情况下对圆进行着色;
- `\tkzClipCircle`命令用于用圆进行裁剪;
- `\tkzLabelCircle`命令用于标注一个圆.

1.1 `\tkzDefCircle`命令: 定义圆

`\tkzDefCircle[⟨ 命令选项 ⟩](⟨A,B⟩) or (⟨A,B,C⟩)`

注意，参数可以是 2 个或 3 个点。该命令结合`\tkzGetPoint`命令和/或`\tkzGetLength`命令，得到圆心和圆的半径，或使用`\tkzPointResult`命令和`\tkzLengthResult`命令使用这些值，但不命名。

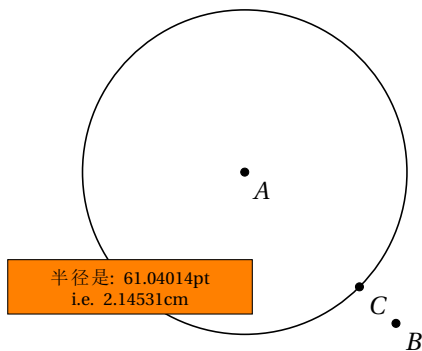
参数	样例	说明
<code>(⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)</code>	<code>(⟨A,B⟩)</code>	<code>[AB]</code> 是半径 $A$ 圆心

选项	默认值	含义
<code>through</code>	<code>through</code>	两点间的距离是半径
<code>diameter</code>	<code>through</code>	两点间的距离是直径
<code>circum</code>	<code>through</code>	三角形的外接圆
<code>in</code>	<code>through</code>	三角形的内切圆
<code>ex</code>	<code>through</code>	三角形的旁切圆
<code>euler or nine</code>	<code>through</code>	三角形的欧拉圆
<code>spieker</code>	<code>through</code>	三角形的 Spieker 圆
<code>apollonius</code>	<code>through</code>	Apollonius 圆
<code>orthogonal</code>	<code>through</code>	与指定圆心的另一个圆正交
<code>orthogonal through</code>	<code>through</code>	与通过两个点的另一个圆正交
<code>K</code>	1	Apollonius 圆的系数

下面的示例中，用到了还未说明的圆的绘制命令。多数情况下，仅需要使用该命令得到圆心和半径。

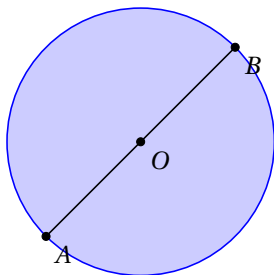
### 1.1.1 使用随机点与through选项示例



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,4){A}
  \tkzDefPoint(2,2){B}
  \tkzDefMidPoint(A,B) \tkzGetPoint{I}
  \tkzDefRandPointOn[segment = I--B]
  \tkzGetPoint{C}
  \tkzDefCircle[through] (A,C)
  \tkzGetLength{rACpt}
  \tkzpttocm(\rACpt){rACcm}
  \tkzDrawCircle(A,C)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B,C)
  \tkzLabelCircle[draw,fill=orange, text width=3cm,
    text centered, font=\scriptsize](A,C)(-90)%
    {半径是: \rACpt pt i.e. \rACcm cm}
\end{tikzpicture}
```

### 1.1.2 diameter选项示例

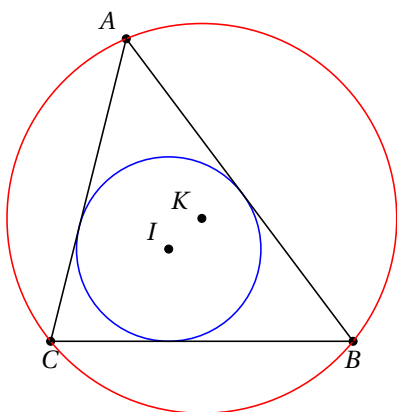
可以通过  $[AB]$  的中点确定圆心。



```
\begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(2,2){B}
  \tkzDefCircle[diameter] (A,B)
  \tkzGetPoint{O}
  \tkzDrawCircle[blue,fill=blue!20] (O,B)
  \tkzDrawSegment(A,B)
  \tkzDrawPoints(A,B,O)
  \tkzLabelPoints(A,B,O)
\end{tikzpicture}
```

### 1.1.3 三角形的内切圆和外接圆示例

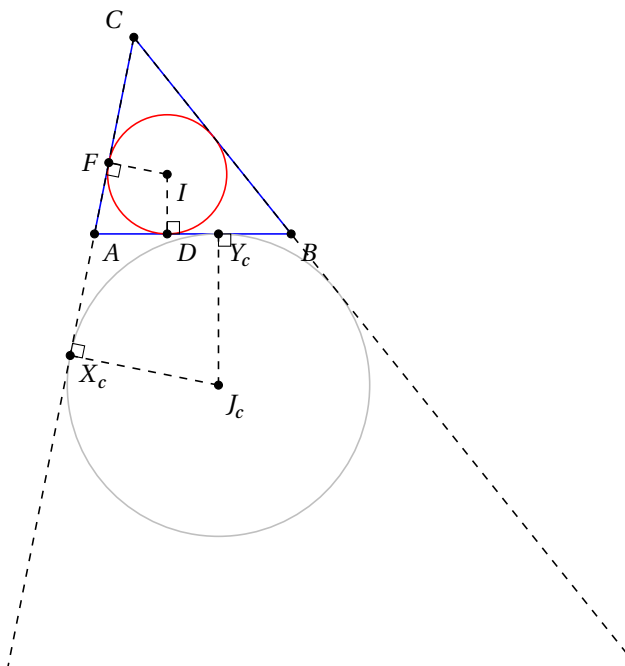
可以使用 `\tkzGetFirstPoint{I}` 和 `\tkzGetSecondPoint{Ib}` 命令得到内切圆在对应边上的投影。



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(2,2){A}
  \tkzDefPoint(5,-2){B}
  \tkzDefPoint(1,-2){C}
  \tkzDefCircle[in] (A,B,C)
  \tkzGetPoint{I} \tkzGetLength{rIN}
  \tkzDefCircle[circum] (A,B,C)
  \tkzGetPoint{K} \tkzGetLength{rCI}
  \tkzDrawPoints(A,B,C,I,K)
  \tkzDrawCircle[R,blue] (I,\rIN pt)
  \tkzDrawCircle[R,red] (K,\rCI pt)
  \tkzLabelPoints[below] (B,C)
  \tkzLabelPoints[above left] (A,I,K)
  \tkzDrawPolygon(A,B,C)
\end{tikzpicture}
```

### 1.1.4 ex选项示例

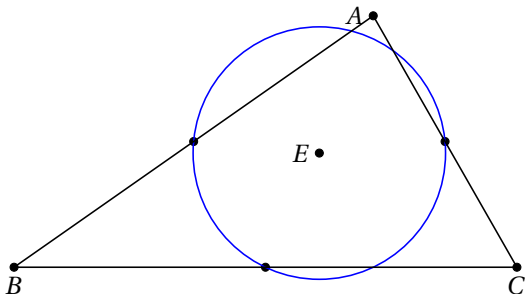
与顶点  $C$  对应的旁切圆。



```
\begin{tikzpicture}[scale=.65]
  \tkzDefPoints{ 0/0/A,4/0/B,0.8/4/C}
  \tkzDefCircle[ex] (B,C,A)
  \tkzGetPoint{J_c} \tkzGetLength{rc}
  \tkzDefPointBy[projection=onto A--C ](J_c)
  \tkzGetPoint{X_c}
  \tkzDefPointBy[projection=onto A--B ](J_c)
  \tkzGetPoint{Y_c}
  \tkzGetPoint{I}
  \tkzDrawPolygon[color=blue] (A,B,C)
  \tkzDrawCircle[R,color=lightgray](J_c,\rc pt)
  % possible \tkzDrawCircle[ex] (A,B,C)
  \tkzDrawCircle[in,color=red] (A,B,C) \tkzGetPoint{I}
  \tkzDefPointBy[projection=onto A--C ](I)
  \tkzGetPoint{F}
  \tkzDefPointBy[projection=onto A--B ](I)
  \tkzGetPoint{D}
  \tkzDrawLines[add=0 and 2.2,dashed] (C,A C,B)
  \tkzDrawSegments[dashed] (J_c,X_c I,D I,F J_c,Y_c)
  \tkzMarkRightAngles(A,F,I B,D,I J_c,X_c,A J_c,Y_c,B)
  \tkzDrawPoints(B,C,A,I,D,F,X_c,J_c,Y_c)
  \tkzLabelPoints(B,A,J_c,I,D,X_c,Y_c)
  \tkzLabelPoints[above left] (C)
  \tkzLabelPoints[left] (F)
\end{tikzpicture}
```

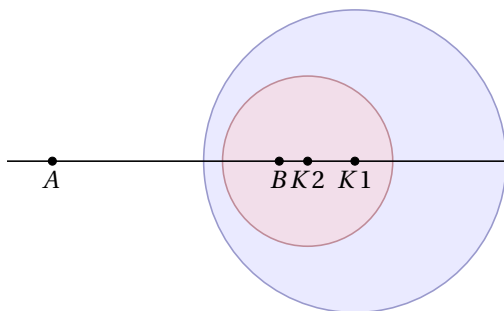
### 1.1.5 euler选项示例

同时验证了欧拉圆会通过三角形三个边的中点。



```
\begin{tikzpicture}[scale=.95]
  \tkzDefPoint(5,3.5){A}
  \tkzDefPoint(0,0){B}
  \tkzDefPoint(7,0){C}
  \tkzDefCircle[euler] (A,B,C)
  \tkzGetPoint{E}
  \tkzGetLength{rEuler}
  \tkzDefSpcTriangle[medial] (A,B,C){M_a,M_b,M_c}
  \tkzDrawPoints(A,B,C,E,M_a,M_b,M_c)
  \tkzDrawCircle[R,blue] (E,\rEuler pt)
  \tkzDrawPolygon(A,B,C)
  \tkzLabelPoints[below] (B,C)
  \tkzLabelPoints[left] (A,E)
\end{tikzpicture}
```

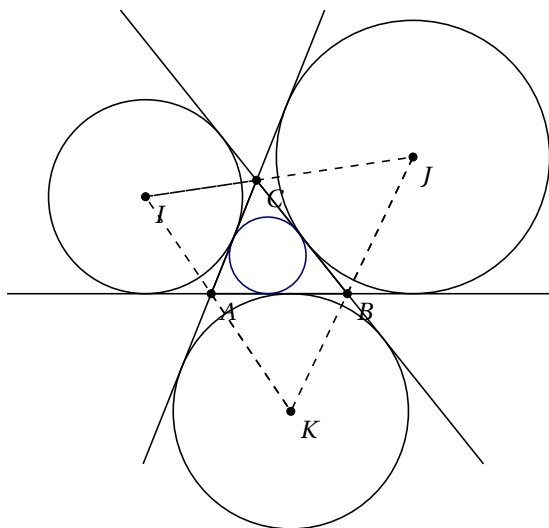
## 1.1.6 apollonius选项示例



```
\begin{tikzpicture}[scale=0.75]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefCircle[apollonius,K=2](A,B)
\tkzGetPoint{K1}
\tkzGetLength{rAp}
\tkzDrawCircle[R,color=blue!50!black,
  fill=blue!20,opacity=.4](K1,\rAp pt)
\tkzDefCircle[apollonius,K=3](A,B)
\tkzGetPoint{K2} \tkzGetLength{rAp}
\tkzDrawCircle[R,color=red!50!black,
  fill=red!20,opacity=.4](K2,\rAp pt)
\tkzLabelPoints[below](A,B,K1,K2)
\tkzDrawPoints(A,B,K1,K2)
\tkzDrawLine[add=.2 and 1](A,B)
\end{tikzpicture}
```

## 1.1.7 ex选项示例

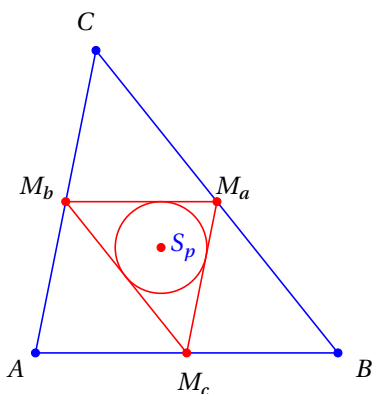
可以使用`\tkzGetFirstPoint{Jb}`和`\tkzGetSecondPoint{Tb}`命令，得到旁切圆圆心在边上的投影。



```
\begin{tikzpicture}[scale=.6]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,0){B}
\tkzDefPoint(1,2.5){C}
\tkzDefCircle[ex](A,B,C) \tkzGetPoint{I}
\tkzGetLength{rI}
\tkzDefCircle[ex](C,A,B) \tkzGetPoint{J}
\tkzGetLength{rJ}
\tkzDefCircle[ex](B,C,A) \tkzGetPoint{K}
\tkzGetLength{rK}
\tkzDefCircle[in](B,C,A) \tkzGetPoint{O}
\tkzGetLength{rO}
\tkzDrawLines[add=1.5 and 1.5](A,B A,C B,C)
\tkzDrawPoints(I,J,K)
\tkzDrawPolygon(A,B,C)
\tkzDrawPolygon[dashed](I,J,K)
\tkzDrawCircle[R,blue!50!black](O,\rO)
\tkzDrawSegments[dashed](A,K B,J C,I)
\tkzDrawPoints(A,B,C)
\tkzDrawCircles[R](J,{\rJ} I,{\rI} K,{\rK})
\tkzLabelPoints(A,B,C,I,J,K)
\end{tikzpicture}
```

### 1.1.8 spieker选项示例

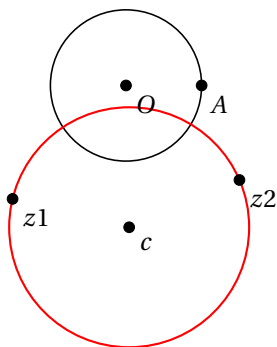
三角形三个边的中点构成的三角形  $M_a M_b M_c$  的内切圆是 Spieker 圆:



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{ 0/0/A,4/0/B,0.8/4/C}
  \tkzDefSpcTriangle[medial](A,B,C){M_a,M_b,M_c}
  \tkzDefTriangleCenter[spieker](A,B,C)
  \tkzGetPoint{S_p}
  \tkzDrawPolygon[blue](A,B,C)
  \tkzDrawPolygon[red](M_a,M_b,M_c)
  \tkzDrawPoints[blue](B,C,A)
  \tkzDrawPoints[red](M_a,M_b,M_c,S_p)
  \tkzDrawCircle[in,red](M_a,M_b,M_c)
  \tkzAutoLabelPoints[center=S_p,dist=.3](M_a,M_b,M_c)
  \tkzLabelPoints[blue,right](S_p)
  \tkzAutoLabelPoints[center=S_p](A,B,C)
\end{tikzpicture}
```

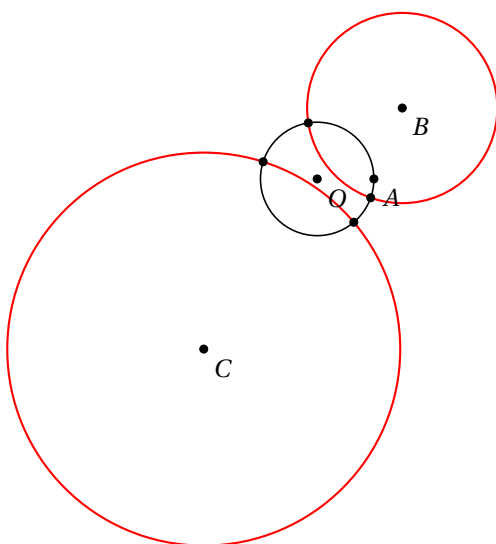
### 1.1.9 orthogonal through选项示例

过指定两个点的另一个圆的正交圆。



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){A}
  \tkzDrawCircle(O,A)
  \tkzDefPoint(-1.5,-1.5){z1}
  \tkzDefPoint(1.5,-1.25){z2}
  \tkzDefCircle[orthogonal through=z1 and z2](O,A)
  \tkzGetPoint{c}
  \tkzDrawCircle[thick,color=red](tkzPointResult,z1)
  \tkzDrawPoints[fill=red,color=black,
    size=4](O,A,z1,z2,c)
  \tkzLabelPoints(O,A,z1,z2,c)
\end{tikzpicture}
```

### 1.1.10 指定圆心的另一个圆的正交圆示例



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O,1/0/A}
  \tkzDefPoints{1.5/1.25/B,-2/-3/C}
  \tkzDefCircle[orthogonal from=B](O,A)
  \tkzGetPoints{z1}{z2}
  \tkzDefCircle[orthogonal from=C](O,A)
  \tkzGetPoints{t1}{t2}
  \tkzDrawCircle(O,A)
  \tkzDrawCircle[thick,color=red](B,z1)
  \tkzDrawCircle[thick,color=red](C,t1)
  \tkzDrawPoints(t1,t2,C)
  \tkzDrawPoints(z1,z2,O,A,B)
  \tkzLabelPoints(O,A,B,C)
\end{tikzpicture}
```

## 2 圆的绘制和标注

- `\tkzDrawCircle`命令用于绘制一个圆,
- `\tkzFillCircle`命令用于在不绘制圆的情况下, 对圆进行着色,
- `\tkzClipCircle`命令用于设置圆形裁剪区域,
- `\tkzLabelCircle`命令用于对圆进行标注.

2.1 `\tkzDrawCircle`命令: 绘制圆

`\tkzDrawCircle[< 命令选项>](A,B)`

注意: 只能用两个点指定半径或直径, 使用`R`选项, 则需要直接指定半径。

参数	样例	说明
<code>(\pt1,\pt2)</code>	<code>(A,B)</code>	两个点定义半径或直径

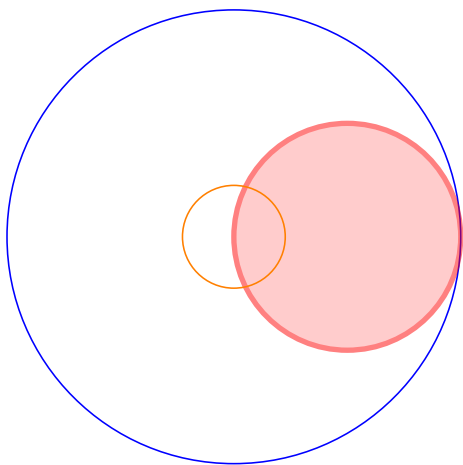
  

选项	默认值	定义
<code>through</code>	<code>through</code>	两个点定义半径
<code>diameter</code>	<code>through</code>	两个点定义直径
<code>R</code>	<code>through</code>	需要指定半径

可使用所有有效 TikZ 样式。

## 2.1.1 绘制一个圆并对其进行着色

能够在绘制中对圆进行着色。



```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(3,0){A}
% 圆心是 O, 通过 A 点
\tkzDrawCircle[color=blue](O,A)
% 直径是 $[OA]$
\tkzDrawCircle[diameter,color=red,%
               line width=2pt,fill=red!40,%
               opacity=.5](O,A)
% 圆心是 O, 半径 =exp(1) cm
\edef\rayon{\fpeval{0.25*exp(1)}}
\tkzDrawCircle[R,color=orange](O,\rayon cm)
\end{tikzpicture}
```

## 2.2 \tkzDrawCircles命令：绘制多个圆

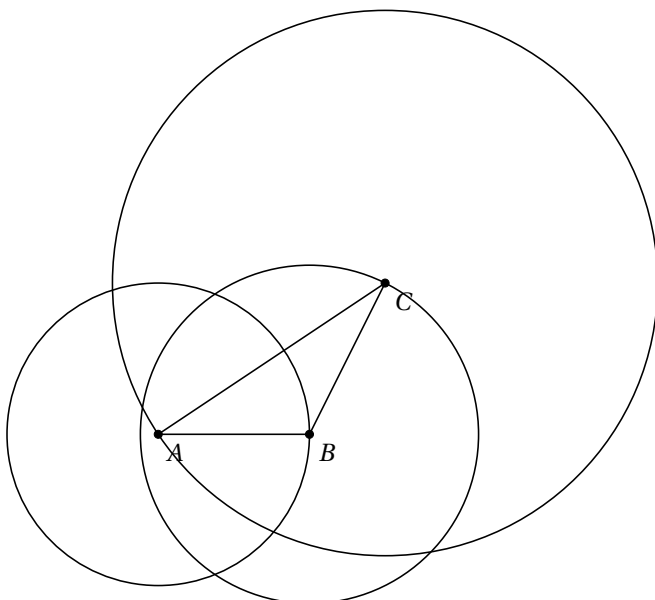
`\tkzDrawCircles[< 命令选项>](⟨A,B C,D⟩)`

注意：参数是空格分隔的构成圆的点对列表，点对中的两个点之间用逗号分隔。使用**R**选项，则需要直接指定半径。

参数	样例	说明
<code>(⟨pt1,pt2 pt3,pt4,...⟩)</code>	<code>(⟨A,B C,D⟩)</code>	点集列表
选项	默认值	含义
<code>through</code>	<code>through</code>	两个点定义半径
<code>diameter</code>	<code>through</code>	两个点定义直径
<code>R</code>	<code>through</code>	通过指定半径

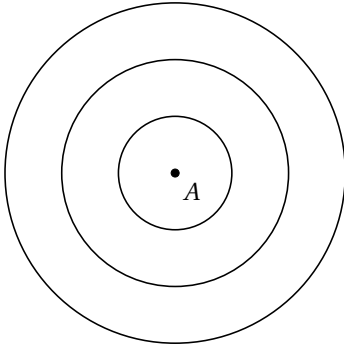
当然，可以使用所有的有效 TikZ 样式。

## 2.2.1 通过三角形定义圆示例



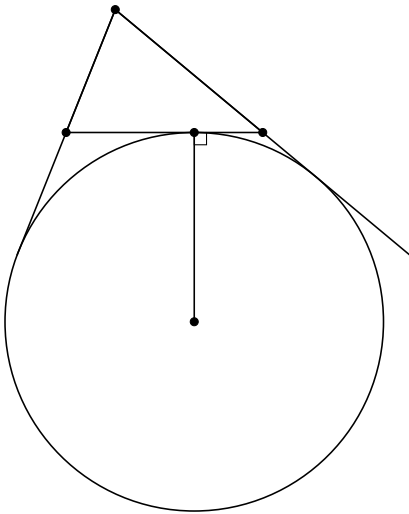
```
\begin{tikzpicture}[scale=1.0]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,0){B}
\tkzDefPoint(3,2){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawCircles(A,B B,C C,A)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

## 2.2.2 同心圆示例



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){A}
  \tkzDrawCircles[R](A,1cm A,2cm A,3cm)
  \tkzDrawPoint(A)
  \tkzLabelPoints(A)
\end{tikzpicture}
```

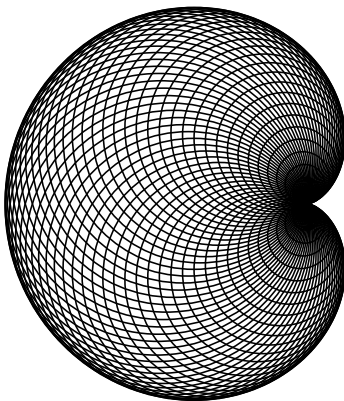
## 2.2.3 旁切圆示例



```
\begin{tikzpicture}[scale=0.65]
  \tkzDefPoints{0/0/A,4/0/B,1/2.5/C}
  \tkzDrawPolygon(A,B,C)
  \tkzDefCircle[ex](B,C,A)
  \tkzGetPoint{J_c} \tkzGetSecondPoint{T_c}
  \tkzGetLength{rJc}
  \tkzDrawCircle[R](J_c,{\rJc pt})
  \tkzDrawLines[add=0 and 1](C,A C,B)
  \tkzDrawSegment(J_c,T_c)
  \tkzMarkRightAngle(J_c,T_c,B)
  \tkzDrawPoints(A,B,C,J_c,T_c)
\end{tikzpicture}
```

## 2.2.4 心形线示例

基于 O. Reboux 用 D. Rodriguez 开发的 `pst-eucl` 宏包绘制的图形进行绘制。名称来源于希腊语中的 *kardia* (*heart*), 是根据其形状命名的。这个名称最先是由 Johan Castillon 给出的 (Wikipedia)。



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,0){A}
  \foreach \ang in {5,10,...,360}{%
    \tkzDefPoint(\ang:2){M}
    \tkzDrawCircle(M,A)
  }
\end{tikzpicture}
```

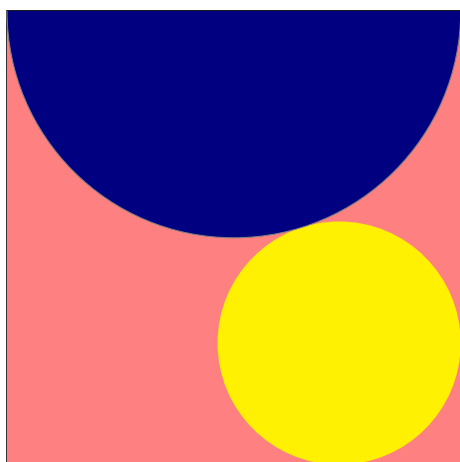


2.3 `\tkzDrawSemiCircle`命令：绘制半圆

`\tkzDrawSemiCircle[< 命令选项>](<A,B>)`

参数	样例	说明
<code>(&lt;pt1,pt2&gt;)</code>	<code>(&lt;O,A&gt;)</code> or <code>(&lt;A,B&gt;)</code>	半径或直径
选项	默认值	含义
<code>through</code>	<code>through</code>	两个点定义半径
<code>diameter</code>	<code>through</code>	两个点定义直径

## 2.3.1 示例



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(6,0){B}
\tkzDefSquare(A,B) \tkzGetPoints{C}{D}
\tkzDrawPolygon(B,C,D,A)
\tkzDefPoint(3,6){F}
\tkzDefTriangle[equilateral](C,D) \tkzGetPoint{I}
\tkzDefPointBy[projection=onto B--C](I) \tkzGetPoint{J}
\tkzInterLL(D,B)(I,J) \tkzGetPoint{K}
\tkzDefPointBy[symmetry=center K](B) \tkzGetPoint{M}
\tkzDrawCircle(M,I)
\tkzCalcLength(M,I) \tkzGetLength{dMI}
\tkzFillPolygon[color = red!50](A,B,C,D)
\tkzFillCircle[R,color = yellow](M,\dMI pt)
\tkzDrawSemiCircle[fill = blue!50!black](F,D)%
\end{tikzpicture}
```

2.4 `\tkzFillCircle`命令：给圆着色

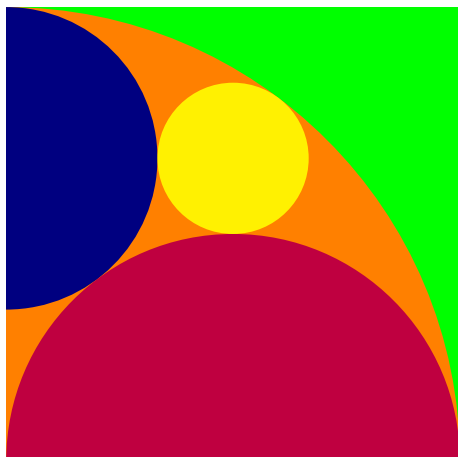
在绘制圆时，也可以实现着色，但该命令不绘制圆，仅对圆形区域进行着色。

`\tkzFillCircle[< 命令选项>](<A,B>)`

选项	默认值	含义
<code>radius</code>	<code>radius</code>	两个点定义半径
<code>R</code>	<code>radius</code>	需要定义半径

可以使用所有有效 TikZ 样式。

## 2.4.1 sangaku 圆示例



```
\begin{tikzpicture}
  \tkzInit[xmin=0,xmax = 6,ymin=0,ymax=6]
  \tkzDefPoint(0,0){B}   \tkzDefPoint(6,0){C}%
  \tkzDefSquare(B,C)      \tkzGetPoints{D}{A}
  \tkzClipPolygon(B,C,D,A)
  \tkzDefMidPoint(A,D)   \tkzGetPoint{F}
  \tkzDefMidPoint(B,C)   \tkzGetPoint{E}
  \tkzDefMidPoint(B,D)   \tkzGetPoint{Q}
  \tkzDefTangent[from = B](F,A) \tkzGetPoints{G}{H}
  \tkzInterLL(F,G)(C,D) \tkzGetPoint{J}
  \tkzInterLL(A,J)(F,E) \tkzGetPoint{K}
  \tkzDefPointBy[projection=onto B--A](K)
  \tkzGetPoint{M}
  \tkzFillPolygon[color = green](A,B,C,D)
  \tkzFillCircle[color = orange](B,A)
  \tkzFillCircle[color = blue!50!black](M,A)
  \tkzFillCircle[color = purple](E,B)
  \tkzFillCircle[color = yellow](K,Q)
\end{tikzpicture}
```

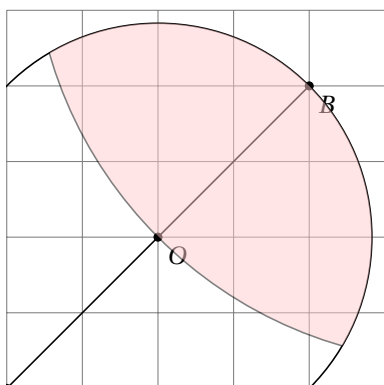
## 2.5 \tkzClipCircle命令：用圆裁剪

`\tkzClipCircle[< 命令选项>](A,B)` 或 `(A,r)`

参数	样例	说明
<code>(A,B)</code> or <code>(A,r)</code>	<code>(A,B)</code> 或 <code>(A,2cm)</code>	AB 是半径或直径

选项	默认值	含义
radius	radius	两个点确定半径
R	radius	指定半径

## 2.5.1 示例



```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzGrid
  \tkzClip
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(2,2){O}
  \tkzDefPoint(4,4){B}
  \tkzDefPoint(6,6){C}
  \tkzDrawPoints(O,A,B,C)
  \tkzLabelPoints(O,A,B,C)
  \tkzDrawCircle(O,A)
  \tkzClipCircle(O,A)
  \tkzDrawLine(A,C)
  \tkzDrawCircle[fill=red!20,opacity=.5](C,O)
\end{tikzpicture}
```

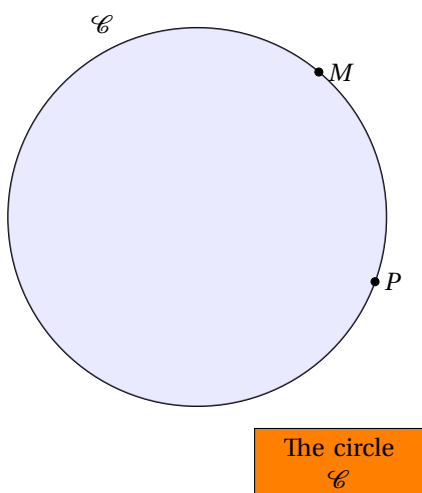
2.6 `\tkzLabelCircle`命令：为圆添加标注

`\tkzLabelCircle`[< 命令选项>](<A,B>)(< 角度>){< 标注>}

选项	默认值	含义
radius	radius	两个点确定半径
R	radius	指定半径

可以使用所有有效 TikZ 样式，标注内容通过“传递”给大括号中的参数指定。

## 2.6.1 标注示例



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,0){N}
\tkzDefPointBy[rotation=center O angle 50](N)
\tkzGetPoint{M}
\tkzDefPointBy[rotation=center O angle -20](N)
\tkzGetPoint{P}
\tkzDefPointBy[rotation=center O angle 125](N)
\tkzGetPoint{P'}
\tkzLabelCircle[above=4pt](O,N)(120){\mathcal{C}}
\tkzDrawCircle(O,M)
\tkzFillCircle[color=blue!20,opacity=.4](O,M)
\tkzLabelCircle[R,draw,fill=orange, text width=2cm,
text centered](O,3 cm)(-60)%
{The circle\ \mathcal{C}}
\tkzDrawPoints(M,P)
\tkzLabelPoints[right](M,P)
\end{tikzpicture}
```