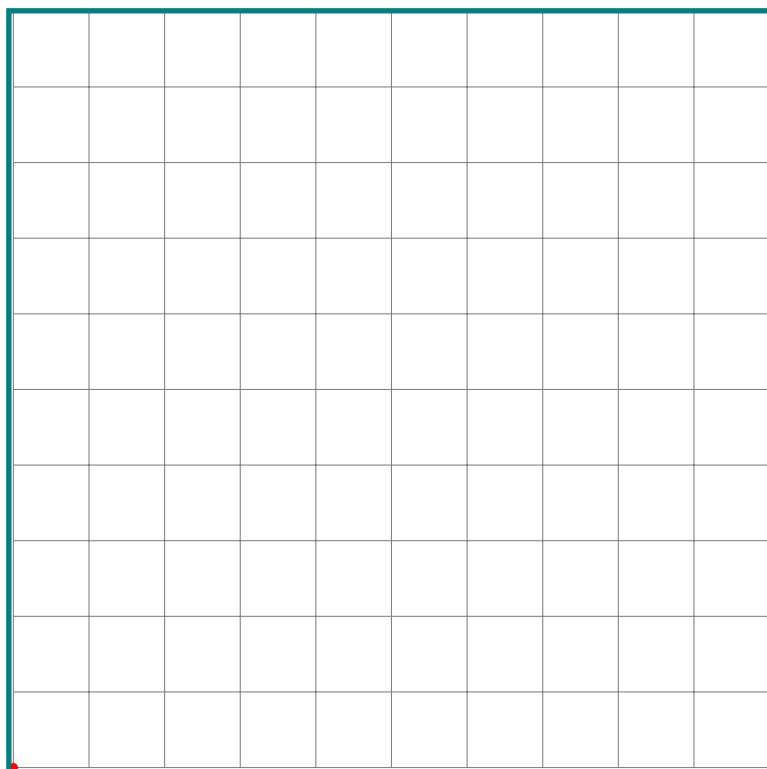


1 点的定义

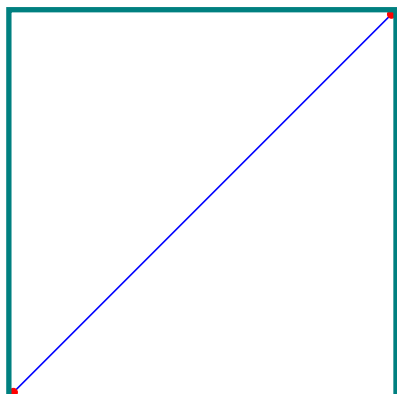
可以通过如下方式定义点:

- 笛卡尔坐标点;
- 极坐标点;
- 命名点;
- 相对点;

虽然可以使用坐标定义点, 但建议使用命名点定义。如果一个名称唯一与一对十进制数对应, 则可以为该点命名, 如 (x,y) 或 $(a:d)$, 其中 x 是横坐标, y 是纵坐标; a 是角度, d 是距离。这是因为正交笛卡尔坐标系定义了一个平面, 其坐标轴是正交的, 并且度量单位等于 1 cm 或 0.39370 in 。现在, 默认使用网格或坐标轴, 则矩形由 $(0,0)$ 和 $(10,10)$ 两个坐标定义。可以用 `tkz-base` 宏包的 `\tkzInit` 命令创建该矩形, 可以查看两个代码及其绘制结果。



```
\begin{tikzpicture}
\tkzGrid
\tkzDefPoint(0,0){O}
\tkzDrawPoint[red](O)
\tkzShowBB[line width=2pt,teal]
\end{tikzpicture}
```

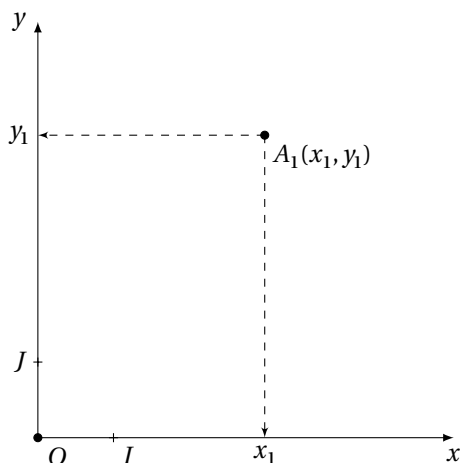


```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(5,5){A}
\tkzDrawSegment[blue](O,A)
\tkzDrawPoints[red](O,A)
\tkzShowBB[line width=2pt,teal]
\end{tikzpicture}
```

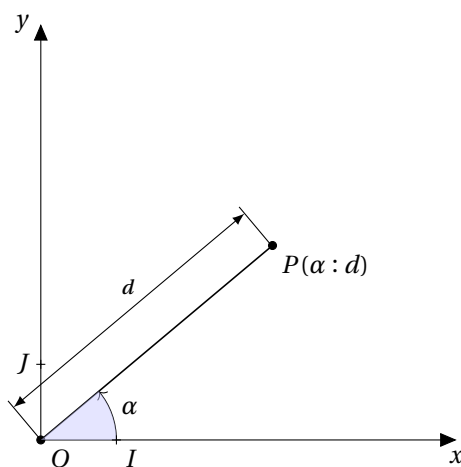
笛卡尔坐标 (a,b) 表示该点的 x 坐标是 $a\text{ cm}$, y 坐标是 $b\text{ cm}$ 。

极坐标中表示一个点需要一个角度 α (度) 和一个从原点度量的距离 d (默认单位是 cm)。

笛卡尔坐标



极坐标



```
\begin{tikzpicture}[scale=1]
\tkzInit[xmax=5,ymax=5]
\tkzDefPoints{0/0/0,1/0/I,0/1/J}
\tkzDrawXY[noticks,>=latex]
\tkzDefPoint(3,4){A}
\tkzDrawPoints(0,A)
\tkzLabelPoint(A){$A_1 (x_1,y_1)$}
\tkzShowPointCoord[xlabel=$x_1$,
ylabel=$y_1$](A)
\tkzLabelPoints(0,I)
\tkzLabelPoints[left](J)
\tkzDrawPoints[shape=cross](I,J)
\end{tikzpicture}
```

```
\begin{tikzpicture}[scale=1]
\tkzInit[xmax=5,ymax=5]
\tkzDefPoints{0/0/0,1/0/I,0/1/J}
\tkzDefPoint(40:4){P}
\tkzDrawXY[noticks,>=triangle 45]
\tkzDrawSegment[dim={d$,
16pt,above=6pt}](0,P)
\tkzDrawPoints(0,P)
\tkzMarkAngle[mark=none,->](I,0,P)
\tkzFillAngle[fill=blue!20,
opacity=.5](I,0,P)
\tkzLabelAngle[pos=1.25](I,0,P){$\alpha$}
\tkzLabelPoint(P){$P (\alpha : d)$}
\tkzDrawPoints[shape=cross](I,J)
\tkzLabelPoints(0,I)
\tkzLabelPoints[left](J)
\end{tikzpicture}
```

在命令中

用`\tkzDefPoint`命令，通过指定坐标值定义一个点。该命令源于TikZ的`\coordinate`命令，因此，可以使用类似`shift`的TikZ的参数。该命令使用`xfp`宏包实现必要的计算。在定义点时，既可以使用笛卡尔坐标，也可以使用极坐标。

1.1 \tkzDefPoint命令：定义命名点

`\tkzDefPoint[< 命令选项>](<x,y>){< 名称>} or (<alpha:d>){< 名称>}`

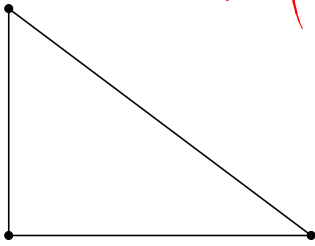
参数	默认值	含义
(x,y)	无	x 和 y 分别是 2 维坐标，默认单位是 cm.
(alpha:d)	无	alpha 是角度 (度), d 是距离 (cm)
{名称}	无	点的名称，如: A, T _a , P1, ...

该命令的必选参数是十进制表示的 2 维坐标值，笛卡尔坐标表示两个长度值，极坐标表示角度和距离。

选项	默认值	含义
label	无	按预设的距离添加标注
shift	无	为 (x,y) 或 (alpha:d) 添加偏移

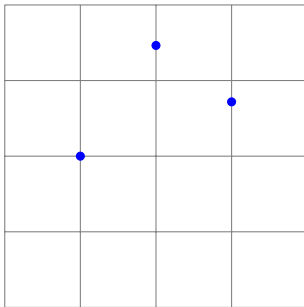
1.1.1 笛卡尔坐标

示例



```
\begin{tikzpicture}
\tkzInit[xmax=5,ymax=5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(0,3){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

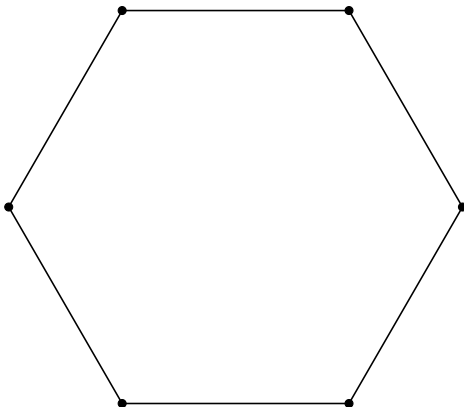
1.1.2 使用xfp宏包实现计算



```
\begin{tikzpicture}[scale=1]
\tkzInit[xmax=4,ymax=4]
\tkzGrid
\tkzDefPoint(-1+2,sqrt(4)){O}
\tkzDefPoint({3*ln(exp(1))},{exp(1))}{A}
\tkzDefPoint({4*sin(pi/6)},{4*cos(pi/6)}){B}
\tkzDrawPoints[color=blue](O,B,A)
\end{tikzpicture}
```

1.1.3 极坐标

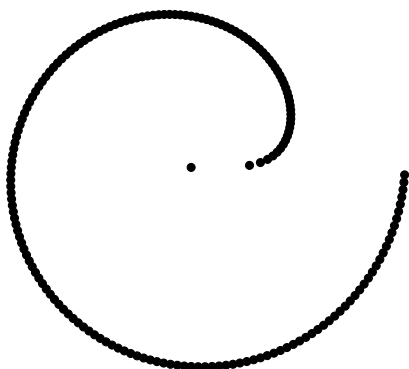
示例



```
\begin{tikzpicture}
\foreach \an [count=\i] in {0,60,...,300}
{\tkzDefPoint(\an:3){A_\i}}
\tkzDrawPolygon(A_1,A_2,...,A_6)
\tkzDrawPoints(A_1,A_2,...,A_6)
\end{tikzpicture}
```

1.1.4 计算和坐标

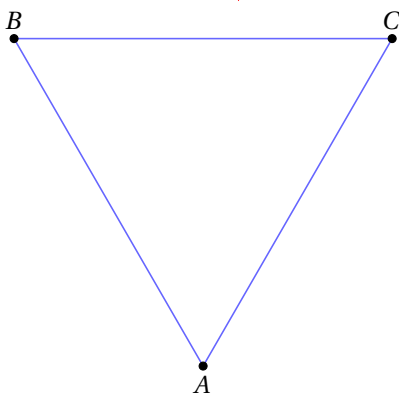
在计算时,需要遵循`xfp`宏包的语法。另外,也有可能用至`pgfmath`宏包进行计算,此时,需要在使用`\tkzDefPoint`命令之前完成计算。



```
\begin{tikzpicture}[scale=.5]
\foreach \an [count=\i] in {0,2,...,358}
{\tkzDefPoint(\an:sqrt(sqrt(\an mm))){A_\i}}
\tkzDrawPoints(A_1,A_...A_180)
\end{tikzpicture}
```

1.1.5 相对点

使用相对点时,需要使用 TikZ 的`scope`环境。下面的示例代码给出了一种定义等边三角形的方法:



```
\begin{tikzpicture}[scale=1]
\tkzSetUpLine[color=blue!60]
\begin{scope}[rotate=30]
\tkzDefPoint(2,3){A}
\begin{scope}[shift=(A)]
\tkzDefPoint(90:5){B}
\tkzDefPoint(30:5){C}
\end{scope}
\end{scope}
\tkzDrawPolygon(A,B,C)
\tkzLabelPoints[above](B,C)
\tkzLabelPoints[below](A)
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

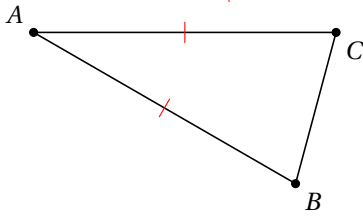
1.2 `\tkzDefShiftPoint` 命令: 通过平移定义点

`\tkzDefShiftPoint[< 参考点>](<x,y>){< 名称>} 或 (< α :d>){< 名称>}`

参数	默认值	含义
(x,y)	无	x 和 y 是 2 维坐标, 默认单位是 cm.
$(\alpha:d)$	无	α 是角度 (度), d 是距离
选项	默认值	含义
[参考点]	无	例如: <code>\tkzDefShiftPoint[A](0:4){B}</code>

1.2.1 使用 `\tkzDefShiftPoint` 命令定义等腰三角形

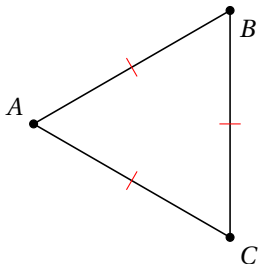
This macro allows you to place one point relative to another. This is equivalent to placing a point at a distance r from a point A at an angle α relative to the horizontal. The command allows you to define a point relative to another point, equivalent to translating the point. The following code gives a method for defining an isosceles triangle through point A and a 30° angle.



```
\begin{tikzpicture}[rotate=-30]
\tkzDefPoint(2,3){A}
\tkzDefShiftPoint[A](0:4){B}
\tkzDefShiftPoint[A](30:4){C}
\tkzDrawSegments(A,B B,C C,A)
\tkzMarkSegments[mark=|,color=red](A,B A,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(B,C)
\tkzLabelPoints[above left](A)
\end{tikzpicture}
```

1.2.2 等边三角形

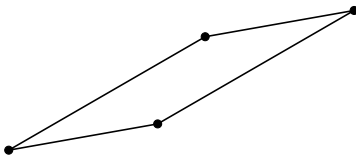
The following code gives a very simple method for defining an equilateral triangle.



```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(2,3){A}
\tkzDefShiftPoint[A](30:3){B}
\tkzDefShiftPoint[A](-30:3){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(B,C)
\tkzLabelPoints[above left](A)
\tkzMarkSegments[mark=|,color=red](A,B A,C B,C)
\end{tikzpicture}
```

1.2.3 平行四边形

A simple way to define a parallelogram is:



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(30:3){B}
\tkzDefShiftPointCoord[B](10:2){C}
\tkzDefShiftPointCoord[A](10:2){D}
\tkzDrawPolygon(A,...,D)
\tkzDrawPoints(A,...,D)
\end{tikzpicture}
```

1.3 `\tkzDefPoints` 命令: 定义点集

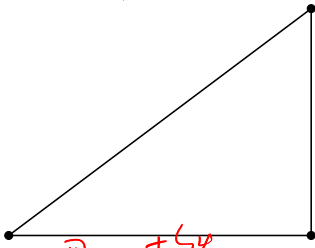
`\tkzDefPoints[<命令选项>]{(x1/y1/n1,x2/y2/n2,...)}`

x_i 和 y_i 是 n_i 点的 2 维坐标

参数	默认值	样例
$x_i/y_i/n_i$	无	<code>\tkzDefPoints{0/0/0,2/2/A}</code>
选项	默认值	含义
shift	无	为所有点添加 (x,y) 或 $(\alpha:d)$ 偏移

2.1 用---构造

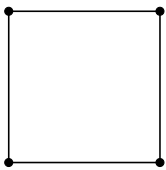
1.4 创建三角形



```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{0/0/A,4/0/B,4/3/C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

1.5 创建正方形

注意该代码中绘制多边形的语法。



```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{0/0/A,2/0/B,2/2/C,0/2/D}
\tkzDrawPolygon(A,...,D)
\tkzDrawPoints(A,B,C,D)
\end{tikzpicture}
```

2 特殊点

tkz-base宏包中点的定义方法就有这些，其中，最为重要的是`\tkzDefPoint`命令。当然，还有一些特殊点的定义方法。

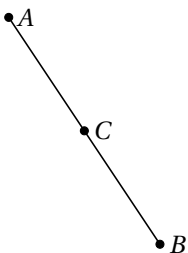
2.1 \tkzDefMidPoint命令：定义线段的中点

```
\tkzDefMidPoint(<pt1,pt2>)
```

该命令定义的点存储于`\tkzPointResult`命令中，可以通过`\tkzGetPoint`命令得到该点。

参数	默认值	含义
(pt1,pt2)	无	pt1 和 pt2 是线段的两个端点

2.1.1 使用\tkzDefMidPoint命令



```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(2,3){A}
\tkzDefPoint(4,0){B}
\tkzDefMidPoint(A,B) \tkzGetPoint{C}
\tkzDrawSegment(A,B)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints[right](A,B,C)
\end{tikzpicture}
```

2.2 重心坐标

设有 pt_1, pt_2, \dots, pt_n 共计 n 个点，则它们定义了 n 个向量 $\vec{v_1}, \vec{v_2}, \dots, \vec{v_n}$ 。令 $\alpha_1, \alpha_2, \dots, \alpha_n$ 是 n 常数，因此可按下式得到一个向量：

$$\frac{\alpha_1 \vec{v_1} + \alpha_2 \vec{v_2} + \dots + \alpha_n \vec{v_n}}{\alpha_1 + \alpha_2 + \dots + \alpha_n}$$

则，该向量可以定义一个点。

`\tkzDefBarycentricPoint(<pt1= α_1 ,pt2= α_2 ,...>)`

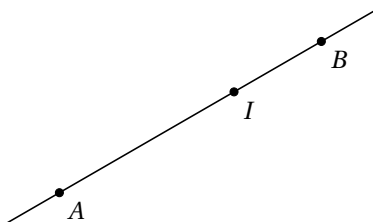
参数	默认值	含义
<code>(pt1=α_1,pt2=α_2,...)</code>	无	每个点的权重

注意：至少需要两个已知点，才能实现计算。

2.2.1 用 `\tkzDefBarycentricPoint` 命令计算两个点的重心

下面的代码中，通过系数 1 和 2 得到了点 A 和点 B 的重心：

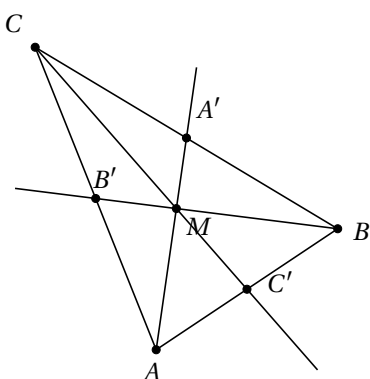
$$\overrightarrow{AI} = \frac{2}{3}\overrightarrow{AB}$$



```
\begin{tikzpicture}
\tkzDefPoint(2,3){A}
\tkzDefShiftPointCoord[2,3](30:4){B}
\tkzDefBarycentricPoint(A=1,B=2)
\tkzGetPoint{I}
\tkzDrawPoints(A,B,I)
\tkzDrawLine(A,B)
\tkzLabelPoints(A,B,I)
\end{tikzpicture}
```

2.2.2 用 `\tkzDefBarycentricPoint` 命令计算 3 个点的重心

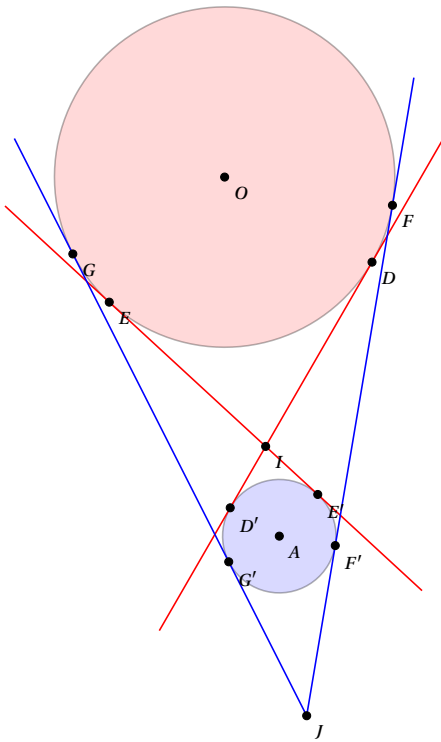
下面的代码中，M 是三角形的重心。当然，为了简化操作，还有另外一个用于计算三角形重心的 `\tkzCentroid` 命令。



```
\begin{tikzpicture}[scale=.8]
\tkzDefPoint(2,1){A}
\tkzDefPoint(5,3){B}
\tkzDefPoint(0,6){C}
\tkzDefBarycentricPoint(A=1,B=1,C=1)
\tkzGetPoint{M}
\tkzDefMidPoint(A,B) \tkzGetPoint{C'}
\tkzDefMidPoint(A,C) \tkzGetPoint{B'}
\tkzDefMidPoint(C,B) \tkzGetPoint{A'}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A',B',C')
\tkzDrawPoints(A,B,C,M)
\tkzDrawLines[add=0 and 1](A,M B,M C,M)
\tkzLabelPoint(M){M}
\tkzAutoLabelPoints[center=M](A,B,C)
\tkzAutoLabelPoints[center=M,above right](A',B',C')
\end{tikzpicture}
```

2.3 内部相似中心

两个圆对应的两个同构物的中心称为外部和内部相似中心。



```

\begin{tikzpicture}[scale=.75,rotate=-30]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(4,-5){A}
  \tkzDefIntSimilitudeCenter(0,3)(A,1)
  \tkzGetPoint{I} 内部相似中心
  \tkzExtSimilitudeCenter(0,3)(A,1)
  \tkzGetPoint{J} 外部相似中心
  \tkzDefTangent[from with R = I](0,3 cm)
  \tkzGetPoints{D}{E}
  \tkzDefTangent[from with R = I](A,1 cm)
  \tkzGetPoints{D'}{E'}
  \tkzDefTangent[from with R = J](0,3 cm)
  \tkzGetPoints{F}{G}
  \tkzDefTangent[from with R = J](A,1 cm)
  \tkzGetPoints{F'}{G'}
  \tkzDrawCircle[R,fill=red!50,opacity=.3](0,3 cm)
  \tkzDrawCircle[R,fill=blue!50,opacity=.3](A,1 cm)
  \tkzDrawSegments[add = .5 and .5,color=red](D,D' E,E')
  \tkzDrawSegments[add= 0 and 0.25,color=blue](J,F J,G)
  \tkzDrawPoints(O,A,I,J,D,E,F,G,D',E',F',G')
  \tkzLabelPoints[font=\scriptsize](O,A,I,J,D,E,F,G,D',
    E',F',G')
\end{tikzpicture}

```