

AlterMundus



著: Alain Matthes


译: 耿楠 (陕西·杨凌)

December 31, 2020 Documentation V.3.06c
<http://altermundus.fr>

tkz-euclide

AlterMundus

Alain Matthes

 **tkz-euclide** 是一个用于在笛卡尔坐标系中绘制平面图形 (点、线、三角形、圆等基本二维对象) 的尽可能方便使用的宏集, 它主要用于绘制经典欧几里得图形。**tkz-euclide** 是基于 PGF 设计的, 可以看作是一个 TikZ 的前端, 并且是符合 (La)TeX 语法的绘图宏包。其目标是提供一个高层的用户接口, 以 ~~实现~~ 相对简单语法进行绘图。它使用 **tkz-base** 宏包提供的正交笛卡尔坐标系和点的定义及点的操作宏命令为基础进行绘图。该宏包的基本思路是尽可能符合欧氏几何中手工绘图的绘图方式。

该宏包需要 TikZ3.0 及以上版本支持, 另外, 由于英语不是母语, 该说明文档中, 可能会存在描述错误。

 首先, 感谢 Till Tantau 开发了强大的 TikZ 绘图工具 **TikZ**

 另外, 在宏包开发中, 收到了大量有价值的建议、修订、勘误和排版样例。它们主要来自 Jean-Côme Charpentier, Josselin Noirel, Manuel Pégourié-Gonnard, Franck Pastor, David Arnold, Ulrike Fischer, Stefan Kottwitz, Christian Tellechea, Nicolas Kisselhoff, David Arnold, Wolfgang Büchel, John Kitzmiller, Dimitri Kapetas, Gaétan Marris, Mark Wibrow, Yves Combe (量角器样例), Paul Gaborit and Laurent.

 感谢 MathWorld 的创造人 Eric Weisstein: [MathWorld](#)

如果发现该文档的错误或有其他任何意见和建议, 请发信至: [Alain Matthes](#).

如果发现译文的错误或其他任何意见和建议, 请发信至: [耿楠](#).

可以在 [CTAN](#) 发布的 “LATEX Project Public License” 协议下发布和修改该文档。

Contents

1	概述	10
1.1	为什么要开发tkz-euclide?	10
1.2	tkz-euclide与TikZ	10
1.3	工作模式	10
1.3.1	示例 I: 黄金三角形	10
1.3.2	示例 II: 另外两种方式	12
1.3.3	最小工作示例	13
1.4	tkz 代码的基本要素	16
1.5	符号和约定	16
1.6	使用tkz-euclide宏包	18
1.6.1	一个经典示例	18
1.6.2	Set, Calculate, Draw, Mark, Label	19
2	安装	20
2.1	tkzbase和tkzeuclide目录中的文件	20
3	新特性与兼容性	22
4	点的定义	23
4.1	\tkzDefPoint命令: 定义命名点	24
4.1.1	笛卡尔坐标	25
4.1.2	使用xfp宏包实现计算	25
4.1.3	极坐标	25
4.1.4	计算和坐标	26
4.1.5	相对点	26
4.2	\tkzDefShiftPoint命令: 通过平移定义点	26
4.2.1	使用\tkzDefShiftPoint命令定义等腰三角形	27
4.2.2	等边三角形	27
4.2.3	平行四边形	27
4.3	\tkzDefPoints命令: 定义点集	27
4.4	创建三角形	28
4.5	创建正方形	28
5	特殊点	28
5.1	\tkzDefMidPoint命令: 定义线段的中点	28
5.1.1	使用\tkzDefMidPoint命令	28
5.2	重心坐标	29
5.2.1	用\tkzDefBarycentricPoint命令计算两个点的重心	29
5.2.2	用\tkzDefBarycentricPoint命令计算3个点的重心	29
5.3	内部相似中心	30
6	与三角形相关的特殊点	31
6.1	\tkzDefTriangleCenter命令: 获取三角形中心	31
6.1.1	ortho或orthic选项	31
6.1.2	centroid选项	32
6.1.3	circum选项	32
6.1.4	in选项	32
6.1.5	ex选项	33
6.1.6	euler选项	33
6.1.7	symmedian选项	34
6.1.8	nagel选项	34
6.1.9	mittenpunkt选项	35
7	点的绘制	35
7.0.1	使用\tkzDrawPoint命令绘制点	35
7.0.2	点的绘制实例	35
7.0.3	第1个示例代码	36
7.0.4	第2个示例代码	36

8	直线或圆上的点	36
8.1	直线上的点	36
8.1.1	pos 选项	37
8.2	圆上的点	37
9	利用变换命令 <code>\tkzDefPointBy</code> 定义点	38
9.1	变换示例	39
9.1.1	平移示例	39
9.1.2	反射示例 (orthogonal symmetry)	39
9.1.3	homothety 和 projection 示例	40
9.1.4	投影示例	40
9.1.5	对称示例	41
9.1.6	旋转示例 (度)	41
9.1.7	旋转示例 (弧度)	41
9.1.8	点反转示例	42
9.1.9	点的反转: 正交圆	42
9.2	用 <code>\tkzDefPointsBy</code> 命令实现多点变换	42
9.2.1	变换示例	43
10	通过向量定义点	44
10.1	<code>\tkzDefPointWith</code> 命令	44
10.1.1	colinear at 选项	44
10.1.2	colinear at 带 K 选项	45
10.1.3	colinear at 带 $K = \frac{\sqrt{2}}{2}$ 选项	45
10.1.4	orthogonal	45
10.1.5	orthogonal 带 $K = -1$ 选项	45
10.1.6	orthogonal 选项的综合实例	46
10.1.7	colinear 和 orthogonal 选项	46
10.1.8	orthogonal normed 带 $K = 1$ 选项	46
10.1.9	orthogonal normed 和 $K = 2$ 选项	47
10.1.10	linear 选项	47
10.1.11	linear normed 选项	47
10.2	<code>\tkzGetVectxy</code> 命令	47
10.2.1	使用 <code>\tkzGetVectxy</code> 命令实现坐标变换	48
11	定义随机点	49
11.1	得到随机点	49
11.2	矩形内的随机点	49
11.3	线段上的随机点	50
11.4	直线上的随机点	50
11.4.1	随机点综合实例	50
11.5	圆上的随机点	51
11.5.1	Apollonius 圆的随机示例	51
11.6	罗盘线段中点	52
12	直线	53
12.1	定义直线	53
12.1.1	mediator 选项	53
12.1.2	bisector 和 normed 选项	54
12.1.3	orthogonal 和 parallel 选项	54
12.1.4	循环画图	54
12.1.5	抛物线	55
12.2	圆的切线	55
12.2.1	圆上指定点的切线	55
12.2.2	过圆外指定点的切线	56
12.2.3	Andrew Mertz 示例	56
12.2.4	from with R 和 at 选项	56
12.2.5	from 选项	57

13 绘制和命名直线	57
13.1 绘制直线	57
13.1.1 <code>add</code> 选项	58
13.1.2 <code>\tkzDrawLines</code> 选项	58
13.1.3 <code>add</code> 选项	58
13.1.4 三角形的中线	59
13.1.5 三角形的高	59
13.1.6 三角形的角平分线	59
13.2 用 <code>\tkzLabelLine</code> 命令为直线添加标注	59
13.2.1 <code>\tkzLabelLine</code> 示例代码	60
14 绘制和标记线段	60
14.1 用 <code>\tkzDrawSegment</code> 命令绘制线段	60
14.1.1 简单示例	60
14.1.2 使用 <code>add</code> 选项扩展线段两端	61
14.1.3 使用 <code>dim</code> 选项标注尺寸	61
14.2 用 <code>\tkzDrawSegments</code> 命令绘制多条线段	62
14.2.1 为线段添加箭头	62
14.3 用 <code>\tkzMarkSegment</code> 命令标记线段	62
14.3.1 几种标记	63
14.3.2 <code>mark</code> 选项	63
14.4 用 <code>\tkzMarkSegments</code> 标记多条线段	63
14.4.1 标记等腰三角形	63
14.4.2 其它标记	64
14.5 线段标注命令 <code>\tkzLabelSegment</code>	64
14.5.1 多个标注	64
14.5.2 直角三角形的标注和标记	65
14.6 多条线段标注命令 <code>\tkzLabelSegments</code>	65
14.6.1 等腰三角形的标注	65
15 三角形	66
15.1 定义三角形命令 <code>\tkzDefTriangle</code>	66
15.1.1 <code>golden</code> 选项	66
15.1.2 <code>equilateral</code> 选项	67
15.1.3 <code>gold</code> 或 <code>euclide</code> 选项	67
15.2 绘制三角形	67
15.2.1 <code>pythagore</code> 选项	68
15.2.2 <code>school</code> 选项	68
15.2.3 <code>golden</code> 选项	68
15.2.4 <code>gold</code> 选项	68
15.2.5 <code>euclide</code> 选项	68
16 使用 <code>\tkzDefSpcTriangle</code> 命令定义特殊三角形	69
16.0.1 <code>medial</code> 或 <code>centroid</code> 选项	69
16.0.2 <code>in</code> 或 <code>incentral</code> 选项	70
16.0.3 <code>ex</code> 或 <code>excentral</code> 选项	70
16.0.4 <code>intouch</code> 选项	71
16.0.5 <code>extouch</code> 选项	71
16.0.6 <code>feuerbach</code> 选项	72
16.0.7 <code>tangential</code> 选项	72
16.0.8 <code>euler</code> 选项	73
17 定义多边形	74
17.1 定义正方形	74
17.1.1 用 <code>\tkzDefSquare</code> 命令通过两个点定义正方形	74
17.1.2 用 <code>\tkzDefSquare</code> 命令绘制等腰直角三角形	74
17.1.3 利用 <code>\tkzDefSquare</code> 绘制 Pythagorean 定理示意图	75
17.2 定义平行四边形	75

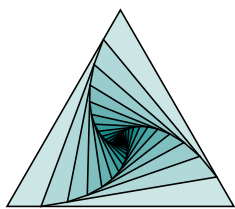
17.3	定义平行四边形的顶点	75
17.3.1	平行四边形定义	75
17.3.2	简单示例	76
17.3.3	黄金比例矩形	76
17.4	绘制正方形	76
17.4.1	在半圆内绘制两个正方形	77
17.5	定义黄金矩形	77
17.6	绘制黄金矩形	77
17.6.1	黄金矩形示例	77
17.7	绘制多边形命令	78
17.7.1	<code>\tkzDrawPolygon</code> 命令	78
17.8	绘制多边形顶点链条	78
17.8.1	多边形顶点链条	79
17.8.2	多边形顶点链条：循环实现	79
17.9	多边形裁剪	79
17.9.1	<code>\tkzClipPolygon</code> 命令	79
17.9.2	使用“裁剪”将 Sangaku 限制在正方形内	80
17.10	多边形着色	80
17.10.1	<code>\tkzFillPolygon</code> 命令	80
17.11	正多边形	81
17.11.1	<code>center</code> 选项	81
17.11.2	<code>side</code> 选项	81
18	圆	82
18.1	定义圆的命令: <code>\tkzDefCircle</code>	82
18.1.1	使用随机点与 <code>through</code> 选项	83
18.1.2	<code>diameter</code> 选项	83
18.1.3	三角形的内切圆和外接圆	83
18.1.4	<code>ex</code> 选项	84
18.1.5	<code>euler</code> 选项	84
18.1.6	<code>apollonius</code> 选项	85
18.1.7	<code>ex</code> 选项	85
18.1.8	<code>spieker</code> 选项	86
18.1.9	<code>orthogonal through</code> 选项	86
18.1.10	指定圆心的另一个圆的正交圆	86
19	圆的绘制和标注	87
19.1	绘制圆	87
19.1.1	绘制一个圆并对其进行着色	87
19.2	绘制多个圆	88
19.2.1	通过三角形定义圆	88
19.2.2	同心圆	89
19.2.3	旁切圆	89
19.2.4	心形线	89
19.3	绘制半圆	90
19.3.1	<code>\tkzDrawSemiCircle</code> 命令	90
19.4	给圆着色	90
19.4.1	sangaku 圆	91
19.5	圆形裁剪	91
19.5.1	样例	91
19.6	圆的标注	92
19.6.1	标注样例	92
20	交点	93
20.1	两条直线的交点	93
20.1.1	两个直线交点示例	93
20.2	一条直线和一个圆的交点	93
20.2.1	直线与圆的交点示例	94
20.2.2	直线与圆的交点复杂示例	94

20.2.3	由圆心和半径定义圆	95
20.2.4	更为复杂的例子	95
20.2.5	半径计算例 1	95
20.2.6	半径计算例 2	95
20.2.7	半径计算例 3	96
20.2.8	半圆中的矩形	96
20.2.9	“with nodes”选项	97
20.3	两个圆的交点	97
20.3.1	构造等边三角形	97
20.3.2	求中点	98
20.3.3	求等腰三角形	98
20.3.4	三等分线段	99
20.3.5	with nodes 选项	99
21	角度	100
21.1	颜色与角度: 填充	100
21.1.1	size 选项	100
21.1.2	改变点的顺序	100
21.1.3	多个角度	101
21.2	角度标记	101
21.2.1	mark = x 选项	103
21.2.2	mark = 选项	104
21.3	为角度添加标注	104
21.3.1	pos 选项	104
21.4	直角标记	105
21.4.1	直角标记示例	105
21.4.2	使用 german 样式添加直角标记	106
21.4.3	混合样式	106
21.4.4	完整示例	106
21.5	\tkzMarkRightAngles 命令	107
22	角度工具	107
22.1	\tkzGetAngle 命令	107
22.2	\tkzGetAngle 命令示例	107
22.3	三个点定义的角度	107
22.3.1	角度测量	108
22.4	\tkzFindAngle 命令	108
22.4.1	确定三角形的三个角	109
22.5	确定斜率	109
22.6	直线与横轴夹角计算命令: \tkzFindSlopeAngle	110
22.6.1	折叠	110
22.6.2	\tkzFindSlopeAngle 示例	111
23	扇形	112
23.1	\tkzDrawSector 命令	112
23.1.1	\tkzDrawSector 命令和 towards 选项	112
23.1.2	\tkzDrawSector 命令和 rotate 选项	113
23.1.3	\tkzDrawSector 命令和 R 选项	113
23.1.4	\tkzDrawSector 命令和 R 选项	113
23.1.5	\tkzDrawSector 命令和 R with nodes 选项	114
23.2	\tkzFillSector 命令	114
23.2.1	\tkzFillSector 命令和 towards 选项	115
23.2.2	\tkzFillSector 命令和 rotate 选项	115
23.3	\tkzClipSector 命令	115
23.3.1	\tkzClipSector 命令	116
24	圆弧	117
24.1	towards 选项	117
24.2	towards 选项	118

24.3	<code>rotate</code> 选项	118
24.4	<code>R</code> 选项	118
24.5	<code>R with nodes</code> 选项	119
24.6	<code>delta</code> 选项	119
24.7	<code>angles</code> 选项: 示例 1	120
24.8	<code>angles</code> 选项: 示例 2	120
25	杂项命令	121
25.1	线段复制	121
25.1.1	用 <code>\tkzDuplicateSegment</code> 命令得到黄金分割	122
25.2	计算线段长度命令 <code>\tkzCalcLength</code>	122
25.2.1	构建罗盘矩形	123
25.3	将 pt 转换为 cm	123
25.4	将 cm 转换为 pt	123
25.4.1	示例代码	124
25.5	获取点的坐标分量	124
25.5.1	<code>\tkzGetPointCoord</code> 命令示例代码	124
25.5.2	使用 <code>\tkzGetPointCoord</code> 命令求向量和	124
26	使用罗盘	125
26.1	主命令 <code>\tkzCompass</code>	125
26.1.1	<code>length</code> 选项	125
26.1.2	<code>delta</code> 选项	125
26.2	多重命令 <code>\tkzCompass</code>	125
26.3	设置命令 <code>\tkzSetUpCompass</code>	126
26.3.1	<code>\tkzSetUpCompass</code> 示例代码	126
27	显示尺规标记	127
27.1	显示直线尺规作图标记命令 <code>\tkzShowLine</code>	127
27.1.1	<code>\tkzShowLine</code> 命令和 <code>parallel</code> 选项	127
27.1.2	<code>\tkzShowLine</code> 命令和 <code>perpendicular</code> 选项	127
27.1.3	<code>\tkzShowLine</code> 命令和 <code>bisector</code> 选项	128
27.1.4	<code>\tkzShowLine</code> 命令和 <code>mediator</code> 选项	128
27.2	显示部分变换过程的尺规作图标记命令 <code>\tkzShowTransformation</code>	128
27.2.1	<code>\tkzShowTransformation</code> 示例	129
27.2.2	<code>\tkzShowTransformation</code> 的另一个示例	129
28	差分点	130
28.1	<code>\tkzDefEquiPoints</code> 命令	130
28.1.1	使用带选项的 <code>\tkzDefEquiPoints</code> 命令	130
29	量角器	131
29.1	正向圆量角器	131
29.2	反向圆量角器	131
30	实例分析	132
30.1	一些有趣的例子	132
30.1.1	相似等腰三角形	132
30.1.2	“Tangente”方法的修订版本	133
30.1.3	“Le Monde”版本	134
30.1.4	三角形的高	135
30.1.5	三角形的高 - 另一种构造方式	135
30.2	其他作者提供的实例	136
30.2.1	整数的算术平方根	136
30.2.2	直角三角	136
30.2.3	阿基米德	136
30.2.4	示例: Dimitris Kapeta	137
30.2.5	示例 1: John Kitzmiller	138
30.2.6	示例 2: John Kitzmiller	139
30.2.7	示例 3: John Kitzmiller	140

30.2.8 示例 4: John Kitzmiller	141
30.2.9 示例 1: from Indonesia	142
30.2.10 示例 2: from Indonesia	143
30.2.11 三个圆	145
30.2.12 APOLLONIUS 圆	146
31 个性化设置	148
31.1 使用 <code>\tkzSetUpLine</code> 命令	148
31.1.1 示例 1: 改变线宽	148
31.1.2 示例 2: 改变线型	149
31.1.3 示例 3: 延伸	149
31.2 点的样式	149
31.2.1 <code>\tkzSetUpPoint</code> 命令	150
31.2.2 在组内使用 <code>\tkzSetUpPoint</code> 命令	150
31.3 <code>\tkzSetUpCompass</code> 命令	150
31.3.1 角平分线的 <code>\tkzSetUpCompass</code> 命令	150
31.3.2 使用 <code>\tkzSetUpCompass</code> 的另一个示例	151
31.4 Own style	151
32 tkz-base 小结	152
32.1 <code>tkz-base</code> 宏包的工具	152
32.2 <code>\tkzInit</code> 命令和 <code>\tkzShowBB</code> 命令	152
32.3 <code>\tkzClip</code> 命令	152
32.4 <code>\tkzClip</code> 命令和 <code>space</code> 选项	152
33 FAQ	154
33.1 常见错误	154
Index	155

1 概述



```
\begin{tikzpicture}[scale=.25]
  \tkzDefPoints{00/0/A,12/0/B,6/12*sind(60)/C}
  \foreach \density in {20,30,...,240}{%
    \tkzDrawPolygon[fill=teal!\density](A,B,C)
    \pgfnodealias{X}{A}
    \tkzDefPointWith[linear,K=.15](A,B) \tkzGetPoint{A}
    \tkzDefPointWith[linear,K=.15](B,C) \tkzGetPoint{B}
    \tkzDefPointWith[linear,K=.15](C,X) \tkzGetPoint{C}
  }
\end{tikzpicture}
```

1.1 为什么要开发tkz-euclide?

开发该宏包的最初想法是为自己和其它数学老师开发一个 \LaTeX 绘图工具，以实现欧氏几何图形的快速绘制，而不需要费力和掌握一门新的绘图语言。当然，**tkz-euclide** 适用于所有使用 \LaTeX 老师，以实现在 \LaTeX 中轻松、正确地绘图。

显然，最简单的绘图方法是按手工绘图的方式和思维进行绘图。为了描述一个几何图形结构，必须定义图形对象以及对对象所执行的操作。因此，如果与数学家或学习数学的学生使用的数学语言语法相近，这些语法则更容易被他们理解和掌握。当然，宏包的语法也必须符合 \LaTeX 用户的使用习惯。

基于此，该宏包定义了点、线段、直线、三角形、多边形和圆六种图形对象。并且设计了定义、创建、绘制、标记和标注五个对图形对象的基本操作。

虽然这样定义的语法会比较冗长，但它很容易理解和使用。因此，一般学生也可以像熟练使用该宏包的老师一样，能够轻松访问此宏包提供的命令进行绘图。

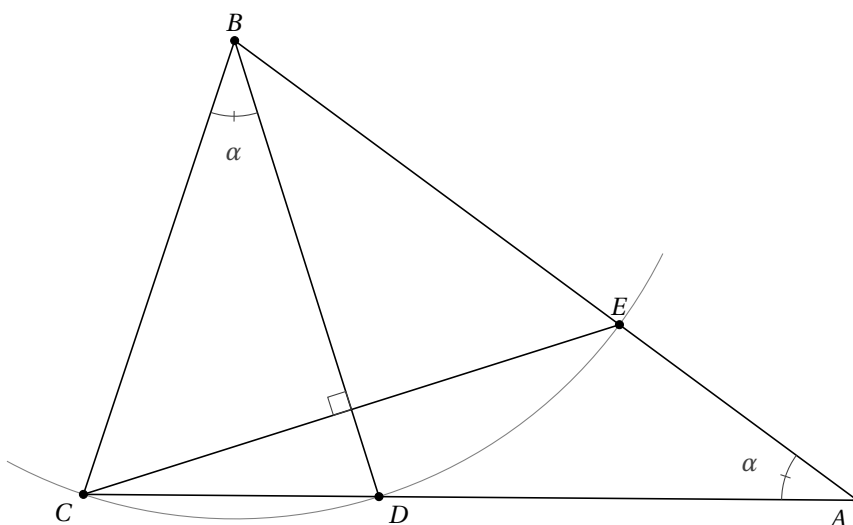
1.2 tkz-euclide与TikZ

当然，TikZ 本身的绘图功能也极其强大，如果没有 TikZ 的话，则永远不会有 **tkz-euclide** 宏包。**tkz-euclide** 是基于 TikZ 开发的，并且可以在 **tkz-euclide** 绘图代码中随时使用 TikZ 代码，**tkz-euclide** 并不限制对 TikZ 的直接调用。不过，并不建议采用混合语法绘图，这样会降低代码的可读性。

其实，无需比较 TikZ 和 **tkz-euclide**，两者的用户不完全相同。前者可以实现更为复杂的图形绘制，后者则擅长绘制欧氏几何图形。当然，前者完全能够实现后者所有的功能。

1.3 工作模式

1.3.1 示例 I: 黄金三角形



其绘制过程如下：

1. CBD 和 DBE 都是等腰三角形；
2. $BC = BE$, BD 是角 CBE 的角平分线。
3. 由此推出角 CBD 和角 DBE 相等，在此，记为 α 。

$$\widehat{BAC} + \widehat{ABC} + \widehat{BCA} = 180^\circ \text{ (在三角形 } BAC \text{ 中)}$$

$$3\alpha + \widehat{BCA} = 180^\circ \text{ (在三角形 } CBD \text{ 中)}$$

因此

$$\alpha + 2\widehat{BCA} = 180^\circ$$

或

$$\widehat{BCA} = 90^\circ - \alpha/2$$

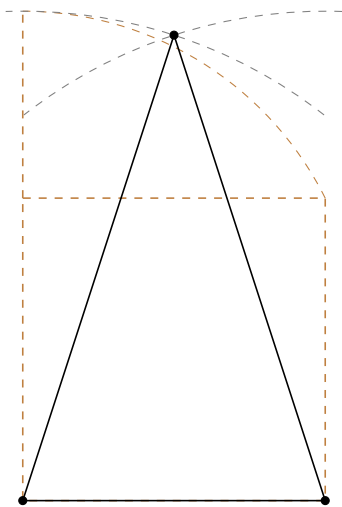
4. 从而可得

$$\widehat{CBD} = \alpha = 36^\circ$$

所以， CBD 是一个“黄金”三角形。

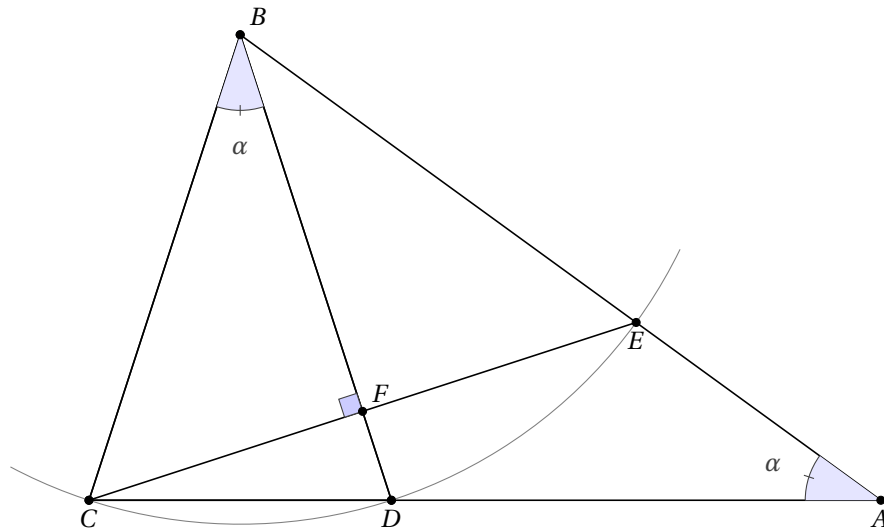
如何构造一个黄金三角形或者顶角为 36° 的等腰三角形呢？

1. 先使用 `\tkzDefPoint(0,0){C}` 和 `\tkzDefPoint(4,0){D}` 定义两个已知点 C 和 D 。
2. 构造一个正方形 $CDef$ ，并且取 Cf 的中点 m 。当然，可以使用圆规和直尺来完成所有这些工作。
3. 然后，以 m 为圆心作过 e 的圆弧，该圆弧与直线 Cf 相交于 n 。
4. 至此，分别以 C 和 D 为圆心，以 Cn 为半径的两个圆弧可定义点 B 。



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(4,0){D}
  \tkzDefSquare(C,D)
  \tkzGetPoints{e}{f}
  \tkzDefMidPoint(C,f)
  \tkzGetPoint{m}
  \tkzInterLC(C,f)(m,e)
  \tkzGetSecondPoint{n}
  \tkzInterCC[with nodes](C,C,n)(D,C,n)
  \tkzGetFirstPoint{B}
  \tkzDrawSegment[brown,dashed](f,n)
  \pgfinterruptboundingbox
  \tkzDrawPolygon[brown,dashed](C,D,e,f)
  \tkzDrawArc[brown,dashed](m,e)(n)
  \tkzCompass[brown,dashed,delta=20](C,B)
  \tkzCompass[brown,dashed,delta=20](D,B)
  \endpgfinterruptboundingbox
  \tkzDrawPoints(C,D,B)
  \tkzDrawPolygon(B,...,D)
\end{tikzpicture}
```

构造了黄金三角形 BCD 后，可以通过以 D 为圆心， BD 为半径的圆与直线 CD 的交点来定义 $BD = DA$ 的点 A 。再通过以 B 为圆心， BC 为半径的圆与直线 BA 的交点来定义的然后再定义点 E 。最后通过直线 BA 与直线 BD 的交点定义点 F 。



```

\begin{tikzpicture}
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(4,0){D}
  \tkzDefSquare(C,D)
  \tkzGetPoints{e}{f}
  \tkzDefMidPoint(C,f)
  \tkzGetPoint{m}
  \tkzInterLC(C,f)(m,e)
  \tkzGetSecondPoint{n}
  \tkzInterCC[with nodes](C,C,n)(D,C,n)
  \tkzGetFirstPoint{B}
  \tkzInterLC(C,D)(D,B) \tkzGetSecondPoint{A}
  \tkzInterLC(B,A)(B,D) \tkzGetSecondPoint{E}
  \tkzInterLL(B,D)(C,E) \tkzGetPoint{F}
  \tkzDrawPoints(C,D,B)
  \tkzDrawPolygon(B,...,D)
  \tkzDrawPolygon(B,C,D)
  \tkzDrawSegments(D,A A,B C,E)
  \tkzDrawArc[delta=10](B,C)(E)
  \tkzDrawPoints(A,...,F)
  \tkzMarkRightAngle[fill=blue!20](B,F,C)
  \tkzFillAngles[fill=blue!10](C,B,D E,A,D)
  \tkzMarkAngles(C,B,D E,A,D)
  \tkzLabelAngles[pos=1.5](C,B,D E,A,D){$\alpha$}
  \tkzLabelPoints[below](A,C,D,E)
  \tkzLabelPoints[above right](B,F)
\end{tikzpicture}

```

1.3.2 示例 II: 另外两种方式

tkz-euclide宏包可以通过“gold”或“euclide”参数定义三角形，因此可按如下方式定义 BCD 和 BCA 黄金三角形：

建图

```

\begin{tikzpicture}
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(4,0){D}
  \tkzDefTriangle[euclide](C,D)
  \tkzGetPoint{B}
  \tkzDefTriangle[euclide](B,C)
  \tkzGetPoint{A}
  \tkzInterLC(B,A)(B,D) \tkzGetSecondPoint{E}

```

```

\tkzInterLL(B,D)(C,E) \tkzGetPoint{F}
\tkzDrawPoints(C,D,B)
\tkzDrawPolygon(B,...,D)
\tkzDrawPolygon(B,C,D)
\tkzDrawSegments(D,A A,B C,E)
\tkzDrawArc[delta=10](B,C)(E)
\tkzDrawPoints(A,...,F)
\tkzMarkRightAngle[fill=blue!20](B,F,C)
\tkzFillAngles[fill=blue!10](C,B,D E,A,D)
\tkzMarkAngles(C,B,D E,A,D)
\tkzLabelAngles[pos=1.5](C,B,D E,A,D){$\alpha$}
\tkzLabelPoints[below](A,C,D,E)
\tkzLabelPoints[above right](B,F)
\end{tikzpicture}

```

也可以使用坐标旋转的方法定义三角形:

```

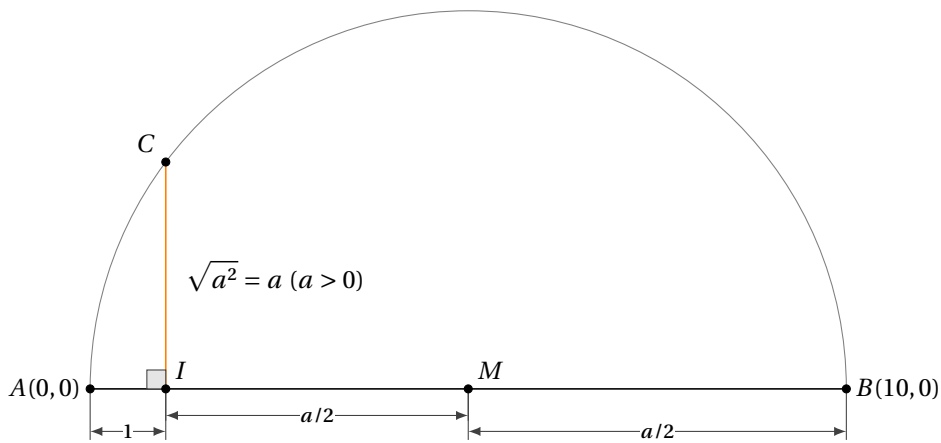
\begin{tikzpicture}
\tkzDefPoint(0,0){C} % 也可以在定义点的同时用
% 类似\tkzDefPoint[label=below:$C$](0,0){C} 的方式进行标注
% 但不建议这样做
\tkzDefPoint(2,6){B}
% 通过旋转定义点 D 和点 E
\tkzDefPointBy[rotation= center B angle 36](C) \tkzGetPoint{D}
\tkzDefPointBy[rotation= center B angle 72](C) \tkzGetPoint{E}
% 通过直线交点定义点 A 和点 H
\tkzInterLL(B,E)(C,D) \tkzGetPoint{A}
\tkzInterLL(C,E)(B,D) \tkzGetPoint{H}
% 绘制
\tkzDrawArc[delta=10](B,C)(E)
\tkzDrawPolygon(C,B,D)
\tkzDrawSegments(D,A B,A C,E)
% 角度标记
\tkzMarkAngles(C,B,D E,A,D) % 绘制圆弧
\tkzLabelAngles[pos=1.5](C,B,D E,A,D){$\alpha$}
\tkzMarkRightAngle(B,H,C)
\tkzDrawPoints(A,...,E)
% 仅实现标注
\tkzLabelPoints[below left](C,A)
\tkzLabelPoints[below right](D)
\tkzLabelPoints[above](B,E)
\end{tikzpicture}

```

1.3.3 最小工作示例

该示例说明了如何使用尺规绘图的方式从长度为 a 的线段得到长度为 \sqrt{a} 的线段

$IB = a, AI = 1$



建议

- 导言区

如果需要使用`xcolor`宏包，则必须在`tkz-euclide`之前载入，也就是说，在使用`TikZ`宏包前载入。否则`TikZ`宏包可能会造成冲突，但`TikZ`宏包提供的`babel`库能解决这些冲突。

```
\documentclass{standalone} % 或其它文档类
% \usepackage{xcolor} % 需要在 tikz 或 tkz-euclide 宏包之前载入
\usepackage{tkz-euclide} % 不再需要显式载入 TikZ 宏包
% \usetikzobj{all} 不再需要该命令
% \usetikzlibrary{babel} 如果有冲突，则可以使用该库进行解决
```

该代码可以分解为以下几个部分：

- 定义已知点：第1部分是定义用于构成图形的已知点：多数情况下，不需要使用`\tkzInit`和`\tkzClip`命令。

```
\tkzDefPoint(0,0){A}
\tkzDefPoint(1,0){I}
```

- 第2部分是利用已知点通过计算得到其它需要的点： B 点位于从 A 开始的10 cm处。 M 是 $[AB]$ 的中点，然后得到通过 I 点的 (AB) 的正交线。然后得到该正交线与以 M 为圆心，过 A 的半圆的交点。

```
\tkzDefPointBy[homothety=center A ratio 10](I)
\tkzGetPoint{B}
\tkzDefMidPoint(A,B)
\tkzGetPoint{M}
\tkzDefPointWith[orthogonal](I,M)
\tkzGetPoint{H}
\tkzInterLC(I,H)(M,A)
\tkzGetSecondPoint{C}
```

- 第3部分是图形绘制：

```
\tkzDrawSegment[style=orange](I,H)
\cancel{\tkzDrawPoints(0,1,A,B,M)}
\tkzDrawArc(M,0)(I)
\tkzDrawSegment[dim={1$, -16pt,}] (0,I)
\tkzDrawSegment[dim={a/2$, -10pt,}] (A,M)
\tkzDrawSegment[dim={a/2$, -16pt,}] (M,B)
```

- 第4部分是绘制标记：

`\tkzMarkRightAngle(A,I,B)`

– 最后是布置标注:

`\tkzLabelPoint[left](A){$A(0,0)$}`
~~`\tkzLabelPoint[right](A){$B(10,0)$}`~~
~~`\tkzLabelSegment[right=4pt](I,B){$\sqrt{a^2}=a \ (a>0)$}`~~

复制代码至此!

– 完整的代码:

```
\begin{tikzpicture}[scale=1,ra/.style={fill=gray!20}]
% 已知点
\tkzDefPoint(0,0){A}
\tkzDefPoint(1,0){I}
% 求解点
\tkzDefPointBy[homothety=center A ratio 10](I) \tkzGetPoint{B}
\tkzDefMidPoint(A,B) \tkzGetPoint{M}
\tkzDefPointWith[orthogonal](I,M) \tkzGetPoint{H}
\tkzInterLC(I,H)(M,B) \tkzGetSecondPoint{C}
% 绘图
\tkzDrawSegment[style=orange](I,C)
\tkzDrawArc(M,B)(A)
\tkzDrawSegment[dim={1$, -16pt,}] (A,I)
\tkzDrawSegment[dim={a/2$, -10pt,}] (I,M)
\tkzDrawSegment[dim={a/2$, -16pt,}] (M,B)
% 标记
\tkzMarkRightAngle[ra](A,I,C)
% 标注
\tkzDrawPoints(I,A,B,C,M)
\tkzLabelPoint[left](A){$A(0,0)$}
\tkzLabelPoints[above right](I,M)
\tkzLabelPoints[above left](C)
\tkzLabelPoint[right](B){$B(10,0)$}
\tkzLabelSegment[right=4pt](I,C){$\sqrt{a^2}=a \ (a>0)$}
\end{tikzpicture}
```

1.4 tkz-euclide 代码的基本要素

接下来，讨论使用 **tkz-euclide** 绘制图形时的“规则”和“符号系统”

绘图的基本要素是点，可以在任何时候通过定义一个点时，点的命名来引用一个点（也可以在后续操作中为点赋予不同的名称）。

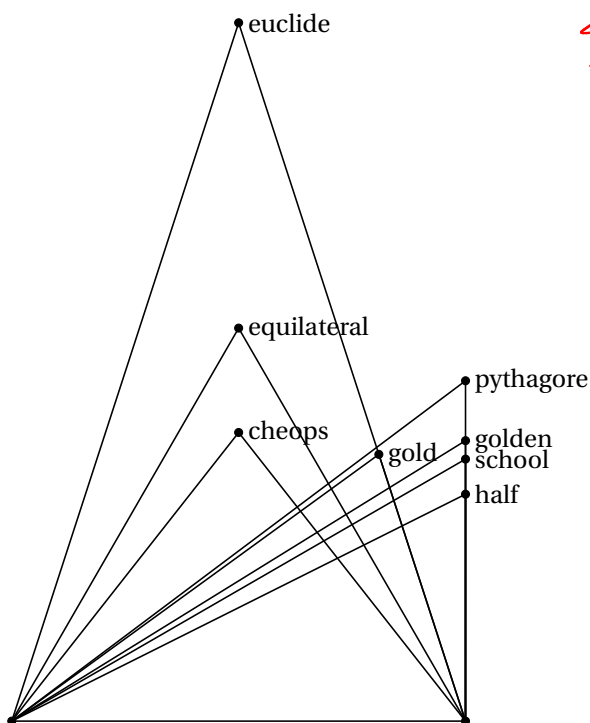
通常，**tkz-euclide** 的命令都是以 **tkz** 作为前缀，主要的四类命令的前缀是：**\tkzDef...**、**\tkzDraw...**、**\tkzMark...** 和 **\tkzLabel...**。

在第 1 类命令中，**\tkzDefPoint** 命令用于通过坐标值定义一个点，关于该命令的细节过后会进行讨论。

在此，首先详细说明一下 **\tkzDefTriangle** 命令。

该命令能够通过将一个点与一对点相关联，从而通过已知三角形来定义一个点。可以通过 **\tkzPointResult** 使用该点，当然，也可以通过 **\tkzGetPoint{C}** 引用该点。圆括号用于传递参数，在 (A,B) 中的 A 和 B 是用于定义第 3 个点。

然而，在 {C} 中，使用大括号得到新的点，三角形可以是：equilateral, half, pythagoras, school, golden 或 sublime, euclide, gold, cheops... 和 two angles，可以在其中进行选择，例如：**\tkzDefTriangle[euclide] (A,B)** **\tkzGetPoint{C}**



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoints{0/0/A,8/0/B}
  \foreach \tr in {equilateral,half,pythagore,%
    school,golden,euclide, gold,cheops}
  {\tkzDefTriangle[\tr](A,B) \tkzGetPoint{C}}
  \tkzDrawPoint(C)
  \tkzLabelPoint[right](C){\tr}
  \tkzDrawSegments(A,C C,B)}
  \tkzDrawPoints(A,B)
  \tkzDrawSegments(A,B)
\end{tikzpicture}
```

1.5 符号和约定

该宏包选择法国几何符号和个人约定对几何图形进行描述，**tkz-euclide** 宏包定义和表示的对象有平面中的点、线和圆，它们是欧氏几何的主要元素，可以根据这些基本元素构成几何图形。

根据欧几里得的说法，这些仅使用这些图形就可以表示各种几何想法。因此，一个点并没有具体尺寸，而在现实中是不存在的。同样，一条线没有宽度，而在现实中也是不存在的。需要考虑的是代表理想数学对象的对象，**tkz-euclide** 将遵循古希腊人的步骤，使用尺规进行绘图。

以下是要使用的符号：

- 点可以用圆盘或十字线（两条直线、一条直线和圆或两个圆的交点）表示。

• B

```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B}
\tkzDrawPoints(A,B)
\tkzLabelPoints(A,B)
\end{tikzpicture}
```

• A

或

+ B

```
\begin{tikzpicture}
\tkzSetUpPoint[shape=cross, color=red]
\tkzDefPoints{0/0/A,4/2/B}
\tkzDrawPoints(A,B)
\tkzLabelPoints(A,B)
\end{tikzpicture}
```

+ A

对于存在的点，可能使用类似 A 、 B 或 C 这样的大写字母(当然，也会有例外)进行标注，例如：

- O 表示圆心、旋转中心等；
- M 表示中点；
- H 表示高；
- P' 表示变换后，点 P 的镜像；

需要注意的是：一个点在代码中的引用名称和文本中的标注名称可能是不一样的，所以可以定义一个点 A ，但是将其标注为 P 。

• P

```
\begin{tikzpicture}
\tkzDefPoints{0/0/A}
\tkzDrawPoints(A)
\tkzLabelPoint(A){P}
\end{tikzpicture}
```

注意也有例外情况：一些如三角形各边的中点，是只有边的特征的，因此，常常用带有边特征的标注，如： M_a 、 M_b 和 M_c 或 M_A 、 M_B 和 M_C 。

在代码中，可以使用诸如 M_A 、 M_B 和 M_C 的形式引用这些点。

另外一种例外情况是无需标注的内部点，这些点在代码中常常用小写字母表示。

- 线段使用方括号中的两个点表示，如： $[AB]$ 。
- 在欧氏几何中，直线用两个点表示，因此， A 和 B 定义的直线表示为 (AB) 。也可以使用希腊字母表示直线，并将其命名为 (δ) 或 (Δ) 。也可以使用小写字母表示直线，如 d 和 d' 。
- 射线可表示为 $[AB)$ 。
- 对于直线间的关系，例如 (AB) 和 (CD) 两条直线，垂直表示为 $(AB) \perp (CD)$ ，平行表示为 $(AB) \parallel (CD)$ 。
- 三角形 ABC 的边长表示为 AB 、 AC 和 BC 。长度值一般用小写字母表示如： $AB = c$ 、 $AC = b$ 和 $BC = a$ 。字母 a 也常常用于表示一个角度， r 常常用于表示半径， d 表示直径， l 表示长度， d 也可以表示距离。
- 多边形用其顶点表示，如三角形表示为 ABC ，四边形表示为 $EFGH$ 。
- 角度用度为单位(例如： 60°)，对于等边三角形 ABC ，可以表示为 $\widehat{ABC} = \widehat{B} = 60^\circ$ 。
- 圆弧使用起止点表示，如，若 A 和 B 是同一个圆上的两个点，则可以用 \widehat{AB} 表示圆弧。
- 如果没有歧义，则一个圆可以表示为 \mathcal{C} ，或用 $\mathcal{C}(O; A)$ 表示圆心在 O 点并通过点 A 的圆或用 $\mathcal{C}(O; 1)$ 表示圆心在点 O 半径为 1 cm 的圆。
- 三角形中的特殊线可表示有内角平分线、角外平分线等。
- (x_1, y_1) 表示点 A_1 的坐标分量， (x_A, y_A) 表示点 A 的坐标分量。

1.6 使用tkz-euclide宏包

1.6.1 一个经典示例

在此，通过绘制一个等边三角形为例，展示正确使用方式。当然，绘制该图形可以有多种方式，本例中将遵循欧几里得步骤。

- 首先需要引入文档类，对于单个图形而言，比较方便的一种方式是使用standalone文档类。

```
\documentclass{standalone}
```

- 然后使用tkz-euclide载入tkz-euclide宏包：

```
\usepackage{tkz-euclide}
```

注意，由于tkz-euclide宏包是基于TikZ宏包实现的，并载入该宏包，因此，无需在代码中载入TikZ宏包

- ~~\usetkzobj{all}~~ 在3.03版以后，无需再使用该命令载入需要的对象，默认情况下，载入所有对象。
- 开始编写文章，并使用tikzpicture环境进行欧氏几何图形绘制：

```
\begin{document}
\begin{tikzpicture}
```

- 定义两个已知点：

```
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,2){B}
```

- 使用这两个点定义两个圆，并使用这两个圆定义交点：

~~(A,B)为圆以A点为圆心通过B点，另一个圆以B点为圆心通过A点，这两个圆共用这两个点。~~

```
\tkzInterCC(A,B)(B,A)
```

```
\tkzGetPoints{C}{D}
```

- 至此，便完成了所有点的定义，接下来，就可以进行绘图了。

```
\tkzDrawCircles[gray,dashed](A,B B,A)
\tkzDrawPolygon(A,B,C)% 三角形
```

- 绘制所有点 A、B、C 和 D：

```
\tkzDrawPoints(A,...,D)
```

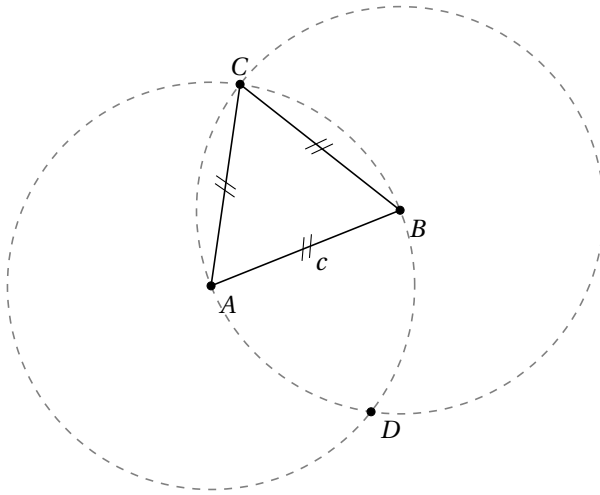
- 最后一步是绘制标注，在绘制标注时，可以为其指定位置参数。

```
\tkzLabelSegments[swap](A,B){$c$}
\tkzLabelPoints(A,B,D)
\tkzLabelPoints[above](C)
```

- 最后，结束各个环境

```
\end{tikzpicture}
\end{document}
```

- 完整的代码



```
\begin{tikzpicture}[scale=0.5]
% 已知点
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,2){B}
% 计算得到的点
\tkzInterCC(A,B)(B,A)
\tkzGetPoints{C}{D}
% 绘图
\tkzDrawCircles[gray,dashed](A,B B,A)
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,...,D)
% 标记
\tkzMarkSegments[mark=s||](A,B B,C C,A)
% 标注
\tkzLabelSegments[swap](A,B){$c$}
\tkzLabelPoints(A,B,D)
\tkzLabelPoints[above](C)
\end{tikzpicture}
```

1.6.2 Set, Calculate, Draw, Mark, Label

这个标题的含义应该是：计算与绘制的分离

在使用 \LaTeX 排版时，源代码分为正文和导言两大部分。通过这种方式，可以将出版物进行结构化设计，并通过样式和排版命令集简化用户的排版过程。 tkz-euclide 是基于这种内容与格式分离的思想进行设计的，以简化用户的绘图过程。

首先是定义已知点，然后计算其他需要的点，这是绘图的两个主要内容。接下来是绘制、标记和标注。

2 安装

`tkz-euclide`和`tkz-base`现在已被收录于 CTAN¹。如果需要测试开发版，则需要按 `texmf` 的目录结构将必要的文件得到到指定的目录。并且要注意以下几点：

- `latex`必须能够识别`tkz-base`和`tkz-euclide`目录。
- 必须正确安装`xfp`²、`numprint`和`tikz 3.00`及以上版本才能够正常使用 `tkz-euclide`。
- 该说明文档和所有的示例都是使用`lualatex-dev`编译的，但使用`pdflatex`和`xelatex`也能够正常编译。

2.1 `tkzbase`和`tkzeuclide`目录中的文件

在`base`目录中有如下文件：

- `tkz-base.cfg`
- `tkz-base.sty`
- `tkz-lib-marks.tex`
- `tkz-obj-axes.tex`
- `tkz-obj-grids.tex`
- `tkz-obj-marks.tex`
- `tkz-obj-points.tex`
- `tkz-obj-rep.tex`
- `tkz-tools-arith.tex`
- `tkz-tools-base.tex`
- `tkz-tools-BB.tex`
- `tkz-tools-misc.tex`
- `tkz-tools-modules.tex`
- `tkz-tools-print.tex`
- `tkz-tools-text.tex`
- `tkz-tools-utilities.tex`

在`euclide`目录中有如下文件：

- `tkz-euclide.sty`
- `tkz-obj-eu-angles.tex`
- `tkz-obj-eu-arcs.tex`
- `tkz-obj-eu-circles.tex`
- `tkz-obj-eu-compass.tex`
- `tkz-obj-eu-draw-circles.tex`
- `tkz-obj-eu-draw-lines.tex`
- `tkz-obj-eu-draw-polygons.tex`
- `tkz-obj-eu-draw-triangles.tex`
- `tkz-obj-eu-lines.tex`
- `tkz-obj-eu-points-by.tex`
- `tkz-obj-eu-points-rnd.tex`

¹ `tkz-base`和`tkz-euclide` 已被`TeXLive`发行版收录，可以通过`tlmgr`进行安装。在 Windows平台，`MiKTeX`发行版也收录了这两个宏包。

² `xfp`替代了`fp`。

- tkz-obj-eu-points-with.tex
- tkz-obj-eu-points.tex
- tkz-obj-eu-polygons.tex
- tkz-obj-eu-protractor.tex
- tkz-obj-eu-sectors.tex
- tkz-obj-eu-show.tex
- tkz-obj-eu-triangles.tex
- tkz-tools-angles.tex
- tkz-tools-intersections.tex
- tkz-tools-math.tex



如果目录结构及文件正确，则`tkz-euclide`能够自动载入所有需要的文件。

3 新特性与兼容性

更新后，语法一致性得到增强。尤其是区分了点的定义及点的计算后，方便了后续的绘图、标记和标注操作。在未来，会更好地区分宏的定义，以使得引入 **Lua** 来计算坐标更加容易。

一个重要的新特性是使用 **xfp** 宏包替换了 **fp** 宏包，这进一步提升了计算性能，并使得用户使用更为便捷。

以下是主要的更新：

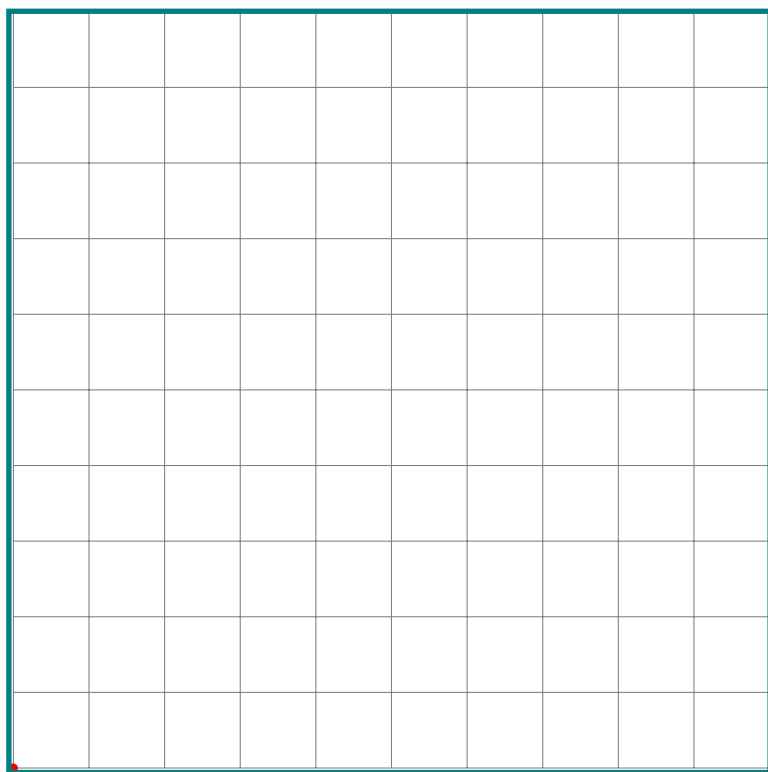
- 优化了代码并进行了 Bug 修复;
- 由于 **tkz-euclide** 默认载入了所有对象，因此无需再使用 `\usetkzobj{all}` 进行载入操作;
- 绘图区域移至各个命令单独实现，从而避免使用紧跟有 **tkzClip** 命令的 `\tkzInit` 命令
- 增加了 `\tkzSaveBB`、`\tkzClipBB` 等绘图区域管理命令;
- 逻辑上，由于所有命令都可以直接使用 **TikZ** 的选项，所以尽可能移除了“重复”选项，如“label options”选项;
- 使用 **tkz-euclide** 产生随机点，并且使用 `tkzDefRandPointOn` 命令代替了 `\tkzGetRandPointOn` 命令。同样的原因，必须使用 `\tkzGetPoint` 引用随机点;
- 新增了 `\tkzLabelLine` 命令，用于为直线添加标注。删除了原来的直线标注可选参数 `end` 和 `start`。
- 新增了用于为点添加标注的 `quotes` 和 `angles` 库。
- 删除了向量操作，为 `\tkzDrawSegment` 命令添加了“->”以绘制向量;
- 仍然存在但已过时并将要删除的宏命令有：
 - 用于跟踪和创建三角形各边中点的命令 `\tkzDrawMedians`，由于不符合计算和绘制分离的思想因此，建议首先使用 `\tkzSpcTriangle[median]` 命令创建中点，然后使用 `\tkzDrawSegments` 或 `\tkzDrawLines` 进行绘制。
 - 将 `\tkzDrawMedians(A,B)(C)` 命令改为了 `\tkzDrawMedians(A,C,B)`，该命令用于定义中点 *C*;
 - `\tkzDrawTriangle[equilateral]` 命令也将被删除，更好的方式是先用 `\tkzDefTriangle[equilateral]` 命令定义点，然后用 `\tkzDrawPolygon` 绘制多边形;
 - 用 `\tkzGetRandPointOn` 命令替换了 `\tkzDefRandPointOn` 命令;
 - 用 `\tkzDefTangent` 命令替换了 `\tkzTangent` 命令;
 - 可以使用 `global path name` 参数求交点，但像在 **TikZ** 中一样，它的计算非常慢。
- 引入了 `\usetkztool` 命令，以加载新的“tools”库。

4 点的定义

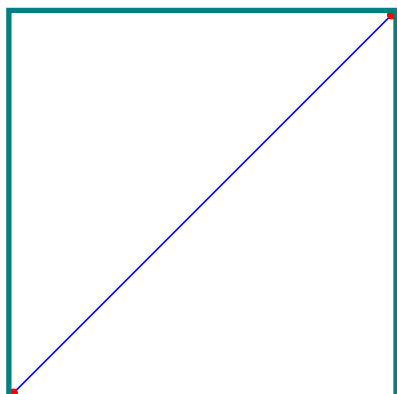
可以通过如下方式定义点:

- 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}
```

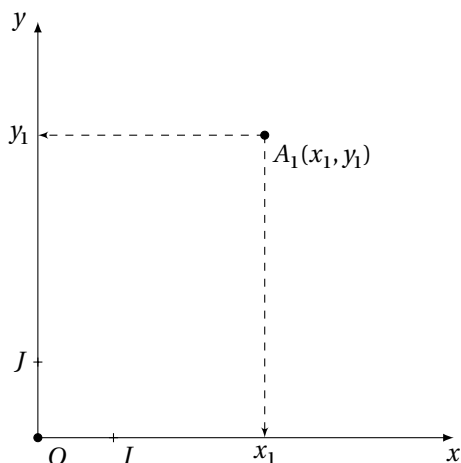


```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(5,5){A}
  \tkzDrawSegment[blue](O,A)
  \tkzDrawPoints[red](O,A)
  \tkzShowBB[line width=2pt,teal]
\end{tikzpicture}
```

笛卡尔坐标 (a,b) 表示该点的 x 坐标是 a cm, y 坐标是 b cm。

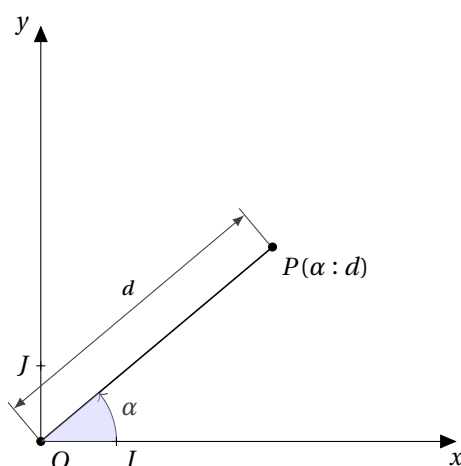
极坐标中表示一个点需要一个角度 α (度) 和一个从原点度量的距离 d (默认单位是 cm)。

笛卡尔坐标



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

极坐标



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

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。

4.1 \tkzDefPoint命令：定义命名点

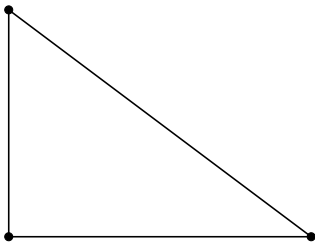
`\tkzDefPoint[< 命令选项>](<x,y>){< 名称>} or (<alpha:d>){< 名称>}`

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	α 是角度 (度)， d 是距离 (cm)
{名称}	无	点的名称，如： A , T_a , P_1 , ...

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。

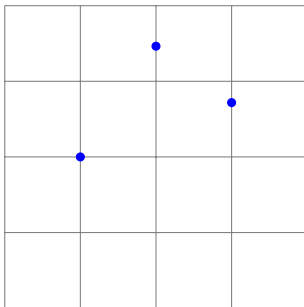
选项	默认值	含义
label	无	按预设的距离添加标注
shift	无	为 (x,y) 或 $(\alpha:d)$ 添加偏移

4.1.1 笛卡尔坐标



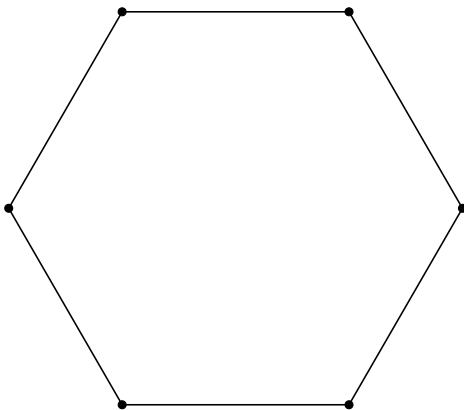
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(0,3){C}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

4.1.2 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}
```

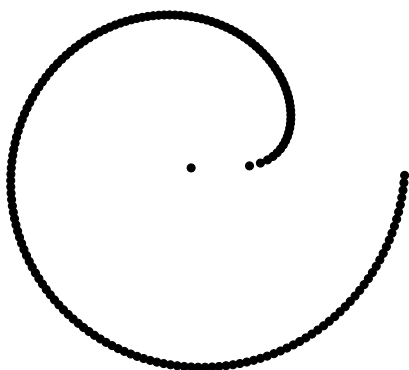
4.1.3 极坐标



```
\begin{tikzpicture}
  \foreach \an [count=\i] in {0,60,...,300}
    {\tkzDefPoint(\an:3){A_\i}}
  \tkzDrawPolygon(A_1,A_2,A_3,A_4,A_5,A_6)
  \tkzDrawPoints(A_1,A_2,A_3,A_4,A_5,A_6)
\end{tikzpicture}
```

4.1.4 计算和坐标

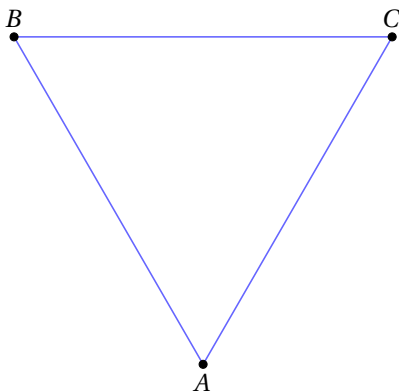
在计算时,需要遵循`xfp`宏包的语法。另外,也有可能会用至`pgfmath`宏包进行计算,此时,需要在`\tkzDefPoint`命令之前完成计算。



```
\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}
```

4.1.5 相对点

在使用相对点时,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}
```

4.2 `\tkzDefShiftPoint`命令: 通过平移定义点

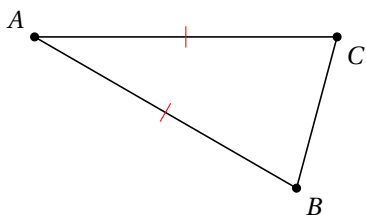
`\tkzDefShiftPoint[⟨ 参考点 ⟩](⟨ x,y ⟩){⟨ 名称 ⟩}` 或 `(⟨ $\alpha:d$ ⟩){⟨ 名称 ⟩}`

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>

4.2.1 使用\tkzDefShiftPoint命令定义等腰三角形

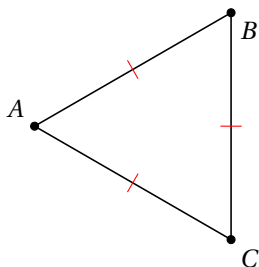
This macro allows you to place one point relative to another. This is equivalent 该命令允许相对于另一个点定义点，等价于对点进行平移。下面的代码给出了一种通过点 A 和 30° 顶角定义等腰三角形的方法。



```
\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}
```

4.2.2 等边三角形

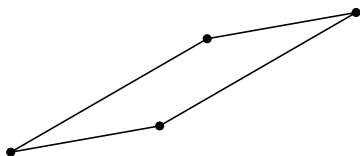
下面的代码给出了一种极为简单的定义等边三角形的方法。



```
\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}
```

4.2.3 平行四边形

简单的定义平行四边形的方式为：



```
\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}
```

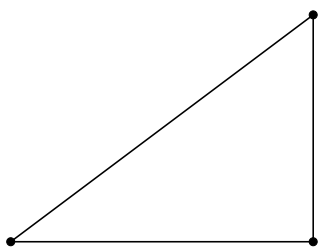
4.3 \tkzDefPoints命令：定义点集

\tkzDefPoints[<命令选项>]{< $x_1/y_1/n_1, x_2/y_2/n_2, \dots$ >}

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	\tkzDefPoints{0/0/0,2/2/A}
选项	默认值	含义
shift	无	为所有点添加 (x,y) 或 (α:d) 偏移

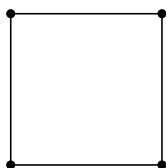
4.4 创建三角形



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

4.5 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}
```

5 特殊点

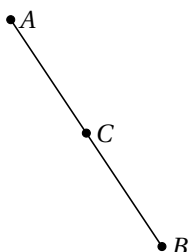
tkz-base宏包中点的定义方法就有这些，其中，最为重要的是**tkzDefPoint**命令，当然，还有一些特殊点的定义方法。

5.1 **\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

5.1.1 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}
```

5.2 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。

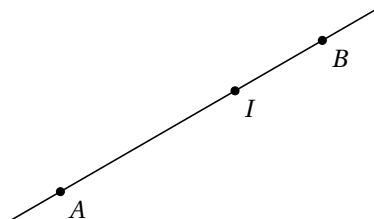
```
\tkzDefBarycentricPoint(<pt1= $\alpha_1$ ,pt2= $\alpha_2$ ,...>)
```

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

注意: 至少需要两个已知点, 才能实现计算。

5.2.1 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:

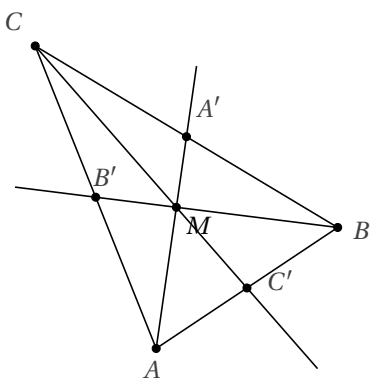


$$\vec{AI} = \frac{2}{3} \vec{AB}$$

```
\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}
```

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

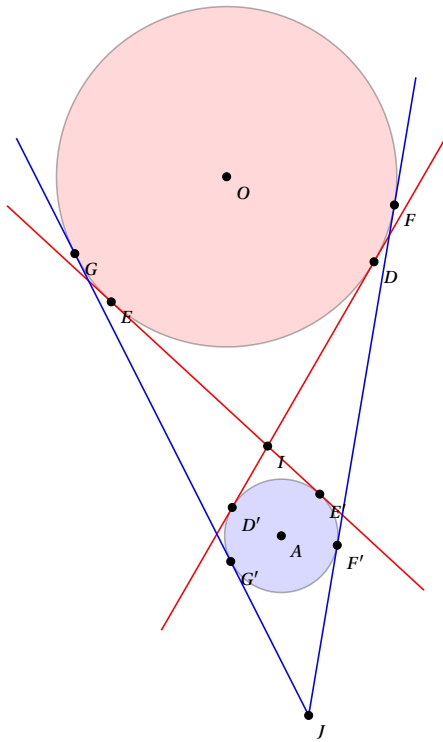
下面的代码中, M 是三角形的重心, 当然, 为了简化操作, 还有另外一个用于计算三角形重心的 `\tkzCentroid` 命令。



```
\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}
```

5.3 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}
\tkzExtSimilitudeCenter(0,3)(A,1)
\tkzGetPoint{J}
\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}
```

6 与三角形相关的特殊点

6.1 \tkzDefTriangleCenter命令：获取三角形中心

该命令用于获取三角形的中心。

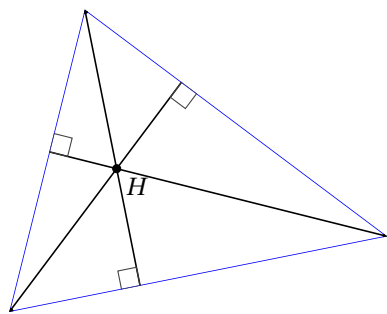
`\tkzDefTriangleCenter[(命令选项)](A,B,C)`

需要注意的是，该命令的参数必须是3个点的列表，通过该命令结合\tkzGetPoint能够得到中心。当然，如果不需要保留该中心，也可以使用tkzPointResult命令使用该中心。

参数	默认值	含义
(pt1,pt2,pt3)	无	逗号分隔的 3 个点的列表
选项	默认值	含义
ortho	circum	垂心，三条高的交点
centroid	circum	重心，三条中线的交点
circum	circum	外心，外接圆圆心
in	circum	内心，内切圆圆心
ex	circum	旁心，外切圆圆心
euler	circum	欧拉点，欧拉圆/费尔巴哈圆/九点圆圆心
symmedian	circum	Lemoine 点或中间点或 Grebe 点
spieker	circum	Spieker 圆心
nagel	circum	Nagel 点（界心），三个旁切圆切点与对应顶点连线的交点
mittenpunkt	circum	中间点
feuerbach	circum	Feuerbach 点

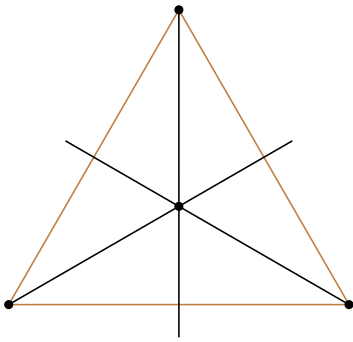
6.1.1 ortho或orthic选项

三条高的交点称为三角形的垂心



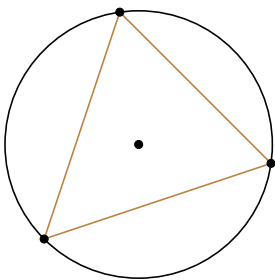
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,1){B}
\tkzDefPoint(1,4){C}
\tkzClipPolygon(A,B,C)
\tkzDefTriangleCenter[ortho](B,C,A)
\tkzGetPoint{H}
\tkzDefSpcTriangle[orthic,name=H](A,B,C){a,b,c}
\tkzDrawPolygon[color=blue](A,B,C)
\tkzDrawPoints(A,B,C,H)
\tkzDrawLines[add=0 and 1](A,Ha B,Hb C,Hc)
\tkzLabelPoint(H){H}
\tkzAutoLabelPoints[center=H](A,B,C)
\tkzMarkRightAngles(A,Ha,B B,Hb,C C,Hc,A)
\end{tikzpicture}
```

6.1.2 centroid选项



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{-1/1/A,5/1/B}
  \tkzDefEquilateral(A,B)
  \tkzGetPoint{C}
  \tkzDefTriangleCenter[centroid](A,B,C)
  \tkzGetPoint{G}
  \tkzDrawPolygon[color=brown](A,B,C)
  \tkzDrawPoints(A,B,C,G)
  \tkzDrawLines[add = 0 and 2/3](A,G B,G C,G)
\end{tikzpicture}
```

6.1.3 circum选项

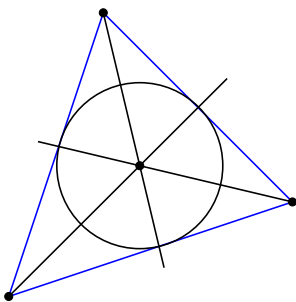


```
\begin{tikzpicture}
  \tkzDefPoints{0/1/A,3/2/B,1/4/C}
  \tkzDefTriangleCenter[circum](A,B,C)
  \tkzGetPoint{G}
  \tkzDrawPolygon[color=brown](A,B,C)
  \tkzDrawCircle(G,A)
  \tkzDrawPoints(A,B,C,G)
\end{tikzpicture}
```

6.1.4 in选项

几何学中，三角形的内切圆是三角形中最大的圆，内切圆的圆心称为三角形的内心。三角形的内心是三角形三个内角角平分线的交点。旁切圆圆心是一个顶点(例如顶点 A)的内角角平分线与另外两个外角角平分线的交点。该旁切圆圆心是相对于顶点 A 的一个旁心，或叫作 A 的旁心。因为三角形内角角平分线与对应的外角角平分线垂直，所以，内心与 3 个旁心构成了一个正交系统。(https://en.wikipedia.org/wiki/Incircle_and_excircles_of_a_triangle)

得到的内心存储在 `tkzPointResult` 命令中，可以通过 `\tkzGetPoint` 检索该点。

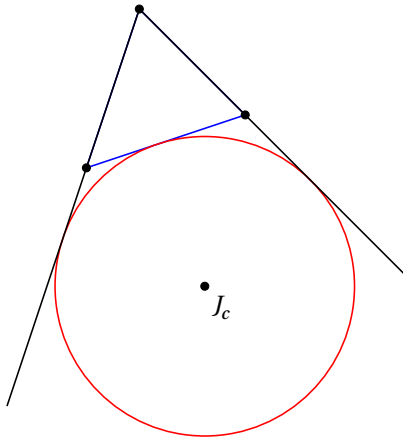


```
\begin{tikzpicture}[scale=1.25]
  \tkzDefPoints{0/1/A,3/2/B,1/4/C}
  \tkzDefTriangleCenter[in](A,B,C)\tkzGetPoint{I}
  \tkzDefPointBy[projection=onto A--C](I)
  \tkzGetPoint{Ib}
  \tkzDrawPolygon[color=blue](A,B,C)
  \tkzDrawPoints(A,B,C,I)
  \tkzDrawLines[add = 0 and 2/3](A,I B,I C,I)
  \tkzDrawCircle(I,Ib)
\end{tikzpicture}
```


6.1.5 ex选项

旁切圆是位于三角形外部与某条边及另外两条边的延长线相切的圆，一个三角形有 3 个旁切圆。(https://en.wikipedia.org/wiki/Incircle_and_excircles_of_a_triangle)

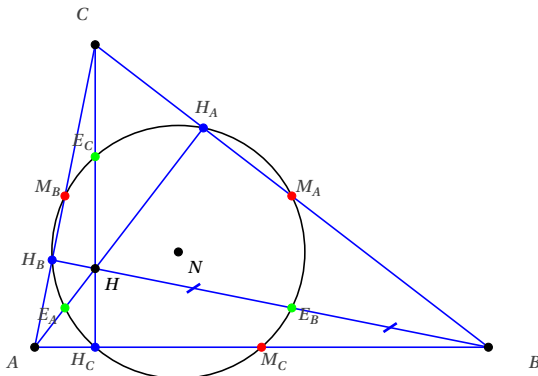
得到的旁心存储在`tkzPointResult`命令中，可以通过`\tkzGetPoint`检索该点。



```
\begin{tikzpicture}[scale=0.70]
  \tkzDefPoints{0/1/A,3/2/B,1/4/C}
  \tkzDefTriangleCenter[ex](B,C,A)
  \tkzGetPoint{J_c}
  \tkzDefPointBy[projection=onto A--B](J_c)
  \tkzGetPoint{Tc}
  %or
  % \tkzDefCircle[ex](B,C,A)
  % \tkzGetFirstPoint{J_c}
  % \tkzGetSecondPoint{Tc}
  \tkzDrawPolygon[color=blue](A,B,C)
  \tkzDrawPoints(A,B,C,J_c)
  \tkzDrawCircle[red](J_c,Tc)
  \tkzDrawLines[add=1.5 and 0](A,C B,C)
  \tkzLabelPoints(J_c)
\end{tikzpicture}
```

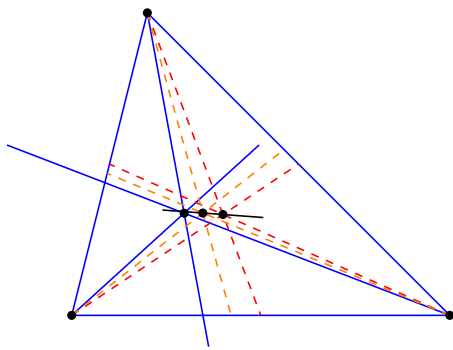
6.1.6 euler选项

该命令用于获得 9 点圆或称欧拉圆或称费尔巴哈圆的圆心。欧拉圆是通过三角形 ABC 三个顶点向对边作垂线形成的三个垂脚 H_A 、 H_B 和 H_C 的圆，欧拉在 1765 年证明该圆同时通过三角形 ABC 三个边的中点 M_A 、 M_B 和 M_C 。根据费尔巴哈定理，9 点圆也通过三角形 ABC 三个顶点与重心 H 连线线段的中点 E_A 、 E_B 和 E_C 。欧拉圆的圆心也称为欧拉点。(https://mathworld.wolfram.com/Nine-PointCircle.html)



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
  \tkzDefSpcTriangle[medial,
    name=M](A,B,C){_A,_B,_C}
  \tkzDefTriangleCenter[euler](A,B,C)
  \tkzGetPoint{N} % I= N nine points
  \tkzDefTriangleCenter[ortho](A,B,C)
  \tkzGetPoint{H}
  \tkzDefMidPoint(A,H) \tkzGetPoint{E_A}
  \tkzDefMidPoint(C,H) \tkzGetPoint{E_C}
  \tkzDefMidPoint(B,H) \tkzGetPoint{E_B}
  \tkzDefSpcTriangle[ortho,name=H](A,B,C){_A,_B,_C}
  \tkzDrawPolygon[color=blue](A,B,C)
  \tkzDrawCircle(N,E_A)
  \tkzDrawSegments[blue](A,H_A B,H_B C,H_C)
  \tkzDrawPoints(A,B,C,N,H)
  \tkzDrawPoints[red](M_A,M_B,M_C)
  \tkzDrawPoints[blue](H_A,H_B,H_C)
  \tkzDrawPoints[green](E_A,E_B,E_C)
  \tkzAutoLabelPoints[center=N, font=\scriptsize]%
  (A,B,C,M_A,M_B,M_C,H_A,H_B,H_C,E_A,E_B,E_C)
  \tkzLabelPoints[font=\scriptsize](H,N)
  \tkzMarkSegments[mark=s|,size=3pt,
    color=blue,line width=1pt](B,E_B E_B,H)
\end{tikzpicture}
```

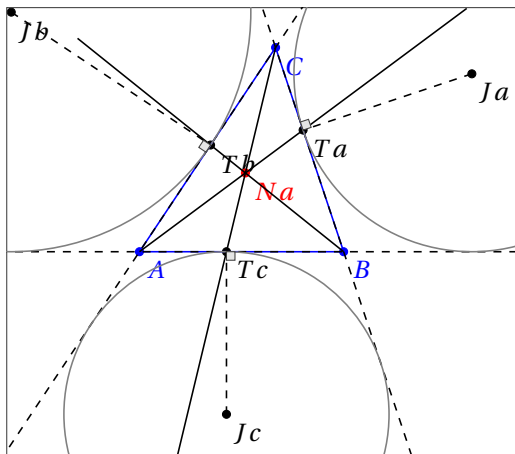
6.1.7 symmedian选项



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,0){B}
\tkzDefPoint(1,4){C}
\tkzDefTriangleCenter[symmedian](A,B,C) \tkzGetPoint{K}
\tkzDefTriangleCenter[median](A,B,C) \tkzGetPoint{G}
\tkzDefTriangleCenter[in](A,B,C) \tkzGetPoint{I}
\tkzDefSpcTriangle[centroid,name=M](A,B,C){a,b,c}
\tkzDefSpcTriangle[incentral,name=I](A,B,C){a,b,c}
\tkzDrawPolygon[color=blue](A,B,C)
\tkzDrawLines[add = 0 and 2/3,blue](A,K B,K C,K)
\tkzDrawSegments[red,dashed](A,Ma B,Mb C,Mc)
\tkzDrawSegments[orange,dashed](A,Ia B,Ib C,Ic)
\tkzDrawLine[add=2 and 2](G,I)
\tkzDrawPoints(A,B,C,K,G,I)
\end{tikzpicture}
```

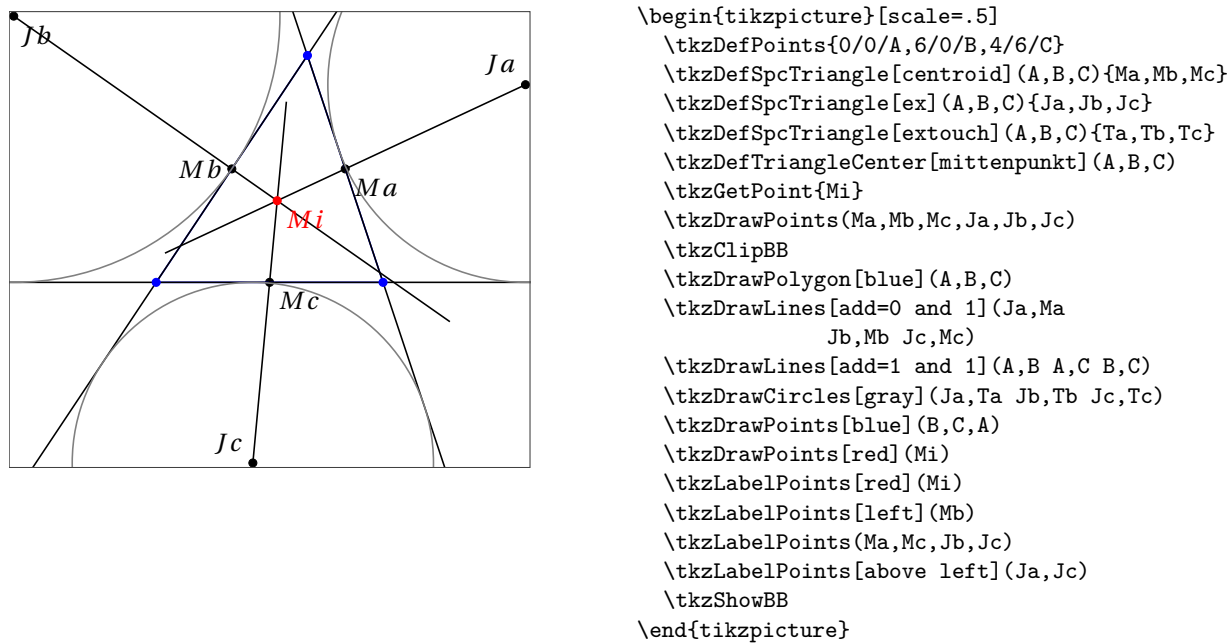
6.1.8 nagel选项

令 Ta 、 Tb 和 Tc 分别为旁切圆与三角形三条边的切点，连线 ATa 、 BTb 和 CTc ，其交点称为 Nagel 点，俗称三角形的“界心”。Weisstein, Eric W. “Nagel point”. From MathWorld—A Wolfram Web Resource.



```
\begin{tikzpicture}[scale=0.45]
\tkzDefPoints{0/0/A,6/0/B,4/6/C}
\tkzDefSpcTriangle[ex](A,B,C){Ja,Jb,Jc}
\tkzDefSpcTriangle[extouch](A,B,C){Ta,Tb,Tc}
\tkzDrawPoints(Ja,Jb,Jc,Ta,Tb,Tc)
\tkzLabelPoints(Ja,Jb,Jc,Ta,Tb,Tc)
\tkzDrawPolygon[blue](A,B,C)
\tkzDefTriangleCenter[nagel](A,B,C) \tkzGetPoint{Na}
\tkzDrawPoints[blue](B,C,A)
\tkzDrawPoints[red](Na)
\tkzLabelPoints[blue](B,C,A)
\tkzLabelPoints[red](Na)
\tkzDrawLines[add=0 and 1](A,Ta B,Tb C,Tc)
\tkzShowBB\tkzClipBB
\tkzDrawLines[add=1 and 1,dashed](A,B B,C C,A)
\tkzDrawCircles[ex,gray](A,B,C C,A,B B,C,A)
\tkzDrawSegments[dashed](Ja,Ta Jb,Tb Jc,Tc)
\tkzMarkRightAngles[fill=gray!20](Ja,Ta,C
Jb,Tb,A Jc,Tc,B)
\end{tikzpicture}
```

6.1.9 mittenpunkt选项



7 点的绘制

7.0.1 使用\tkzDrawPoint命令绘制点

\tkzDrawPoint[< 命令选项>](< 名称>)

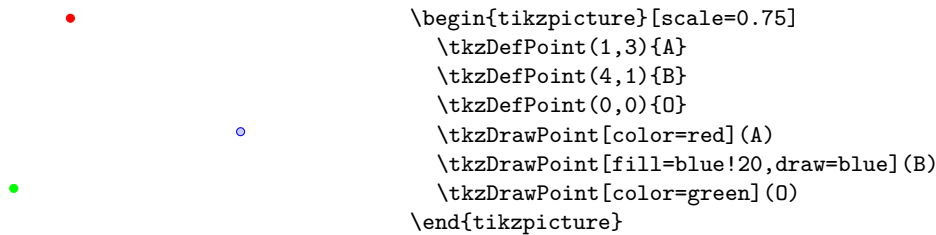
参数	默认值	含义
点的名称	无	只能有一个点的名称

这是一个必选参数，圆盘采用填充色绘制，但颜色较浅。可以通过选项实现更多绘制效果，点采用 node 的方式进行绘制，因此，当执行缩放操作时，点的大小不受影响。

选项	默认值	含义
shape	circle	可以使用cross或cross out 能够创建如cross的形式。
size	6	6× \pgflinewidth
color	black	默认颜色，可以被修改

7.0.2 点的绘制实例

注意，缩放不会影响点的形状，多数情况下，无论采用宏还是通过修改配置文件从一开始就定义一个点的形状，一般都可以得到令人满意的效果。



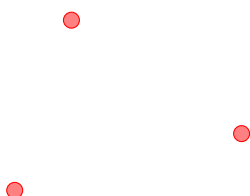
也可以一次绘制多个点，但是该命令会比绘制单个点的命令慢。另外，一次绘制多个点时，则表示对所有点使用相同的绘制选项。

`\tkzDrawPoints[< 命令选项>](< 点列表>)`

参数	默认值	含义
点列表	无	例如: <code>\tkzDrawPoints(A,B,C)</code> , 注意各点间用逗号分隔。
选项	默认值	含义
shape	circle	可以是cross或cross out
size	6	6× <code>\pgflinewidth</code>
color	black	默认为黑色, 可以被修改

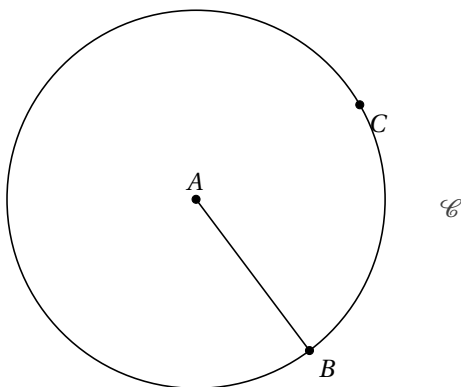
注意命令最后的“s”，如果没有这个“s”则会发生错误。

7.0.3 第 1 个示例代码



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(1,3){A}
  \tkzDefPoint(4,1){B}
  \tkzDefPoint(0,0){C}
  \tkzDrawPoints[size=6,color=red,
                  fill=red!50](A,B,C)
\end{tikzpicture}
```

7.0.4 第 2 个示例代码



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(2,3){A} \tkzDefPoint(5,-1){B}
\tkzDefPoint[label=below:$\mathcal{C}$,
shift={(2,3)}](-30:5.5){E}
\begin{scope}[shift=(A)]
\tkzDefPoint(30:5){C}
\end{scope}
\tkzCalcLength[cm](A,B)\tkzGetLength{rAB}
\tkzDrawCircle[R](A,\rAB cm)
\tkzDrawSegment(A,B)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(B,C)
\tkzLabelPoints[above](A)
\end{tikzpicture}
```

8 直线或圆上的点

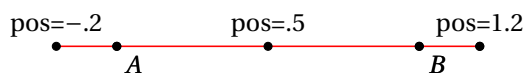
8.1 直线上的点

$$\backslash\mathrm{tkzDefPointOnLine}[\langle \text{命令选项} \rangle](\langle A,B \rangle)$$

参数	默认值	含义
pt1,pt2	无	定义直线的两个点

选项	默认值	含义
pos=nb	无	比例, nb 是一个十进制数

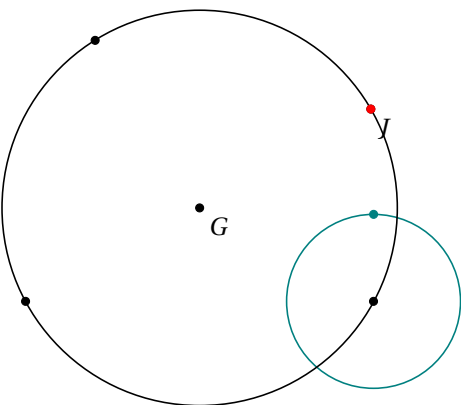
8.1.1 pos选项



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/0/B}
\tkzDrawLine[red](A,B)
\tkzDefPointOnLine[pos=1.2](A,B)
\tkzGetPoint{P}
\tkzDefPointOnLine[pos=-0.2](A,B)
\tkzGetPoint{R}
\tkzDefPointOnLine[pos=0.5](A,B)
\tkzGetPoint{S}
\tkzDrawPoints(A,B,P)
\tkzLabelPoints(A,B)
\tkzLabelPoint[above](P){pos=$1.2$}
\tkzLabelPoint[above](R){pos=$-.2$}
\tkzLabelPoint[above](S){pos=$.5$}
\tkzDrawPoints(A,B,P,R,S)
\tkzLabelPoints(A,B)
\end{tikzpicture}
```

8.2 圆上的点

\tkzDefPointOnCircle[< 命令选项>]		
选项	默认值	含义
angle	0	与横轴夹角
center	\tkzPointResult	圆心
radius	\tkzLengthResult	半径



```
\begin{tikzpicture}[scale=1.15]
\tkzDefPoints{0/0/A,4/0/B,0.8/3/C}
\tkzDefPointOnCircle[angle=90,center=B,radius=1 cm]
\tkzGetPoint{I}
\tkzDefCircle[circum](A,B,C)
\tkzGetPoint{G} \tkzGetLength{rG}
\tkzDefPointOnCircle[angle=30,center=G,radius=\rG pt]
\tkzGetPoint{J}
\tkzDrawCircle[R,teal](B,1cm)
\tkzDrawPoint[teal](I)
\tkzDrawPoints(A,B,C)
\tkzDrawCircle(G,J)
\tkzDrawPoints(G,J)
\tkzDrawPoint[red](J)
\tkzLabelPoints(G,J)
\end{tikzpicture}
```

9 利用变换命令\tkzDefPointBy定义点

这些变换主要有：

- 平移;
- 放大;
- 正交反射或对称;
- 中心对称;
- 正交投影;
- 旋转 (度或弧度);
- 相对于圆的反转.

使用\tkzDefPointBy命令实现单点变换，通过\tkzDefPointsBy实现多点变换，变换方式通过可选项指定。默认用点 A' 表示点 A 的变换结果，例如：

```
\tkzDefPointBy[translation= from A to A'](B)
```

结果保存于tkzPointResult命令中。

\tkzDefPointBy[< 命令选项>](<pt>)

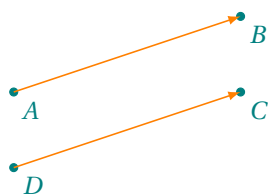
参数是一个已存在的点，其变换结果存储于tkzPointResult中，可使用\tkzGetPoint{M}命令将结果点命名为M。

参数	含义	样例
pt	已存在的一个点的名称	(A)
选项		样例
translation	= from #1 to #2	[translation=from A to B] (E)
homothety	= center #1 ratio #2	[homothety=center A ratio .5] (E)
reflection	= over #1--#2	[reflection=over A--B] (E)
symmetry	= center #1	[symmetry=center A] (E)
projection	= onto #1--#2	[projection=onto A--B] (E)
rotation	= center #1 angle #2	[rotation=center O angle 30] (E)
rotation in rad	= center #1 angle #2	[rotation in rad=center O angle pi/3] (E)
inversion	= center #1 through #2	[inversion =center O through A] (E)

该命令仅定义一个变换后的点，而并不绘制该点。

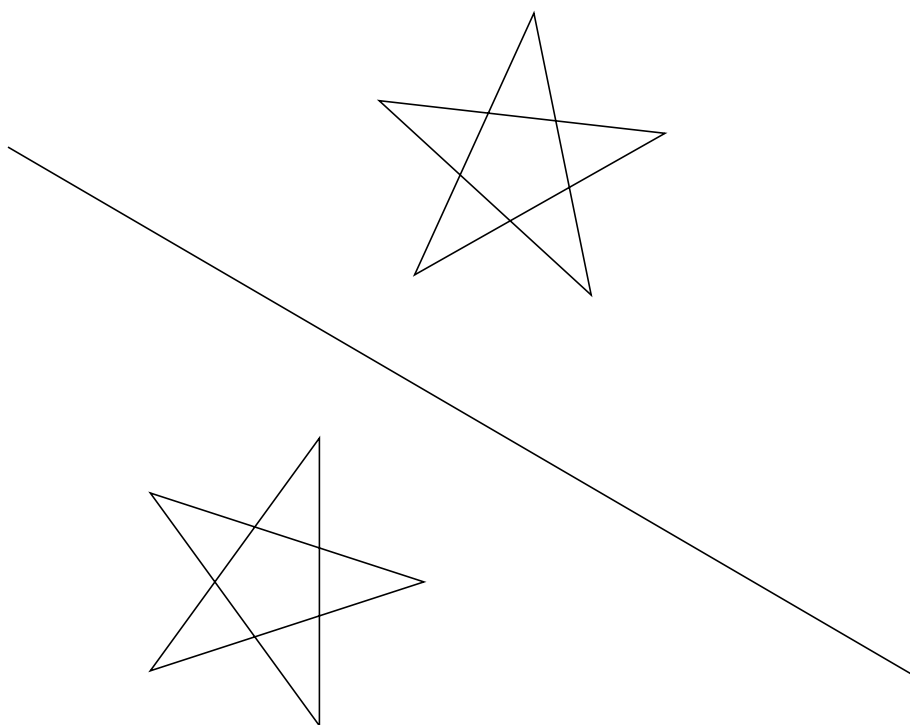
9.1 变换示例

9.1.1 平移示例



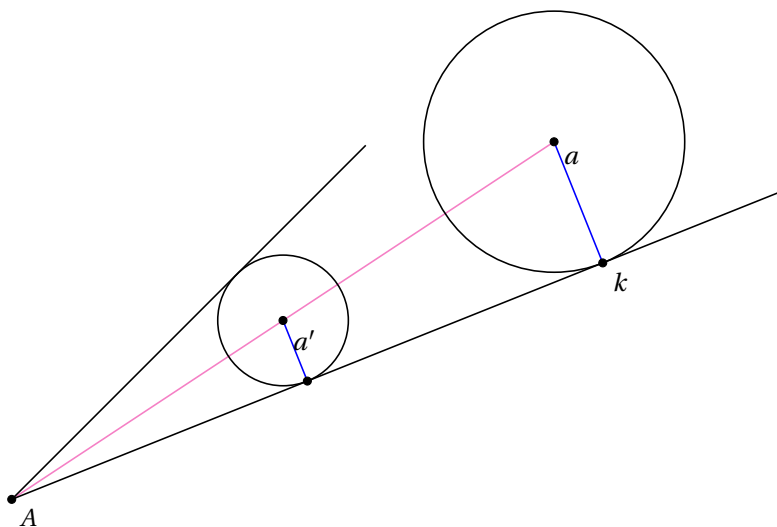
```
\begin{tikzpicture}[>=latex]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,1){B}
\tkzDefPoint(3,0){C}
\tkzDefPointBy[translation= from B to A](C)
\tkzGetPoint{D}
\tkzDrawPoints[teal](A,B,C,D)
\tkzLabelPoints[color=teal](A,B,C,D)
\tkzDrawSegments[orange,->](A,B D,C)
\end{tikzpicture}
```

9.1.2 反射示例 (orthogonal symmetry)



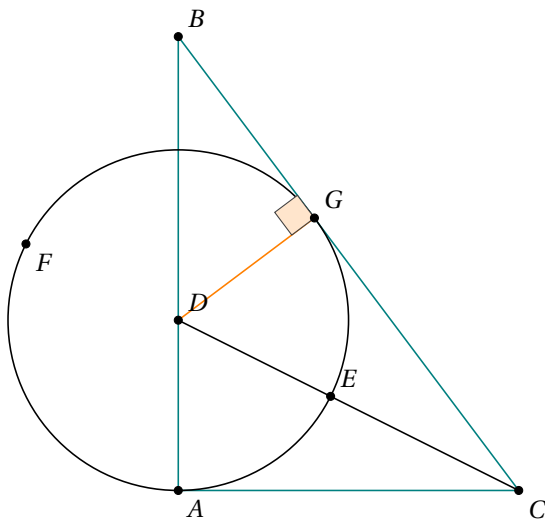
```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{1.5/-1.5/C,-4.5/2/D}
\tkzDefPoint(-4,-2){O}
\tkzDefPoint(-2,-2){A}
\foreach \i in {0,1,...,4}{%
  \pgfmathparse{0+\i * 72}
  \tkzDefPointBy[rotation=%
    center O angle \pgfmathresult](A)
  \tkzGetPoint{A\i}
  \tkzDefPointBy[reflection = over C--D](A\i)
  \tkzGetPoint{A'\i}}
\tkzDrawPolygon(A0, A2, A4, A1, A3)
\tkzDrawPolygon(A0', A2', A4', A1', A3')
\tkzDrawLine[add= .5 and .5](C,D)
\end{tikzpicture}
```

9.1.3 homothety和projection示例



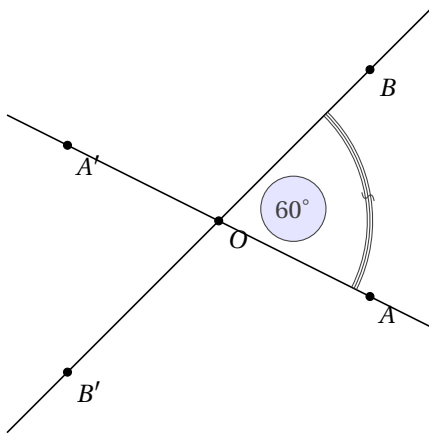
```
\begin{tikzpicture}[scale=1.2]
\tkzDefPoint(0,1){A} \tkzDefPoint(5,3){B} \tkzDefPoint(3,4){C}
\tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
\tkzDrawLine[add=0 and 0,color=magenta!50](A,a)
\tkzDefPointBy[homothety=center A ratio .5](a) \tkzGetPoint{a'}
\tkzDefPointBy[projection=onto A--B](a') \tkzGetPoint{k'}
\tkzDefPointBy[projection=onto A--B](a) \tkzGetPoint{k}
\tkzDrawLines[add=0 and .3](A,k,A,C)
\tkzDrawSegments[blue](a',k'a,k)
\tkzDrawPoints(a,a',k,k',A)
\tkzDrawCircles(a',k'a,k)
\tkzLabelPoints(a,a',k,A)
\end{tikzpicture}
```

9.1.4 投影示例



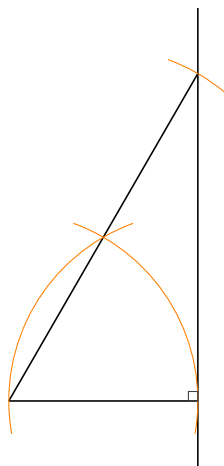
```
\begin{tikzpicture}[scale=1.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(0,4){B}
\tkzDefTriangle[pythagore](B,A) \tkzGetPoint{C}
\tkzDefLine[bisector](B,C,A) \tkzGetPoint{c}
\tkzInterLL(C,c)(A,B) \tkzGetPoint{D}
\tkzDefPointBy[projection=onto B--C](D)
\tkzGetPoint{G}
\tkzInterLC(C,D)(D,A) \tkzGetPoints{E}{F}
\tkzDrawPolygon[teal](A,B,C)
\tkzDrawSegment(C,D) \tkzDrawCircle(D,A)
\tkzDrawSegment[orange](D,G)
\tkzMarkRightAngle[fill=orange!20](D,G,B)
\tkzDrawPoints(A,C,F) \tkzLabelPoints(A,C,F)
\tkzDrawPoints(B,D,E,G)
\tkzLabelPoints[above right](B,D,E,G)
\end{tikzpicture}
```


9.1.5 对称示例



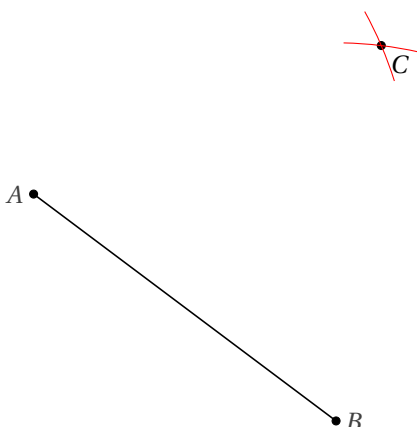
```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,-1){A}
\tkzDefPoint(2,2){B}
\tkzDefPointsBy[symmetry=center O](B,A){}
\tkzDrawLine(A,A')
\tkzDrawLine(B,B')
\tkzMarkAngle[mark=s,arc=lll,
size=2 cm,mkcolor=red](A,O,B)
\tkzLabelAngle[pos=1,circle,draw,
fill=blue!10](A,O,B){$60^\circ$}
\tkzDrawPoints(A,B,O,A',B')
\tkzLabelPoints(A,B,O,A',B')
\end{tikzpicture}
```

9.1.6 旋转示例 (度)



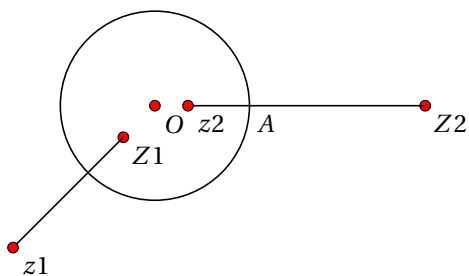
```
\begin{tikzpicture}[scale=0.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,0){B}
\tkzDrawSegment(A,B)
\tkzDefPointBy[rotation=center A angle 60](B)
\tkzGetPoint{C}
\tkzDefPointBy[symmetry=center C](A)
\tkzGetPoint{D}
\tkzDrawSegment(A,t kzPointResult)
\tkzDrawLine(B,D)
\tkzDrawArc[orange,delta=10](A,B)(C)
\tkzDrawArc[orange,delta=10](B,C)(A)
\tkzDrawArc[orange,delta=10](C,D)(D)
\tkzMarkRightAngle(D,B,A)
\end{tikzpicture}
```

9.1.7 旋转示例 (弧度)



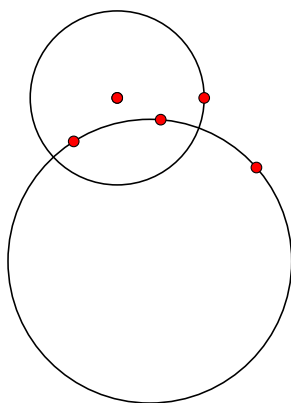
```
\begin{tikzpicture}
\tkzDefPoint["$A$" left](1,5){A}
\tkzDefPoint["$B$" right](5,2){B}
\tkzDefPointBy[rotation in rad=center A angle pi/3](B)
\tkzGetPoint{C}
\tkzDrawSegment(A,B)
\tkzDrawPoints(A,B,C)
\tkzCompass[color=red](A,C)
\tkzCompass[color=red](B,C)
\tkzLabelPoints(C)
\end{tikzpicture}
```

9.1.8 点反转示例



```
\begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){A}
  \tkzDefPoint(-1.5,-1.5){z1}
  \tkzDefPoint(0.35,0){z2}
  \tkzDefPointBy[inversion = center O through A](z1)
  \tkzGetPoint{Z1}
  \tkzDefPointBy[inversion = center O through A](z2)
  \tkzGetPoint{Z2}
  \tkzDrawCircle(O,A)
  \tkzDrawPoints[color=black,fill=red,size=4](Z1,Z2)
  \tkzDrawSegments(z1,Z1 z2,Z2)
  \tkzDrawPoints[color=black,fill=red,size=4](O,z1,z2)
  \tkzLabelPoints(O,A,z1,z2,Z1,Z2)
\end{tikzpicture}
```

9.1.9 点的反转：正交圆



```
\begin{tikzpicture}[scale=1.15]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){A}
  \tkzDrawCircle(O,A)
  \tkzDefPoint(0.5,-0.25){z1}
  \tkzDefPoint(-0.5,-0.5){z2}
  \tkzDefPointBy[inversion = center O through A](z1)
  \tkzGetPoint{Z1}
  \tkzCircumCenter(z1,z2,Z1)
  \tkzGetPoint{c}
  \tkzDrawCircle(c,Z1)
  \tkzDrawPoints[color=black,fill=red,size=4]%(O,z1,z2,Z1,O,A)
\end{tikzpicture}
```

9.2 用\tkzDefPointsBy命令实现多点变换

是单点变换命令的变体命令，用于定义多个点的变换。必须通过参数指定变换点名称，或是指定变换后点的名称，并让参数留空。

```
\tkzDefPointsBy[translation= from A to A'](B,C){}
```

变换后的点是 B' 和 C' 。

```
\tkzDefPointsBy[translation= from A to A'](B,C){D,E}
```

变换后的点是 D 和 E 。

```
\tkzDefPointsBy[translation= from A to A'](B)
```

变换后的点是 B'

`\tkzDefPointsBy[⟨ 命令选项 ⟩](⟨ 需要变换点列表 ⟩){⟨ 变换后点列表 ⟩}`

参数

示例

`(⟨ 需要变换点列表 ⟩){⟨ 变换后点列表 ⟩}` `(A,B){E,F}` E 是 A 的变换, F 是 B 的变换。

如果变换后点列表为空, 变换后点的名称在原名称后添加“'”。

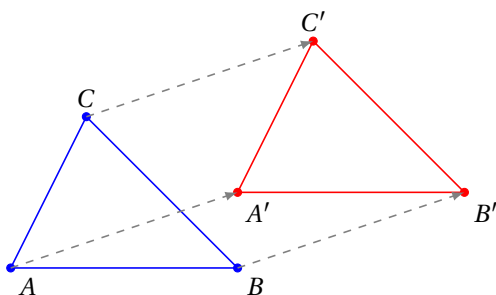
选项

示例

<code>translation = from #1 to #2</code>	<code>[translation=from A to B] (E){}</code>
<code>homothety = center #1 ratio #2</code>	<code>[homothety=center A ratio .5] (E){F}</code>
<code>reflection = over #1--#2</code>	<code>[reflection=over A--B] (E){F}</code>
<code>symmetry = center #1</code>	<code>[symmetry=center A] (E){F}</code>
<code>projection = onto #1--#2</code>	<code>[projection=onto A--B] (E){F}</code>
<code>rotation = center #1 angle #2</code>	<code>[rotation=center angle 30] (E){F}</code>
<code>rotation in rad = center #1 angle #2</code>	<code>for instance angle pi/3</code>

该命令仅定义变换后的点, 而并不绘制这些点。

9.2.1 变换示例



```
\begin{tikzpicture}[>=latex]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,1){A'}
\tkzDefPoint(3,0){B}
\tkzDefPoint(1,2){C}
\tkzDefPointsBy[translation= from A to A'](B,C){}
\tkzDrawPolygon[color=blue](A,B,C)
\tkzDrawPolygon[color=red](A',B',C')
\tkzDrawPoints[color=blue](A,B,C)
\tkzDrawPoints[color=red](A',B',C')
\tkzLabelPoints(A,B,A',B')
\tkzLabelPoints[above](C,C')
\tkzDrawSegments[color = gray,->,%
style=dashed](A,A' B,B' C,C')
\end{tikzpicture}
```

10 通过向量定义点

10.1 \tkzDefPointWith命令

可以通过多种方案定义满足特定向量条件的点, There are several possibilities to create points that meet certain vector conditions. 可以使用\tkzDefPointWith命令定义这样的点。一般原理为: 需要用两个点作为参数, 也就是一个向量。不同的选项可以实现通过共线向量或正交向量的方式生成新的点, 向量的长度可以与第1个向量的长度成正比, 也可以与单位向量成正比。由于该点仅做临时使用, 因此, 不需要立即命名。其结果保存在tkzPointResult中, 可以使用\tkzGetPoint命令检索该点, 并为其命名。

可以通过选项设置指定点与所求点之间的距离, 通常, 该距离是参数中给定2个点之间的距离, 如果使用了“normed”选项, 则给定2点之间的距离和获得的点的距离为1 cm。然后通过比例系数 K 选项对其进行缩放。

\tkzDefPointWith(<pt1,pt2>)

是满足向量条件的点的定义。

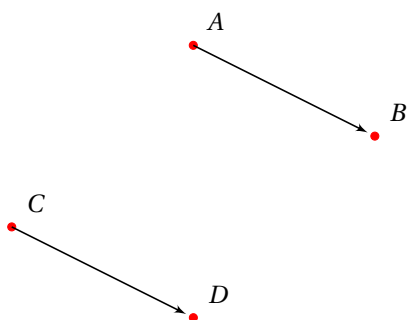
参数	含义	说明
(pt1,pt2)	点对	结果是保存在tkzPointResult命令中的点

在下文中, 假定由\tkzGetPoint{C}得到了该点。

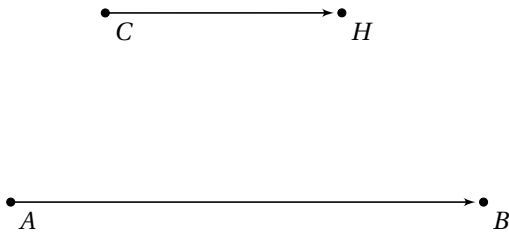
选项	样例	说明
orthogonal	[orthogonal] (A,B)	$AC = AB$ 和 $\overrightarrow{AC} \perp \overrightarrow{AB}$
orthogonal normed	[orthogonal normed] (A,B)	$AC = 1$ 和 $\overrightarrow{AC} \perp \overrightarrow{AB}$
linear	[linear] (A,B)	$\overrightarrow{AC} = K \times \overrightarrow{AB}$
linear normed	[linear normed] (A,B)	$AC = K$ 和 $\overrightarrow{AC} = k \times \overrightarrow{AB}$
colinear= at #1	[colinear= at C] (A,B)	$\overrightarrow{CD} = \overrightarrow{AB}$
colinear normed= at #1	[colinear normed= at C] (A,B)	$\overrightarrow{CD} = \overrightarrow{AB}$
K	[linear] (A,B), K=2	$\overrightarrow{AC} = 2 \times \overrightarrow{AB}$

10.1.1 colinear at选项

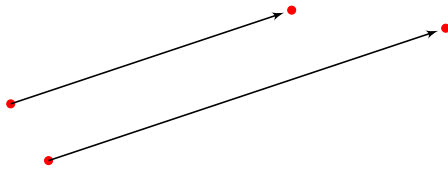
$$(\overrightarrow{AB} = \overrightarrow{CD})$$



```
\begin{tikzpicture}[vect/.style={->, shorten >=3pt,
>=latex'}, scale=1.2]
\tkzDefPoint(2,3){A}
\tkzDefPoint(4,2){B}
\tkzDefPoint(0,1){C}
\tkzDefPointWith[colinear=at C] (A,B)
\tkzGetPoint{D}
\tkzDrawPoints[color=red] (A,B,C,D)
\tkzLabelPoints[above right=3pt] (A,B,C,D)
\tkzDrawSegments[vect] (A,B C,D)
\end{tikzpicture}
```

10.1.2 colinear at带 K 选项

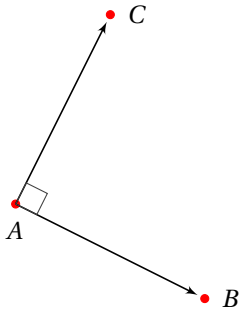
```
\begin{tikzpicture}[vect/.style={->, shorten >=3pt,
    >=latex'}, scale=1.25]
\tkzDefPoint(0,0){A} \tkzDefPoint(5,0){B}
\tkzDefPoint(1,2){C}
\tkzDefPointWith[colinear=at C](A,B)
\tkzGetPoint{G}
\tkzDefPointWith[colinear=at C,K=0.5](A,B)
\tkzGetPoint{H}
\tkzLabelPoints(A,B,C,G,H)
\tkzDrawPoints(A,B,C,G,H)
\tkzDrawSegments[vect](A,B C,H)
\end{tikzpicture}
```

10.1.3 colinear at带 $K = \frac{\sqrt{2}}{2}$ 选项

```
\begin{tikzpicture}[vect/.style={->, shorten >=3pt,
    >=latex'}, scale=1.75]
\tkzDefPoint(1,1){A} \tkzDefPoint(4,2){B}
\tkzDefPoint(2,2){C}
\tkzDefPointWith[colinear=at C,K=sqrt(2)/2](A,B)
\tkzGetPoint{D}
\tkzDrawPoints[color=red](A,B,C,D)
\tkzDrawSegments[vect](A,B C,D)
\end{tikzpicture}
```

10.1.4 orthogonal

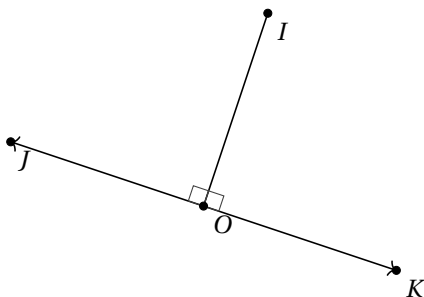
因 $K = 1$, 所以 $AB = AC$ 。



```
\begin{tikzpicture}[vect/.style={->,shorten >=3pt,
    >=latex'},scale=1.25]
\tkzDefPoint(2,3){A} \tkzDefPoint(4,2){B}
\tkzDefPointWith[orthogonal,K=1](A,B)
\tkzGetPoint{C}
\tkzDrawPoints[color=red](A,B,C)
\tkzLabelPoints[right=3pt](B,C)
\tkzLabelPoints[below=3pt](A)
\tkzDrawSegments[vect](A,B A,C)
\tkzMarkRightAngle(B,A,C)
\end{tikzpicture}
```

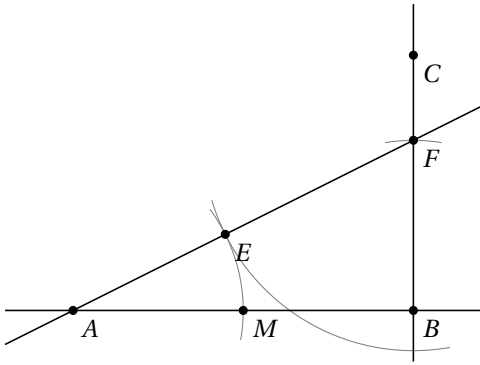
10.1.5 orthogonal带 $K = -1$ 选项

因 $|K| = 1$, 所以 $OI = OJ = OK$ 。



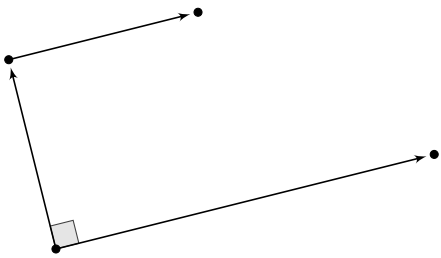
```
\begin{tikzpicture}[scale=0.85]
\tkzDefPoint(1,2){O} \tkzDefPoint(2,5){I}
\tkzDefPointWith[orthogonal](O,I)
\tkzGetPoint{J}
\tkzDefPointWith[orthogonal,K=-1](O,I)
\tkzGetPoint{K}
\tkzDrawSegment(O,I)
\tkzDrawSegments[->](O,J O,K)
\tkzMarkRightAngles(I,O,J I,O,K)
\tkzDrawPoints(O,I,J,K)
\tkzLabelPoints(O,I,J,K)
\end{tikzpicture}
```

10.1.6 orthogonal选项的综合实例



```
\begin{tikzpicture}[scale=0.75]
\tkzDefPoints{0/0/A,6/0/B}
\tkzDefMidPoint(A,B)
\tkzGetPoint{I}
\tkzDefPointWith[orthogonal,K=-.75](B,A)
\tkzGetPoint{C}
\tkzInterLC(B,C)(B,I)
\tkzGetPoints{D}{F}
\tkzDuplicateSegment(B,F)(A,F)
\tkzGetPoint{E}
\tkzDrawArc[delta=10](F,E)(B)
\tkzInterLC(A,B)(A,E)
\tkzGetPoints{N}{M}
\tkzDrawArc[delta=10](A,M)(E)
\tkzDrawLines(A,B B,C A,F)
\tkzCompass(B,F)
\tkzDrawPoints(A,B,C,F,M,E)
\tkzLabelPoints(A,B,C,F,M,E)
\end{tikzpicture}
```

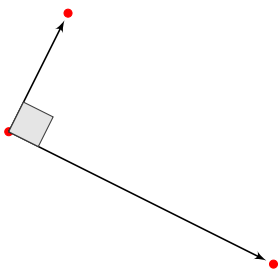
10.1.7 colinear和orthogonal选项



```
\begin{tikzpicture}[vect/.style={->,shorten >=3pt,
>=latex'}, scale=1.25]
\tkzDefPoint(2,1){A}
\tkzDefPoint(6,2){B}
\tkzDefPointWith[orthogonal,K=.5](A,B)
\tkzGetPoint{C}
\tkzDefPointWith[colinear=at C,K=.5](A,B)
\tkzGetPoint{D}
\tkzMarkRightAngle[fill=gray!20](B,A,C)
\tkzDrawSegments[vect](A,B A,C C,D)
\tkzDrawPoints(A,...,D)
\end{tikzpicture}
```

10.1.8 orthogonal normed带 $K=1$ 选项

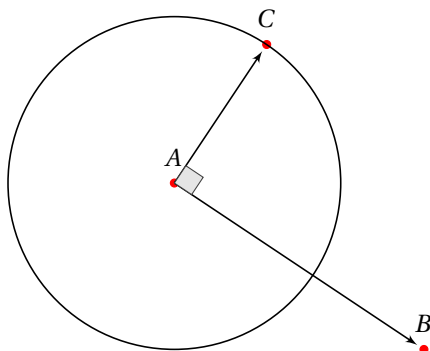
$AC = 1$.



```
\begin{tikzpicture}[vect/.style={->,shorten >=3pt,
>=latex'}, scale=1.75]
\tkzDefPoint(2,3){A}
\tkzDefPoint(4,2){B}
\tkzDefPointWith[orthogonal normed](A,B)
\tkzGetPoint{C}
\tkzDrawPoints[color=red](A,B,C)
\tkzDrawSegments[vect](A,B A,C)
\tkzMarkRightAngle[fill=gray!20](B,A,C)
\end{tikzpicture}
```

10.1.9 orthogonal normed和 $K=2$ 选项

因 $K=2$ ，所以 $AC=2$ 。

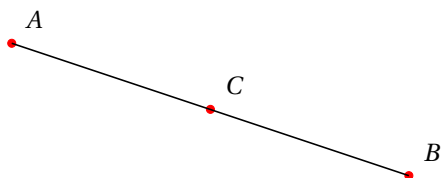


```
\begin{tikzpicture}[vect/.style={->,shorten >=3pt,
>=latex'}, scale=1.10]
\tkzDefPoint(2,3){A}
\tkzDefPoint(5,1){B}
\tkzDefPointWith[orthogonal normed,K=2](A,B)
\tkzGetPoint{C}
\tkzDrawPoints[color=red](A,B,C)
\tkzDrawCircle[R](A,2cm)
\tkzDrawSegments[vect](A,B A,C)
\tkzMarkRightAngle[fill=gray!20](B,A,C)
\tkzLabelPoints[above=3pt](A,B,C)
\end{tikzpicture}
```

10.1.10 linear选项

在此，取 $K=0.5$ 。

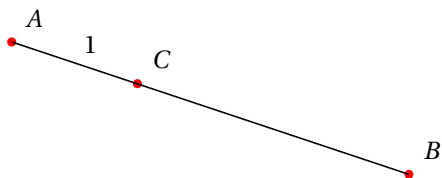
这相当于给一个向量乘了一个实数，本例中是 $[AB]$ 的中点。



```
\begin{tikzpicture}[scale=1.75]
\tkzDefPoint(1,3){A}
\tkzDefPoint(4,2){B}
\tkzDefPointWith[linear,K=0.5](A,B)
\tkzGetPoint{C}
\tkzDrawPoints[color=red](A,B,C)
\tkzDrawSegment(A,B)
\tkzLabelPoints[above right=3pt](A,B,C)
\end{tikzpicture}
```

10.1.11 linear normed选项

在下面的实例中， $AC=1$ 并且 C 属于 (AB) 。



```
\begin{tikzpicture}[scale=1.75]
\tkzDefPoint(1,3){A}
\tkzDefPoint(4,2){B}
\tkzDefPointWith[linear normed](A,B)
\tkzGetPoint{C}
\tkzDrawPoints[color=red](A,B,C)
\tkzDrawSegment(A,B)
\tkzLabelSegment(A,C){$1$}
\tkzLabelPoints[above right=3pt](A,B,C)
\end{tikzpicture}
```

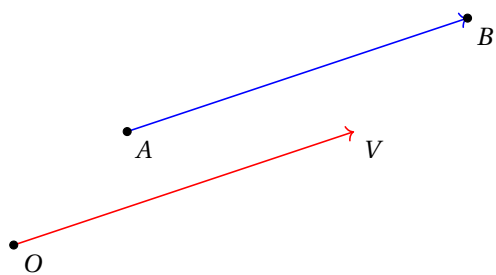
10.2 \tkzGetVectxy命令

检索一个向量的坐标。

```
\tkzGetVectxy(<A,B>){<text>}
```

获得一个向量的坐标。

参数	样例	说明
(point){name of macro}	\tkzGetVectxy(A,B){V}	V_x, V_y : 向量 \overrightarrow{AB} 的坐标

10.2.1 使用`\tkzGetVectxy`命令实现坐标变换

```

\begin{tikzpicture}[scale=1.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,1){A}
  \tkzDefPoint(4,2){B}
  \tkzGetVectxy(A,B){v}
  \tkzDefPoint(\vx,\vy){V}
  \tkzDrawSegment[->,color=red](O,V)
  \tkzDrawSegment[->,color=blue](A,B)
  \tkzDrawPoints(A,B,O)
  \tkzLabelPoints(A,B,O,V)
\end{tikzpicture}

```


11 定义随机点

可以使用以下四种方式定义随机点

1. 矩形内的点;
2. 线段上的点;
3. 直线上的点;
4. 圆上的点.

11.1 得到随机点

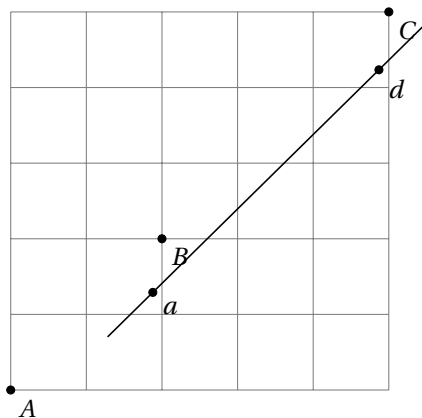
使用新的方式取代原`\tkzGetRandPointOn`命令，以定义随机点。

`\tkzDefRandPointOn`[< 命令选项>]

结果是可以由`\tkzGetPoint`命名的随机点，如果不需要保留结果，可以使用`\tkzPointResult`命令返回结果。

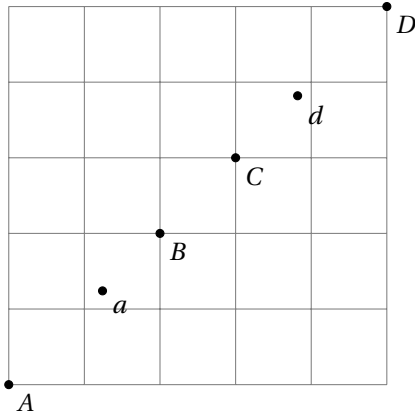
选项	默认值	含义
<code>rectangle=pt1 and pt2</code>		<code>[rectangle=A and B]</code>
<code>segment= pt1--pt2</code>		<code>[segment=A--B]</code>
<code>line=pt1--pt2</code>		<code>[line=A--B]</code>
<code>circle =center pt1 radius dim</code>		<code>[circle = center A radius 2 cm]</code>
<code>circle through=center pt1 through pt2</code>		<code>[circle through= center A through B]</code>
<code>disk through=center pt1 through pt2</code>		<code>[disk through=center A through B]</code>

11.2 矩形内的随机点



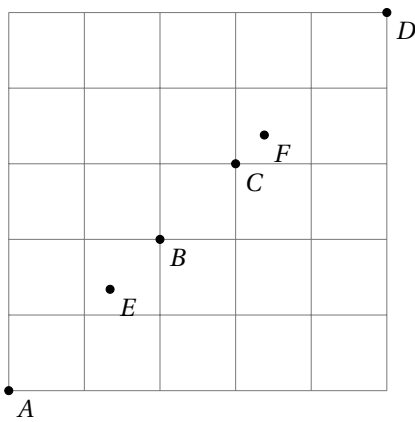
```
\begin{tikzpicture}
\tkzInit[xmax=5,ymax=5]
\tkzGrid
\tkzDefPoints{0/0/A,2/2/B,5/5/C}
\tkzDefRandPointOn[rectangle = A and B]
\tkzGetPoint{a}
\tkzDefRandPointOn[rectangle = B and C]
\tkzGetPoint{d}
\tkzDrawLine(a,d)
\tkzDrawPoints(A,B,C,a,d)
\tkzLabelPoints(A,B,C,a,d)
\end{tikzpicture}
```

11.3 线段上的随机点



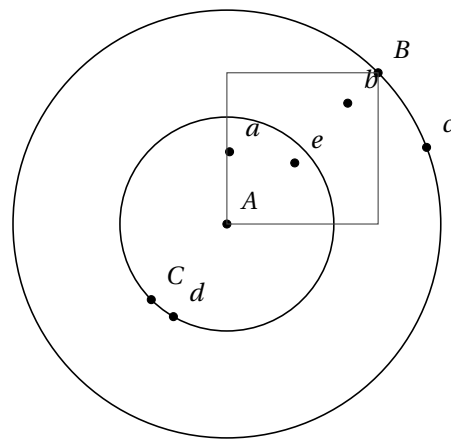
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzGrid
  \tkzDefPoints{0/0/A,2/2/B,3/3/C,5/5/D}
  \tkzDefRandPointOn[segment = A--B]\tkzGetPoint{a}
  \tkzDefRandPointOn[segment = C--D]\tkzGetPoint{d}
  \tkzDrawPoints(A,B,C,D,a,d)
  \tkzLabelPoints(A,B,C,D,a,d)
\end{tikzpicture}
```

11.4 直线上的随机点



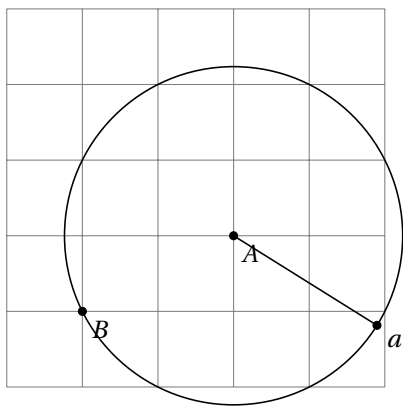
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzGrid
  \tkzDefPoints{0/0/A,2/2/B,3/3/C,5/5/D}
  \tkzDefRandPointOn[line = A--B]\tkzGetPoint{E}
  \tkzDefRandPointOn[line = C--D]\tkzGetPoint{F}
  \tkzDrawPoints(A,...,F)
  \tkzLabelPoints(A,...,F)
\end{tikzpicture}
```

11.4.1 随机点综合实例



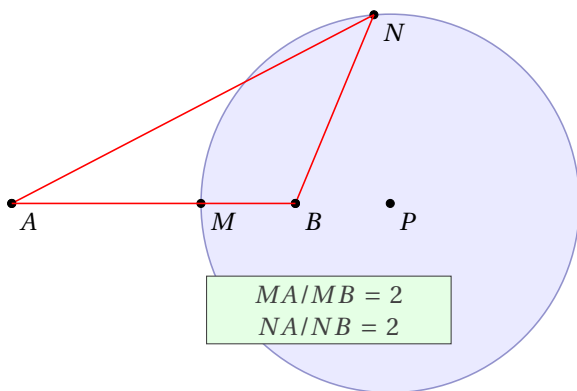
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,2/2/B,-1/-1/C}
  \tkzDefCircle[through=](A,C)
  \tkzGetLength{rAC}
  \tkzDrawCircle(A,C)
  \tkzDrawCircle(A,B)
  \tkzDefRandPointOn[rectangle=A and B]
  \tkzGetPoint{a}
  \tkzDefRandPointOn[segment=A--B]
  \tkzGetPoint{b}
  \tkzDefRandPointOn[circle=center A radius \rAC pt]
  \tkzGetPoint{d}
  \tkzDefRandPointOn[circle through= center A through B]
  \tkzGetPoint{c}
  \tkzDefRandPointOn[disk through=center A through B]
  \tkzGetPoint{e}
  \tkzLabelPoints[above right=3pt](A,B,C,a,b,...,e)
  \tkzDrawPoints[] (A,B,C,a,b,...,e)
  \tkzDrawRectangle(A,B)
\end{tikzpicture}
```

11.5 圆上的随机点



```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzGrid
  \tkzDefPoints{3/2/A,1/1/B}
  \tkzCalcLength[cm](A,B) \tkzGetLength{rAB}
  \tkzDrawCircle[R](A,\rAB cm)
  \tkzDefRandPointOn[circle = center A radius
    \rAB cm]\tkzGetPoint{a}
  \tkzDrawSegment(A,a)
  \tkzDrawPoints(A,B,a)
  \tkzLabelPoints(A,B,a)
\end{tikzpicture}
```

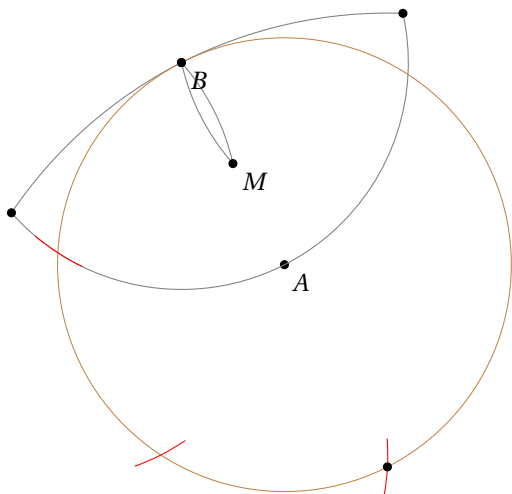
11.5.1 Apollonius 圆的随机示例



```
\begin{tikzpicture}[scale=1.25]
  \tkzDefPoints{0/0/A,3/0/B}
  \def\coeffK{2}
  \tkzApolloniusCenter[K=\coeffK](A,B)
  \tkzGetPoint{P}
  \tkzDefApolloniusPoint[K=\coeffK](A,B)
  \tkzGetPoint{M}
  \tkzDefApolloniusRadius[K=\coeffK](A,B)
  \tkzDrawCircle[R,color = blue!50!black, fill=blue!20,
    opacity=.4](\tkzPointResult,\tkzLengthResult pt)
  \tkzDefRandPointOn[circle through= center P through M]
  \tkzGetPoint{N}
  \tkzDrawPoints(A,B,P,M,N)
  \tkzLabelPoints(A,B,P,M,N)
  \tkzDrawSegments[red](N,A N,B)
  \tkzDrawPoints(A,B)
  \tkzDrawSegments[red](A,B)
  \tkzLabelCircle[R,draw,fill=green!10, text width=3cm,
    text centered](P,\tkzLengthResult pt-20pt)(-120)
    {$MA/MB=\coeffK$\$NA/NB=\coeffK$}
\end{tikzpicture}
```

11.6 罗盘线段中点

作为小结，这是较为复杂例子，它用罗盘来确定线段的中点。



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A}
  \tkzDefRandPointOn[circle= center A radius 4cm]
  \tkzGetPoint{B}
  \tkzDrawPoints(A,B)
  \tkzDefPointBy[rotation= center A angle 180](B)
  \tkzGetPoint{C}
  \tkzInterCC[R](A,4 cm)(B,4 cm)
  \tkzGetPoints{I}{I'}
  \tkzInterCC[R](A,4 cm)(I,4 cm)
  \tkzGetPoints{J}{B}
  \tkzInterCC(B,A)(C,B)
  \tkzGetPoints{D}{E}
  \tkzInterCC(D,B)(E,B)
  \tkzGetPoints{M}{M'}
  \tikzset{arc/.style={color=brown,style=dashed,delta=10}}
  \tkzDrawArc[arc](C,D)(E)
  \tkzDrawArc[arc](B,E)(D)
  \tkzDrawCircle[color=brown,line width=.2pt](A,B)
  \tkzDrawArc[arc](D,B)(M)
  \tkzDrawArc[arc](E,M)(B)
  \tkzCompassss[color=red,style=solid](B,I I,J J,C)
  \tkzDrawPoints(B,C,D,E,M)
  \tkzLabelPoints(A,B,M)
\end{tikzpicture}
```

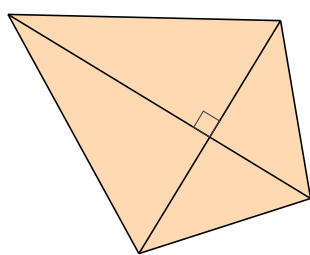
12 直线

在绘制直线之前，需要确定直线上的两个点，也可以通过垂直平分线、角平分线、平行线或垂线等方式确定特定的直线。

12.1 定义直线

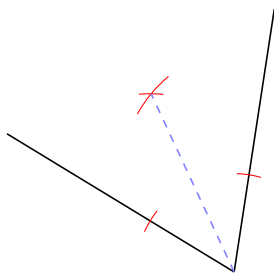
\tkzDefLine[< 命令选项>](<pt1,pt2>) or (<pt1,pt2,pt3>)		
参数是 2 个或 3 个点的列表,根据具体问题,该命令得到 1 个或 2 个点,这些结果需要使用\tkzGetPoint或\tkzGetPoints命令获得并对结果点进行命名。		
参数	样例	说明
(<pt1,pt2>)	(<A,B>)	[mediator] (A,B)
(<pt1,pt2,pt3>)	(<A,B,C>)	[bisector] (B,A,C)
选项	默认值	含义
mediator		定义 2 个点
perpendicular=through...	mediator	通过指定点的垂线
orthogonal=through...	mediator	同上
parallel=through...	mediator	通过指定点的平行线
bisector	mediator	通过 3 个点定义的角的角平分线
bisector out	mediator	外角角平分线
K	1	垂线的比例系数
normed	false	线段规一化

12.1.1 mediator选项



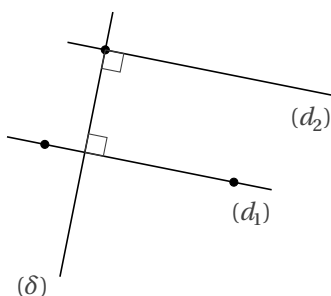
```
\begin{tikzpicture}[rotate=25]
\tkzDefPoints{-2/0/A,1/2/B}
\tkzDefLine[mediator](A,B)\tkzGetPoints{C}{D}
\tkzDefPointWith[linear,K=.75](C,D)\tkzGetPoint{D}
\tkzDefMidPoint(A,B)\tkzGetPoint{I}
\tkzFillPolygon[color=orange!30](A,C,B,D)
\tkzDrawSegments(A,B C,D)
\tkzMarkRightAngle(B,I,C)
\tkzDrawSegments(D,B D,A)
\tkzDrawSegments(C,B C,A)
\end{tikzpicture}
```

12.1.2 bisector和normed选项



```
\begin{tikzpicture}[rotate=25,scale=.65]
\tkzDefPoints{0/0/C, 2/-3/A, 4/0/B}
\tkzDefLine[bisector,normed](B,A,C)
\tkzGetPoint{a}
\tkzDrawLines[add= 0 and .5](A,B A,C)
\tkzShowLine[bisector,gap=4,size=2,color=red](B,A,C)
\tkzDrawLines[blue!50,dashed,add= 0 and 3](A,a)
\end{tikzpicture}
```

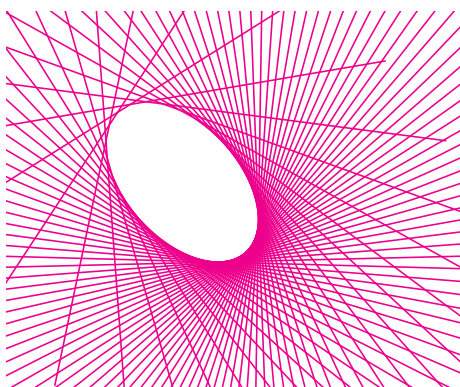
12.1.3 orthogonal和parallel选项



```
\begin{tikzpicture}
\tkzDefPoints{-1.5/-0.25/A,1/-0.75/B,-0.7/1/C}
\tkzDrawLine(A,B)
\tkzLabelLine[pos=1.25,below left](A,B){$(d_1)$}
\tkzDrawPoints(A,B,C)
\tkzDefLine[orthogonal=through C](B,A) \tkzGetPoint{c}
\tkzDrawLine(C,c)
\tkzLabelLine[pos=1.25,left](C,c){$(\delta)$}
\tkzInterLL(A,B)(C,c) \tkzGetPoint{I}
\tkzMarkRightAngle(C,I,B)
\tkzDefLine[parallel=through C](A,B) \tkzGetPoint{c'}
\tkzDrawLine(C,c')
\tkzLabelLine[pos=1.25,below left](C,c'){$(d_2)$}
\tkzMarkRightAngle(I,C,c')
\end{tikzpicture}
```

12.1.4 循环画图

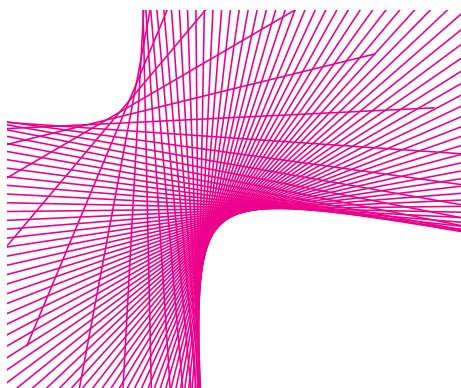
基于 D Rodriguez 用 pst-eucl 宏包绘制的 O. Reboux 设计的图案进行绘制。



```
\begin{tikzpicture}[scale=.5]
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6] % necessary
\tkzClip
\tkzDefPoint(0,0){O}
\tkzDefPoint(132:4){A}
\tkzDefPoint(5,0){B}
\foreach \ang in {5,10,...,360}{%
\tkzDefPoint(\ang:5){M}
\tkzDefLine[mediator](A,M)
\tkzDrawLine[color=magenta,add= 3 and
3](tkzFirstPointResult,tkzSecondPointResult)}
\end{tikzpicture}
```

12.1.5 抛物线

基于 D Rodriguez 用 pst-eucl 宏包绘制的 O. Reboux 设计的图案进行绘制。本例中，对定义的垂直平分线的两个端点进行命名，是有必要的。



```
\begin{tikzpicture}[scale=.5]
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6]
\tkzClip
\tkzDefPoint(0,0){O}
\tkzDefPoint(132:5){A}
\tkzDefPoint(4,0){B}
\foreach \ang in {5,10,...,360}{%
\tkzDefPoint(\ang:4){M}
\tkzDefLine[mediator](A,M)
\tkzDrawLine[color=magenta,add= 3 and
3](tkzFirstPointResult,tkzSecondPointResult)}
\end{tikzpicture}
```

12.2 圆的切线

圆的切线可以有两种，一是在圆上某点处的切线，另一种是过圆外某点的圆的切线。

`\tkzDefTangent[< 命令选项>](<pt1,pt2>) or (<pt1,dim>)`

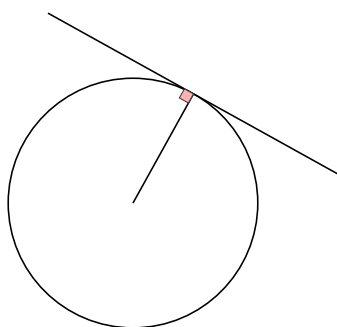
圆括号中的参数是圆心、圆心和圆上的一个点或是圆心和半径，该命令取代了 `\tkzTangent` 命令。

参数	样例	说明
<code>(<pt1,pt2 or (<pt1,dim>))</code>	<code>(<A,B>) or (<A,2cm>)</code>	<code>[AB]</code> 是半径, <code>A</code> 是圆心

选项	默认值	含义
<code>at=pt</code>	<code>at</code>	圆上指定点的切线
<code>from=pt</code>	<code>at</code>	通过圆外指定点的圆的切线
<code>from with R=pt</code>	<code>at</code>	同上，但圆需要通过圆心和半径指定

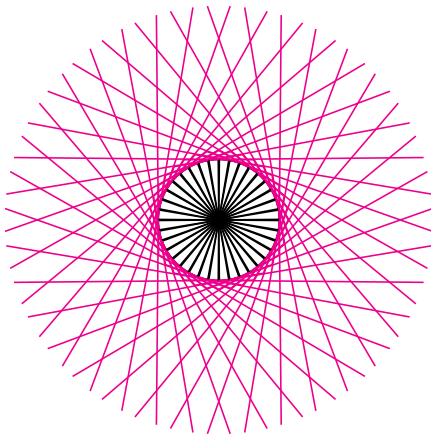
注意，该命令仅定义切线，并不绘制切线。切线的第 2 个点可通过 `tkzPointResult` 命令得到。

12.2.1 圆上指定点的切线



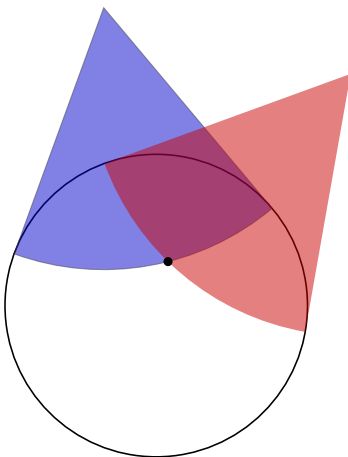
```
\begin{tikzpicture}[scale=.55]
\tkzDefPoint(0,0){O} \tkzDefPoint(6,6){E}
\tkzDefRandPointOn[circle=center O radius 3cm]
\tkzGetPoint{A} \tkzDrawSegment(O,A)
\tkzDrawCircle(O,A)
% 重围包围盒 (删除圆外的空白)
\pgfresetboundingbox
\tkzDefTangent[at=A](O)
\tkzGetPoint{h}
\tkzDrawLine[add = 4 and 3](A,h)
\tkzMarkRightAngle[fill=red!30](O,A,h)
\end{tikzpicture}
```

12.2.2 过圆外指定点的切线



```
\begin{tikzpicture}[scale=.8]
  \tkzDefPoint(3,3){c}
  \tkzDefPoint(6,3){a0}
  \tkzRadius=1 cm
  \tkzDrawCircle[R](c,\tkzRadius)
  \foreach \an in {0,10,...,350}{
    \tkzDefPointBy[rotation=center c angle \an](a0)
    \tkzGetPoint{a}
    \tkzDefTangent[from with R = a](c,\tkzRadius)
    \tkzGetPoints{e}{f}
    \tkzDrawLines[color=magenta](a,f a,e)
    \tkzDrawSegments(c,e c,f)
  }%
\end{tikzpicture}
```

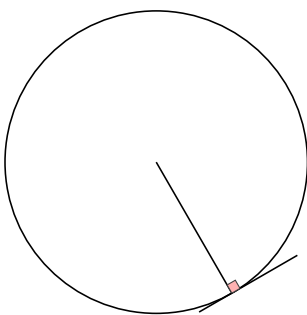
12.2.3 Andrew Mertz 示例



```
\begin{tikzpicture}[scale=0.5]
  \tkzDefPoint(100:8){A}
  \tkzDefPoint(50:8){B}
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(0,4){R}
  \tkzDrawCircle(C,R)
  \tkzDefTangent[from = A](C,R)
  \tkzGetPoints{D}{E}
  \tkzDefTangent[from = B](C,R)
  \tkzGetPoints{F}{G}
  \tkzDrawSector[fill=blue!80!black,opacity=0.5](A,D)(E)
  \tkzFillSector[color=red!80!black,opacity=0.5](B,F)(G)
  \tkzInterCC(A,D)(B,F) \tkzGetSecondPoint{I}
  \tkzDrawPoint[color=black](I)
\end{tikzpicture}
```

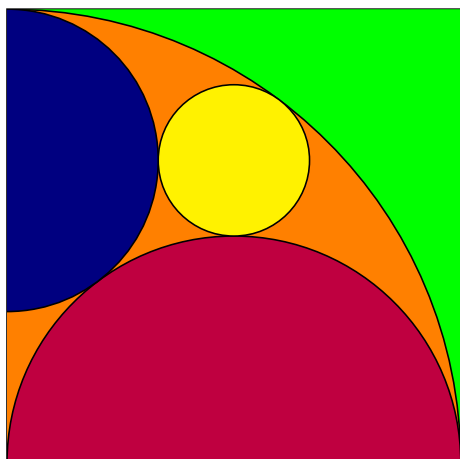
<http://www.texample.net/tikz/examples/>

12.2.4 from with R 和 at 选项



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoint(0,0){O}
  \tkzDefRandPointOn[circle=center O radius 4cm]
  \tkzGetPoint{A}
  \tkzDefTangent[at=A](O)
  \tkzGetPoint{h}
  \tkzDrawSegments(O,A)
  \tkzDrawCircle(O,A)
  \tkzDrawLine[add = 1 and 1](A,h)
  \tkzMarkRightAngle[fill=red!30](O,A,h)
\end{tikzpicture}
```


12.2.5 from 选项



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){B}
  \tkzDefPoint(0,8){A}
  \tkzDefSquare(A,B)
  \tkzGetPoints{C}{D}
  \tkzDrawSquare(A,B)
  \tkzClipPolygon(A,B,C,D)
  \tkzDefPoint(4,8){F}
  \tkzDefPoint(4,0){E}
  \tkzDefPoint(4,4){Q}
  \tkzFillPolygon[color = green](A,B,C,D)
  \tkzDrawCircle[fill = orange](B,A)
  \tkzDrawCircle[fill = purple](E,B)
  \tkzDefTangent[from=B](F,A)
  \tkzInterLL(F,t kzFirstPointResult)(C,D)
  \tkzInterLL(A,t kzPointResult)(F,E)
  \tkzDrawCircle[fill = yellow](t kzPointResult,Q)
  \tkzDefPointBy[projection= onto B--A](t kzPointResult)
  \tkzDrawCircle[fill = blue!50!black](t kzPointResult,A)
\end{tikzpicture}
```

13 绘制和命名直线

下列命令用于绘制和命名直线

13.1 绘制直线

只需要给出一条直线上的一对点，就可以绘制一条直线。在绘制直线时，可以使用`add`参数设置端点处向外扩展的距离(该选项由 **Mark Wibrow** 设计，具体代码如下)。

```
\tikzset{%
  add/.style args={#1 and #2}{
    to path={%
      ($(\tikztostart)!-#1!(\tikztotarget)$)--$(\tikztotarget)!-#2!(\tikztostart)$)%
    \tikztonodes}}}
```

如果直线由三角形定义，则参数是3个点的列表(三角形的3个顶点)。其中，第2个点是直线的起点，第1个和第2个点是直线终点。因此，旧的命令`\tkzDrawMedian`中的 $(A,B)(C)$ 应该改为 (B,C,A) 。

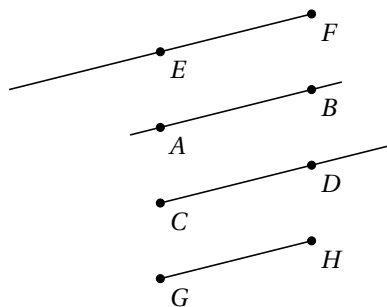
`\tkzDrawLine[< 命令选项>]($\langle pt1,pt2 \rangle$) or ($\langle pt1,pt2,pt3 \rangle$)`

参数是2个或3个点。

选项	默认值	含义
median	无	[median] (A,B,C) 基点 B 处的中线
altitude	无	[altitude] (C,A,B) 顶点 A 的高
bisector	无	[bisector] (B,C,A) 顶点 C 的角平分线
none	无	绘制直线 (AB)
add= nb1 and nb2	.2 and .2	扩展线段

`add`定义了通过点 $pt1$ 和 $pt2$ 的扩展长度，两个数字均为百分比。也可以在绘图中使用 TikZ 样式。

13.1.1 add选项



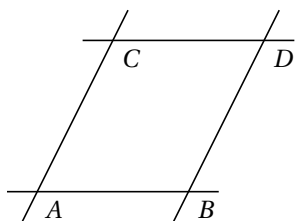
```
\begin{tikzpicture}
\tkzInit[xmin=-2,xmax=3,ymin=-2.25,ymax=2.25]
\tkzClip[space=.25]
\tkzDefPoint(0,0){A} \tkzDefPoint(2,0.5){B}
\tkzDefPoint(0,-1){C} \tkzDefPoint(2,-0.5){D}
\tkzDefPoint(0,1){E} \tkzDefPoint(2,1.5){F}
\tkzDefPoint(0,-2){G} \tkzDefPoint(2,-1.5){H}
\tkzDrawLine(A,B)
\tkzDrawLine[add = 0 and .5](C,D)
\tkzDrawLine[add = 1 and 0](E,F)
\tkzDrawLine[add = 0 and 0](G,H)
\tkzDrawPoints(A,B,C,D,E,F,G,H)
\tkzLabelPoints(A,B,C,D,E,F,G,H)
\end{tikzpicture}
```

也可以使用相同的选项绘制多条线。

```
\tkzDrawLines[< 命令选项>](\pt1,pt2 \pt3,pt4,...)
```

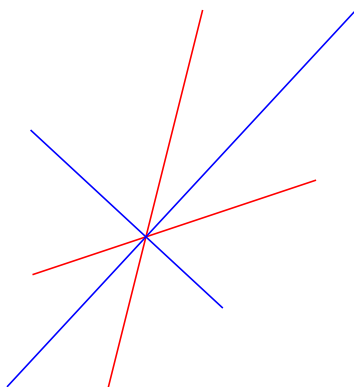
参数是用空格分隔的直线点对，点对之间用逗号分隔。也可以在绘制中使用 TikZ 样式。

13.1.2 \tkzDrawLines选项



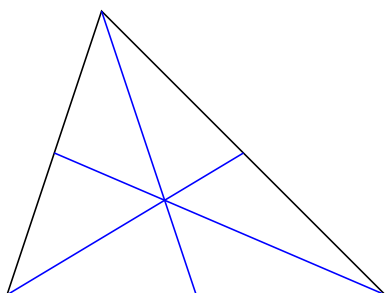
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,0){B}
\tkzDefPoint(1,2){C}
\tkzDefPoint(3,2){D}
\tkzDrawLines(A,B C,D A,C B,D)
\tkzLabelPoints(A,B,C,D)
\end{tikzpicture}
```

13.1.3 add选项



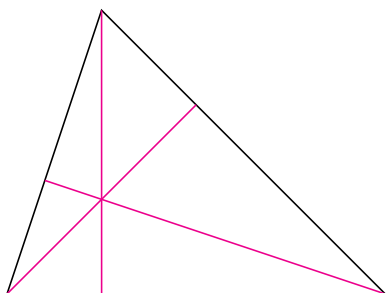
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,0){O}
\tkzDefPoint(3,1){I}
\tkzDefPoint(1,4){J}
\tkzDefLine[bisector](I,O,J)
\tkzGetPoint{i}
\tkzDefLine[bisector out](I,O,J)
\tkzGetPoint{j}
\tkzDrawLines[add = 1 and .5,color=red](O,I O,J)
\tkzDrawLines[add = 1 and .5,color=blue](O,i O,j)
\end{tikzpicture}
```

13.1.4 三角形的中线



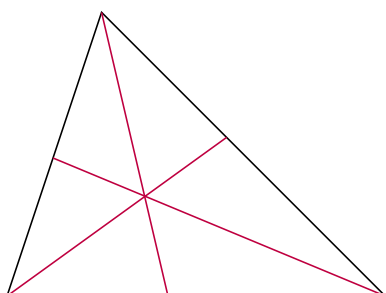
```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(1,3){C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpLine[color=blue]
\tkzDrawLine[median](B,C,A)
\tkzDrawLine[median](C,A,B)
\tkzDrawLine[median](A,B,C)
\end{tikzpicture}
```

13.1.5 三角形的高



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(1,3){C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpLine[color=magenta]
\tkzDrawLine[altitude](B,C,A)
\tkzDrawLine[altitude](C,A,B)
\tkzDrawLine[altitude](A,B,C)
\end{tikzpicture}
```

13.1.6 三角形的角平分线



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(1,3){C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpLine[color=purple]
\tkzDrawLine[bisector](B,C,A)
\tkzDrawLine[bisector](C,A,B)
\tkzDrawLine[bisector](A,B,C)
\end{tikzpicture}
```

13.2 用\tkzLabelLine命令为直线添加标注

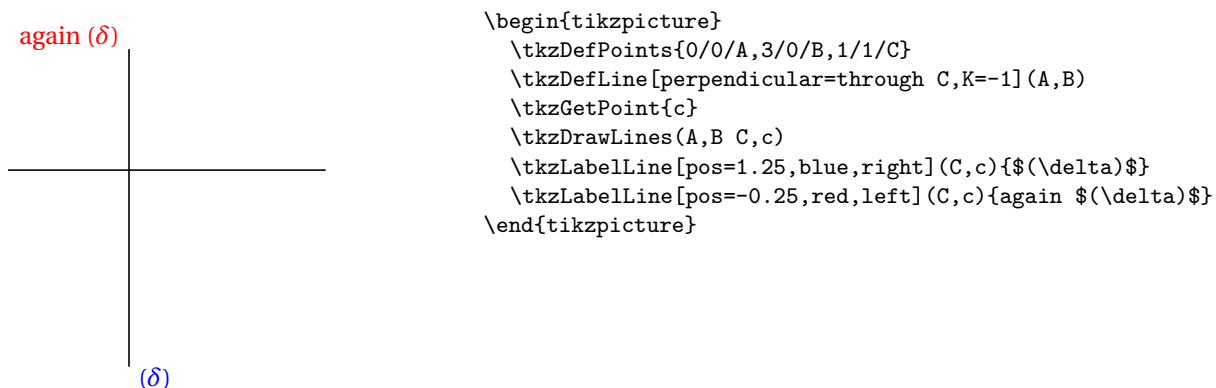
```
\tkzLabelLine[< 命令选项>](\pt1,\pt2){\label}
```

参数	默认值	含义
label	\tkzLabelLine(A,B){ Δ }	
选项	默认值	含义
pos	.5	pos是 TikZ 的一个选项

在选项中，除了可以使用pos外，可以使用任何 TikZ 样式，特别是用于设置标注位置的above、right、...等样式。

13.2.1 \tkzLabelLine 示例代码

`pos` 是一个重要的选项，该选项的取值可以是大于 1，也可以是负值。



14 绘制和标记线段

14.1 用 \tkzDrawSegment 命令绘制线段

`\tkzDrawSegment[< 命令选项>](<pt1,pt2>)`

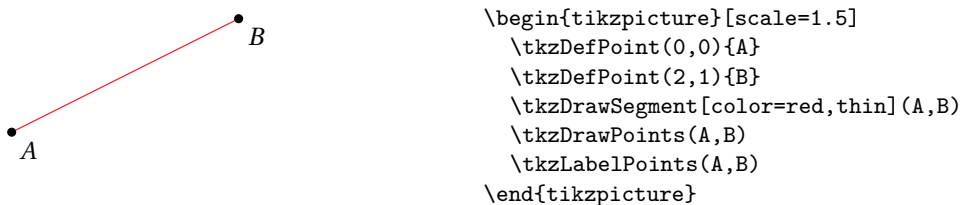
参数是用逗号分隔的线段端点列表，可以使用任何合法的 TikZ 样式。

参数	样例	含义
(pt1,pt2)	(A,B)	绘制线段 [A,B]

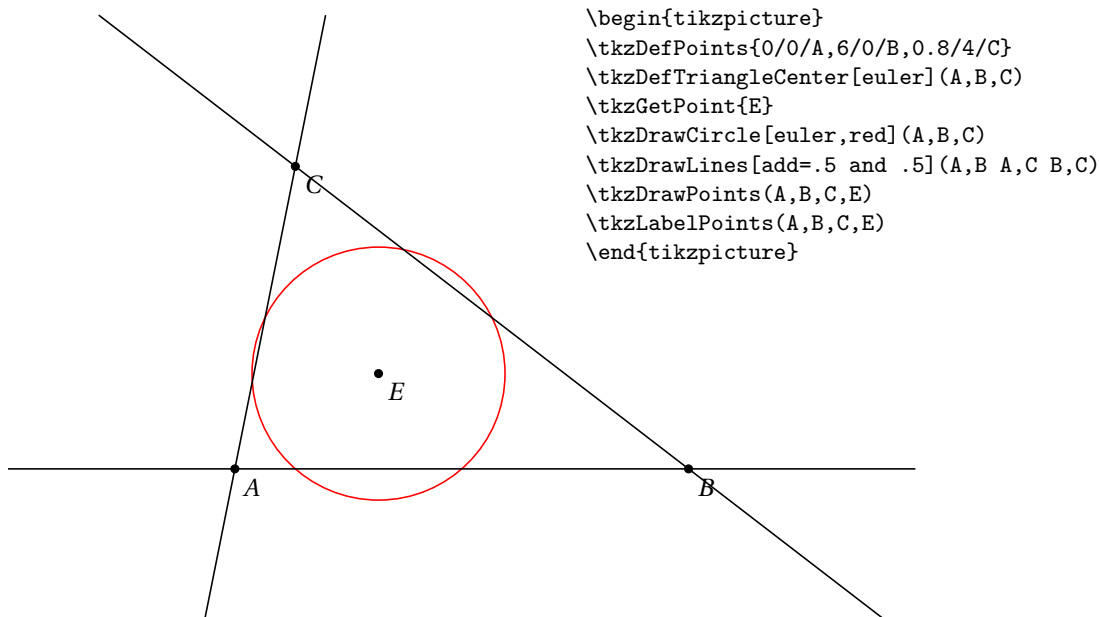
选项	样例	含义
TikZ 选项		所有 TikZ 有效选项
add	0 and 0	add = kl and kr , ...
...	...	允许线段向左右扩展
dim	无	dim = {label,dim,option}, ...
...	...	允许为图形添加尺寸

该命令等效于 `\draw (A)--(B);` 命令

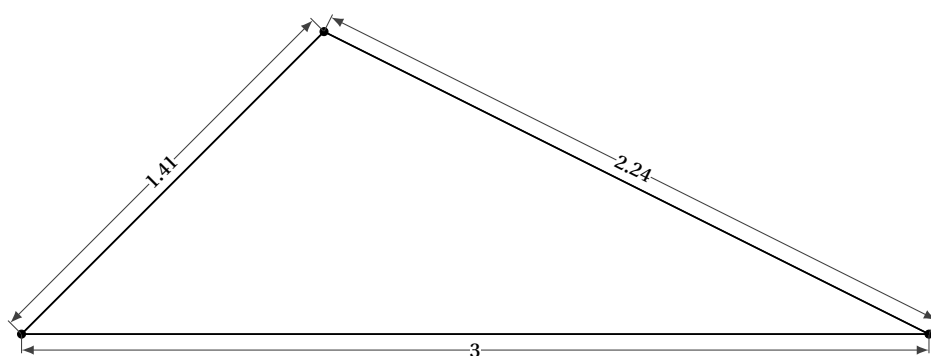
14.1.1 简单示例



14.1.2 使用add选项扩展线段两端



14.1.3 使用dim选项标注尺寸



```

\begin{tikzpicture}[scale=4]
\pgfkeys{/pgf/number format/.cd, fixed, precision=2}
% 定义已知点
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,0){B}
\tkzDefPoint(1,1){C}
% 绘制三角形和点
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
% 计算长度
\tkzCalcLength[cm](A,B)\tkzGetLength{AB1}
\tkzCalcLength[cm](B,C)\tkzGetLength{BC1}
\tkzCalcLength[cm](A,C)\tkzGetLength{AC1}
% 添加尺寸标注
\tkzDrawSegment[dim={\pgfmathprintnumber\BC1,6pt,transform shape}](C,B)
\tkzDrawSegment[dim={\pgfmathprintnumber\AC1,6pt,transform shape}](A,C)
\tkzDrawSegment[dim={\pgfmathprintnumber\AB1,-6pt,transform shape}](A,B)
\end{tikzpicture}

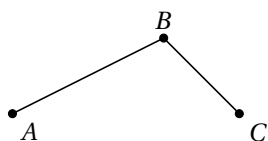
```

14.2 用\tkzDrawSegments命令绘制多条线段

可以用该命令绘制选项相同的多条线段

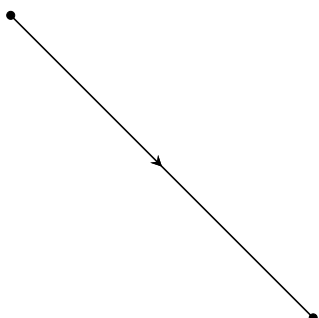
\tkzDrawSegments[< 命令选项>](<pt1,pt2 pt3,pt4,...>)

参数是一组用空格分隔的线段的端点列表，两个端点之间用逗号分隔。在绘制中可以使用任何有效的 TikZ 样式。



```
\begin{tikzpicture}
\tkzInit[xmin=-1,xmax=3,ymin=-1,ymax=2]
\tkzClip[space=1]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,1){B}
\tkzDefPoint(3,0){C}
\tkzDrawSegments(A,B B,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(A,C)
\tkzLabelPoints[above](B)
\end{tikzpicture}
```

14.2.1 为线段添加箭头



```
\begin{tikzpicture}
\tikzset{
  arr/.style={postaction=decorate,
  decoration={markings,
  mark=at position .5 with {\arrow[thick]{#1}}}
}
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,-4){B}
\tkzDrawSegments[arr=stealth](A,B)
\tkzDrawPoints(A,B)
\end{tikzpicture}
```

14.3 用\tkzMarkSegment命令标记线段

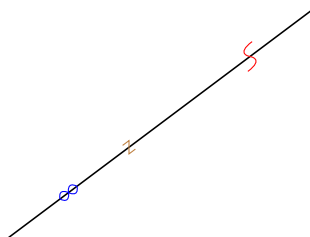
\tkzMarkSegment[< 命令选项>](<pt1,pt2>)

该命令用于为线段添加标记。

选项	默认值	含义
pos	.5	标记的位置
color	black	标记的颜色
mark	none	标记类型
size	4pt	标记尺寸

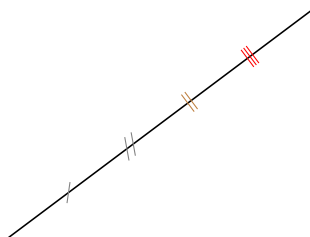
标记样式由 TikZ 提供，但也可以使用基于 Yves Combe 的方法自定义标记样式。

14.3.1 几种标记



```
\begin{tikzpicture}
\tkzDefPoint(2,1){A}
\tkzDefPoint(6,4){B}
\tkzDrawSegment(A,B)
\tkzMarkSegment[color=brown,size=2pt,pos=0.4, mark=z](A,B)
\tkzMarkSegment[color=blue,pos=0.2, mark=oo](A,B)
\tkzMarkSegment[pos=0.8,mark=s,color=red](A,B)
\end{tikzpicture}
```

14.3.2 mark选项



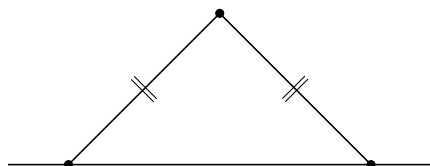
```
\begin{tikzpicture}
\tkzDefPoint(2,1){A}
\tkzDefPoint(6,4){B}
\tkzDrawSegment(A,B)
\tkzMarkSegment[color=gray,pos=0.2,mark=s](A,B)
\tkzMarkSegment[color=gray,pos=0.4,mark=s||](A,B)
\tkzMarkSegment[color=brown,pos=0.6,mark=s||](A,B)
\tkzMarkSegment[color=red,pos=0.8,mark=s|||](A,B)
\end{tikzpicture}
```

14.4 用\tkzMarkSegments标记多条线段

```
\tkzMarkSegments[< 命令选项>](<pt1,pt2 pt3,pt4,...>)
```

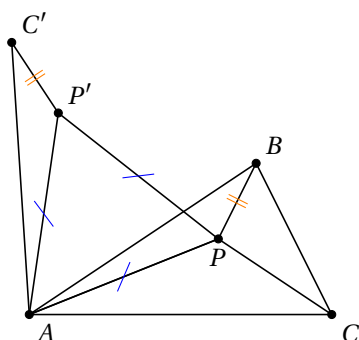
参数是用空格分隔的线段端点列表，每对端点用逗号分隔。可以使用任何有效的 TikZ 样式。

14.4.1 标记等腰三角形



```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{0/0/O,2/2/A,4/0/B,6/2/C}
\tkzDrawSegments(O,A A,B)
\tkzDrawPoints(O,A,B)
\tkzDrawLine(O,B)
\tkzMarkSegments[mark=||,size=6pt](O,A A,B)
\end{tikzpicture}
```

14.4.2 其它标记



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){A}\tkzDefPoint(3,2){B}
  \tkzDefPoint(4,0){C}\tkzDefPoint(2.5,1){P}
  \tkzDrawPolygon(A,B,C)
  \tkzDefEquilateral(A,P) \tkzGetPoint{P'}
  \tkzDrawPolygon(A,P,P')
  \tkzDrawPolySeg(P',C',A,P,B)
  \tkzDrawSegment(C,P)
  \tkzDrawPoints(A,B,C,C',P,P')
  \tkzMarkSegments[mark=s|,size=6pt,
    color=blue](A,P P,P' P',A)
  \tkzMarkSegments[mark=||,color=orange](B,P P',C')
  \tkzLabelPoints(A,C) \tkzLabelPoints[below](P)
  \tkzLabelPoints[above right](P',C',B)
\end{tikzpicture}
```

14.5 线段标注命令\tkzLabelSegment

`\tkzLabelSegment[< 命令选项>](pt1,pt2){label}`

该命令用于为一个线段或直线添加标注，可以使用类似`pos`的任何有效 TikZ 样式。

参数	样例	含义
label (pt1,pt2)	<code>\tkzLabelSegment(A,B){5}</code> (A,B)	标注文本 被标注线段 [AB]

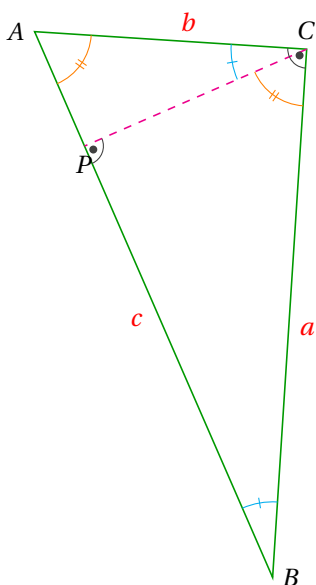
选项	默认值	含义
pos	.5	标注的位置

14.5.1 多个标注



```
\begin{tikzpicture}
  \tkzInit
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(6,0){B}
  \tkzDrawSegment(A,B)
  \tkzLabelSegment[above,pos=.8](A,B){$a$}
  \tkzLabelSegment[below,pos=.2](A,B){$4$}
\end{tikzpicture}
```


14.5.2 直角三角形的标注和标记



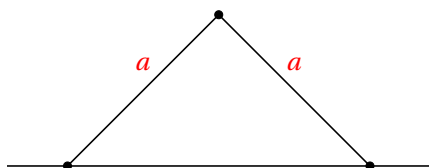
```
\begin{tikzpicture}[rotate=-60]
  \tikzset{label seg style/.append style = {%
    color      = red,
  }}
  \tkzDefPoint(0,1){A}
  \tkzDefPoint(2,4){C}
  \tkzDefPointWith[orthogonal normed,K=7](C,A)
  \tkzGetPoint{B}
  \tkzDrawPolygon[green!60!black](A,B,C)
  \tkzDrawLine[altitude,dashed,color=magenta](B,C,A)
  \tkzGetPoint{P}
  \tkzLabelPoint[left](A){$A$}
  \tkzLabelPoint[right](B){$B$}
  \tkzLabelPoint[above](C){$C$}
  \tkzLabelPoint[below](P){$P$}
  \tkzLabelSegment[] (B,A){$c$}
  \tkzLabelSegment[swap] (B,C){$a$}
  \tkzLabelSegment[swap] (C,A){$b$}
  \tkzMarkAngles[size=1cm,
    color=cyan,mark=|](C,B,A A,C,P)
  \tkzMarkAngle[size=0.75cm,
    color=orange,mark=||](P,C,B)
  \tkzMarkAngle[size=0.75cm,
    color=orange,mark=||](B,A,C)
  \tkzMarkRightAngles[german](A,C,B B,P,C)
\end{tikzpicture}
```

14.6 多条线段标注命令\tkzLabelSegments

```
\tkzLabelSegments[< 命令选项>](<pt1,pt2 pt3,pt4,...>)
```

如果多条线段的标注样式相同，则可以使用该命令一次性进行标注。

14.6.1 等腰三角形的标注



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{0/0/0,2/2/A,4/0/B,6/2/C}
  \tkzDrawSegments(0,A A,B)
  \tkzDrawPoints(0,A,B)
  \tkzDrawLine(0,B)
  \tkzLabelSegments[color=red,above=4pt](0,A A,B){$a$}
\end{tikzpicture}
```

15 三角形

15.1 定义三角形命令`\tkzDefTriangle`

定义三角形的命令允许使用至少2个点构建一个三角形。可以按如下方式定义三角形：

- **two angles** 已知2个角的三角形;
- **equilateral** 等边三角形;
- **half** 直角边之和与斜边之比等于2的直角三角形;
- **pythagore** 满足如3、4和5的勾股直角三角形;
- **school** 三个角分别是30、60和90的直角三角形;
- **golden** 直角边之和与斜边比等于 $\Phi = 1.618034$ 的直角三角形，“golden triangle”名称来自“golden rectangle”，底角为72度的等腰三角形称为“golden triangle”或“Euclid's triangle”;
- **euclide or gold** 同上;
- **cheops** 通过三边比例为2、 Φ 和 Φ 确定三角形的第3个点。

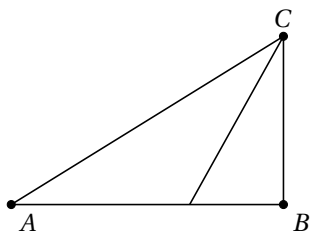
`\tkzDefTriangle[< 命令选项>](A,B)`

由于三角形按照三角圆的方式定义，因此，参数中的点的顺序决定的另外一个点的位置。使用该命令后，可以使用`\tkzGetPoint`得到计算结果并为点命名，也可以使用`tkzPointResult`得到计算结果，但不为该点命名。

选项	默认值	含义
<code>two angles= #1 and #2</code>	无	三角形两个已知角度
<code>equilateral</code>	无	等边三角形
<code>pythagore</code>	无	勾股三角形，三边满足类似 3-4-5 的关系
<code>school</code>	无	三个角分别是 30、60 和 90 度
<code>gold</code>	无	三个角分别是 72、72 和 36 度，A 是顶点
<code>euclide</code>	无	同上，但 [AB] 是底边
<code>golden</code>	无	AB 构成矩形，并且 $AB/AC = \Phi$
<code>cheops</code>	无	$AC = BC$ ，AC 和 BC 及第三边满足 2 和 Φ 的比例关系

使用`\tkzGetPoint`可以保存并命名得到的点，也可以使用`tkzPointResult`命令临时使用得到的点。

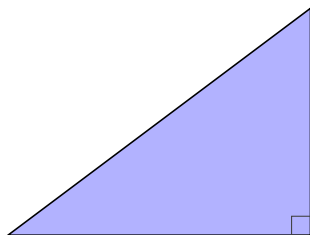
15.1.1 golden选项



```
\begin{tikzpicture}[scale=0.9]
\tkzInit[xmax=5,ymax=3]
\tkzClip[space=.5]
\tkzDefPoint(0,0){A} \tkzDefPoint(4,0){B}
\tkzDefTriangle[golden](A,B)\tkzGetPoint{C}
\tkzDrawPolygon(A,B,C) \tkzDrawPoints(A,B,C)
\tkzLabelPoints(A,B) \tkzDrawBisector(A,C,B)
\tkzLabelPoints[above](C)
\end{tikzpicture}
```


15.2.1 pythagore选项

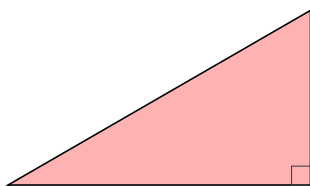
三角形的三个边满足类似3、4和5的关系。



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDrawTriangle[pythagore,fill=blue!30](A,B)
\tkzMarkRightAngles(A,B,t kzPointResult)
\end{tikzpicture}
```

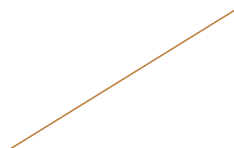
15.2.2 school选项

三角形的三个内角分别是30、60和90度。



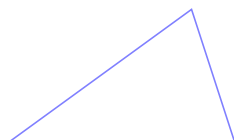
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDrawTriangle[school,fill=red!30](A,B)
\tkzMarkRightAngles(t kzPointResult,B,A)
\end{tikzpicture}
```

15.2.3 golden选项



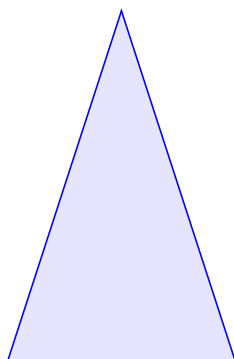
```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,-10){M}
\tkzDefPoint(3,-10){N}
\tkzDrawTriangle[golden,color=brown](M,N)
\end{tikzpicture}
```

15.2.4 gold选项



```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(5,-5){I}
\tkzDefPoint(8,-5){J}
\tkzDrawTriangle[gold,color=blue!50](I,J)
\end{tikzpicture}
```

15.2.5 euclide选项



```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(10,-5){K}
\tkzDefPoint(13,-5){L}
\tkzDrawTriangle[euclide,color=blue,fill=blue!10](K,L)
\end{tikzpicture}
```

16 使用\tkzDefSpcTriangle命令定义特殊三角形

在“点”的定义小节中，定义了一些三角形中的特殊点，在此，可以使用这些点确定三角形的三个顶点。

`\tkzDefSpcTriangle[< 命令选项>](A,B,C)`

注意，点的顺序决定了计算结果。

选项	默认值	含义
in or incentral	centroid	已知两角的三角形
ex or excentral	centroid	等边三角形
extouch	centroid	勾股三角形
intouch or contact	centroid	30、60 和 90 度三角形
centroid or medial	centroid	底角是 72，顶点是 A 的等腰三角形
orthic	centroid	同上，但底边是 [AB]
feuerbach	centroid	$AB/AC = \Phi$
euler	centroid	AC=BC, AC 和 BC 与第 3 边满足 2 和 Φ 的关系
tangential	centroid	同上
name	无	同上

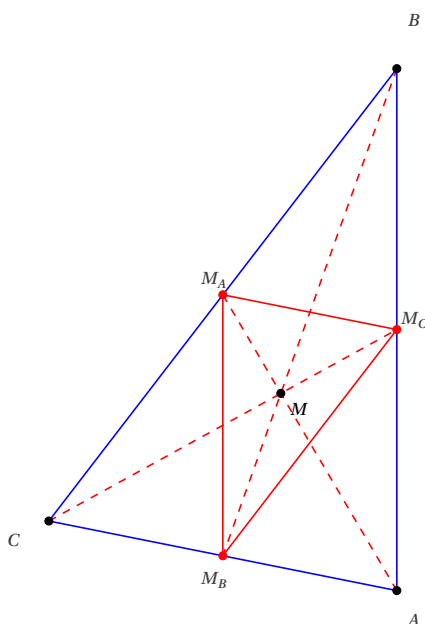
使用\tkzGetPoint可以保存并命名得到的点，也可以使用tkzPointResult命令临时使用得到的点。

16.0.1 medial或centroid选项

三角形的质心用 G 表示 (有时也用 M 表示)，它是三角形三条中线的将点，该点也称为重心，重心总是位于三角形内部。

Weisstein, Eric W. “Centroid triangle” From MathWorld—A Wolfram Web Resource.

下面的例子中，通过预先定义的点，得到通过这些点的欧拉圆。

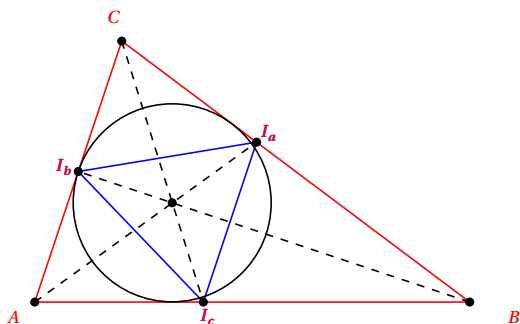


```
\begin{tikzpicture}[rotate=90,scale=1.15]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefTriangleCenter[centroid](A,B,C)
\tkzGetPoint{M}
\tkzDefSpcTriangle[medial,name=M](A,B,C){_A,_B,_C}
\tkzDrawPolygon[color=blue](A,B,C)
\tkzDrawSegments[dashed,red](A,M_A B,M_B C,M_C)
\tkzDrawPolygon[color=red](M_A,M_B,M_C)
\tkzDrawPoints(A,B,C,M)
\tkzDrawPoints[red](M_A,M_B,M_C)
\tkzAutoLabelPoints[center=M,font=\scriptsize]%(A,B,C,M_A,M_B,M_C)
\tkzLabelPoints[font=\scriptsize](M)
\end{tikzpicture}
```

16.0.2 in或incentral选项

中心三角形是由另一个三角形的三个内角平分线与对边交点确定的三角形。

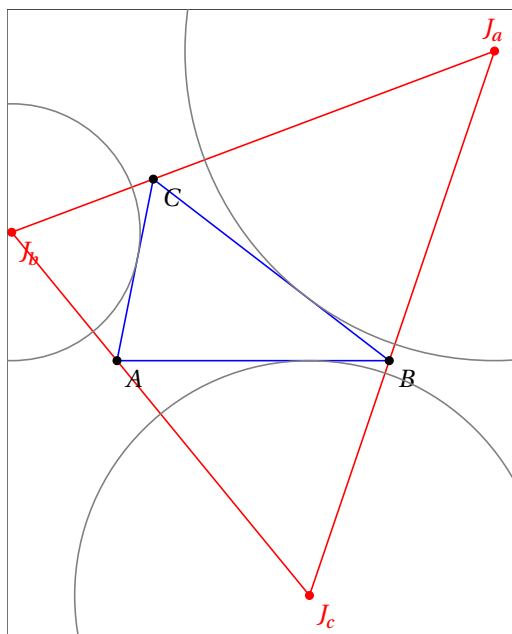
Weissstein, Eric W. “Incentral triangle” From MathWorld—A Wolfram Web Resource.



```
\begin{tikzpicture}[scale=1.15]
\tkzDefPoints{ 0/0/A,5/0/B,1/3/C}
\tkzDefSpcTriangle[in,name=I](A,B,C){_a,_b,_c}
\tkzInCenter(A,B,C)
\tkzGetPoint{I}
\tkzDrawPolygon[red](A,B,C)
\tkzDrawPolygon[blue](I_a,I_b,I_c)
\tkzDrawPoints(A,B,C,I,I_a,I_b,I_c)
\tkzDrawCircle[in](A,B,C)
\tkzDrawSegments[dashed](A,I_a B,I_b C,I_c)
\tkzAutoLabelPoints[center=I,
blue,font=\scriptsize](I_a,I_b,I_c)
\tkzAutoLabelPoints[center=I,red,
font=\scriptsize](A,B,C,I_a,I_b,I_c)
\end{tikzpicture}
```

16.0.3 ex或excentral选项

旁心三角形是由另一个三角形的三个旁心构成的三角形。

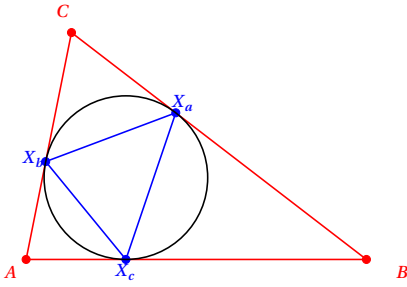


```
\begin{tikzpicture}[scale=0.60]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[excentral,name=J](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[extouch,name=T](A,B,C){_a,_b,_c}
\tkzDrawPolygon[blue](A,B,C)
\tkzDrawPolygon[red](J_a,J_b,J_c)
\tkzDrawPoints(A,B,C)
\tkzDrawPoints[red](J_a,J_b,J_c)
\tkzLabelPoints(A,B,C)
\tkzLabelPoints[red](J_b,J_c)
\tkzLabelPoints[red,above](J_a)
\tkzClipBB
\tkzShowBB
\tkzDrawCircles[gray](J_a,T_a J_b,T_b J_c,T_c)
\end{tikzpicture}
```

16.0.4 intouch选项

三角形的内接三角形是由三角形的内切圆的三个切点构成的三角形。

Weisstein, Eric W. “Contact triangle” From MathWorld—A Wolfram Web Resource.



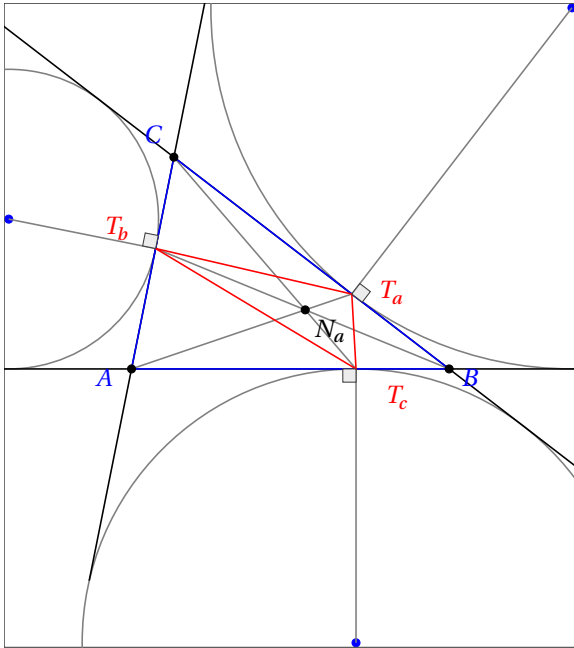
```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[intouch,name=X](A,B,C){_a,_b,_c}
\tkzInCenter(A,B,C)\tkzGetPoint{I}
\tkzDrawPolygon[red](A,B,C)
\tkzDrawPolygon[blue](X_a,X_b,X_c)
\tkzDrawPoints[red](A,B,C)
\tkzDrawPoints[blue](X_a,X_b,X_c)
\tkzDrawCircle[in](A,B,C)
\tkzAutoLabelPoints[center=I,blue,font=\scriptsize]%(X_a,X_b,X_c)
\tkzAutoLabelPoints[center=I,red,font=\scriptsize]%(A,B,C)
\end{tikzpicture}
```

16.0.5 extouch选项

外触三角形 $T_a T_b T_c$ 是由三角形 ABC 的三个旁切圆 J_a 、 J_b 和 J_c 的切点构成的三角形。

Weisstein, Eric W. “Extouch triangle” From MathWorld—A Wolfram Web Resource.

可以得到旁切圆的切点和由这三个切点构成的三角形。

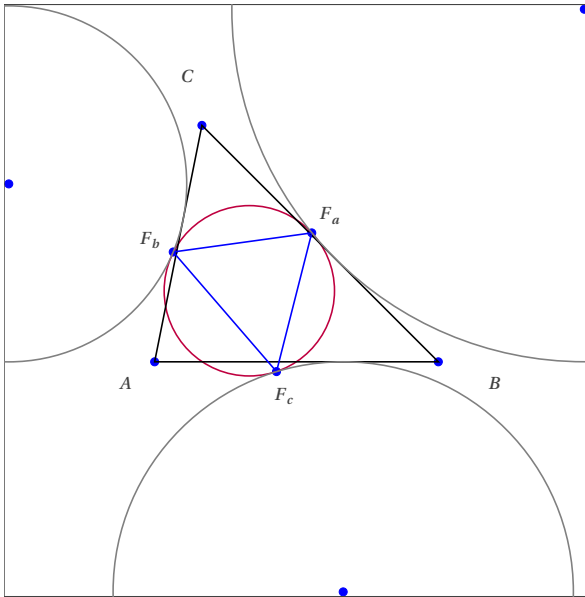


```
\begin{tikzpicture}[scale=.7]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[excentral,
name=J](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[extouch,
name=T](A,B,C){_a,_b,_c}
\tkzDefTriangleCenter[nagel](A,B,C)
\tkzGetPoint{N_a}
\tkzDefTriangleCenter[centroid](A,B,C)
\tkzGetPoint{G}
\tkzDrawPoints[blue](J_a,J_b,J_c)
\tkzClipBB \tkzShowBB
\tkzDrawCircles[gray](J_a,T_a J_b,T_b J_c,T_c)
\tkzDrawLines[add=1 and 1](A,B,C C,A)
\tkzDrawSegments[gray](A,T_a B,T_b C,T_c)
\tkzDrawSegments[gray](J_a,T_a J_b,T_b J_c,T_c)
\tkzDrawPolygon[blue](A,B,C)
\tkzDrawPolygon[red](T_a,T_b,T_c)
\tkzDrawPoints(A,B,C,N_a)
\tkzLabelPoints(N_a)
\tkzAutoLabelPoints[center=N_a,blue](A,B,C)
\tkzAutoLabelPoints[center=G,red,
dist=.4](T_a,T_b,T_c)
\tkzMarkRightAngles[fill=gray!15](J_a,T_a,B
J_b,T_b,C J_c,T_c,A)
\end{tikzpicture}
```

16.0.6 feuerbach选项

Feuerbach 三角形是由九点圆与三个旁切圆的三个切点构成的三角形。

Weisstein, Eric W. “Feuerbach triangle” From MathWorld—A Wolfram Web Resource.

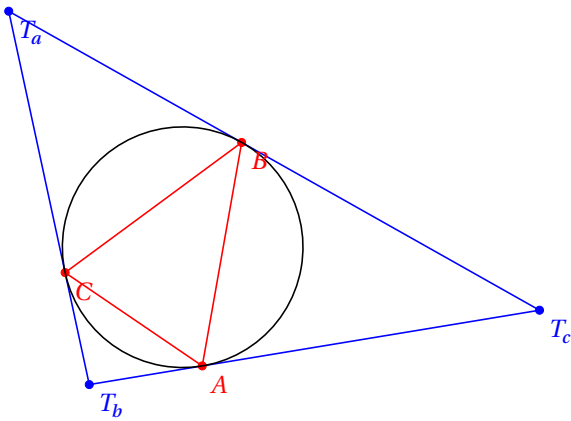


```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,0){B}
\tkzDefPoint(0.5,2.5){C}
\tkzDefCircle[euler](A,B,C) \tkzGetPoint{N}
\tkzDefSpcTriangle[feuerbach,
name=F](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[excentral,
name=J](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[extouch,
name=T](A,B,C){_a,_b,_c}
\tkzDrawPoints[blue](J_a,J_b,J_c,F_a,F_b,F_c,A,B,C)
\tkzClipBB
\tkzShowBB
\tkzDrawCircle[purple](N,F_a)
\tkzDrawPolygon(A,B,C)
\tkzDrawPolygon[blue](F_a,F_b,F_c)
\tkzDrawCircles[gray](J_a,F_a J_b,F_b J_c,F_c)
\tkzAutoLabelPoints[center=N,dist=.3,
font=\scriptsize](A,B,C,F_a,F_b,F_c,J_a,J_b,J_c)
\end{tikzpicture}
```

16.0.7 tangential选项

切向三角形是三角形 ABC 外接圆在三个顶点处的切线构成的三角形 $T_a T_b T_c$ 。因此，它是相对于三角形 ABC 外心的反三角形。

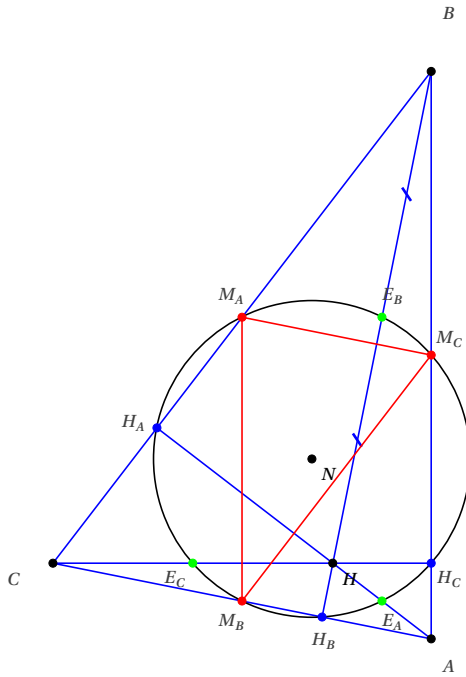
Weisstein, Eric W. “Tangential Triangle” From MathWorld—A Wolfram Web Resource.



```
\begin{tikzpicture}[scale=.5,rotate=80]
\tkzDefPoints{0/0/A,6/0/B,1.8/4/C}
\tkzDefSpcTriangle[tangential,
name=T](A,B,C){_a,_b,_c}
\tkzDrawPolygon[red](A,B,C)
\tkzDrawPolygon[blue](T_a,T_b,T_c)
\tkzDrawPoints[red](A,B,C)
\tkzDrawPoints[blue](T_a,T_b,T_c)
\tkzDefCircle[circum](A,B,C)
\tkzGetPoint{O}
\tkzDrawCircle(O,A)
\tkzLabelPoints[red](A,B,C)
\tkzLabelPoints[blue](T_a,T_b,T_c)
\end{tikzpicture}
```


16.0.8 euler选项

欧拉三角形是由三角形 ABC 的垂心 H 与三个顶点连线中点构成的三角形 $E_A E_B E_C$ ，欧拉三角形的顶点是欧拉点，它们位于三角形的九点圆上。



```
\begin{tikzpicture}[rotate=90,scale=1.25]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[medial,
                    name=M](A,B,C){_A,_B,_C}
\tkzDefTriangleCenter[euler](A,B,C)
\tkzGetPoint{N} % I= N nine points
\tkzDefTriangleCenter[ortho](A,B,C)
\tkzGetPoint{H}
\tkzDefMidPoint(A,H) \tkzGetPoint{E_A}
\tkzDefMidPoint(C,H) \tkzGetPoint{E_C}
\tkzDefMidPoint(B,H) \tkzGetPoint{E_B}
\tkzDefSpcTriangle[ortho,name=H](A,B,C){_A,_B,_C}
\tkzDrawPolygon[color=blue](A,B,C)
\tkzDrawCircle(N,E_A)
\tkzDrawSegments[blue](A,H_A B,H_B C,H_C)
\tkzDrawPoints(A,B,C,N,H)
\tkzDrawPoints[red](M_A,M_B,M_C)
\tkzDrawPoints[blue](H_A,H_B,H_C)
\tkzDrawPoints[green](E_A,E_B,E_C)
\tkzAutoLabelPoints[center=N,font=\scriptsize]%
(A,B,C,M_A,M_B,M_C,H_A,H_B,H_C,E_A,E_B,E_C)
\tkzLabelPoints[font=\scriptsize](H,N)
\tkzMarkSegments[mark=s|,size=3pt,
                  color=blue,line width=1pt](B,E_B E_B,H)
\tkzDrawPolygon[color=red](M_A,M_B,M_C)
\end{tikzpicture}
```

17 定义多边形

17.1 定义正方形

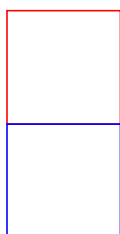
`\tkzDefSquare(<pt1,pt2>)`

通过两个点按逆时针方向推算另外两个点后,得到正方形。结果保存在`tkzFirstPointResult`和`tkzSecondPointResult`中。当然,可以使用`\tkzGetPoints`命令为这两个点重命名。

参数	样例	说明
<code>(<pt1,pt2>)</code>	<code>\tkzDefSquare(<A,B>)</code>	按指定的方向定义正方形

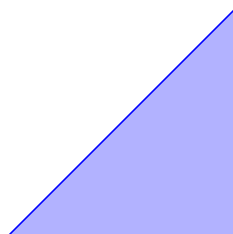
17.1.1 用`\tkzDefSquare`命令通过两个点定义正方形

需要注意点的方向问题。



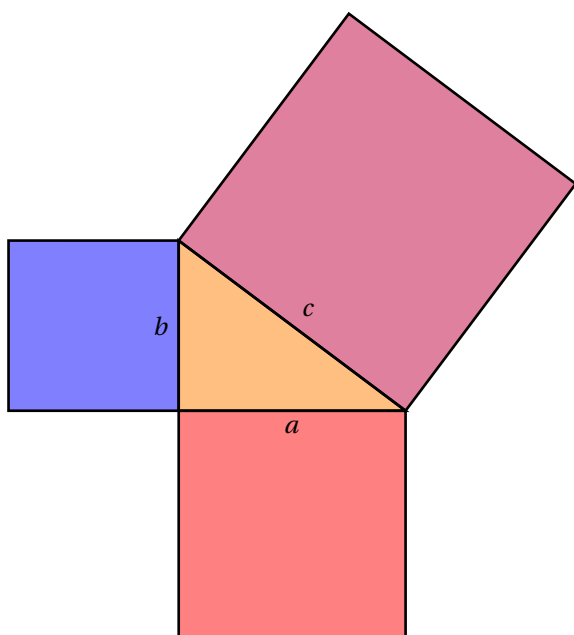
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,0){A} \tkzDefPoint(3,0){B}
\tkzDefSquare(A,B)
\tkzDrawPolygon[color=red](A,B,tkzFirstPointResult,%
tkzSecondPointResult)
\tkzDefSquare(B,A)
\tkzDrawPolygon[color=blue](B,A,tkzFirstPointResult,%
tkzSecondPointResult)
\end{tikzpicture}
```

可以使用`\tkzGetFirstPoint`或`\tkzGetSecondPoint`命令利用其中的1个点绘制等腰直角三角形。

17.1.2 用`\tkzDefSquare`命令绘制等腰直角三角形

```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,0){B}
\tkzDefSquare(A,B) \tkzGetFirstPoint{C}
\tkzDrawPolygon[color=blue,fill=blue!30](A,B,C)
\end{tikzpicture}
```

17.1.3 利用\tkzDefSquare绘制 Pythagorean 定理示意图



```
\begin{tikzpicture}[scale=.75]
\tkzInit
\tkzDefPoint(0,0){C}
\tkzDefPoint(4,0){A}
\tkzDefPoint(0,3){B}
\tkzDefSquare(B,A)\tkzGetPoints{E}{F}
\tkzDefSquare(A,C)\tkzGetPoints{G}{H}
\tkzDefSquare(C,B)\tkzGetPoints{I}{J}
\tkzFillPolygon[fill = red!50 ](A,C,G,H)
\tkzFillPolygon[fill = blue!50 ](C,B,I,J)
\tkzFillPolygon[fill = purple!50](B,A,E,F)
\tkzFillPolygon[fill = orange,opacity=.5](A,B,C)
\tkzDrawPolygon[line width = 1pt](A,B,C)
\tkzDrawPolygon[line width = 1pt](A,C,G,H)
\tkzDrawPolygon[line width = 1pt](C,B,I,J)
\tkzDrawPolygon[line width = 1pt](B,A,E,F)
\tkzLabelSegment[](A,C){$a$}
\tkzLabelSegment[](C,B){$b$}
\tkzLabelSegment[swap](A,B){$c$}
\end{tikzpicture}
```

17.2 定义平行四边形

17.3 定义平行四边形的顶点

可以通过 3 个点来定义一个平行四边形。

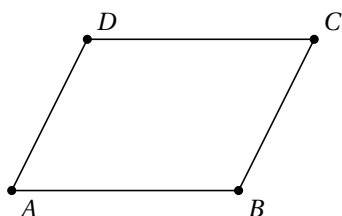
`\tkzDefParallelogram(<pt1,pt2,pt3>)`

通过 3 个点，通过计算另一个点，构成一个平行四边形，结果保存在 `\tkzPointResult` 中。

可以使用命令为其命名 `\tkzGetPoint...`。

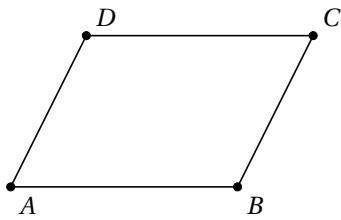
参数	默认值	含义
<code>(<pt1,pt2,pt3>)</code>	无	必须的 3 个顶点

17.3.1 平行四边形定义



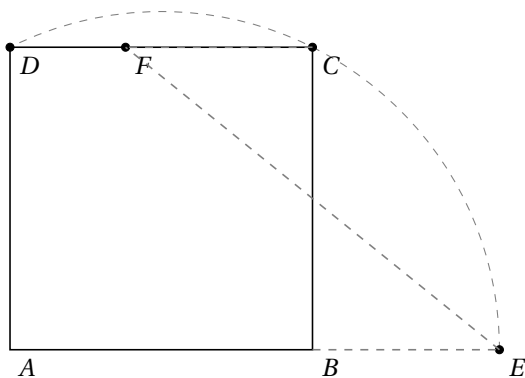
```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{0/0/A,3/0/B,4/2/C}
\tkzDefParallelogram(A,B,C)
\tkzGetPoint{D}
\tkzDrawPolygon(A,B,C,D)
\tkzLabelPoints(A,B)
\tkzLabelPoints[above right](C,D)
\tkzDrawPoints(A,...,D)
\end{tikzpicture}
```

17.3.2 简单示例



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{0/0/A,3/0/B,4/2/C}
  \tkzDefPointWith[colinear= at C](B,A)
  \tkzGetPoint{D}
  \tkzDrawPolygon(A,B,C,D)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above right](C,D)
  \tkzDrawPoints(A,...,D)
\end{tikzpicture}
```

17.3.3 黄金比例矩形



```
\begin{tikzpicture}[scale=.5]
  \tkzInit[xmax=14,ymax=10]
  \tkzClip[space=1]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(8,0){B}
  \tkzDefMidPoint(A,B)\tkzGetPoint{I}
  \tkzDefSquare(A,B)\tkzGetPoints{C}{D}
  \tkzDrawSquare(A,B)
  \tkzInterLC(A,B)(I,C)\tkzGetPoints{G}{E}
  \tkzDrawArc[style=dashed,color=gray](I,E)(D)
  \tkzDefPointWith[colinear= at C](E,B)
  \tkzGetPoint{F}
  \tkzDrawPoints(C,D,E,F)
  \tkzLabelPoints(A,B,C,D,E,F)
  \tkzDrawSegments[style=dashed,color=gray](E,F C,F B,E)
\end{tikzpicture}
```

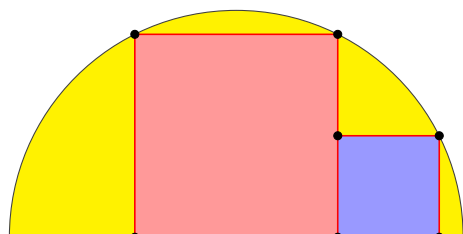
17.4 绘制正方形

\tkzDrawSquare[< 命令选项>](<pt1,pt2>)

用于绘制一个正方形，但不绘制顶点。可以对内部进行着色，点的顺序是逆时针方向。

参数	样例	说明
(<pt1,pt2>)	\tkzDrawSquare(<A,B>)	\tkzGetPoints{C}{D}
选项	样例	说明
Options TikZ	red,line width=1pt	

17.4.1 在半圆内绘制两个正方形



```
\begin{tikzpicture}[scale=.75]
  \tkzInit[ymax=8,xmax=8]
  \tkzClip[space=.25]   \tkzDefPoint(0,0){A}
  \tkzDefPoint(8,0){B}   \tkzDefPoint(4,0){I}
  \tkzDefSquare(A,B)   \tkzGetPoints{C}{D}
  \tkzInterLC(I,C)(I,B) \tkzGetPoints{E'}{E}
  \tkzInterLC(I,D)(I,B) \tkzGetPoints{F'}{F}
  \tkzDefPointsBy[projection=onto A--B](E,F){H,G}
  \tkzDefPointsBy[symmetry = center H](I){J}
  \tkzDefSquare(H,J)   \tkzGetPoints{K}{L}
  \tkzDrawSector[fill=yellow](I,B)(A)
  \tkzFillPolygon[color=red!40](H,E,F,G)
  \tkzFillPolygon[color=blue!40](H,J,K,L)
  \tkzDrawPolySeg[color=red](H,E,F,G)
  \tkzDrawPolySeg[color=red](J,K,L)
  \tkzDrawPoints(E,G,H,F,J,K,L)
\end{tikzpicture}
```

17.5 定义黄金矩形

`\tkzDefGoldRectangle(<point,point>)`

定义长宽比为黄金分割比 Φ 的黄金矩形。结果保存在 `\tkzFirstPointResult` 的 `\tkzSecondPointResult` 中。可以用 `\tkzGetPoints` 为这两个点重命名，并在后续代码中进行引用。

参数	样例	说明
<code>(<pt1,pt2>)</code>	<code>(<A,B>)</code>	如果用 C 和 D 表示得到的点，则 $AB/BC = \Phi$ 。

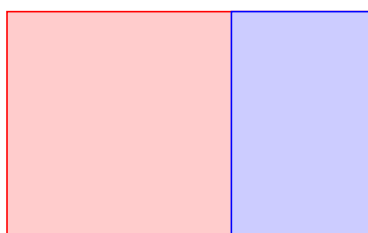
17.6 绘制黄金矩形

`\tkzDrawGoldRectangle[<命令选项>](<point,point>)`

参数	样例	说明
<code>(<pt1,pt2>)</code>	<code>(<A,B>)</code>	根据线段 $[AB]$ 绘制黄金矩形

选项	样例	说明
Options TikZ	<code>red,line width=1pt</code>	

17.6.1 黄金矩形示例



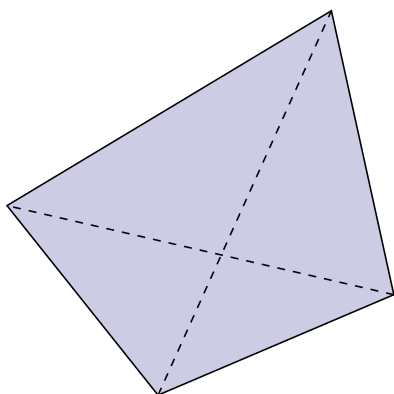
```
\begin{tikzpicture}[scale=.6]
  \tkzDefPoint(0,0){A}   \tkzDefPoint(8,0){B}
  \tkzDefGoldRectangle(A,B) \tkzGetPoints{C}{D}
  \tkzDefGoldRectangle(B,C) \tkzGetPoints{E}{F}
  \tkzDrawPolygon[color=red,fill=red!20](A,B,C,D)
  \tkzDrawPolygon[color=blue,fill=blue!20](B,C,E,F)
\end{tikzpicture}
```

17.7 绘制多边形命令

`\tkzDrawPolygon[< 命令选项>](< 点集列表>)`

用给定的点集，根据指定的 TikZ 选项绘制多边形。连续的点可以省略中间的点，例如，可以使用 (A, \dots, E) 表示点集 (A, B, C, D, E) ，用 $(P_1, P \dots, P_5)$ 表示点集 $(P_1, P_2, P_3, P_4, P_5)$ 。

参数	样例	说明
<code>(\langle pt1, pt2, pt3, \dots \rangle)</code>	<code>\tkzDrawPolygon[gray,dashed](A,B,C)</code>	绘制一个三角形
选项	默认值	样例
Options TikZ ...		<code>\tkzDrawPolygon[red,line width=2pt](A,B,C)</code>

17.7.1 `\tkzDrawPolygon` 命令

```
\begin{tikzpicture}[rotate=18,scale=1.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2.25,0.2){B}
\tkzDefPoint(2.5,2.75){C}
\tkzDefPoint(-0.75,2){D}
\tkzDrawPolygon[fill=black!50!blue!20!](A,B,C,D)
\tkzDrawSegments[style=dashed](A,C B,D)
\end{tikzpicture}
```

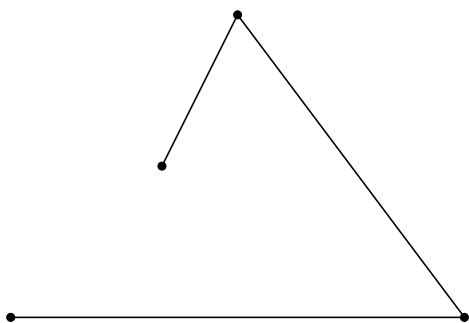
17.8 绘制多边形顶点链条

`\tkzDrawPolySeg[< 命令选项>](< 点集列表>)`

绘制多边形顶点构成的拆线。

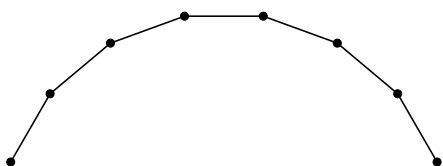
参数	样例	说明
<code>(\langle pt1, pt2, pt3, \dots \rangle)</code>	<code>\tkzDrawPolySeg[gray,dashed](A,B,C)</code>	绘制一个三角形
选项	默认值	样例
Options TikZ ...		<code>\tkzDrawPolySeg[red,line width=2pt](A,B,C)</code>

17.8.1 多边形顶点链条



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,6/0/B,3/4/C,2/2/D}
\tkzDrawPolySeg(A,...,D)
\tkzDrawPoints(A,...,D)
\end{tikzpicture}
```

17.8.2 多边形顶点链条：循环实现



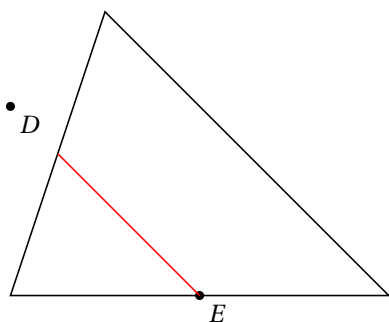
```
\begin{tikzpicture}
\foreach \pt in {1,2,...,8} {%
\tkzDefPoint(\pt*20:3){P_\pt}}
\tkzDrawPolySeg(P_1,P_...,P_8)
\tkzDrawPoints(P_1,P_...,P_8)
\end{tikzpicture}
```

17.9 多边形裁剪

`\tkzClipPolygon[< 命令选项>](< 点集列表>)`

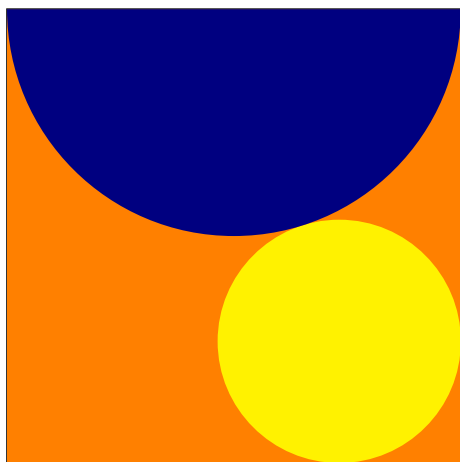
用指定的多边形对图形实行裁剪。

参数	样例	说明
<code>(<pt1,pt2>)</code>	<code>(<A,B>)</code>	

17.9.1 `\tkzClipPolygon` 命令

```
\begin{tikzpicture}[scale=1.25]
\tkzInit[xmin=0,xmax=4,ymin=0,ymax=3]
\tkzClip[space=.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(1,3){C}
\tkzDrawPolygon(A,B,C)
\tkzDefPoint(0,2){D}
\tkzDefPoint(2,0){E}
\tkzDrawPoints(D,E)
\tkzLabelPoints(D,E)
\tkzClipPolygon(A,B,C)
\tkzDrawLine[color=red](D,E)
\end{tikzpicture}
```

17.9.2 使用"裁剪"将 Sangaku 限制在正方形内



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
\tkzDefSquare(A,B) \tkzGetPoints{C}{D}
\tkzDrawPolygon(B,C,D,A)
\tkzClipPolygon(B,C,D,A)
\tkzDefPoint(4,8){F}
\tkzDefTriangle[equilateral](C,D)
\tkzGetPoint{I}
\tkzDrawPoint(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 = orange](A,B,C,D)
\tkzFillCircle[R,color = yellow](M,\dMI pt)
\tkzFillCircle[R,color = blue!50!black](F,4 cm)%
\end{tikzpicture}
```

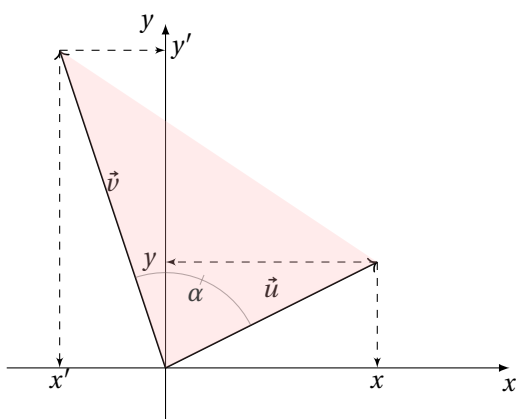
17.10 多边形着色

`\tkzFillPolygon[< 命令选项>](< 点集列表>)`

可以在绘制多边形时着色，但该命令仅对多边形内部进行着色，而不绘制多边形。

参数	样例	说明
<code>(\pt1,\pt2,...)</code>	<code>((A,B,...))</code>	

17.10.1 \tkzFillPolygon命令

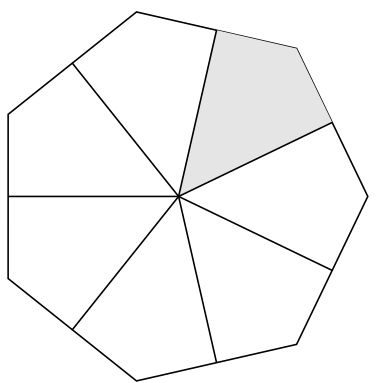


```
\begin{tikzpicture}[scale=0.7]
\tkzInit[xmin=-3,xmax=6,ymin=-1,ymax=6]
\tkzDrawX[noticks]
\tkzDrawY[noticks]
\tkzDefPoint(0,0){O} \tkzDefPoint(4,2){A}
\tkzDefPoint(-2,6){B}
\tkzPointShowCoord[xlabel=$x$,ylabel=$y$](A)
\tkzPointShowCoord[xlabel=$x'$,ylabel=$y'$,$
ystyle={right=2pt}](B)
\tkzDrawSegments[->](O,A O,B)
\tkzLabelSegment[above=3pt](O,A){$\vec{u}$}
\tkzLabelSegment[above=3pt](O,B){$\vec{v}$}
\tkzMarkAngle[fill= yellow,size=1.8cm,%
opacity=.5](A,O,B)
\tkzFillPolygon[red!30,opacity=0.25](A,B,O)
\tkzLabelAngle[pos = 1.5](A,O,B){$\alpha$}
\end{tikzpicture}
```


17.11 正多边形

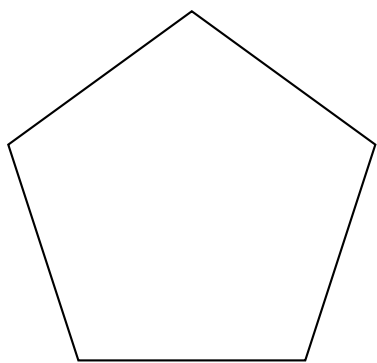
\tkzDefRegPolygon[< 命令选项>](<pt1,pt2>)		
根据选项中指定的边数，以指定的点为中心或是指定的边，定义一个正多边形。		
参数	样例	说明
(<pt1,pt2>)	(<O,A>)	如果使用"center"选项，则 O 是多边形中心
(<pt1,pt2>)	(<A,B>)	如果使用"side"选项，[AB] 一条边
选项	默认值	样例
name	P	顶点命名为 P1, P2, ...
sides	5	边数
center	center	第 1 个点 是正多边形中心
side	center	批定的两个顶点是一个边
Options TikZ	...	

17.11.1 center 选项



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoints{0/0/P0,0/0/Q0,2/0/P1}
\tkzDefMidPoint(P0,P1) \tkzGetPoint{Q1}
\tkzDefRegPolygon[center,sides=7](P0,P1)
\tkzDefMidPoint(P1,P2) \tkzGetPoint{Q1}
\tkzDefRegPolygon[center,sides=7,name=Q](P0,Q1)
\tkzDrawPolygon(P1,P...,P7)
\tkzFillPolygon[gray!20](Q0,Q1,P2,Q2)
\foreach \j in {1,...,7} {
\tkzDrawSegment[black](P0,Q\j)}
\end{tikzpicture}
```

17.11.2 side 选项



```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{-4/0/A, -1/0/B}
\tkzDefRegPolygon[side,sides=5,name=P](A,B)
\tkzDrawPolygon[thick](P1,P...,P5)
\end{tikzpicture}
```

18 圆

通过本节的命令中，可以定义并绘制圆。为此，需要知道圆的圆心以及圆的半径或是圆上的点。可能最常用的方法就是用给定的中心绘制过指定的点的圆，这将是默认的方法，否则则需要给出圆的半径 R 。另外，还有一些特定的圆，例如三角形的外接圆。

- `\tkzDefCircle`命令根据指定的圆心和半径(单位:cm)定义一个圆,通过`\tkzGetPoint`和`\tkzGetLength`命令实现计算;
- `\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`命令用于对圆内的内容进行裁剪;
- `\tkzLabelCircle`命令用于标注一个圆.

18.1 定义圆的命令: `\tkzDefCircle`

该命令用指定的圆心和半径定义一下圆。

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

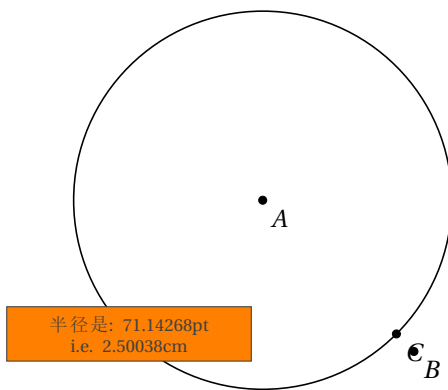
注意，参数可以是2个点或3个点。该命令结合`\tkzGetPoint`命令和/或`\tkzGetLength`命令，以得到圆心和圆的半径，或使用`\tkzPointResult`命令和`\tkzLengthResult`命令临时使用这些值，但不保存结果。

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

选项	默认值	含义
<code>through</code>	<code>through</code>	两个点间的距离是半径
<code>diameter</code>	<code>through</code>	两个点间的距离是直径
<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>	<code>1</code>	Apollonius 圆的系数

下面的示例中，会用到还未说明的圆的绘制命令，不过这不是必须的，多数情况下，仅需要得到圆心和半径就可以了。

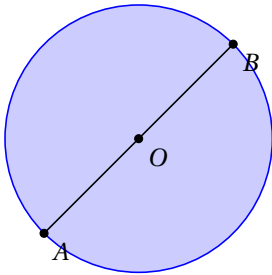
18.1.1 使用随机点与through选项



```
\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}
```

18.1.2 diameter选项

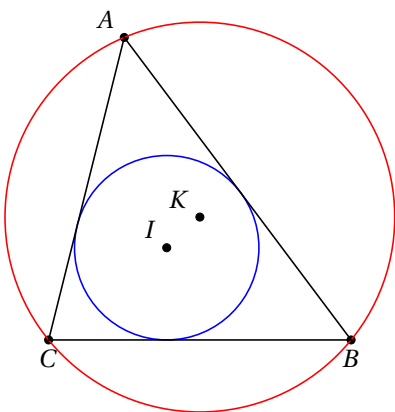
可以通过 $[AB]$ 的中点确定圆心。



```
\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}
```

18.1.3 三角形的内切圆和外接圆

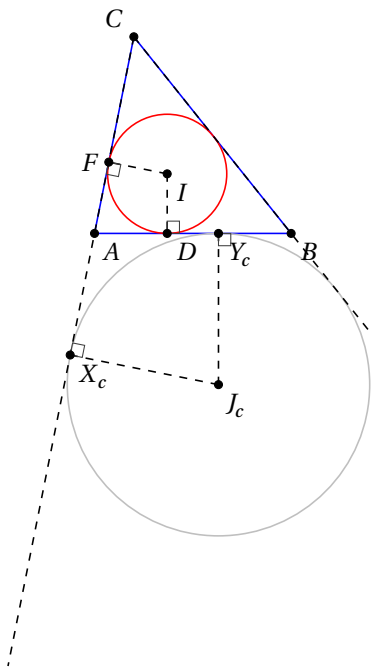
可以使用 `\tkzGetFirstPointI` 和 `\tkzGetSecondPointIb` 命令得到内切圆在对应边上的投影。



```
\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}
```

18.1.4 ex选项

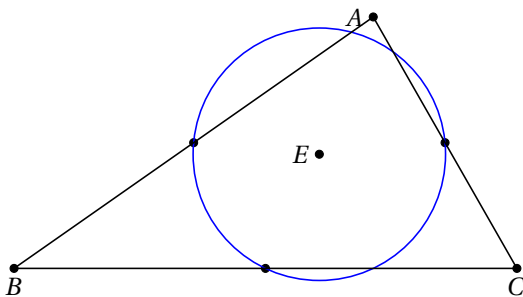
与顶点 C 对应的旁切圆。



```
\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}
```

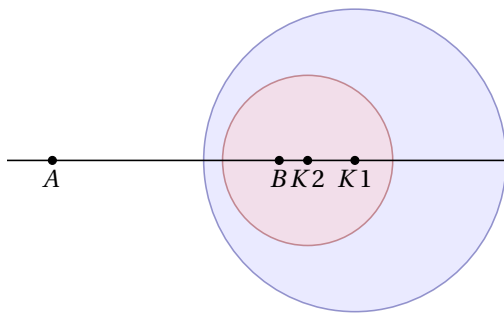
18.1.5 euler选项

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}
```

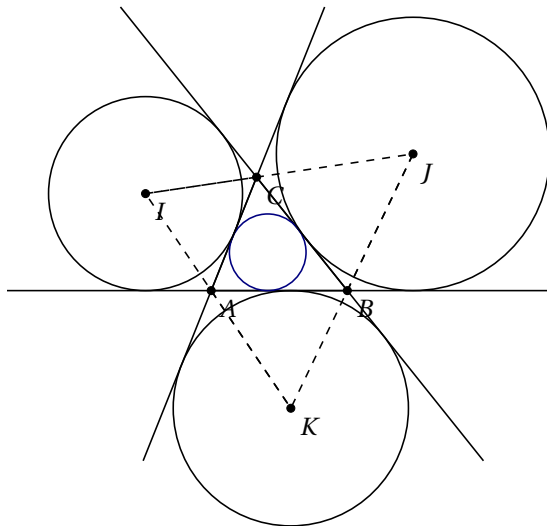
18.1.6 apollonius选项



```
\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}
```

18.1.7 ex选项

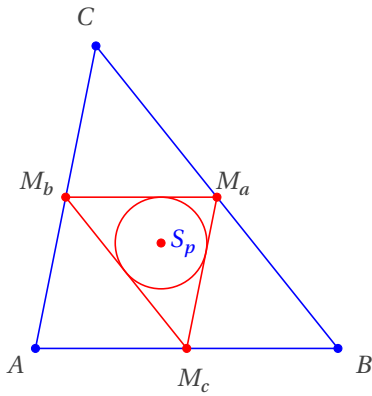
可以使用`\tkzGetFirstPoint{Jb}`和`\tkzGetSecondPoint{Tb}`命令，得到旁切圆圆心在边上的投影。



```
\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}
```

18.1.8 spieker选项

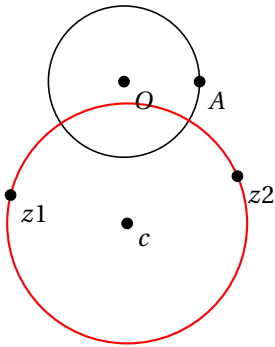
三角形三个边的中点构成的三角形 $M_aM_bM_c$ 的内切圆是 Spieker 圆:



```
\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}
```

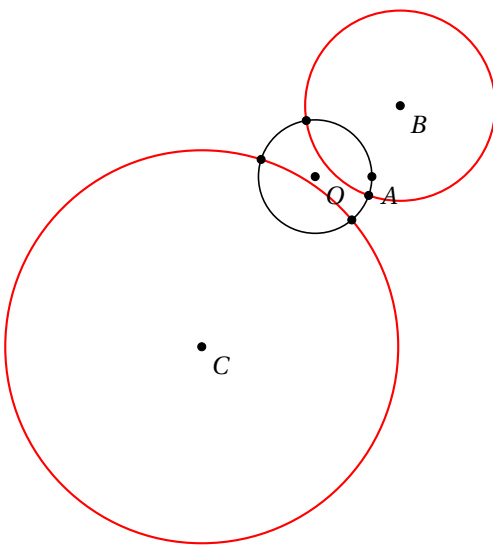
18.1.9 orthogonal through选项

通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}
```

18.1.10 指定圆心的另一个圆的正交圆



```
\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}
```

19 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.

19.1 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。可以通过R选项直接指定长度。

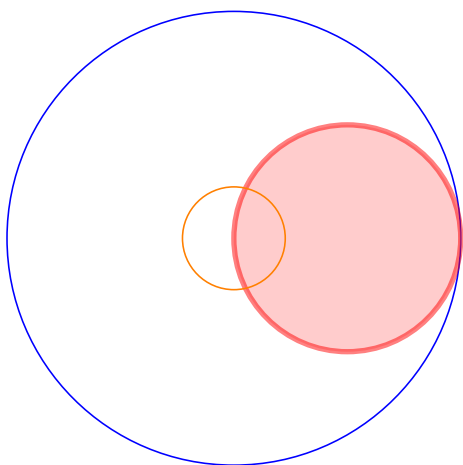
参数	样例	说明
<code>(pt1,pt2)</code>	<code>(A,B)</code>	两个点定义半径或直径

选项	默认值	定义
<code>through</code>	<code>through</code>	用两个点定义半径
<code>diameter</code>	<code>through</code>	用两个点定义直径
<code>R</code>	<code>through</code>	半径由指定的点确定

当然, 可以使用所有有效 TikZ 样式。

19.1.1 绘制一个圆并对其进行着色

可以看到, 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}
```

19.2 绘制多个圆

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

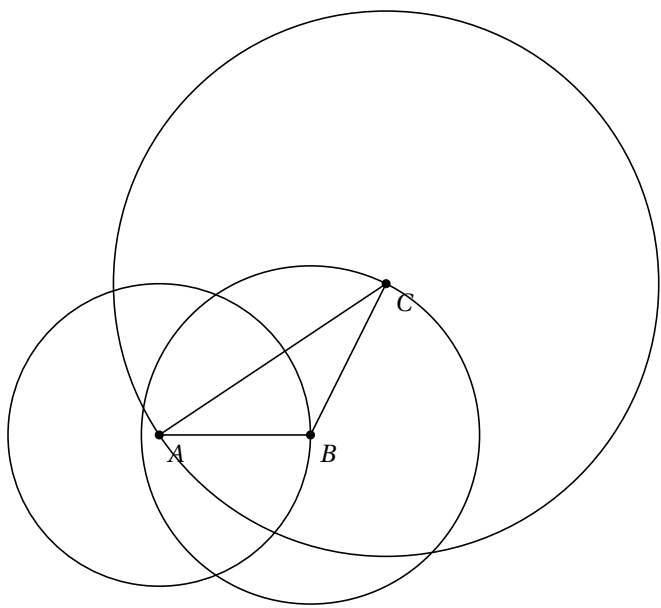
 注意：参数是空格分隔的构成圆的点对列表，点对中两个点之间和逗号分隔。可以使用R参数直接指定半径度量点。

参数	样例	说明
(<pt1,pt2 pt3,pt4,...>)	(<A,B C,D>)	点集列表

选项	默认值	含义
through	through	点对定义的是每个圆的半径
diameter	through	点对定义的是第个圆的直径
R	through	通过指定的点计算每个圆的半径

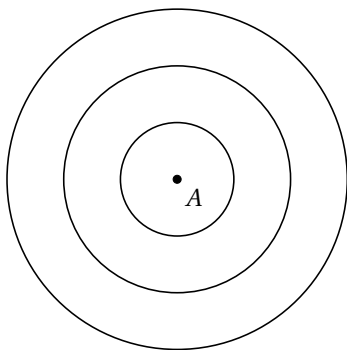
当然，可以使用所有的有效 TikZ 样式。

19.2.1 通过三角形定义圆



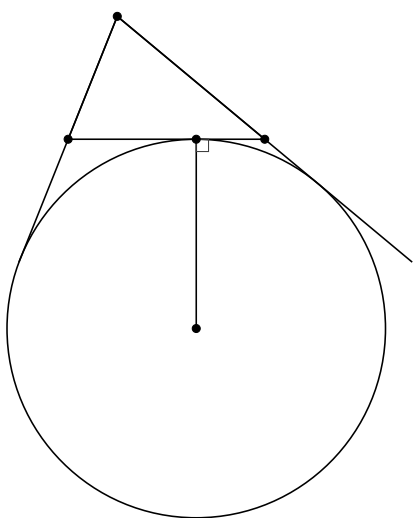
```
\begin{tikzpicture}[scale=1.0]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,0){B}
\tkzDefPoint(3,2){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawCircles(A,B B,C C,A)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(A,B,C)
\end{tikzpicture}
```


19.2.2 同心圆



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

19.2.3 旁切圆

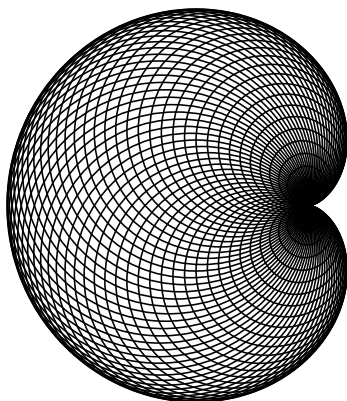


```
\begin{tikzpicture}[scale=0.65]
\tkzDefPoints{0/0/A,4/0/B,1/2.5/C}
\tkzDrawPolygon(A,B,C)
\tkzDefCircle[ex](B,C,A)
\tkzGetPoint{J_c} \tkzGetSecondPoint{T_c}
\tkzGetLength{rJc}
\tkzDrawCircle[R](J_c,{\rJc pt})
\tkzDrawLines[add=0 and 1](C,A C,B)
\tkzDrawSegment(J_c,T_c)
\tkzMarkRightAngle(J_c,T_c,B)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

19.2.4 心形线

基于 O. Rebourg 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)。

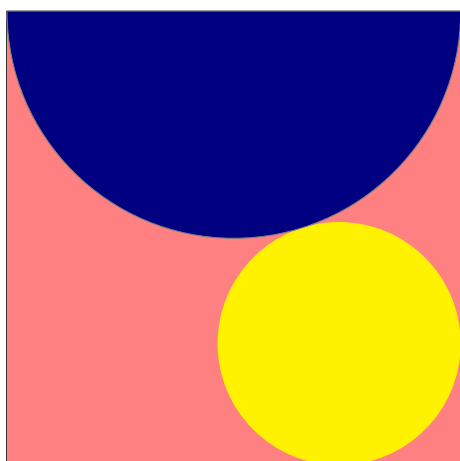


```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,0){A}
\foreach \ang in {5,10,...,360}{%
\tkzDefPoint(\ang:2){M}
\tkzDrawCircle(M,A)
}
\end{tikzpicture}
```

19.3 绘制半圆

`\tkzDrawSemiCircle[< 命令选项>](⟨A,B⟩)`

参数	样例	说明
<code>(⟨pt1,pt2⟩)</code>	<code>(⟨0,A⟩)</code> or <code>(⟨A,B⟩)</code>	半径或直径
选项	默认值	含义
<code>through</code>	<code>through</code>	两个点定义了半径
<code>diameter</code>	<code>through</code>	两个点定义了直径

19.3.1 `\tkzDrawSemiCircle` 命令

```
\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}
```

19.4 给圆着色

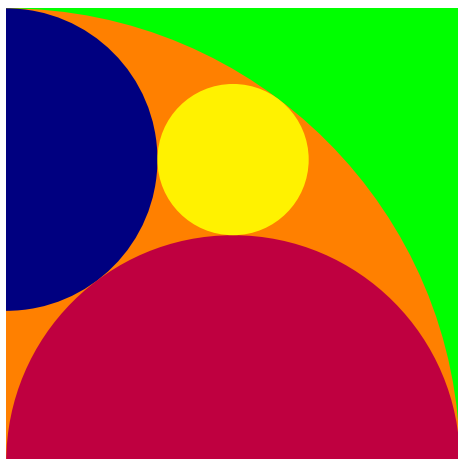
在绘制圆时，也可以实现着色，但该命令可以不绘制圆，仅对圆形区域进行着色处理。

`\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>	用另外一个点与圆心定义半径

当然，可以使用所有有效 TikZ 样式。

19.4.1 sangaku 圆



```
\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}
```

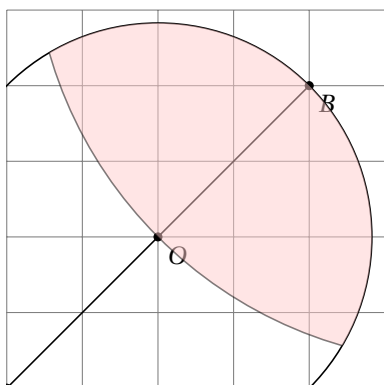
19.5 圆形裁剪

`\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	用指定点确定半径

19.5.1 样例



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

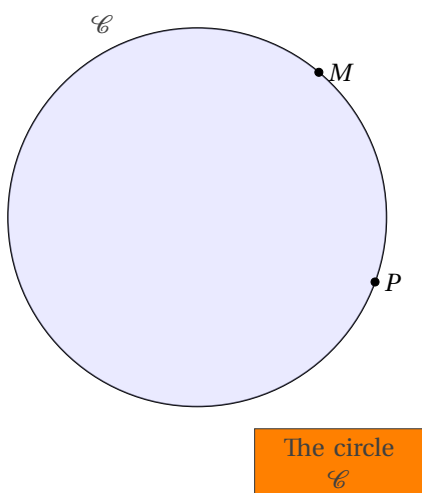
19.6 圆的标注

`\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	用指定的点确定半径

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

19.6.1 标注样例



```
\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}
```

20 交点

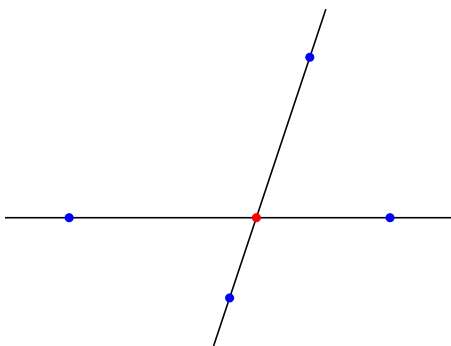
可以求得两条直线、一条直线与一个圆及两个圆之间的交点。
求交点的相关命令没有可选参数，用户必须确保存在交点。

20.1 两条直线的交点

`\tkzInterLL($\langle A,B \rangle$)($\langle C,D \rangle$)`

求直线 (AB) 和 (CD) 的交点，并保存于 `\tkzPointResult` 命令中，两条直线分别由两个圆括号中的两个点定义。可以通过 `\tkzDefPoint` 为该交点命名，并在后续代码中进行引用。

20.1.1 两个直线交点示例



```
\begin{tikzpicture}[rotate=-45,scale=.75]
\tkzDefPoint(2,1){A}
\tkzDefPoint(6,5){B}
\tkzDefPoint(3,6){C}
\tkzDefPoint(5,2){D}
\tkzDrawLines(A,B C,D)
\tkzInterLL(A,B)(C,D)
\tkzGetPoint{I}
\tkzDrawPoints[color=blue](A,B,C,D)
\tkzDrawPoint[color=red](I)
\end{tikzpicture}
```

20.2 一条直线和一个圆的交点

如前所述，一条直线可以由两个点定义，一个圆可以按如下方式进行定义：

- (O,C) 是一对点，第 1 个是圆心，第 2 个是圆上的一个点。
- (O,r) O 是圆心， r 半径，单位可以是 *cm* 可 *pt*。

`\tkzInterLC[\langle 命令选项 \rangle]($\langle A,B \rangle$)($\langle O,C \rangle$) 或 ($\langle O,r \rangle$) 可 ($\langle O,C,D \rangle$)`

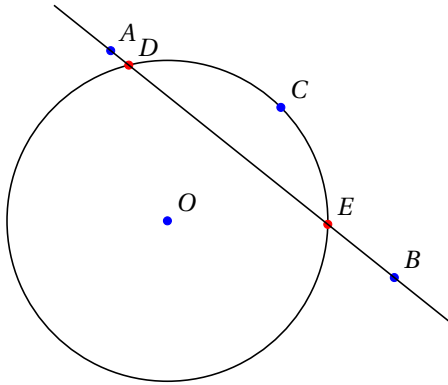
参数必须是一条直线和一个圆。

选项	默认值	含义
N	N	(O,C) 确定圆
R	N	$(O, 1 \text{ cm})$ 或 $(O, 120 \text{ pt})$
with nodes	N	(O,C,D) CD 是半径

该命令定义了直线与由圆心 O 和半径 r 定义的圆的交点 I 和 J ，如果出现错误，则记录在中.log。

20.2.1 直线与圆的交点示例

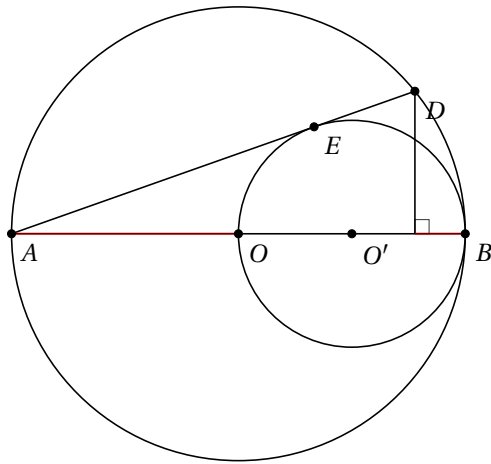
在下面示例代码中，圆用两个点表示，直线与圆的有两个交点。



```
\begin{tikzpicture}[scale=.75]
  \tkzInit[xmax=5,ymax=4]
  \tkzDefPoint(1,1){O}
  \tkzDefPoint(0,4){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPoint(3,3){C}
  \tkzInterLC(A,B)(O,C) \tkzGetPoints{D}{E}
  \tkzDrawCircle(O,C)
  \tkzDrawPoints[color=blue](O,A,B,C)
  \tkzDrawPoints[color=red](D,E)
  \tkzDrawLine(A,B)
  \tkzLabelPoints[above right](O,A,B,C,D,E)
\end{tikzpicture}
```

20.2.2 直线与圆的交点复杂示例

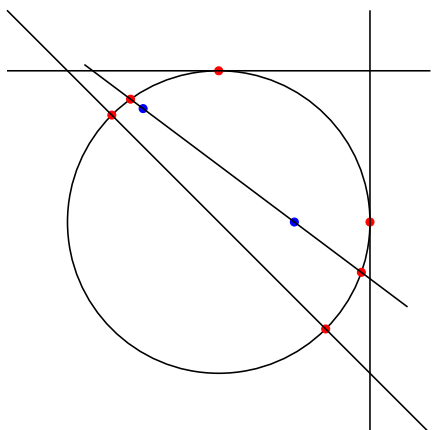
可参阅http://www.gogeometry.com/problem/p190_tangent_circle_diameter_perpendicular.htm



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
  \tkzDefMidPoint(A,B) \tkzGetPoint{O}
  \tkzDrawCircle(O,B)
  \tkzDefMidPoint(O,B) \tkzGetPoint{O'}
  \tkzDrawCircle(O',B)
  \tkzDefTangent[from=A](O',B)
  \tkzGetSecondPoint{E}
  \tkzInterLC(A,E)(O,B)
  \tkzGetSecondPoint{D}
  \tkzDefPointBy[projection=onto A--B](D)
  \tkzGetPoint{F}
  \tkzMarkRightAngle(D,F,B)
  \tkzDrawSegments(A,D A,B D,F)
  \tkzDrawSegments[color=red,line width=1pt,
    opacity=.4](A,O F,B)
  \tkzDrawPoints(A,B,O,O',E,D)
  \tkzLabelPoints(A,B,O,O',E,D)
\end{tikzpicture}
```


20.2.3 由圆心和半径定义圆

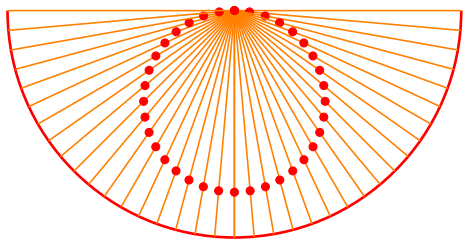
直线与圆相切的特例：



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,8){A} \tkzDefPoint(8,0){B}
\tkzDefPoint(8,8){C} \tkzDefPoint(4,4){I}
\tkzDefPoint(2,7){E} \tkzDefPoint(6,4){F}
\tkzDrawCircle[R](I,4 cm)
\tkzInterLC[R](A,C)(I,4 cm) \tkzGetPoints{I1}{I2}
\tkzInterLC[R](B,C)(I,4 cm) \tkzGetPoints{J1}{J2}
\tkzInterLC[R](A,B)(I,4 cm) \tkzGetPoints{K1}{K2}
\tkzDrawPoints[color=red](I1,J1,K1,K2)
\tkzDrawLines(A,B B,C A,C)
\tkzInterLC[R](E,F)(I,4 cm) \tkzGetPoints{I2}{J2}
\tkzDrawPoints[color=blue](E,F)
\tkzDrawPoints[color=red](I2,J2)
\tkzDrawLine(I2,J2)
\end{tikzpicture}
```

20.2.4 更为复杂的例子

 注意语法细节：首先，在传递参数的同时，可以计算点的坐标，但是必须嵌套`xfp`语法。由于`xfp`宏包能够使用弧度，在此使用了`pi`，当然，也可以使用度，但需要使用类似`sind`或`cosd`命令进行计算。其次，当计算中需要圆括号时，需要使用分组命令：`...TeX{...}`。



```
\begin{tikzpicture}[scale=1.2]
\tkzDefPoint(0,1){J} \tkzDefPoint(0,0){O}
\tkzDrawArc[R,line width=1pt,color=red](J,2.5 cm)(180,0)
\foreach \i in {0,-5,-10,...,-85,-90}{
\tkzDefPoint({2.5*cosd(\i)},{1+2.5*sind(\i)}){P}
\tkzDrawSegment[color=orange](J,P)
\tkzInterLC[R](P,J)(0,1 cm)
\tkzGetPoints{M}{N}
\tkzDrawPoints[red](N)
}
\foreach \i in {-90,-95,...,-175,-180}{
\tkzDefPoint({2.5*cosd(\i)},{1+2.5*sind(\i)}){P}
\tkzDrawSegment[color=orange](J,P)
\tkzInterLC[R](P,J)(0,1 cm)
\tkzGetPoints{M}{N}
\tkzDrawPoints[red](M)
}
\end{tikzpicture}
```

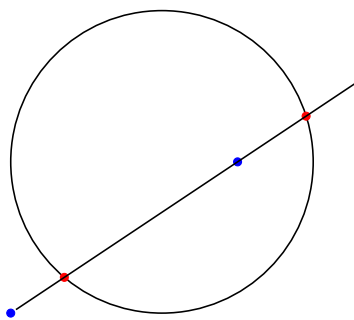
20.2.5 半径计算例 1

使用`pgfmath`模块的`\pgfmathsetmacro`命令进行计算。

半径的计算是提前完成的，不是在交点计算命令中计算的。The radius measurement may be the result of a calculation that is not done within the intersection macro, but before. 可以有多种方式计算长度，一种方式是使用`pgfmath`模块的`\pgfmathsetmacro`命令进行计算。某些情况下，这种计算精度不足，如 $0.0002 \div 0.0001$ 的结果是 1.98，此时，如使用 `xfp` 宏包进行计算其结果为 2。

20.2.6 半径计算例 2

使用`xfp`宏包的`\fpeval`命令：

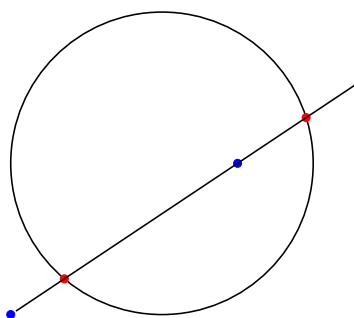


```
\begin{tikzpicture}
\tkzDefPoint(2,2){A}
\tkzDefPoint(5,4){B}
\tkzDefPoint(4,4){O}
\edef\tkzLen{\fpeval{0.0002/0.0001}}
\tkzDrawCircle[R](O,\tkzLen cm)
\tkzInterLC[R](A,B)(O,\tkzLen cm)
\tkzGetPoints{I}{J}
\tkzDrawPoints[color=blue](A,B)
\tkzDrawPoints[color=red](I,J)
\tkzDrawLine(I,J)
\end{tikzpicture}
```

20.2.7 半径计算例 3

使用 \TeX 的 `\tkzLength` 命令计算。

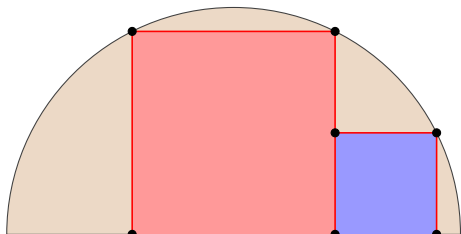
用 `\newdimen` 命令定义一尺寸。当然，用的是 \TeX 进行计算。



```
\begin{tikzpicture}
\tkzDefPoints{2/2/A,5/4/B,4/4/O}
\tkzLength=2cm
\tkzDrawCircle[R](O,\tkzLength)
\tkzInterLC[R](A,B)(O,\tkzLength)
\tkzGetPoints{I}{J}
\tkzDrawPoints[color=blue](A,B)
\tkzDrawPoints[color=red](I,J)
\tkzDrawLine(I,J)
\end{tikzpicture}
```

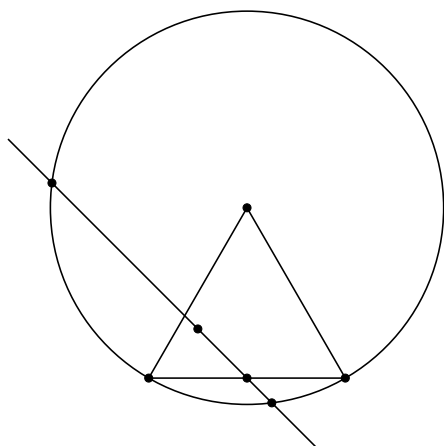
20.2.8 半圆中的矩形

在一个半圆内同时画两个正方形是比较困难的，并且需要通过圆的半径计算正方形的边长。



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,8/0/B,4/0/I}
\tkzDefSquare(A,B)\tkzGetPoints{C}{D}
\tkzInterLC(I,C)(I,B)\tkzGetPoints{E'}{E}
\tkzInterLC(I,D)(I,B)\tkzGetPoints{F'}{F}
\tkzDefPointsBy[projection = onto A--B](E,F){H,G}
\tkzDefPointsBy[symmetry = center H](I){J}
\tkzDefSquare(H,J)\tkzGetPoints{K}{L}
\tkzDrawSector[fill=brown!30](I,B)(A)
\tkzFillPolygon[color=red!40](H,E,F,G)
\tkzFillPolygon[color=blue!40](H,J,K,L)
\tkzDrawPolySeg[color=red](H,E,F,G)
\tkzDrawPolySeg[color=red](J,K,L)
\tkzDrawPoints(E,G,H,F,J,K,L)
\end{tikzpicture}
```


20.2.9 "with nodes"选项



```
\begin{tikzpicture}[scale=.65]
\tkzDefPoints{0/0/A,4/0/B,1/1/D,2/0/E}
\tkzDefTriangle[equilateral](A,B)
\tkzGetPoint{C}
\tkzDrawCircle(C,A)
\tkzInterLC[with nodes](D,E)(C,A,B)
\tkzGetPoints{F}{G}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,...,G)
\tkzDrawLine(F,G)
\end{tikzpicture}
```

20.3 两个圆的交点

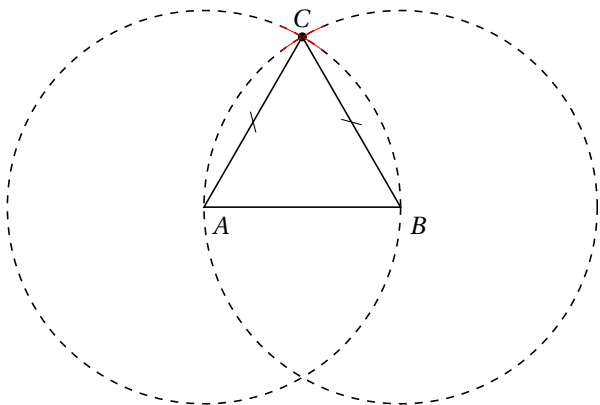
通常，两个圆是由圆心和一点确定的，但也可以用R指定半径点。

`\tkzInterCC[< 命令选项>](<O,A>)(<O',A'>)` 或 `(<O,r>)(<O',r'>)` 或 `(<O,A,B>)(<O',C,D>)`

选项	默认值	含义
N	N	OA 和 $O'A'$ 是半径， O 和 O' 是圆心
R	N	r 和 r' 是半径
with nodes	N	在 $(A,A,C)(C,B,F)$ 中 AC 和 BF 是半径

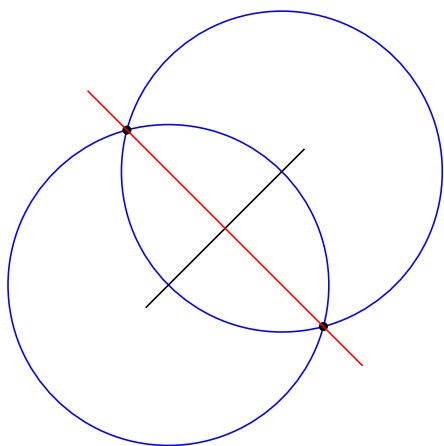
该命令定义了 O 和 O' 两个圆的交点 I 和 J ，如果两个圆没有共点，则返回错误。也可以直接使用 `\tkzInterCCN` 命令和 `\tkzInterCCR` 命令进行计算。

20.3.1 构造等边三角形



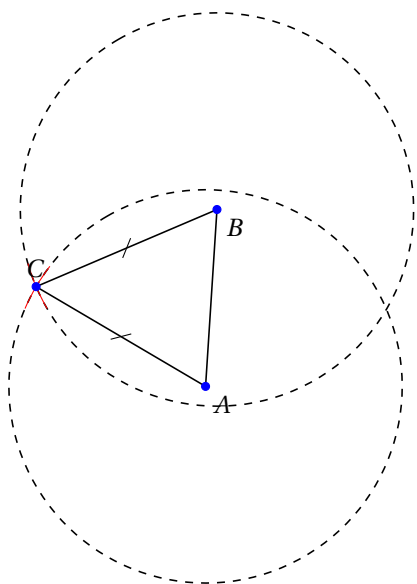
```
\begin{tikzpicture}[trim left=-1cm,scale=0.65]
\tkzDefPoint(1,1){A}
\tkzDefPoint(5,1){B}
\tkzInterCC(A,B)(B,A)\tkzGetPoints{C}{D}
\tkzDrawPoint[color=black](C)
\tkzDrawCircle[dashed](A,B)
\tkzDrawCircle[dashed](B,A)
\tkzCompass[color=red](A,C)
\tkzCompass[color=red](B,C)
\tkzDrawPolygon(A,B,C)
\tkzMarkSegments[mark=s|](A,C B,C)
\tkzLabelPoints[] (A,B)
\tkzLabelPoint[above](C){C}
\end{tikzpicture}
```

20.3.2 求中点



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(2,2){B}
  \tkzDrawCircle[color=blue](B,A)
  \tkzDrawCircle[color=blue](A,B)
  \tkzInterCC(B,A)(A,B)\tkzGetPoints{M}{N}
  \tkzDrawLine(A,B)
  \tkzDrawPoints(M,N)
  \tkzDrawLine[color=red](M,N)
\end{tikzpicture}
```

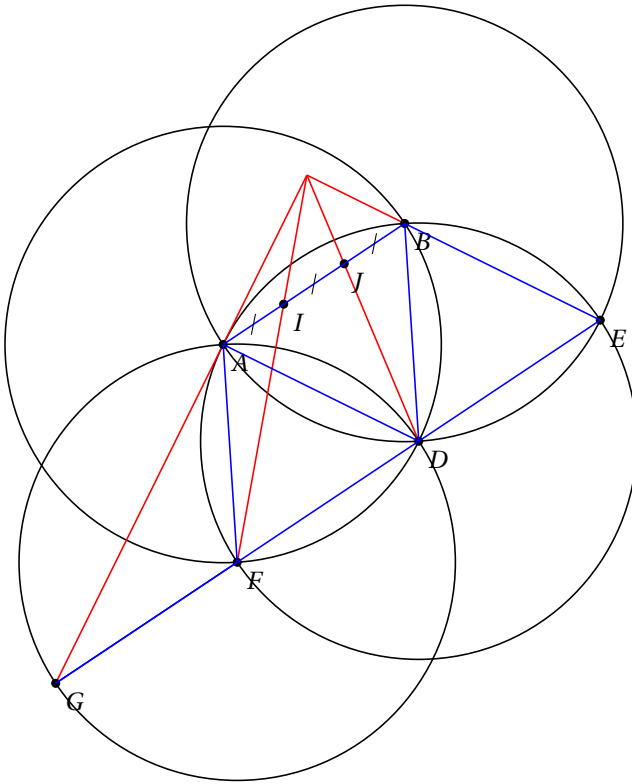
20.3.3 求等腰三角形



```
\begin{tikzpicture}[rotate=120,scale=0.65]
  \tkzDefPoint(1,2){A}
  \tkzDefPoint(4,0){B}
  \tkzInterCC[R](A,4cm)(B,4cm)
  \tkzGetPoints{C}{D}
  \tkzDrawCircle[R,dashed](A,4 cm)
  \tkzDrawCircle[R,dashed](B,4 cm)
  \tkzCompass[color=red](A,C)
  \tkzCompass[color=red](B,C)
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints[color=blue](A,B,C)
  \tkzMarkSegments[mark=s](A,C B,C)
  \tkzLabelPoints[] (A,B)
  \tkzLabelPoint[above](C){C$}
\end{tikzpicture}
```

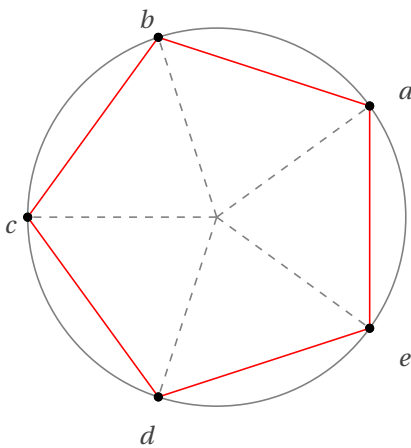
20.3.4 三等分线段

该例与尺规作图求线段的三等分操作一至。



```
\begin{tikzpicture}[scale=0.80]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,2){B}
\tkzInterCC(A,B)(B,A)
\tkzGetPoints{C}{D}
\tkzInterCC(D,B)(B,A)
\tkzGetPoints{A}{E}
\tkzInterCC(D,B)(A,B)
\tkzGetPoints{F}{B}
\tkzInterLC(E,F)(F,A)
\tkzGetPoints{D}{G}
\tkzInterLL(A,G)(B,E)
\tkzGetPoint{O}
\tkzInterLL(O,D)(A,B)
\tkzGetPoint{J}
\tkzInterLL(O,F)(A,B)
\tkzGetPoint{I}
\tkzDrawCircle(D,A)
\tkzDrawCircle(A,B)
\tkzDrawCircle(B,A)
\tkzDrawCircle(F,A)
\tkzDrawSegments[color=red](O,G
O,B O,D O,F)
\tkzDrawPoints(A,B,D,E,F,G,I,J)
\tkzLabelPoints(A,B,D,E,F,G,I,J)
\tkzDrawSegments[blue](A,B B,D A,D
A,F F,G E,G B,E)
\tkzMarkSegments[mark=s|](A,I I,J J,B)
\end{tikzpicture}
```

20.3.5 with nodes 选项



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/a,0/5/B,5/0/C}
\tkzDefPoint(54:5){F}
\tkzDrawCircle[color=gray](A,C)
\tkzInterCC[with nodes](A,A,C)(C,B,F)
\tkzGetPoints{a}{e}
\tkzInterCC(A,C)(a,e) \tkzGetFirstPoint{b}
\tkzInterCC(A,C)(b,a) \tkzGetFirstPoint{c}
\tkzInterCC(A,C)(c,b) \tkzGetFirstPoint{d}
\tkzDrawPoints(a,b,c,d,e)
\tkzDrawPolygon[color=red](a,b,c,d,e)
\foreach \vertex/\num in {a/36,b/108,c/180,
d/252,e/324}{%
\tkzDrawPoint(\vertex)
\tkzLabelPoint[label=\num:$\vertex$](\vertex){}
\tkzDrawSegment[color=gray,style=dashed](A,\vertex)
}
\end{tikzpicture}
```

21 角度

21.1 颜色与角度：填充

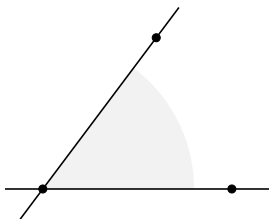
`\tkzFillAngle[<命令选项>](A,O,B)`

O 是角度顶点， OA 和 OB 两条边，注意角度方向由点的顺序决定。

选项	默认值	含义
size	1 cm	着色扇形的半径

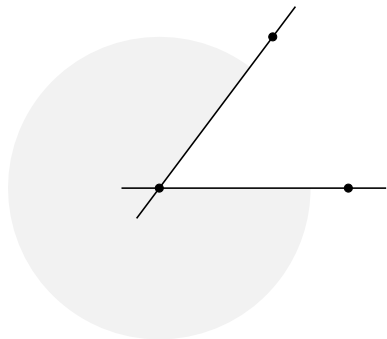
当然，可以使用所有有效的 TikZ 样式，如 fill 和 shade 等。

21.1.1 size 选项

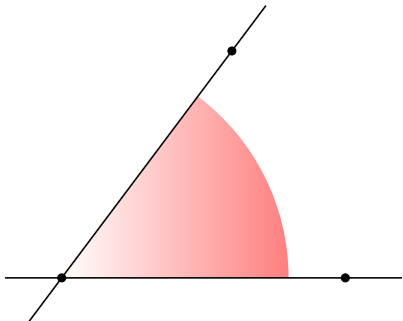


```
\begin{tikzpicture}
\tkzInit
\tkzDefPoints{0/0/0,2.5/0/A,1.5/2/B}
\tkzFillAngle[size=2cm, fill=gray!10](A,O,B)
\tkzDrawLines(O,A O,B)
\tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

21.1.2 改变点的顺序



```
\begin{tikzpicture}
\tkzInit
\tkzDefPoints{0/0/0,2.5/0/A,1.5/2/B}
\tkzFillAngle[size=2cm,fill=gray!10](B,O,A)
\tkzDrawLines(O,A O,B)
\tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

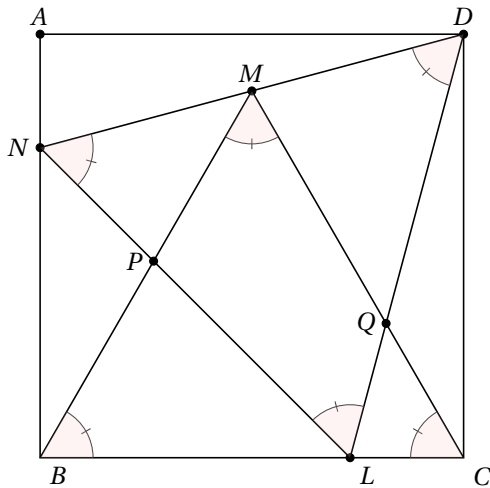


```
\begin{tikzpicture}[scale=0.75]
\tkzInit
\tkzDefPoints{0/0/0,5/0/A,3/4/B}
% Don't forget {} to get, () to use
\tkzFillAngle[size=4cm,left color=white,
right color=red!50](A,O,B)
\tkzDrawLines(O,A O,B)
\tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

`\tkzFillAngles[(< 命令选项>)(\langle A,O,B\rangle)(\langle A',O',B'\rangle)` 等

绘制多个角度。

21.1.3 多个角度



```
\begin{tikzpicture}[scale=0.7]
  \tkzDefPoint(0,0){B}
  \tkzDefPoint(8,0){C}
  \tkzDefPoint(0,8){A}
  \tkzDefPoint(8,8){D}
  \tkzDrawPolygon(B,C,D,A)
  \tkzDefTriangle[equilateral](B,C)
  \tkzGetPoint{M}
  \tkzInterLL(D,M)(A,B) \tkzGetPoint{N}
  \tkzDefPointBy[rotation=center N angle -60](D)
  \tkzGetPoint{L}
  \tkzInterLL(N,L)(M,B) \tkzGetPoint{P}
  \tkzInterLL(M,C)(D,L) \tkzGetPoint{Q}
  \tkzDrawSegments(D,N N,L L,D B,M M,C)
  \tkzDrawPoints(L,N,P,Q,M,A,D)
  \tkzLabelPoints[left](N,P,Q)
  \tkzLabelPoints[above](M,A,D)
  \tkzLabelPoints(L,B,C)
  \tkzMarkAngles(C,B,M B,M,C M,C,B
    D,L,N L,N,D N,D,L)
  \tkzFillAngles[fill=red!20,opacity=.2](C,B,M
    B,M,C M,C,B D,L,N L,N,D N,D,L)
\end{tikzpicture}
```

21.2 角度标记

有丰富的选项可以使绘图更为精细，为 TikZ 新添加的标记定义在 `tkz-lib-marks.tex` 文件中，主要的标记有：

`|`, `||`, `|||`, `z`, `s`, `x`, `o`, `oo`

它们的定义如下 Their definitions are as follows

```
\pgfdeclareplotmark{|||}
  %double bar
{%
  \pgfpathmoveto{\pgfqpoint{2\pgflinewidth}{\pgfplotmarksizex}}
  \pgfpathlineto{\pgfqpoint{2\pgflinewidth}{-\pgfplotmarksizex}}
  \pgfpathmoveto{\pgfqpoint{-2\pgflinewidth}{\pgfplotmarksizex}}
  \pgfpathlineto{\pgfqpoint{-2\pgflinewidth}{-\pgfplotmarksizex}}
  \pgfusepathqstroke
}
```

```

%triple bar
\pgfdeclareplotmark{|||}
{%
  \pgfpathmoveto{\pgfqpoint{0 pt}{\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{0 pt}{-\pgfplotmarksizesize}}
  \pgfpathmoveto{\pgfqpoint{-3\pgflinewidth}{\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{-3\pgflinewidth}{-\pgfplotmarksizesize}}
  \pgfpathmoveto{\pgfqpoint{3\pgflinewidth}{\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{3\pgflinewidth}{-\pgfplotmarksizesize}}
  \pgfusepathqstroke
}

% An bar slant
\pgfdeclareplotmark{s|}
{%
  \pgfpathmoveto{\pgfqpoint{- .70710678\pgfplotmarksizesize}%
    {- .70710678\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{.70710678\pgfplotmarksizesize}%
    {.70710678\pgfplotmarksizesize}}
  \pgfusepathqstroke
}

% An double bar slant
\pgfdeclareplotmark{s||}
{%
  \pgfpathmoveto{\pgfqpoint{-0.75\pgfplotmarksizesize}{-\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{0.25\pgfplotmarksizesize}{\pgfplotmarksizesize}}
  \pgfpathmoveto{\pgfqpoint{0\pgfplotmarksizesize}{-\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{1\pgfplotmarksizesize}{\pgfplotmarksizesize}}
  \pgfusepathqstroke
}

% z
\pgfdeclareplotmark{z}
{%
  \pgfpathmoveto{\pgfqpoint{0.75\pgfplotmarksizesize}{-\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{-0.75\pgfplotmarksizesize}{-\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{0.75\pgfplotmarksizesize}{\pgfplotmarksizesize}}
  \pgfpathlineto{\pgfqpoint{-0.75\pgfplotmarksizesize}{\pgfplotmarksizesize}}
  \pgfusepathqstroke
}

% s
\pgfdeclareplotmark{s}
{%
  \pgfpathmoveto{\pgfqpoint{0pt}{0pt}}
  \pgfpathcurveto
    {\pgfpoint{0pt}{0pt}}
    {\pgfpoint{-\pgfplotmarksizesize}{\pgfplotmarksizesize}}
    {\pgfpoint{\pgfplotmarksizesize}{\pgfplotmarksizesize}}
  \pgfpathmoveto{\pgfqpoint{0pt}{0pt}}
  \pgfpathcurveto
    {\pgfpoint{0pt}{0pt}}
    {\pgfpoint{\pgfplotmarksizesize}{-\pgfplotmarksizesize}}
    {\pgfpoint{-\pgfplotmarksizesize}{-\pgfplotmarksizesize}}
  \pgfusepathqstroke
}

% infinity
\pgfdeclareplotmark{oo}
{%
  \pgfpathmoveto{\pgfqpoint{0pt}{0pt}}

```

```

\pgfpathcurveto
  {\pgfpoint{0pt}{0pt}}
  {\pgfpoint{.5\pgfplotmarksizesize}{1\pgfplotmarksizesize}}
  {\pgfpoint{\pgfplotmarksizesize}{0pt}}
\pgfpathmoveto{\pgfpoint{0pt}{0pt}}
\pgfpathcurveto
  {\pgfpoint{0pt}{0pt}}
  {\pgfpoint{-.5\pgfplotmarksizesize}{1\pgfplotmarksizesize}}
  {\pgfpoint{-\pgfplotmarksizesize}{0pt}}
\pgfpathmoveto{\pgfpoint{0pt}{0pt}}
\pgfpathcurveto
  {\pgfpoint{0pt}{0pt}}
  {\pgfpoint{.5\pgfplotmarksizesize}{-1\pgfplotmarksizesize}}
  {\pgfpoint{\pgfplotmarksizesize}{0pt}}
\pgfpathmoveto{\pgfpoint{0pt}{0pt}}
\pgfpathcurveto
  {\pgfpoint{0pt}{0pt}}
  {\pgfpoint{-.5\pgfplotmarksizesize}{-1\pgfplotmarksizesize}}
  {\pgfpoint{-\pgfplotmarksizesize}{0pt}}
\pgfusepathqstroke
}

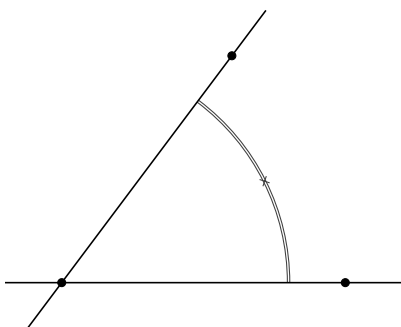
```

`\tkzMarkAngle[(< 命令选项>)](<A,0,B>)`

O 是顶点, 注意参数需要随选项而变化。可以使用任意一种标记, 甚至可以绘制一个圆弧, 然后为该圆弧添加一个标记。圆弧的样式通过 `arc` 选项指定, 圆弧的半径由 `mksize` 选项指定。当然, 圆弧也可以着色。

选项	默认值	含义
<code>arc</code>	1	选择单线、双线或三线样式
<code>size</code>	1 cm	圆弧半径
<code>mark</code>	无	标记类型
<code>mksize</code>	4pt	标记符号尺寸
<code>mkcolor</code>	black	标记符号颜色
<code>mkpos</code>	0.5	标记位置

21.2.1 mark = x 选项

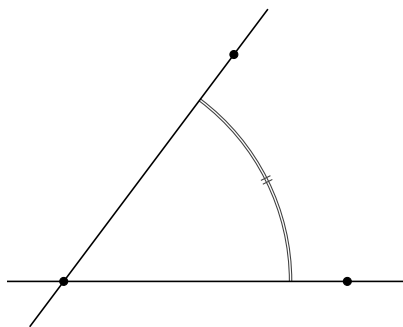


```

\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{O/0/0,5/0/A,3/4/B}
  \tkzMarkAngle[size = 4cm,mark = x,
    arc=11,mkcolor = red](A,0,B)
  \tkzDrawLines(O,A O,B)
  \tkzDrawPoints(O,A,B)
\end{tikzpicture}

```

21.2.2 mark = || 选项



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{O/0/0,5/0/A,3/4/B}
  \tkzMarkAngle[size = 4cm,mark = ||,
    arc=ll,mkcolor = red](A,O,B)
  \tkzDrawLines(O,A O,B)
  \tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

`\tkzMarkAngles[< 命令选项>](A,O,B)(A',O',B')` 等

对于具有相同选项的标记，可以一次标记多个角度。

21.3 为角度添加标注

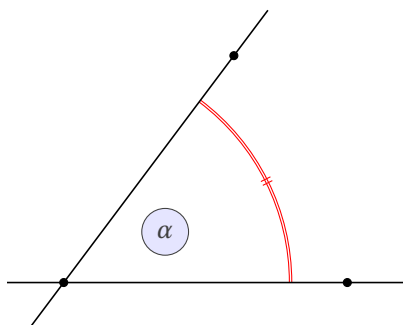
`\tkzLabelAngle[< 命令选项>](A,O,B)`

该命令只有一个 `dist` 选项 (带或不带单位)，该选项可以被 TikZ 的选项 (不带单位) 替代，默认情况下，其单位是 `cm`。

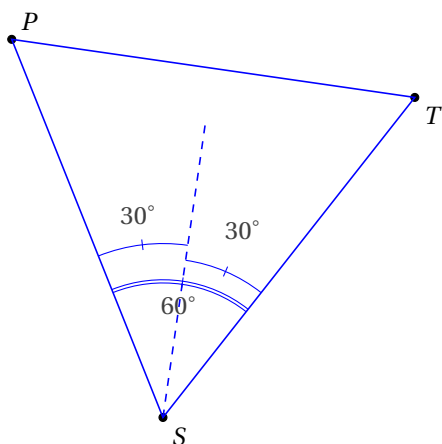
选项	默认值	含义
<code>pos</code>	1	或是 <code>dist</code> ，用于控制标注的距离

可以使用 TikZ 的 `rotate`、`shift`、`below` 等选项调整标注的位置。

21.3.1 pos 选项



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{O/0/0,5/0/A,3/4/B}
  \tkzMarkAngle[size = 4cm,mark = ||,
    arc=ll,color = red](A,O,B)%
  \tkzDrawLines(O,A O,B)
  \tkzDrawPoints(O,A,B)
  \tkzLabelAngle[pos=2,draw,circle,
    fill=blue!10](A,O,B){$\alpha$}
\end{tikzpicture}
```

```
\begin{tikzpicture}[rotate=30]
\tkzDefPoint(2,1){S}
\tkzDefPoint(7,3){T}
\tkzDefPointBy[rotation=center S angle 60](T)
\tkzGetPoint{P}
\tkzDefLine[bisector, normed](T,S,P)
\tkzGetPoint{s}
\tkzDrawPoints(S,T,P)
\tkzDrawPolygon[color=blue](S,T,P)
\tkzDrawLine[dashed,color=blue,add=0 and 3](S,s)
\tkzLabelPoint[above right](P){$P$}
\tkzLabelPoints(S,T)
\tkzMarkAngle[size = 1.8cm,mark = |,arc=ll,
               color = blue](T,S,P)
\tkzMarkAngle[size = 2.1cm,mark = |,arc=l,
               color = blue](T,S,s)
\tkzMarkAngle[size = 2.3cm,mark = |,arc=l,
               color = blue](s,S,P)
\tkzLabelAngle[pos = 1.5](T,S,P){$60^\circ$}%
\tkzLabelAngles[pos = 2.7](T,S,s s,S,P){$30^\circ$}%
\end{tikzpicture}
```

`\tkzLabelAngles[< 命令选项>](A,O,B)(A',O',B')` 等

当选项相同时，可以用该命令为多个角度添加标注。

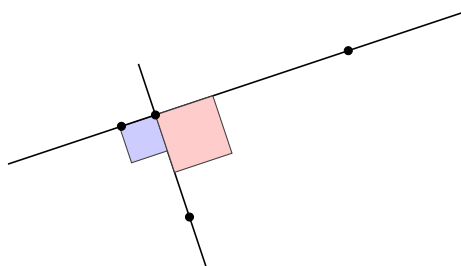
21.4 直角标记

`\tkzMarkRightAngle[< 命令选项>](A,O,B)`

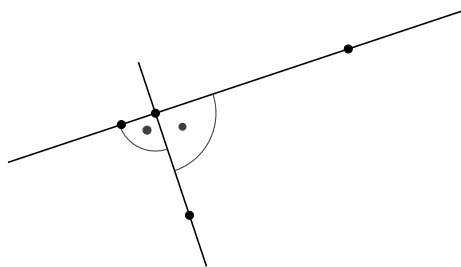
`german`选项用于改变样式，`size`选项用于改变尺寸。

选项	默认值	含义
<code>german</code>	<code>normal</code>	带内点的圆弧
<code>size</code>	<code>0.2</code>	标记边的尺寸

21.4.1 直角标记示例

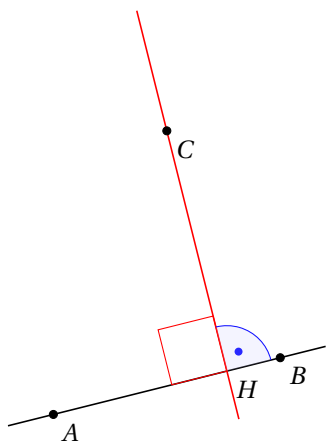


```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
\tkzDefPointBy[projection = onto B--A](P)
\tkzGetPoint{H}
\tkzDrawLines[add=.5 and .5](P,H)
\tkzMarkRightAngle[fill=blue!20,size=.5,draw](A,H,P)
\tkzDrawLines[add=.5 and .5](A,B)
\tkzMarkRightAngle[fill=red!20,size=.8](B,H,P)
\tkzDrawPoints(A,B,P,H)
\end{tikzpicture}
```

21.4.2 使用 `german` 样式添加直角标记

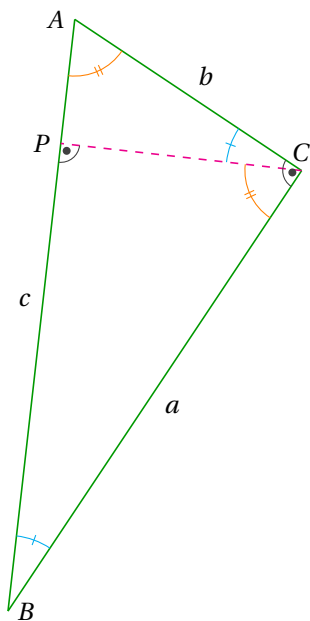
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
\tkzDefPointBy[projection = onto B--A](P)
\tkzGetPoint{H}
\tkzDrawLines[add=.5 and .5](P,H)
\tkzMarkRightAngle[german,size=.5,draw](A,H,P)
\tkzDrawPoints[](A,B,P,H)
\tkzDrawLines[add=.5 and .5,fill=blue!20](A,B)
\tkzMarkRightAngle[german,size=.8](P,H,B)
\end{tikzpicture}
```

21.4.3 混合样式



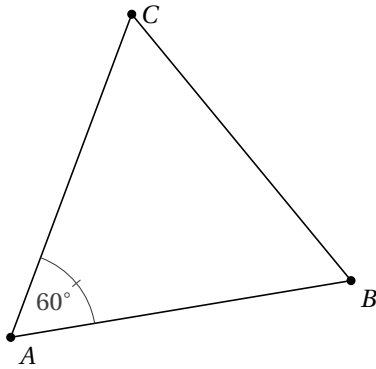
```
\begin{tikzpicture}[scale=0.75]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,1){B}
\tkzDefPoint(2,5){C}
\tkzDefPointBy[projection=onto B--A](C)
\tkzGetPoint{H}
\tkzDrawLine(A,B)
\tkzDrawLine[add = .5 and .2,color=red](C,H)
\tkzMarkRightAngle[,size=1,color=red](C,H,A)
\tkzMarkRightAngle[german,size=.8,color=blue](B,H,C)
\tkzFillAngle[opacity=.2,fill=blue!20,size=.8](B,H,C)
\tkzLabelPoints(A,B,C,H)
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

21.4.4 完整示例



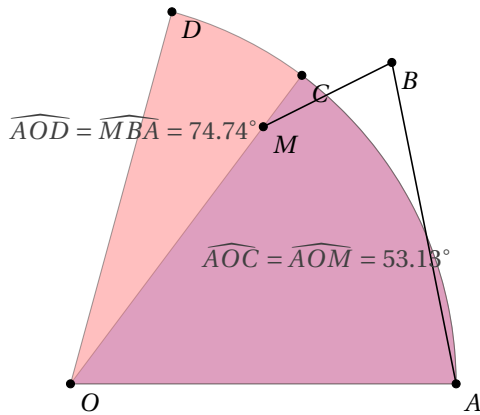
```
\begin{tikzpicture}[rotate=-90]
\tkzDefPoint(0,1){A}
\tkzDefPoint(2,4){C}
\tkzDefPointWith[orthogonal normed,K=7](C,A)
\tkzGetPoint{B}
\tkzDrawSegment[green!60!black](A,C)
\tkzDrawSegment[green!60!black](C,B)
\tkzDrawSegment[green!60!black](B,A)
\tkzDrawLine[altitude,dashed,color=magenta](B,C,A)
\tkzGetPoint{P}
\tkzLabelPoint[left](A){$A$}
\tkzLabelPoint[right](B){$B$}
\tkzLabelPoint[above](C){$C$}
\tkzLabelPoint[left](P){$P$}
\tkzLabelSegment[auto](B,A){$c$}
\tkzLabelSegment[auto,swap](B,C){$a$}
\tkzLabelSegment[auto,swap](C,A){$b$}
\tkzMarkAngle[size=1cm,color=cyan,mark=|](C,B,A)
\tkzMarkAngle[size=0.75cm,color=orange,mark=||](P,C,B)
\tkzMarkAngle[size=0.75cm,color=orange,mark=||](B,A,C)
\tkzMarkRightAngle[german](A,C,B)
\tkzMarkRightAngle[german](B,P,C)
\end{tikzpicture}
```


22.3.1 角度测量



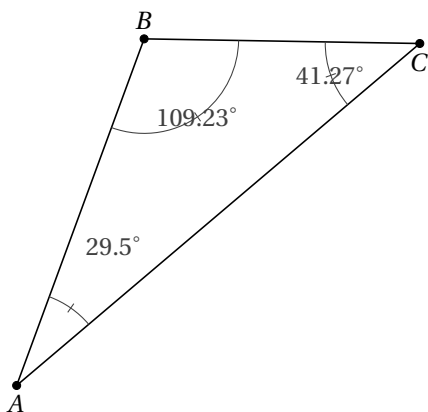
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(-1,1){A} \tkzDefPoint(5,2){B}
  \tkzDefEquilateral(A,B)
  \tkzGetPoint{C}
  \tkzDrawPolygon(A,B,C)
  \tkzFindAngle(B,A,C)
  \tkzGetAngle{angleBAC}
  \edef\angleBAC{\fpeval{round(\angleBAC)}}
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B)
  \tkzLabelPoint[right](C){C}
  \tkzLabelAngle(B,A,C){\angleBAC^\circ}
  \tkzMarkAngle[size=1.5cm](B,A,C)
\end{tikzpicture}
```

22.4 \tkzFindAngle命令



```
\begin{tikzpicture}[scale=0.85]
  \tkzInit[xmin=-1,ymin=-1,xmax=7,ymax=7]
  \tkzClip
  \tkzDefPoint(0,0){O} \tkzDefPoint(6,0){A}
  \tkzDefPoint(5,5){B} \tkzDefPoint(3,4){M}
  \tkzFindAngle(A,O,M) \tkzGetAngle{an}
  \tkzDefPointBy[rotation=center O angle \an](A)
  \tkzGetPoint{C}
  \tkzDrawSector[fill = blue!50,opacity=.5](O,A)(C)
  \tkzFindAngle(M,B,A) \tkzGetAngle{am}
  \tkzDefPointBy[rotation = center O angle \am](A)
  \tkzGetPoint{D}
  \tkzDrawSector[fill = red!50,opacity = .5](O,A)(D)
  \tkzDrawPoints(O,A,B,M,C,D)
  \tkzLabelPoints(O,A,B,M,C,D)
  \edef\an{\fpeval{round(\an,2)}}\edef\am{\fpeval{round(\am,2)}}
  \tkzDrawSegments(M,B B,A)
  \tkzText(4,2){\widehat{AOC}=\widehat{AOM}=\an^\circ}
  \tkzText(1,4){\widehat{AOD}=\widehat{MBA}=\am^\circ}
\end{tikzpicture}
```

22.4.1 确定三角形的三个角



```
\begin{tikzpicture}[scale=1.25,rotate=30]
\tkzDefPoints{0.5/1.5/A, 3.5/4/B, 6/2.5/C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints[below](A,C)
\tkzLabelPoints[above](B)
\tkzMarkAngle[size=1cm](B,C,A)
\tkzFindAngle(B,C,A) \tkzGetAngle{angleBCA}
\edef\angleBCA{\fpeval{round(\angleBCA,2)}}
\tkzLabelAngle[pos = 1](B,C,A){$\angle BCA^{\circ}$}
\tkzMarkAngle[size=1cm](C,A,B)
\tkzFindAngle(C,A,B) \tkzGetAngle{angleBAC}
\edef\angleBAC{\fpeval{round(\angleBAC,2)}}
\tkzLabelAngle[pos = 1.8](C,A,B){$\angle BAC^{\circ}$}
\tkzMarkAngle[size=1cm](A,B,C)
\tkzFindAngle(A,B,C) \tkzGetAngle{angleABC}
\edef\angleABC{\fpeval{round(\angleABC,2)}}
\tkzLabelAngle[pos = 1](A,B,C){$\angle ABC^{\circ}$}
\end{tikzpicture}
```

22.5 确定斜率

斜率由直线上两个点确定，该命令不检测其存在性。

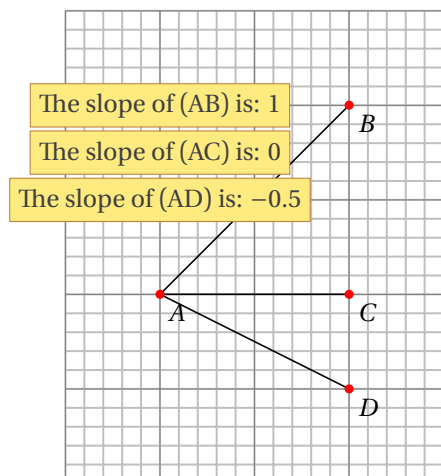
`\tkzFindSlope(<pt1,pt2>){<宏名称>}`

斜率保存在宏中。

参数	样例	说明
(pt1,pt2)pt3 进行计算	<code>\tkzFindSlope(A,B){slope}</code>	<code>\slope</code> 斜率通过 $\frac{y_B - y_A}{x_B - x_A}$



当 $x_B = x_A$ 时，没有斜率。



```
\begin{tikzpicture}[scale=1.25]
\tkzInit[xmax=4,ymax=5]\tkzGrid[sub]
\tkzDefPoint(1,2){A} \tkzDefPoint(3,4){B}
\tkzDefPoint(3,2){C} \tkzDefPoint(3,1){D}
\tkzDrawSegments(A,B A,C A,D)
\tkzDrawPoints[color=red](A,B,C,D)
\tkzLabelPoints(A,B,C,D)
\tkzFindSlope(A,B){SAB} \tkzFindSlope(A,C){SAC}
\tkzFindSlope(A,D){SAD}
\pgfkeys{/pgf/number format/.cd,fixed,precision=2}
\tkzText[fill=Gold!50,draw=brown](1,4)%
{The slope of (AB) is: $\pgfmathprintnumber{\SAB}$}
\tkzText[fill=Gold!50,draw=brown](1,3.5)%
{The slope of (AC) is: $\pgfmathprintnumber{\SAC}$}
\tkzText[fill=Gold!50,draw=brown](1,3)%
{The slope of (AD) is: $\pgfmathprintnumber{\SAD}$}
\end{tikzpicture}
```

22.6 直线与横轴夹角计算命令: `\tkzFindSlopeAngle`

结果在-180 度与 +180 度之间。

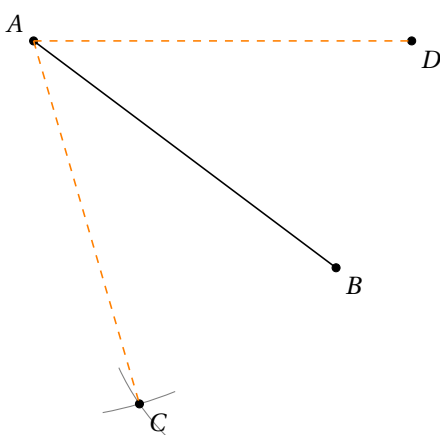
`\tkzFindSlopeAngle(<A,B>)`

计算直线 (AB) 的斜率并将其保存在 `\tkzAngleResult` 中。

参数	样例	说明
(pt1,pt2)	<code>\tkzFindSlopeAngle(A,B)</code>	

使用 `\tkzGetAngle` 可以得到结果, 如果无需再次使用, 可以使用 `\tkzAngleResult` 得到结果。

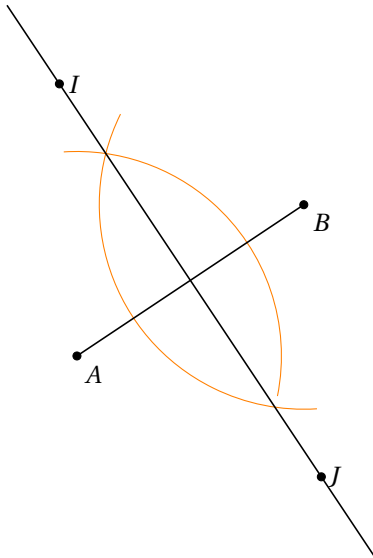
22.6.1 折叠



```
\begin{tikzpicture}
\tkzDefPoint(1,5){A}
\tkzDefPoint(5,2){B}
\tkzDrawSegment(A,B)
\tkzFindSlopeAngle(A,B)
\tkzGetAngle{\tkzang}
\tkzDefPointBy[rotation=center A angle \tkzang](B)
\tkzGetPoint{C}
\tkzDefPointBy[rotation=center A angle - \tkzang](B)
\tkzGetPoint{D}
\tkzCompass[orange,length=1](A,C)
\tkzCompass[orange,delta=10](B,C)
\tkzDrawPoints(A,B,C,D)
\tkzLabelPoints(B,C,D)
\tkzLabelPoints[above left](A)
\tkzDrawSegments[style=dashed,color=orange](A,C A,D)
\end{tikzpicture}
```

22.6.2 \tkzFindSlopeAngle 示例

这是计算中点的另一个实例。



```
\begin{tikzpicture}
\tkzInit
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,2){B}
\tkzDefLine[mediator](A,B)
\tkzGetPoints{I}{J}
\tkzCalcLength[cm](A,B)
\tkzGetLength{dAB}
\tkzFindSlopeAngle(A,B)
\tkzGetAngle{tkzangle}
\begin{scope}[rotate=\tkzangle]
\tikzset{arc/.style={color=gray,delta=10}}
\tkzDrawArc[orange,R,arc](B,3/4*\dAB)(120,240)
\tkzDrawArc[orange,R,arc](A,3/4*\dAB)(-45,60)
\tkzDrawLine(I,J)
\tkzDrawSegment(A,B)
\end{scope}
\tkzDrawPoints(A,B,I,J)
\tkzLabelPoints(A,B) \tkzLabelPoints[right](I,J)
\end{tikzpicture}
```

23 扇形

23.1 \tkzDrawSector命令

 注意参数需要根据选项变化。

\tkzDrawSector[< 命令选项>](<O,...>)(<...>)		
选项	默认值	含义
towards	towards	O 是圆心并且圆弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始并且用角度确定长度
R	towards	给定半径和两个角度
R with nodes	towards	给定半径和两个点

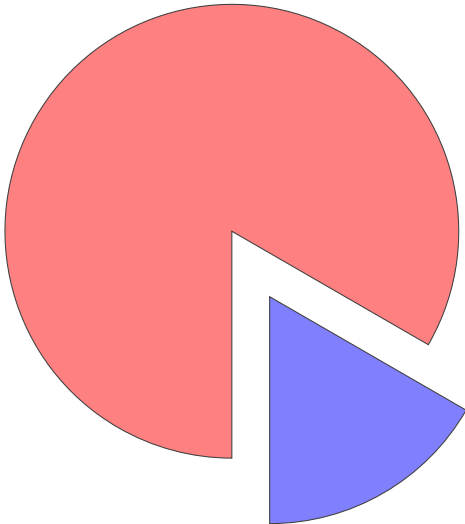
可以使用所有有效的 TikZ 样式。

选项	参数	样例
towards	(<pt,pt>)(<pt>)	\tkzDrawSector(O,A)(B)
rotate	(<pt,pt>)(<an>)	\tkzDrawSector[rotate,color=red](O,A)(90)
R	(<pt,r>)(<an,an>)	\tkzDrawSector[R,color=blue](O,2 cm)(30,90)
R with nodes	(<pt,r>)(<pt,pt>)	\tkzDrawSector[R with nodes](O,2 cm)(A,B)

几个样例

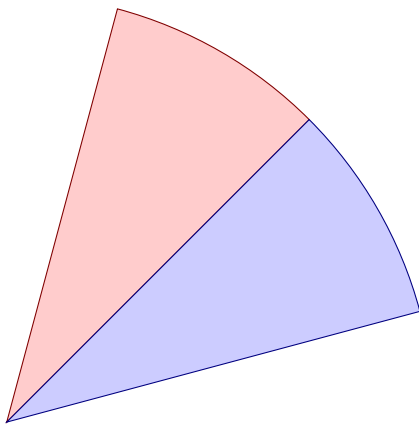
23.1.1 \tkzDrawSector命令和towards选项

不必使用towards，也可以使用fill选项。



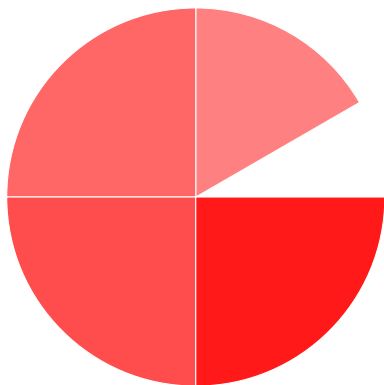
```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,0){O}
\tkzDefPoint(-30:3){A}
\tkzDefPointBy[rotation = center O angle -60](A)
\tkzDrawSector[fill=red!50](O,A)(tkzPointResult)
\begin{scope}[shift={(-60:1cm)}]
\tkzDefPoint(0,0){O}
\tkzDefPoint(-30:3){A}
\tkzDefPointBy[rotation = center O angle -60](A)
\tkzDrawSector[fill=blue!50](O,tkzPointResult)(A)
\end{scope}
\end{tikzpicture}
```


23.1.2 \tkzDrawSector命令和rotate选项



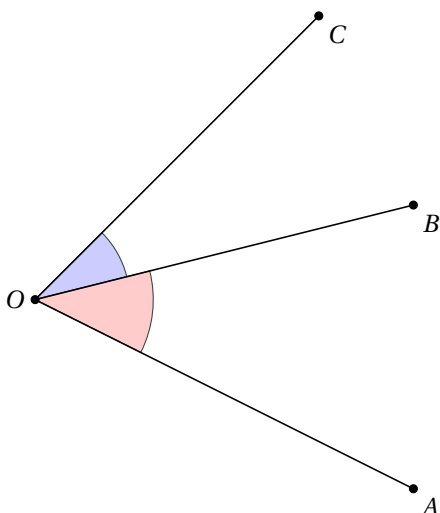
```
\begin{tikzpicture}[scale=2]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,2){A}
\tkzDrawSector[rotate,draw=red!50!black,%
fill=red!20](O,A)(30)
\tkzDrawSector[rotate,draw=blue!50!black,%
fill=blue!20](O,A)(-30)
\end{tikzpicture}
```

23.1.3 \tkzDrawSector命令和R选项



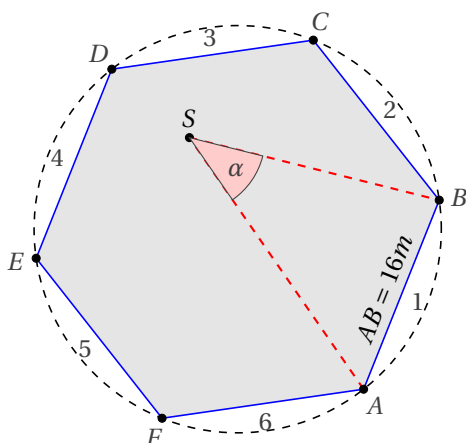
```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,-1){A}
\tkzDrawSector[R,draw=white,%
fill=red!50](O,2cm)(30,90)
\tkzDrawSector[R,draw=white,%
fill=red!60](O,2cm)(90,180)
\tkzDrawSector[R,draw=white,%
fill=red!70](O,2cm)(180,270)
\tkzDrawSector[R,draw=white,%
fill=red!90](O,2cm)(270,360)
\end{tikzpicture}
```

23.1.4 \tkzDrawSector命令和R选项



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){O}
\tkzDefPoint(4,-2){A}
\tkzDefPoint(4,1){B}
\tkzDefPoint(3,3){C}
\tkzDrawSector[R with nodes,%
fill=blue!20](O,1 cm)(B,C)
\tkzDrawSector[R with nodes,%
fill=red!20](O,1.25 cm)(A,B)
\tkzDrawSegments(O,A O,B O,C)
\tkzDrawPoints(O,A,B,C)
\tkzLabelPoints(A,B,C)
\tkzLabelPoints[left](O)
\end{tikzpicture}
```

23.1.5 \tkzDrawSector命令和R with nodes选项



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(-1,-2){A}
\tkzDefPoint(1,3){B}
\tkzDefRegPolygon[side,sides=6](A,B)
\tkzGetPoint{O}
\tkzDrawPolygon[fill=black!10,
draw=blue](P1,P...,P6)
\tkzLabelRegPolygon[sep=1.05](O){A,...,F}
\tkzDrawCircle[dashed](O,A)
\tkzLabelSegment[above,sloped,
midway](A,B){\(\text{A B} = 16\text{m}\)}
\foreach \i [count=\xi from 1] in {2,...,6,1}
{
\tkzDefMidPoint(P\xi,P\i)
\path (O) to [pos=1.1] node {\xi} (tkzPointResult) ;
}
\tkzDefRandPointOn[segment = P3--P5]
\tkzGetPoint{S}
\tkzDrawSegments[thick,dashed,red](A,S S,B)
\tkzDrawPoints(P1,P...,P6,S)
\tkzLabelPoint[left,above](S){\(\text{S}\)}
\tkzDrawSector[R with nodes,fill=red!20](S,2 cm)(A,B)
\tkzLabelAngle[pos=1.5](A,S,B){\(\alpha\)}
\end{tikzpicture}
```

23.2 \tkzFillSector命令

 注意参数需要根据选项变化。

\tkzFillSector[< 命令选项>](<O,...>)(<...>)

选项	默认值	含义
towards	towards	O 是圆心并且圆弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始并且通过角度确定长度
R	towards	给定半径和两个角度
R with nodes	towards	给定半径和两个点

当然，可以使用所有有效的 TikZ 样式。

选项	参数	样例
towards	(<pt,pt>)(<pt>)	\tkzFillSector(O,A)(B)
rotate	(<pt,pt>)(<an>)	\tkzFillSector[rotate,color=red](O,A)(90)
R	(<pt,r>)(<an,an>)	\tkzFillSector[R,color=blue](O,2 cm)(30,90)
R with nodes	(<pt,r>)(<pt,pt>)	\tkzFillSector[R with nodes](O,2 cm)(A,B)

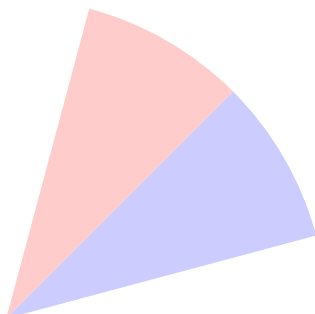
23.2.1 \tkzFillSector命令和towards选项

可以不使用**towards**，并且不绘制轮廓，仅对表面进行着色。




```
\begin{tikzpicture}[scale=.6]
\tkzDefPoint(0,0){O}
\tkzDefPoint(-30:3){A}
\tkzDefPointBy[rotation = center O angle -60](A)
\tkzFillSector[fill=red!50](O,A)(tkzPointResult)
\begin{scope}[shift={(-60:1cm)}]
\tkzDefPoint(0,0){O}
\tkzDefPoint(-30:3){A}
\tkzDefPointBy[rotation = center O angle -60](A)
\tkzFillSector[color=blue!50](O,tkzPointResult)(A)
\end{scope}
\end{tikzpicture}
```

23.2.2 \tkzFillSector命令和rotate选项



```
\begin{tikzpicture}[scale=1.5]
\tkzDefPoint(0,0){O} \tkzDefPoint(2,2){A}
\tkzFillSector[rotate,color=red!20](O,A)(30)
\tkzFillSector[rotate,color=blue!20](O,A)(-30)
\end{tikzpicture}
```

23.3 \tkzClipSector命令

 注意参数需要根据选项变化。

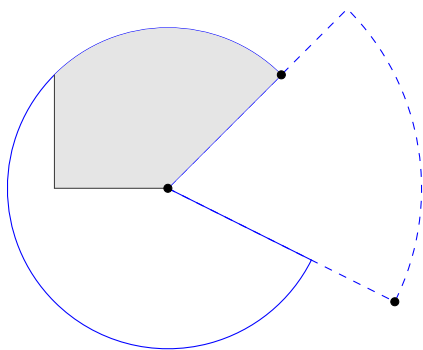
\tkzClipSector[< 命令选项>](<O,...>)(<...>)

选项	默认值	含义
towards	towards	O 是圆心，并且圆弧从 A 开始到 (OB)
rotate	towards	扇形从 A 开始并且由角度确定其幅度
R	towards	给定半径和两个角度

当然，可以使用所有有效的 TikZ 样式。

选项	参数	样例
towards	(<pt,pt>)(<pt>)	\tkzClipSector(O,A)(B)
rotate	(<pt,pt>)(<angle>)	\tkzClipSector[rotate](O,A)(90)
R	(<pt,r>)(<angle 1,angle 2>)	\tkzClipSector[R](O,2 cm)(30,90)

23.3.1 \tkzClipSector 命令



```

\begin{tikzpicture}[scale=1.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzDrawSector[color=blue,dashed](O,A)(B)
  \tkzDrawSector[color=blue](O,B)(A)
  \tkzClipBB
  \begin{scope}
    \tkzClipSector(O,B)(A)
    \draw[fill=gray!20] (-1,0) rectangle (3,3);
  \end{scope}
  \tkzDrawPoints(A,B,O)
\end{tikzpicture}

```

24 圆弧

`\tkzDrawArc[< 命令选项>](<O,...>)(<...>)`

该命令绘制圆心在 O 点的圆弧，根据选项不同，其参数不同。其中，关键是如何确定圆弧的起点和终点。可以指定圆弧起点，或给定圆弧半径。对于后一种情况而言，需要两个角度。或者直接给定角度，或者用与圆心关联 node 确定角度。角度的单位是度。

选项	默认值	含义
towards	towards	O 是圆心，并且圆弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始并且角度确定了长度
R	towards	给定半径和两个角度
R with nodes	towards	给定半径和两个点
angles	towards	给定半径和两个点
delta	0	角度加上两个边

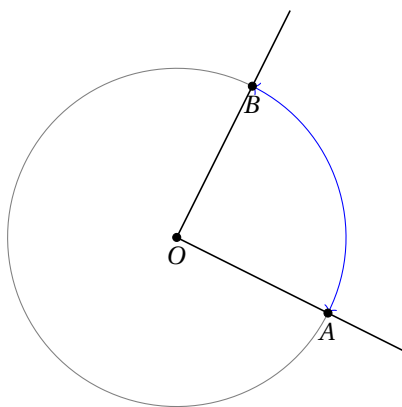
当然，可以使用所有有效的 TikZ 样式。

选项	参数	样例
towards	<code>(<pt,pt>)(<pt>)</code>	<code>\tkzDrawArc[delta=10](O,A)(B)</code>
rotate	<code>(<pt,pt>)(<an>)</code>	<code>\tkzDrawArc[rotate,color=red](O,A)(90)</code>
R	<code>(<pt,r>)(<an,an>)</code>	<code>\tkzDrawArc[R](O,2 cm)(30,90)</code>
R with nodes	<code>(<pt,r>)(<pt,pt>)</code>	<code>\tkzDrawArc[R with nodes](O,2 cm)(A,B)</code>
angles	<code>(<pt,pt>)(<an,an>)</code>	<code>\tkzDrawArc[angles](O,A)(0,90)</code>

以下是几个示例代码：

24.1 towards 选项

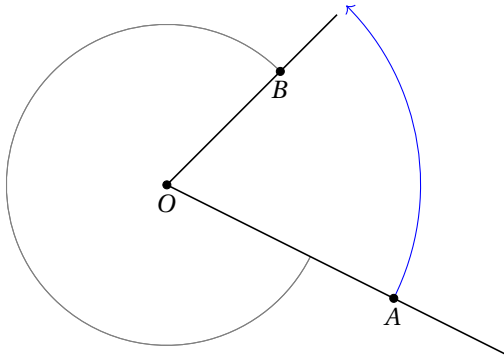
towards 选项不是必须的，第 1 个例子里的圆弧从 A 开始到 B 。当然，从 B 到 A 会得到不同的结果。圆弧凸向由逆时针方向指定的指向确定。



```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,-1){A}
\tkzDefPointBy[rotation= center O angle 90](A)
\tkzGetPoint{B}
\tkzDrawArc[color=blue,<->](O,A)(B)
\tkzDrawArc(O,B)(A)
\tkzDrawLines[add = 0 and .5](O,A O,B)
\tkzDrawPoints(O,A,B)
\tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

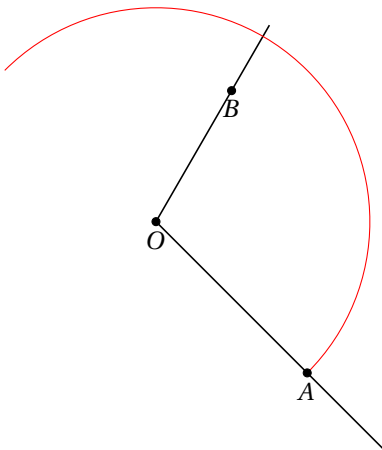
24.2 towards选项

该例中，圆弧从A开始，直到(OB)结束。



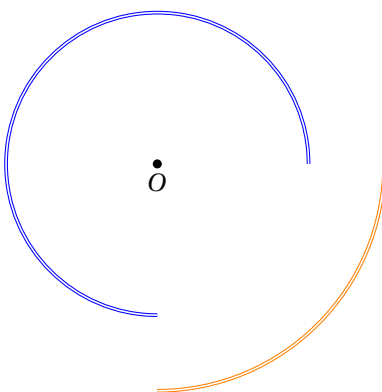
```
\begin{tikzpicture}[scale=1.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzDrawArc[color=blue,->](O,A)(B)
  \tkzDrawArc[color=gray](O,B)(A)
  \tkzDrawArc(O,B)(A)
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

24.3 rotate选项



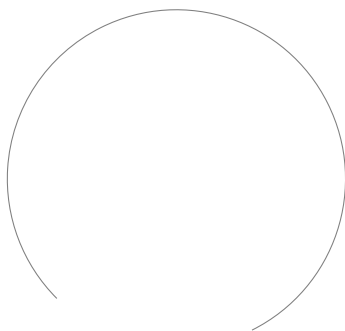
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-2){A}
  \tkzDefPoint(60:2){B}
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawArc[rotate,color=red](O,A)(180)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

24.4 R选项



```
\begin{tikzpicture}
  \tkzDefPoints{O/O/O}
  \tikzset{compass style/.append style={<->}}
  \tkzDrawArc[R,color=orange,double](O,3cm)(270,360)
  \tkzDrawArc[R,color=blue,double](O,2cm)(0,270)
  \tkzDrawPoint(O)
  \tkzLabelPoint[below](O){$O$}
\end{tikzpicture}
```

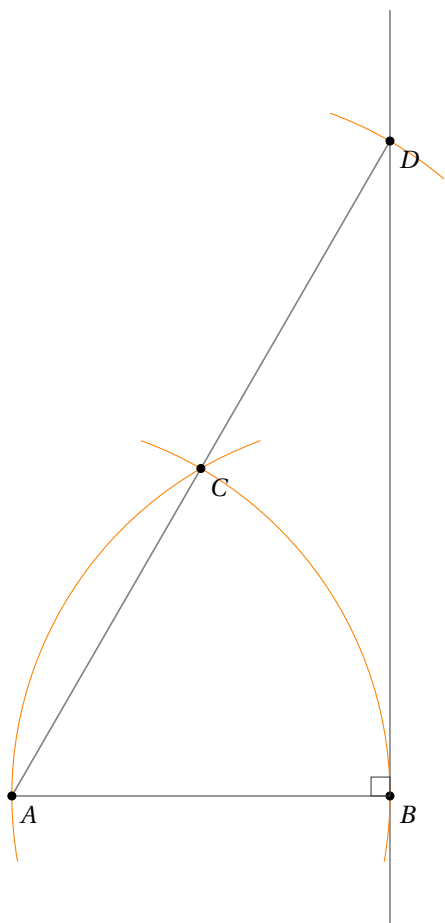
24.5 R with nodes选项



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzCalcLength(B,A)\tkzGetLength{radius}
  \tkzDrawArc[R with nodes](B,\radius pt)(A,O)
\end{tikzpicture}
```

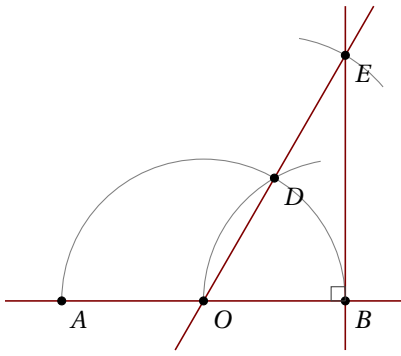
24.6 delta选项

该选项与`\tkzCompass`结果类似，它能够使圆弧覆盖两边。`delta`的单位是度。



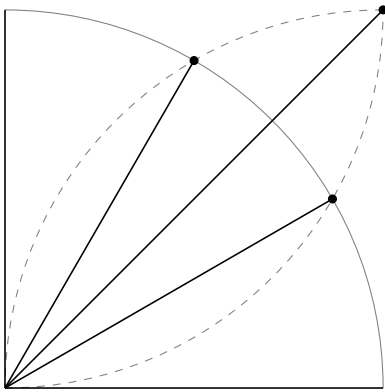
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPointBy[rotation= center A angle 60](B)
  \tkzGetPoint{C}
  \tkzSetUpLine[color=gray]
  \tkzDefPointBy[symmetry= center C](A)
  \tkzGetPoint{D}
  \tkzDrawSegments(A,B A,D)
  \tkzDrawLine(B,D)
  \tkzSetUpCompass[color=orange]
  \tkzDrawArc[orange,delta=10](A,B)(C)
  \tkzDrawArc[orange,delta=10](B,C)(A)
  \tkzDrawArc[orange,delta=10](C,D)(D)
  \tkzDrawPoints(A,B,C,D)
  \tkzLabelPoints(A,B,C,D)
  \tkzMarkRightAngle(D,B,A)
\end{tikzpicture}
```

24.7 angles选项: 示例 1



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPoint(2.5,0){O}
  \tkzDefPointBy[rotation=center O angle 60](B)
  \tkzGetPoint{D}
  \tkzDefPointBy[symmetry=center D](O)
  \tkzGetPoint{E}
  \tkzSetUpLine[color=Maroon]
  \tkzDrawArc[angles](O,B)(0,180)
  \tkzDrawArc[angles,](B,O)(100,180)
  \tkzCompass[delta=20](D,E)
  \tkzDrawLines(A,B O,E B,E)
  \tkzDrawPoints(A,B,O,D,E)
  \tkzLabelPoints(A,B,O,D,E)
  \tkzMarkRightAngle(O,B,E)
\end{tikzpicture}
```

24.8 angles选项: 示例 2



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(5,0){I}
  \tkzDefPoint(0,5){J}
  \tkzInterCC(O,I)(I,0)\tkzGetPoints{B}{C}
  \tkzInterCC(O,I)(J,0)\tkzGetPoints{D}{A}
  \tkzInterCC(I,0)(J,0)\tkzGetPoints{L}{K}
  \tkzDrawArc[angles](O,I)(0,90)
  \tkzDrawArc[angles,color=gray,style=dashed](I,0)(90,180)
  \tkzDrawArc[angles,color=gray,style=dashed](J,0)(-90,0)
  \tkzDrawPoints(A,B,K)
  \foreach \point in {I,A,B,J,K}{\tkzDrawSegment(O,\point)}
\end{tikzpicture}
```


25 杂项命令

25.1 线段复制

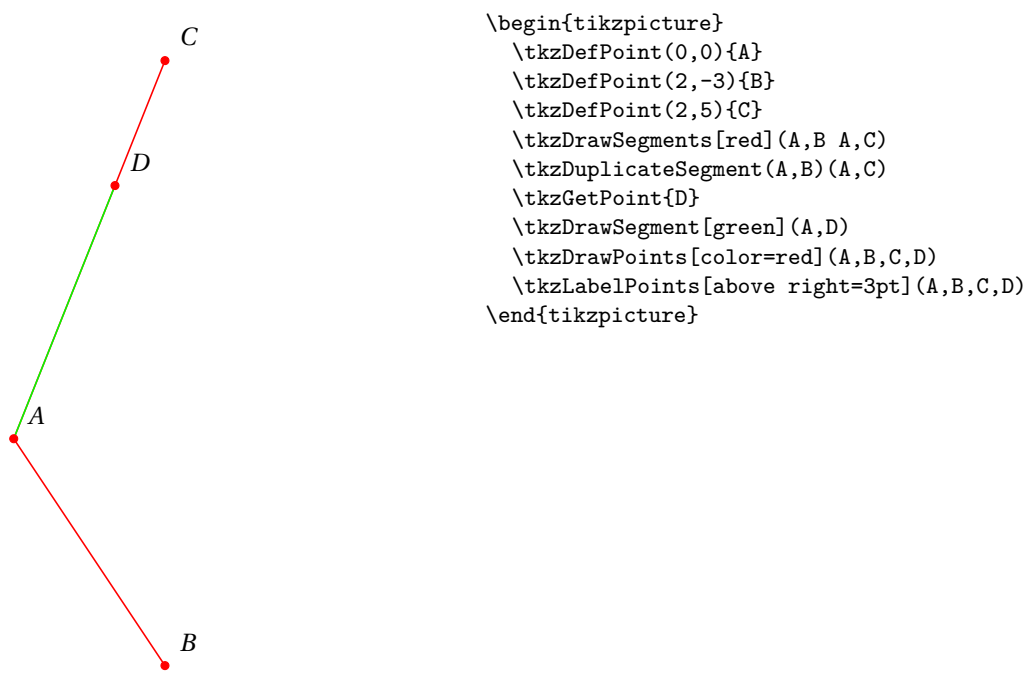
这涉及到在给定的半直线上构造与给定线段长度相同的线段。

`\tkzDuplicateSegment(\langle pt1,pt2\rangle)(\langle pt3,pt4\rangle)\{\langle pt5\rangle\}`

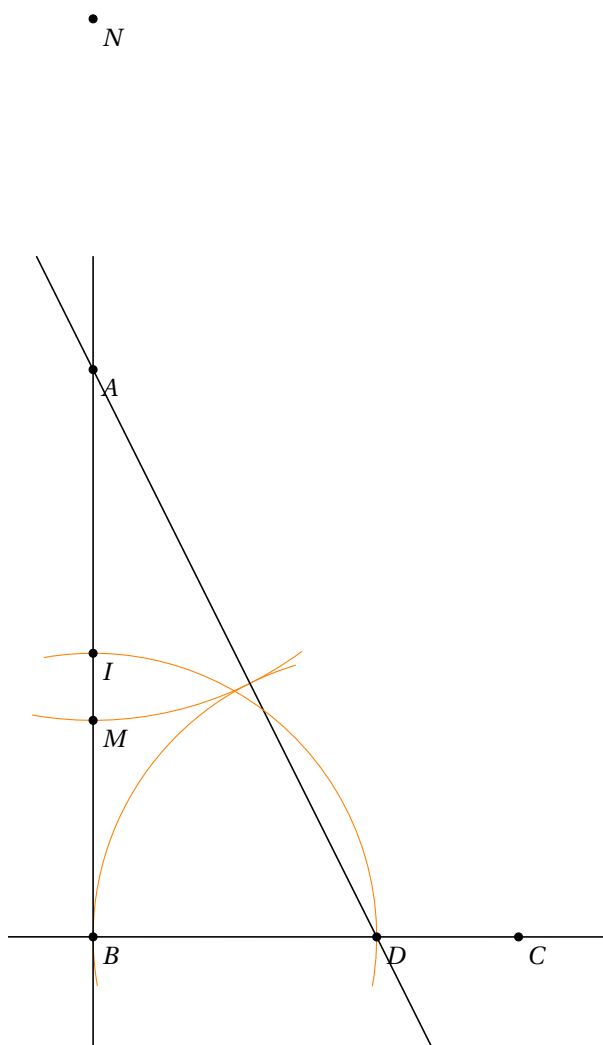
这涉及到在给定的半直线上构造与给定线段长度相同的线段，该命令实际是一个点的定义
`\tkzDuplicateSegment`命令是`\tkzDuplicateLen`的更新版本。

参数	样例	说明
<code>(pt1,pt2)(pt3,pt4)\{pt5\}</code>	<code>\tkzDuplicateSegment(A,B)(E,F)\{C\}</code>	$AC = EF$ 并且 $C \in [AB)$

`\tkzDuplicateLength`命令与该命令相同。



25.1.1 用`\tkzDuplicateSegment`命令得到黄金分割



```
\begin{tikzpicture}[rotate=-90,scale=.75]
\tkzDefPoint(0,0){A}
\tkzDefPoint(10,0){B}
\tkzDefMidPoint(A,B)
\tkzGetPoint{I}
\tkzDefPointWith[orthogonal,K=-.75](B,A)
\tkzGetPoint{C}
\tkzInterLC(B,C)(B,I) \tkzGetSecondPoint{D}
\tkzDuplicateSegment(B,D)(D,A) \tkzGetPoint{E}
\tkzInterLC(A,B)(A,E) \tkzGetPoints{N}{M}
\tkzDrawArc[orange,delta=10](D,E)(B)
\tkzDrawArc[orange,delta=10](A,M)(E)
\tkzDrawLines(A,B B,C A,D)
\tkzDrawArc[orange,delta=10](B,D)(I)
\tkzDrawPoints(A,B,D,C,M,I,N)
\tkzLabelPoints(A,B,D,C,M,I,N)
\end{tikzpicture}
```

25.2 计算线段长度命令`\tkzCalcLength`

也可以用 TikZ 的 `vecLen` 计算长度，该选项能够计算 A 点和 B 点间的距离 AB。

问题是 TikZ 的计算结果精度不足，因此该命令用 `xfp` 宏包实现了计算，虽然其计算比较慢，但精度更高。

`\tkzCalcLength[< 命令选项>](<pt1,pt2>){< 宏名称>}`

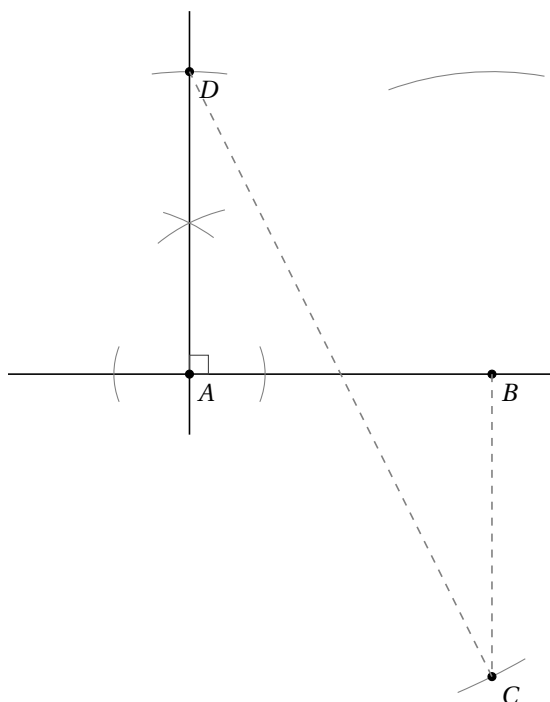
计算结果保存有指定的宏中。

参数	样例	说明
(pt1,pt2){宏名称}	<code>\tkzCalcLength(A,B){dAB}</code>	<code>\dAB</code> 得到 AB 的长度，单位是 pt

仅有 1 个选项。

选项	默认值	样例
cm	false	<code>\tkzCalcLength[cm](A,B){dAB}</code> <code>\dAB</code> 得到 AB 的长度，单位是 cm

25.2.1 构建罗盘矩形



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDrawLine[add= .6 and .2](A,B)
  \tkzCalcLength[cm](A,B)\tkzGetLength{dAB}
  \tkzDefLine[perpendicular=through A](A,B)
  \tkzDrawLine(A,\tkzPointResult)
  \tkzGetPoint{D}
  \tkzShowLine[orthogonal=through A,gap=2](A,B)
  \tkzMarkRightAngle(B,A,D)
  \tkzVecKOrth[-1](B,A)\tkzGetPoint{C}
  \tkzCompass(A,D D,C)
  \tkzDrawArc[R](B,\dAB)(80,110)
  \tkzDrawPoints(A,B,C,D)
  \tkzDrawSegments[color=gray,style=dashed](B,C C,D)
  \tkzLabelPoints(A,B,C,D)
\end{tikzpicture}
```

25.3 将 pt 转换为 cm

不能确定该命令是否有用，仅仅用 28.45274 进行了乘除运算。

`\tkzpttocm(<nombre>){<宏名称>}`

参数	样例	说明
(number) 宏名称	<code>\tkzpttocm(120){len}</code>	<code>\len</code> 得到cm值

结果保存在宏中。

25.4 将 cm 转换为 pt

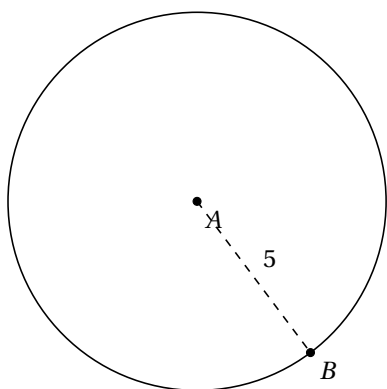
`\tkzcmtopt(<nombre>){<宏名称>}`

参数	样例	说明
(nombre){宏名称}	<code>\tkzcmtopt(5){len}</code>	<code>\len</code> 得到pt值

结果保存在宏中。pt.

25.4.1 示例代码

`\tkzDefCircle[radius](A,B)` 命令用于计算半径, 可以用 `\tkzGetLength` 命令得到计算结果, 但其单位是 **pt**。



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,-4){B}
\tkzDefCircle[through](A,B)
\tkzGetLength{rABpt}
\tkzpttocm(\rABpt){rABcm}
\tkzDrawCircle(A,B)
\tkzDrawPoints(A,B)
\tkzLabelPoints(A,B)
\tkzDrawSegment[dashed](A,B)
\tkzLabelSegment(A,B){$\pgfmathprintnumber{\rABcm}$}
\end{tikzpicture}
```

25.5 获取点的坐标分量

`\tkzGetPointCoord(<A>){<宏名称>}`

参数

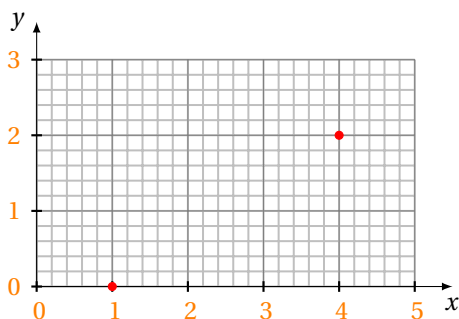
样例

说明

(point){宏名称} `\tkzGetPointCoord(A){A}` `\Ax`和`\Ay`保存点 A 的坐标分量

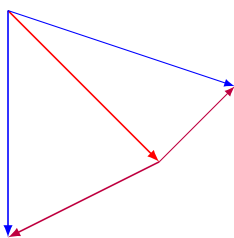
将点的坐标分量保存在两个宏中, 如果宏名称是 **p**, 则将坐标分量保存在 `\px`和`\py`宏中, 单位是 **cm**。

25.5.1 \tkzGetPointCoord命令示例代码



```
\begin{tikzpicture}
\tkzInit[xmax=5,ymax=3]
\tkzGrid[sub,orange]
\tkzAxeXY
\tkzDefPoint(1,0){A}
\tkzDefPoint(4,2){B}
\tkzGetPointCoord(A){a}
\tkzGetPointCoord(B){b}
\tkzDefPoint(\ax,\ay){C}
\tkzDefPoint(\bx,\by){D}
\tkzDrawPoints[color=red](C,D)
\end{tikzpicture}
```

25.5.2 使用\tkzGetPointCoord命令求向量和



```
\begin{tikzpicture}[>=latex]
\tkzDefPoint(1,4){a} \tkzDefPoint(3,2){b}
\tkzDefPoint(1,1){c}
\tkzDrawSegment[->,red](a,b) \tkzGetPointCoord(c){c}
\draw[blue,->](a) -- ([shift=(b)]\cx,\cy) ;
\draw[purple,->](b) -- ([shift=(b)]\cx,\cy) ;
\tkzDrawSegment[->,blue](a,c)
\tkzDrawSegment[->,purple](b,c)
\end{tikzpicture}
```

26 使用罗盘

26.1 主命令\tkzCompass

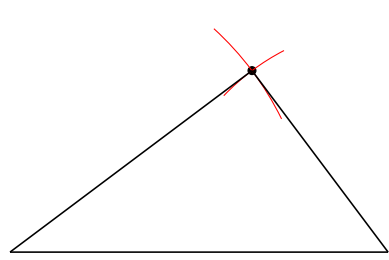
\tkzCompass[< 命令选项>](A,B)

该命令可以为指定点留下罗盘轨迹，也就是一小段圆弧。在使用该命令时，必须指定圆心。可以使用 TikZ 的 style、color、line thickness 等样式设置轨迹的外观。

可以使用length或delta选项指定圆弧长度。

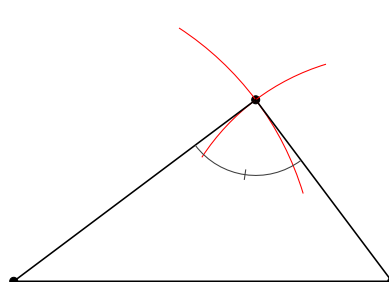
选项	默认值	含义
delta	0 (deg)	通过增加对称度修改圆弧长度 (度)
length	1 (cm)	改变圆弧长度 (cm)

26.1.1 length选项



```
\begin{tikzpicture}
\tkzDefPoint(1,1){A}
\tkzDefPoint(6,1){B}
\tkzInterCC[R](A,4cm)(B,3cm)
\tkzGetPoints{C}{D}
\tkzDrawPoint(C)
\tkzCompass[color=red,length=1.5](A,C)
\tkzCompass[color=red](B,C)
\tkzDrawSegments(A,B A,C B,C)
\end{tikzpicture}
```

26.1.2 delta选项



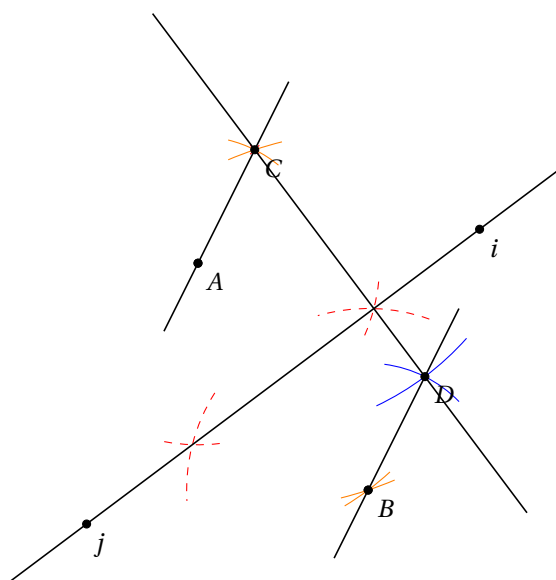
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,0){B}
\tkzInterCC[R](A,4cm)(B,3cm)
\tkzGetPoints{C}{D}
\tkzDrawPoints(A,B,C)
\tkzCompass[color=red,delta=20](A,C)
\tkzCompass[color=red,delta=20](B,C)
\tkzDrawPolygon(A,B,C)
\tkzMarkAngle(A,C,B)
\end{tikzpicture}
```

26.2 多重命令\tkzCompassss

\tkzCompassss[< 命令选项>](pt1,pt2, pt3,pt4,...)

注意：参数是点对列表。

选项	默认值	含义
delta	0	通过增加对称度改变圆弧的角度
length	1	改变圆弧长度



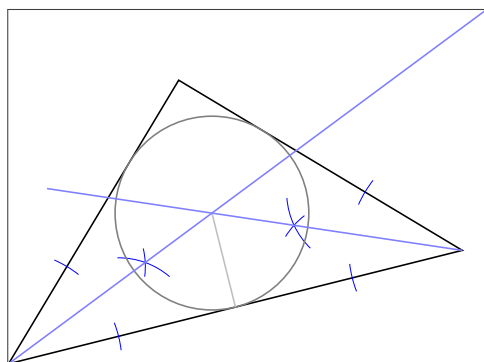
```
\begin{tikzpicture}[scale=.75]
\tkzDefPoint(2,2){A}
\tkzDefPoint(5,-2){B}
\tkzDefPoint(3,4){C}
\tkzDrawPoints(A,B)
\tkzDrawPoint[color=red,shape=cross out](C)
\tkzCompass[color=orange](A,B A,C B,C C,B)
\tkzShowLine[mediator,color=red,
dashed,length = 2](A,B)
\tkzShowLine[parallel = through C,
color=blue,length=2](A,B)
\tkzDefLine[mediator](A,B)
\tkzGetPoints{i}{j}
\tkzDefLine[parallel=through C](A,B)
\tkzGetPoint{D}
\tkzDrawLines[add=.6 and .6](C,D A,C B,D)
\tkzDrawLines(i,j) \tkzDrawPoints(A,B,C,i,j,D)
\tkzLabelPoints(A,B,C,i,j,D)
\end{tikzpicture}
```

26.3 设置命令\tkzSetUpCompass

\tkzSetUpCompass[< 命令选项>]

选项	默认值	含义
line width	0.4pt	线宽
color	black!50	颜色
style	solid	线型: solid、dashed、dotted、...

26.3.1 \tkzSetUpCompass示例代码



```
\begin{tikzpicture}[showbi/.style={bisector,
size=2,gap=3}, scale=.75]
\tkzSetUpCompass[color=blue,line width=.3 pt]
\tkzDefPoints{O/1/A, 8/3/B, 3/6/C}
\tkzDrawPolygon(A,B,C)
\tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
\tkzDefLine[bisector](C,B,A) \tkzGetPoint{b}
\tkzShowLine[showbi](B,A,C)
\tkzShowLine[showbi](C,B,A)
\tkzInterLL(A,a)(B,b) \tkzGetPoint{I}
\tkzDefPointBy[projection= onto A--B](I)
\tkzGetPoint{H}
\tkzDrawCircle[radius,color=gray](I,H)
\tkzDrawSegments[color=gray!50](I,H)
\tkzDrawLines[add=0 and -.2,color=blue!50](A,a B,b)
\tkzShowBB
\end{tikzpicture}
```

27 显示尺规标记

27.1 显示直线尺规作图标记命令\tkzShowLine

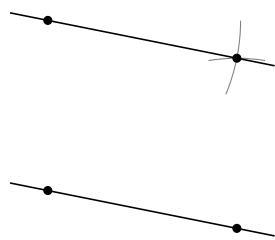
`\tkzShowLine[< 命令选项>](<pt1,pt2>) 或 (<pt1,pt2,pt3>)`

这个命令的原型来自 **Yves Combe**，它用于显示中垂线、过指定点的平行线或垂线、角平分线的尺规作图标记。其参数是两个或三个点，可以通过命令选项对结果进行调整。

选项	默认值	含义
mediator	mediator	中垂线
perpendicular	mediator	垂线
orthogonal	mediator	同上
bisector	mediator	角平分线
K	1	三角形内圆
length	1	圆弧长度，单位是 cm
ratio	.5	圆弧长度比例
gap	2	符号间隙
size	1	圆弧半径（参见 bisector）

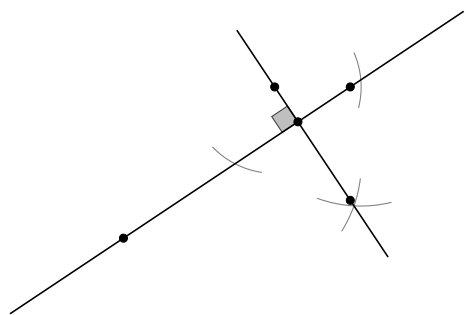
当然，可以使用所有有效的 TikZ 样式。

27.1.1 \tkzShowLine命令和parallel选项



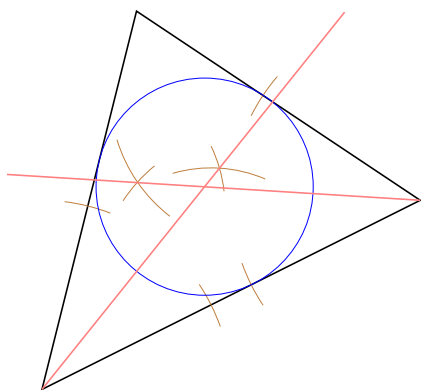
```
\begin{tikzpicture}
\tkzDefPoints{-1.5/-0.25/A,1/-0.75/B,-1.5/2/C}
\tkzDrawLine(A,B)
\tkzDefLine[parallel=through C](A,B) \tkzGetPoint{c}
\tkzShowLine[parallel=through C](A,B)
\tkzDrawLine(C,c) \tkzDrawPoints(A,B,C,c)
\end{tikzpicture}
```

27.1.2 \tkzShowLine命令和perpendicular选项



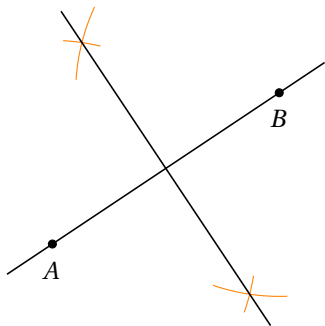
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A, 3/2/B, 2/2/C}
\tkzDefLine[perpendicular=through C,K=-.5](A,B)
\tkzGetPoint{c}
\tkzShowLine[perpendicular=through C,K=-.5,gap=3](A,B)
\tkzDefPointBy[projection=onto A--B](c)
\tkzGetPoint{h}
\tkzMarkRightAngle[fill=lightgray](A,h,C)
\tkzDrawLines[add=.5 and .5](A,B C,c)
\tkzDrawPoints(A,B,C,h,c)
\end{tikzpicture}
```

27.1.3 \tkzShowLine命令和bisector选项



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoints{0/0/A, 4/2/B, 1/4/C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpCompass[color=brown,line width=.1 pt]
\tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
\tkzDefLine[bisector](C,B,A) \tkzGetPoint{b}
\tkzInterLL(A,a)(B,b) \tkzGetPoint{I}
\tkzDefPointBy[projection = onto A--B](I)
\tkzGetPoint{H}
\tkzShowLine[bisector,size=2,gap=3,blue](B,A,C)
\tkzShowLine[bisector,size=2,gap=3,blue](C,B,A)
\tkzDrawCircle[radius,color=blue,line width=.2pt](I,H)
\tkzDrawSegments[color=red!50](I,t kzPointResult)
\tkzDrawLines[add=0 and -0.3,color=red!50](A,a B,b)
\end{tikzpicture}
```

27.1.4 \tkzShowLine命令和mediator选项



```
\begin{tikzpicture}
\tkzDefPoint(2,2){A}
\tkzDefPoint(5,4){B}
\tkzDrawPoints(A,B)
\tkzShowLine[mediator,color=orange,length=1](A,B)
\tkzGetPoints{i}{j}
\tkzDrawLines[add=-0.1 and -0.1](i,j)
\tkzDrawLines(A,B)
\tkzLabelPoints[below =3pt](A,B)
\end{tikzpicture}
```

27.2 显示部分变换过程的尺规作图标记命令\tkzShowTransformation

`\tkzShowTransformation[< 命令选项>](pt1,pt2)` 或 `(pt1,pt2,pt3)`

这个命令的原型来源的 **Yves Combe**，用于显示正交对称、中心对称、正交投影和平移的尺规作图标记。可以通过命令选项对结果进行调整。

选项	默认值	含义
reflection= over pt1--pt2	reflection	正交对称
symmetry=center pt	reflection	中心对称
projection=onto pt1--pt2	reflection	投影
translation=from pt1 to pt2	reflection	平移
K	1	三角形内的圆
length	1	圆弧长度
ratio	.5	圆弧长度比例
gap	2	标记间隙
size	1	圆弧半径 (参见 bisector)

28 差分点

28.1 \tkzDefEquiPoints命令

该命令用于定义一条直线与给定的点等距的两个点。

\tkzDefEquiPoints[< 命令选项>](pt1,pt2)		
参数	默认值	含义
(pt1,pt2)	无	顺序任意的两个点
选项	默认值	含义
dist	2 cm	两个点间距离的一半
from=pt	无	参考点
show	false	如为 true, 则显示尺规标记
/compass/delta	0	尺规标记尺寸

28.1.1 使用带选项的\tkzDefEquiPoints命令

```
\begin{tikzpicture}
\tkzSetUpCompass[color=purple,line width=1pt]
\tkzDefPoint(0,1){A}
\tkzDefPoint(5,2){B}
\tkzDefPoint(3,4){C}
\tkzDefEquiPoints[from=C,dist=1,show,
\tkzcompass/delta=20](A,B)
\tkzGetPoints{E}{H}
\tkzDrawLines[color=blue](C,E C,H A,B)
\tkzDrawPoints[color=blue](A,B,C)
\tkzDrawPoints[color=red](E,H)
\tkzLabelPoints(E,H)
\tkzLabelPoints[color=blue](A,B,C)
\end{tikzpicture}
```

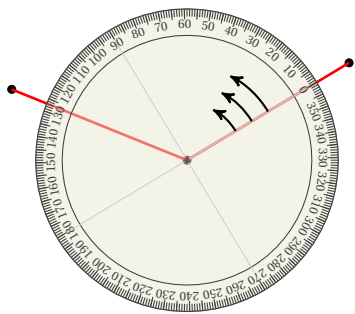
29 量角器

基于 **Yves Combe** 的方法，该命令用于绘制量角器。其工作原理更为简单，仅半条直线 (射线)，量角器原点位于点 O ，射线方向由 A 确定。角度方向由指定的测量圆方向决定。

\tkzProtractor[< 命令选项>](O,A)		
选项	默认值	含义
lw	0.4 pt	线宽
scale	1	比例：用于调整量角器尺寸
return	false	反向测量圆

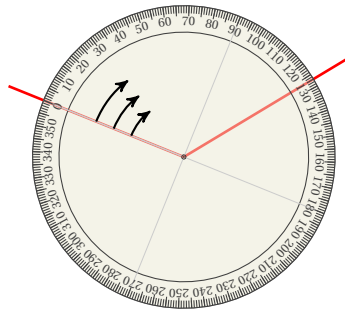
29.1 正向圆量角器

正向测量圆方向



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(2,0){A}\tkzDefPoint(0,0){O}
\tkzDefShiftPoint[A](31:5){B}
\tkzDefShiftPoint[A](158:5){C}
\tkzDrawPoints(A,B,C)
\tkzDrawSegments[color = red,
  line width = 1pt](A,B A,C)
\tkzProtractor[scale = 1](A,B)
\end{tikzpicture}
```

29.2 反向圆量角器



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(2,3){A}
\tkzDefShiftPoint[A](31:5){B}
\tkzDefShiftPoint[A](158:5){C}
\tkzDrawSegments[color=red,line width=1pt](A,B A,C)
\tkzProtractor[return](A,C)
\end{tikzpicture}
```

30 实例分析

30.1 一些有趣的例子

30.1.1 相似等腰三角形

以下示例选自精彩的 **Descartes et les Mathématiques** 网站,在此,未对原文本进行任何修改,仅用 `tkz-euclide` 宏包对其进行了编码实现。

<https://debart.pagesperso-orange.fr/seconde/triangle.html>

参考文献:

- Géométrie au Bac - Tangente, special issue no. 8 - Exercice 11, page 11
- Elisabeth Busser and Gilles Cohen: 200 nouveaux problèmes du “Monde” - POLE 2007 (200 new problems of “Le Monde”)
- Affaire de logique n° 364 - Le Monde February 17, 2004

构造相似等腰三角形有两种方式, 一是 *Tangente* 杂志, 另一个是 *Le Monde*。

“*Tangente*”杂志编辑: 两个相似等腰三角形 AXB 和 BYC 由主顶点 X 和 Y 构成的, 其中, A 、 B 和 C 共线。令 α 为顶点的角度 ($\widehat{AXB} = \widehat{BYC}$), 然后可以构造与前两个等腰三角形相似的第三个等腰三角形 XZY , 其主顶点是 Z 点, 需要证明 Z 点属于直线 (AC) 。

“*Le Monde*”杂志编辑: 可以构造两个相似等腰三角形 AXB 和 BYC 由主顶点 X 和 Y , 其中 A 、 B 和 C 点共线。令 α 为顶点的角度 ($\widehat{AXB} = \widehat{BYC}$), 线段 $[AC]$ 上的点 Z 与 X 和 Y 的距离相等。那么这两个顶点的角度是多少?

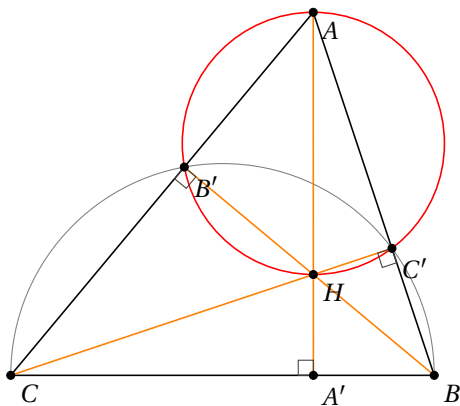
下面两页给出了这两种方法的代码和构造过程, 但在查看代码前, 可以先搜索相关文献。这些代码展示了这两种推导过程。

30.1.4 三角形的高

以下示例选自精彩的 **Descartes et les Mathématiques** 网站 (Descartes and the Mathematics)。

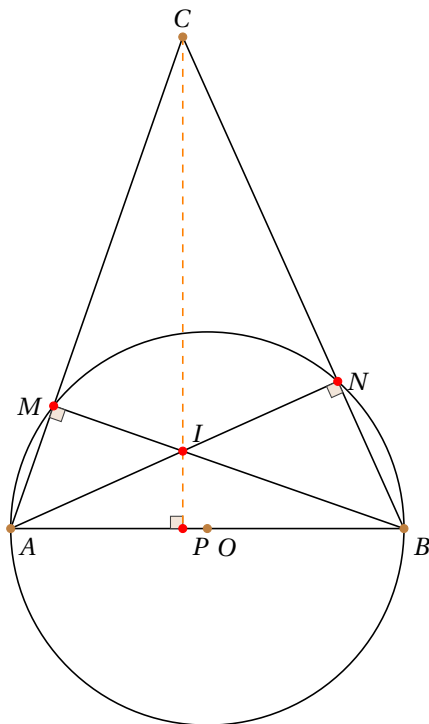
https://debart.pagesperso-orange.fr/geoplan/geometrie_triangle.html

三条高相交于 H 点。



```
\begin{tikzpicture}[scale=.8]
\tkzDefPoint(0,0){C}
\tkzDefPoint(7,0){B}
\tkzDefPoint(5,6){A}
\tkzDrawPolygon(A,B,C)
\tkzDefMidPoint(C,B)
\tkzGetPoint{I}
\tkzDrawArc(I,B)(C)
\tkzInterLC(A,C)(I,B)
\tkzGetSecondPoint{B'}
\tkzInterLC(A,B)(I,B)
\tkzGetFirstPoint{C'}
\tkzInterLL(B,B')(C,C')
\tkzGetPoint{H}
\tkzInterLL(A,H)(C,B)
\tkzGetPoint{A'}
\tkzDefCircle[circum](A,B',C')
\tkzGetPoint{O}
\tkzDrawCircle[color=red](O,A)
\tkzDrawSegments[color=orange](B,B' C,C' A,A')
\tkzMarkRightAngles(C,B',B B,C',C C,A',A)
\tkzDrawPoints(A,B,C,A',B',C',H)
\tkzLabelPoints(A,B,C,A',B',C',H)
\end{tikzpicture}
```

30.1.5 三角形的高 - 另一种构造方式

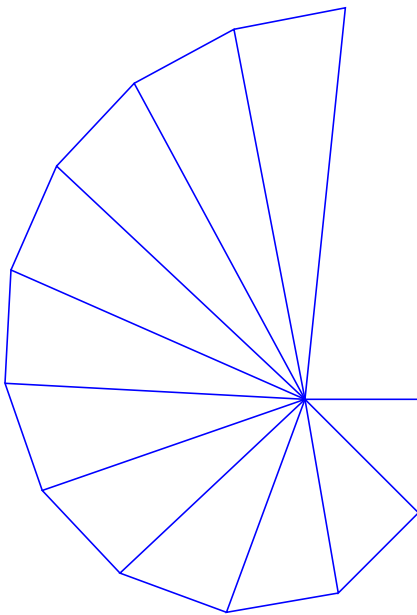


```
\begin{tikzpicture}[scale=0.65]
\tkzDefPoint(0,0){A}
\tkzDefPoint(8,0){B}
\tkzDefPoint(3.5,10){C}
\tkzDefMidPoint(A,B)
\tkzGetPoint{O}
\tkzDefPointBy[projection=onto A--B](C)
\tkzGetPoint{P}
\tkzInterLC(C,A)(O,A)
\tkzGetSecondPoint{M}
\tkzInterLC(C,B)(O,A)
\tkzGetFirstPoint{N}
\tkzInterLL(B,M)(A,N)
\tkzGetPoint{I}
\tkzDrawCircle[diameter](A,B)
\tkzDrawSegments(C,A C,B A,B B,M A,N)
\tkzMarkRightAngles[fill=brown!20](A,M,B A,N,B A,P,C)
\tkzDrawSegment[style=dashed,color=orange](C,P)
\tkzLabelPoints(O,A,B,P)
\tkzLabelPoint[left](M){M$}
\tkzLabelPoint[right](N){N$}
\tkzLabelPoint[above](C){C$}
\tkzLabelPoint[above right](I){I$}
\tkzDrawPoints[color=red](M,N,P,I)
\tkzDrawPoints[color=brown](O,A,B,C)
\end{tikzpicture}
```

30.2 其他作者提供的实例

30.2.1 整数的算术平方根

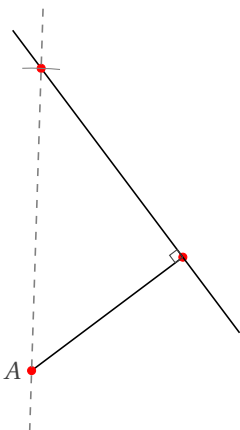
本例演示了如何用尺规求解 1 、 $\sqrt{2}$ 和 $\sqrt{3}$ 的方法。



```
\begin{tikzpicture}[scale=1.5]
\tkzDefPoint(0,0){O}
\tkzDefPoint(1,0){a0}
\tkzDrawSegment[blue](O,a0)
\foreach \i [count=\j] in {0,...,10}{%
\tkzDefPointWith[orthogonal normed](a\i,0)
\tkzGetPoint{a\j}
\tkzDrawPolySeg[color=blue](a\i,a\j,0)}
\end{tikzpicture}
```

30.2.2 直角三角

有一线段 $[AB]$ ，然后确定一点 C 使 $AC = 8 \text{ cm}$ ，并且 ABC 是以 B 为直角的直角三角形。



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint["$A$" left](2,1){A}
\tkzDefPoint(6,4){B}
\tkzDrawSegment(A,B)
\tkzDrawPoint[color=red](A)
\tkzDrawPoint[color=red](B)
\tkzDefPointWith[orthogonal,K=-1](B,A)
\tkzDrawLine[add = .5 and .5](B,t kzPointResult)
\tkzInterLC[R](B,t kzPointResult)(A,8 cm)
\tkzGetPoints{C}{J}
\tkzDrawPoint[color=red](C)
\tkzCompass(A,C)
\tkzMarkRightAngle(A,B,C)
\tkzDrawLine[color=gray,style=dashed](A,C)
\end{tikzpicture}
```

30.2.3 阿基米德

这是伟大的希腊数学家阿基米德证明的一个古老问题。下图有一个直径为 AB 的半圆，一条直线在 B 点与半圆相切，在 C 点有半圆的另一条切线。把 C 点投影到线段 $[AB]$ 上的 D 点。两条切线相交于 T 点。

现证明直线 (AT) 平分直线 (CD)

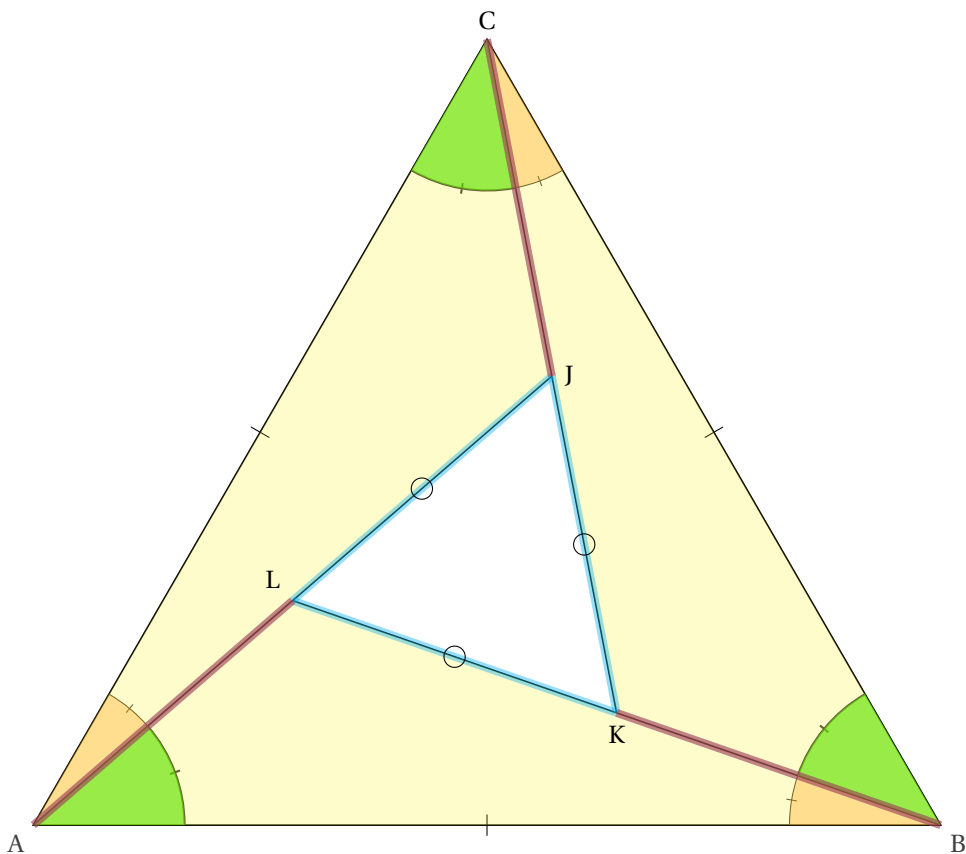

```

\begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2.5,0){N}
  \tkzDefPoint(-4.2,0.5){M}
  \tkzDefPointBy[rotation=center O angle 30](N)
  \tkzGetPoint{B}
  \tkzDefPointBy[rotation=center O angle -50](N)
  \tkzGetPoint{A}
  \tkzInterLC(M,B)(O,N) \tkzGetFirstPoint{C}
  \tkzInterLC(M,A)(O,N) \tkzGetSecondPoint{A'}
  \tkzMarkAngle[mkpos=.2, size=0.5](A,C,B)
  \tkzMarkAngle[mkpos=.2, size=0.5](A,M,C)
  \tkzDrawSegments(A,C M,A M,B)
  \tkzDrawCircle(O,N)
  \tkzLabelCircle[above left](O,N)(120){$\mathcal{C}$}
  \tkzMarkAngle[mkpos=.2, size=1.2](C,A,M)
  \tkzDrawPoints(O, A, B, M, B, C)
  \tkzLabelPoints[right](O,A,B)
  \tkzLabelPoints[above left](M,C)
  \tkzLabelPoint[below left](A'){$A'$}
\end{tikzpicture}

```

30.2.5 示例 1: John Kitzmiller

证明 $\triangle LKJ$ 是等边三角形。



```

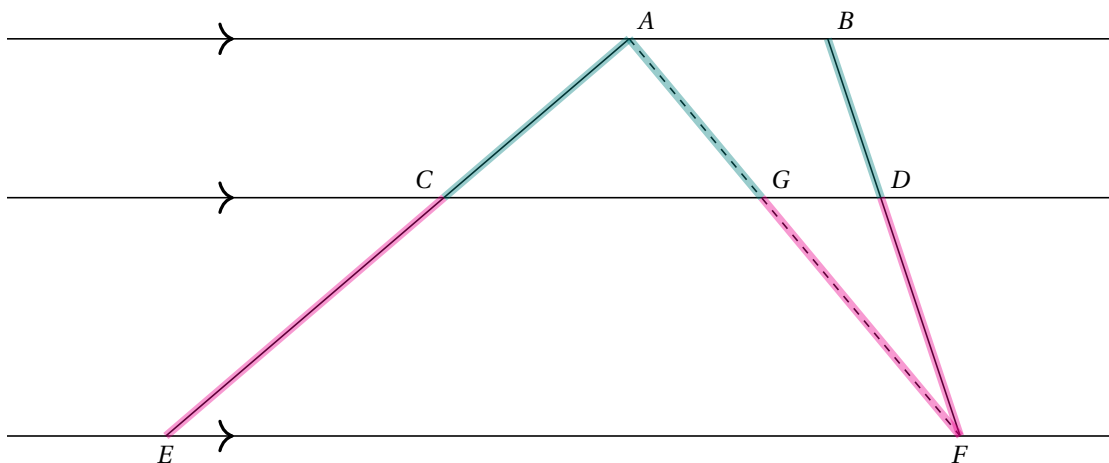
\begin{tikzpicture}[scale=2]
  \tkzDefPoint[label=below left:A](0,0){A}
  \tkzDefPoint[label=below right:B](6,0){B}
  \tkzDefTriangle[equilateral](A,B) \tkzGetPoint{C}
  \tkzMarkSegments[mark=|](A,B A,C B,C)
  \tkzDefBarycentricPoint(A=1,B=2) \tkzGetPoint{C'}
  \tkzDefBarycentricPoint(A=2,C=1) \tkzGetPoint{B'}
  \tkzDefBarycentricPoint(C=2,B=1) \tkzGetPoint{A'}
  \tkzInterLL(A,A')(C,C') \tkzGetPoint{J}
  \tkzInterLL(C,C')(B,B') \tkzGetPoint{K}
  \tkzInterLL(B,B')(A,A') \tkzGetPoint{L}
  \tkzLabelPoint[above](C){C}
  \tkzDrawPolygon(A,B,C) \tkzDrawSegments(A,J B,L C,K)
  \tkzMarkAngles[size=1 cm](J,A,C K,C,B L,B,A)
  \tkzMarkAngles[thick,size=1 cm](A,C,J C,B,K B,A,L)
  \tkzMarkAngles[opacity=.5](A,C,J C,B,K B,A,L)
  \tkzFillAngles[fill= orange,size=1 cm,opacity=.3](J,A,C K,C,B L,B,A)
  \tkzFillAngles[fill=orange, opacity=.3,thick,size=1,](A,C,J C,B,K B,A,L)
  \tkzFillAngles[fill=green, size=1, opacity=.5](A,C,J C,B,K B,A,L)
  \tkzFillPolygon[color=yellow, opacity=.2](J,A,C)
  \tkzFillPolygon[color=yellow, opacity=.2](K,B,C)
  \tkzFillPolygon[color=yellow, opacity=.2](L,A,B)
  \tkzDrawSegments[line width=3pt,color=cyan,opacity=0.4](A,J C,K B,L)
  \tkzDrawSegments[line width=3pt,color=red,opacity=0.4](A,L B,K C,J)
  \tkzMarkSegments[mark=o](J,K K,L L,J)
  \tkzLabelPoint[right](J){J}
  \tkzLabelPoint[below](K){K}
  \tkzLabelPoint[above left](L){L}
\end{tikzpicture}

```

30.2.6 示例 2: John Kitzmiller

证明: $\frac{AC}{CE} = \frac{BD}{DF}$

John 的另一个有趣的例子是如何在 `tkz-euclide` 宏包中使用 TikZ 的类似 `decoration` 和 `postaction` 选项。



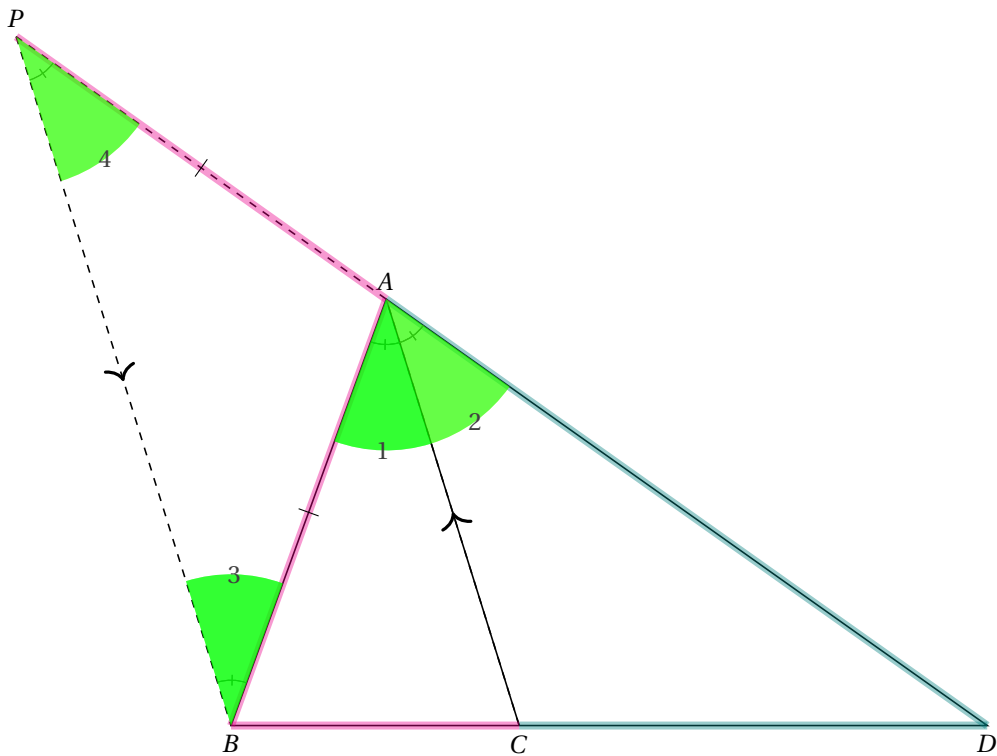
```

\begin{tikzpicture}[scale=1.75,decoration={markings,
  mark=at position 3cm with {\arrow[scale=2]{>}}}
\tkzDefPoints{0/0/E, 6/0/F, 0/1.8/P, 6/1.8/Q, 0/3/R, 6/3/S}
\tkzDrawLines[postaction={decorate}] (E,F P,Q R,S)
\tkzDefPoints{3.5/3/A, 5/3/B}
\tkzDrawSegments(E,A F,B)
\tkzInterLL(E,A)(P,Q) \tkzGetPoint{C}
\tkzInterLL(B,F)(P,Q) \tkzGetPoint{D}
\tkzLabelPoints[above right] (A,B)
\tkzLabelPoints[below] (E,F)
\tkzLabelPoints[above left] (C)
\tkzDrawSegments[style=dashed] (A,F)
\tkzInterLL(A,F)(P,Q) \tkzGetPoint{G}
\tkzLabelPoints[above right] (D,G)
\tkzDrawSegments[color=teal, line width=3pt, opacity=0.4] (A,C A,G)
\tkzDrawSegments[color=magenta, line width=3pt, opacity=0.4] (C,E G,F)
\tkzDrawSegments[color=teal, line width=3pt, opacity=0.4] (B,D)
\tkzDrawSegments[color=magenta, line width=3pt, opacity=0.4] (D,F)
\end{tikzpicture}

```

30.2.7 示例 3: John Kitzmiller

证明: $\frac{BC}{CD} = \frac{AB}{AD}$ (角平分线).

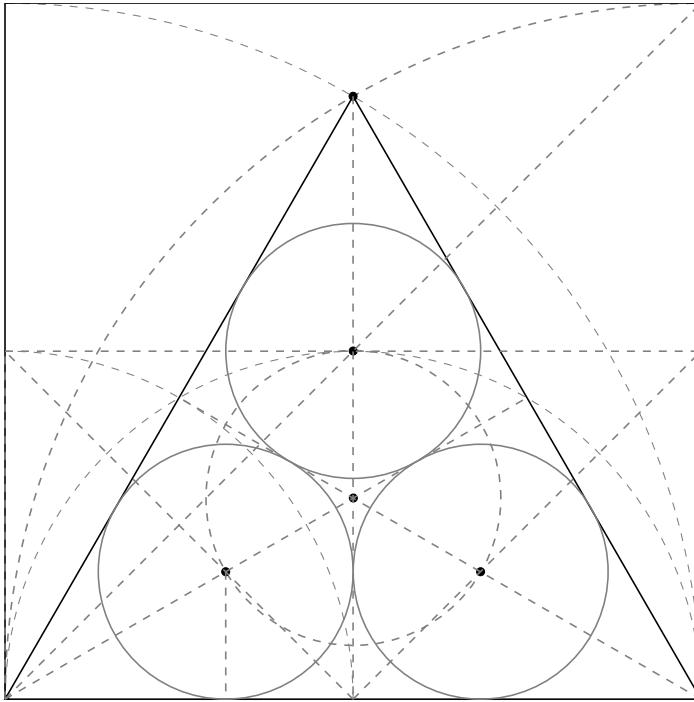



```

\begin{tikzpicture}[pol/.style={fill=brown!40,opacity=.5},
                    seg/.style={tkzdotted,color=gray},
                    hidden pt/.style={fill=gray!40},
                    mra/.style={color=gray!70,tikzdotted,/tkzrightangle/size=.2},
                    scale=3]
\tkzSetUpPoint[size=2]
\tkzDefPoints{O/0/A,2.5/0/B,1.33/0.75/D,0/2.5/E,2.5/2.5/F}
\tkzDefLine[parallel=through D](A,B) \tkzGetPoint{I1}
\tkzDefLine[parallel=through B](A,D) \tkzGetPoint{I2}
\tkzInterLL(D,I1)(B,I2) \tkzGetPoint{C}
\tkzDefLine[parallel=through E](A,D) \tkzGetPoint{I3}
\tkzDefLine[parallel=through D](A,E) \tkzGetPoint{I4}
\tkzInterLL(E,I3)(D,I4) \tkzGetPoint{H}
\tkzDefLine[parallel=through F](E,H) \tkzGetPoint{I5}
\tkzDefLine[parallel=through H](E,F) \tkzGetPoint{I6}
\tkzInterLL(F,I5)(H,I6) \tkzGetPoint{G}
\tkzDefMidPoint(G,H) \tkzGetPoint{P}
\tkzDefMidPoint(G,C) \tkzGetPoint{Q}
\tkzDefMidPoint(B,C) \tkzGetPoint{R}
\tkzDefMidPoint(A,B) \tkzGetPoint{S}
\tkzDefMidPoint(A,E) \tkzGetPoint{T}
\tkzDefMidPoint(E,H) \tkzGetPoint{U}
\tkzDefMidPoint(A,D) \tkzGetPoint{M}
\tkzDefMidPoint(D,C) \tkzGetPoint{N}
\tkzInterLL(B,D)(S,R) \tkzGetPoint{L}
\tkzInterLL(H,F)(U,P) \tkzGetPoint{K}
\tkzDefLine[parallel=through K](D,H) \tkzGetPoint{I7}
\tkzInterLL(K,I7)(B,D) \tkzGetPoint{O}
\tkzFillPolygon[pol](P,Q,R,S,T,U)
\tkzDrawSegments[seg](K,O K,L P,Q R,S T,U
                     C,D H,D A,D M,N B,D)
\tkzDrawSegments(E,H B,C G,F G,H G,C Q,R S,T U,P H,F)
\tkzDrawPolygon(A,B,F,E)
\tkzDrawPoints(A,B,C,E,F,G,H,P,Q,R,S,T,U,K)
\tkzDrawPoints[hidden pt](M,N,O,D)
\tkzMarkRightAngle[mra](L,O,K)
\tkzMarkSegments[mark=|,size=1pt,thick,color=gray](A,S B,S B,R C,R
              Q,C Q,G G,P H,P
              E,U H,U E,T A,T)
\tkzLabelAngle[pos=.3](K,L,O){$\alpha$}
\tkzLabelPoints[below](O,A,S,B)
\tkzLabelPoints[above](H,P,G)
\tkzLabelPoints[left](T,E)
\tkzLabelPoints[right](C,Q)
\tkzLabelPoints[above left](U,D,M)
\tkzLabelPoints[above right](L,N)
\tkzLabelPoints[below right](F,R)
\tkzLabelPoints[below left](K)
\end{tikzpicture}

```


30.2.11 三个圆

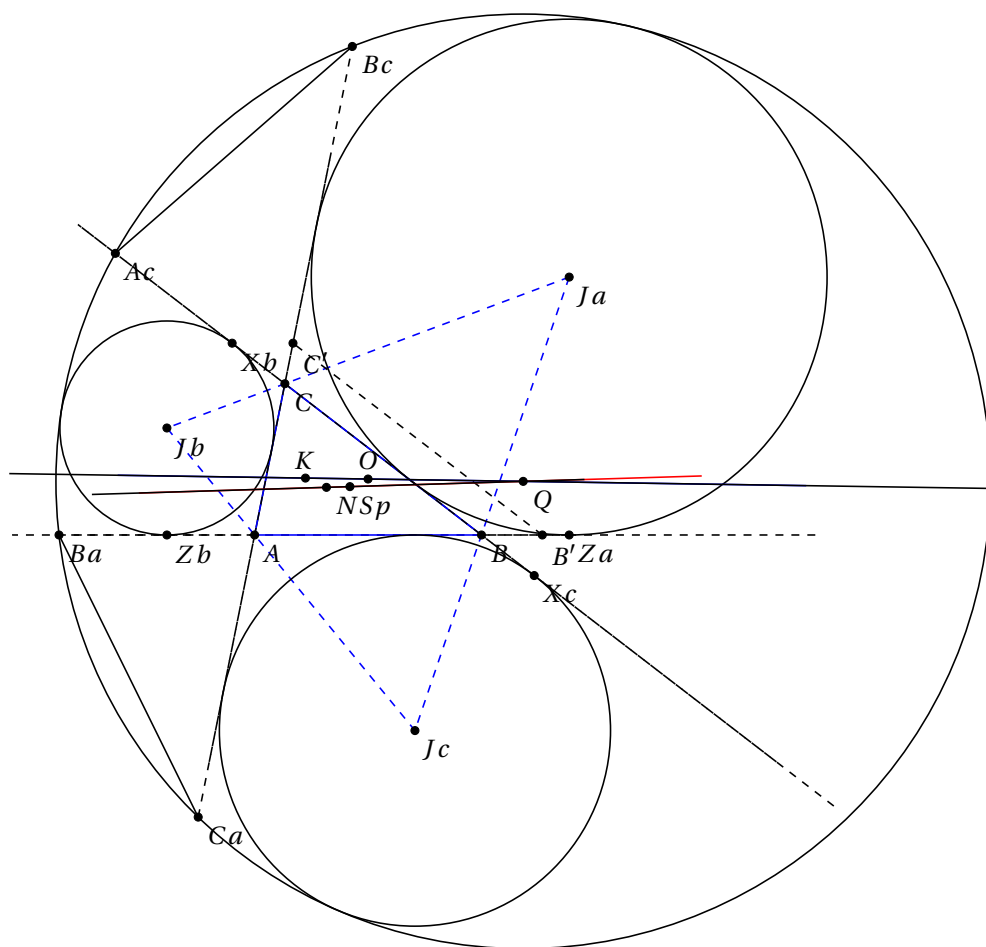


```

\begin{tikzpicture}[scale=1.15]
  \tkzDefPoints{0/0/A,8/0/B,0/4/a,8/4/b,8/8/c}
  \tkzDefTriangle[equilateral](A,B) \tkzGetPoint{C}
  \tkzDrawPolygon(A,B,C)
  \tkzDefSquare(A,B) \tkzGetPoints{D}{E}
  \tkzClipBB
  \tkzDefMidPoint(A,B) \tkzGetPoint{M}
  \tkzDefMidPoint(B,C) \tkzGetPoint{N}
  \tkzDefMidPoint(A,C) \tkzGetPoint{P}
  \tkzDrawSemiCircle[gray,dashed](M,B)
  \tkzDrawSemiCircle[gray,dashed](A,M)
  \tkzDrawSemiCircle[gray,dashed](A,B)
  \tkzDrawCircle[gray,dashed](B,A)
  \tkzInterLL(A,N)(M,a) \tkzGetPoint{Ia}
  \tkzDefPointBy[projection = onto A--B](Ia)
  \tkzGetPoint{ha}
  \tkzDrawCircle[gray](Ia,ha)
  \tkzInterLL(B,P)(M,b) \tkzGetPoint{Ib}
  \tkzDefPointBy[projection = onto A--B](Ib)
  \tkzGetPoint{hb}
  \tkzDrawCircle[gray](Ib,hb)
  \tkzInterLL(A,c)(M,C) \tkzGetPoint{Ic}
  \tkzDefPointBy[projection = onto A--C](Ic)
  \tkzGetPoint{hc}
  \tkzDrawCircle[gray](Ic,hc)
  \tkzInterLL(A,Ia)(B,Ib) \tkzGetPoint{G}
  \tkzDrawCircle[gray,dashed](G,Ia)
  \tkzDrawPolySeg(A,E,D,B)
  \tkzDrawPoints(A,B,C)
  \tkzDrawPoints(G,Ia,Ib,Ic)
  \tkzDrawSegments[gray,dashed](C,M A,N B,P M,a M,b A,a a,b b,B A,D Ia,ha)
\end{tikzpicture}

```

30.2.12 APOLLONIUS 圆



```

\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{O/0/A,6/0/B,0.8/4/C}
  \tkzDefTriangleCenter[euler](A,B,C)      \tkzGetPoint{N}
  \tkzDefTriangleCenter[circum](A,B,C)     \tkzGetPoint{O}
  \tkzDefTriangleCenter[lemoine](A,B,C)    \tkzGetPoint{K}
  \tkzDefTriangleCenter[spieker](A,B,C)    \tkzGetPoint{Sp}
  \tkzDefExCircle(A,B,C)      \tkzGetPoint{Jb}
  \tkzDefExCircle(C,A,B)      \tkzGetPoint{Ja}
  \tkzDefExCircle(B,C,A)      \tkzGetPoint{Jc}
  \tkzDefPointBy[projection=onto B--C](Jc) \tkzGetPoint{Xc}
  \tkzDefPointBy[projection=onto B--C](Jb) \tkzGetPoint{Xb}
  \tkzDefPointBy[projection=onto A--B](Ja)  \tkzGetPoint{Za}
  \tkzDefPointBy[projection=onto A--B](Jb)  \tkzGetPoint{Zb}
  \tkzDefLine[parallel=through Xc](A,C)     \tkzGetPoint{X'c}
  \tkzDefLine[parallel=through Xb](A,B)     \tkzGetPoint{X'b}
  \tkzDefLine[parallel=through Za](C,A)     \tkzGetPoint{Z'a}
  \tkzDefLine[parallel=through Zb](C,B)     \tkzGetPoint{Z'b}
  \tkzInterLL(Xc,X'c)(A,B)                 \tkzGetPoint{B'}
  \tkzInterLL(Xb,X'b)(A,C)                 \tkzGetPoint{C'}
  \tkzInterLL(Za,Z'a)(C,B)                 \tkzGetPoint{A''}
  \tkzInterLL(Zb,Z'b)(C,A)                 \tkzGetPoint{B''}
  \tkzDefPointBy[reflection= over Jc--Jb](B') \tkzGetPoint{Ca}
  \tkzDefPointBy[reflection= over Jc--Jb](C') \tkzGetPoint{Ba}
  \tkzDefPointBy[reflection= over Ja--Jb](A'') \tkzGetPoint{Bc}
  \tkzDefPointBy[reflection= over Ja--Jb](B'') \tkzGetPoint{Ac}
  \tkzDefCircle[circum](Ac,Ca,Ba)           \tkzGetPoint{Q}
  \tkzDrawCircle[circum](Ac,Ca,Ba)
  \tkzDefPointWith[linear,K=1.1](Q,Ac)      \tkzGetPoint{nAc}
  \tkzClipCircle[through](Q,nAc)
  \tkzDrawLines[add=1.5 and 1.5,dashed](A,B B,C A,C)
  \tkzDrawPolygon[color=blue](A,B,C)
  \tkzDrawPolygon[dashed,color=blue](Ja,Jb,Jc)
  \tkzDrawCircles[ex](A,B,C B,C,A C,A,B)
  \tkzDrawLines[add=0 and 0,dashed](Ca,Bc B,Za A,Ba B',C')
  \tkzDrawLine[add=1 and 1,dashed](Xb,Xc)
  \tkzDrawLine[add=7 and 3,blue](O,K)
  \tkzDrawLine[add=8 and 15,red](N,Sp)
  \tkzDrawLines[add=10 and 10](K,O N,Sp)
  \tkzDrawSegments(Ba,Ca Bc,Ac)
  \tkzDrawPoints(A,B,C,N,Ja,Jb,Jc,Xb,Xc,B',C',Za,Zb,Ba,Ca,Bc,Ac,Q,Sp,K,O)
  \tkzLabelPoints(A,B,C,N,Ja,Jb,Jc,Xb,Xc,B',C',Za,Zb,Ba,Ca,Bc,Ac,Q,Sp)
  \tkzLabelPoints[above](K,O)
\end{tikzpicture}

```

31 个性化设置

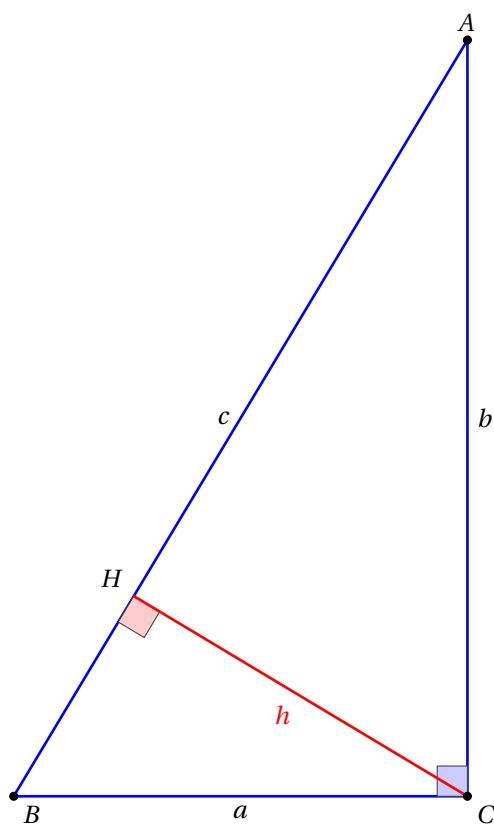
31.1 使用\tkzSetUpLine命令

该命令用于设置所有线条的样式。

`\tkzSetUpLine[< 命令选项>]`

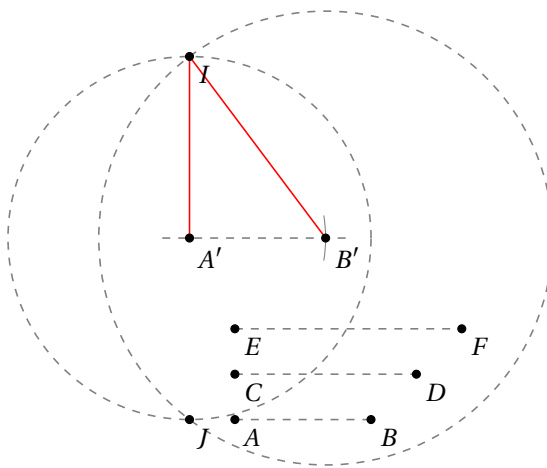
选项	默认值	含义
color	black	颜色
line width	0.4pt	线宽
style	solid	线型
add	.2 and .2	线段两端延伸长度

31.1.1 示例 1：改变线宽



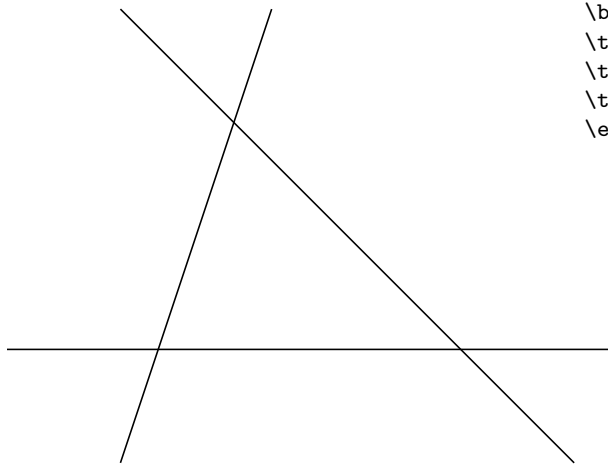
```
\begin{tikzpicture}
\tkzSetUpLine[color=blue,line width=1pt]
\begin{scope}[rotate=-90]
\tkzDefPoint(10,6){C}
\tkzDefPoint( 0,6){A}
\tkzDefPoint(10,0){B}
\tkzDefPointBy[projection = onto B--A](C)
\tkzGetPoint{H}
\tkzDrawPolygon(A,B,C)
\tkzMarkRightAngle[size=.4,fill=blue!20](B,C,A)
\tkzMarkRightAngle[size=.4,fill=red!20](B,H,C)
\tkzDrawSegment[color=red](C,H)
\end{scope}
\tkzLabelSegment[below](C,B){$a$}
\tkzLabelSegment[right](A,C){$b$}
\tkzLabelSegment[left](A,B){$c$}
\tkzLabelSegment[color=red](C,H){$h$}
\tkzDrawPoints(A,B,C)
\tkzLabelPoints[above left](H)
\tkzLabelPoints(B,C)
\tkzLabelPoints[above](A)
\end{tikzpicture}
```

31.1.2 示例 2: 改变线型



```
\begin{tikzpicture}[scale=.6]
\tkzDefPoint(1,0){A} \tkzDefPoint(4,0){B}
\tkzDefPoint(1,1){C} \tkzDefPoint(5,1){D}
\tkzDefPoint(1,2){E} \tkzDefPoint(6,2){F}
\tkzDefPoint(0,4){A'}\tkzDefPoint(3,4){B'}
\tkzCalcLength[cm](C,D) \tkzGetLength{rCD}
\tkzCalcLength[cm](E,F) \tkzGetLength{rEF}
\tkzInterCC[R](A',\rCD cm)(B',\rEF cm)
\tkzGetPoints{I}{J}
\tkzSetUpLine[style=dashed,color=gray]
\tkzDrawLine(A',B')
\tkzCompass(A',B')
\tkzDrawSegments(A,B C,D E,F)
\tkzDrawCircle[R](A',\rCD cm)
\tkzDrawCircle[R](B',\rEF cm)
\tkzSetUpLine[color=red]
\tkzDrawSegments(A',I B',I)
\tkzDrawPoints(A,B,C,D,E,F,A',B',I,J)
\tkzLabelPoints(A,B,C,D,E,F,A',B',I,J)
\end{tikzpicture}
```

31.1.3 示例 3: 延伸



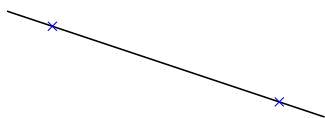
```
\begin{tikzpicture}
\tkzSetUpLine[add=.5 and .5]
\tkzDefPoints{0/0/A,4/0/B,1/3/C}
\tkzDrawLines(A,B B,C A,C)
\end{tikzpicture}
```

31.2 点的样式

`\tkzSetUpPoint[< 命令选项>]`

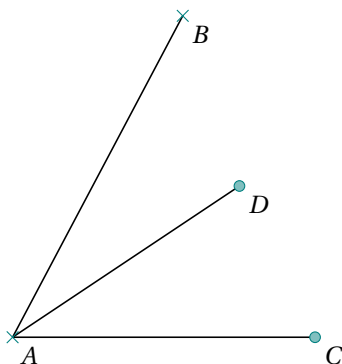
选项	默认值	含义
color	black	颜色
size	3pt	尺寸
fill	black!50	填充色
shape	circle	圆或十字线

31.2.1 \tkzSetUpPoint 命令



```
\begin{tikzpicture}
\tkzSetUpPoint[shape = cross out,color=blue]
\tkzInit[xmax=100,xstep=20,ymax=.5]
\tkzDefPoint(20,1){A}
\tkzDefPoint(80,0){B}
\tkzDrawLine(A,B)
\tkzDrawPoints(A,B)
\end{tikzpicture}
```

31.2.2 在组内使用\tkzSetUpPoint 命令



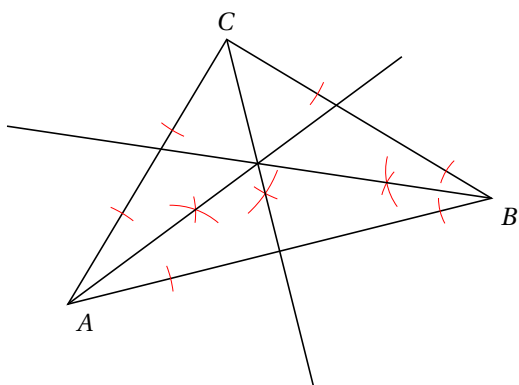
```
\begin{tikzpicture}
\tkzInit[ymin=-0.5,ymax=3,xmin=-0.5,xmax=7]
\tkzDefPoint(0,0){A}
\tkzDefPoint(02.25,04.25){B}
\tkzDefPoint(4,0){C}
\tkzDefPoint(3,2){D}
\tkzDrawSegments(A,B A,C A,D)
\tkzSetUpPoint[shape=cross out,
fill= teal!50,
size=4,color=teal]
\tkzDrawPoints(A,B)
\tkzSetUpPoint[fill= teal!50,size=4,
color=teal]
\tkzDrawPoints(C,D)
\tkzLabelPoints(A,B,C,D)
\end{tikzpicture}
```

31.3 \tkzSetUpCompass 命令

\tkzSetUpCompass[< 命令选项>]

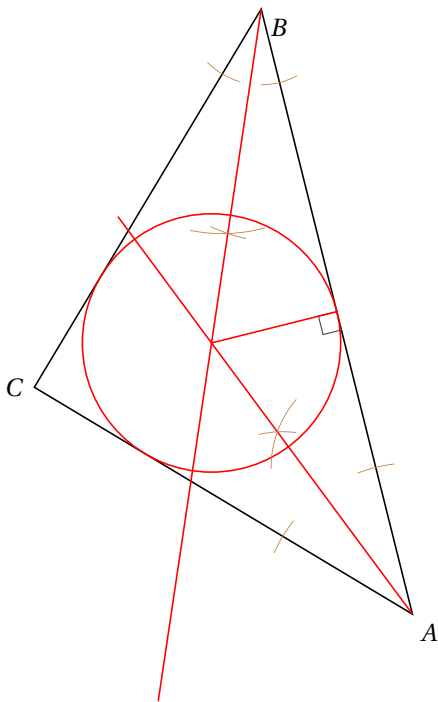
选项	默认值	含义
color	black	圆弧颜色
line width	0.4pt	圆弧线宽
style	solid	圆弧线型

31.3.1 角平分线的\tkzSetUpCompass 命令



```
\begin{tikzpicture}[scale=0.7]
\tkzDefPoints{0/1/A, 8/3/B, 3/6/C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpCompass[color=red,line width=.2 pt]
\tkzDefLine[bisector](A,C,B) \tkzGetPoint{c}
\tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
\tkzDefLine[bisector](C,B,A) \tkzGetPoint{b}
\tkzShowLine[bisector,size=2,gap=3](A,C,B)
\tkzShowLine[bisector,size=2,gap=3](B,A,C)
\tkzShowLine[bisector,size=1,gap=2](C,B,A)
\tkzDrawLines[add=0 and 0](B,b)
\tkzDrawLines[add=0 and -.4](A,a C,c)
\tkzLabelPoints(A,B) \tkzLabelPoints[above](C)
\end{tikzpicture}
```

31.3.2 使用\tkzSetUpCompass的另一个示例



```
\begin{tikzpicture}[scale=1,rotate=90]
\tkzDefPoints{0/1/A, 8/3/B, 3/6/C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpCompass[color=brown,
    line width=.3 pt,style=tikzdotted]
\tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
\tkzDefLine[bisector](C,B,A) \tkzGetPoint{b}
\tkzInterLL(A,a)(B,b) \tkzGetPoint{I}
\tkzDefPointBy[projection= onto A--B](I)
\tkzGetPoint{H}
\tkzMarkRightAngle(I,H,A)
\tkzDrawCircle[radius,color=red](I,H)
\tkzDrawSegments[color=red](I,H)
\tkzDrawLines[add=0 and -.5,,color=red](A,a)
\tkzDrawLines[add=0 and 0,color=red](B,b)
\tkzShowLine[bisector,size=2,gap=3](B,A,C)
\tkzShowLine[bisector,size=1,gap=3](C,B,A)
\tkzLabelPoints(A,B)\tkzLabelPoints[left](C)
\end{tikzpicture}
```

31.4 Own style

既可以使用tkzSetUpPoint设置全局样式，也使用TikZ选项设置局部样式。

```
○ A \tkzSetUpPoint[color=blue!50!white, fill=gray!20!red!50!white]
\tikzset{/tikz/mystyle/.style={color=blue!20!black,fill=blue!20}}
• O \begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(0,1){A}
\tkzDrawPoints(O) % general style
\tkzDrawPoints[mystyle,size=4](A) % my style
\tkzLabelPoints(O,A)
\end{tikzpicture}
```

32 tkz-base 小结

32.1 tkz-base宏包的工具

首先, 无需处理 TikZ 的包围盒尺寸, 早期的 **tkz-euclide** 宏包未对包围盒进行控制, 现在提供了包围盒设置命令。

然而, 有时也需要控制显示的尺寸。为此, 需要设计工具区域包围盒, 这由 **tkz-base** 宏包实现, 该宏包提供的主要命令是 **\tkzInit**, 并建议使用 **lcm** 的绘图单位。对某些绘图的情况, 则需要指定画面大小 (**xmin**, **xmax**, **ymin** and **ymax**), 并使用“裁剪”矩形尽可能控制图形尺寸。

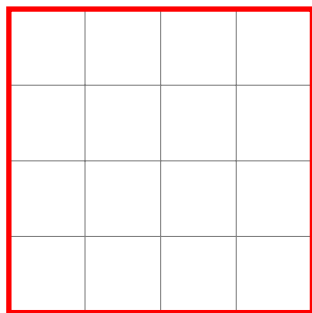
tkz-euclide 宏包使用的 **tkz-base** 宏包提供的两个命令是:

- **\tkzInit**
- **\tkzClip**

为实现该功能, 直接设计了一个命令用于操作包围盒, 可以查看、备份、恢复包围盒 (参见 **tkz-base** 宏包的 Bounding Box 小节)。

32.2 \tkzInit命令和\tkzShowBB命令

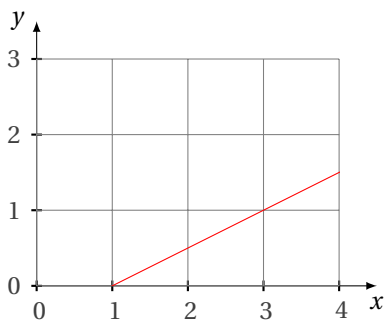
用包围在图形周围的矩形表示包围盒。



```
\begin{tikzpicture}
  \tkzInit[xmin=-1,xmax=3,ymin=-1,ymax=3]
  \tkzGrid
  \tkzShowBB[red,line width=2pt]
\end{tikzpicture}
```

32.3 \tkzClip命令

该命令的作用是通过初始绘图矩形的“裁剪”, 以便仅显示指定矩形范围的内容。



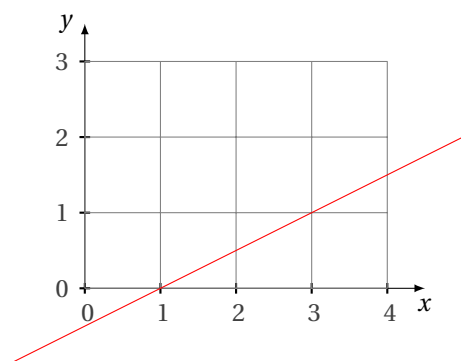
```
\begin{tikzpicture}
  \tkzInit[xmax=4,ymax=3]
  \tkzAxeXY
  \tkzGrid
  \tkzClip
  \draw[red] (-1,-1)--(5,2);
\end{tikzpicture}
```

可以通过命令选项在裁剪区域四周添加指定的空间。

```
\tkzClip[space=1]
```

32.4 \tkzClip命令和space选项

该选项可以裁剪区域四周添加指定的空间。



```
\begin{tikzpicture}
  \tkzInit[xmax=4, ymax=3]
  \tkzAxeXY
  \tkzGrid
  \tkzClip[space=1]
  \draw[red] (-1,-1)--(5,2);
\end{tikzpicture}
```

使用`space`选项后，“裁剪”矩形区域大小为：`xmin-1`、`ymin-1`、`xmax+1`和`ymax+1`。

33 FAQ

33.1 常见错误

目前, 根据需要, 已对语法进行了多次修改, 这可能会带来很多错误, 主要有:

- 绘制多个点时, `\tkzDrawPoint(A,B)`命令中需要使用`\tkzDrawPoints`命令。
- 使用`\tkzGetPoint(A)`得到并命名一个点时, 需要使用大括号而不是小括号, 因此, 应该写成: `\tkzGetPoint{A}`.
- 可以用`\tkzGetPoint{A}`命令代替`\tkzGetFirstPoint{A}`命令。当返回两个点时, 可以使用`\tkzGetPoints{A}{B}`命令一次得到两个点, 也可以使用`\tkzGetFirstPoint{A}`命令或`\tkzGetSecondPoint{A}`命令得到两个点中的一个点。也可以通过`\tkzFirstPointResult`命令或`\tkzSecondPointResult`命令引用这两个点中的一个点。第3个点可以用`\tkzPointResult`命令得到。
- 需要绘制多条线段时, 应该使用`\tkzDrawSegments`带“s”的命令, 而不能使用`\tkzDrawSegment(A,B A,C)`命令, 但注意该命令效率不高。
- 命令选项与参数需要配合使用, 所有圆相关的命令都需要知道圆的半径, 如果半径需要通过计算得到, 则需要使用`\tkzR`命令选项。
- `\tkzDrawSegments[color = gray,style=dashed]{B,B' C,C'}`是错误语法, 仅对象定义命令才使用大括号。
- 角度的度量单位常用度, 极少使用弧度。
- 如果传入参数时需要计算, 并发生了错误, 那么建议在使用该命令前完成计算。
- 不要混合使用`pgfmath`和`\xfp`的语法, 本宏包主要使用`\xfp`语法, 但如果更喜欢 `pgfmath` 语法, 则建议在传入参数前完成计算。
- 有关`\tkzClip`的用法: 为了得到更为精确的结果, 该宏包尽量避免向量归一化计算。向量归一化的优点是能够更好的控制对象尺寸, 但其缺点是在使用 \TeX 计算时, 会带来精度损失。这些误差通常很小, 约为千分之一, 但是当图幅较大时, 其累积误差可能会导致灾难。不归一化, 意味着某些点可能会远离工作区域, `\tkzClip`能够实现图形的裁剪。
- 如果角度太小时, 使用`\tkzDrawAngle`命令, 则会发生错误。该错误是使用在圆弧上放置标记的`decoration`装饰库产生的。即使没有标记, 该错误仍然存在。例如, 可以使用 `mkpos=.2` 选项绕过该错误, 该选项能够在绘制圆弧之前布置标记另一种方法是使用`\tkzFillAngle`命令。

Index

`\ang`, 107
`\Ax`, 124
`\Ay`, 124

`\coordinate`, 24

`\dAB`, 122
`\Delta`, 59
`\draw (A)--(B);` 命令, 60

Environment
 scope, 26

`\fpeval`, 95

`\len`, 123

`\newdimen`, 96

Operating System
 Windows, 20

Package
 fp, 20, 22
 numprint, 20
 pgfmath, 26, 154
 tikz 3.00, 20
 tkz-base, 20, 23, 152
 tkz-euclide, 20, 132, 152
 tzk-base, 152
 xfp, 20, 22, 24–26, 95, 122, 154
`\pgflinewidth`, 35, 36
`\pgfmathsetmacro`, 95
`\px`, 124
`\py`, 124

`\slope`, 109
standalone, 18

TeX Distributions
 MiKTeX, 20
 TeXLive, 20
TikZ Library
 angles, 22
 babel, 14
 decoration, 154
 quotes, 22
`\tkzAngleResult`, 107, 110
`\tkzCalcLength`, 122
`\tkzCalcLength:` arguments
 (`pt1`,`pt2`){宏名称}, 122
`\tkzCalcLength:` options
 cm, 122
`\tkzCalcLength[<命令选项>](<pt1,pt2>){<宏名称>}`, 122
`\tkzCentroid`, 29
`\tkzClip`, 14, 152, 154
`\tkzClipBB`, 22
`\tkzClipCircle`, 82, 87, 91
`\tkzClipCircle:` arguments
 (`A,B`) or (`A,r`), 91
`\tkzClipCircle:` options
 R, 91
 radius, 91

`\tkzClipCircle`[<命令选项>]($\langle A, B \rangle$) 或 ($\langle A, r \rangle$), 91
`\tkzClipPolygon`, 79
`\tkzClipPolygon`: arguments
 ($\langle pt1, pt2 \rangle$), 79
`\tkzClipPolygon`[<命令选项>](\langle 点集列表 \rangle), 79
`\tkzClipSector`(0,A)(B), 115
`\tkzClipSector`[R](0,2 cm)(30,90), 115
`\tkzClipSector`[rotate](0,A)(90), 115
`\tkzClipSector`, 115, 116
`\tkzClipSector`: options
 R, 115
 rotate, 115
 towards, 115
`\tkzClipSector`[<命令选项>]($\langle 0, \dots \rangle$)($\langle \dots \rangle$), 115
`\tkzcmtopt`, 123
`\tkzcmtopt`: arguments
 (nombre){宏名称}, 123
`\tkzcmtopt`(\langle nombre \rangle){ \langle 宏名称 \rangle }, 123
`\tkzCompass`, 119, 125
`\tkzCompass`: options
 delta, 125
 length, 125
`\tkzCompass`s, 125
`\tkzCompass`s: options
 delta, 125
 length, 125
`\tkzCompass`s[<命令选项>]($\langle pt1, pt2, pt3, pt4, \dots \rangle$), 125
`\tkzCompass`[<命令选项>]($\langle A, B \rangle$), 125
`\tkzDefBarycentricPoint`, 29
`\tkzDefBarycentricPoint`: arguments
 ($pt1=a_1, pt2=a_2, \dots$), 29
`\tkzDefBarycentricPoint`($\langle pt1=a_1, pt2=a_2, \dots \rangle$), 29
`\tkzDefCircle`[radius](A,B), 124
`\tkzDefCircle`, 82
`\tkzDefCircle`: arguments
 ($\langle pt1, pt2 \rangle$) or ($\langle pt1, pt2, pt3 \rangle$), 82
`\tkzDefCircle`: options
 K, 82
 apollonius, 82
 circum, 82
 diameter, 82
 euler or nine, 82
 ex, 82
 in, 82
 orthogonal through, 82
 orthogonal, 82
 spieker, 82
 through, 82
`\tkzDefCircle`[<命令选项>]($\langle A, B \rangle$) or ($\langle A, B, C \rangle$), 82
`\tkzDefEquiPoints`, 130
`\tkzDefEquiPoints`: arguments
 (pt1,pt2), 130
`\tkzDefEquiPoints`: options
 /compass/delta, 130
 dist, 130
 from=pt, 130
 show, 130
`\tkzDefEquiPoints`[<命令选项>]($\langle pt1, pt2 \rangle$), 130
`\tkzDefGoldRectangle`, 77
`\tkzDefGoldRectangle`: arguments
 ($\langle pt1, pt2 \rangle$), 77
`\tkzDefGoldRectangle`(\langle point,point \rangle), 77
`\tkzDefLine`, 53
`\tkzDefLine`: arguments

```

    ((pt1,pt2,pt3)), 53
    ((pt1,pt2)), 53
\tkzDefLine: options
    K, 53
    bisector out, 53
    bisector, 53
    mediator, 53
    normed, 53
    orthogonal=through..., 53
    parallel=through..., 53
    perpendicular=through..., 53
\tkzDefLine[⟨命令选项⟩]((pt1,pt2)) or ((pt1,pt2,pt3)), 53
\tkzDefMidPoint, 28
\tkzDefMidPoint: arguments
    (pt1,pt2), 28
\tkzDefMidPoint(⟨pt1,pt2⟩), 28
\tkzDefParallelogram, 75
\tkzDefParallelogram: arguments
    ((pt1,pt2,pt3)), 75
\tkzDefParallelogram(⟨pt1,pt2,pt3⟩), 75
\tkzDefPoint, 24, 26, 93
\tkzDefPoint: arguments
    ( $\alpha:d$ ), 24
    ( $x,y$ ), 24
    {名称}, 24
\tkzDefPoint: options
    label, 24
    shift, 24
\tkzDefPointBy, 38
\tkzDefPointBy: arguments
    pt, 38
\tkzDefPointBy: options
    homothety, 38
    inversion, 38
    projection , 38
    reflection, 38
    rotation in rad, 38
    rotation , 38
    symmetry , 38
    translation, 38
\tkzDefPointBy[⟨命令选项⟩](⟨pt⟩), 38
\tkzDefPointOnCircle, 37
\tkzDefPointOnCircle: options
    angle, 37
    center, 37
    radius, 37
\tkzDefPointOnCircle[⟨命令选项⟩], 37
\tkzDefPointOnLine, 36
\tkzDefPointOnLine: arguments
    pt1,pt2, 36
\tkzDefPointOnLine: options
    pos=nb, 36
\tkzDefPointOnLine[⟨命令选项⟩](⟨A,B⟩), 36
\tkzDefPoints{0/0/0,2/2/A}, 27
\tkzDefPoints, 27
\tkzDefPoints: arguments
     $x_i/y_i/n_i$ , 27
\tkzDefPoints: options
    shift, 27
\tkzDefPointsBy, 38, 42, 43
\tkzDefPointsBy: arguments
    ((需要变换点列表)){⟨变换后点列表⟩}, 43
\tkzDefPointsBy: options
    homothety = center #1 ratio #2, 43

```

```

projection = onto #1--#2, 43
reflection = over #1--#2, 43
rotation = center #1 angle #2, 43
rotation in rad = center #1 angle #2, 43
symmetry = center #1, 43
translation = from #1 to #2, 43
\tkzDefPointsBy[(命令选项)]((需要变换点列表))((变换后点列表)), 43
\tkzDefPoints[(命令选项)]{(x1/y1/n1, x2/y2/n2, ...)}, 27
\tkzDefPointWith, 44
\tkzDefPointWith: arguments
    (pt1, pt2), 44
\tkzDefPointWith: options
    K, 44
    colinear normed= at #1, 44
    colinear= at #1, 44
    linear normed, 44
    linear, 44
    orthogonal normed, 44
    orthogonal, 44
\tkzDefPointWith((pt1, pt2)), 44
\tkzDefPoint[(命令选项)]((x, y))((名称)) or ((α:d))((名称)), 24
\tkzDefRandPointOn, 49
\tkzDefRandPointOn: options
    circle =center pt1 radius dim, 49
    circle through=center pt1 through pt2, 49
    disk through=center pt1 through pt2, 49
    line=pt1--pt2, 49
    rectangle=pt1 and pt2, 49
    segment= pt1--pt2, 49
\tkzDefRandPointOn[(命令选项)], 49
\tkzDefRegPolygon, 81
\tkzDefRegPolygon: arguments
    ((pt1, pt2)), 81
\tkzDefRegPolygon: options
    Options TikZ, 81
    center, 81
    name, 81
    sides, 81
    side, 81
\tkzDefRegPolygon[(命令选项)]((pt1, pt2)), 81
\tkzDefShiftPoint, 26, 27
\tkzDefShiftPoint: arguments
    (α:d), 26
    (x, y), 26
\tkzDefShiftPoint: options
    [参考点], 26
\tkzDefShiftPoint[(参考点)]((x, y))((名称)) 或 ((α:d))((名称)), 26
\tkzDefSpcTriangle, 69
\tkzDefSpcTriangle: options
    centroid or medial, 69
    euler, 69
    ex or excentral, 69
    extouch, 69
    feuerbach, 69
    in or incentral, 69
    intouch or contact, 69
    name, 69
    orthic, 69
    tangential, 69
\tkzDefSpcTriangle[(命令选项)]((A, B, C)), 69
\tkzDefSquare, 74, 75
\tkzDefSquare: arguments
    ((pt1, pt2)), 74
\tkzDefSquare((pt1, pt2)), 74

```

```

\tkzDefTangent, 55
\tkzDefTangent: arguments
    ( $\langle pt1, pt2 \text{ or } \langle pt1, dim \rangle \rangle$ ), 55
\tkzDefTangent: options
    at=pt, 55
    from with R=pt, 55
    from=pt, 55
\tkzDefTangent[ $\langle \text{命令选项} \rangle$ ]( $\langle pt1, pt2 \rangle$ ) or ( $\langle pt1, dim \rangle$ ), 55
\tkzDefTriangle, 66
\tkzDefTriangle: options
    cheops, 66
    equilateral, 66
    euclide, 66
    golden, 66
    gold, 66
    pythagore, 66
    school, 66
    two angles= #1 and #2, 66
\tkzDefTriangleCenter, 31
\tkzDefTriangleCenter: arguments
    (pt1, pt2, pt3), 31
\tkzDefTriangleCenter: options
    centroid, 31
    circum, 31
    euler, 31
    ex, 31
    feuerbach, 31
    in, 31
    mittenpunkt, 31
    nagel, 31
    ortho, 31
    spieker, 31
    symmedian, 31
\tkzDefTriangleCenter[ $\langle \text{命令选项} \rangle$ ]( $\langle A, B, C \rangle$ ), 31
\tkzDefTriangle[ $\langle \text{命令选项} \rangle$ ]( $\langle A, B \rangle$ ), 66
\tkzDrawAngle, 154
\tkzDrawArc[angles] (O, A) (O, 90), 117
\tkzDrawArc[delta=10] (O, A) (B), 117
\tkzDrawArc[R with nodes] (O, 2 cm) (A, B), 117
\tkzDrawArc[R] (O, 2 cm) (30, 90), 117
\tkzDrawArc[rotate, color=red] (O, A) (90), 117
\tkzDrawArc, 117
\tkzDrawArc: options
    R with nodes, 117
    R, 117
    angles, 117
    delta, 117
    rotate, 117
    towards, 117
\tkzDrawArc[ $\langle \text{命令选项} \rangle$ ]( $\langle O, \dots \rangle$ )( $\langle \dots \rangle$ ), 117
\tkzDrawCircle, 82, 87
\tkzDrawCircle: arguments
    ( $\langle pt1, pt2 \rangle$ ), 87
\tkzDrawCircle: options
    R, 87
    diameter, 87
    through, 87
\tkzDrawCircles, 88
\tkzDrawCircles: arguments
    ( $\langle pt1, pt2 \text{ } pt3, pt4, \dots \rangle$ ), 88
\tkzDrawCircles: options
    R, 88
    diameter, 88
    through, 88

```

`\tkzDrawCircles`[<命令选项>]($(A,B\ C,D)$), 88
`\tkzDrawCircle`[<命令选项>]((A,B)), 87
`\tkzDrawGoldRectangle`, 77
`\tkzDrawGoldRectangle: arguments`
 ($\langle pt1,pt2 \rangle$), 77
`\tkzDrawGoldRectangle: options`
 Options TikZ, 77
`\tkzDrawGoldRectangle`[<命令选项>]($\langle point,point \rangle$), 77
`\tkzDrawLine`, 57
`\tkzDrawLine: options`
 add= nb1 and nb2, 57
 altitude, 57
 bisector, 57
 median, 57
 none, 57
`\tkzDrawLines`, 58
`\tkzDrawLines`[<命令选项>]($\langle pt1,pt2\ pt3,pt4,... \rangle$), 58
`\tkzDrawLine`[<命令选项>]($\langle pt1,pt2 \rangle$) or ($\langle pt1,pt2,pt3 \rangle$), 57
`\tkzDrawMedian`, 57
`\tkzDrawPoint`(A,B), 154
`\tkzDrawPoint`, 35
`\tkzDrawPoint: arguments`
 点的名称, 35
`\tkzDrawPoint: options`
 color, 35
 shape, 35
 size, 35
`\tkzDrawPoints`(A,B,C), 36
`\tkzDrawPoints`, 36, 154
`\tkzDrawPoints: arguments`
 点列表, 36
`\tkzDrawPoints: options`
 color, 36
 shape, 36
 size, 36
`\tkzDrawPoints`[<命令选项>]($\langle \text{点列表} \rangle$), 36
`\tkzDrawPoint`[<命令选项>]($\langle \text{名称} \rangle$), 35
`\tkzDrawPolygon`, 78
`\tkzDrawPolygon: arguments`
 ($\langle pt1,pt2,pt3,... \rangle$), 78
`\tkzDrawPolygon: options`
 Options TikZ, 78
`\tkzDrawPolygon`[<命令选项>]($\langle \text{点集列表} \rangle$), 78
`\tkzDrawPolySeg`, 78
`\tkzDrawPolySeg: arguments`
 ($\langle pt1,pt2,pt3,... \rangle$), 78
`\tkzDrawPolySeg: options`
 Options TikZ, 78
`\tkzDrawPolySeg`[<命令选项>]($\langle \text{点集列表} \rangle$), 78
`\tkzDrawSector`(O,A)(B), 112
`\tkzDrawSector`[R with nodes]($O,2\ cm$)(A,B), 112
`\tkzDrawSector`[R,color=blue]($O,2\ cm$)($30,90$), 112
`\tkzDrawSector`[rotate,color=red](O,A)(90), 112
`\tkzDrawSector`, 112–114
`\tkzDrawSector: options`
 R with nodes, 112
 R, 112
 rotate, 112
 towards, 112
`\tkzDrawSector`[<命令选项>]($\langle O,... \rangle$)($\langle ... \rangle$), 112
`\tkzDrawSegment`($A,B\ A,C$), 154
`\tkzDrawSegment`, 22, 60
`\tkzDrawSegment: arguments`
 ($pt1,pt2$), 60

- \tkzDrawSegment: options
 - TikZ 选项, 60
 - ..., 60
 - add, 60
 - dim, 60
- \tkzDrawSegments[color = gray,style=dashed]{B,B' C,C'}, 154
- \tkzDrawSegments, 62, 154
- \tkzDrawSegments[⟨命令选项⟩](⟨pt1,pt2 pt3,pt4,...⟩), 62
- \tkzDrawSegment[⟨命令选项⟩](⟨pt1,pt2⟩), 60
- \tkzDrawSemiCircle, 90
- \tkzDrawSemiCircle: arguments
 - (⟨pt1,pt2⟩), 90
- \tkzDrawSemiCircle: options
 - diameter, 90
 - through, 90
- \tkzDrawSemiCircle[⟨命令选项⟩](⟨A,B⟩), 90
- \tkzDrawSquare, 76
- \tkzDrawSquare: arguments
 - (⟨pt1,pt2⟩), 76
- \tkzDrawSquare: options
 - Options TikZ, 76
- \tkzDrawSquare[⟨命令选项⟩](⟨pt1,pt2⟩), 76
- \tkzDrawTriangle, 67
- \tkzDrawTriangle: options
 - cheops, 67
 - equilateral, 67
 - euclide, 67
 - golden, 67
 - gold, 67
 - pythagore, 67
 - school, 67
 - two angles= #1 and #2, 67
- \tkzDrawTriangle[⟨命令选项⟩](⟨A,B⟩), 67
- \tkzDuplicateLen, 121
- \tkzDuplicateLength, 121
- \tkzDuplicateSegment, 121, 122
- \tkzDuplicateSegment: arguments
 - (pt1,pt2)(pt3,pt4){pt5}, 121
- \tkzDuplicateSegment(⟨pt1,pt2⟩)(⟨pt3,pt4⟩){⟨pt5⟩}, 121
- \tkzFillAngle, 100, 137, 154
- \tkzFillAngle: options
 - size, 100
- \tkzFillAngles, 101
- \tkzFillAngles[⟨命令选项⟩](⟨A,O,B⟩)(⟨A',O',B'⟩) 等, 101
- \tkzFillAngle[⟨命令选项⟩](⟨A,O,B⟩), 100
- \tkzFillCircle, 82, 87, 90
- \tkzFillCircle: options
 - R, 90
 - radius, 90
- \tkzFillCircle[⟨命令选项⟩](⟨A,B⟩), 90
- \tkzFillPolygon, 80
- \tkzFillPolygon: arguments
 - (⟨pt1,pt2,...⟩), 80
- \tkzFillPolygon[⟨命令选项⟩](⟨点集列表⟩), 80
- \tkzFillSector(O,A)(B), 114
- \tkzFillSector[R with nodes](O,2 cm)(A,B), 114
- \tkzFillSector[R,color=blue](O,2 cm)(30,90), 114
- \tkzFillSector[rotate,color=red](O,A)(90), 114
- \tkzFillSector, 114, 115
- \tkzFillSector: options
 - R with nodes, 114
 - R, 114
 - rotate, 114
 - towards, 114

`\tkzFillSector[⟨命令选项⟩](⟨0,...⟩)(⟨...⟩)`, 114
`\tkzFindAngle`, 107, 108
`\tkzFindAngle: arguments`
 $(pt1, pt2, pt3)$, 107
`\tkzFindAngle(⟨pt1, pt2, pt3⟩)`, 107
`\tkzFindSlope`, 109
`\tkzFindSlope: arguments`
 $(pt1, pt2)pt3$, 109
`\tkzFindSlopeAngle`, 110, 111
`\tkzFindSlopeAngle: arguments`
 $(pt1, pt2)$, 110
`\tkzFindSlopeAngle(⟨A, B⟩)`, 110
`\tkzFindSlope(⟨pt1, pt2⟩){⟨宏名称⟩}`, 109
`\tkzGetAngle`, 107, 110
`\tkzGetAngle: arguments`
 宏名称, 107
`\tkzGetAngle(⟨宏名称⟩)`, 107
`\tkzGetFirstPoint{A}`, 154
`\tkzGetFirstPoint{Jb}`, 85
`\tkzGetFirstPoint`, 74
`\tkzGetFirstPointI`, 83
`\tkzGetLength`, 82, 124
`\tkzGetPoint(A)`, 154
`\tkzGetPoint{A}`, 154
`\tkzGetPoint{C}`, 44
`\tkzGetPoint{M}`, 38
`\tkzGetPoint`, 22, 28, 31–33, 44, 49, 53, 66, 69, 75, 82
`\tkzGetPointCoord`, 124
`\tkzGetPointCoord: arguments`
 $(point){⟨宏名称⟩}$, 124
`\tkzGetPointCoord(⟨A⟩){⟨宏名称⟩}`, 124
`\tkzGetPoints{A}{B}`, 154
`\tkzGetPoints{C}{D}`, 76
`\tkzGetPoints`, 53, 74, 77
`\tkzGetRandPointOn`, 22, 49
`\tkzGetSecondPoint{A}`, 154
`\tkzGetSecondPoint{Tb}`, 85
`\tkzGetSecondPoint`, 74
`\tkzGetSecondPointIb`, 83
`\tkzGetVectxy`, 47, 48
`\tkzGetVectxy: arguments`
 $(point){name\ of\ macro}$, 47
`\tkzGetVectxy(⟨A, B⟩){⟨text⟩}`, 47
`\tkzInit`, 14, 22, 23, 152
`\tkzInterCC`, 97
`\tkzInterCC: options`
 N, 97
 R, 97
 with nodes, 97
`\tkzInterCCN`, 97
`\tkzInterCCR`, 97
`\tkzInterCC[⟨命令选项⟩](⟨O, A⟩)(⟨O', A'⟩) 或 (⟨O, r⟩)(⟨O', r'⟩) 或 (⟨O, A, B⟩)(⟨O', C, D⟩)`, 97
`\tkzInterLC`, 93
`\tkzInterLC: options`
 N, 93
 R, 93
 with nodes, 93
`\tkzInterLC[⟨命令选项⟩](⟨A, B⟩)(⟨O, C⟩) 或 (⟨O, r⟩) 可 (⟨O, C, D⟩)`, 93
`\tkzInterLL`, 93
`\tkzInterLL(⟨A, B⟩)(⟨C, D⟩)`, 93
`\tkzLabelAngle`, 104
`\tkzLabelAngle: options`
 pos, 104
`\tkzLabelAngles`, 105

`\tkzLabelAngles`[<命令选项>]($\langle A, O, B \rangle$)($\langle A', O', B' \rangle$) 等, 105
`\tkzLabelAngle`[<命令选项>]($\langle A, O, B \rangle$), 104
`\tkzLabelCircle`, 82, 87, 92
`\tkzLabelCircle`: options
 R, 92
 radius, 92
`\tkzLabelCircle`[<命令选项>]($\langle A, B \rangle$)(<角度>){<标注>}, 92
`\tkzLabelLine`(A,B), 59
`\tkzLabelLine`, 22, 59, 60
`\tkzLabelLine`: arguments
 label, 59
`\tkzLabelLine`: options
 pos, 59
`\tkzLabelLine`[<命令选项>]($\langle pt1, pt2 \rangle$){<label>}, 59
`\tkzLabelSegment`(A,B){5}, 64
`\tkzLabelSegment`, 64
`\tkzLabelSegment`: arguments
 (pt1,pt2), 64
 label, 64
`\tkzLabelSegment`: options
 pos, 64
`\tkzLabelSegments`, 65
`\tkzLabelSegments`[<命令选项>]($\langle pt1, pt2 \ pt3, pt4, \dots \rangle$), 65
`\tkzLabelSegment`[<命令选项>]($\langle pt1, pt2 \rangle$){<label>}, 64
`\tkzLength`, 96
`\tkzMarkAngle`, 103, 137
`\tkzMarkAngle`: options
 arc, 103
 mark, 103
 mkcolor, 103
 mkpos, 103
 mksize, 103
 size, 103
`\tkzMarkAngles`, 104
`\tkzMarkAngles`[<命令选项>]($\langle A, O, B \rangle$)($\langle A', O', B' \rangle$) 等, 104
`\tkzMarkAngle`[<命令选项>]($\langle A, O, B \rangle$), 103
`\tkzMarkRightAngle`, 105
`\tkzMarkRightAngle`: options
 german, 105
 size, 105
`\tkzMarkRightAngles`, 107
`\tkzMarkRightAngles`[<命令选项>]($\langle A, O, B \rangle$)($\langle A', O', B' \rangle$) 等, 107
`\tkzMarkRightAngle`[<命令选项>]($\langle A, O, B \rangle$), 105
`\tkzMarkSegment`, 62
`\tkzMarkSegment`: options
 color, 62
 mark, 62
 pos, 62
 size, 62
`\tkzMarkSegments`, 63
`\tkzMarkSegments`[<命令选项>]($\langle pt1, pt2 \ pt3, pt4, \dots \rangle$), 63
`\tkzMarkSegment`[<命令选项>]($\langle pt1, pt2 \rangle$), 62
`\tkzProtractor`, 131
`\tkzProtractor`: options
 lw, 131
 return, 131
 scale, 131
`\tkzProtractor`[<命令选项>]($\langle O, A \rangle$), 131
`\tkzpttocm`, 123
`\tkzpttocm`: arguments
 (number) 宏名称, 123
`\tkzpttocm`(<nombre>){<宏名称>}, 123
`\tkzSaveBB`, 22
`\tkzSetUpCompass`, 126, 150, 151

- \tkzSetUpCompass: options
 - color, 126, 150
 - line width, 126, 150
 - style, 126, 150
- \tkzSetUpCompass[⟨命令选项⟩], 126, 150
- \tkzSetUpLine, 148
- \tkzSetUpLine: options
 - add, 148
 - color, 148
 - line width, 148
 - style, 148
- \tkzSetUpLine[⟨命令选项⟩], 148
- \tkzSetUpPoint, 149, 150
- \tkzSetUpPoint: options
 - color, 149
 - fill, 149
 - shape, 149
 - size, 149
- \tkzSetUpPoint[⟨命令选项⟩], 149
- \tkzShowBB, 152
- \tkzShowLine, 127, 128
- \tkzShowLine: options
 - K, 127
 - bisector, 127
 - gap, 127
 - length, 127
 - mediator, 127
 - orthogonal, 127
 - perpendicular, 127
 - ratio, 127
 - size, 127
- \tkzShowLine[⟨命令选项⟩](⟨pt1,pt2⟩) 或 (⟨pt1,pt2,pt3⟩), 127
- \tkzShowTransformation, 128, 129
- \tkzShowTransformation: options
 - K, 128
 - gap, 128
 - length, 128
 - projection=onto pt1--pt2, 128
 - ratio, 128
 - reflection= over pt1--pt2, 128
 - size, 128
 - symmetry=center pt, 128
 - translation=from pt1 to pt2, 128
- \tkzShowTransformation[⟨命令选项⟩](⟨pt1,pt2⟩) 或 (⟨pt1,pt2,pt3⟩), 128
- \tkzTangent, 55
- \usetkzobj{all}, 22
- \usetkztool, 22
- \Vx, 47
- \Vy, 47