

AlterMundus



著：Alain Matthes

译：耿楠 (陕西·杨凌)

January 1, 2021 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包,它主要用于绘制经典欧氏几何图形。**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,以通过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有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: 另外两种黄金三角形绘制方式	12
1.3.3	最小工作示例	13
1.4	tkz 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	经典示例	18
1.6.2	点集、计算、绘制、标记和标注	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	定义点	23
4.1	\tkzDefPoint命令: 定义命名点	24
4.1.1	笛卡尔坐标示例	25
4.1.2	使用xfp宏包实现计算	25
4.1.3	极坐标示例	25
4.1.4	坐标计算	26
4.1.5	相对位置点	26
4.2	\tkzDefShiftPoint命令: 定义平移点	26
4.2.1	用\tkzDefShiftPoint命令构造等腰三角形	27
4.2.2	用\tkzDefShiftPoint命令构造等边三角形	27
4.2.3	用\tkzDefShiftPoint命令构造平行四边形	27
4.3	\tkzDefPoints命令: 定义点集	27
4.3.1	用\tkzDefPoints命令构造三角形	28
4.3.2	用\tkzDefPoints命令构造正方形	28
5	定义特殊点	28
5.1	\tkzDefMidPoint命令: 定义线段中点	28
5.1.1	\tkzDefMidPoint命令示例	28
5.2	重心坐标	29
5.2.1	用\tkzDefBarycentricPoint命令计算 2 个点的重心	29
5.2.2	用\tkzDefBarycentricPoint命令计算 3 个点的重心	29
5.3	相似中心	30
6	三角形特殊点	31
6.1	\tkzDefTriangleCenter命令: 定义三角形特殊点	31
6.1.1	ortho或orthic选项: 垂心	31
6.1.2	centroid选项: 重心	32
6.1.3	circum选项: 外心	32
6.1.4	in选项: 内心	32
6.1.5	ex选项: 旁心	33
6.1.6	euler选项: 欧拉点	33
6.1.7	symmedian选项: 陪位重心	33
6.1.8	nagel选项: 界心 (Nagel 点)	34
6.1.9	mittenpunkt选项	35
7	绘制点	35
7.1	\tkzDrawPoint命令: 绘制点	35
7.1.1	点绘制示例	35
7.2	\tkzDrawPoints命令: 绘制多个点	36
7.2.1	示例 1	37

7.2.2	示例 2	37
8	直线或圆上的点	37
8.1	<code>\tkzDefPointOnLine</code> 命令: 定义直线上的点	37
8.1.1	<code>pos</code> 选项	38
8.2	<code>\tkzDefPointOnCircle</code> 命令: 定义圆上的点	38
9	利用坐标变换定义点	39
9.1	<code>\tkzDefPointBy</code> : 通过变换定义一个点	39
9.2	变换示例	40
9.2.1	平移示例	40
9.2.2	轴对称示例	40
9.2.3	<code>homothety</code> 和 <code>projection</code> 示例	41
9.2.4	投影示例	41
9.2.5	中心对称示例	42
9.2.6	旋转示例(度)	42
9.2.7	旋转示例(弧度)	42
9.2.8	反转示例	43
9.2.9	点的反转: 正交圆	43
9.3	<code>\tkzDefPointsBy</code> 命令: 通过变换定义多个点	43
9.3.1	变换示例	44
10	通过向量定义点	45
10.1	<code>\tkzDefPointWith</code> 命令: 定义向量点	45
10.1.1	<code>colinear at</code> 选项示例	45
10.1.2	<code>colinear at</code> 带 K 选项示例	46
10.1.3	<code>colinear at</code> 带 $K = \frac{\sqrt{2}}{2}$ 选项示例	46
10.1.4	<code>orthogonal</code> 选项示例	46
10.1.5	<code>orthogonal</code> 带 $K = -1$ 选项示例	46
10.1.6	<code>orthogonal</code> 选项综合示例	47
10.1.7	<code>colinear</code> 和 <code>orthogonal</code> 选项示例	47
10.1.8	<code>orthogonal normed</code> 带 $K = 1$ 选项示例	47
10.1.9	<code>orthogonal normed</code> 和 $K = 2$ 选项示例	48
10.1.10	<code>linear</code> 选项示例	48
10.1.11	<code>linear normed</code> 选项示例	48
10.2	<code>\tkzGetVectxy</code> 命令: 获取向量坐标分量	48
10.2.1	使用 <code>\tkzGetVectxy</code> 命令实现坐标变换	49
11	定义随机点	50
11.1	<code>\tkzDefRandPointOn</code> 命令: 定义随机点	50
11.2	矩形内的随机点	50
11.3	线段上的随机点	51
11.4	直线上的随机点	51
11.4.1	随机点综合示例	51
11.5	圆上的随机点	52
11.5.1	Apollonius 圆的随机示例	52
11.5.2	线段中点	53
12	直线	54
12.1	<code>\tkzDefLine</code> 命令: 定义直线	54
12.1.1	<code>mediator</code> 选项示例	54
12.1.2	<code>bisector</code> 和 <code>normed</code> 选项示例	55
12.1.3	<code>orthogonal</code> 和 <code>parallel</code> 选项示例	55
12.1.4	循环画图	55
12.1.5	抛物线	56
12.2	<code>\tkzDefTangent</code> 命令: 定义圆的切线	56
12.2.1	圆上指定点的切线示例	56
12.2.2	过圆外指定点的切线示例	57
12.2.3	Andrew Mertz 示例	57

12.2.4	from with R 和 at 选项示例	57
12.2.5	from 选项示例	58
13	直线绘制和命名	58
13.1	\tkzDrawLine 命令: 绘制直线	58
13.1.1	add 选项示例	59
13.1.2	\tkzDrawLines 命令示例	59
13.1.3	add 选项示例	59
13.1.4	三角形的中线示例	60
13.1.5	三角形的高示例	60
13.1.6	三角形的角平分线示例	60
13.2	\tkzLabelLine 命令: 添加直线标注	60
13.2.1	\tkzLabelLine 命令示例	61
14	绘制和标记线段	61
14.1	\tkzDrawSegment 命令: 绘制线段	61
14.1.1	简单示例	61
14.1.2	使用 add 选项延伸线段	62
14.1.3	使用 dim 选项标注尺寸	62
14.2	\tkzDrawSegments 命令: 绘制多条线段	63
14.2.1	为线段添加箭头	63
14.3	\tkzMarkSegment 命令: 标记线段	63
14.3.1	几种标记示例	64
14.3.2	mark 选项示例	64
14.4	\tkzMarkSegments 命令: 标记多条线段	64
14.4.1	标记等腰三角形示例	64
14.4.2	其它标记示例	65
14.5	\tkzLabelSegment 命令: 标注线段	65
14.5.1	多个标注示例	65
14.5.2	直角三角形的标注和标记示例	66
14.6	\tkzLabelSegments 命令: 标注多条线段	66
14.6.1	等腰三角形的标注示例	66
15	定义与绘制三角形	67
15.1	\tkzDefTriangle 命令: 定义三角形	67
15.1.1	golden 选项示例	67
15.1.2	equilateral 选项示例	68
15.1.3	gold 或 euclide 选项示例	68
15.2	\tkzDrawTriangle 命令: 绘制三角形	68
15.2.1	pythagore 选项示例	69
15.2.2	school 选项示例	69
15.2.3	golden 选项示例	69
15.2.4	gold 选项示例	69
15.2.5	euclide 选项示例	69
15.3	\tkzDefSpcTriangle 命令: 定义特殊三角形	70
15.3.1	medial 或 centroid 选项示例	70
15.3.2	in 或 incentral 选项示例	71
15.3.3	ex 或 excentral 选项示例	71
15.3.4	intouch 选项示例	72
15.3.5	extouch 选项示例	72
15.3.6	feuerbach 选项示例	73
15.3.7	tangential 选项示例	73
15.3.8	euler 选项示例	74
16	多边形定义	75
16.1	\tkzDefSquare 命令: 定义正方形	75
16.1.1	通过两个点定义正方形	75
16.1.2	绘制等腰直角三角形	75
16.1.3	绘制 Pythagorean 定理示意图	76

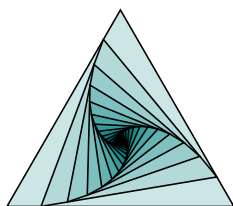
16.2	<code>\tkzDefParallelogram</code> 命令：定义平行四边形第4个顶点	76
16.2.1	平行四边形定义示例	76
16.2.2	黄金矩形示例	77
16.3	<code>\tkzDrawSquare</code> 命令：绘制正方形	77
16.3.1	在半圆内绘制两个正方形示例	78
16.4	<code>\tkzDefGoldRectangle</code> 命令：定义黄金矩形	78
16.5	<code>\tkzDrawGoldRectangle</code> 命令：绘制黄金矩形	78
16.5.1	黄金矩形示例	78
16.6	<code>\tkzDrawPolygon</code> 命令：绘制多边形	79
16.6.1	<code>\tkzDrawPolygon</code> 命令示例	79
16.7	<code>\tkzDrawPolySeg</code> 命令：绘制多边形顶点折线	79
16.7.1	多边形顶点折线示例	80
16.7.2	多边形顶点折线：循环实现	80
16.8	<code>\tkzClipPolygon</code> 命令：使用多边形裁剪	80
16.8.1	<code>\tkzClipPolygon</code> 命令示例	80
16.8.2	使用“裁剪”将 Sangaku 图形限制在正方形内	81
16.9	<code>\tkzFillPolygon</code> 命令：多边形着色	81
16.9.1	<code>\tkzFillPolygon</code> 命令示例	81
16.10	<code>\tkzDefRegPolygon</code> 命令：定义正多边形	82
16.10.1	<code>center</code> 选项示例	82
16.10.2	<code>side</code> 选项示例	82
17	圆	83
17.1	<code>\tkzDefCircle</code> 命令：定义圆	83
17.1.1	使用随机点与 <code>through</code> 选项示例	84
17.1.2	<code>diameter</code> 选项示例	84
17.1.3	三角形的内切圆和外接圆示例	84
17.1.4	<code>ex</code> 选项示例	85
17.1.5	<code>euler</code> 选项示例	85
17.1.6	<code>apollonius</code> 选项示例	86
17.1.7	<code>ex</code> 选项示例	86
17.1.8	<code>spieker</code> 选项示例	87
17.1.9	<code>orthogonal through</code> 选项示例	87
17.1.10	指定圆心的另一个圆的正交圆示例	87
18	圆的绘制和标注	88
18.1	<code>\tkzDrawCircle</code> 命令：绘制圆	88
18.1.1	绘制一个圆并对其进行着色	88
18.2	<code>\tkzDrawCircles</code> 命令：绘制多个圆	89
18.2.1	通过三角形定义圆示例	89
18.2.2	同心圆示例	90
18.2.3	旁切圆示例	90
18.2.4	心形线示例	90
18.3	<code>\tkzDrawSemiCircle</code> 命令：绘制半圆	91
18.3.1	示例	91
18.4	<code>\tkzFillCircle</code> 命令：给圆着色	91
18.4.1	sangaku 圆示例	92
18.5	<code>\tkzClipCircle</code> 命令：用圆裁剪	92
18.5.1	示例	92
18.6	<code>\tkzLabelCircle</code> 命令：为圆添加标注	93
18.6.1	标注示例	93
19	交点	94
19.1	<code>\tkzInterLL</code> 命令：求两条直线的交点	94
19.1.1	直线交点示例	94
19.2	<code>\tkzInterLC</code> 命令：定义一条直线和一个圆的交点	94
19.2.1	直线与圆的交点示例	95
19.2.2	直线与圆的交点复杂示例	95
19.2.3	由圆心和半径定义圆示例	96

19.2.4	更为复杂的示例	96
19.3	半径计算	96
19.3.1	半径计算示例 1	96
19.3.2	半径计算示例 2	97
19.3.3	半圆中的矩形示例	97
19.3.4	“with nodes”选项示例	98
19.4	<code>\tkzInterCC</code> 命令：求两个圆的交点	98
19.4.1	构造等边三角形示例	98
19.4.2	求中点示例	99
19.4.3	求等腰三角形示例	99
19.4.4	三等分线段示例	100
19.4.5	with nodes 选项示例	100
20	角	101
20.1	<code>\tkzFillAngle</code> 命令：填充角	101
20.1.1	size 选项示例	101
20.1.2	改变点的顺序示例	101
20.2	<code>\tkzFillAngles</code> 命令：填充多个角	102
20.2.1	填充多个角示例	102
20.3	标记角	102
20.4	<code>\tkzMarkAngle</code> 命令：标记角	104
20.4.1	mark = x 选项示例	104
20.4.2	mark = 选项示例	105
20.5	<code>\tkzMarkAngles</code> 命令：标记多个角	105
20.6	<code>\tkzLabelAngle</code> 命令：标注角	105
20.6.1	pos 选项示例	105
20.7	<code>\tkzLabeAngles</code> 命令：标注多个角	106
20.8	<code>\tkzMarkRightAngle</code> 命令：标记直角	106
20.8.1	直角标记示例	106
20.8.2	使用 german 样式添加直角标记	107
20.8.3	混合样式	107
20.8.4	完整示例	107
20.9	<code>\tkzMarkRightAngles</code> 命令：标记多个直角	108
21	角度测量命令	108
21.1	<code>\tkzGetAngle</code> 命令：计算角度	108
21.1.1	<code>\tkzGetAngle</code> 命令示例	108
21.2	<code>\tkzFindAngle</code> 命令：计算三个点定义的角的角度	108
21.2.1	角度测量示例	109
21.3	角度计算示例	109
21.3.1	三角形内角计算示例	110
21.4	<code>\tkzFindSlope</code> 命令：计算斜率	110
21.5	<code>\tkzFindSlopeAngle</code> 命令：计算直线与横轴夹角	111
21.5.1	折叠示例	111
21.5.2	中点计算示例	112
22	扇形	113
22.1	<code>\tkzDrawSector</code> 命令：绘制扇形	113
22.1.1	towards 选项示例	113
22.1.2	rotate 选项示例	114
22.1.3	R 选项示例	114
22.1.4	R 选项示例	114
22.1.5	R with nodes 选项示例	115
22.2	<code>\tkzFillSector</code> 命令：填充扇形	115
22.2.1	towards 选项示例	116
22.2.2	rotate 选项示例	116
22.3	<code>\tkzClipSector</code> 命令：扇形裁剪	116
22.3.1	示例	117

23	圆弧	118
23.1	<code>\tkzDrawArc</code> 命令: 绘制圆弧	118
23.2	<code>towards</code> 选项示例 1	118
23.3	<code>towards</code> 选项示例 2	119
23.4	<code>rotate</code> 选项示例	119
23.5	<code>R</code> 选项示例	119
23.6	<code>R with nodes</code> 选项示例	120
23.7	<code>delta</code> 选项示例	120
23.8	<code>angles</code> 选项示例 1	121
23.9	<code>angles</code> 选项示例 2	121
24	杂项命令	122
24.1	<code>\tkzDuplicateSegment</code> 命令: 复制线段	122
24.1.1	黄金分割示例	123
24.2	<code>\tkzCalcLength</code> 命令: 计算线段长度	123
24.2.1	绘制尺规标记	124
24.3	<code>\tkzptocm</code> 命令: 将 pt 转换为 cm	124
24.4	<code>\tkzcmtopt</code> 命令: 将 cm 转换为 pt	124
24.4.1	示例代码	125
24.5	<code>\tkzGetPointCoord</code> 命令: 提取点的坐标分量	125
24.5.1	点定义示例	125
24.5.2	求向量和示例	125
25	尺规标记	126
25.1	<code>\tkzCompass</code> 命令: 绘制尺规标记	126
25.1.1	<code>length</code> 选项示例	126
25.1.2	<code>delta</code> 选项示例	126
25.2	<code>\tkzCompassss</code> 命令: 绘制多个尺规标记	126
25.3	<code>\tkzSetUpCompass</code> 命令: 设置尺规标记样式	127
25.3.1	示例	127
25.4	<code>\tkzShowLine</code> 命令: 显示直线尺规标记	128
25.4.1	<code>parallel</code> 选项示例	128
25.4.2	<code>perpendicular</code> 选项示例	128
25.4.3	<code>bisector</code> 选项示例	129
25.4.4	<code>mediator</code> 选项示例	129
25.5	<code>\tkzShowTransformation</code> 命令: 显示部分变换过程尺规标记	129
25.5.1	示例 1	130
25.5.2	示例 2	130
26	差分点	131
26.1	<code>\tkzDefEquiPoints</code> 命令: 定义一条直线上与给定点距离相等的两个点	131
26.1.1	示例	131
27	量角器	132
27.1	<code>\tkzProtractor</code> 命令: 绘制量角器	132
27.2	正向圆量角器	132
27.3	反向圆量角器	132
28	实例分析	133
28.1	一些有趣的例子	133
28.1.1	相似等腰三角形	133
28.1.2	“Tangente”方法	134
28.1.3	“Le Monde”方法	135
28.1.4	三角形的高	136
28.1.5	三角形的高 - 另一种构造方式	136
28.2	其他作者提供的实例	137
28.2.1	整数的算术平方根	137
28.2.2	直角三角	137
28.2.3	阿基米德等分	137

28.2.4	示例 (Dimitris Kapeta)	138
28.2.5	证明示例 1(John Kitzmiller)	139
28.2.6	证明示例 2(John Kitzmiller)	140
28.2.7	证明示例 3(John Kitzmiller)	141
28.2.8	证明示例 4(John Kitzmiller)	142
28.2.9	构图示例 1(Indonesia)	143
28.2.10	构图示例 2(Indonesia)	144
28.2.11	三个相切圆	146
28.2.12	APOLLONIUS 圆	147
29	个性化设置	149
29.1	<code>\tkzSetUpLine</code> 命令: 设置线条样式	149
29.1.1	改变线宽示例	149
29.1.2	改变线型示例	150
29.1.3	线段延伸示例	150
29.2	<code>\tkzSetUpPoint</code> 命令: 设置点样式	150
29.2.1	示例 1	151
29.2.2	示例 2	151
29.3	<code>\tkzSetUpCompass</code> 命令: 设置尺规标记样式	151
29.3.1	示例 1	151
29.3.2	示例 2	152
29.4	局部样式设置	152
30	<code>tkz-base</code> 小结	153
30.1	<code>tkz-base</code> 宏包工具	153
30.2	<code>\tkzInit</code> 命令和 <code>\tkzShowBB</code> 命令	153
30.3	<code>\tkzClip</code> 命令	153
30.3.1	<code>\tkzClip</code> 命令和 <code>space</code> 选项示例	153
31	FAQ	155
31.1	常见错误	155
	Index	156

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设计一个 \LaTeX 绘图工具，以实现欧氏几何图形的快速绘制，而不需要费力学习和掌握一门新的绘图语言。当然，**tkz-euclide** 适用于所有使用 \LaTeX 的教师，以实现在 \LaTeX 中轻松、正确地欧氏几何绘图。

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。为描述一个几何图形，须定义图形对象以及对这些对象所执行的操作。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。

虽然这会使语法比较冗长，但却更容易理解和使用。因此，用户能够轻松使用该宏包提供的命令进行绘图。

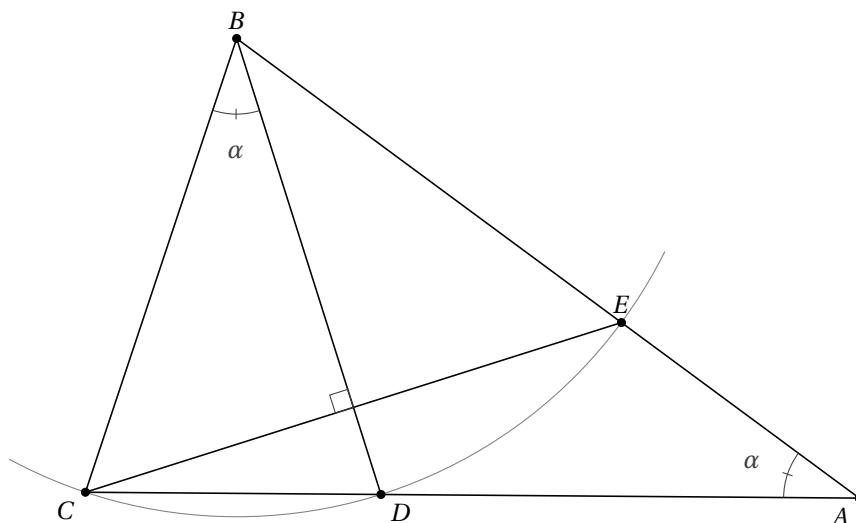
1.2 tkz-euclide与TikZ

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

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

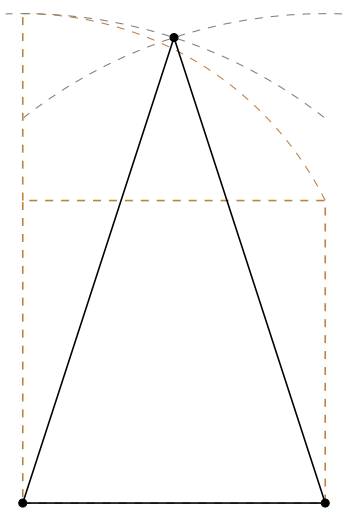
4. 从而可得

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

所以， CBD 是一个“黄金”三角形。

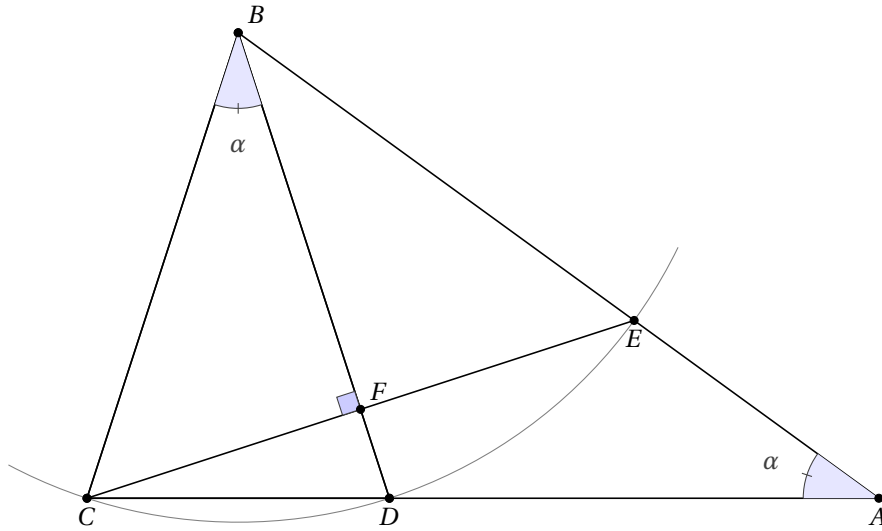
如何构造一个黄金三角形或者顶角为 36° 的等腰三角形呢？

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：另外两种黄金三角形绘制方式

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

```

\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:

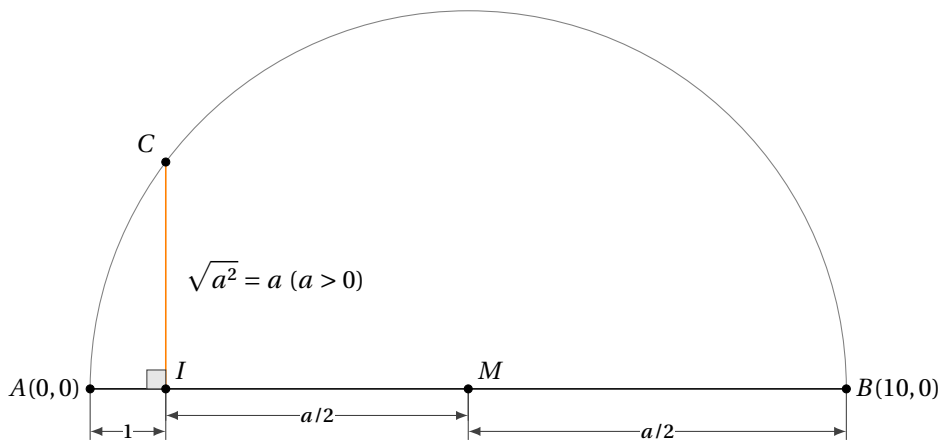
```

\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`之前载入该宏包，也就是在`TikZ`宏包之前载入。否则可能会与`TikZ`宏包冲突，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部分是定义已知点：多数情况下，并不需要使用`\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点：`B`点位于从`A`开始的10 cm处。`M`是线段`[AB]`的中点，然后得到通过`I`点的直线`(AB)`的正交直线。从而得到该正交线与以`M`为圆心，过`A`点的半圆的交点。

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

– 第3部分是图形绘制：

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

– 第4部分是绘制标记：

```
\tkzMarkRightAngle(A,I,C)
```

– 最后是布置标注：

```
\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){$\sqrt{a^2}=a \ (a>0)$}
```

– 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,}] (A,M)
\tkzDrawSegment[dim={a/2$, -16pt,}] (M,B)
\tkzDrawPoints(I,A,B,C,M)
% 标记
\tkzMarkRightAngle[ra](A,I,C)
% 标注
\tkzLabelPoint[left](A){$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 的基本要素

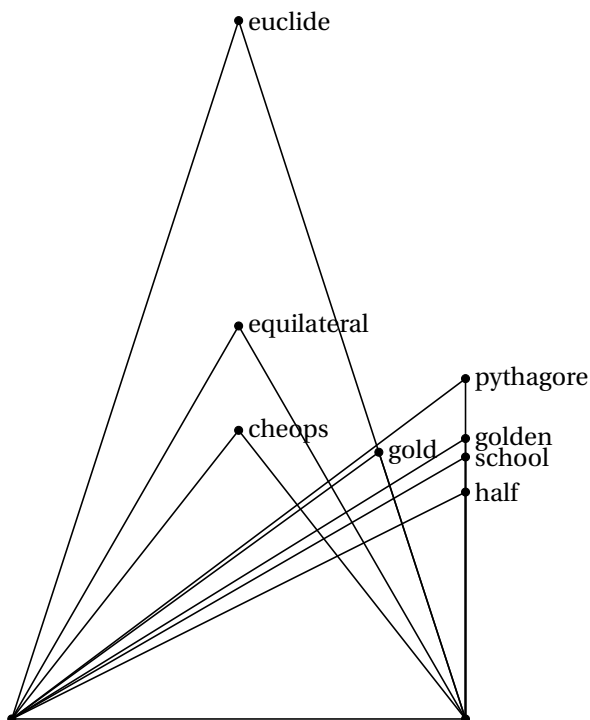
tkz-euclide 绘图中的基本要素是点，可以在任何时候通过定义一个点时命名该点，并通过该点的名称引用一个点（当然，也可以在后续操作中为点赋予不同的名称）。

通常，**tkz-euclide** 的命令都是以 **tkz** 为前缀，其主要四类前缀是：**\tkzDef...**、**\tkzDraw...**、**\tkzMark...** 和 **\tkzLabel...**。

在第 1 类命令中，**\tkzDefPoint** 命令用于通过坐标定义一个点，该命令的细节过后会进行讨论。在此，首先详细讨论 **\tkzDefTriangle** 命令。

该命令利用一个点与一对已知点的关系，通过已知类型的三角形来定义这个点。可以通过 **tkzPointResult** 使用该点，当然，也可以通过 **\tkzGetPoint{C}** 保存并为该点命名，以便后续引用该点。命令中的圆括号用于传递参数，例如在 (A,B) 中的 A 和 B 是已知点，用于根据三角形类型定义第三个点。

然而，在 **\tkzGetPoint{C}** 命令中，用大括号中的参数命名新点，三角形类型有：**equilateral**、**half**、**pythagoras**、**school**、**golden** 或 **sublime**、**euclidean**、**gold**、**cheops**... 和 **two angles**，例如：**\tkzDefTriangle[euclidean](A,B)**
\tkzGetPoint{C}



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoints{0/0/A,8/0/B}
  \foreach \tr in {equilateral,half,pythagore,%
    school,golden,euclidean, 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

该宏包选用法国的几何符号和作者习惯描述几何图形，**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欧氏几何图形。

根据欧几里得的定义，仅使用这些基本图形就可以构成各种复杂图形。因此，一个点并没有具体的尺寸，它在现实中不存在。同样，一条线也没有宽度，也不存在。在 **tkz-euclide** 宏包中需要考虑的是代表理想数学概念的对象，**tkz-euclide** 遵循古希腊尺规作图的方式进行绘图。

以下是使用的基本符号：


- 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。

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

对于一个点，可以使用类似 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 可以表示圆心、旋转中心等；
- M 可以表示中点；
- H 可以表示高；
- P' 可以表示变换后，点 P 的镜像点；

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

	<pre> \begin{tikzpicture} \tkzDefPoints{0/0/A} \tkzDrawPoints(A) \tkzLabelPoint(A){P} \end{tikzpicture} </pre>
--	--

注意也有例外情况：一些如三角形各边的中点，应带有边的特征，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 的形式命名并引用这些点。

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，点 A 和点 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，例如对于直线 (AB) 和直线 (CD) ，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 $EFGH$ 。
- 角度的单位是度 (例如： 60°)，对于等边三角形 ABC ，可以表示为 $\widehat{ABC} = \widehat{B} = 60^\circ$ 。
- 圆弧用起止点表示，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，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。
- 三角形中的特殊线有：内角角平分线、外角角平分线等。
- (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 经典示例

在此，以绘制一个等边三角形为例，展示tkz-euclide宏包的正确使用方式。当然，绘制该图形可以有多种方式，本例中将遵循欧氏几何尺规绘图步骤。

- 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}
```

- 然后载入tkz-euclide宏包：

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

注意，由于tkz-euclide宏包是基于TikZ宏包开发的，会同时载入该宏包，因此，无需再次载入TikZ宏包

- ~~\usetkzobj{all}~~，在3.03版以后，无需再使用该命令载入绘图对象，默认情况下会载入所有对象。
- 开始文档，并使用tikzpicture环境绘制欧氏几何图形：

```
\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) 表示以 A 点为圆心通过 B 点， (B,A) 表示以 B 点为圆心通过 A 点，两个圆共用这 A 点和 B 点。

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

得到两个圆的交点，并命名为 C 和 D

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

- 至此，便完成了所有点的定义，接下来进行绘图。

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

- 绘制 A 、 B 、 C 和 D ：

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

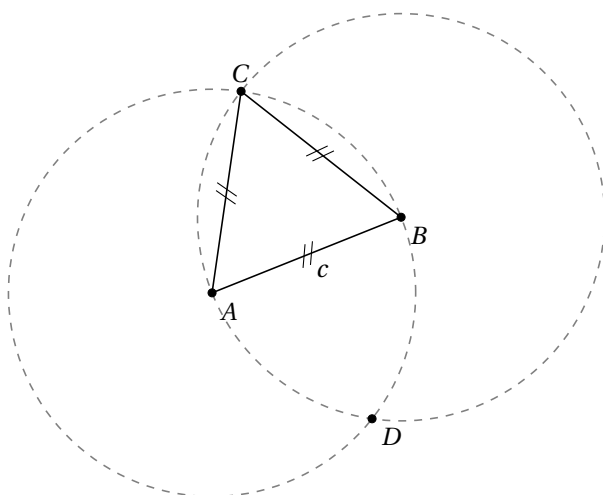
- 绘制标注，在绘制标注时，可以为其指定位置参数。

```
\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 点集、计算、绘制、标记和标注

该标题的含义是：计算与绘制分离

在使用 \LaTeX 排版时，源代码分为导言和正文两大部分。通过这种方式，可以将排版内容进行结构化设计，并通过样式和排版命令集简化用户的排版过程。**tkz-euclide**正是基于这种内容与格式分离思想进行设计的，以简化用户绘图过程。

首先定义已知点，然后计算其他点，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复制到指定目录。并注意以下几点：

- \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`
- `tkz-obj-eu-points-with.tex`

¹ `tkz-base`和`tkz-euclide` 已被`TeXLive`发行版收录，可通过`tlmgr`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.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

宏包更新后，增强了语法一致性。尤其是将点的定义与点的计算分离后，为后续绘图、标记和标注操作带来了方便。在未来，将更好地分离宏定义，并引入 **Lua**，以方便坐标计算。

一个重要的新特性是使用 **xfp** 宏包替换了 **fp** 宏包，从而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：

- 优化了代码并修复了部分 Bug;
- 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;
- 在各个命令中单独处理包围盒，从而避免使用 `\tkzInit` 命令和 `\tkzClip` 命令
- 增加了 `\tkzSaveBB`、`\tkzClipBB` 等包围盒处理命令;
- 逻辑上，由于所有命令都可以直接使用 TikZ 选项，所以尽可能移除了如 “label options” 等 “重复” 选项;
- 用 **tkz-euclide** 宏包生成随机点，并且使用 `\tkzDefRandPointOn` 命令代替了 `\tkzGetRandPointOn` 命令，并要求必须使用 `\tkzGetPoint` 引用生成的随机点;
- 新增 `\tkzLabelLine` 命令，用于为直线添加标注。删除了原来的直线标注选项 `end` 和 `start`。
- 新增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命令有：
 - 用于跟踪和创建三角形各边中点的命令 `\tkzDrawMedians`，不符合计算和绘制分离的思想，因此，建议首先使用 `\tkzSpcTriangle[median]` 命令计算中点，然后使用 `\tkzDrawSegments` 或 `\tkzDrawLines` 绘制。
 - 将 `\tkzDrawMedians(A,B)(C)` 命令改为 `\tkzDrawMedians(A,C,B)`，以定义中点 *C*;
 - `\tkzDrawTriangle[equilateral]` 命令将被删除，建议先用 `\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，其计算速度非常慢。
- 引入了 `\usetkztool` 命令，以加载新的 “tools” 库。

4 定义点

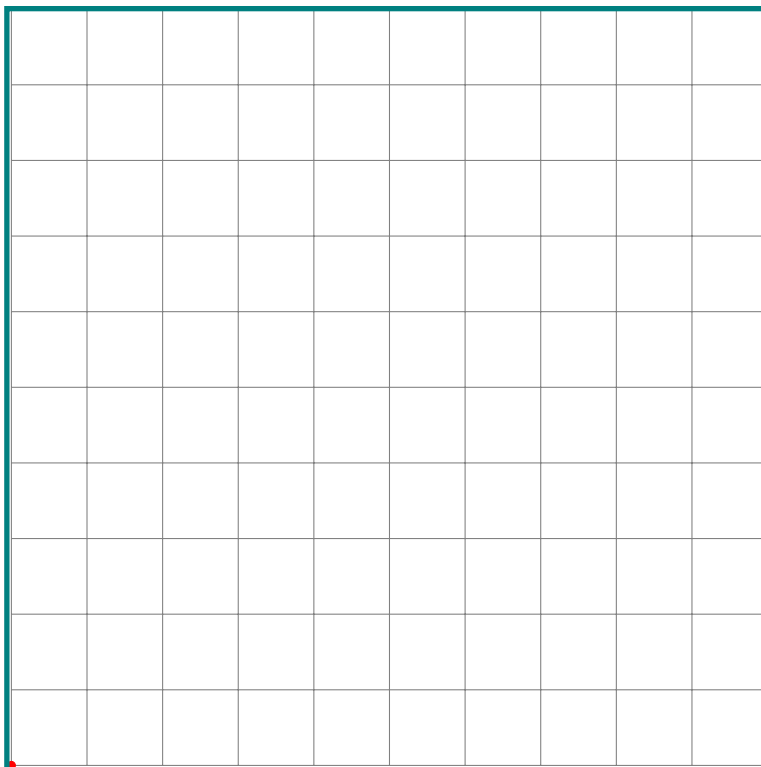
可以通过如下方式定义一个点:

- 笛卡尔坐标;
- 极坐标;
- 为点命名;
- 相对位置.

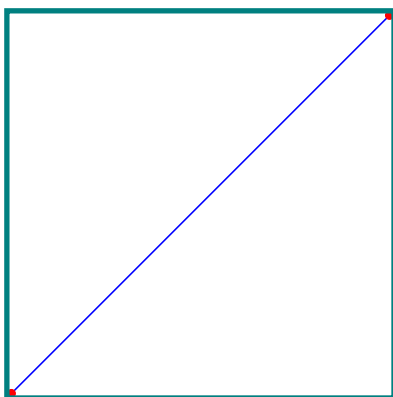
虽然仅用坐标就可以定义一个点, 但建议为点进行命名。如果一个名称唯一地与一个十进制数对应, 则可作为该点的名称, 由于正交笛卡尔坐标系定义了一个平面, 其坐标轴是正交的, 因此, 可以用 (x,y) 或 $(a:d)$ 表示该平面中的一个点, 其中 x 是横坐标, y 是纵坐标; a 是角度, d 是距离, 其度量单位等于 1 cm 或 0.39370 in。

现在, 默认情况下, 如果使用网格或坐标轴, 则可由 (0,0) 和 (10,10) 两个坐标定义一个矩形。

可以用 `tkz-base` 宏包的 `\tkzInit` 命令创建该矩形, 请参阅以下两个代码及其绘制结果, 以体会这些命令的作用。



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

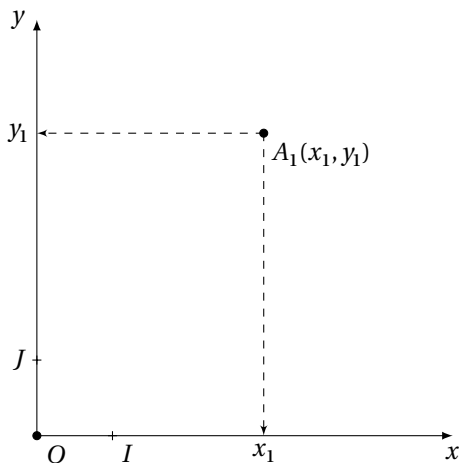


```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(5,5){A}
\tkzDrawSegment[blue](O,A)
\tkzDrawPoints[red](O,A)
\tkzShowBB[line width=2pt,teal]
\end{tikzpicture}
```

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

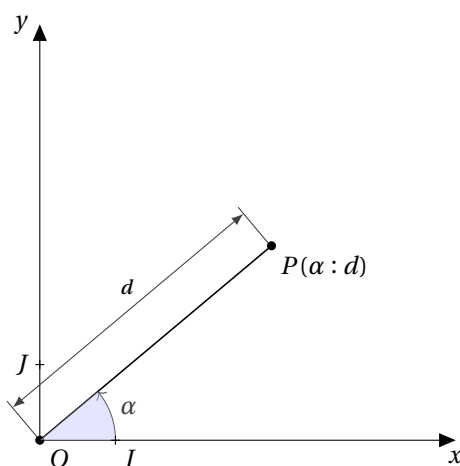
极坐标中表示一个点则需要一个角度 α (度) 和一个从原点度量的距离 d (默认单位是 cm)。

笛卡尔坐标



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

极坐标



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

可以用`\tkzDefPoint`命令，通过指定坐标定义一个点。该命令基于TikZ的`\coordinate`命令实现，因此，可在该命令中使用类似`shift`的TikZ的选项，该命令使用`xfp`宏包实现计算。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

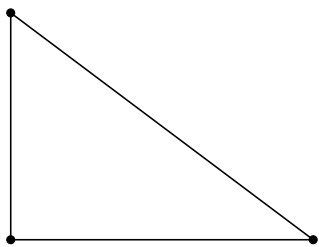
`\tkzDefPoint[< 命令选项>](<x,y>){< 名称>} or (<alpha:d>){< 名称>}`

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 ， $P1$ ，...

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，极坐标表示角度和距离。

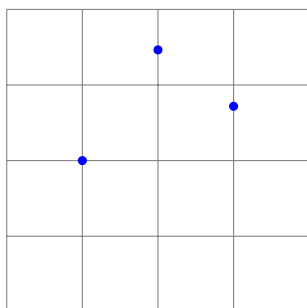
选项	默认值	含义
label	无	按预设的距离添加标注
shift	无	为 (x,y) 或 $(\alpha:d)$ 添加坐标偏移

4.1.1 笛卡尔坐标示例



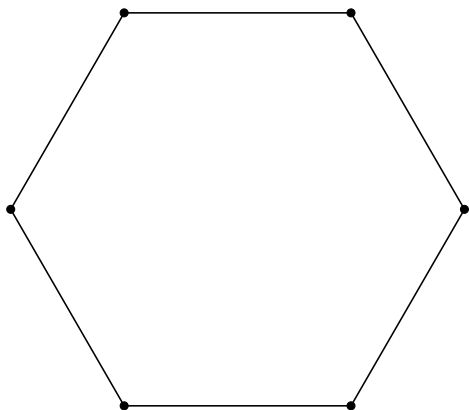
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(0,3){C}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

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}
```

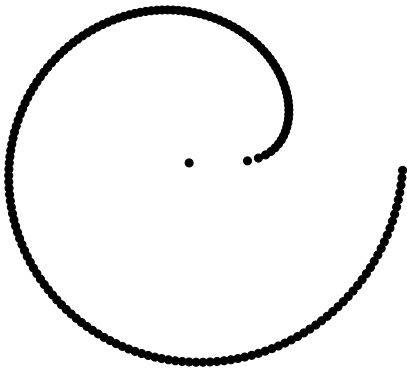
4.1.3 极坐标示例



```
\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 坐标计算

在计算坐标时，需遵循`xfp`宏包语法。另外，如使用`pgfmath`库计算，则必须在使用`\tkzDefPoint`命令前完成计算。

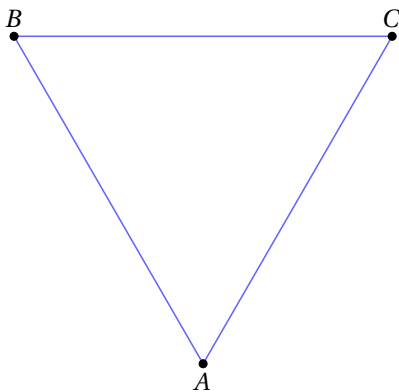


```
\begin{tikzpicture}[scale=.5]
\foreach \an [count=\i] in {0,2,...,358}
{\tkzDefPoint(\an:sqrt(sqrt(\an mm)))}{A_\i}}
\tkzDrawPoints(A_1,A_...A_180)
\end{tikzpicture}
```

4.1.5 相对位置点

可以使用相对位置定义一个点，此时，需使用 TikZ 的`scope`环境。

下面的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}
```

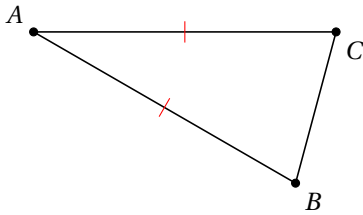
4.2 \tkzDefShiftPoint 命令：定义平移点

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

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

4.2.1 用\tkzDefShiftPoint命令构造等腰三角形

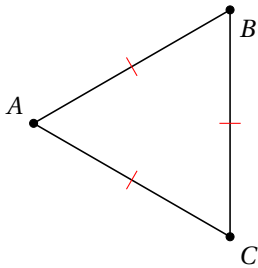
该命令允许基于一个点定义另一个点，等价于点的平移。下面的代码给出了一种通过点 A 和 30° 顶角等腰三角形的构造方法。



```
\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 用\tkzDefShiftPoint命令构造等边三角形

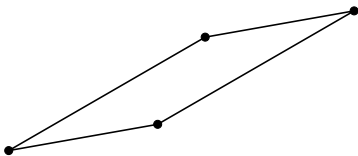
下面的代码给出了一种极为简捷的等边三角形构造方法。



```
\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 用\tkzDefShiftPoint命令构造平行四边形

简单的定义平行四边形的方式为：



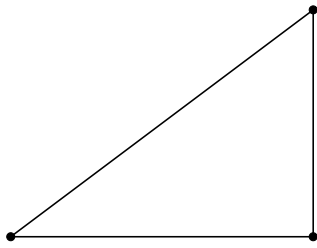
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(30:3){B}
\tkzDefShiftPointCoord[B](10:2){C}
\tkzDefShiftPointCoord[A](10:2){D}
\tkzDrawPolygon(A,...,D)
\tkzDrawPoints(A,...,D)
\end{tikzpicture}
```

4.3 \tkzDefPoints命令：定义点集

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

x_i 和 y_i 是 n_i 点的 2 维坐标

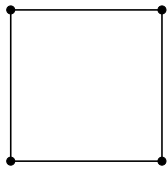
参数	默认值	样例
$x_i/y_i/n_i$	无	<code>\tkzDefPoints{0/0/0,2/2/A}</code>
选项	默认值	含义
shift	无	为所有点添加 (x,y) 或 $(\alpha:d)$ 坐标偏移

4.3.1 用`\tkzDefPoints`命令构造三角形

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

4.3.2 用`\tkzDefPoints`命令构造正方形

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 定义特殊点

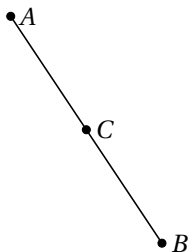
以上是`tkz-base`宏包中基本的定义一个点的命令，其中，最为重要的是`\tkzDefPoint`命令。当然，该宏包也提供了方便地直接定义一些特殊点的命令。

5.1 `\tkzDefMidPoint`命令：定义线段中点

`\tkzDefMidPoint(<pt1,pt2>)`

定义的点存储于`\tkzPointResult`命令中，也可以通过`\tkzGetPoint`命令得到该点，并为其命名。

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

5.1.1 `\tkzDefMidPoint`命令示例

```
\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

设共有 pt_1, pt_2, \dots, pt_n 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新向量:

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

则，该向量可用于定义一个点。

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

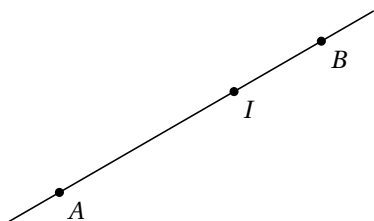
参数	默认值	含义
<code>(pt1=α_1,pt2=α_2,...)</code>	无	每个点的权重

注意：至少需要两个已知点，才能实现计算。

5.2.1 用 `\tkzDefBarycentricPoint` 命令计算 2 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：

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

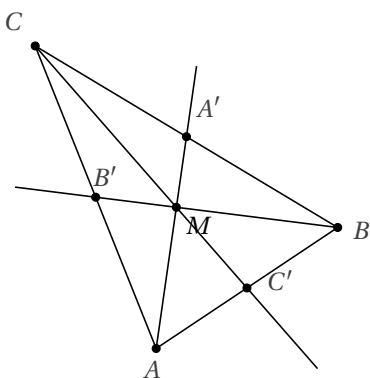


```
\begin{tikzpicture}
\tkzDefPoint(2,3){A}
\tkzDefShiftPointCoord[2,3](30:4){B}
\tkzDefBarycentricPoint(A=1,B=2)
\tkzGetPoint{I}
\tkzDrawPoints(A,B,I)
\tkzDrawLine(A,B)
\tkzLabelPoints(A,B,I)
\end{tikzpicture}
```

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。

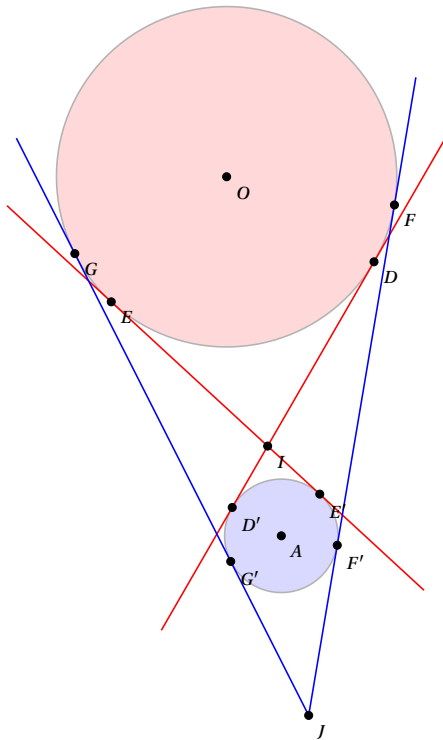
注意，为简化操作，该宏包还提供了另外一个用于直接计算三角形重心的 `\tkzCentroid` 命令。



```
\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 相似中心

两个圆对应外部和内部相似中心。




```
\begin{tikzpicture}[scale=.75,rotate=-30]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(4,-5){A}
  % 内部相似中心
  \tkzDefIntSimilitudeCenter(0,3)(A,1)
  \tkzGetPoint{I}
  % 外部相似中心
  \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 三角形特殊点

6.1 \tkzDefTriangleCenter命令：定义三角形特殊点

该命令用于定义与三角形相关的几个特殊点。

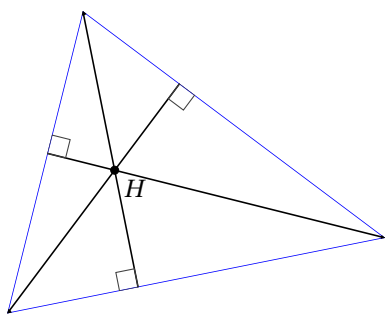
`\tkzDefTriangleCenter[(命令选项)](A,B,C)`

 注意：该命令的参数必须是一个三角形的 3 个顶点列表，结合 `\tkzGetPoint` 命令保存定义的点，并为其命名。当然，仅临时使用，则可使用 `\tkzPointResult` 命令使用该点。

参数	默认值	含义
(pt1,pt2,pt3)	无	逗号分隔的三角形 3 个顶点列表
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	陪位重心，Lemoine 点或中间点或 Grebe 点
spieker	circum	Spieker 点，中点三角形内切圆圆心
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	三个旁切圆圆心与对应边中点连线的交点
feuerbach	circum	Feuerbach 点，内切圆与九点圆的公切点

6.1.1 ortho或orthic选项：垂心

三角形三条高的交点称为三角形的垂心。



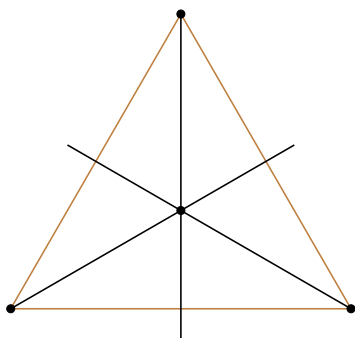
```

\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：重心

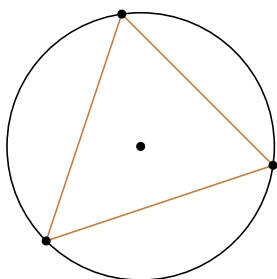
三角形三条中线的交点称为重心。



```
\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：外心

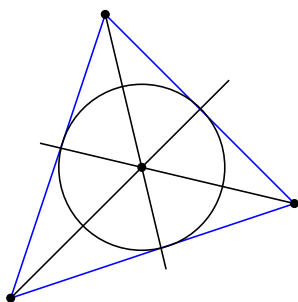
三角形外接圆圆心称为外心，也是三个边垂直平分线的交点。



```
\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：内心

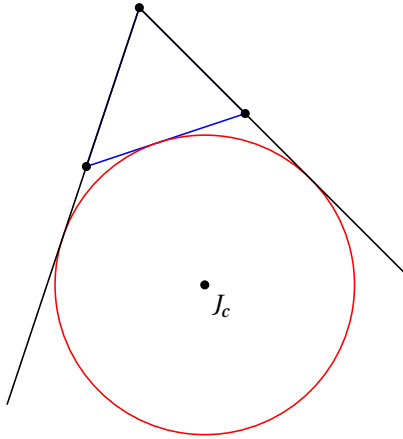
几何学中，三角形的内切圆是三角形中最大的圆，内切圆的圆心称为三角形的内心。三角形的内也是三角形三个内角角平分线的交点。(https://en.wikipedia.org/wiki/Incircle_and_excircles_of_a_triangle)



```
\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：旁心

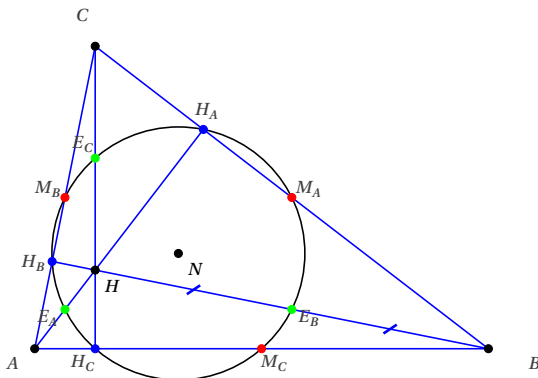
旁切圆圆心是一个顶点(例如顶点 A)的内角平分线与另外两个外角平分线的交点。该旁切圆圆心是相对于顶点 A 的一个旁心，或叫作 A 的旁心。因为三角形内角平分线与对应的外角平分线垂直，所以，内心与3个旁心构成了一个正交系统。旁切圆是位于三角形外部与某条边及另外两条边的延长线相切的圆，一个三角形有3个旁切圆。(https://en.wikipedia.org/wiki/Incircle_and_excircles_of_a_triangle)



```
\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：欧拉点

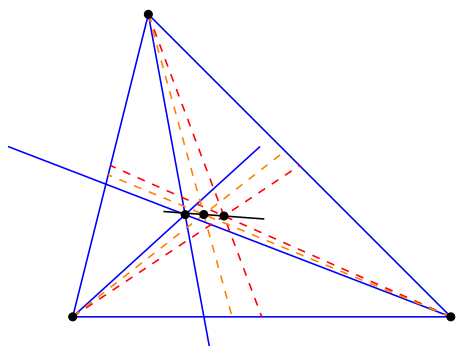
欧拉点是欧拉圆的圆心，欧拉圆又称九点圆或费尔巴哈圆的圆心。欧拉圆是通过三角形 ABC 三个顶点向对边作垂线形成的三个垂脚 H_A 、 H_B 和 H_C 的圆，欧拉在1765年证明该圆同时通过三角形 ABC 三个边的中点 M_A 、 M_B 和 M_C 。根据费尔巴哈定理，欧拉圆也通过三角形 ABC 三个顶点与重心 H 连线线段的中点 E_A 、 E_B 和 E_C 。(https://mathworld.wolfram.com/Nine-PointCircle.html)



```
\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：陪位重心

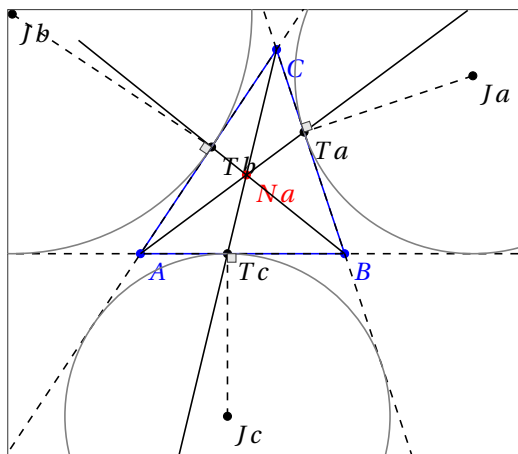
设 AN 、 BM 、 CE 是三角形 ABC 的三条中线， N' 、 M' 、 E' 分别是三条边 BC 、 CA 、 AB 上的点，若 $\widehat{BAN'} = \widehat{NAC}$ 、 $\widehat{CBM'} = \widehat{M'BA}$ 、 $\widehat{ACE} = \widehat{E'CB}$ ，则三条直线 AN' 、 BM' 、 CE' 交于一点 K ，该点称为陪位重心。



```
\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：界心（Nagel 点）

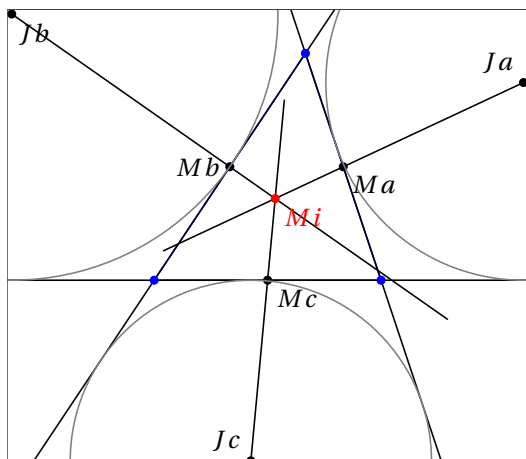
令 Ta 、 Tb 和 Tc 分别为旁切圆与三角形三条边的切点，连线 ATa 、 BTb 和 CTc ，其交点称为 Nagel 点，俗称三角形的“界心”。Weisstein, Eric W. “Nagel point”. From MathWorld—A Wolfram Web Resource.



```
\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

三个旁切圆圆心与对应边中点连线的交点。



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/A,6/0/B,4/6/C}
  \tkzDefSpcTriangle[centroid](A,B,C){Ma,Mb,Mc}
  \tkzDefSpcTriangle[ex](A,B,C){Ja,Jb,Jc}
  \tkzDefSpcTriangle[extouch](A,B,C){Ta,Tb,Tc}
  \tkzDefTriangleCenter[mittenpunkt](A,B,C)
  \tkzGetPoint{Mi}
  \tkzDrawPoints(Ma,Mb,Mc,Ja,Jb,Jc)
  \tkzClipBB
  \tkzDrawPolygon[blue](A,B,C)
  \tkzDrawLines[add=0 and 1](Ja,Ma
    Jb,Mb Jc,Mc)
  \tkzDrawLines[add=1 and 1](A,B A,C B,C)
  \tkzDrawCircles[gray](Ja,Ta Jb,Tb Jc,Tc)
  \tkzDrawPoints[blue](B,C,A)
  \tkzDrawPoints[red](Mi)
  \tkzLabelPoints[red](Mi)
  \tkzLabelPoints[left](Mb)
  \tkzLabelPoints(Ma,Mc,Jb,Jc)
  \tkzLabelPoints[above left](Ja,Jc)
  \tkzShowBB
\end{tikzpicture}
```

7 绘制点

7.1 \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

必选参数，点圆盘用填充色绘制，但颜色较浅，可以通过选项实现更多效果。由于采用 node 的方式绘制，因此，缩放操作不影响点的尺寸。

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	$6 \times \text{\pgflinewidth}$
color	black	默认颜色，可以修改

7.1.1 点绘制示例

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。

```

•
•
•
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(1,3){A}
  \tkzDefPoint(4,1){B}
  \tkzDefPoint(0,0){O}
  \tkzDrawPoint[color=red](A)
  \tkzDrawPoint[fill=blue!20,draw=blue](B)
  \tkzDrawPoint[color=green](O)
\end{tikzpicture}
```

7.2 `\tkzDrawPoints`命令：绘制多个点

可以一次绘制多个点，但该命令比绘制单点慢。另外，一次绘制多个点时，所有点使用相同选项。

`\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> , 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	$6 \times \texttt{\textbackslash pgflinewidth}$
color	black	默认为黑色, 可以被修改

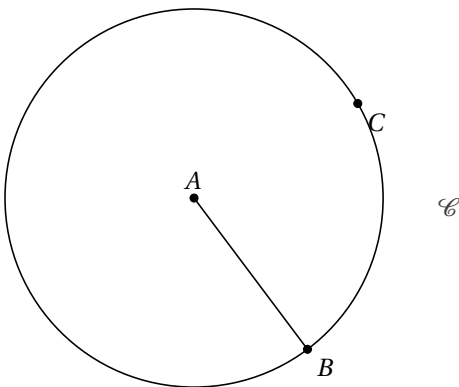
命令最后有一个“s”，如果没有“s”则会发生错误。

7.2.1 示例 1



```
\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}
```

7.2.2 示例 2



```
\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

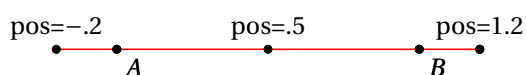
8.1 \tkzDefPointOnLine命令：定义直线上的点

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

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	距离起点 A 的比例

8.1.1 pos选项

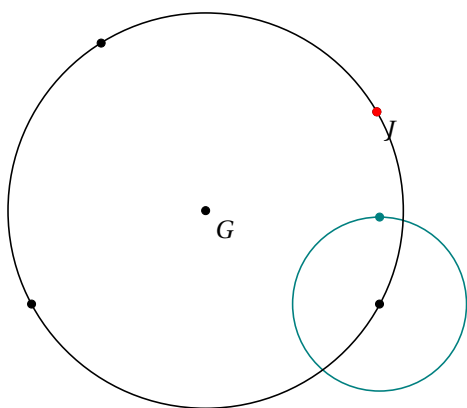


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawLine[red](A,B)
  \tkzDefPointOnLine[pos=1.2](A,B)
  \tkzGetPoint{P}
  \tkzDefPointOnLine[pos=-.2](A,B)
  \tkzGetPoint{R}
  \tkzDefPointOnLine[pos=0.5](A,B)
  \tkzGetPoint{S}
  \tkzDrawPoints(A,B,P)
  \tkzLabelPoints(A,B)
  \tkzLabelPoint[above](P){pos=$1.2$}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

8.2 \tkzDefPointOnCircle命令：定义圆上的点

`\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 利用坐标变换定义点

这些变换主要有：

- 平移;
- 缩放;
- 轴对称;
- 中心对称;
- 正交投影;
- 旋转(度或弧度);
- 相对于圆的反转.

可以使用`\tkzDefPointBy`命令实现单点变换,也可以通过`\tkzDefPointsBy`实现多点变换,变换方式用选项。默认用 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。

9.1 `\tkzDefPointBy`: 通过变换定义一个点

`\tkzDefPointBy`[< 命令选项>](`<pt>`)

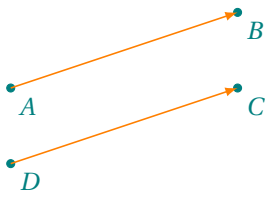
参数是一个已知点,结果存储于`\tkzPointResult`命令,可用`\tkzGetPoint{M}`命令保存该点,并为点命名。

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 0 angle 30] (E)
rotation in rad	= center #1 angle #2	[rotation in rad=center 0 angle pi/3] (E)
inversion	= center #1 through #2	[inversion =center 0 through A] (E)

该命令仅定义一个点,并不绘制该点。

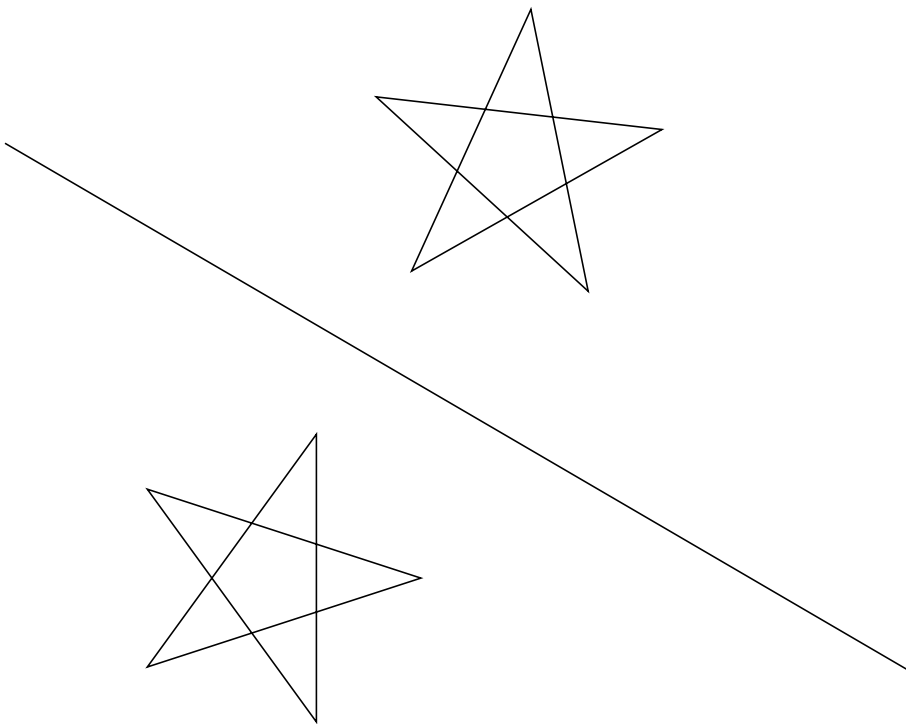
9.2 变换示例

9.2.1 平移示例



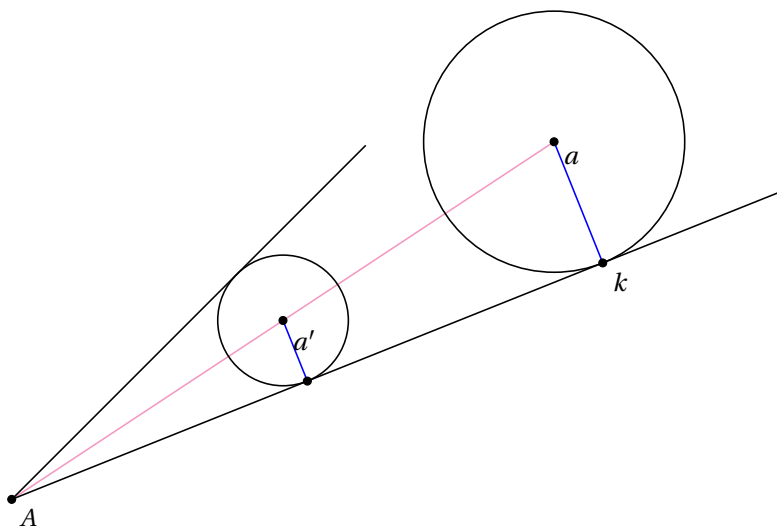
```
\begin{tikzpicture}[>=latex]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,1){B}
\tkzDefPoint(3,0){C}
\tkzDefPointBy[translation= from B to A](C)
\tkzGetPoint{D}
\tkzDrawPoints[teal](A,B,C,D)
\tkzLabelPoints[color=teal](A,B,C,D)
\tkzDrawSegments[orange,->](A,B D,C)
\end{tikzpicture}
```

9.2.2 轴对称示例



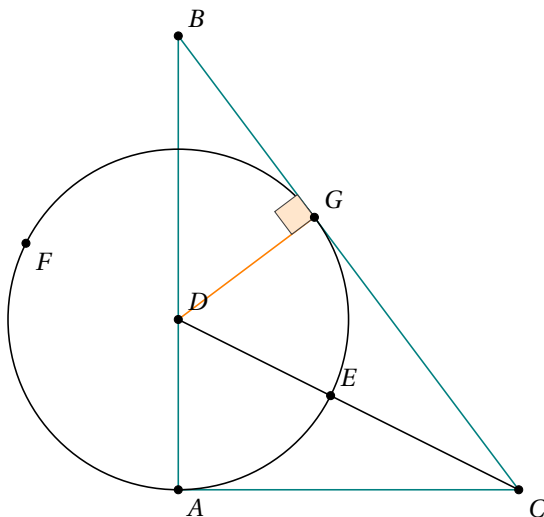
```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{1.5/-1.5/C,-4.5/2/D}
\tkzDefPoint(-4,-2){O}
\tkzDefPoint(-2,-2){A}
\foreach \i in {0,1,...,4}{%
  \pgfmathparse{0+\i * 72}
  \tkzDefPointBy[rotation=%
    center O angle \pgfmathresult](A)
  \tkzGetPoint{A\i}
  \tkzDefPointBy[reflection = over C--D](A\i)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


9.2.3 homothety和projection示例



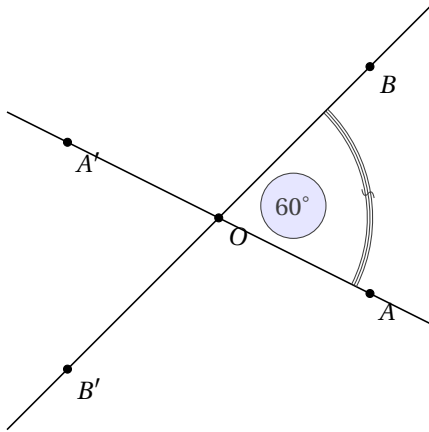
```
\begin{tikzpicture}[scale=1.2]
\tkzDefPoint(0,1){A} \tkzDefPoint(5,3){B} \tkzDefPoint(3,4){C}
\tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
\tkzDrawLine[add=0 and 0,color=magenta!50](A,a)
\tkzDefPointBy[homothety=center A ratio .5](a) \tkzGetPoint{a'}
\tkzDefPointBy[projection = onto A--B](a') \tkzGetPoint{k'}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

9.2.4 投影示例



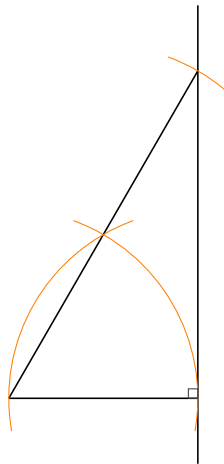
```
\begin{tikzpicture}[scale=1.5]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(0,4){B}
  \tkzDefTriangle[pythagore](B,A) \tkzGetPoint{C}
  \tkzDefLine[bisector](B,C,A) \tkzGetPoint{c}
  \tkzInterLL(C,c)(A,B) \tkzGetPoint{D}
  \tkzDefPointBy[projection=onto B--C](D)
  \tkzGetPoint{G}
  \tkzInterLC(C,D)(D,A) \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}
```

9.2.5 中心对称示例



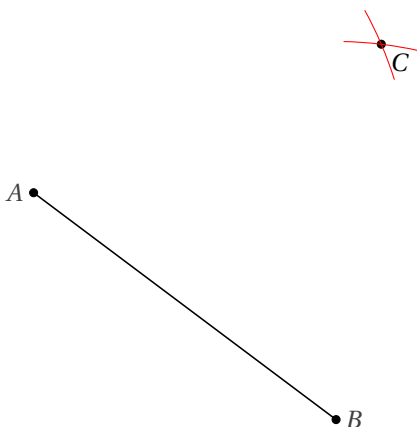
```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(2,2){B}
  \tkzDefPointsBy[symmetry=center O](B,A){}
  \tkzDrawLine(A,A')
  \tkzDrawLine(B,B')
  \tkzMarkAngle[mark=s,arc=lll,
    size=2 cm,mkcolor=red](A,O,B)
  \tkzLabelAngle[pos=1,circle,draw,
    fill=blue!10](A,O,B){$60^\circ$}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

9.2.6 旋转示例 (度)



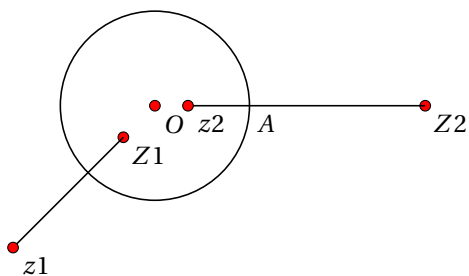
```
\begin{tikzpicture}[scale=0.5]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,0){B}
  \tkzDrawSegment(A,B)
  \tkzDefPointBy[rotation=center A angle 60](B)
  \tkzGetPoint{C}
  \tkzDefPointBy[symmetry=center C](A)
  \tkzGetPoint{D}
  \tkzDrawSegment(A,t kzPointResult)
  \tkzDrawLine(B,D)
  \tkzDrawArc[orange,delta=10](A,B)(C)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

9.2.7 旋转示例 (弧度)



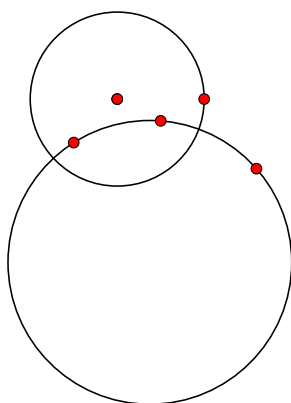
```
\begin{tikzpicture}
  \tkzDefPoint["$A$" left](1,5){A}
  \tkzDefPoint["$B$" right](5,2){B}
  \tkzDefPointBy[rotation in rad=center A angle pi/3](B)
  \tkzGetPoint{C}
  \tkzDrawSegment(A,B)
  \tkzDrawPoints(A,B,C)
  \tkzCompass[color=red](A,C)
  \tkzCompass[color=red](B,C)
  \tkzLabelPoints(C)
\end{tikzpicture}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

9.2.8 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}
```

9.2.9 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}
```

9.3 \tkzDefPointsBy命令：通过变换定义多个点

该命令是单点变换命令的变体，用于定义多点变换。必须在圆括号中，通过参数指定变换点名称，也可以在大括号中给出变换后点的名称。

```
\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 ⟩](⟨ 变换点列表 ⟩){⟨ 变换结果点名称列表 ⟩}`

参数

示例

`(⟨ 变换点列表 ⟩){⟨ 变换结果点名称列表 ⟩}` `(A,B){E,F}` E 是 A 的变换, F 是 B 的变换。

如果变换结果点名称列表为空, 变换结果点的名称是在原名称后添加“'”号。

选项

示例

`translation = from #1 to #2`

`[translation=from A to B] (E){}`

`homothety = center #1 ratio #2`

`[homothety=center A ratio .5] (E){F}`

`reflection = over #1--#2`

`[reflection=over A--B] (E){F}`

`symmetry = center #1`

`[symmetry=center A] (E){F}`

`projection = onto #1--#2`

`[projection=onto A--B] (E){F}`

`rotation = center #1 angle #2`

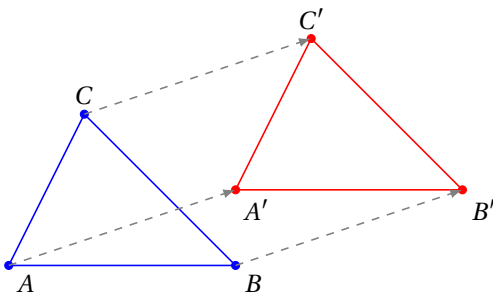
`[rotation=center angle 30] (E){F}`

`rotation in rad = center #1 angle #2`

for instance `angle pi/3`

该命令仅定义变换结果点, 并不绘制这些点。

9.3.1 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：定义向量点

可通过多种方案定义满足特定向量条件的点，此时，需要用两个点作为参数，也就是一个向量。不同的选项用于设置通过共线向量或正交向量的方式定义新点，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。如果该点仅做临时使用，则不需要立即命名，使用\tkzPointResult命令即可。也可使用\tkzGetPoint命令保存该点，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，则定义的点的距离为 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。

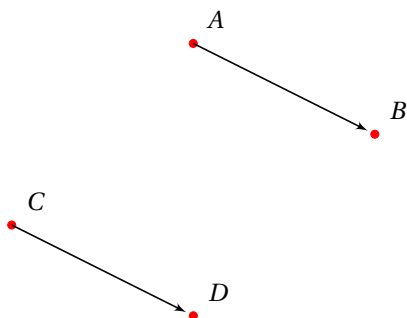
参数	含义	说明
(pt1,pt2)	点对	结果是保存于\tkzPointResult命令

假定由\tkzGetPoint{C}得到该点。

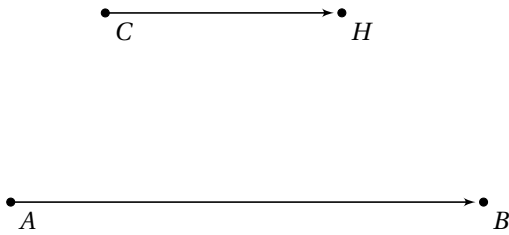
选项	样例	说明
orthogonal	[orthogonal] (A,B)	$AC = AB$ 和 $\overrightarrow{AC} \perp \overrightarrow{AB}$
orthogonal normed	[orthogonal normed] (A,B)	$AC = 1$ 和 $\overrightarrow{AC} \perp \overrightarrow{AB}$
linear	[linear] (A,B)	$\overrightarrow{AC} = K \times \overrightarrow{AB}$
linear normed	[linear normed] (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选项示例

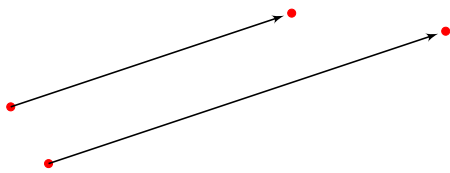
$$(\overrightarrow{AB} = \overrightarrow{CD})$$



```
\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 选项示例

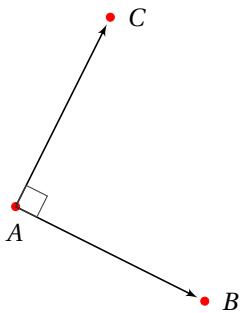
```
\begin{tikzpicture}[vect/.style={->, shorten >=3pt,
    >=latex'}, scale=1.25]
\tkzDefPoint(0,0){A} \tkzDefPoint(5,0){B}
\tkzDefPoint(1,2){C}
\tkzDefPointWith[colinear=at C](A,B)
\tkzGetPoint{G}
\tkzDefPointWith[colinear=at C,K=0.5](A,B)
\tkzGetPoint{H}
\tkzLabelPoints(A,B,C,G,H)
\tkzDrawPoints(A,B,C,G,H)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

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

```
\begin{tikzpicture}[vect/.style={->, shorten >=3pt,
    >=latex'}, scale=1.75]
\tkzDefPoint(1,1){A} \tkzDefPoint(4,2){B}
\tkzDefPoint(2,2){CU}
\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选项示例

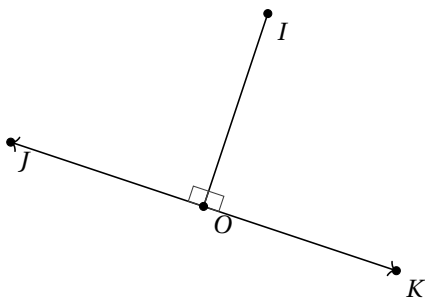
因 $K = 1$, 所以 $AB = AC$ 。



```
\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}
```

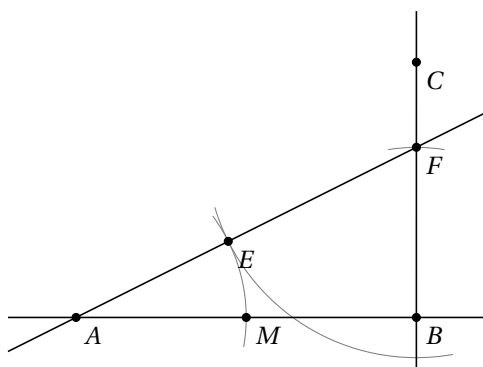
10.1.5 orthogonal带 $K = -1$ 选项示例

因 $|K| = 1$, 所以 $OI = OJ = OK$ 。



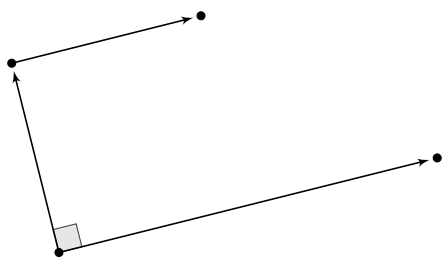
```
\begin{tikzpicture}[scale=0.85]
\tkzDefPoint(1,2){O} \tkzDefPoint(2,5){I}
\tkzDefPointWith[orthogonal](O,I)
\tkzGetPoint{J}
\tkzDefPointWith[orthogonal,K=-1](O,I)
\tkzGetPoint{K}
\tkzDrawSegment(O,I)
\tkzDrawSegments[->](O,J O,K)
\tkzMarkRightAngles(I,O,J I,O,K)
\tkzDrawPoints(O,I,J,K)
\tkzLabelPoints(O,I,J,K)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

10.1.6 orthogonal选项综合示例



```
\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}
```

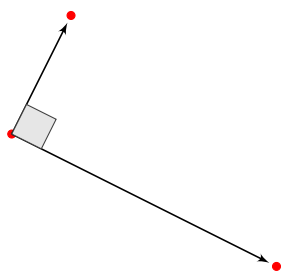
10.1.7 colinear和orthogonal选项示例



```
\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 $K=1$ 选项示例

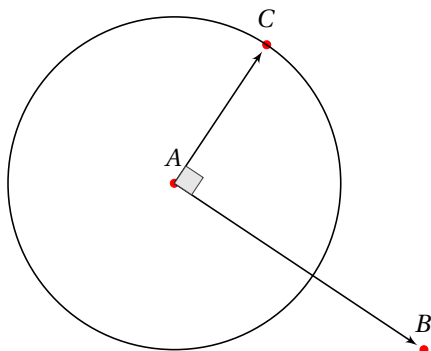
$AC = 1$.



```
\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 $K=2$ 选项示例

因 $K=2$ ，所以 $AC=2$ 。

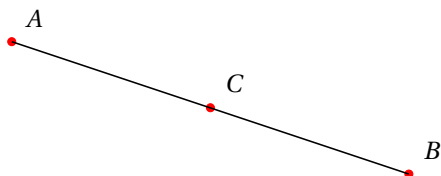


```
\begin{tikzpicture}[vect/.style={->,shorten >=3pt,
=>=latex'}, scale=1.10]
\tkzDefPoint(2,3){A}
\tkzDefPoint(5,1){B}
\tkzDefPointWith[orthogonal normed,K=2](A,B)
\tkzGetPoint{C}
\tkzDrawPoints[color=red](A,B,C)
\tkzDrawCircle[R](A,2cm)
\tkzDrawSegments[vect](A,B A,C)
\tkzMarkRightAngle[fill=gray!20](B,A,C)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

10.1.10 linear选项示例

在此，取 $K=0.5$ 。

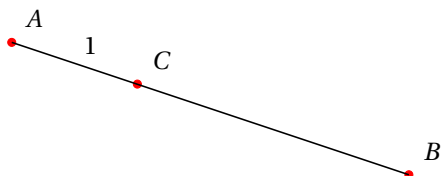
这相当于给一个向量乘了一个实数，本例中是 $[AB]$ 的中点。



```
\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选项示例

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：获取向量坐标分量

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

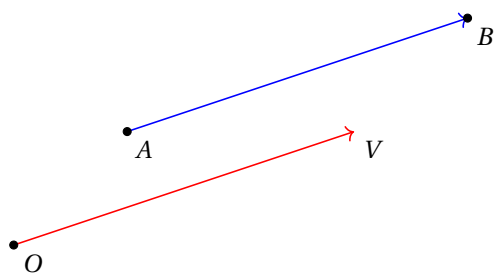
获得一个向量的坐标分量。

参数

样例

说明

(point){name of macro} \tkzGetVectxy(A,B){V} V_x, V_y 向量 \overrightarrow{AB} 的坐标分量

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 \tkzDefRandPointOn命令：定义随机点

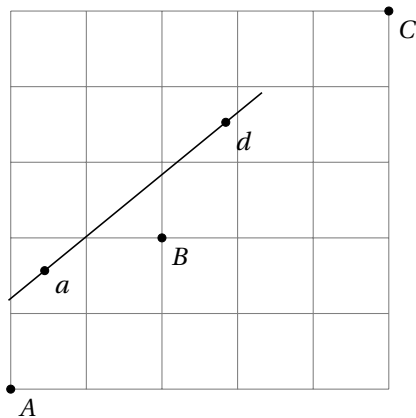
该命令取代了旧的\tkzGetRandPointOn命令，用于定义随机点。

\tkzDefRandPointOn[< 命令选项>]

可以用\tkzGetPoint保存并命名定义的随机点，如仅为临时使用，则可使用\tkzPointResult命令。

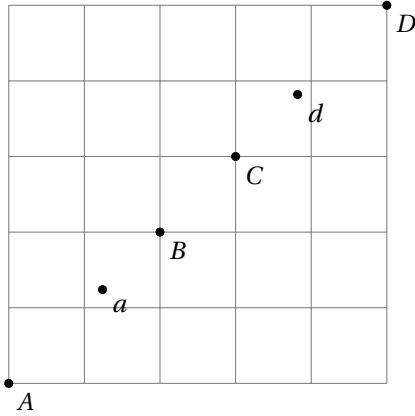
选项	默认值	含义
rectangle=pt1 and pt2		[rectangle=A and B]
segment= pt1--pt2		[segment=A--B]
line=pt1--pt2		[line=A--B]
circle =center pt1 radius dim		[circle = center A radius 2 cm]
circle through=center pt1 through pt2		[circle through= center A through B]
disk through=center pt1 through pt2		[disk through=center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

11.2 矩形内的随机点



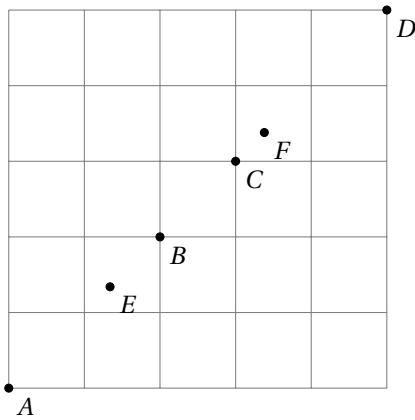
```
\begin{tikzpicture}
\tkzInit[xmax=5,ymax=5]
\tkzGrid
\tkzDefPoints{0/0/A,2/2/B,5/5/C}
\tkzDefRandPointOn[rectangle = A and B]
\tkzGetPoint{a}
\tkzDefRandPointOn[rectangle = B and C]
\tkzGetPoint{d}
\tkzDrawLine(a,d)
\tkzDrawPoints(A,B,C,a,d)
\tkzLabelPoints(A,B,C,a,d)
\end{tikzpicture}
```

11.3 线段上的随机点



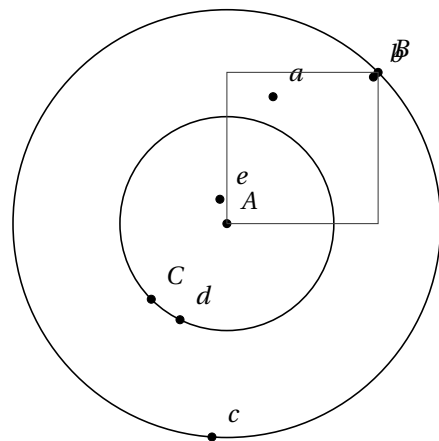
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzGrid
  \tkzDefPoints{0/0/A,2/2/B,3/3/C,5/5/D}
  \tkzDefRandPointOn[segment = A--B]\tkzGetPoint{a}
  \tkzDefRandPointOn[segment = C--D]\tkzGetPoint{d}
  \tkzDrawPoints(A,B,C,D,a,d)
  \tkzLabelPoints(A,B,C,D,a,d)
\end{tikzpicture}
```

11.4 直线上的随机点



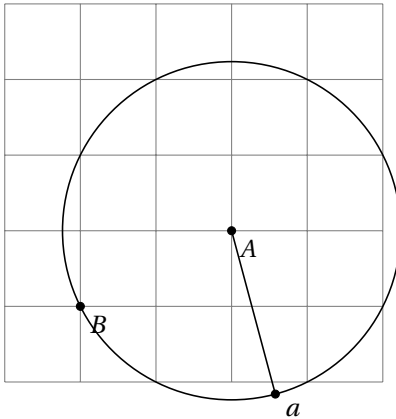
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzGrid
  \tkzDefPoints{0/0/A,2/2/B,3/3/C,5/5/D}
  \tkzDefRandPointOn[line = A--B]\tkzGetPoint{E}
  \tkzDefRandPointOn[line = C--D]\tkzGetPoint{F}
  \tkzDrawPoints(A,...,F)
  \tkzLabelPoints(A,...,F)
\end{tikzpicture}
```

11.4.1 随机点综合示例



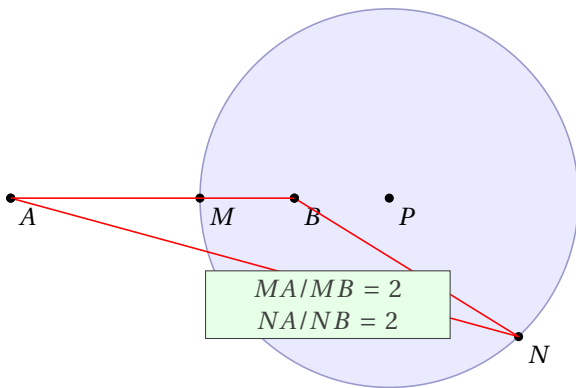
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,2/2/B,-1/-1/C}
  \tkzDefCircle[through=](A,C)
  \tkzGetLength{rAC}
  \tkzDrawCircle(A,C)
  \tkzDrawCircle(A,B)
  \tkzDefRandPointOn[rectangle=A and B]
  \tkzGetPoint{a}
  \tkzDefRandPointOn[segment=A--B]
  \tkzGetPoint{b}
  \tkzDefRandPointOn[circle=center 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}
```

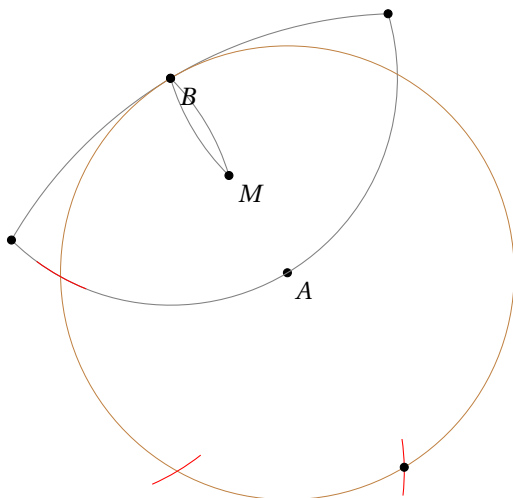
11.5.1 Apollonius 圆的随机示例



```
\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}
```

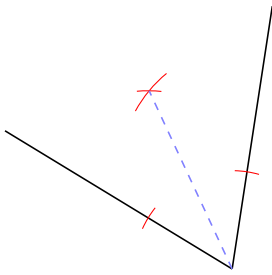
11.5.2 线段中点

可以用尺规作图的方式求得线段的中点。



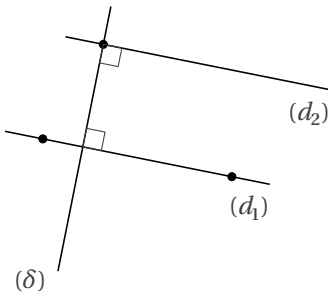
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A}
  \tkzDefRandPointOn[circle= center A radius 4cm]
  \tkzGetPoint{B}
  \tkzDrawPoints(A,B)
  \tkzDefPointBy[rotation= center A angle 180](B)
  \tkzGetPoint{C}
  \tkzInterCC[R](A,4 cm)(B,4 cm)
  \tkzGetPoints{I}{I'}
  \tkzInterCC[R](A,4 cm)(I,4 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}
```


12.1.2 bisector和normed选项示例



```
\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}
```

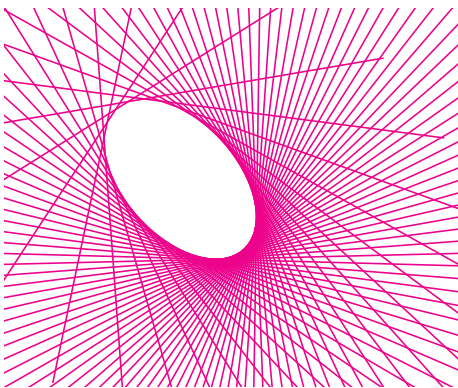
12.1.3 orthogonal和parallel选项示例



```
\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 设计的图形绘制。

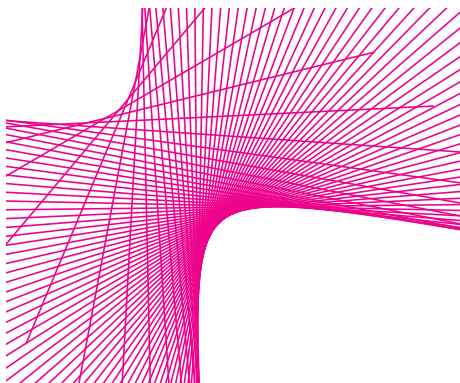


```
\begin{tikzpicture}[scale=.5]
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6] % necessary
\tkzClip
\tkzDefPoint(0,0){O}
\tkzDefPoint(132:4){A}
\tkzDefPoint(5,0){B}
\foreach \ang in {5,10,...,360}{%
\tkzDefPoint(\ang:5){M}
\tkzDefLine[mediator](A,M)
\tkzDrawLine[color=magenta,add= 3 and
3](tkzFirstPointResult,tkzSecondPointResult)}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

12.1.5 抛物线

基于 D Rodriguez 用 pst-eucl 宏包绘制的 O. Reboux 设计的图形绘制。

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 \tkzDefTangent 命令：定义圆的切线

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。

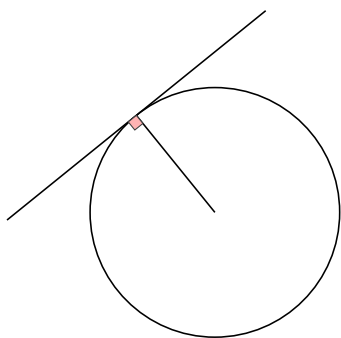
`\tkzDefTangent[< 命令选项>](<pt1,pt2>) or (<pt1,dim>)`

圆括号中的参数是圆心和圆上的一个点或是圆心和半径，该命令取代了旧的 `\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>	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参数指定

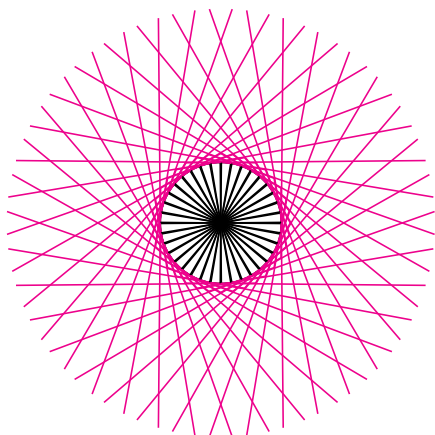
该命令仅定义切线，并不绘制切线。切线的第 2 个点可通过 `\tkzPointResult` 命令得到。

12.2.1 圆上指定点的切线示例



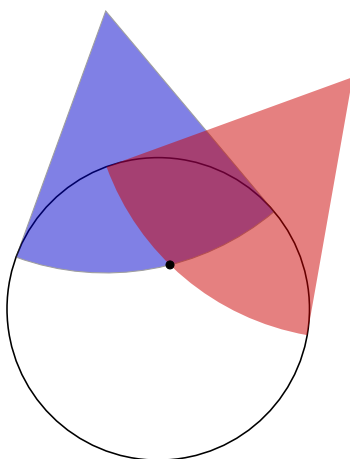
```
\begin{tikzpicture}[scale=.55]
\tkzDefPoint(0,0){O} \tkzDefPoint(6,6){E}
\tkzDefRandPointOn[center O radius 3cm]
\tkzGetPoint{A} \tkzDrawSegment(O,A)
\tkzDrawCircle(O,A)
% 重置包围盒 (删除圆外的空白)
\pgfresetboundingbox
\tkzDefTangent[at=A](O)
\tkzGetPoint{h}
\tkzDrawLine[add = 4 and 3](A,h)
\tkzMarkRightAngle[fill=red!30](O,A,h)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


12.2.2 过圆外指定点的切线示例



```
\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}
```

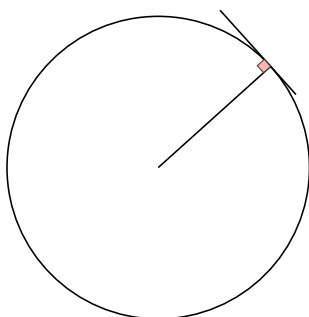
12.2.3 Andrew Mertz 示例



```
\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}
```

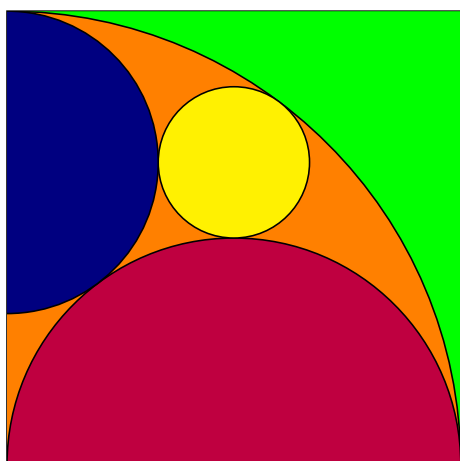
<http://www.texample.net/tikz/examples/>

12.2.4 from with R 和 at 选项示例



```
\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 选项示例



```
\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 直线绘制和命名

下列命令用于绘制和命名直线

13.1 \tkzDrawLine命令：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={%
      ($(\tikztostart)!-#1!(\tikztotarget)$)--($(\tikztotarget)!-#2!(\tikztostart)$)%
    }
  }
}
```

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，旧命令`\tkzDrawMedian`中的(A,B)(C)应该改为(B,C,A)。

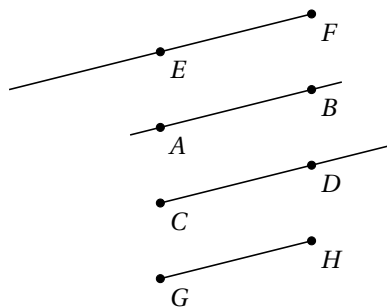
`\tkzDrawLine[< 命令选项>](pt1,pt2) or (pt1,pt2,pt3)`

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	延伸线段

`add`定义了通过点 pt1 和 pt2 的延伸长度，两个数字均为百分比，也可在绘图中使用所有有效 TikZ 样式。

13.1.1 add选项示例



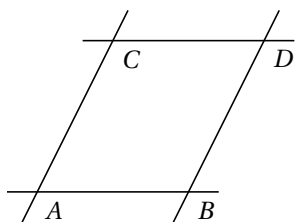
```
\begin{tikzpicture}
  \tkzInit[xmin=-2,xmax=3,ymin=-2.25,ymax=2.25]
  \tkzClip[space=.25]
  \tkzDefPoint(0,0){A} \tkzDefPoint(2,0.5){B}
  \tkzDefPoint(0,-1){C} \tkzDefPoint(2,-0.5){D}
  \tkzDefPoint(0,1){E} \tkzDefPoint(2,1.5){F}
  \tkzDefPoint(0,-2){G} \tkzDefPoint(2,-1.5){H}
  \tkzDrawLine(A,B)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

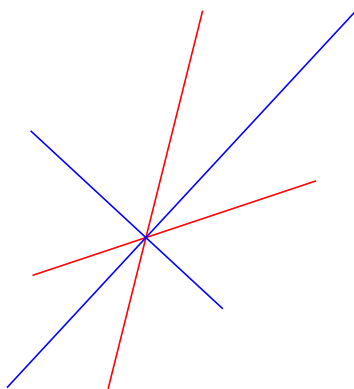
参数是用空格分隔的直线点对，点对之间用逗号分隔。也可在绘制中使用所有有效 TikZ 样式。

13.1.2 \tkzDrawLines命令示例



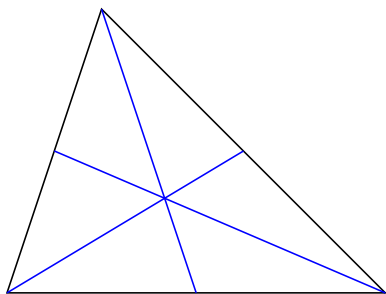
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(2,0){B}
  \tkzDefPoint(1,2){C}
  \tkzDefPoint(3,2){D}
  \tkzDrawLines(A,B C,D A,C B,D)
  \tkzLabelPoints(A,B,C,D)
\end{tikzpicture}
```

13.1.3 add选项示例



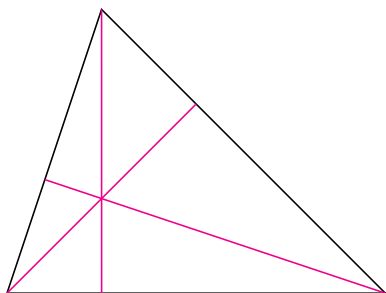
```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(3,1){I}
  \tkzDefPoint(1,4){J}
  \tkzDefLine[bisector](I,O,J)
  \tkzGetPoint{i}
  \tkzDefLine[bisector out](I,O,J)
  \tkzGetPoint{j}
  \tkzDrawLines[add = 1 and .5,color=red](O,I O,J)
  \tkzDrawLines[add = 1 and .5,color=blue](O,i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

13.1.4 三角形的中线示例



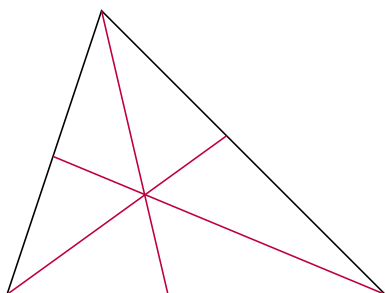
```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(1,3){C}
\tkzDrawPolygon(A,B,C)
\tkzSetUpLine[color=blue]
\tkzDrawLine[median](B,C,A)
\tkzDrawLine[median](C,A,B)
\tkzDrawLine[median](A,B,C)
\end{tikzpicture}
```

13.1.5 三角形的高示例



```
\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 三角形的角平分线示例



```
\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 \tkzLabelLine命令：添加直线标注

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

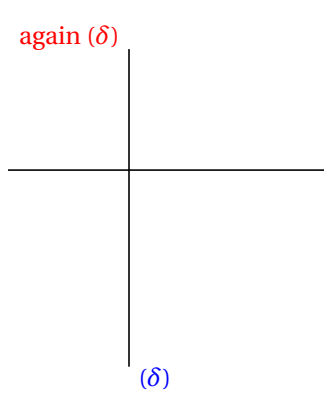
参数	默认值	含义
label		Δ

选项	默认值	含义
pos	.5	pos是 TikZ 的一个选项

可以使用`pos`外，所有有效 TikZ 样式，特别是用于设置标注位置的`above`、`right`、...等样式选项。

13.2.1 \tkzLabelLine命令示例

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。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,3/0/B,1/1/C}
\tkzDefLine[perpendicular=through C,K=-1](A,B)
\tkzGetPoint{c}
\tkzDrawLines(A,B C,c)
\tkzLabelLine[pos=1.25,blue,right](C,c){$(\delta)$}
\tkzLabelLine[pos=-0.25,red,left](C,c){again $(\delta)$}
\end{tikzpicture}
```

14 绘制和标记线段

14.1 \tkzDrawSegment命令：绘制线段

```
\tkzDrawSegment[< 命令选项>](<pt1,pt2>)
```

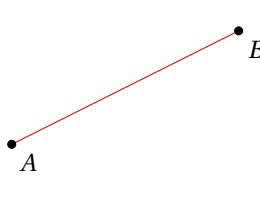
参数是用逗号分隔的线段端点列表，可以使用所有有效 TikZ 样式。

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		所有有效 TikZ 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

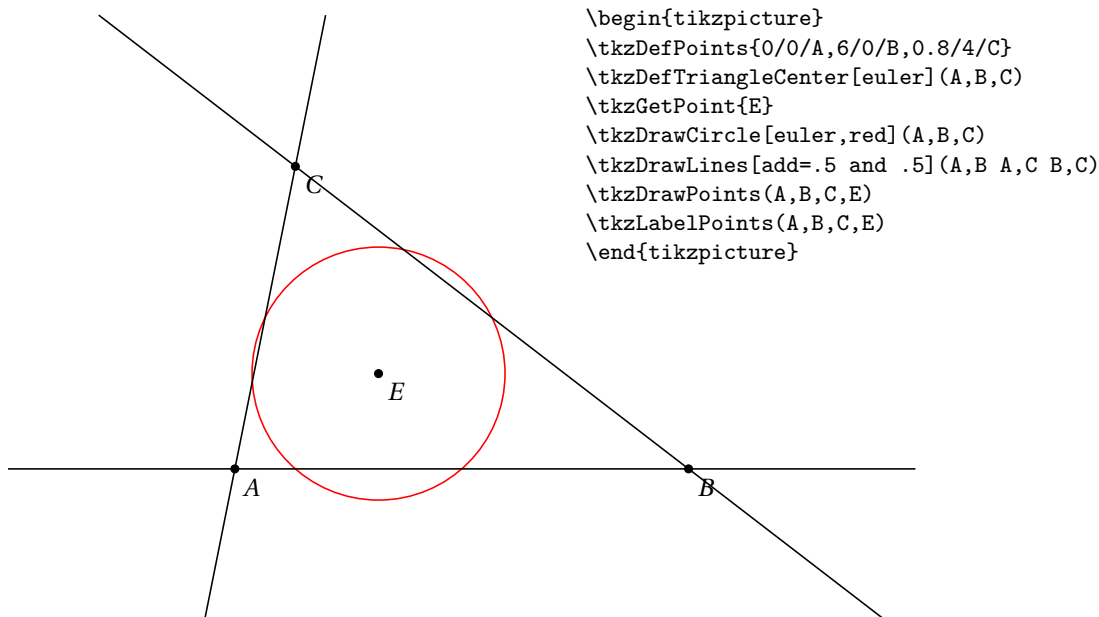
该命令等效于\draw (A)--(B); 命令

14.1.1 简单示例

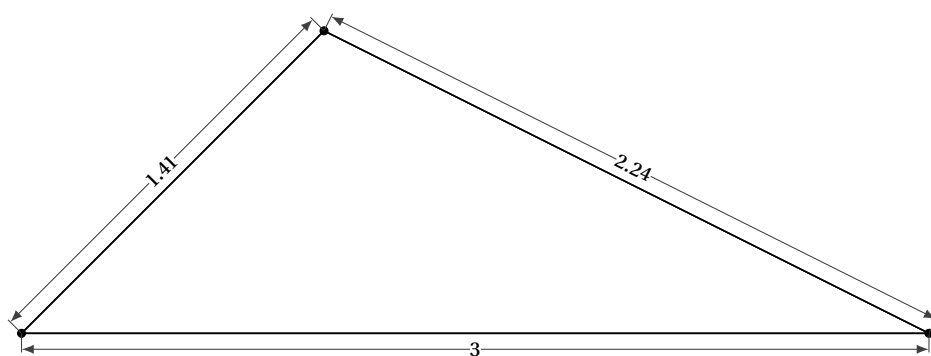


```
\begin{tikzpicture}[scale=1.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,1){B}
\tkzDrawSegment[color=red,thin](A,B)
\tkzDrawPoints(A,B)
\tkzLabelPoints(A,B)
\end{tikzpicture}
```

14.1.2 使用add选项延伸线段



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}

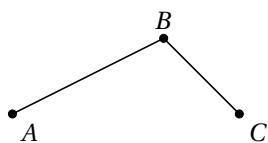
```

14.2 \tkzDrawSegments命令：绘制多条线段

该命令用于绘制选项相同的多条线段

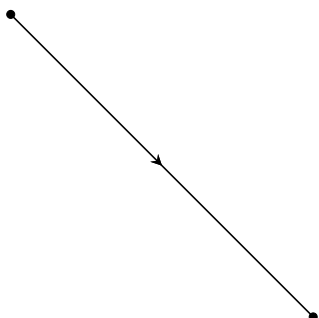
\tkzDrawSegments[< 命令选项>](<pt1,pt2 pt3,pt4,...>)

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。在绘制中可以使用所有有效 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 \tkzMarkSegment命令：标记线段

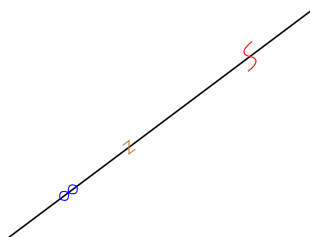
\tkzMarkSegment[< 命令选项>](<pt1,pt2>)

该命令用于为线段添加标记。

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

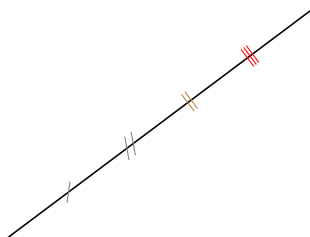
标记样式由 TikZ 提供，但也可使用基于 Yves Combe 方法的自定义标记样式。

14.3.1 几种标记示例



```
\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选项示例



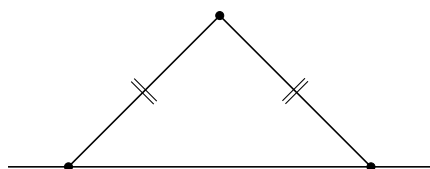
```
\begin{tikzpicture}
\tkzDefPoint(2,1){A}
\tkzDefPoint(6,4){B}
\tkzDrawSegment(A,B)
\tkzMarkSegment[color=gray,pos=0.2,mark=s|](A,B)
\tkzMarkSegment[color=gray,pos=0.4,mark=s||](A,B)
\tkzMarkSegment[color=brown,pos=0.6,mark=s||](A,B)
\tkzMarkSegment[color=red,pos=0.8,mark=s|||](A,B)
\end{tikzpicture}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

14.4 \tkzMarkSegments命令：标记多条线段

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

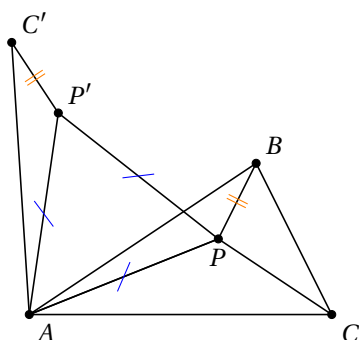
参数是用空格分隔的线段端点列表，每对端点用逗号分隔。可以使用所有有效 TikZ 样式。

14.4.1 标记等腰三角形示例



```
\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 其它标记示例



```
\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'}
  \tkzDefPointsBy[rotation=center A angle 60](P,B){P',C'}
  \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}
```

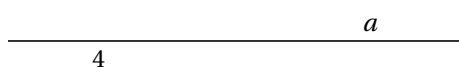
14.5 \tkzLabelSegment 命令：标注线段

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

用于标注一个线段或直线，可使用类似pos的所有有效 TikZ 样式。

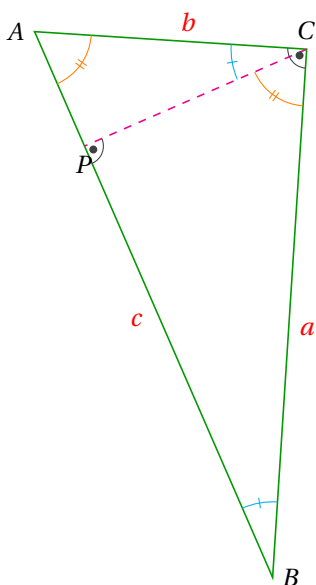
参数	样例	含义
label (pt1,pt2)	<code>\tkzLabelSegment(A,B){5}</code> (A,B)	标注文本 被标注线段 [AB]
选项	默认值	含义
pos	.5	标注的位置

14.5.1 多个标注示例



```
\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示例



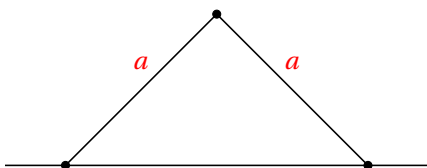
```
\begin{tikzpicture}[rotate=-60]
  \tikzset{label seg style/.append style = {%
    color      = red,
  }}
  \tkzDefPoint(0,1){A}
  \tkzDefPoint(2,4){C}
  \tkzDefPointWith[orthogonal normed,K=7](C,A)
  \tkzGetPoint{B}
  \tkzDrawPolygon[green!60!black](A,B,C)
  \tkzDrawLine[altitude,dashed,color=magenta](B,C,A)
  \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 \tkzLabelSegments命令：标注多条线段

`\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 等腰三角形的标注示例



```
\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 定义与绘制三角形

15.1 `\tkzDefTriangle`命令：定义三角形

三角形定义命令允许使用至少2个点构造一个三角形。可以按如下方式定义三角形：

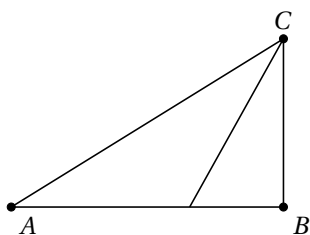
- `two angles`已知2个角的三角形;
- `equilateral`等边三角形;
- `half`直角边之和与斜边之比等于2的直角三角形;
- `pythagore`勾股直角三角形;
- `school`三个角分别是30、60和90的直角三角形;
- `golden`直角边之和与斜边比等于黄金分割比 $\Phi = 1.618034$ 的直角三角形, ;
- `euclide or gold`底角为72度的等腰三角形称为“golden triangle”或“Euclid's triangle”;
- `cheops`三边为2、 Φ 和 Φ 的等腰三角形。

`\tkzDefTriangle`[< 命令选项>](A,B)

参数中的点的顺序决定的另外一个点的位置。可使用`\tkzGetPoint`保存并命名定义的点,也可使用`tkzPointResult`命令使用计算结果,但不命名。

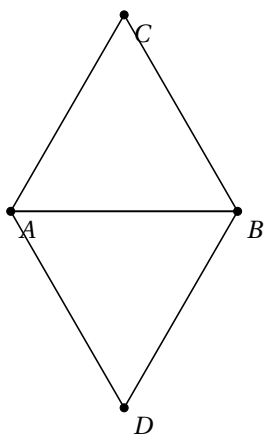
选项	默认值	含义
<code>two angles= #1 and #2</code>	无	三角形两个已知角
<code>equilateral</code>	无	等边三角形
<code>pythagore</code>	无	勾股三角形
<code>school</code>	无	三个角分别是 30、60 和 90 度
<code>gold</code>	无	三个角分别是 72、72 和 36 度, A 是顶点
<code>euclide</code>	无	同上, 但 [AB] 是底边
<code>golden</code>	无	AB 构成矩形, 并且 $AB/BC = \Phi$
<code>cheops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

使用`\tkzGetPoint`保存并命名得到的点, 使用`tkzPointResult`命令使用得到的点, 但不命名。

15.1.1 `golden`选项示例

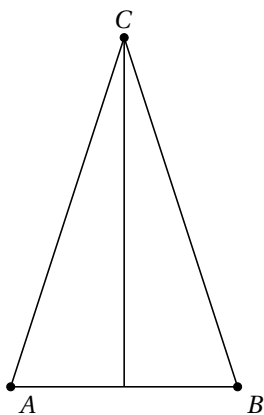
```
\begin{tikzpicture}[scale=0.9]
  \tkzInit[xmax=5,ymax=3]
  \tkzClip[space=.5]
  \tkzDefPoint(0,0){A} \tkzDefPoint(4,0){B}
  \tkzDefTriangle[golden](A,B)\tkzGetPoint{C}
  \tkzDrawPolygon(A,B,C) \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B) \tkzDrawBisector(A,C,B)
  \tkzLabelPoints[above](C)
\end{tikzpicture}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

15.1.2 equilateral选项示例



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,0){B}
\tkzDefTriangle[equilateral](A,B)
\tkzGetPoint{C}
\tkzDrawPolygon(A,B,C)
\tkzDefTriangle[equilateral](B,A)
\tkzGetPoint{D}
\tkzDrawPolygon(B,A,D)
\tkzDrawPoints(A,B,C,D)
\tkzLabelPoints(A,B,C,D)
\end{tikzpicture}
```

15.1.3 gold或euclide选项示例



```
\begin{tikzpicture}[scale=0.75]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefTriangle[euclide](A,B)
\tkzGetPoint{C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(A,B)
\tkzLabelPoints[above](C)
\tkzDrawBisector(A,C,B)
\end{tikzpicture}
```

15.2 \tkzDrawTriangle命令：绘制三角形

`\tkzDrawTriangle[< 命令选项>](A,B)`

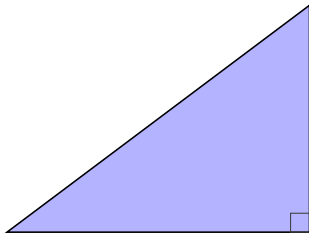
与三角形定义命令类似，但可绘制三角形。

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

定义中，三角形尺寸取决于起始的两个点。

15.2.1 pythagore选项示例

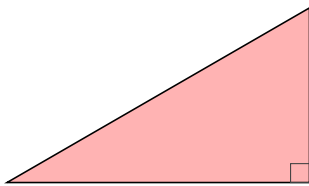
三角形的三个边满足类似3、4和5的勾股关系。



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

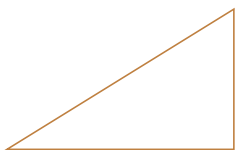
15.2.2 school选项示例

三角形的三个内角分别是30、60和90度。



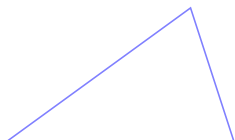
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDrawTriangle[school,fill=red!30](A,B)
\tkzMarkRightAngles(t kzPointResult,B,A)
\end{tikzpicture}
```

15.2.3 golden选项示例



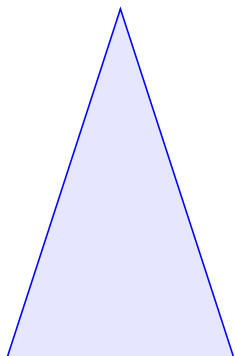
```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,-10){M}
\tkzDefPoint(3,-10){N}
\tkzDrawTriangle[golden,color=brown](M,N)
\end{tikzpicture}
```

15.2.4 gold选项示例



```
\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选项示例



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

15.3 \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。

\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	内心三角形
ex or excentral	centroid	旁心三角形
extouch	centroid	外切三角形
intouch or contact	centroid	内切三角形
centroid or medial	centroid	三边中点构成的三角形
orthic	centroid	正交三角形
feuerbach	centroid	九点圆与旁切圆的切点构成的三角形
euler	centroid	欧拉三角形
tangential	centroid	切身三角形
name	无	顶点命名前缀

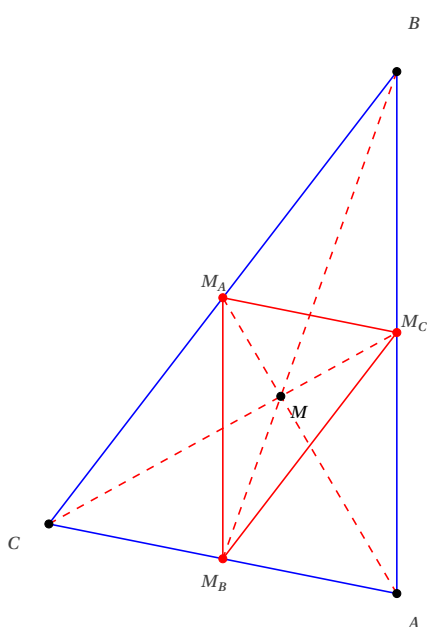
使用\tkzGetPoint保存并命名点，使用\tkzPointResult命令使用得到的点，但不命名。

15.3.1 medial或centroid选项示例

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。

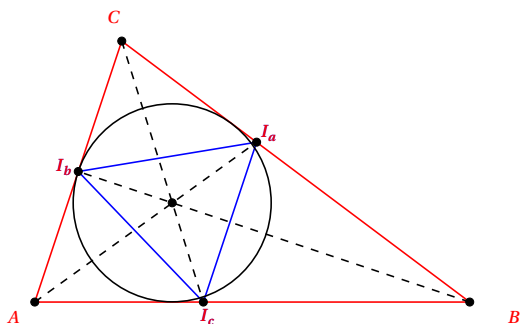


```
\begin{tikzpicture}[rotate=90,scale=1.15]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefTriangleCenter[centroid](A,B,C)
\tkzGetPoint{M}
\tkzDefSpcTriangle[medial,name=M](A,B,C){_A,_B,_C}
\tkzDrawPolygon[color=blue](A,B,C)
\tkzDrawSegments[dashed,red](A,M_A B,M_B C,M_C)
\tkzDrawPolygon[color=red](M_A,M_B,M_C)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

15.3.2 in或incentral选项示例

内心三角形是由一个三角形的三个内角平分线与对边交点确定的三角形。

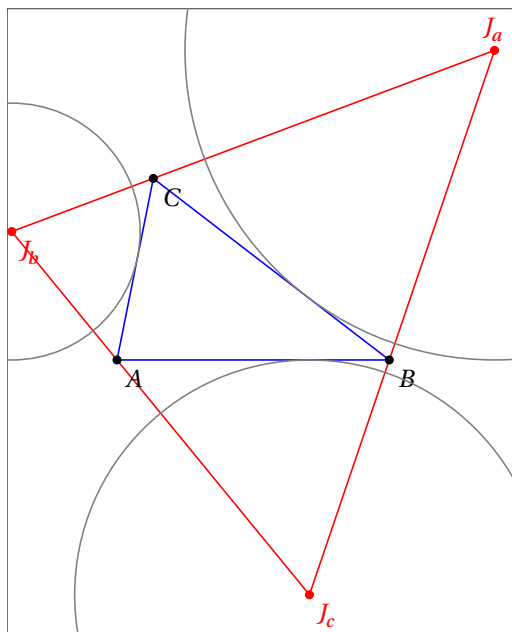
Weissstein, Eric W. “Incentral triangle” From MathWorld—A Wolfram Web Resource.



```
\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}
```

15.3.3 ex或excentral选项示例

旁心三角形是由一个三角形的三个旁心构成的三角形。

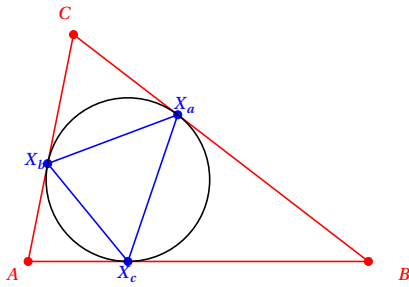


```
\begin{tikzpicture}[scale=0.60]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[excentral,name=J](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[extouch,name=T](A,B,C){_a,_b,_c}
\tkzDrawPolygon[blue](A,B,C)
\tkzDrawPolygon[red](J_a,J_b,J_c)
\tkzDrawPoints(A,B,C)
\tkzDrawPoints[red](J_a,J_b,J_c)
\tkzLabelPoints(A,B,C)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

15.3.4 intouch选项示例

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。

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



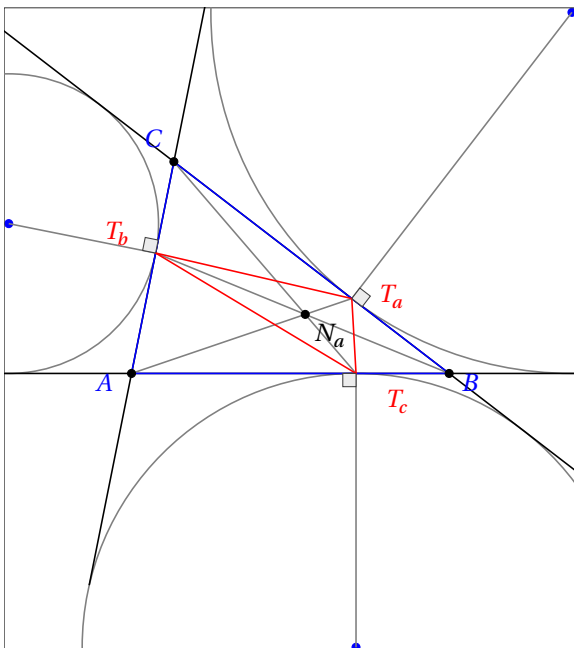
```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[intouch,name=X](A,B,C){_a,_b,_c}
\tkzInCenter(A,B,C)\tkzGetPoint{I}
\tkzDrawPolygon[red](A,B,C)
\tkzDrawPolygon[blue](X_a,X_b,X_c)
\tkzDrawPoints[red](A,B,C)
\tkzDrawPoints[blue](X_a,X_b,X_c)
\tkzDrawCircle[in](A,B,C)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

15.3.5 extouch选项示例

外切三角形 $T_a T_b T_c$ 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。

Weissstein, 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。

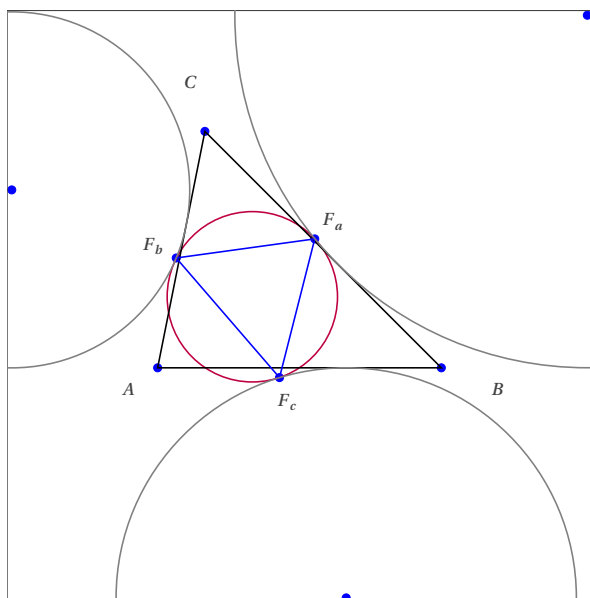


```
\begin{tikzpicture}[scale=.7]
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefSpcTriangle[excentral,
name=J](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[extouch,
name=T](A,B,C){_a,_b,_c}
\tkzDefTriangleCenter[nagel](A,B,C)
\tkzGetPoint{N_a}
\tkzDefTriangleCenter[centroid](A,B,C)
\tkzGetPoint{G}
\tkzDrawPoints[blue](J_a,J_b,J_c)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


15.3.6 feuerbach选项示例

Feuerbach 三角形是由九点圆与三个旁切圆的 3 个切点构成的三角形。

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

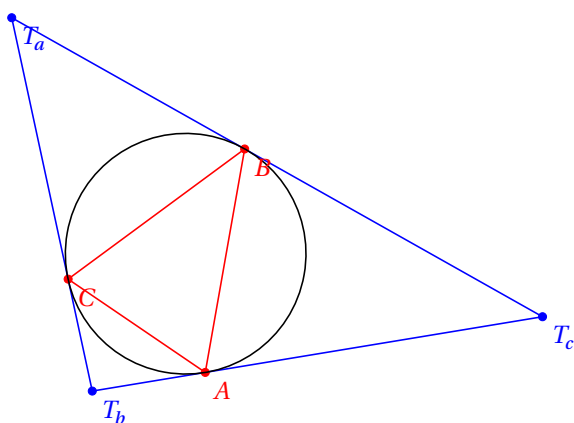


```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){A}
\tkzDefPoint(3,0){B}
\tkzDefPoint(0.5,2.5){C}
\tkzDefCircle[euler](A,B,C) \tkzGetPoint{N}
\tkzDefSpcTriangle[feuerbach,
name=F](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[excentral,
name=J](A,B,C){_a,_b,_c}
\tkzDefSpcTriangle[extouch,
name=T](A,B,C){_a,_b,_c}
\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}
```

15.3.7 tangential选项示例

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 外心的反角三角形。

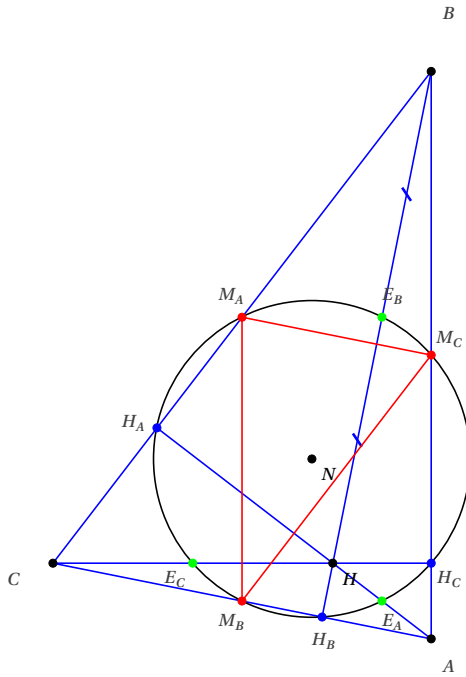
Weissstein, Eric W. “Tangential Triangle” From MathWorld—A Wolfram Web Resource.



```
\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}
```

15.3.8 euler选项示例

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}
```

16 多边形定义

16.1 `\tkzDefSquare`命令：定义正方形

`\tkzDefSquare(<pt1,pt2>)`

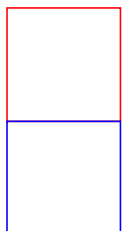
通过两个点按逆时针方向推算另外两个点后,得到正方形。结果保存在`\tkzFirstPointResult`和`\tkzSecondPointResult`命令中。

当然,可以使用`\tkzGetPoints`保存并为这两个点重命名。

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

16.1.1 通过两个点定义正方形

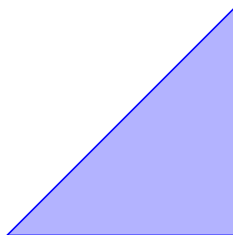
需要注意点的方向问题。



```
\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}
```

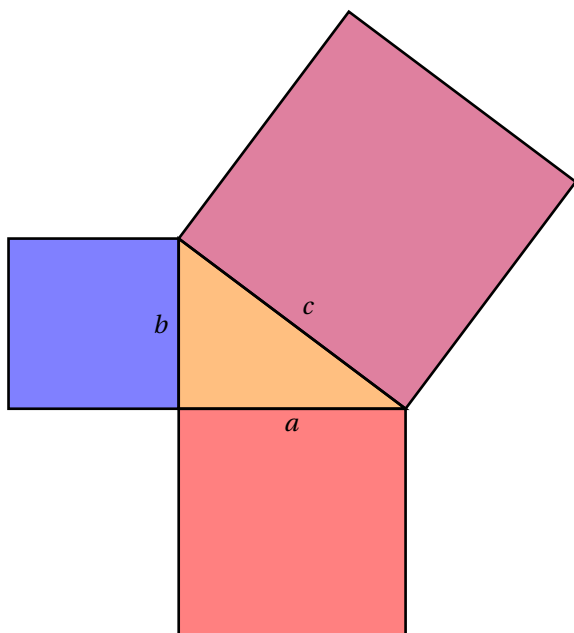
可以使用`\tkzGetFirstPoint`或`\tkzGetSecondPoint`命令利用其中的1个点绘制等腰直角三角形。

16.1.2 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}
```

16.1.3 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}
```

16.2 \tkzDefParallelogram命令：定义平行四边形第 4 个顶点

可以通过 3 个点定义一个平行四边形。

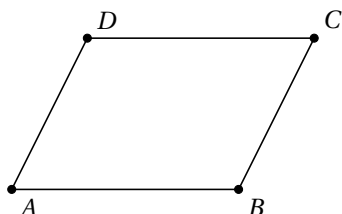
`\tkzDefParallelogram(<pt1,pt2,pt3>)`

通过 3 个点，通过计算另一个点，构成平行四边形，结果保存在\textkzPointResult中。

可使用\textkzGetPoint命令保存并命名结果...

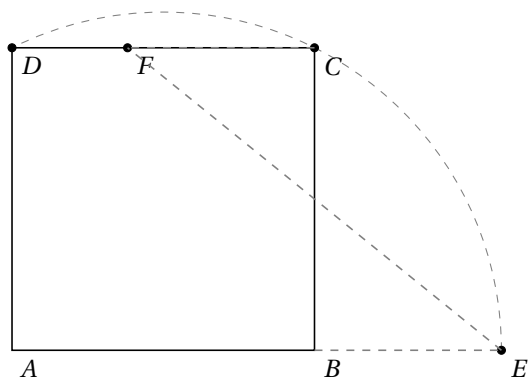
参数	默认值	含义
(<pt1,pt2,pt3>)	无	必须的 3 个顶点

16.2.1 平行四边形定义示例



```
\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}
```

16.2.2 黄金矩形示例



```
\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}
```

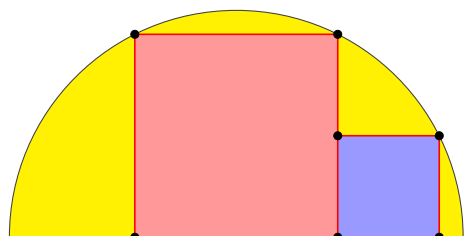
16.3 \tkzDrawSquare命令：绘制正方形

`\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
<code>(<pt1,pt2>)</code>	<code>\tkzDrawSquare(<A,B>)</code>	<code>\tkzGetPoints{C}{D}</code>
选项	样例	说明
TikZ 选项	<code>red,line width=1pt</code>	所有有效 TikZ 选项

16.3.1 在半圆内绘制两个正方形示例



```
\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}
```

16.4 \tkzDefGoldRectangle命令：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` 保存并命名这两个点。

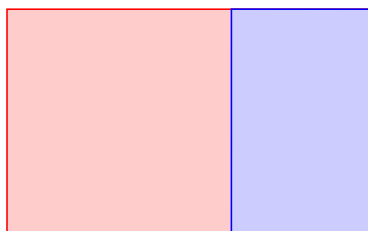
参数	样例	说明
<code>(\pt1,\pt2)</code>	<code>((A,B))</code>	如果用 C 和 D 表示得到的点，则 $AB/BC = \Phi$ 。

16.5 \tkzDrawGoldRectangle命令：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
TikZ 选项	<code>red,line width=1pt</code>	所有有效 TikZ 选项

16.5.1 黄金矩形示例

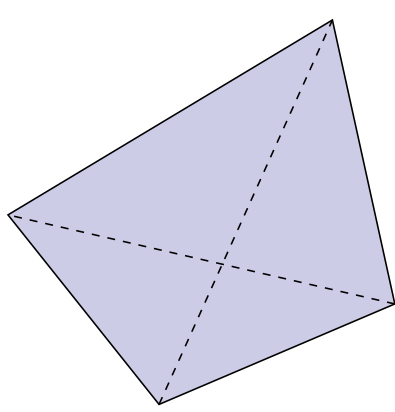


```
\begin{tikzpicture}[scale=.6]
\tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
\tkzDefGoldRectangle(A,B) \tkzGetPoints{C}{D}
\tkzDefGoldRectangle(B,C) \tkzGetPoints{E}{F}
\tkzDrawPolygon[color=red,fill=red!20](A,B,C,D)
\tkzDrawPolygon[color=blue,fill=blue!20](B,C,E,F)
\end{tikzpicture}
```

16.6 \tkzDrawPolygon命令：绘制多边形

\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,...,E) 表示点集 (A,B,C,D,E)，用 (P ₁ ,P...,P ₅) 表示点集 (P ₁ ,P ₂ ,P ₃ ,P ₄ ,P ₅)。		
参数	样例	说明
(<pt1,pt2,pt3,...>)	\tkzDrawPolygon[gray,dashed](A,B,C)	绘制一个三角形
选项	默认值	样例
TikZ 选项 ...		\tkzDrawPolygon[red,line width=2pt](A,B,C)

16.6.1 \tkzDrawPolygon命令示例

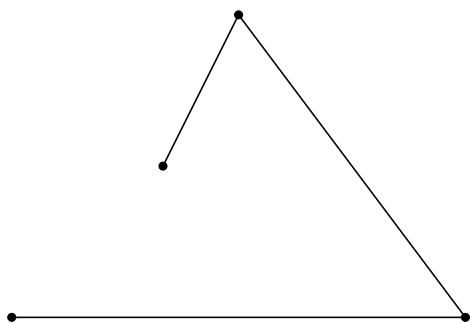


```
\begin{tikzpicture}[rotate=18,scale=1.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2.25,0.2){B}
\tkzDefPoint(2.5,2.75){C}
\tkzDefPoint(-0.75,2){D}
\tkzDrawPolygon[fill=black!50!blue!20!](A,B,C,D)
\tkzDrawSegments[style=dashed](A,C B,D)
\end{tikzpicture}
```

16.7 \tkzDrawPolySeg命令：绘制多边形顶点折线

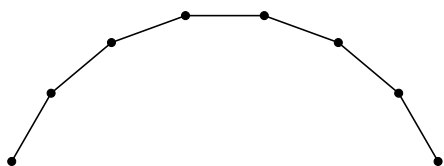
\tkzDrawPolySeg[< 命令选项>](< 点集列表>)		
绘制多边形顶点构成的折线。		
参数	样例	说明
(<pt1,pt2,pt3,...>)	\tkzDrawPolySeg[gray,dashed](A,B,C)	绘制一个三角形
选项	默认值	样例
TikZ 选项 ...		\tkzDrawPolySeg[red,line width=2pt](A,B,C)

16.7.1 多边形顶点折线示例



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

16.7.2 多边形顶点折线：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}
```

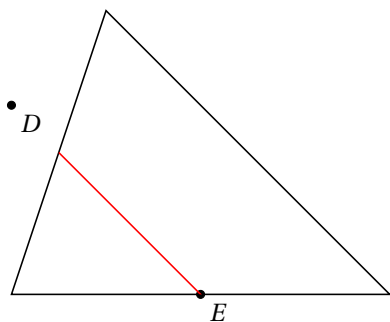
16.8 \tkzClipPolygon命令：使用多边形裁剪

`\tkzClipPolygon[< 命令选项>](< 点集列表>)`

用指定的多边形对图形进行裁剪。

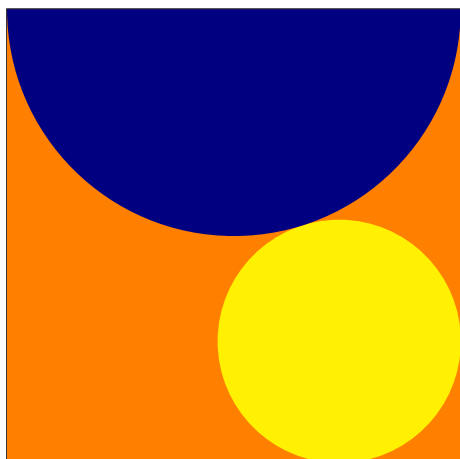
参数	样例	说明
<code>(\pt1,\pt2)</code>	<code>((A,B))</code>	

16.8.1 \tkzClipPolygon命令示例



```
\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}
```


16.8.2 使用"裁剪"将 Sangaku 图形限制在正方形内



```
\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}
```

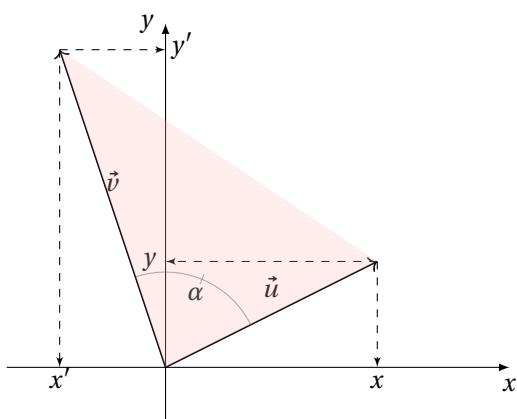
16.9 \tkzFillPolygon命令：多边形着色

`\tkzFillPolygon[< 命令选项>](< 点集列表>)`

可以在对多边形着色，但该命令仅对内部着色，不绘制多边形。

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

16.9.1 \tkzFillPolygon命令示例



```
\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}
```

16.10 \tkzDefRegPolygon命令：定义正多边形

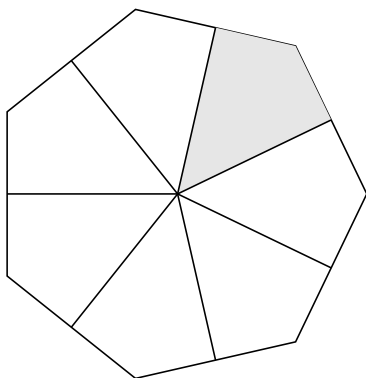
`\tkzDefRegPolygon[< 命令选项>](<pt1,pt2>)`

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
$(\langle pt1, pt2 \rangle)$	$(\langle O, A \rangle)$	如果使用"center"选项，则 O 是多边形中心
$(\langle pt1, pt2 \rangle)$	$(\langle A, B \rangle)$	如果使用"side"选项， $[AB]$ 一条边

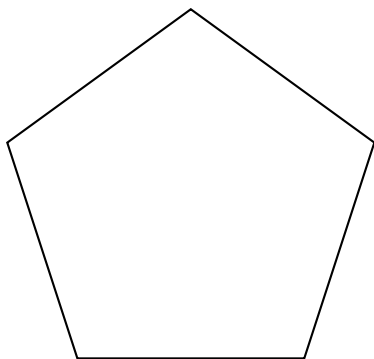
选项	默认值	样例
name	P	顶点命名为 P_1, P_2, \dots
sides	5	边数
center	center	第 1 个点是正多边形中心
side	center	指定的两个顶点构成一条边
TikZ 选项	...	

16.10.1 center选项示例



```
\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}
```

16.10.2 side选项示例



```
\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}
```

17 圆

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

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

17.1 `\tkzDefCircle`命令：定义圆

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

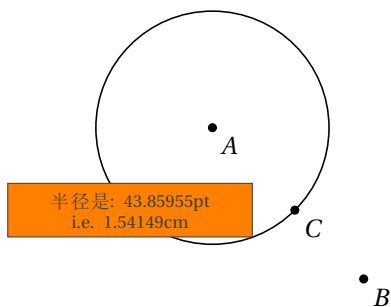
注意，参数可以是 2 个或 3 个点。该命令结合`\tkzGetPoint`命令和/或`\tkzGetLength`命令，得到圆心和圆的半径，或使用`\tkzPointResult`命令和`\tkzLengthResult`命令使用这些值，但不命名。

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

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

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

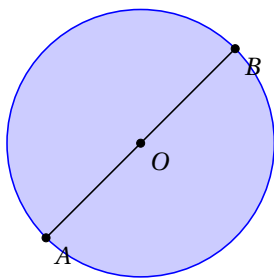
17.1.1 使用随机点与through选项示例



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

17.1.2 diameter选项示例

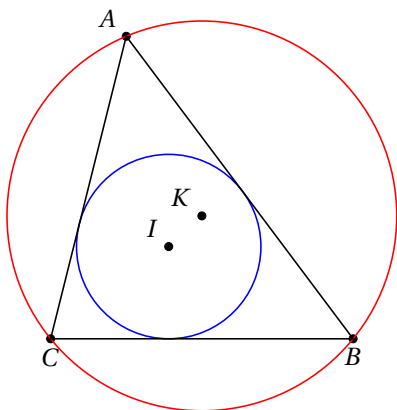
可以通过 $[AB]$ 的中点确定圆心。



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

17.1.3 三角形的内切圆和外接圆示例

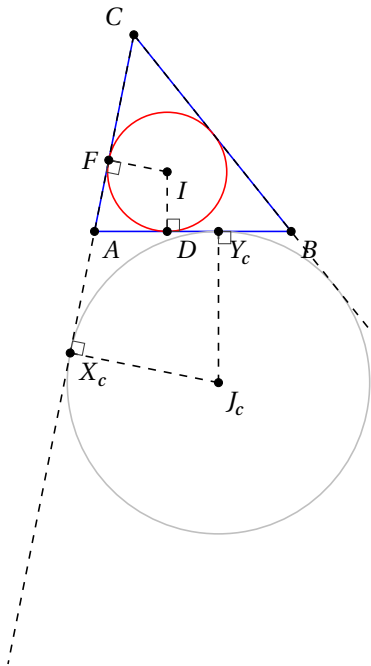
可以使用 `\tkzGetFirstPoint{I}` 和 `\tkzGetSecondPoint{Ib}` 命令得到内切圆在对应边上的投影。



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

17.1.4 ex选项示例

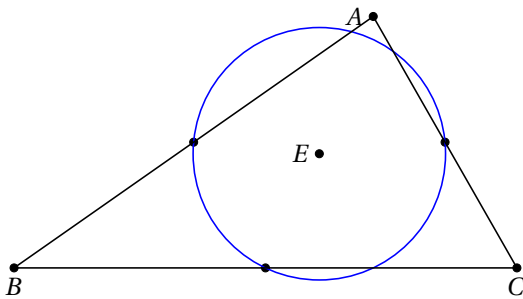
与顶点 C 对应的旁切圆。



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

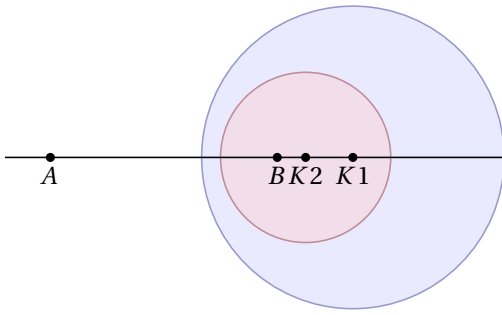
17.1.5 euler选项示例

同时验证了欧拉圆会通过三角形三个边的中点。



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

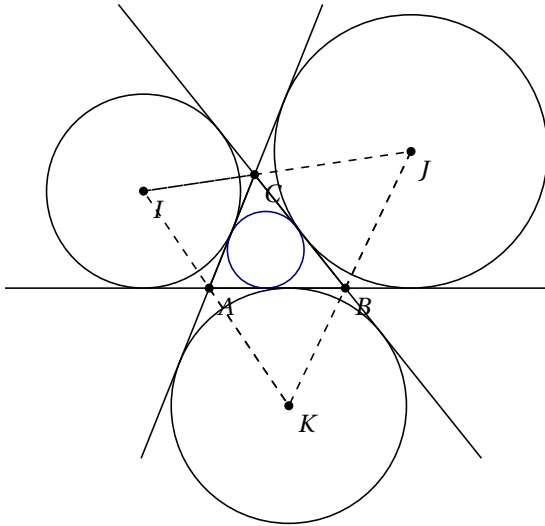
17.1.6 apollonius选项示例



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

17.1.7 ex选项示例

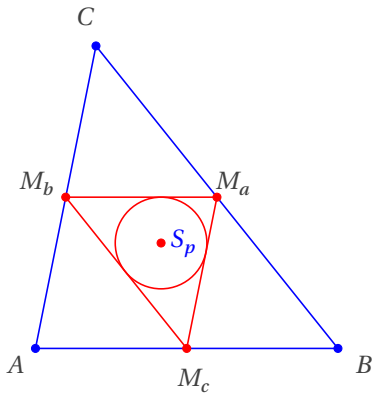
可以使用`\tkzGetFirstPoint{Jb}`和`\tkzGetSecondPoint{Tb}`命令，得到旁切圆圆心在边上的投影。



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

17.1.8 spieker选项示例

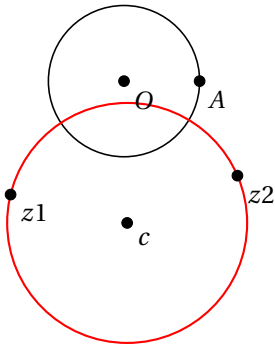
三角形三个边的中点构成的三角形 $M_aM_bM_c$ 的内切圆是 Spieker 圆:



```
\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}
```

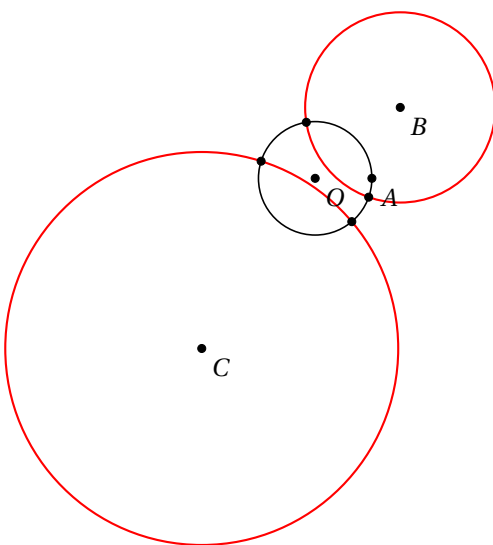
17.1.9 orthogonal through选项示例

过指定两个点的另一个圆的正交圆。



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){A}
  \tkzDrawCircle(0,A)
  \tkzDefPoint(-1.5,-1.5){z1}
  \tkzDefPoint(1.5,-1.25){z2}
  \tkzDefCircle[orthogonal through=z1 and z2](0,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}
```

17.1.10 指定圆心的另一个圆的正交圆示例



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

18 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.

18.1 `\tkzDrawCircle`命令: 绘制圆

`\tkzDrawCircle[⟨ 命令选项 ⟩](⟨A,B⟩)`



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

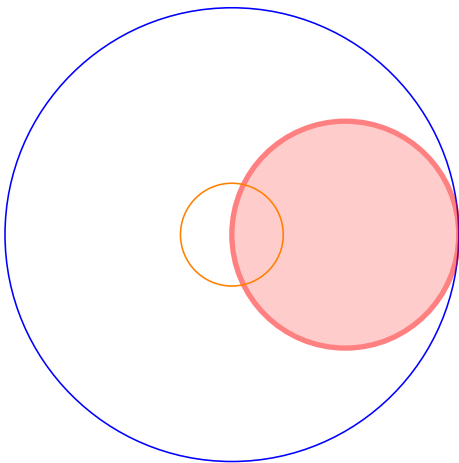
参数	样例	说明
<code>(⟨pt1,pt2⟩)</code>	<code>(⟨A,B⟩)</code>	两个点定义半径或直径

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

可使用所有有效 TikZ 样式。

18.1.1 绘制一个圆并对其进行着色

能够在绘制中对圆进行着色。



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


18.2 \tkzDrawCircles命令：绘制多个圆

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

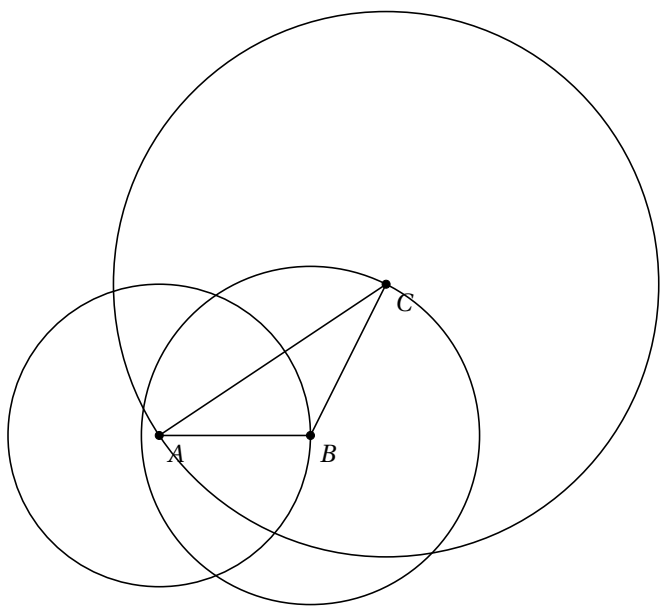
 注意：参数是空格分隔的构成圆的点对列表，点对中的两个点之间用逗号分隔。使用**R**选项，则需要直接指定半径。

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

选项	默认值	含义
through	through	两个点定义半径
diameter	through	两个点定义直径
R	through	通过指定半径

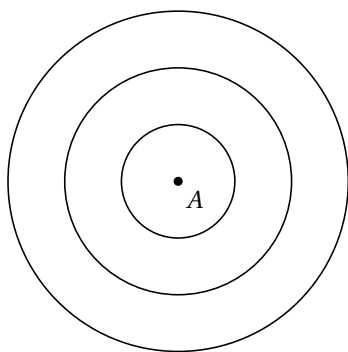
当然，可以使用所有的有效 TikZ 样式。

18.2.1 通过三角形定义圆示例



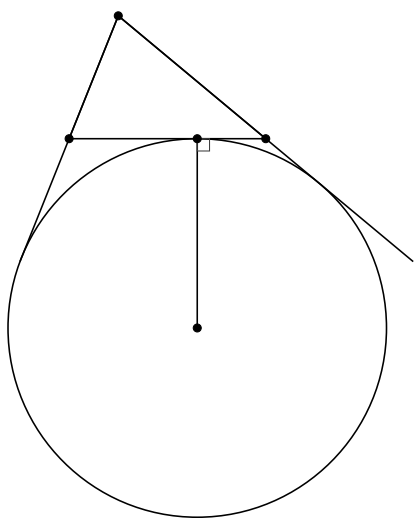
```
\begin{tikzpicture}[scale=1.0]
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,0){B}
\tkzDefPoint(3,2){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawCircles(A,B B,C C,A)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

18.2.2 同心圆示例



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

18.2.3 旁切圆示例

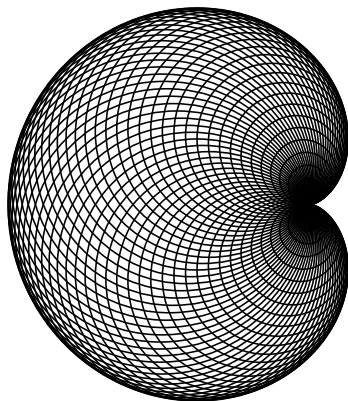


```
\begin{tikzpicture}[scale=0.65]
  \tkzDefPoints{0/0/A,4/0/B,1/2.5/C}
  \tkzDrawPolygon(A,B,C)
  \tkzDefCircle[ex](B,C,A)
  \tkzGetPoint{J_c} \tkzGetSecondPoint{T_c}
  \tkzGetLength{rJc}
  \tkzDrawCircle[R](J_c,{\rJc pt})
  \tkzDrawLines[add=0 and 1](C,A C,B)
  \tkzDrawSegment(J_c,T_c)
  \tkzMarkRightAngle(J_c,T_c,B)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

18.2.4 心形线示例

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}
```

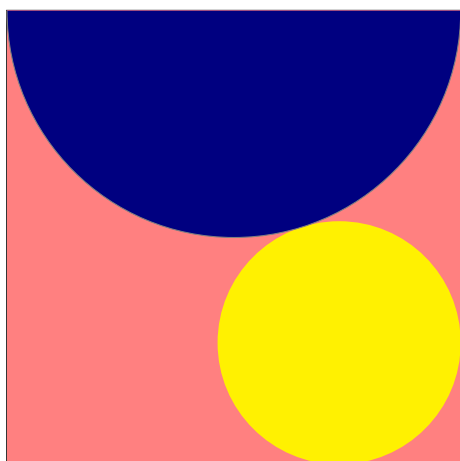
18.3 \tkzDrawSemiCircle命令：绘制半圆

`\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>	两个点定义半径
<code>diameter</code>	<code>through</code>	两个点定义直径

18.3.1 示例



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

18.4 \tkzFillCircle命令：给圆着色

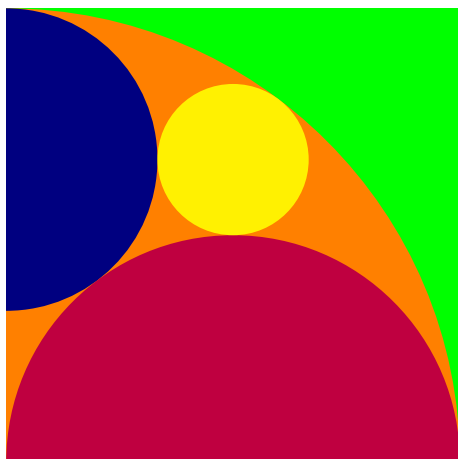
在绘制圆时，也可以实现着色，但该命令不绘制圆，仅对圆形区域进行着色。

`\tkzFillCircle[< 命令选项>](⟨A,B⟩)`

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

可以使用所有有效 TikZ 样式。

18.4.1 sangaku 圆示例



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

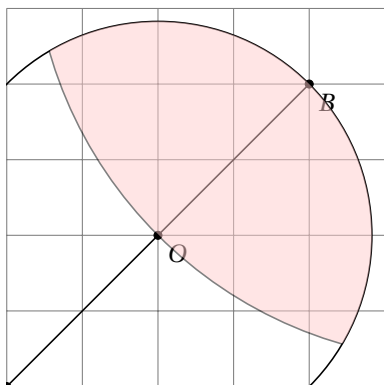
18.5 \tkzClipCircle命令：用圆裁剪

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

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

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

18.5.1 示例



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

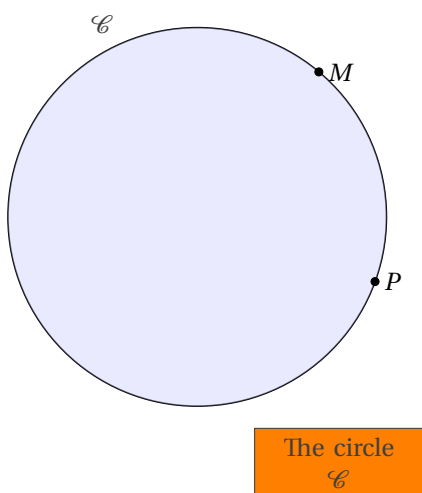
18.6 `\tkzLabelCircle`命令：为圆添加标注

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

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

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

18.6.1 标注示例



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

19 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。

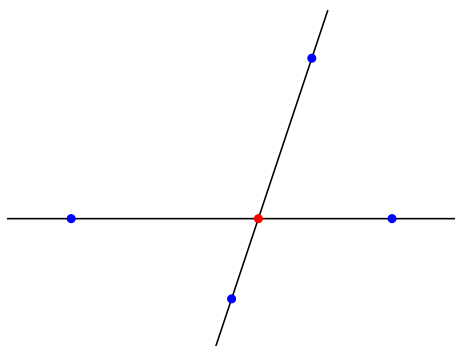
求交点的相关命令没有可选参数，用户必须确保交点存在。

19.1 `\tkzInterLL`命令：求两条直线的交点

`\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点对定义。可以通过 `\tkzDefPoint` 保存并命名交点。

19.1.1 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}
```

19.2 `\tkzInterLC`命令：定义一条直线和一个圆的交点

直线可以由两个点定义，圆可以按如下方式进行定义：

- (O,C) O 是圆心， C 是圆上的一个点。
- (O,r) O 是圆心， r 半径，单位可以是 *cm* 或 *pt*。

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

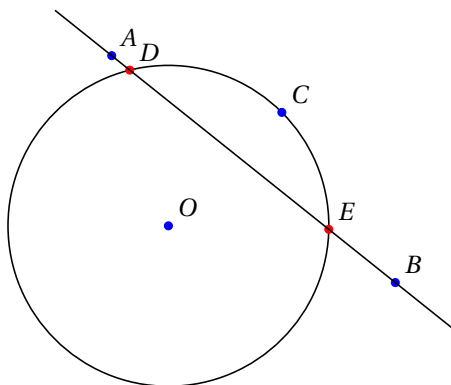
参数必须是一条直线和一个圆。

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

定义直线与由圆心 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 `.log` 日志文件中。

19.2.1 直线与圆的交点示例

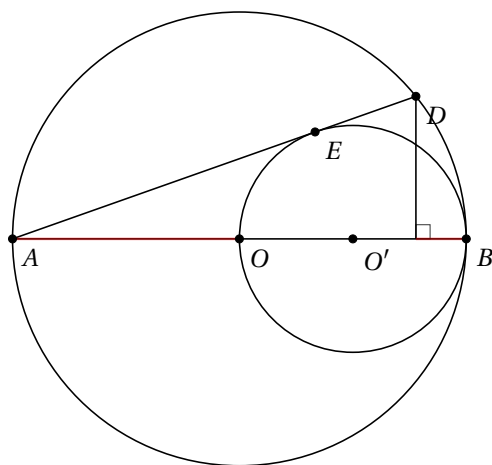
在下面示例代码中，圆用两个点表示，直线与圆的有两个交点。



```
\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}
```

19.2.2 直线与圆的交点复杂示例

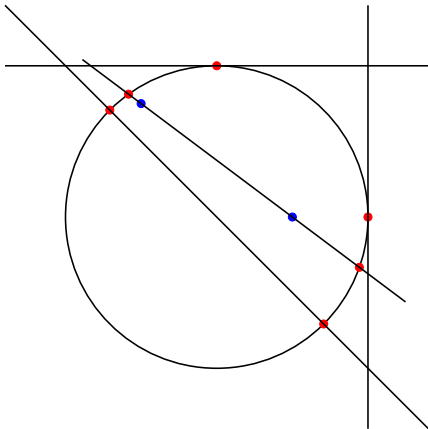
可参阅http://www.gogeometry.com/problem/p190_tangent_circle_diameter_perpendicular.htm



```
\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}
```


19.2.3 由圆心和半径定义圆示例

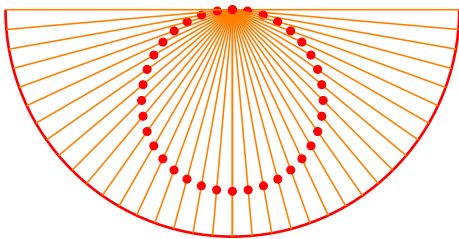
直线与圆相切的特例：



```
\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}
```

19.2.4 更为复杂的示例

 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，如使用`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)(O,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)(O,1 cm)
\tkzGetPoints{M}{N}
\tkzDrawPoints[red](M)
}
\end{tikzpicture}
```

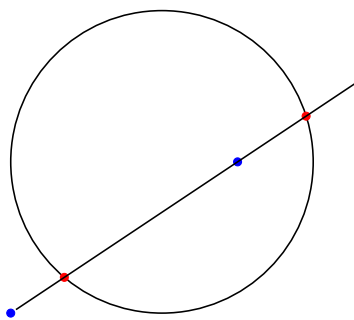
19.3 半径计算

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。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，其精度不足，如 $0.0002 \div 0.0001$ 的结果是1.98，但如使用`xfp`宏包进行计算，则其结果为2。

19.3.1 半径计算示例 1

使用`xfp`宏包的`\fpeval`命令：

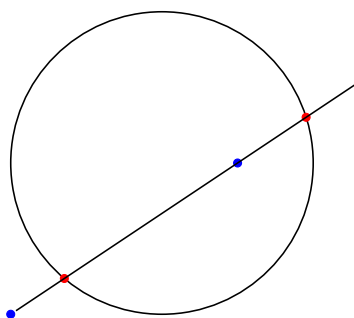


```
\begin{tikzpicture}
\tkzDefPoint(2,2){A}
\tkzDefPoint(5,4){B}
\tkzDefPoint(4,4){O}
\edef\tkzLen{\fpeval{0.0002/0.0001}}
\tkzDrawCircle[R](O,\tkzLen cm)
\tkzInterLC[R](A,B)(O,\tkzLen cm)
\tkzGetPoints{I}{J}
\tkzDrawPoints[color=blue](A,B)
\tkzDrawPoints[color=red](I,J)
\tkzDrawLine(I,J)
\end{tikzpicture}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

19.3.2 半径计算示例 2

使用 \TeX 的 `\tkzLength` 命令计算。

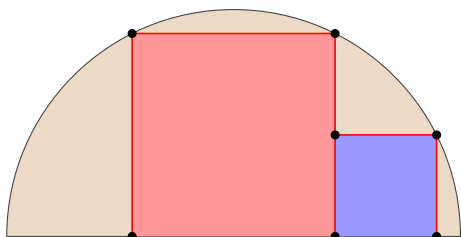
用 `\newdimen` 命令定义一尺寸。当然，用的是 \TeX 进行计算。



```
\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}
```

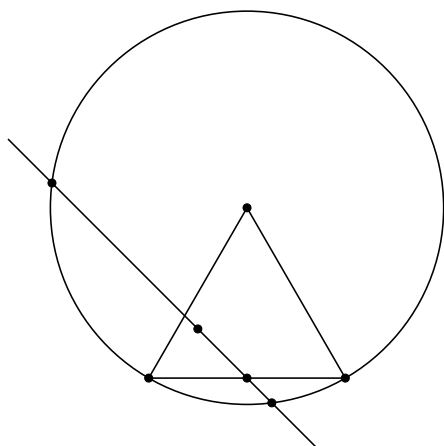
19.3.3 半圆中的矩形示例

在一个半圆内同时画两个正方形是比较困难的，需要通过圆的半径计算正方形 S 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}
```

19.3.4 "with nodes"选项示例



```
\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}
```

19.4 \tkzInterCC命令：求两个圆的交点

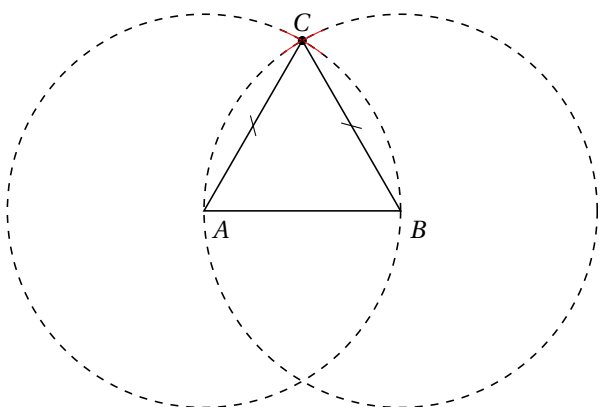
通常，两个圆是由圆心和另一个点确定的，但也可以用R选项后，在参数中指定半径。

`\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

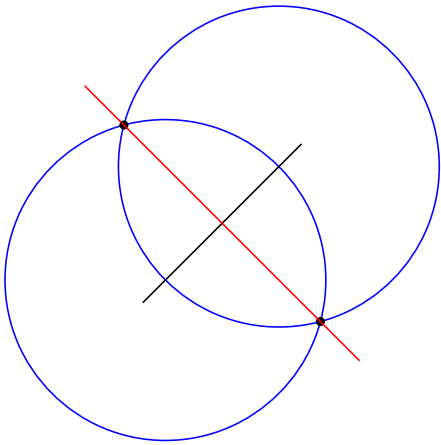
该命令定义了 O 和 O' 两个圆的交点 I 和 J ，如果两个圆没有交点，则返回错误。也可以直接使用 `\tkzInterCCN` 命令和 `\tkzInterCCR` 命令进行计算。

19.4.1 构造等边三角形示例



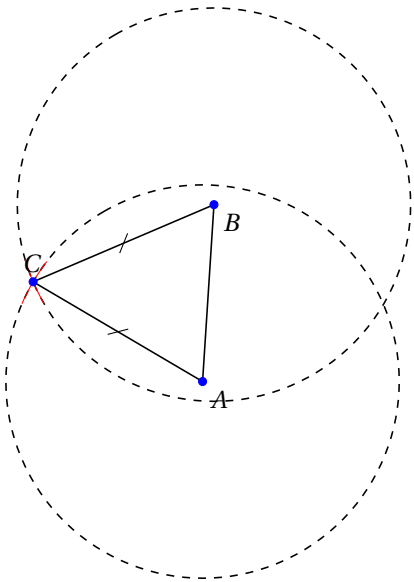
```
\begin{tikzpicture}[trim left=-1cm,scale=0.65]
\tkzDefPoint(1,1){A}
\tkzDefPoint(5,1){B}
\tkzInterCC(A,B)(B,A)\tkzGetPoints{C}{D}
\tkzDrawPoint[color=black](C)
\tkzDrawCircle[dashed](A,B)
\tkzDrawCircle[dashed](B,A)
\tkzCompass[color=red](A,C)
\tkzCompass[color=red](B,C)
\tkzDrawPolygon(A,B,C)
\tkzMarkSegments[mark=s|](A,C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

19.4.2 求中点示例



```
\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}
```

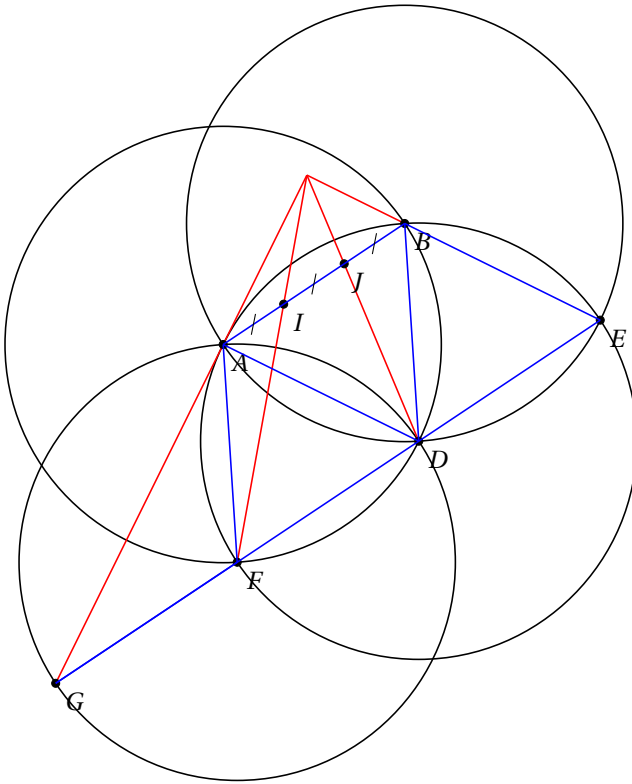
19.4.3 求等腰三角形示例



```
\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}
```

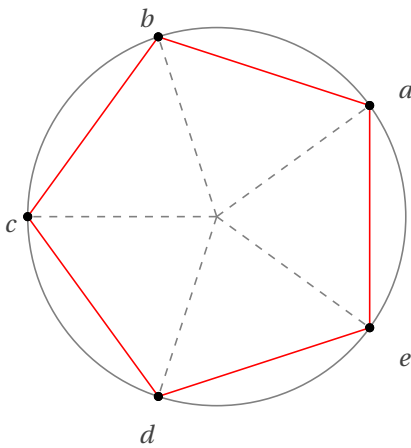
19.4.4 三等分线段示例

与尺规作图方式将线段三等分操作一致。



```
\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}
```

19.4.5 with nodes 选项示例



```
\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}
```

20 角

20.1 \tkzFillAngle命令：填充角

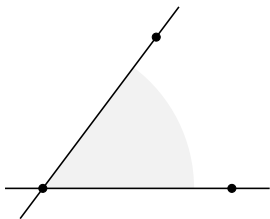
<code>\tkzFillAngle[< 命令选项>](A,O,B)</code>
--

O 是角顶点, OA 和 OB 是两条边, 点的顺序决定角的方向。

选项	默认值	含义
size	1 cm	着色扇形的半径

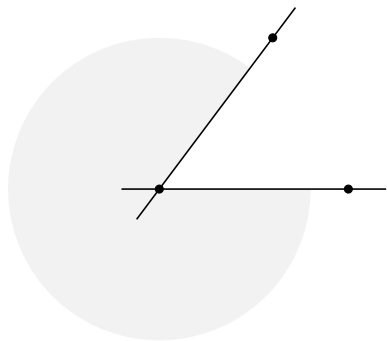
可以使用所有有效的 TikZ 样式, 如 fill 和 shade 等。

20.1.1 size选项示例

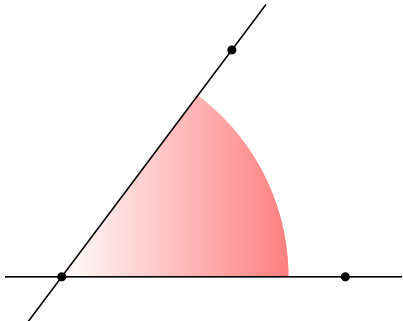


```
\begin{tikzpicture}
\tkzInit
\tkzDefPoints{0/0/0,2.5/0/A,1.5/2/B}
\tkzFillAngle[size=2cm, fill=gray!10](A,O,B)
\tkzDrawLines(O,A O,B)
\tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

20.1.2 改变点的顺序示例



```
\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}
```



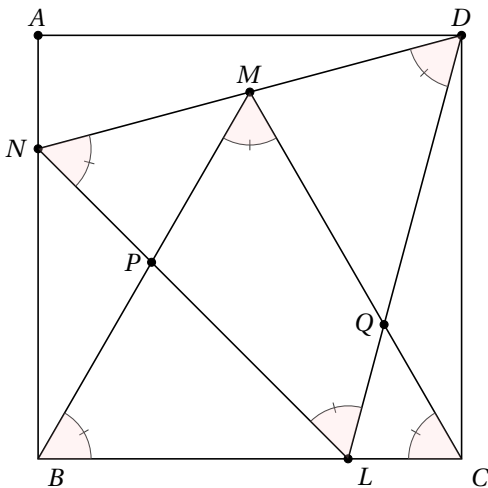
```
\begin{tikzpicture}[scale=0.75]
\tkzInit
\tkzDefPoints{0/0/0,5/0/A,3/4/B}
% Don't forget {} to get, () to use
\tkzFillAngle[size=4cm,left color=white,
right color=red!50](A,O,B)
\tkzDrawLines(O,A O,B)
\tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

20.2 \tkzFillAngles命令：填充多个角

`\tkzFillAngles[⟨命令选项⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)` 等

绘制多个角。

20.2.1 填充多个角示例



```
\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}
```

20.3 标记角

在 TikZ 中，绘图选项非常丰富，本宏包又增加了需要的一些标记，它们定义在 `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：

```
\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
}

```

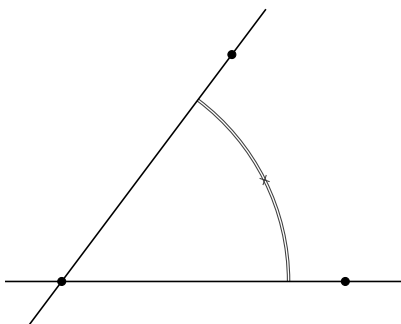
20.4 \tkzMarkAngle命令：标记角

`\tkzMarkAngle[< 命令选项>](A,O,B)`

O 是顶点，注意参数需随选项变化。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，也可为圆弧着色。

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

20.4.1 mark = x 选项示例

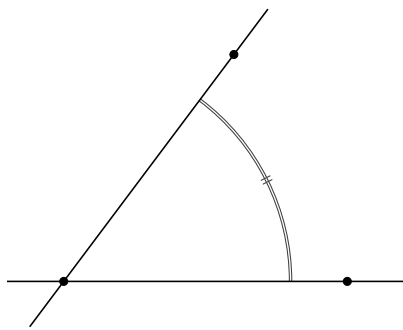


```

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

```


20.4.2 mark = || 选项示例



```
\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}
```

20.5 \tkzMarkAngles命令：标记多个角

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

对于具有相同选项的多个标记，可以一次标记多个角。

20.6 \tkzLabelAngle命令：标注角

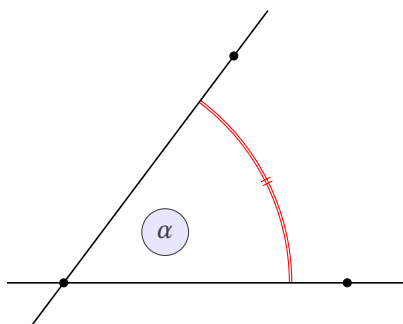
`\tkzLabelAngle[< 命令选项>](A,O,B)`

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`。

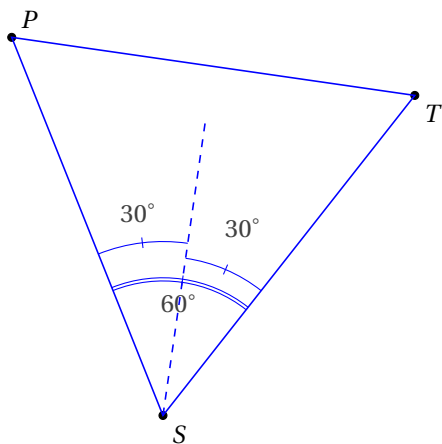
选项	默认值	含义
<code>pos</code>	<code>1</code>	或是 <code>dist</code> ，用于控制标注的距离

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

20.6.1 pos选项示例



```
\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=11,
    color = blue](T,S,P)
  \tkzMarkAngle[size = 2.1cm,mark = |,arc=1,
    color = blue](T,S,s)
  \tkzMarkAngle[size = 2.3cm,mark = |,arc=1,
    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}
```

20.7 \tkzLabelAngles命令：标注多个角

`\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。

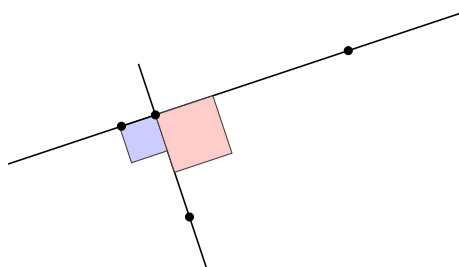
20.8 \tkzMarkRightAngle命令：标记直角

`\tkzMarkRightAngle[< 命令选项>](A,O,B)`

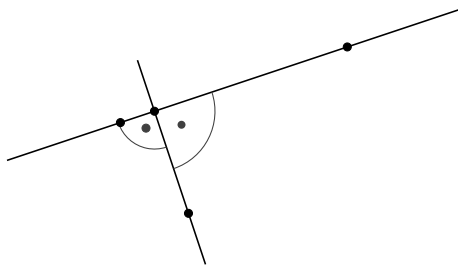
german选项用于改变样式，**size**选项用于改变尺寸。

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

20.8.1 直角标记示例

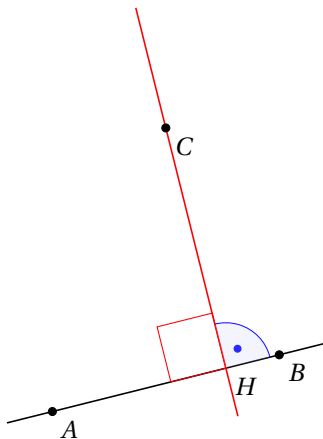


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
  \tkzDefPointBy[projection = onto B--A](P)
  \tkzGetPoint{H}
  \tkzDrawLines[add=.5 and .5](P,H)
  \tkzMarkRightAngle[fill=blue!20,size=.5,draw](A,H,P)
  \tkzDrawLines[add=.5 and .5](A,B)
  \tkzMarkRightAngle[fill=red!20,size=.8](B,H,P)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

20.8.2 使用 `german` 样式添加直角标记

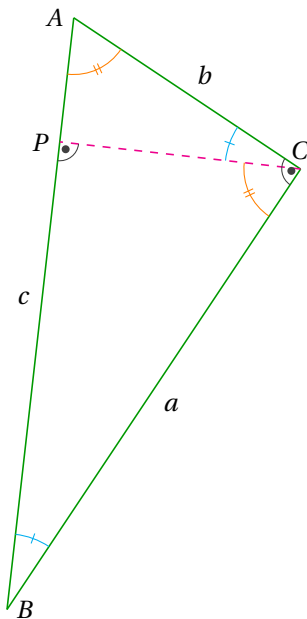
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
\tkzDefPointBy[projection = onto B--A](P)
\tkzGetPoint{H}
\tkzDrawLines[add=.5 and .5](P,H)
\tkzMarkRightAngle[german,size=.5,draw](A,H,P)
\tkzDrawPoints[] (A,B,P,H)
\tkzDrawLines[add=.5 and .5,fill=blue!20](A,B)
\tkzMarkRightAngle[german,size=.8](P,H,B)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

20.8.3 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}
```

20.8.4 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=1cm,color=cyan,mark=|](A,C,P)
\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}
```

20.9 \tkzMarkRightAngles命令：标记多个直角

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

当选项相同时，使用该命令标记多个直角。

21 角度测量命令

21.1 \tkzGetAngle命令：计算角度

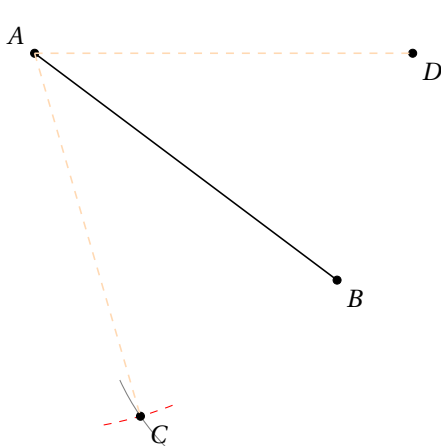
`\tkzGetAngle(< 宏名称>)`

将角度值以度为单位存入指定宏，可用`\tkzAngleResult`命令得到角度值并存入指定宏。

参数	样例	含义
宏名称	<code>\tkzGetAngle{ang}</code>	<code>\ang</code> 保存了角度值（度）

21.1.1 \tkzGetAngle命令示例

直线 (AB) 是角 \widehat{CAD} 的角平分线, 因此 AD 的斜率为 0, 得到 (AB) 的斜率后将其旋转 2 次。



```
\begin{tikzpicture}
  \tkzInit
  \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[length=1,dashed,color=red](A,C)
  \tkzCompass[delta=10,brown](B,C)
  \tkzDrawPoints(A,B,C,D)
  \tkzLabelPoints(B,C,D)
  \tkzLabelPoints[above left](A)
  \tkzDrawSegments[style=dashed,color=orange!30](A,C A,D)
\end{tikzpicture}
```

21.2 \tkzFindAngle命令：计算三个点定义的角的角度

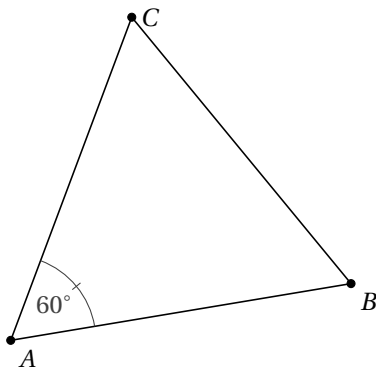
$$\backslash\text{tkzFindAngle}(\langle\text{pt1},\text{pt2},\text{pt3}\rangle)$$

结果保存在\tkzAngleResult中。

参数	样例	说明
(pt1,pt2,pt3)	<code>\tkzFindAngle(A,B,C)</code>	<code>\tkzAngleResult</code> 是 $(\overrightarrow{BA}, \overrightarrow{BC})$ 之间的角度

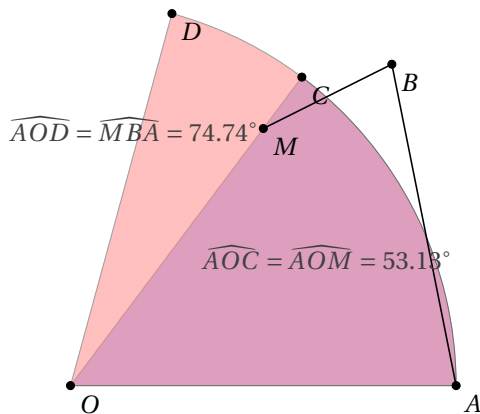
结果在-180 度到 +180 度之间，pt2 是顶点，可以使用\tkzGetAngle得到结果。

21.2.1 角度测量示例



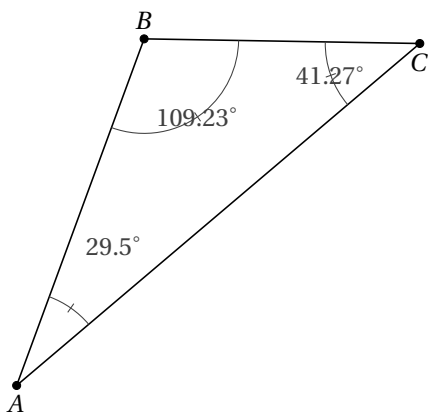
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(-1,1){A} \tkzDefPoint(5,2){B}
  \tkzDefEquilateral(A,B)
  \tkzGetPoint{C}
  \tkzDrawPolygon(A,B,C)
  \tkzFindAngle(B,A,C)
  \tkzGetAngle{angleBAC}
  \edef\angleBAC{\fpeval{round(\angleBAC)}}
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B)
  \tkzLabelPoint[right](C){C}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

21.3 角度计算示例



```
\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}
```

21.3.1 三角形内角计算示例



```
\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}
```

21.4 \tkzFindSlope命令：计算斜率

斜率由直线上两个点确定，该命令不检测其存在性。

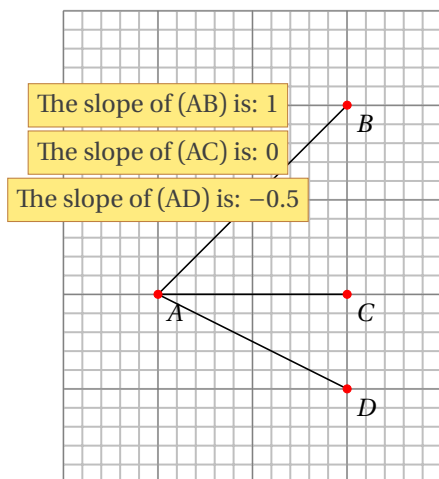
`\tkzFindSlope(<pt1,pt2>){<宏名称>}`

斜率保存在指定宏中。

参数	样例	说明
(pt1,pt2)pt3 计算	<code>\tkzFindSlope(A,B){slope}</code>	<code>\slope</code> 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}
```

21.5 \tkzFindSlopeAngle命令：计算直线与横轴夹角

结果在-180 度与 +180 度之间。

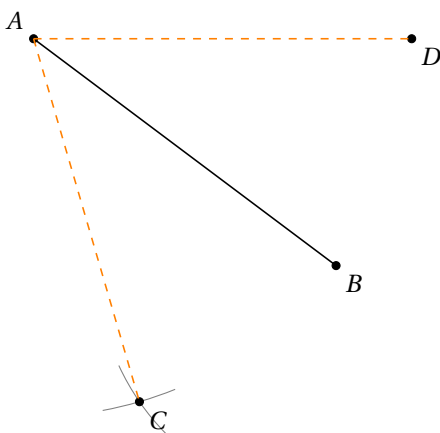
\tkzFindSlopeAngle($\langle A,B \rangle$)

计算直线 (AB) 的斜率并保存在 \tkzAngleResult 命令中。

参数	样例	说明
(pt1,pt2)	\tkzFindSlopeAngle(A,B)	

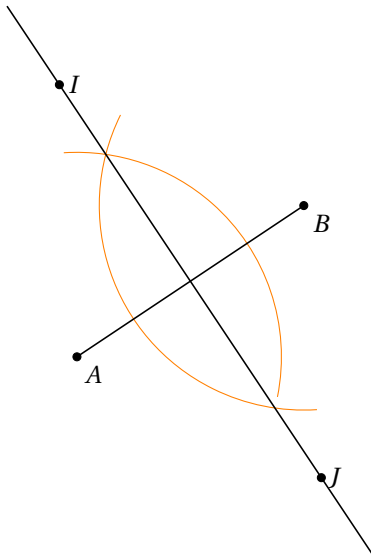
用 \tkzGetAngle 保存结果并命名

21.5.1 折叠示例



```
\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}
```

21.5.2 中点计算示例



```

\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}

```


22 扇形

22.1 \tkzDrawSector命令：绘制扇形

 注意参数需要根据选项变化。

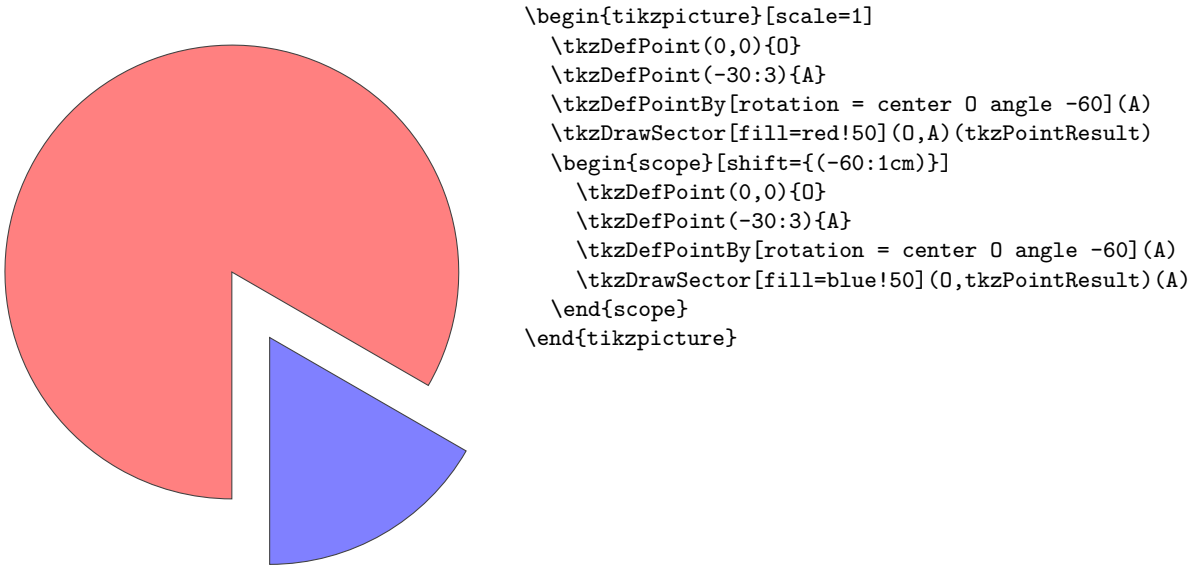
\tkzDrawSector[< 命令选项>](<O,...>)(<...>)		
选项	默认值	含义
towards	towards	O 是圆心并且圆弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始并且用角度确定长度
R	towards	给定半径和两个角度
R with nodes	towards	给定半径和两个点

可以使用所有有效的 TikZ 样式。

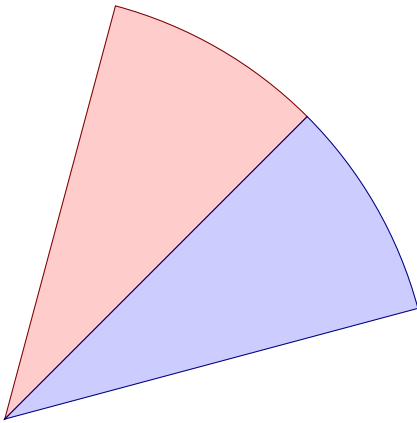
选项	参数	样例
towards	(<pt,pt>)(<pt>)	\tkzDrawSector(O,A)(B)
rotate	(<pt,pt>)(<an>)	\tkzDrawSector[rotate,color=red](O,A)(90)
R	(<pt,r>)(<an,an>)	\tkzDrawSector[R,color=blue](O,2 cm)(30,90)
R with nodes	(<pt,r>)(<pt,pt>)	\tkzDrawSector[R with nodes](O,2 cm)(A,B)

22.1.1 towards选项示例

towards是默认选项，同时也可以使用fill选项。

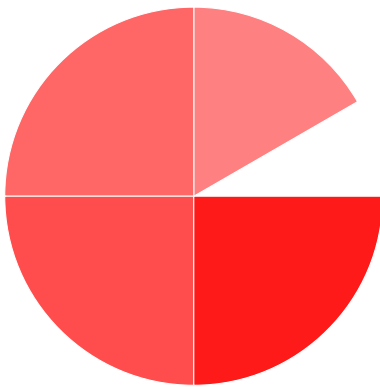


22.1.2 rotate选项示例



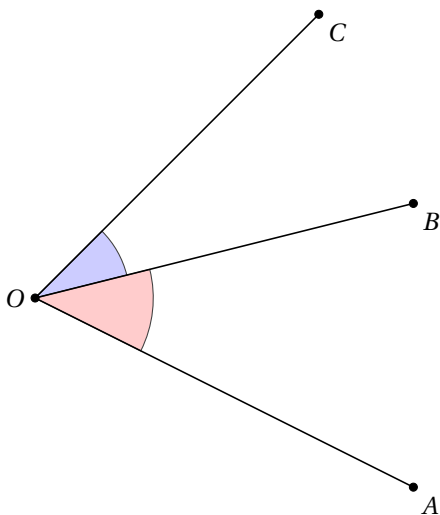
```
\begin{tikzpicture}[scale=2]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,2){A}
\tkzDrawSector[rotate,draw=red!50!black,%
fill=red!20](O,A)(30)
\tkzDrawSector[rotate,draw=blue!50!black,%
fill=blue!20](O,A)(-30)
\end{tikzpicture}
```

22.1.3 R选项示例



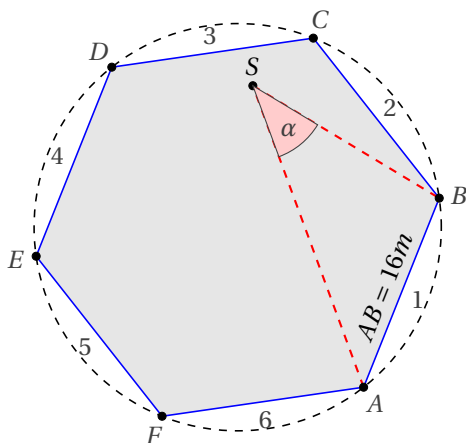
```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoint(0,0){O}
\tkzDefPoint(2,-1){A}
\tkzDrawSector[R,draw=white,%
fill=red!50](O,2cm)(30,90)
\tkzDrawSector[R,draw=white,%
fill=red!60](O,2cm)(90,180)
\tkzDrawSector[R,draw=white,%
fill=red!70](O,2cm)(180,270)
\tkzDrawSector[R,draw=white,%
fill=red!90](O,2cm)(270,360)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

22.1.4 R选项示例



```
\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}
```

22.1.5 R with nodes选项示例



```
\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){S}
\tkzLabelSector[R with nodes,fill=red!20](S,2 cm)(A,B)
\tkzLabelAngle[pos=1.5](A,S,B){\alpha}
\end{tikzpicture}
```

22.2 \tkzFillSector命令：填充扇形

 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)

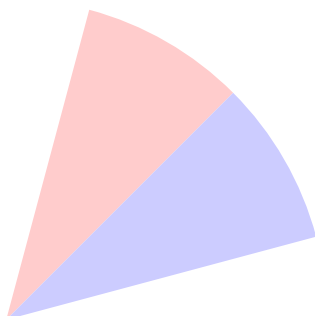
22.2.1 towards选项示例

towards是默认选项，该命令不绘制轮廓，仅对区域进行着色。




```
\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}
```

22.2.2 rotate选项示例



```
\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}
```

22.3 \tkzClipSector命令：扇形裁剪

 注意参数需要根据选项变化。

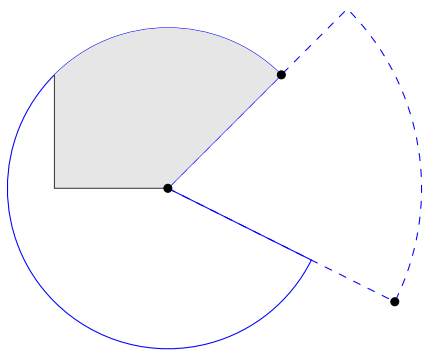
\tkzClipSector[< 命令选项>](<O,...>)(<...>)

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)

22.3.1 示例



```

\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}

```

23 圆弧

23.1 \tkzDrawArc命令：绘制圆弧

`\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。

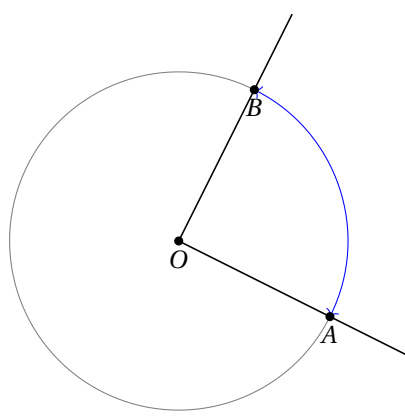
选项	默认值	含义
towards	towards	O 是圆心，并且圆弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始并且角度确定了长度
R	towards	给定半径和两个角度
R with nodes	towards	给定半径和两个点
angles	towards	给定半径和两个点
delta	0	角度加上两个边

可以使用所有有效的 TikZ 样式。

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

23.2 towards选项示例 1

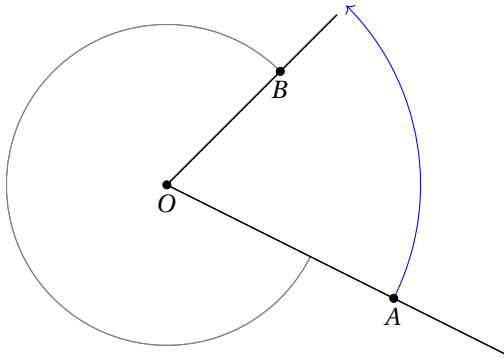
towards是默认选项，第 1 个例子中的圆弧从 A 开始到 B 结束。当然，从 B 到 A 会得到不同的结果。圆弧凸向由逆时针方向确定。



```
\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}
```

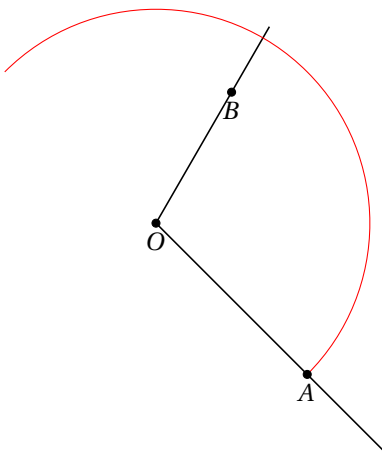
23.3 towards选项示例 2

该例中，圆弧从 A 开始，直到 (OB) 结束。



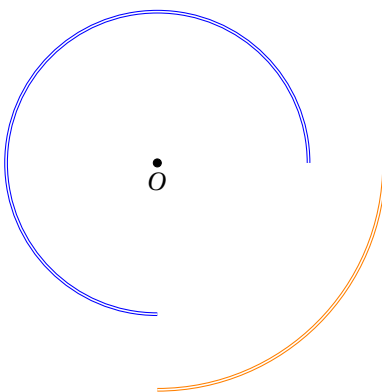
```
\begin{tikzpicture}[scale=1.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzDrawArc[color=blue,->](O,A)(B)
  \tkzDrawArc[color=gray](O,B)(A)
  \tkzDrawArc(O,B)(A)
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}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

23.4 rotate选项示例



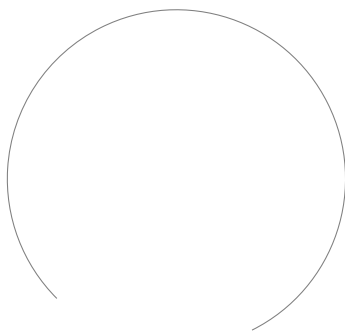
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-2){A}
  \tkzDefPoint(60:2){B}
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawArc[rotate,color=red](O,A)(180)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

23.5 R选项示例



```
\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}
```

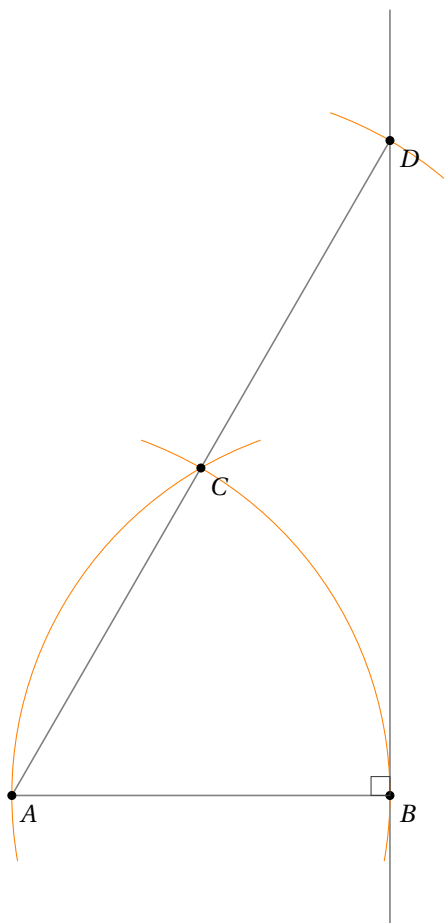
23.6 R with nodes选项示例



```
\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}
```

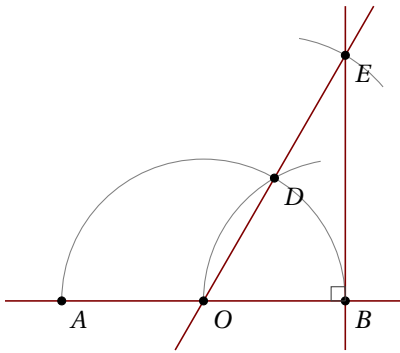
23.7 delta选项示例

该选项与`\tkzCompass`结果类似，它能够延伸圆弧，delta的单位是度。



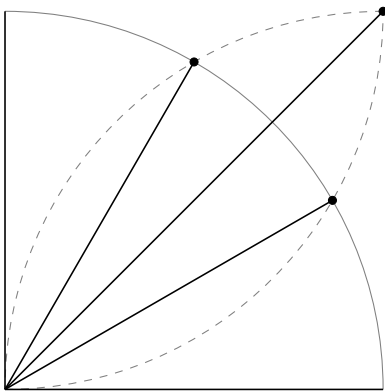
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,0){B}
\tkzDefPointBy[rotation= center A angle 60](B)
\tkzGetPoint{C}
\tkzSetUpLine[color=gray]
\tkzDefPointBy[symmetry= center C](A)
\tkzGetPoint{D}
\tkzDrawSegments(A,B A,D)
\tkzDrawLine(B,D)
\tkzSetUpCompass[color=orange]
\tkzDrawArc[orange,delta=10](A,B)(C)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


23.8 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}
```

23.9 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}
```

24 杂项命令

24.1 `\tkzDuplicateSegment`命令：复制线段

在给定的半直线上复制与给定线段长度相同的线段。

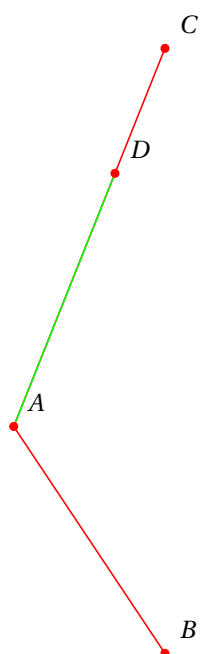
`\tkzDuplicateSegment(\langle pt1,pt2\rangle)(\langle pt3,pt4\rangle)`

该命令实际是一个点的定义命令。

`\tkzDuplicateSegment`命令是`\tkzDuplicateLen`命令的更新版本。

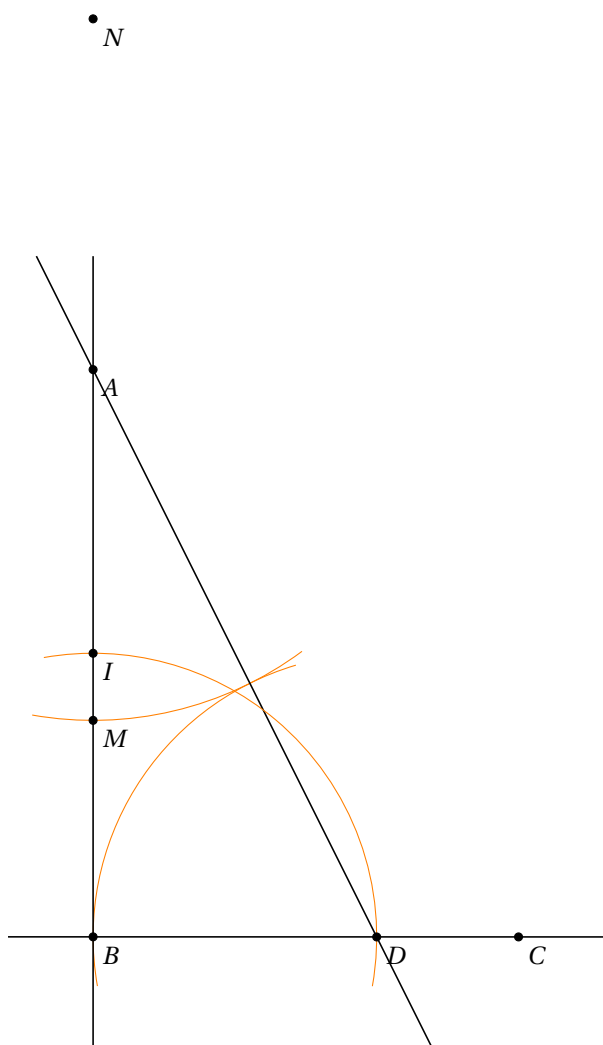
参数	样例	说明
<code>(pt1,pt2)(pt3,pt4)</code>	<code>\tkzDuplicateSegment(A,B)(E,F)</code>	$AC = EF$

`\tkzDuplicateLength`命令与该命令相同。



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(2,-3){B}
\tkzDefPoint(2,5){C}
\tkzDrawSegments[red](A,B A,C)
\tkzDuplicateSegment(A,B)(A,C)
\tkzGetPoint{D}
\tkzDrawSegment[green](A,D)
\tkzDrawPoints[color=red](A,B,C,D)
\tkzLabelPoints[above right=3pt](A,B,C,D)
\end{tikzpicture}
```

24.1.1 黄金分割示例



```

\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}

```

24.2 \tkzCalcLength命令：计算线段长度

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。

但 TikZ 计算精度不足，因此该命令用 **xfp** 宏包实现计算，虽然其计算慢，但精度高。

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

计算结果保存有指定的宏中。

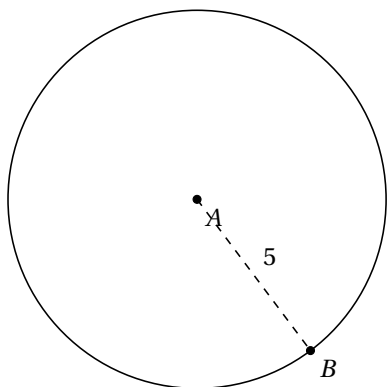
参数	样例	说明
(pt1,pt2){宏名称}	<code>\tkzCalcLength(A,B){dAB}</code>	<code>\dAB</code> 得到 AB 的长度，单位是 pt

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

24.4.1 示例代码

`\tkzDefCircle[radius](A,B)`命令定义一个圆，可以用`\tkzGetLength`命令得到半径，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}
```

24.5 `\tkzGetPointCoord`命令：提取点的坐标分量

`\tkzGetPointCoord(<A>){<宏名称>}`

参数

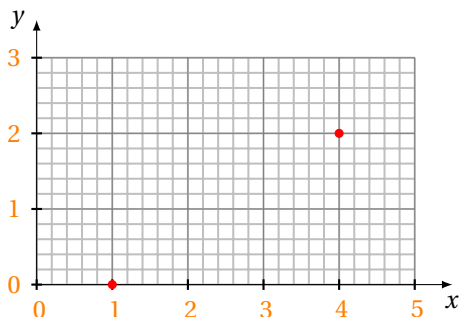
样例

说明

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

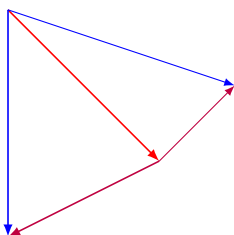
将点的坐标分量保存在两个宏中，如果宏名称是`p`，则将坐标分量保存在`\px`和`\py`宏中，单位是`cm`。

24.5.1 点定义示例



```
\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}
```

24.5.2 求向量和示例



```
\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}
```

25 尺规标记

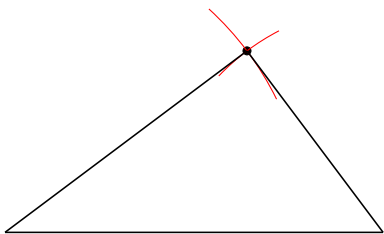
25.1 `\tkzCompass` 命令：绘制尺规标记

`\tkzCompass[⟨命令选项⟩](⟨A,B⟩)`

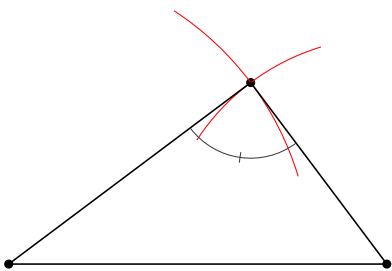
该命令绘制尺规标记，即一小段圆弧。使用该命令时，须指定圆心。可以使用 TikZ 的 `style`、`color`、`line thickness` 等样式设置标记外观。

可以使用 `length` 或 `delta` 选项指定标记长度。

选项	默认值	含义
<code>delta</code>	0 (deg)	延伸长度 (度)
<code>length</code>	1 (cm)	圆弧长度 (cm)

25.1.1 `length` 选项示例

```
\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}
```

25.1.2 `delta` 选项示例

```
\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}
```

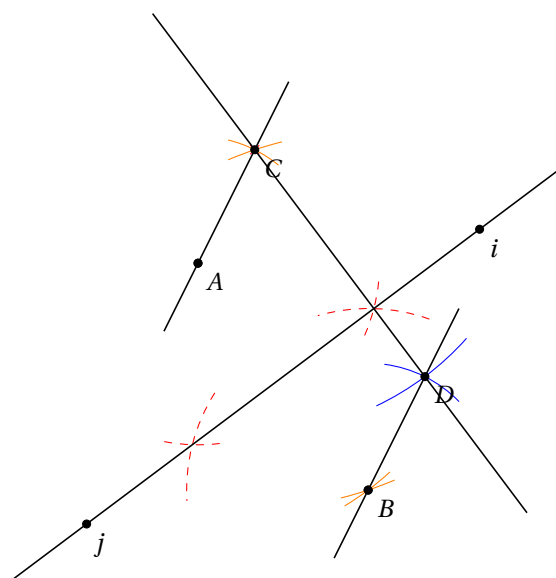
25.2 `\tkzCompassss` 命令：绘制多个尺规标记

`\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
<code>delta</code>	0	延伸角度
<code>length</code>	1	圆弧长度



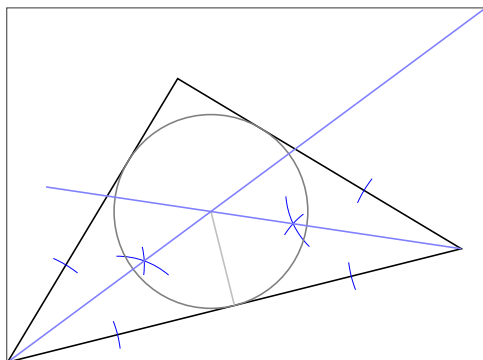
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(2,2){A}
  \tkzDefPoint(5,-2){B}
  \tkzDefPoint(3,4){C}
  \tkzDrawPoints(A,B)
  \tkzDrawPoint[color=red,shape=cross out](C)
  \tkzCompass[color=orange](A,B A,C B,C C,B)
  \tkzShowLine[mediator,color=red,
    dashed,length = 2](A,B)
  \tkzShowLine[parallel = 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}
```

25.3 \tkzSetUpCompass命令：设置尺规标记样式

\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、...

25.3.1 示例



```
\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}
```

25.4 \tkzShowLine命令：显示直线尺规标记

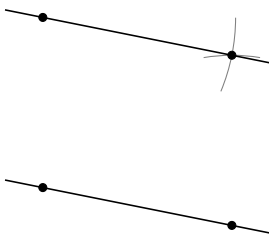
`\tkzShowLine[< 命令选项>](<pt1,pt2>) 或 (<pt1,pt2,pt3>)`

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

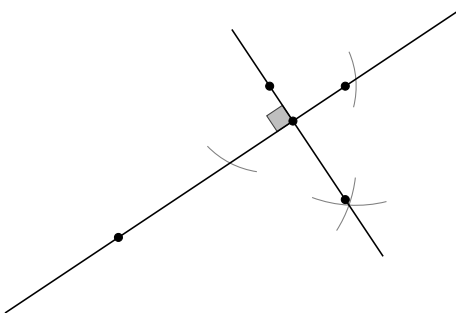
可以使用所有有效 TikZ 样式。

25.4.1 parallel选项示例



```
\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}
```

25.4.2 perpendicular选项示例

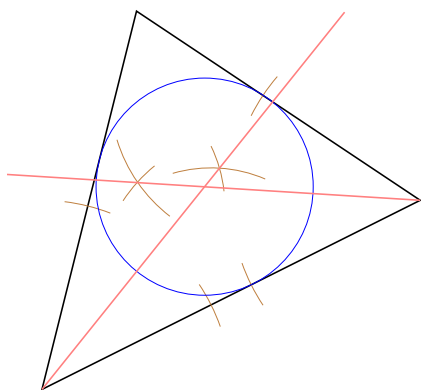


```

\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}

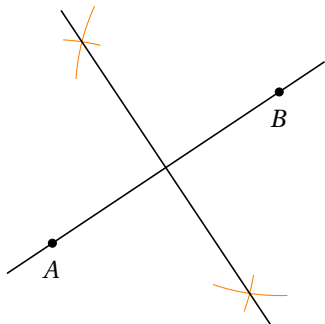
```


25.4.3 bisector选项示例



```
\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}
```

25.4.4 mediator选项示例



```
\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}
```

25.5 \tkzShowTransformation命令：显示部分变换过程尺规标记

`\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)

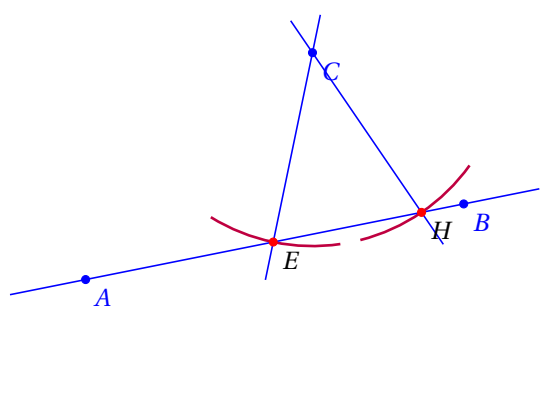
26 差分点

26.1 \tkzDefEquiPoints命令：定义一条直线上与给定点距离相等的两个点

\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	直线上两点间距离的一半
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

26.1.1 示例



```
\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}
```

27 量角器

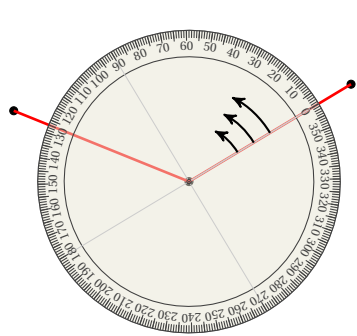
基于 **Yves Combe** 的方法，其工作原理更为简单，仅半条直线 (射线)，量角器原点位于点 O ，射线方向由 A 确定。角度方向由指定的测量圆方向决定。

27.1 `\tkzProtractor`命令：绘制量角器

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

27.2 正向圆量角器

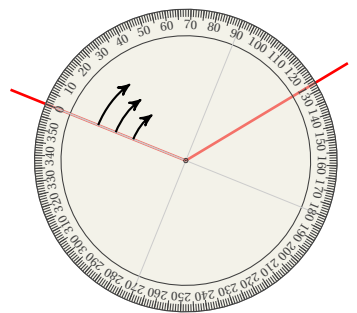
正向测量圆方向



```
\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}
```

27.3 反向圆量角器

逆向测量圆方向



```
\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}
```

28 实例分析

28.1 一些有趣的例子

28.1.1 相似等腰三角形

以下示例选自精彩的 **Descartes et les Mathématiques** 网站,在此,未对原文进行任何修改,仅用 **tkz-euclide** 宏包对其进行了绘制。

<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

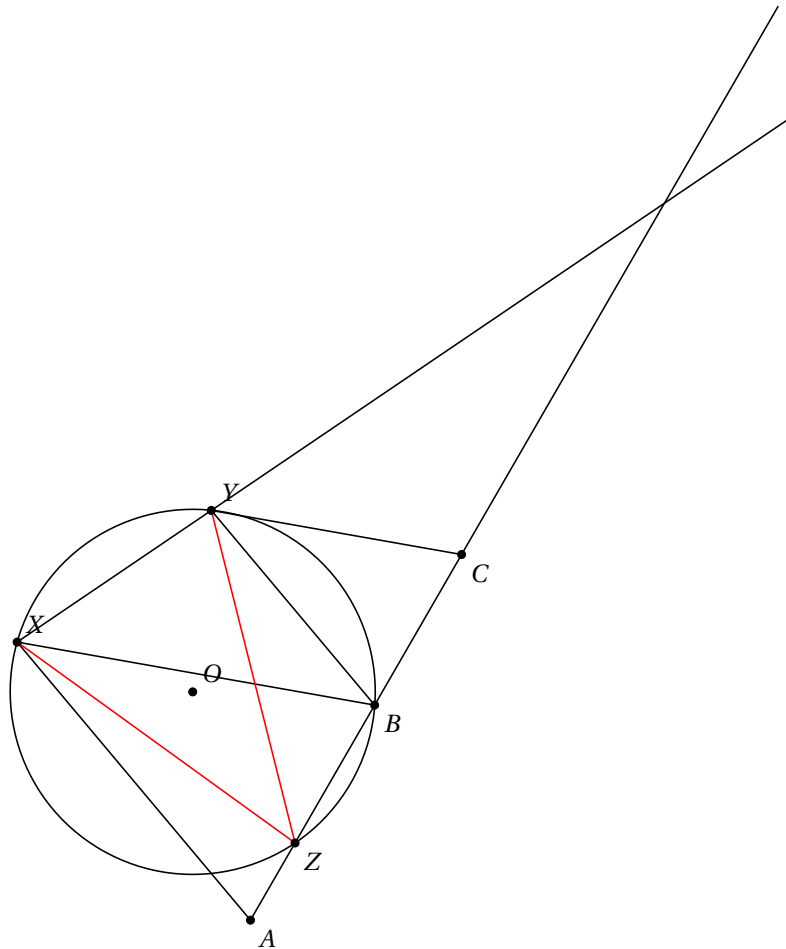
构造相似等腰三角形有两种方式,一是 *Tangente* 杂志提供的方法,另一个是 *Le Monde* 杂志提供的方法。

“*Tangente*”杂志编辑: 两个相似等腰三角形 AXB 和 BYC 的主顶点分别是 X 和 Y , 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: 两个相似等腰三角形 AXB 和 BYC 的主顶点分别是 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?

下面两页给出了解决这两个问题的 **tkz-euclide** 代码和绘图过程,但在查看代码前,建议先搜索相关文献。这些代码展示了这两种推导过程。

28.1.2 "Tangente"方法

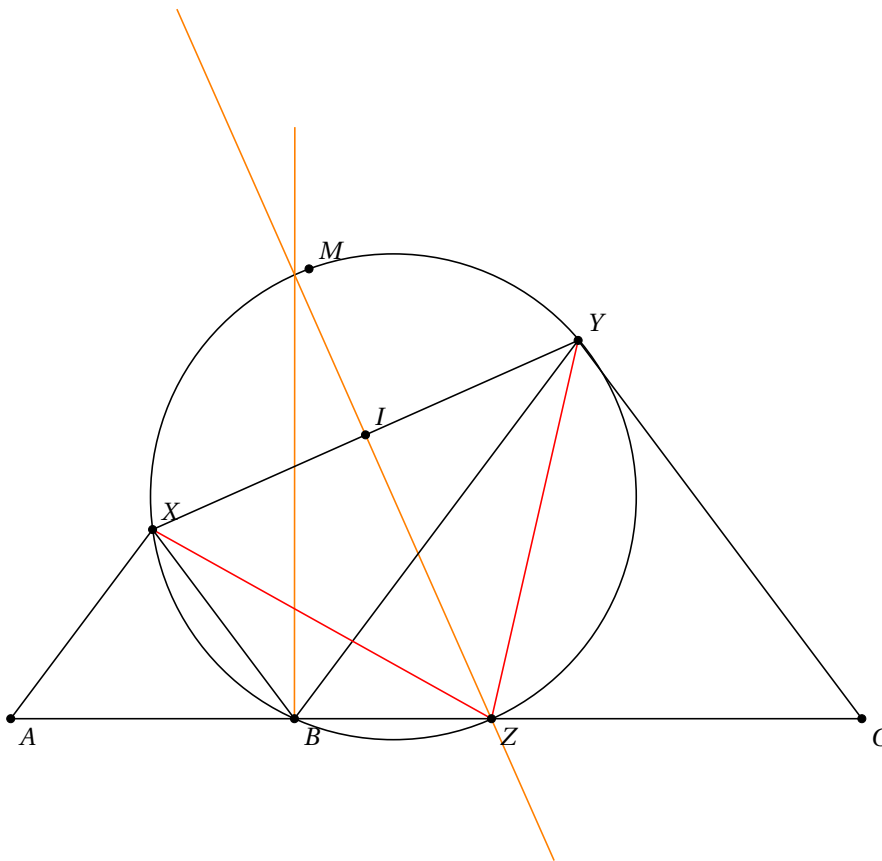


```

\begin{tikzpicture}[scale=.8,rotate=60]
  \tkzDefPoint(6,0){X}    \tkzDefPoint(3,3){Y}
  \tkzDefShiftPoint[X](-110:6){A}    \tkzDefShiftPoint[X](-70:6){B}
  \tkzDefShiftPoint[Y](-110:4.2){A'} \tkzDefShiftPoint[Y](-70:4.2){B'}
  \tkzDefPointBy[translation= from A' to B ](Y) \tkzGetPoint{Y}
  \tkzDefPointBy[translation= from A' to B ](B') \tkzGetPoint{C}
  \tkzInterLL(A,B)(X,Y) \tkzGetPoint{O}
  \tkzDefMidPoint(X,Y) \tkzGetPoint{I}
  \tkzDefPointWith[orthogonal](I,Y)
  \tkzInterLL(I,\tkzPointResult)(A,B) \tkzGetPoint{Z}
  \tkzDefCircle[circum](X,Y,B) \tkzGetPoint{O}
  \tkzDrawCircle(O,X)
  \tkzDrawLines[add = 0 and 1.5](A,C) \tkzDrawLines[add = 0 and 3](X,Y)
  \tkzDrawSegments(A,X B,X B,Y C,Y)   \tkzDrawSegments[color=red](X,Z Y,Z)
  \tkzDrawPoints(A,B,C,X,Y,O,Z)
  \tkzLabelPoints(A,B,C,Z)   \tkzLabelPoints[above right](X,Y,O)
\end{tikzpicture}

```

28.1.3 "Le Monde"方法



```

\begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(3,0){B}
  \tkzDefPoint(9,0){C}
  \tkzDefPoint(1.5,2){X}
  \tkzDefPoint(6,4){Y}
  \tkzDefCircle[circum](X,Y,B)      \tkzGetPoint{O}
  \tkzDefMidPoint(X,Y)             \tkzGetPoint{I}
  \tkzDefPointWith[orthogonal](I,Y) \tkzGetPoint{i}
  \tkzDrawLines[add = 2 and 1,color=orange](I,i)
  \tkzInterLL(I,i)(A,B)            \tkzGetPoint{Z}
  \tkzInterLC(I,i)(O,B)            \tkzGetSecondPoint{M}
  \tkzDefPointWith[orthogonal](B,Z) \tkzGetPoint{b}
  \tkzDrawCircle(O,B)
  \tkzDrawLines[add = 0 and 2,color=orange](B,b)
  \tkzDrawSegments(A,X B,X B,Y C,Y A,C X,Y)
  \tkzDrawSegments[color=red](X,Z Y,Z)
  \tkzDrawPoints(A,B,C,X,Y,Z,M,I)
  \tkzLabelPoints(A,B,C,Z)
  \tkzLabelPoints[above right](X,Y,M,I)
\end{tikzpicture}

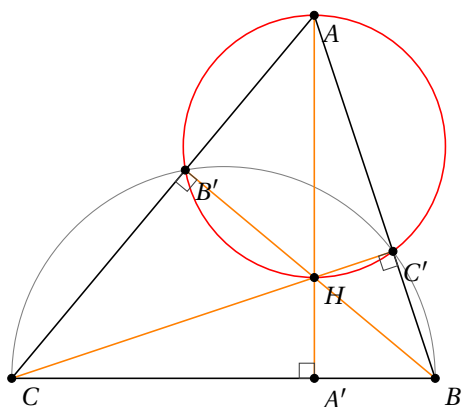
```

28.1.4 三角形的高

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

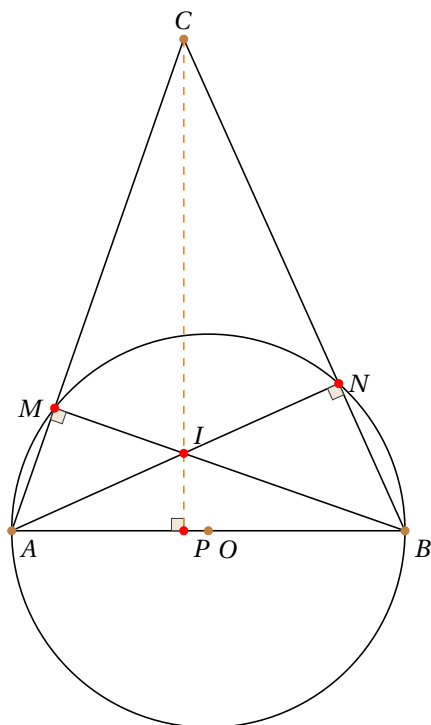
https://debart.pagesperso-orange.fr/geoplan/geometrie_triangle.html

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}
```

28.1.5 三角形的高 - 另一种构造方式

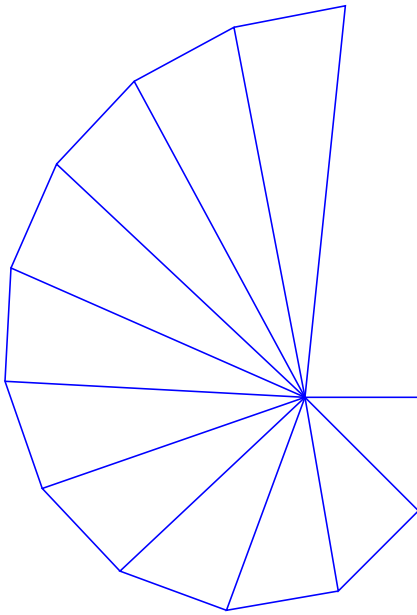


```
\begin{tikzpicture}[scale=0.65]
\tkzDefPoint(0,0){A}
\tkzDefPoint(8,0){B}
\tkzDefPoint(3.5,10){C}
\tkzDefMidPoint(A,B)
\tkzGetPoint{O}
\tkzDefPointBy[projection=onto A--B](C)
\tkzGetPoint{P}
\tkzInterLC(C,A)(O,A)
\tkzGetSecondPoint{M}
\tkzInterLC(C,B)(O,A)
\tkzGetFirstPoint{N}
\tkzInterLL(B,M)(A,N)
\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}
```


28.2 其他作者提供的实例

28.2.1 整数的算术平方根

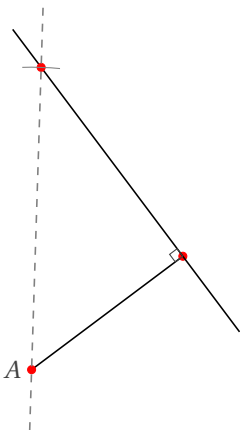
本例演示了如何用尺规求解 1 、 $\sqrt{2}$ 和 $\sqrt{3}$ 的方法。



```
\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}
```

28.2.2 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}
```

28.2.3 阿基米德等分

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)

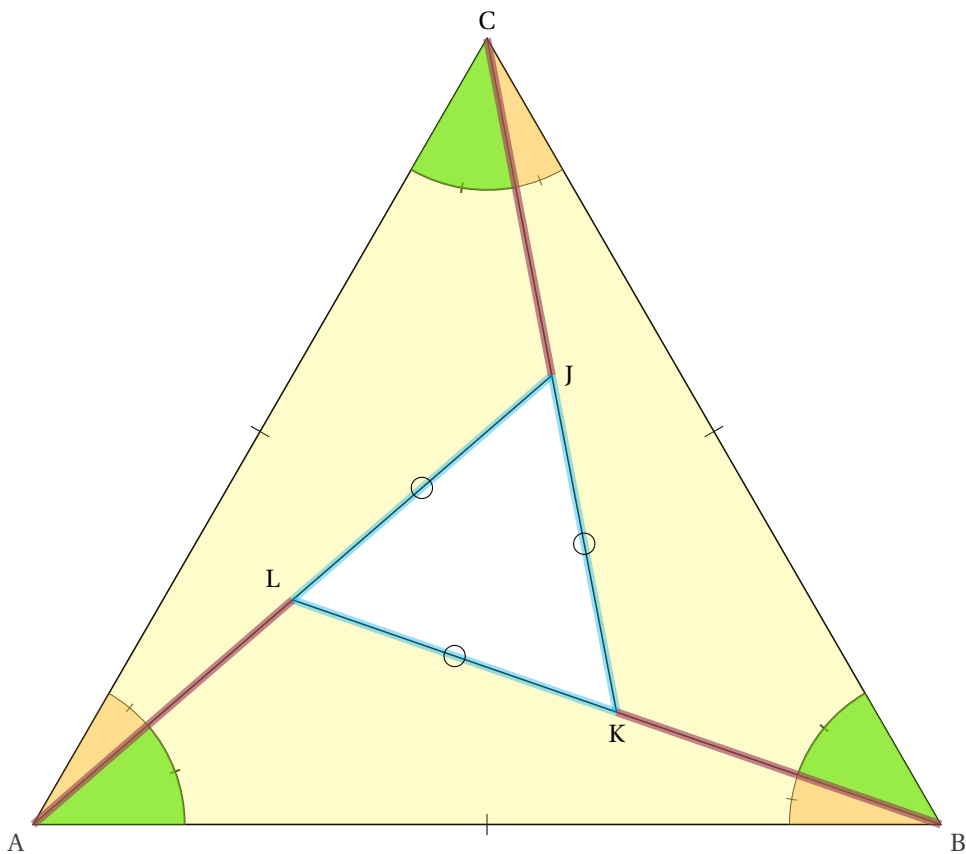

```

\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}

```

28.2.5 证明示例 1(John Kitzmiller)

证明 $\triangle LKJ$ 是等边三角形。



```

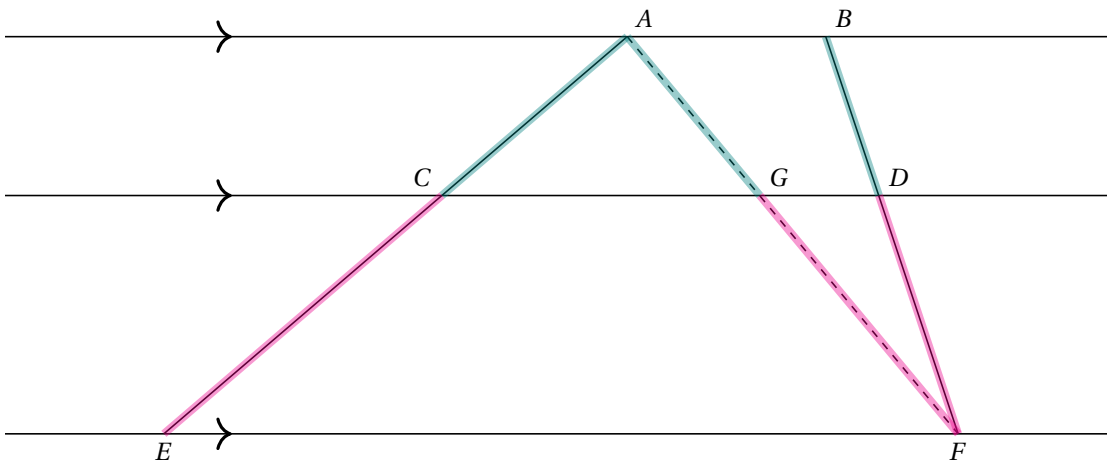
\begin{tikzpicture}[scale=2]
  \tkzDefPoint[label=below left:A](0,0){A}
  \tkzDefPoint[label=below right:B](6,0){B}
  \tkzDefTriangle[equilateral](A,B) \tkzGetPoint{C}
  \tkzMarkSegments[mark=|](A,B A,C B,C)
  \tkzDefBarycentricPoint(A=1,B=2) \tkzGetPoint{C'}
  \tkzDefBarycentricPoint(A=2,C=1) \tkzGetPoint{B'}
  \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}

```

28.2.6 证明示例 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。



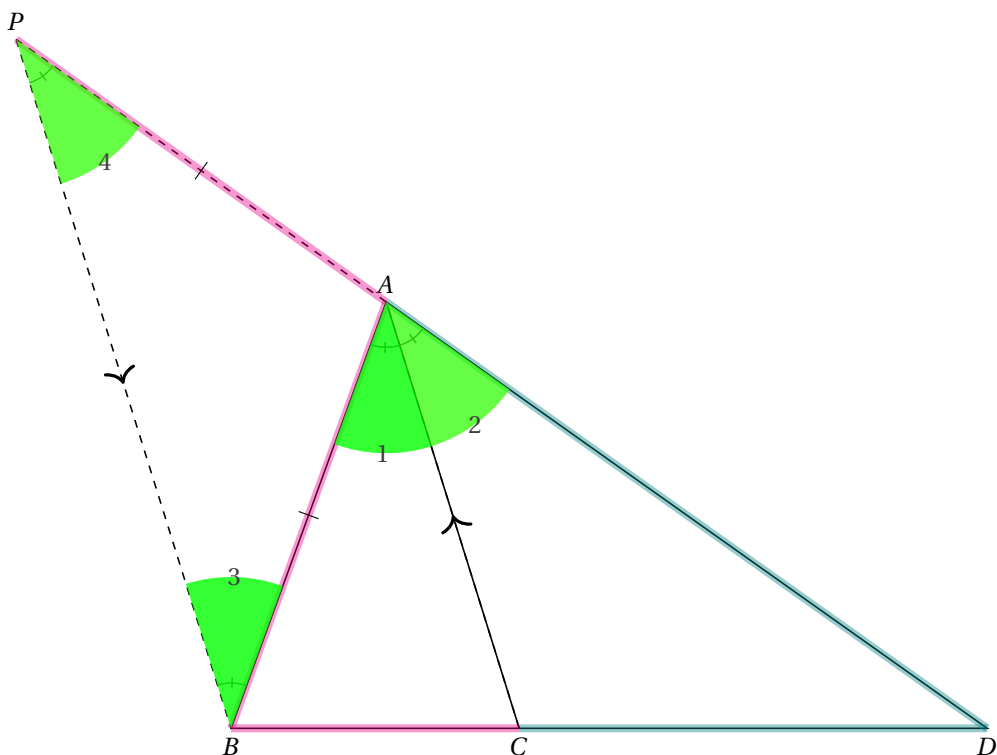
```

\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}

```

28.2.7 证明示例 3(John Kitzmiller)

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



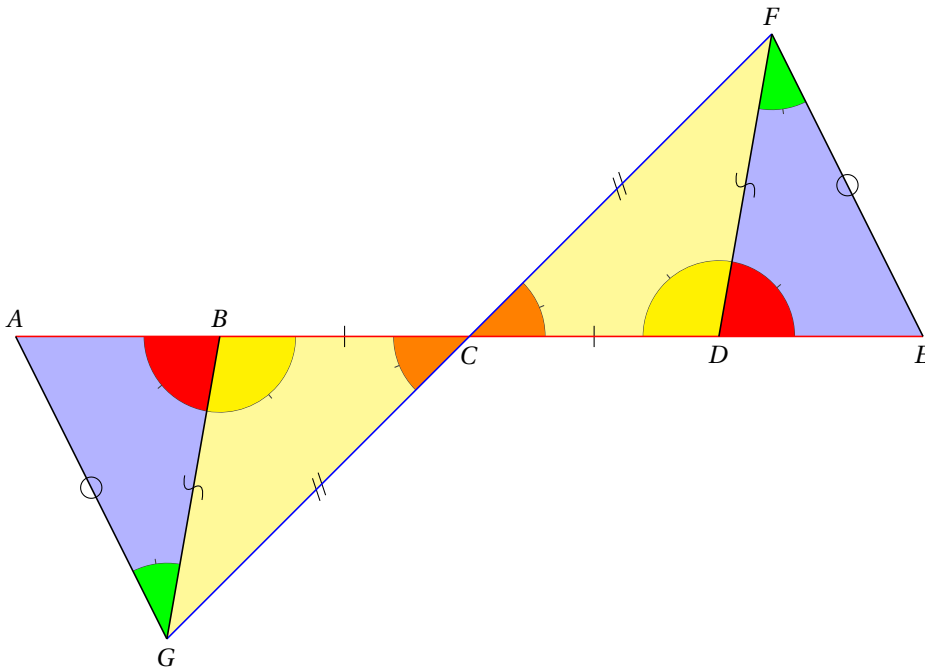
```

\begin{tikzpicture}[scale=2]
  \tkzDefPoints{0/0/B, 5/0/D} \tkzDefPoint(70:3){A}
  \tkzDrawPolygon(B,D,A)
  \tkzDefLine[bisector](B,A,D) \tkzGetPoint{a}
  \tkzInterLL(A,a)(B,D) \tkzGetPoint{C}
  \tkzDefLine[parallel=through B](A,C) \tkzGetPoint{b}
  \tkzInterLL(A,D)(B,b) \tkzGetPoint{P}
  \begin{scope}[decoration={markings, mark=at position .5 with {\arrow[scale=2]{>}}}]
    \tkzDrawSegments[postaction={decorate},dashed](C,A P,B)
  \end{scope}
  \tkzDrawSegment(A,C) \tkzDrawSegment[style=dashed](A,P)
  \tkzLabelPoints[below](B,C,D) \tkzLabelPoints[above](A,P)
  \tkzDrawSegments[color=magenta, line width=3pt, opacity=0.4](B,C P,A)
  \tkzDrawSegments[color=teal, line width=3pt, opacity=0.4](C,D A,D)
  \tkzDrawSegments[color=magenta, line width=3pt, opacity=0.4](A,B)
  \tkzMarkAngles[size=3mm](B,A,C C,A,D)
  \tkzMarkAngles[size=3mm](B,A,C A,B,P)
  \tkzMarkAngles[size=3mm](B,P,A C,A,D)
  \tkzMarkAngles[size=3mm](B,A,C A,B,P B,P,A C,A,D)
  \tkzFillAngles[fill=green, opacity=0.5](B,A,C A,B,P)
  \tkzFillAngles[fill=yellow, opacity=0.3](B,P,A C,A,D)
  \tkzFillAngles[fill=green, opacity=0.6](B,A,C A,B,P B,P,A C,A,D)
  \tkzLabelAngle[pos=1](B,A,C){1} \tkzLabelAngle[pos=1](C,A,D){2}
  \tkzLabelAngle[pos=1](A,B,P){3} \tkzLabelAngle[pos=1](B,P,A){4}
  \tkzMarkSegments[mark=|](A,B A,P)
\end{tikzpicture}

```

28.2.8 证明示例 4(John Kitzmiller)

证明: $\overline{AG} \cong \overline{EF}$ (Detour).

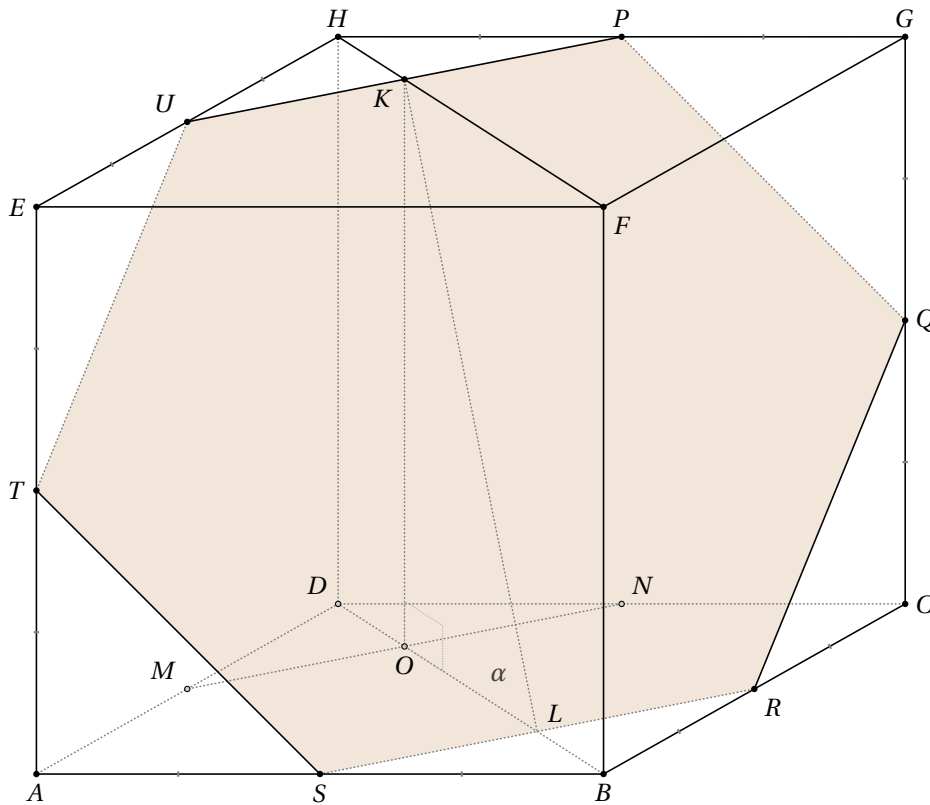



```

\begin{tikzpicture}[scale=3]
  \tkzDefPoints{0/0/A,2/0/B}
  \tkzDefSquare(A,B) \tkzGetPoints{C}{D}
  \tkzDefPointBy[rotation=center D angle 45](C)\tkzGetPoint{G}
  \tkzDefSquare(G,D)\tkzGetPoints{E}{F}
  \tkzInterLL(B,C)(E,F)\tkzGetPoint{H}
  \tkzFillPolygon[gray!10](D,E,H,C,D)
  \tkzDrawPolygon(A,...,D)\tkzDrawPolygon(D,...,G)
  \tkzDrawSegment(B,E)
  \tkzMarkSegments[mark=|,size=3pt,color=gray](A,B B,C C,D D,A E,F F,G G,D D,E)
  \tkzMarkSegments[mark=|,size=3pt,color=gray](B,E E,H)
  \tkzLabelPoints[left](A,D)
  \tkzLabelPoints[right](B,C,F,H)
  \tkzLabelPoints[above](G)\tkzLabelPoints[below](E)
  \tkzMarkRightAngles(D,A,B D,G,F)
\end{tikzpicture}

```

28.2.10 构图示例 2(Indonesia)

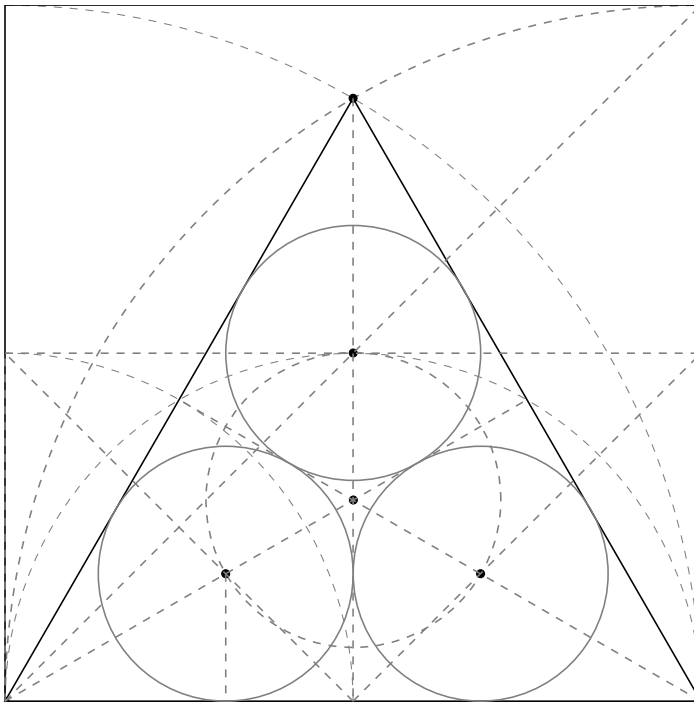



```

\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,tkzdotted,/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}

```

28.2.11 三个相切圆



```

\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}

```



```

\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}

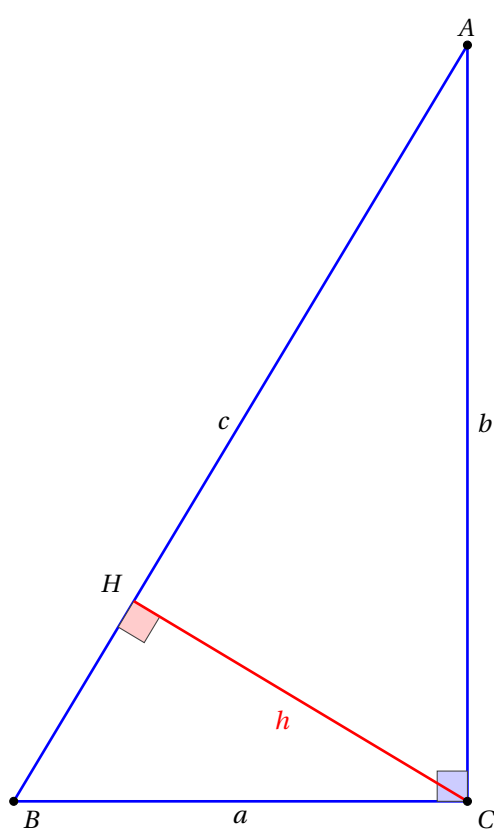
```

29 个性化设置

29.1 \tkzSetUpLine命令：设置线条样式

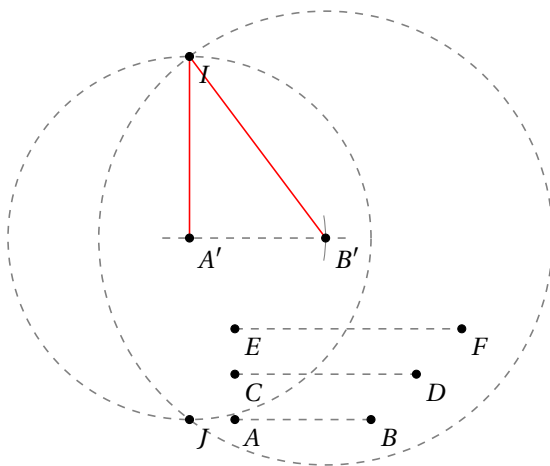
\tkzSetUpLine[< 命令选项>]		
选项	默认值	含义
color	black	颜色
line width	0.4pt	线宽
style	solid	线型
add	.2 and .2	线段两端延伸长度

29.1.1 改变线宽示例



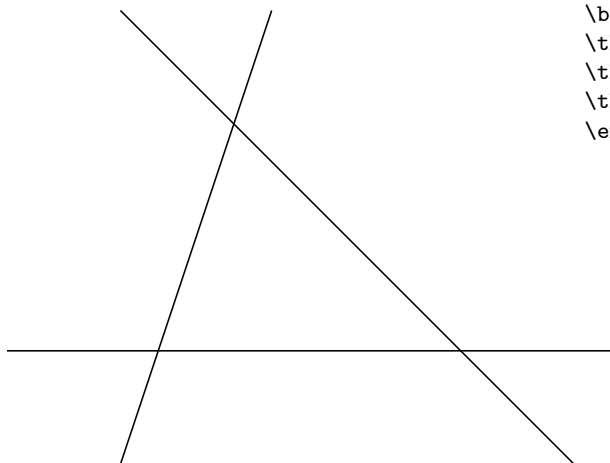
```
\begin{tikzpicture}
\tkzSetUpLine[color=blue,line width=1pt]
\begin{scope}[rotate=-90]
\tkzDefPoint(10,6){C}
\tkzDefPoint( 0,6){A}
\tkzDefPoint(10,0){B}
\tkzDefPointBy[projection = onto B--A](C)
\tkzGetPoint{H}
\tkzDrawPolygon(A,B,C)
\tkzMarkRightAngle[size=.4,fill=blue!20](B,C,A)
\tkzMarkRightAngle[size=.4,fill=red!20](B,H,C)
\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}
```

29.1.2 改变线型示例



```
\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}
```

29.1.3 线段延伸示例



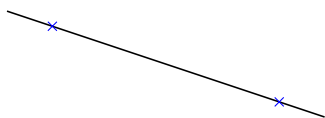
```
\begin{tikzpicture}
\tkzSetUpLine[add=.5 and .5]
\tkzDefPoints{0/0/A,4/0/B,1/3/C}
\tkzDrawLines(A,B B,C A,C)
\end{tikzpicture}
```

29.2 \tkzSetUpPoint命令：设置点样式

`\tkzSetUpPoint[< 命令选项>]`

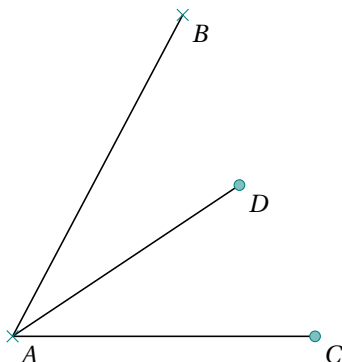
选项	默认值	含义
color	black	颜色
size	3pt	尺寸
fill	black!50	填充色
shape	circle	圆或十字线

29.2.1 示例 1



```
\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}
```

29.2.2 示例 2



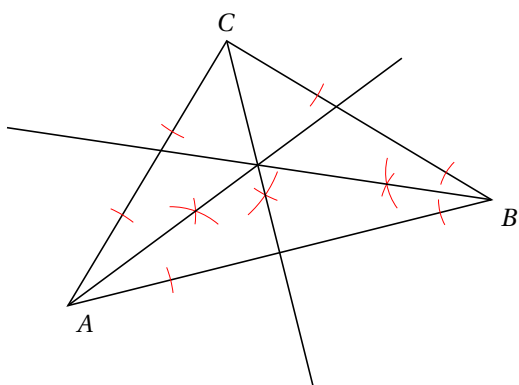
```
\begin{tikzpicture}
  \tkzInit[ymin=-0.5,ymax=3,xmin=-0.5,xmax=7]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(02.25,04.25){B}
  \tkzDefPoint(4,0){C}
  \tkzDefPoint(3,2){D}
  \tkzDrawSegments(A,B A,C A,D)
  {\tkzSetUpPoint[shape=cross out,
    fill= teal!50,
    size=4,color=teal]
  \tkzDrawPoints(A,B)}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

29.3 \tkzSetUpCompass命令：设置尺规标记样式

`\tkzSetUpCompass[< 命令选项>]`

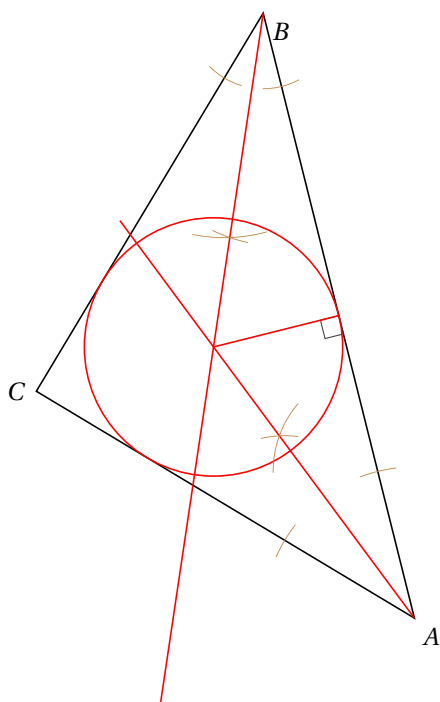
选项	默认值	含义
color	black	圆弧颜色
line width	0.4pt	圆弧线宽
style	solid	圆弧线型

29.3.1 示例 1



```
\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}
```

29.3.2 示例 2



```
\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=tkzdotted]
\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}
```

29.4 局部样式设置

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}}
\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}
```


30 tkz-base 小结

30.1 tkz-base 宏包工具

首先，无需处理 TikZ 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尺寸。为此，需要设置工作区域包围盒，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 1cm 为绘图单位。某些情况下，则需要指定画布大小 (xmin、xmax、ymin 和 ymax)，并使用“裁剪”矩形尽可能控制图形尺寸。

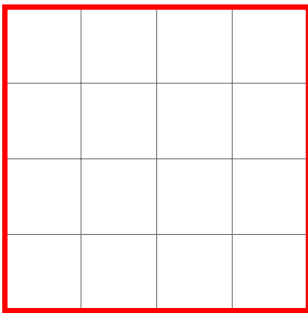
tkz-euclide 宏包使用的 **tkz-base** 宏包提供的两个命令是：

- `\tkzInit`
- `\tkzClip`

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

30.2 \tkzInit 命令和 \tkzShowBB 命令

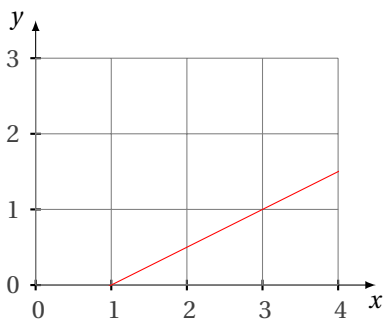
用图形四周的矩形表示包围盒。



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

30.3 \tkzClip 命令

通过对初始绘图矩形的“裁剪”，仅显示指定矩形范围的内容。



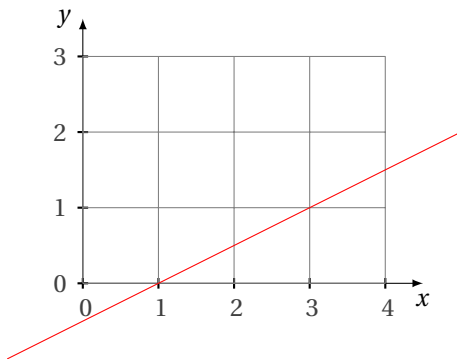
```
\begin{tikzpicture}
  \tkzInit[xmax=4,ymax=3]
  \tkzAxeXY
  \tkzGrid
  \tkzClip
  \draw[red] (-1,-1)--(5,2);
\end{tikzpicture}
```

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]
```

30.3.1 \tkzClip 命令和 space 选项示例

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`。

31 FAQ

31.1 常见错误

目前, 根据需要, 已对语法进行了多次改进, 这可能会带来很多错误, 主要改进有:

- 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}` 命令分别得到两个点。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 `R` 命令选项。
- `\tkzDrawSegments[color = gray,style=dashed]{B,B' C,C'}` 是错误语法, 只有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` 库的语法, 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, 可以使用 `\tkzClip` 命令进行图形裁剪。
- 如果角度太小时, 使用 `\tkzDrawAngle` 命令, 则会发生错误。该错误是使用了在圆弧上放置标记的 `decoration` 装饰库而产生的, 即使没有标记, 该错误仍然存在。可以使用 `mkpos=.2` 选项避免该错误, 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` 命令避免该错误发生。

Index

- `\ang`, 108
- `\Ax`, 125
- `\Ay`, 125

- `\coordinate`, 24

- `\dAB`, 123
- `\Delta`, 60
- `\draw (A)--(B)`; 命令, 61

- Environment
 - scope, 26

- `\fpeval`, 96

- `\len`, 124

- `\newdimen`, 97

- Operating System
 - Windows, 20

- Package
 - fp, 20, 22
 - numprint, 20
 - pgfmath, 26, 155
 - tikz 3.00, 20
 - tkz-base, 20, 23, 153
 - tkz-euclide, 20, 133, 153
 - tzk-base, 153
 - xfp, 20, 22, 24–26, 96, 123, 155
- `\pgflinewidth`, 35, 37
- `\pgfmathsetmacro`, 96
- `\px`, 125
- `\py`, 125

- `\slope`, 110
- standalone, 18

- TeX Distributions
 - MiKTeX, 20
 - TeXLive, 20
- TikZ Library
 - angles, 22
 - babel, 14
 - decoration, 155
 - quotes, 22
- `\tkzAngleResult`, 108, 111
- `\tkzCalcLength`, 123
- `\tkzCalcLength:` arguments
 - (pt1,pt2){宏名称}, 123
- `\tkzCalcLength:` options
 - cm, 123
- `\tkzCalcLength[<命令选项>](<pt1,pt2>){<宏名称>}`, 123
- `\tkzCentroid`, 29
- `\tkzClip`, 14, 22, 153, 155
- `\tkzClipBB`, 22
- `\tkzClipCircle`, 83, 88, 92
- `\tkzClipCircle:` arguments
 - (<A,B>) or (<A,r>), 92
- `\tkzClipCircle:` options
 - R, 92
 - radius, 92

`\tkzClipCircle`[<命令选项>]($\langle A, B \rangle$) 或 ($\langle A, r \rangle$), 92
`\tkzClipPolygon`, 80
`\tkzClipPolygon: arguments`
 ($\langle pt1, pt2 \rangle$), 80
`\tkzClipPolygon`[<命令选项>](\langle 点集列表 \rangle), 80
`\tkzClipSector`(0,A)(B), 116
`\tkzClipSector[R]`(0,2 cm)(30,90), 116
`\tkzClipSector[rotate]`(0,A)(90), 116
`\tkzClipSector`, 116
`\tkzClipSector: options`
 R, 116
 rotate, 116
 towards, 116
`\tkzClipSector`[<命令选项>]($\langle 0, \dots \rangle$)($\langle \dots \rangle$), 116
`\tkzcmtopt`, 124
`\tkzcmtopt: arguments`
 (nombre){宏名称}, 124
`\tkzcmtopt`($\langle nombre \rangle$){ \langle 宏名称 \rangle }, 124
`\tkzCompass`, 120, 126
`\tkzCompass: options`
 delta, 126
 length, 126
`\tkzCompasss`, 126
`\tkzCompasss: options`
 delta, 126
 length, 126
`\tkzCompasss`[<命令选项>]($\langle pt1, pt2, pt3, pt4, \dots \rangle$), 126
`\tkzCompass`[<命令选项>]($\langle A, B \rangle$), 126
`\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), 125
`\tkzDefCircle`, 83
`\tkzDefCircle: arguments`
 ($\langle pt1, pt2 \rangle$) or ($\langle pt1, pt2, pt3 \rangle$), 83
`\tkzDefCircle: options`
 K, 83
 apollonius, 83
 circum, 83
 diameter, 83
 euler or nine, 83
 ex, 83
 in, 83
 orthogonal through, 83
 orthogonal, 83
 spieker, 83
 through, 83
`\tkzDefCircle`[<命令选项>]($\langle A, B \rangle$) or ($\langle A, B, C \rangle$), 83
`\tkzDefEquiPoints`, 131
`\tkzDefEquiPoints: arguments`
 (pt1,pt2), 131
`\tkzDefEquiPoints: options`
 /compass/delta, 131
 dist, 131
 from=pt, 131
 show, 131
`\tkzDefEquiPoints`[<命令选项>]($\langle pt1, pt2 \rangle$), 131
`\tkzDefGoldRectangle`, 78
`\tkzDefGoldRectangle: arguments`
 ($\langle pt1, pt2 \rangle$), 78
`\tkzDefGoldRectangle`($\langle point, point \rangle$), 78
`\tkzDefLine`, 54
`\tkzDefLine: arguments`

- ($\langle \text{pt1}, \text{pt2}, \text{pt3} \rangle$), 54
- ($\langle \text{pt1}, \text{pt2} \rangle$), 54
- \tkzDefLine: options
 - K, 54
 - bisector out, 54
 - bisector, 54
 - mediator, 54
 - normed, 54
 - orthogonal=through..., 54
 - parallel=through..., 54
 - perpendicular=through..., 54
- \tkzDefLine[\langle 命令选项 \rangle]($\langle \text{pt1}, \text{pt2} \rangle$) or ($\langle \text{pt1}, \text{pt2}, \text{pt3} \rangle$), 54
- \tkzDefMidPoint, 28
- \tkzDefMidPoint: arguments
 - (pt1,pt2), 28
- \tkzDefMidPoint($\langle \text{pt1}, \text{pt2} \rangle$), 28
- \tkzDefParallelogram, 76
- \tkzDefParallelogram: arguments
 - ($\langle \text{pt1}, \text{pt2}, \text{pt3} \rangle$), 76
- \tkzDefParallelogram($\langle \text{pt1}, \text{pt2}, \text{pt3} \rangle$), 76
- \tkzDefPoint, 24, 26, 28, 94
- \tkzDefPoint: arguments
 - ($\alpha:d$), 24
 - (x,y), 24
 - {名称}, 24
- \tkzDefPoint: options
 - label, 24
 - shift, 24
- \tkzDefPointBy, 39
- \tkzDefPointBy: arguments
 - pt, 39
- \tkzDefPointBy: options
 - homothety, 39
 - inversion, 39
 - projection , 39
 - reflection, 39
 - rotation in rad, 39
 - rotation , 39
 - symmetry , 39
 - translation, 39
- \tkzDefPointBy[\langle 命令选项 \rangle]($\langle \text{pt} \rangle$), 39
- \tkzDefPointOnCircle, 38
- \tkzDefPointOnCircle: options
 - angle, 38
 - center, 38
 - radius, 38
- \tkzDefPointOnCircle[\langle 命令选项 \rangle], 38
- \tkzDefPointOnLine, 37
- \tkzDefPointOnLine: arguments
 - pt1,pt2, 37
- \tkzDefPointOnLine: options
 - pos=nb, 37
- \tkzDefPointOnLine[\langle 命令选项 \rangle]($\langle A,B \rangle$), 37
- \tkzDefPoints{0/0/0,2/2/A}, 27
- \tkzDefPoints, 27, 28
- \tkzDefPoints: arguments
 - $x_i/y_i/n_i$, 27
- \tkzDefPoints: options
 - shift, 27
- \tkzDefPointsBy, 39, 43, 44
- \tkzDefPointsBy: arguments
 - (\langle 变换点列表 \rangle){ \langle 变换结果点名称列表 \rangle }, 44
- \tkzDefPointsBy: options
 - homothety = center #1 ratio #2, 44

```

projection = onto #1--#2, 44
reflection = over #1--#2, 44
rotation = center #1 angle #2, 44
rotation in rad = center #1 angle #2, 44
symmetry = center #1, 44
translation = from #1 to #2, 44
\tkzDefPointsBy[(命令选项)]((变换点列表)){(变换结果点名称列表)}, 44
\tkzDefPoints[(命令选项)]{(x1/y1/n1, x2/y2/n2, ...)}, 27
\tkzDefPointWith, 45
\tkzDefPointWith: arguments
    (pt1, pt2), 45
\tkzDefPointWith: options
    K, 45
    colinear normed= at #1, 45
    colinear= at #1, 45
    linear normed, 45
    linear, 45
    orthogonal normed, 45
    orthogonal, 45
\tkzDefPointWith((pt1, pt2)), 45
\tkzDefPoint[(命令选项)]((x, y)){(名称)} or ((α:d)){(名称)}, 24
\tkzDefRandPointOn, 22, 50
\tkzDefRandPointOn: options
    circle =center pt1 radius dim, 50
    circle through=center pt1 through pt2, 50
    disk through=center pt1 through pt2, 50
    line=pt1--pt2, 50
    rectangle=pt1 and pt2, 50
    segment= pt1--pt2, 50
\tkzDefRandPointOn[(命令选项)], 50
\tkzDefRegPolygon, 82
\tkzDefRegPolygon: arguments
    ((pt1, pt2)), 82
\tkzDefRegPolygon: options
    TikZ 选项, 82
    center, 82
    name, 82
    sides, 82
    side, 82
\tkzDefRegPolygon[(命令选项)]((pt1, pt2)), 82
\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, 70
\tkzDefSpcTriangle: options
    centroid or medial, 70
    euler, 70
    ex or excentral, 70
    extouch, 70
    feuerbach, 70
    in or incentral, 70
    intouch or contact, 70
    name, 70
    orthic, 70
    tangential, 70
\tkzDefSpcTriangle[(命令选项)]((A, B, C)), 70
\tkzDefSquare, 75
\tkzDefSquare: arguments
    ((pt1, pt2)), 75
\tkzDefSquare((pt1, pt2)), 75

```

```

\tkzDefTangent, 56
\tkzDefTangent: arguments
    (\(pt1,pt2 or (\(pt1,dim\))) , 56
\tkzDefTangent: options
    at=pt, 56
    from with R=pt, 56
    from=pt, 56
\tkzDefTangent[\(命令选项\)](\(pt1,pt2\)) or (\(pt1,dim\)), 56
\tkzDefTriangle, 67
\tkzDefTriangle: options
    cheops, 67
    equilateral, 67
    euclide, 67
    golden, 67
    gold, 67
    pythagore, 67
    school, 67
    two angles= #1 and #2, 67
\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[\(命令选项\)](\(A,B,C\)), 31
\tkzDefTriangle[\(命令选项\)](\(A,B\)), 67
\tkzDrawAngle, 155
\tkzDrawArc[angles](O,A)(O,90), 118
\tkzDrawArc[delta=10](O,A)(B), 118
\tkzDrawArc[R with nodes](O,2 cm)(A,B), 118
\tkzDrawArc[R](O,2 cm)(30,90), 118
\tkzDrawArc[rotate,color=red](O,A)(90), 118
\tkzDrawArc, 118
\tkzDrawArc: options
    R with nodes, 118
    R, 118
    angles, 118
    delta, 118
    rotate, 118
    towards, 118
\tkzDrawArc[\(命令选项\)](\(O,...\))(\(,...\)), 118
\tkzDrawCircle, 83, 88
\tkzDrawCircle: arguments
    (\(pt1,pt2\)), 88
\tkzDrawCircle: options
    R, 88
    diameter, 88
    through, 88
\tkzDrawCircles, 89
\tkzDrawCircles: arguments
    (\(pt1,pt2 pt3,pt4,...\)), 89
\tkzDrawCircles: options
    R, 89
    diameter, 89
    through, 89

```


`\tkzDrawCircles`[<命令选项>]($(A,B\ C,D)$), 89
`\tkzDrawCircle`[<命令选项>]((A,B)), 88
`\tkzDrawGoldRectangle`, 78
`\tkzDrawGoldRectangle: arguments`
 ($\langle pt1,pt2 \rangle$), 78
`\tkzDrawGoldRectangle: options`
 TikZ 选项, 78
`\tkzDrawGoldRectangle`[<命令选项>]($\langle point,point \rangle$), 78
`\tkzDrawLine`, 58
`\tkzDrawLine: options`
 add= nb1 and nb2, 58
 altitude, 58
 bisector, 58
 median, 58
 none, 58
`\tkzDrawLines`, 59
`\tkzDrawLines`[<命令选项>]($\langle pt1,pt2\ pt3,pt4,... \rangle$), 59
`\tkzDrawLine`[<命令选项>]($\langle pt1,pt2 \rangle$) or ($\langle pt1,pt2,pt3 \rangle$), 58
`\tkzDrawMedian`, 58
`\tkzDrawPoint`(A,B), 155
`\tkzDrawPoint`, 35
`\tkzDrawPoint: arguments`
 点的名称, 35
`\tkzDrawPoint: options`
 color, 35
 shape, 35
 size, 35
`\tkzDrawPoints`(A,B,C), 37
`\tkzDrawPoints`, 36, 37, 155
`\tkzDrawPoints: arguments`
 点列表, 37
`\tkzDrawPoints: options`
 color, 37
 shape, 37
 size, 37
`\tkzDrawPoints`[<命令选项>]($\langle \text{点列表} \rangle$), 37
`\tkzDrawPoint`[<命令选项>]($\langle \text{名称} \rangle$), 35
`\tkzDrawPolygon`, 79
`\tkzDrawPolygon: arguments`
 ($\langle pt1,pt2,pt3,... \rangle$), 79
`\tkzDrawPolygon: options`
 TikZ 选项, 79
`\tkzDrawPolygon`[<命令选项>]($\langle \text{点集列表} \rangle$), 79
`\tkzDrawPolySeg`, 79
`\tkzDrawPolySeg: arguments`
 ($\langle pt1,pt2,pt3,... \rangle$), 79
`\tkzDrawPolySeg: options`
 TikZ 选项, 79
`\tkzDrawPolySeg`[<命令选项>]($\langle \text{点集列表} \rangle$), 79
`\tkzDrawSector`(O,A)(B), 113
`\tkzDrawSector`[R with nodes]($O,2\ cm$)(A,B), 113
`\tkzDrawSector`[R,color=blue]($O,2\ cm$)($30,90$), 113
`\tkzDrawSector`[rotate,color=red](O,A)(90), 113
`\tkzDrawSector`, 113
`\tkzDrawSector: options`
 R with nodes, 113
 R, 113
 rotate, 113
 towards, 113
`\tkzDrawSector`[<命令选项>]($\langle O,... \rangle$)($\langle ... \rangle$), 113
`\tkzDrawSegment`($A,B\ A,C$), 155
`\tkzDrawSegment`, 22, 61
`\tkzDrawSegment: arguments`
 ($pt1,pt2$), 61

```

\tkzDrawSegment: options
  TikZ 选项, 61
  ..., 61
  add, 61
  dim, 61
\tkzDrawSegments[color = gray, style=dashed]{B,B' C,C'}, 155
\tkzDrawSegments, 63, 155
\tkzDrawSegments[⟨命令选项⟩](⟨pt1,pt2 pt3,pt4,...⟩), 63
\tkzDrawSegment[⟨命令选项⟩](⟨pt1,pt2⟩), 61
\tkzDrawSemiCircle, 91
\tkzDrawSemiCircle: arguments
  (⟨pt1,pt2⟩), 91
\tkzDrawSemiCircle: options
  diameter, 91
  through, 91
\tkzDrawSemiCircle[⟨命令选项⟩](⟨A,B⟩), 91
\tkzDrawSquare, 77
\tkzDrawSquare: arguments
  (⟨pt1,pt2⟩), 77
\tkzDrawSquare: options
  TikZ 选项, 77
\tkzDrawSquare[⟨命令选项⟩](⟨pt1,pt2⟩), 77
\tkzDrawTriangle, 68
\tkzDrawTriangle: options
  cheops, 68
  equilateral, 68
  euclide, 68
  golden, 68
  gold, 68
  pythagore, 68
  school, 68
  two angles= #1 and #2, 68
\tkzDrawTriangle[⟨命令选项⟩](⟨A,B⟩), 68
\tkzDuplicateLen, 122
\tkzDuplicateLength, 122
\tkzDuplicateSegment, 122
\tkzDuplicateSegment: arguments
  (pt1,pt2)(pt3,pt4), 122
\tkzDuplicateSegment(⟨pt1,pt2⟩)(⟨pt3,pt4⟩), 122
\tkzFillAngle, 101, 138, 155
\tkzFillAngle: options
  size, 101
\tkzFillAngles, 102
\tkzFillAngles[⟨命令选项⟩](⟨A,O,B⟩)(⟨A',O',B'⟩) 等, 102
\tkzFillAngle[⟨命令选项⟩](⟨A,O,B⟩), 101
\tkzFillCircle, 83, 88, 91
\tkzFillCircle: options
  R, 91
  radius, 91
\tkzFillCircle[⟨命令选项⟩](⟨A,B⟩), 91
\tkzFillPolygon, 81
\tkzFillPolygon: arguments
  (⟨pt1,pt2,...⟩), 81
\tkzFillPolygon[⟨命令选项⟩](⟨点集列表⟩), 81
\tkzFillSector(O,A)(B), 115
\tkzFillSector[R with nodes](O,2 cm)(A,B), 115
\tkzFillSector[R,color=blue](O,2 cm)(30,90), 115
\tkzFillSector[rotate,color=red](O,A)(90), 115
\tkzFillSector, 115
\tkzFillSector: options
  R with nodes, 115
  R, 115
  rotate, 115
  towards, 115

```

`\tkzFillSector[⟨命令选项⟩](⟨O,...⟩)(⟨...⟩)`, 115
`\tkzFindAngle`, 108
`\tkzFindAngle: arguments`
 $(pt1, pt2, pt3)$, 108
`\tkzFindAngle(⟨pt1, pt2, pt3⟩)`, 108
`\tkzFindSlope`, 110
`\tkzFindSlope: arguments`
 $(pt1, pt2)pt3$, 110
`\tkzFindSlopeAngle`, 111
`\tkzFindSlopeAngle: arguments`
 $(pt1, pt2)$, 111
`\tkzFindSlopeAngle(⟨A, B⟩)`, 111
`\tkzFindSlope(⟨pt1, pt2⟩){⟨宏名称⟩}`, 110
`\tkzFirstPointResult`, 75, 78, 155
`\tkzGetAngle`, 108, 111
`\tkzGetAngle: arguments`
 宏名称, 108
`\tkzGetAngle(⟨宏名称⟩)`, 108
`\tkzGetFirstPoint{A}`, 155
`\tkzGetFirstPoint{I}`, 84
`\tkzGetFirstPoint{Jb}`, 86
`\tkzGetFirstPoint`, 75
`\tkzGetLength`, 83, 125
`\tkzGetPoint(A)`, 155
`\tkzGetPoint{A}`, 155
`\tkzGetPoint{C}`, 45
`\tkzGetPoint{M}`, 39
`\tkzGetPoint`, 22, 28, 31, 45, 50, 54, 67, 70, 76, 83
`\tkzGetPointCoord`, 125
`\tkzGetPointCoord: arguments`
 $(point){\langle \text{宏名称} \rangle}$, 125
`\tkzGetPointCoord(⟨A⟩){⟨宏名称⟩}`, 125
`\tkzGetPoints{A}{B}`, 155
`\tkzGetPoints{C}{D}`, 77
`\tkzGetPoints`, 54, 75, 78
`\tkzGetRandPointOn`, 22, 50
`\tkzGetSecondPoint{A}`, 155
`\tkzGetSecondPoint{Ib}`, 84
`\tkzGetSecondPoint{Tb}`, 86
`\tkzGetSecondPoint`, 75
`\tkzGetVectxy`, 48, 49
`\tkzGetVectxy: arguments`
 $(point){\text{name of macro}}$, 48
`\tkzGetVectxy(⟨A, B⟩){⟨text⟩}`, 48
`\tkzInit`, 14, 22, 23, 153
`\tkzInterCC`, 98
`\tkzInterCC: options`
 N, 98
 R, 98
 with nodes, 98
`\tkzInterCCN`, 98
`\tkzInterCCR`, 98
`\tkzInterCC[⟨命令选项⟩](⟨O, A⟩)(⟨O', A'⟩) 或 (⟨O, r⟩)(⟨O', r'⟩) 或 (⟨O, A, B⟩)(⟨O', C, D⟩)`, 98
`\tkzInterLC`, 94
`\tkzInterLC: options`
 N, 94
 R, 94
 with nodes, 94
`\tkzInterLC[⟨命令选项⟩](⟨A, B⟩)(⟨O, C⟩) 或 (⟨O, r⟩) 或 (⟨O, C, D⟩)`, 94
`\tkzInterLL`, 94
`\tkzInterLL(⟨A, B⟩)(⟨C, D⟩)`, 94
`\tkzLabeAngles`, 106
`\tkzLabelAngle`, 105
`\tkzLabelAngle: options`

- pos, 105
- \tkzLabelAngles, 106
- \tkzLabelAngles[⟨命令选项⟩](⟨A,O,B⟩)(⟨A',O',B'⟩) 等, 106
- \tkzLabelAngle[⟨命令选项⟩](⟨A,O,B⟩), 105
- \tkzLabelCircle, 83, 88, 93
- \tkzLabelCircle: options
 - R, 93
 - radius, 93
- \tkzLabelCircle[⟨命令选项⟩](⟨A,B⟩)(⟨角度⟩){⟨标注⟩}, 93
- \tkzLabelLine(A,B), 60
- \tkzLabelLine, 22, 60, 61
- \tkzLabelLine: arguments
 - label, 60
- \tkzLabelLine: options
 - pos, 60
- \tkzLabelLine[⟨命令选项⟩](⟨pt1,pt2⟩){⟨label⟩}, 60
- \tkzLabelSegment(A,B){5}, 65
- \tkzLabelSegment, 65
- \tkzLabelSegment: arguments
 - (pt1,pt2), 65
 - label, 65
- \tkzLabelSegment: options
 - pos, 65
- \tkzLabelSegments, 66
- \tkzLabelSegments[⟨命令选项⟩](⟨pt1,pt2 pt3,pt4,...⟩), 66
- \tkzLabelSegment[⟨命令选项⟩](⟨pt1,pt2⟩){⟨label⟩}, 65
- \tkzLength, 97
- \tkzLengthResult, 83
- \tkzMarkAngle, 104, 138
- \tkzMarkAngle: options
 - arc, 104
 - mark, 104
 - mkcolor, 104
 - mkpos, 104
 - mksize, 104
 - size, 104
- \tkzMarkAngles, 105
- \tkzMarkAngles[⟨命令选项⟩](⟨A,O,B⟩)(⟨A',O',B'⟩) 等, 105
- \tkzMarkAngle[⟨命令选项⟩](⟨A,O,B⟩), 104
- \tkzMarkRightAngle, 106
- \tkzMarkRightAngle: options
 - german, 106
 - size, 106
- \tkzMarkRightAngles, 108
- \tkzMarkRightAngles[⟨命令选项⟩](⟨A,O,B⟩)(⟨A',O',B'⟩) 等, 108
- \tkzMarkRightAngle[⟨命令选项⟩](⟨A,O,B⟩), 106
- \tkzMarkSegment, 63
- \tkzMarkSegment: options
 - color, 63
 - mark, 63
 - pos, 63
 - size, 63
- \tkzMarkSegments, 64
- \tkzMarkSegments[⟨命令选项⟩](⟨pt1,pt2 pt3,pt4,...⟩), 64
- \tkzMarkSegment[⟨命令选项⟩](⟨pt1,pt2⟩), 63
- \tkzPointResult, 28, 31, 39, 45, 50, 56, 70, 76, 83, 94, 155
- \tkzProtractor, 132
- \tkzProtractor: options
 - lw, 132
 - return, 132
 - scale, 132
- \tkzProtractor[⟨命令选项⟩](⟨O,A⟩), 132
- \tkzptocm, 124
- \tkzpttocm, 124

- \tkzpttocm: arguments
 - (number) 宏名称, 124
- \tkzpttocm([nombre](#)){([宏名称](#))}, 124
- \tkzSaveBB, 22
- \tkzSecondPointResult, 75, 78, 155
- \tkzSetUpCompass, 127, 151
- \tkzSetUpCompass: options
 - color, 127, 151
 - line width, 127, 151
 - style, 127, 151
- \tkzSetUpCompass[[命令选项](#)], 127, 151
- \tkzSetUpLine, 149
- \tkzSetUpLine: options
 - add, 149
 - color, 149
 - line width, 149
 - style, 149
- \tkzSetUpLine[[命令选项](#)], 149
- \tkzSetUpPoint, 150
- \tkzSetUpPoint: options
 - color, 150
 - fill, 150
 - shape, 150
 - size, 150
- \tkzSetUpPoint[[命令选项](#)], 150
- \tkzShowBB, 153
- \tkzShowLine, 128
- \tkzShowLine: options
 - K, 128
 - bisector, 128
 - gap, 128
 - length, 128
 - mediator, 128
 - orthogonal, 128
 - perpendicular, 128
 - ratio, 128
 - size, 128
- \tkzShowLine[[命令选项](#)]([pt1,pt2](#)) 或 ([pt1,pt2,pt3](#)), 128
- \tkzShowTransformation, 129
- \tkzShowTransformation: options
 - K, 129
 - gap, 129
 - length, 129
 - projection=onto pt1--pt2, 129
 - ratio, 129
 - reflection= over pt1--pt2, 129
 - size, 129
 - symmetry=center pt, 129
 - translation=from pt1 to pt2, 129
- \tkzShowTransformation[[命令选项](#)]([pt1,pt2](#)) 或 ([pt1,pt2,pt3](#)), 129
- \tkzTangent, 56
- \usetkzobj{all}, 22
- \usetkztool, 22
- \Vx, 48
- \Vy, 48