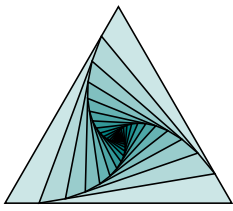


1 概述



```
\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。

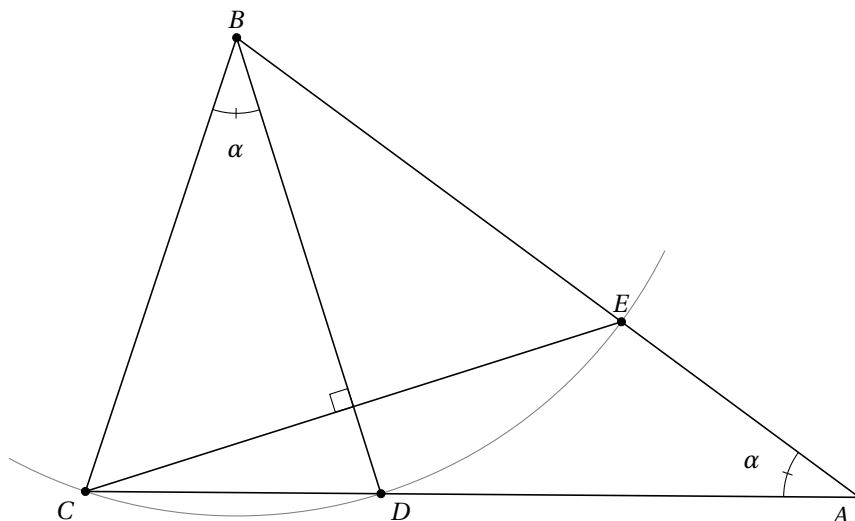
1.2 tkz-euclide与TikZ

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

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

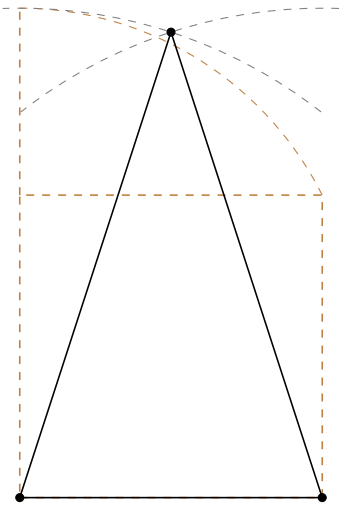
4. 从而可得

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

所以， CBD 是一个“黄金”三角形。

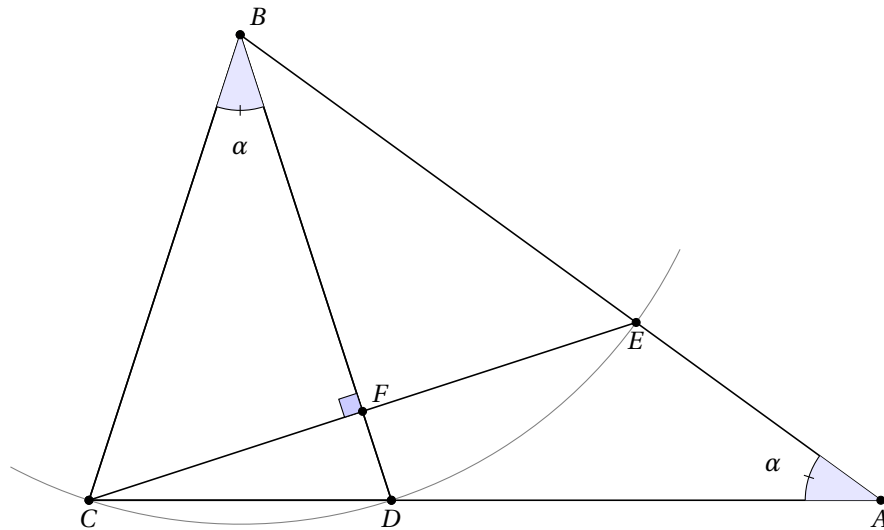
如何构造一个黄金三角形或者顶角为 36° 的等腰三角形呢？

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:

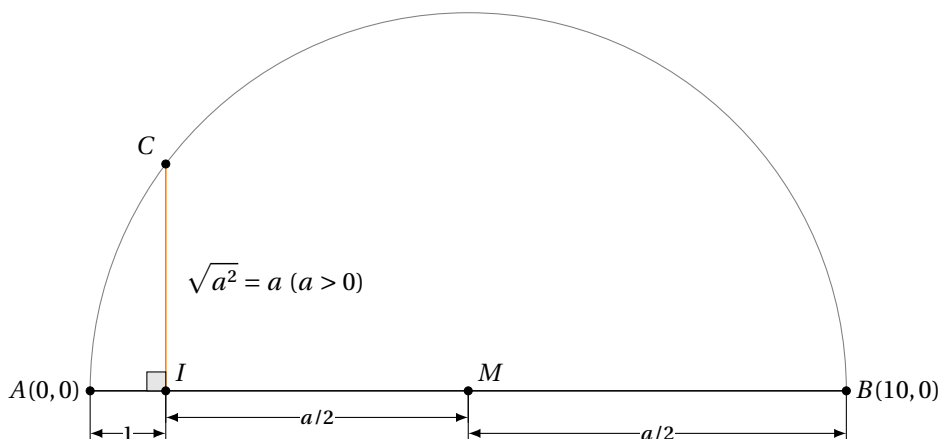
```

\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}

```

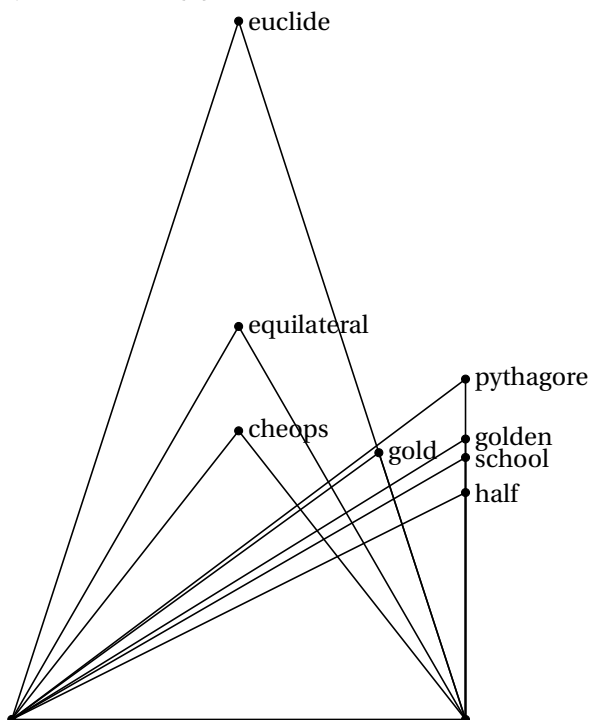
1.4 tkz 的基本要素

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)
```

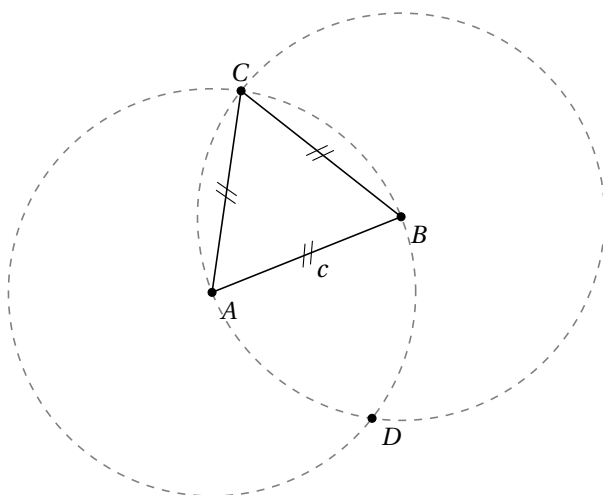
- 绘制标注，在绘制标注时，可以为其指定位置参数。

```
\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，源代码分为引言和正文两大部分。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。