

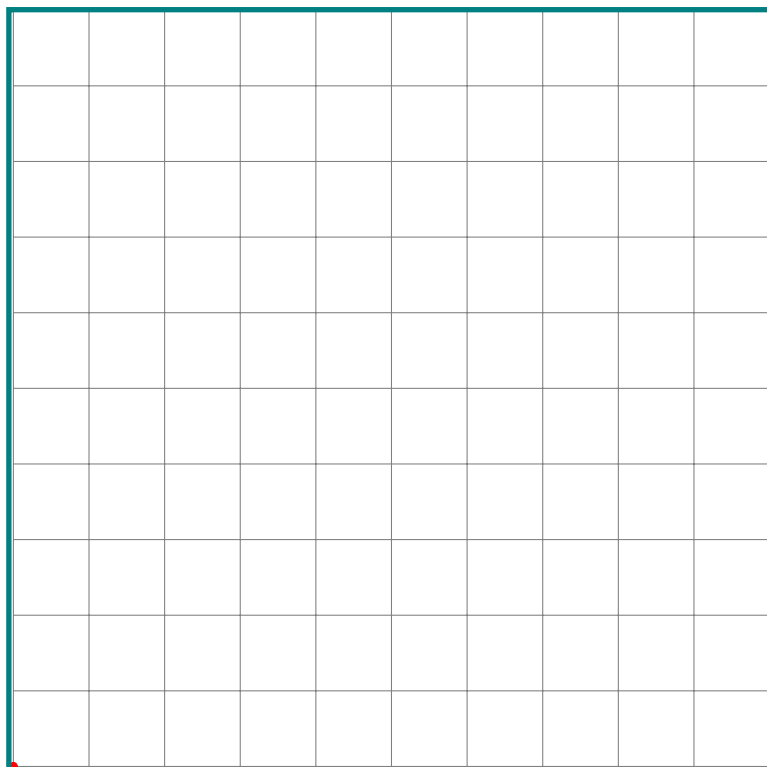
## 1 定义点

可以通过如下方式定义一个点:

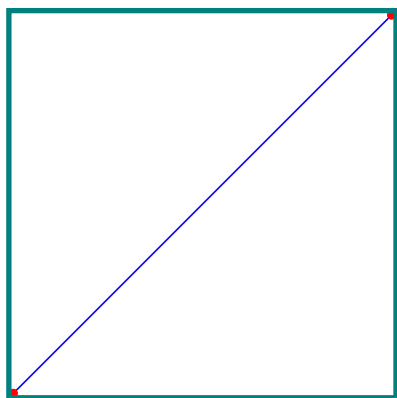
- 笛卡尔坐标;
- 极坐标;
- 为点命名;
- 相对位置.

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}
```

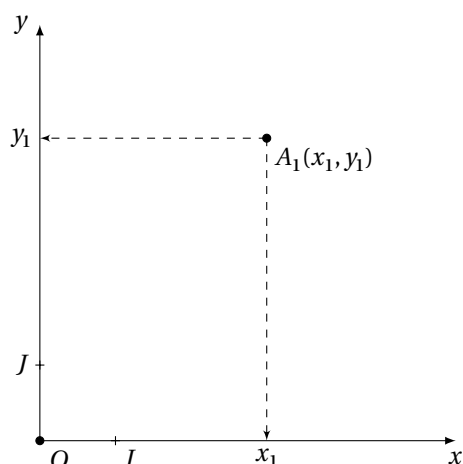


```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\tkzDefPoint(5,5){A}
\tkzDrawSegment[blue](O,A)
\tkzDrawPoints[red](O,A)
\tkzShowBB[line width=2pt,teal]
\end{tikzpicture}
```

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

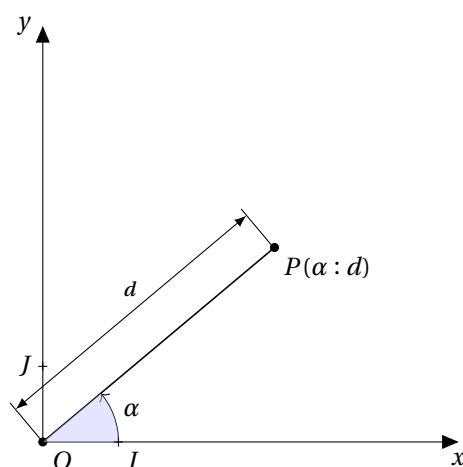
极坐标中表示一个点则需要一个角度  $\alpha$ (度) 和一个从原点度量的距离  $d$ (默认单位是 cm)。

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。

### 1.1 \tkzDefPoint命令：定义命名点

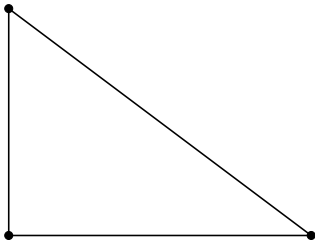
`\tkzDefPoint[< 命令选项>](<x,y>){< 名称>} or (<alpha:d>){< 名称>}`

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

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。

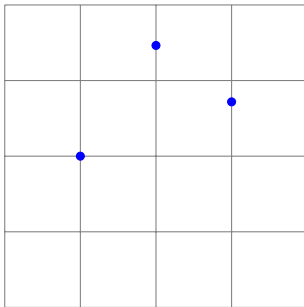
| 选项    | 默认值 | 含义                              |
|-------|-----|---------------------------------|
| label | 无   | 按预设的距离添加标注                      |
| shift | 无   | 为 $(x,y)$ 或 $(\alpha:d)$ 添加坐标偏移 |

## 1.1.1 笛卡尔坐标示例



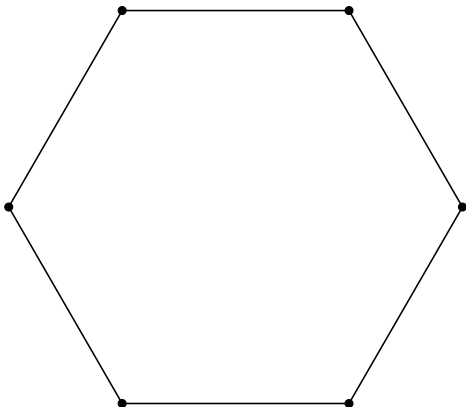
```
\begin{tikzpicture}
  \tkzInit[xmax=5,ymax=5]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(0,3){C}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

## 1.1.2 使用xfp宏包实现计算



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

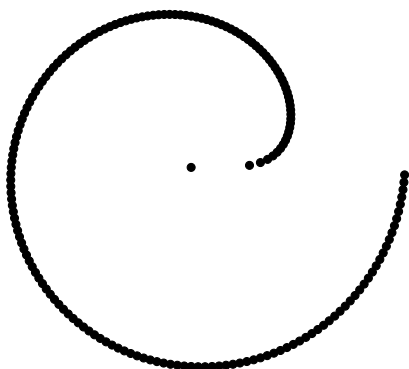
## 1.1.3 极坐标示例



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

## 1.1.4 坐标计算

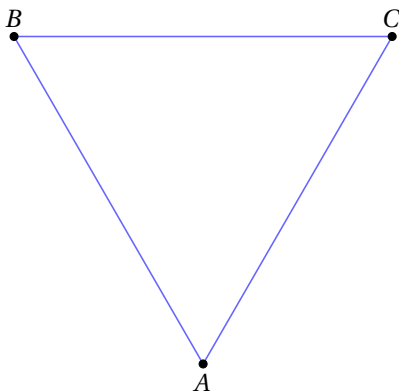
在计算坐标时，需遵循`xfp`宏包语法。另外，如使用`pgfmath`库计算，则必须在使用`\tkzDefPoint`命令前完成计算。



```
\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}
```

## 1.1.5 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}
```

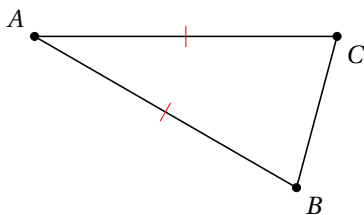
## 1.2 \tkzDefShiftPoint命令：定义平移点

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

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

### 1.2.1 用\tkzDefShiftPoint命令构造等腰三角形

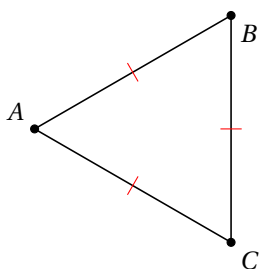
该命令允许基于一个点定义另一个点，等价于点的平移。下面的代码给出了一种通过点  $A$  和  $30^\circ$  顶角等腰三角形的构造方法。



```
\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}
```

### 1.2.2 用\tkzDefShiftPoint命令构造等边三角形

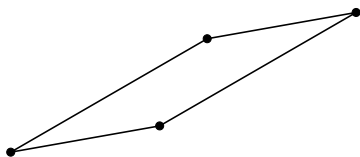
下面的代码给出了一种极为简捷的等边三角形构造方法。



```
\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}
```

### 1.2.3 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}
```

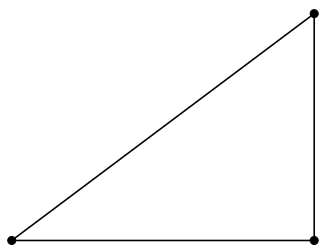
## 1.3 \tkzDefPoints命令：定义点集

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

$x_i$  和  $y_i$  是  $n_i$  点的 2 维坐标

| 参数            | 默认值 | 样例                                      |
|---------------|-----|---|
| $x_i/y_i/n_i$ | 无   | <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      |

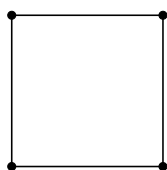
## 1.3.1 用\tkzDefPoints命令构造三角形



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

## 1.3.2 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}
```

## 2 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。

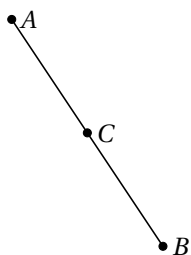
## 2.1 \tkzDefMidPoint命令：定义线段中点

**\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 |

## 2.1.1 \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}
```

## 2.2 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= $\alpha_1$ ,pt2= $\alpha_2$ ,...>)`

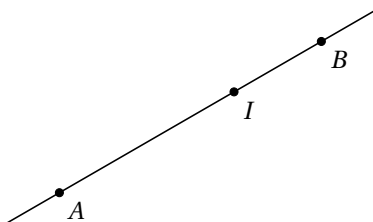
| 参数   | 默认值 | 含义     |
|--|-----|--------|
| <code>(pt1=<math>\alpha_1</math>,pt2=<math>\alpha_2</math>,...)</code> | 无   | 每个点的权重 |

注意：至少需要两个已知点，才能实现计算。

### 2.2.1 用`\tkzDefBarycentricPoint`命令计算 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：

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

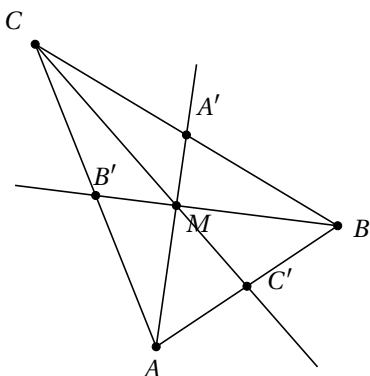


```
\begin{tikzpicture}
\tkzDefPoint(2,3){A}
\tkzDefShiftPointCoord[2,3](30:4){B}
\tkzDefBarycentricPoint(A=1,B=2)
\tkzGetPoint{I}
\tkzDrawPoints(A,B,I)
\tkzDrawLine(A,B)
\tkzLabelPoints(A,B,I)
\end{tikzpicture}
```

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

下面的代码中， $M$  是三角形的重心。

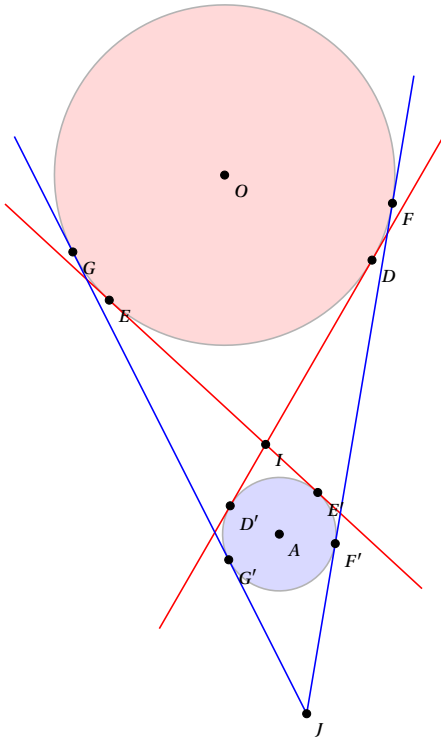
注意，为简化操作，该宏包还提供了另外一个用于直接计算三角形重心的`\tkzCentroid`命令。



```
\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}
```

## 2.3 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}

```