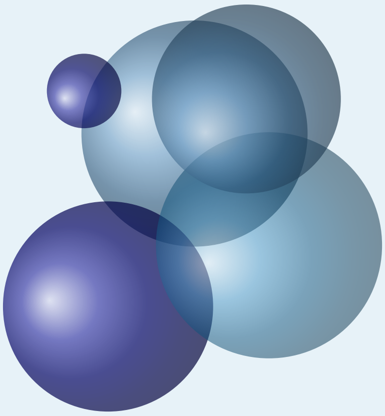
tkz-euclide 5.02c

AlterMundus

tkz-euclide v5

Euclidean Geometry



著：Alain Matthes

译：hnkznhb(海南·聂海波)

May 12, 2023 Documentation V.5.02c

http://altermundus.fr

tkz-euclide

AlterMundus

Alain Matthes

☞ tkz-euclide 在第 5 版中通过了，可以使用 lua 来进行部分计算。更多信息请参见“news 新特性” 和 “lua” 部分。

tkz-euclide 是一组方便的宏，用于在平面（基于二维物体）上用笛卡尔坐标系绘图。它可以处理欧几里得几何中最经典的情况。tkz-euclide 是建立在 PGF 及其相关前端的 Ti*k*Z 之上，是一个 (La)TeX 友好的绘图包。其目的是提供一个高层次的用户界面，以相对简单地建立图形。我们的想法是让你尽可能自然地一步步符合手工绘图的绘图方式。

英语不是我的母语，所以可能有一些错误。

☞ 首先，感谢 **Till Tantau** 开发了强大的 Ti*k*Z 绘图工具 Ti*k*Z

☞ 致谢：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** 关于量角器上的工作, **Paul Gaborit**, **Laurent Van Deik** 的所有更正、评论和问题，**Muzimuzhi Z** 关于选项“dim”的代码。非常感谢 **Chetan Shirore** 和 **Dr. Ajit Kumar**，因为他们在包 **luamaths** 中对复数的工作对我帮助很大。

☞ 感谢 MathWorld 的创造人 Eric Weisstein: MathWorld

☞ 你可以在我的网站上找到一些例子: altermundus.fr.　　　　建设中!

如果发现该文档的错误或有其他任何意见和建议，请发信至：Alain Matthes.
如果发现译文的错误或其有他任何意见和建议，请发信至：聂海波.
参考文献 3.06c 中文版 耿楠.

Contents

Part I.

General survey : a brief but comprehensive review

概述 : 简短而全面的回顾

News and compatibility 新功能和兼容性

Q.1. With 5.0 version 关于版本 5.0

- Finally, I added the ”lua” option for the package `tkz-euclide`. This allows to do the calculations for the main functions using lua; (see 1). The syntax is unchanged. Nothing changes for the user.
- 最后，为包 `tkz-euclide` 添加了“lua”选项。这允许使用 lua 对主要函数进行计算;（请参见 1）。语法不变。对用户来说没有什么变化。
- The ”xfp” option has become ”veclen” see 28.14;
- “xfp”选项已变为“veclen”，请参见 28.14;

Q.2. With 4.2 version 关于版本 4.2

Some changes have been made to make the syntax more homogeneous and especially to distinguish the definition and search for coordinates from the rest, i.e. drawing, marking and labelling.
为了使语法更加统一，特别是为了将定义和坐标搜索与其他内容 (如绘图、标记和标签) 区分开来，进行了一些更改。
Now the definition macros are isolated, it will be easier to introduce a phase of coordinate calculations using **Lua**.
现在定义宏被隔离了，使用 **Lua** 引入一个坐标计算阶段将会更容易。
Here are some of the changes.
以下是一些更改。

- I recently discovered a problem when using the ”scale” option. When plotting certain figures with certain tools, extensive use of `pgfmathreciprocal` involves small computational errors but can add up and render the figures unfit. Here is how to proceed to avoid these problems:
- 我最近在使用“缩放”选项时发现了一个问题。当使用某些工具绘制某些图形时，大量使用`pgfmathreciprocal` 会产生很小的计算误差，但会累积起来，使图形变得不准确。以下是避免这些问题的方法:
  1. On my side I introduced a patch proposed by Muzimuzhi that modifies `pgfmathreciprocal`;
  2. 这里，我介绍了 Muzimuzhi 提出的一个补丁，它修改了`pgfmathreciprocal`
  3. Another idea proposed by Muzimuzhi is to pass as an option for the `tikzpicture` environment this `/pgf/fpu/install only={reciprocal}` after loading of course the `fpu` library;
  4. Muzimuzhi 提出的另一个想法是，在加载`fpu` 库之后，将 `/pgf/fpu/install only={reciprocal}` 作为选项传递给`tikzpicture` 环境；
  5. I have in the methods chosen to define my macros tried to avoid as much as possible the use of `pgfmathreciprocal`;
  6. 在选择定义宏的方法时，我尽可能避免使用 `pgfmathreciprocal`
  7. There is still a foolproof method which consists in avoiding the use of `scale = ...`. It's quite easy if, like me, you only work with fixed points fixed at the beginning of your code. The size of your figure depends only on these fixed points so you just have to adapt the coordinates of these.
  8. 仍然有一种万无一失的方法，包括避免使用`scale = ...`。如果像我一样，你只在代码开头使用固定的点，那就很容易了。你的图形的大小只取决于这些固定点，所以你只需要调整这些点的坐标。

- Now `\tkzDefCircle` gives two points as results: the center of the circle and a point of the circle. When a point of the circle is known, it is enough to use `\tkzGetPoint` or `\tkzGetFirstPoint` to get the center, otherwise `\tkzGetPoints` will give you the center and a point of the circle. You can always get the length of the radius with `\tkzGetLength`. I wanted to favor working with nodes and banish the appearance of numbers in the code.

- 现在 \tkzDefCircle 给出两个点作为结果：圆心和圆点。已知圆的一点时，使用 \tkzGetPoint 或 \tkzGetFirstPoint 求圆心就够了，否则\tkzGetPoints 会给你圆的圆心和 1 圆上一点。总是可以使用\tkzGetLength 获得半径的长度。我想支持使用节点，并消除代码中数字的出现。

- In order to isolate the definitions, I deleted or modified certain macros which are: \tkzDrawLine, \tkzDrawTriangle, \tkzDrawCircle, \tkzDrawSemiCircle and \tkzDrawRectangle;

- 为了分离定义，我删除或修改了某些宏，它们是:\tkzDrawLine, \tkzDrawTriangle, \tkzDrawCircle, \tkzDrawSemiCircle 和 \tkzDrawRectangle;

  Thus \tkzDrawSquare(A,B) becomes \tkzDefSquare(A,B)\tkzGetPoints{C}{D} then

  \tkzDrawPolygon(A,B,C,D);

  因此\tkzDrawSquare(A,B) 变成\tkzDefSquare(A,B)\tkzGetPoints{C}{D}，然后

  \tkzDrawPolygon(A,B,C,D);

  If you want to draw a circle, you can't do so \tkzDrawCircle[R](A,1). First you have to define the point through which the circle passes, so you have to do \tkzDefCircle[R](A,1) \tkzGetPoint{a} and finally \tkzDrawCircle(A,a). Another possibilty is to define a point on the circle \tkzDefShiftPoint[A](1,0){a};

  如果想画一个圆，不能用\tkzDrawCircle[R](A,1)。首先必须定义圆通过的点，所以你必须\tkzDefCircle[R](A,1) \tkzGetPoint{a}，最后 \tkzDrawCircle(A,a)。另一种可能性是在圆上定义一个点\tkzDefShiftPoint[A](1,0){a}

- The following macros tkzDefCircleBy[orthogonal through] and \tkzDefCircleBy[orthogonal from] become tkzDefCircle[orthogonal through] and \tkzDefCircle[orthogonal from] ;

- 下列宏tkzDefCircleBy[orthogonal through] 和\tkzDefCircleBy[orthogonal from] 变成tkzDefCircle[orthog 和 \tkzDefCircle[orthogonal from];

- \tkzDefLine[euler](A,B,C) is a macro that allows you to obtain the line of **Euler** when possible. \tkzDefLine[altitu is possible again, as well as \tkzDefLine[tangent at=A](O) and \tkzDefLine[tangent from=P](O,A) which did not works;

- \tkzDefLine[euler](A,B,C) 是一个宏，允许在可能的情况下获取 **Euler** 线。\tkzDefLine[altitude](A, B, C) 又是可能的，另外\tkzDefLine[tangent at=A](O) 和\tkzDefLine[tangent from=P](O, A) 不再使用；

- \tkzDefTangent is replaced by \tkzDefLine[tangent from = ...] or \tkzDefLine[tangent at = ...];

- \tkzDefTangent 替换为\tkzDefLine[tangent from =...] 或\tkzDefLine[tangent at = ...];

- I added the macro \tkzPicAngle[tikz options](A,B,C) for those who prefer to use Ti*k*Z;

- 添加了宏\tkzPicAngle[tikz options](A,B,C) 给喜欢用 TIKZ 的人；

- The macro \tkzMarkAngle has been corrected;

- 宏\tkzMarkAngle 已被更正；

- The macro linked to the **apollonius** option of the \tkzDefCircle command has been rewritten;

- 链接到\tkzDefCircle 命令的 **apollonius** 选项的宏已被重写；

- (4.23) The macro \tkzDrawSemiCircle has been corrected;

- 4.23) 宏\tkzDrawSemiCircle 已被更正；

- The order of the arguments of the macro \tkzDefPointOnCircle has changed: now it is center, angle and point or radius. I have added two options for working with radians which are through in rad and R in rad.

- 宏\tkzDefPointOnCircle 的参数顺序已更改: 现在是中心、角度和点或半径：center, angle and point or radius。添加了两个使用弧度的选项，分别是 through in rad 和 R in rad。

- I added the option reverse to the arcs paths. This allows to reverse the path and to reverse if necessary the arrows that would be present.

- 选项 reverse 添加到弧线路径。这允许反转路径，并在必要时反转可能出现的箭头。

- I have unified the styles for the labels. There is now only label style left which is valid for points, segments, lines, circles and angles. I have deleted label seg style label line style and label angle style

- 统一了标签的样式。现在只剩下对点、线段、线、圆和角度有效的 label style。我已经删除了 label line style 和 label angle style

- I added the macro tkzFillAngles to use several angles.

- 添加了宏tkzFillAngles 来使用几个角度。

- Correction option return witk \tkzProtractor

- 修正选项 return 随着 \tkzProtractor

  As a reminder, the following changes have been made previously:

  提醒一下，之前已经做了以下更改:

- \tkzDrawMedian, \tkzDrawBisector, \tkzDrawAltitude, \tkzDrawMedians, \tkzDrawBisectors et \tkzDrawAltitudes do not exist anymore. The creation and drawing separation is not respected so it is preferable to first create the coordinates of these points with \tkzDefSpcTriangle[median] and then to choose the ones you are going to draw with \tkzDrawSegments or \tkzDrawLines;

- \tkzDrawMedian, \tkzDrawBisector, \tkzDrawAltitude, \tkzDrawMedians, \tkzDrawBisectors，等等，\tkzDrawAltitudes 不再存在。不考虑创建和绘图的分离，因此最好先用\tkzDefSpcTriangle[median] 创建这些点的坐标，然后选择要用\tkzDrawSegments 或\tkzDrawLines 绘制的点；

- \tkzDrawTriangle has been deleted. \tkzDrawTriangle[equilateral] was handy but it is better to get the third point with \tkzDefTriangle[equilateral] and then draw with \tkzDrawPolygon; idem for \tkzDrawSquare and \tkzDrawGoldRectangle;

- \tkzDrawTriangle 已被删除。\tkzdraw triangle[equirement] 很方便，但是最好用\tkzDefTriangle[equirement] 得到第三个点，然后用\tkzDrawPolygon 绘制；同上，适用于 \tkzDrawSquare 和\tkzDrawGoldRectangle

- The circle inversion was badly defined so I rewrote the macro. The input arguments are always the center and a point of the circle, the output arguments are the center of the image circle and a point of the image circle or two points of the image line if the antecedent circle passes through the pole of the inversion. If the circle passes the inversion center, the image is a straight line, the validity of the procedure depends on the choice of the point on the antecedent circle;

- 圆反演定义不正确，所以我重写了宏。输入参数始终是圆的中心和一个点，输出参数是图像圆的中心和图像圆的一个点，或者是图像线的两个点 (如果先前的圆通过反演的极点)。如果圆经过反演中心，图像是一条直线，程序的有效性取决于先行圆上的点的选择；

- Correct allocation for gold sublime and euclide triangles;

- 修正分配黄金 sublime 和欧几里德三角形；

- I added the option ” next to” for the intersections LC and CC;

- 交点命令 LC 和 CC 添加了选项“next to 旁边”；

- Correction option isoceles right;

- 修正等腰直角三角形选项；

- (4.22 and 4.23) Correction of the macro `\tkzMarkAngle`;

- (4.22 和 4.23) 修正宏`\tkzMarkAngle`；

- `\tkzDefMidArc(O,A,B)` gives the middle of the arc center O from A to B;

- `\tkzDefMidArc(O,A,B)` 给出圆心为点 O、从 A 到 B 的弧的中点；

- Good news : Some useful tools have been added. They are present on an experimental basis and will undoubtedly need to be improved;

- 好消息: 增加了一些有用的工具。它们目前处于试验阶段，无疑需要改进；

- The options ”orthogonal from and through” depend now of `\tkzDefCircleBy`

- 选项“orthogonal from and through”现在取决于`\tkzDefCircleBy`

  1. `\tkzDotProduct(A,B,C)` computes the scalar product in an orthogonal reference system of the vectors $\overrightarrow{A,B}$ and $\overrightarrow{A,C}$.
  2. `\tkzDotProduct(A,B,C)` 在向量 $\overrightarrow{A,B}$ 和 $\overrightarrow{A,C}$ 的直角坐标系中计算标量积。

     `\tkzDotProduct(A,B,C)=aa'+bb' if vec{AB} =(a,b) and vec{AC} =(a',b')`
  3. `\tkzPowerCircle(A)(B,C)` power of point A with respect to the circle of center B passing through C;
  4. `\tkzPowerCircle(A)(B,C)` 点 A 相对于圆心 B 经过点 C 的圆的幂；
  5. `\tkzDefRadicalAxis(A,B)(C,D)` Radical axis of two circles of center A and C;
  6. `\tkzDefRadicalAxis(A,B)(C,D)` 圆心为 A 和 C 的两个圆的根轴；
  7. (4.23) The macro `tkzDefRadicalAxis` has been completed
  8. (4.23) 宏`tkzDefRadicalAxis` 已完成
  9. Some tests : `\tkzIsOrtho(A,B,C)` and `\tkzIsLinear(A,B,C)` The first indicates whether the lines (A, B) and (A, C) are orthogonal. The second indicates whether the points A, B and C are aligned; `\tkzIsLinear(A,B,C)` if A,B,C are aligned then `\tkzLineartrue` you can use `\iftkzLinear` (idem for `\tkzIsOrtho`);

10. 一些测试: \tkzisothor(A,B,C) 指示直线 (A,B) 和 (A,C) 是否正交。\tkzisllinear(A,B,C) 指示点 A 、B 和 C 是否对齐；\tkzilinear(A,B,C) 如果 A，B，C 对齐，则\tkzLineartrue 可以使用 \iftkzLinear(\tkzisothor 同理)；
11. A style for vectors has been added that you can of course modify

    tikzset{vector style/.style={>=Latex,->}};
12. 增加了一个矢量样式，当然可以修改它：

    tikzset{vector style/.style={>=Latex,->}};
13. Now it's possible to add an arrow on a line or a circle with the option tkz arrow.
14. 现在可以用选项tkz arrow 在直线或圆上添加箭头。

0.3. Changes with previous versions 与以前版本相比的变化

- I remind you that an important novelty is the recent replacement of the fp package by xfp. This is to improve the calculations a little bit more and to make it easier to use;

- 提醒，最近用包 xfp 替换了包 fp 是一个重要的创新。这是为了改进计算，使它更容易使用；

- First of all, you don't have to deal with Ti*k*Z the size of the bounding box. Early versions of tkz-euclide did not control the size of the bounding box, The bounding box is now controlled in each macro (hopefully) to avoid the use of \tkzInit followed by \tkzClip;

- 首先，不必处理 Ti*k*Z 边界框的大小。tkz-euclide 的早期版本不控制边界框的大小，现在 (希望) 每个宏都控制边界框以避免使用\tkzInit 后跟\tkzClip。

- With **tkz-euclide** loads all objects, so there's no need to place \usetkzobj{all};

- **tkz-euclide** 加载所有对象，因此无需导入 \usetkzobj{all};

- Added macros for the bounding box: \tkzSaveBB \tkzClipBB and so on;

- 为边界框添加了宏:\tkzSaveBB \tkzClipBB 等等;

- Logically most macros accept Ti*k*Z options. So I removed the "duplicate" options when possible thus the "label options" option is removed;

- 从逻辑上讲，大多数宏都接受 Ti*k*Z 选项。所以我尽可能删除了“重复”选项，因此删除了“标签选项”选项;

- The unit is now the cm;

- 单位现在是厘米 cm;

- \tkzCalcLength \tkzGetLength gives result in cm;

- \tkzCalcLength \tkzGetLength 以厘米为单位给出结果;

- \tkzMarkArc and \tkzLabelArc are new macros;

- \tkzMarkArc 和\tkzLabelArc 是新的宏；
- Now \tkzClipCircle and \tkzClipPolygon have an option **out**. To use this option you must have a Bounding Box that contains the object on which the Clip action will be performed. This can be done by using an object that encompasses the figure or by using the macro \tkzInit;
- 现在 \tkzClipCircle 和 \tkzClipPolygon 有了一个选项 **out**。要使用此选项，必须有一个包含将对其执行裁剪操作的对象的边界框。可以通过使用包含该图形的对象或使用宏\tkzInit 来实现；
- The options `end` and `start` which allowed to give a label to a straight line are removed. You now have to use the macro \tkzLabelLine;
- 允许给直线加标签的选项 `end` 和 `start` 被删除。现在必须使用宏 \tkzLabelLine；
- Introduction of the libraries quotes and angles; it allows to give a label to a point, even if I am not in favour of this practice;
- 介绍库 quotes 和 angles；它允许给一个点贴上标签，但我不赞成这种做法；
- The notion of vector disappears, to draw a vector just pass ”->” as an option to \tkzDrawSegment;
- 矢量的概念消失了，要画一个矢量只需将”->” 作为一个选项传递给\tkzdrawsgment；
- \tkzDefIntSimilitudeCenter and \tkzDefExtSimilitudeCenter do not exist anymore, now you need to use \tkzDefSimilitudeCenter[int] or \tkzDefSimilitudeCenter[ext];
- \tkzDefIntSimilitudeCenter 和\tkzDefExtSimilitudeCenter 不再存在，现在需要使用\tkzDefSimilitudeCenter
或\tkzDefSimilitudeCenter[ext]；
- \tkzDefRandPointOn is replaced by \tkzGetRandPointOn;
- \tkzDefRandPointOn 替换为 \tkzGetRandPointOn;
- An option of the macro \tkzDefTriangle has changed, in the previous version the option was ”euclide” with an ”e”. Now it's ”euclid”;
- 宏\tkzDefTriangle 的一个选项已更改，在以前的版本中，该选项是带“e”的“euclide”。现在是“euclid”；
- Random points are now in `tkz-euclide` and the macro \tkzGetRandPointOn is replaced by \tkzDefRandPointOn. For homogeneity reasons, the points must be retrieved with \tkzGetPoint;
- 在 `tkz-euclide` 中随机点现在可用，宏\tkzGetRandPointOn 被替换为 \tkzDefRandPointOn。出于同质性原因，必须使用\tkzGetPoint 获得这些点。
- New macros have been added : \tkzDrawSemiCircles, \tkzDrawPolygons, \tkzDrawTriangles;
- 添加了新的宏:\tkzDrawSemiCircles, \tkzDrawPolygons, \tkzDrawTriangles;
- Option ”isosceles right” is a new option of the macro \tkzDefTriangle;
- 宏\tkzDefTriangle 有一个新选项 “isosceles right”（等腰直角三角形）；

- Appearance of the macro `\usetkztool` which allows to load new ”tools”;

- 宏`\usetkztool` 允许加载新 “tools（工具）”；

- The styles can be modified with the help of the following macros : `\tkzSetUpPoint`, `\tkzSetUpLine`, `\tkzSetUpArc`, `\tkzSetUpCompass`, `\tkzSetUpLabel` and `\tkzSetUpStyle`. The last one allows you to create a new style.

- 在以下宏的帮助下，可以修改样式：`\tkzSetUpPoint`, `\tkzSetUpLine`, `\tkzSetUpArc`, `\tkzSetUpCompass`, `\tkzSetUpLabel` and `\tkzSetUpStyle`。最后一个允许创建一个新的样式。

1. Working with lua : option lua 使用 lua

You can now use the ”lua” option with **tkz-euclide** version 5. You just have to write in your preamble
现在，您可以在 **tkz-euclide** 版本 5 中使用 “lua” 选项。你只需要在序言中写下
`usepackage[lua]{tkz-euclide}`.
Évidemment vous devrez compiler avec LuaLaTeX. Nothing changes for the syntax.
请使用 LuaLaTeX 开发编译器。语法没有变化。
Without the option you can use **tkz-euclide** with the proposed code of version 4.25.
如果没有该选项，可以将 **tkz-euclide** 与 4.25 版的建议代码一起使用。
This version is not yet finalized although the documentation you are currently reading has been compiled with this option.
尽管当前正在阅读的文档，已经使用此选项进行了编译，但此版本尚未最终确定。
Some information about the method used and the results obtained. Concerning the method, I considered two possibilities. The first one was simply to replace everywhere I could the calculations made by ”xfp” or sometimes by ”lua”. This is how I went from ”fp” to ”xfp” and now to ”lua”. The second and more ambitious possibility would have been to associate to each point a complex number and to make the calculations on the complexes with ”lua”. Unfortunately for that I have to use libraries for which I don’t know the license.
关于所用方法和所得结果的一些信息。关于方法，我考虑了两种可能性。第一个是简单地在任何地方替换由 “xfp” 或者有时由 “lua” 进行的计算。我就是这样从 “fp” 到 “xfp”，再到现在的 “lua”。第二种也是更大胆的可能性是将每个点与一个复数相关联，并用 “lua” 在复数上进行计算。不幸的是，我不得不使用我不知道许可的库。
Otherwise the results are good. This documentation with ”LualaTeX” and ”xfp” compiles in 47s while with ”lua” it takes only 30s for 236 pages.
相反，结果是好的。这个文档用 “lualaTeX” 和 “xfp” 编译需要 47 秒，而用 “Lua” 编译 236 页只需要 30 秒。
Another document of 61 pages is compiled 16s with ”pdflaTeX” and ”xfp” and 13s with ”LualaTeX” and ”xfp”.
另一个 61 页的文档，用 “pdflaTeX” 和 “xfp” 编译需要 16s，用 “LualaTeX” 和 “xfp” 编译需要 13s。
This documentation compiles with `\usepackage{tkz-base}` and `\usepackage[lua]{tkz-euclide}` but I didn’t test all the interactions thoroughly.
本文档使用 `\usepackage{tkz-base}` 和 `\usepackage[lua]{tkz-euclide}` 进行编译，但是我没有彻底测试所有的交互。

2. Installation 安装

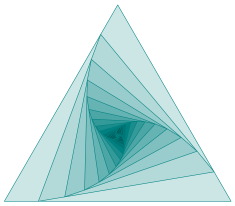
tkz-euclide is on the server of the CTAN[1]. If you want to test a beta version, just put the following files in a texmf folder that your system can find. You will have to check several points:
tkz-euclide 在 CTAN 的服务器上[2]。如果想测试 beta 版本，只需将以下文件放在系统可以找到的 texmf 文件夹中。并注意以下几点:

- The tkz-euclide folder must be located on a path recognized by latex.
- tkz-euclide 文件夹必须位于 latex 识别的路径上。
- The tkz-euclide uses xfp.
- tkz-euclide 使用 xfp。
- You need to have PGF installed on your computer. tkz-euclide use several libraries of Ti*k*Z
- 需要在计算机上安装 PGF 软件。tkz-euclide 使用几个 Ti*k*Z 库

  angles,
  arrows,
  arrows.meta,
  calc,
  decorations,
  decorations.markings,
  decorations.pathreplacing,
  decorations.shapes,
  decorations.text,
  decorations.pathmorphing,
  intersections,
  math,
  plotmarks,
  positioning,
  quotes,
  shapes.misc,
  through
- This documentation and all examples were obtained with lualatex but pdflatex or xelatex should be suitable.
- 本文档和所有示例都是使用 lualatex 获得的，但 pdflatex 或 xelatex 应该也适用。

[1] tkz-euclide is part of TeXLive and tlmgr allows you to install them. This package is also part of MiKTeX under Windows.
[2] tkz-euclide 是 TeXLive 的一部分，并且 tlmgr 允许安装它们。此软件包也是 Windows 下的 MiKTeX 的一部分。

3. 演示和概述



```
\begin{tikzpicture}[scale=.25]
\tkzDefPoints{0/0/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}
```

【例子，可以在后面学习相关内容后再重新阅读。
第一行，定义点。包括直接定义，与计算定义。
第二行，利用循环。
第三行，画多边形。
后面指定循环的方式。】

3.1. 为什么使用 tkz-euclide?

My initial goal was to provide other mathematics teachers and myself with a tool to quickly create Euclidean geometry figures without investing too much effort in learning a new programming language. Of course, **tkz-euclide** is for math teachers who use LaTeX and makes it possible to easily create correct drawings by means of LaTeX.
我最初的目标是为其他数学老师和我自己提供一个工具，可以快速绘制欧几里得几何图形，而无需在学习新的编程语言上投入太多精力。当然，**tkz-euclide** 是为使用 LaTeX 的数学教师设计的，它可以通过 LaTeX 轻松创建正确的绘图。
It appeared that the simplest method was to reproduce the one used to obtain construction by hand.
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。
To describe a construction, you must, of course, define the objects but also the actions that you perform. It seemed to me that syntax close to the language of mathematicians and their students would be more easily understandable; moreover, it also seemed to me that this syntax should be close to that of LaTeX.
当然，要描述一个构造，必须定义对象以及执行的操作。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。
The objects, of course, are points, segments, lines, triangles, polygons and circles. As for actions, I considered five to be sufficient, namely: define, create, draw, mark and label.
当然，对象是点、线段、线、三角形、多边形和圆形。至于动作，我认为五个就足够了，即: 定义、创建、绘制、标记和标注。
The syntax is perhaps too verbose but it is, I believe, easily accessible. As a result, the students like teachers were able to easily access this tool.
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。

3.2. TikZ vs tkz-euclide

I love programming with Ti*k*Z, and without Ti*k*Z I would never have had the idea to create **tkz-euclide** but never forget that behind it there is Ti*k*Z and that it is always possible to insert code from Ti*k*Z. **tkz-euclide** doesn't prevent you from using Ti*k*Z.
我喜欢用 Ti*k*Z 编程，如果没有 Ti*k*Z，我永远也不会有创建 **tkz-euclide** 的想法，但是永远不要忘记在它的背后有 Ti*k*Z，并且总是可以从 Ti*k*Z 插入代码。**tkz-euclide** 不会阻止您使用 Ti*k*Z。
That said, I don't think mixing syntax is a good thing. 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。
There is no need to compare Ti*k*Z and **tkz-euclide**. The latter is not addressed to the same audience as Ti*k*Z. The first one allows you to do a lot of things, the second one only does geometry drawings. The first one can do everything the second one does, but the second one will more easily do what you want.

没有必要比较 Ti*k*Z 和 **tkz-euclide** 。后者的受众不同于 Ti*k*Z。前者允许你做很多事情，后者只做几何图形。前者可以做后者做的一切，但后者会更容易做你想做的事情。
The main purpose is to define points to create geometrical figures. **tkz-euclide** allows you to draw the essential objects of Euclidean geometry from these points but it may be insufficient for some actions like coloring surfaces. In this case you will have to use Ti*k*Z which is always possible.
主要目的是定义点以创建几何图形。**tkz-euclide** 允许从这些点绘制欧几里得几何的基本对象，但对于某些操作 (如给表面着色) 来说，这可能是不够的。在这种情况下，将不得不使用 “TIKZ ”, 它总是可能的。
Here are some comparisons between **TikZ** and **tkz-euclide** 4. For this I will use the geometry examples from the PGFManual. The two most important Euclidean tools used by early Greeks to construct different geometrical shapes and angles were a compass and a straightedge. My idea is to allow you to follow step by step a construction that would be done by hand (with compass and straightedge) as naturally as possible.
以下是 **TikZ** 和 **tkz-euclide** 4 之间的一些比较。为此，我将使用手册中的几何示例。早期希腊人用来构造不同几何形状和角度的两个最重要的欧几里得工具是圆规和直尺。我的想法是让你尽可能自然地一步一步地用手工 (用圆规和直尺) 完成一个绘图。

3.2.1. Book I, 命题 I _几何原本_

Book I, proposition I _Euclid’s Elements_

To construct an equilateral triangle on a given finite straight line.

Book I, 命题 I _几何原本_

在给定的有限直线上构造一个等边三角形。

Explanation :
解释:
The fourth tutorial of the *PgfManual* is about geometric constructions. *T. Tantau* proposes to get the drawing with its beautiful tool Ti*k*Z. Here I propose the same construction with *tkz-elements*. The color of the Ti*k*Z code is orange and that of *tkz-elements* is red.
PgfManual 第四个教程是关于几何构造的。*T. Tantau* 建议使用其漂亮的工具 Ti*k*Z 来获取绘图。在这里，我建议使用 *tkz-elements* 进行相同的构造。Ti*k*Z 代码的颜色为橙色，而 *tkz-elements* 的颜色为红色。

【以下是作者用两种方案做对比，主是想体现本宏包的分离思想。
某种程度上，两种绘图语言都比较烦琐。但是考虑到可以在 pdf 中绘图，并且达到方便，美观的程度，真不容易。
学习过一些时间后，本人感觉，其实与网络画板、GGB、几何画板等作图软件相比，一个是所谓的动态，一个是命令行。其实，本宏包，指定的点的坐标之后，后续的绘图，其实与画板类的软件极其相似，不过是是用手工命令，达成了直观操作而已。画板中的点的动态，只要在宏包里改变坐标即可达到目的。两类软件毕竟还是区别的，但联系并不少。
建议从点、线、面等图形的作法开始，依平面几何学习的次序学习，基本上就可以快速掌握本宏包用法。毕竟，作者的思想，体现了尺规作图的过程，不过是将手工绘图，改写成了命令行而已。】

```
\usepackage{tikz}
\usetikzlibrary{calc,intersections,through,backgrounds}

\usepackage{tkz-euclide}
```

How to get the line AB ? To get this line, we use two fixed points.

怎么得到线段 AB？为了得到这条线，我们使用两个固定点。

```
\coordinate [label=left:$A$] (A) at (0,0);
\coordinate [label=right:$B$] (B) at (1.25,0.25);
\draw (A) -- (B);
```

```
\tkzDefPoint(0,0){A}
\tkzDefPoint(1.25,0.25){B}
\tkzDrawSegment(A,B)
\tkzLabelPoint[left](A){$A$}
\tkzLabelPoint[right](B){$B$}
```

We want to draw a circle around the points A and B whose radius is given by the length of the line AB.
想在点 A 和 B 周围画一个圆，其半径由线段 AB 的长度给出。

```
\draw let \p1 = ($ (B) - (A) $),
\n2 = {veclen(\x1,\y1)} in
          (A) circle (\n2)
          (B) circle (\n2);

\tkzDrawCircles(A,B B,A)
```

The intersection of the circles $\mathscr{D}$ and $\mathscr{E}$
圆 $\mathscr{D}$ 和 $\mathscr{E}$ 的交点

```
draw [name path=A--B] (A) -- (B);
node (D) [name path=D,draw,circle through=(B),label=left:$D$] at (A) {};
node (E) [name path=E,draw,circle through=(A),label=right:$E$] at (B) {};
path [name intersections={of=D and E, by={[label=above:$C$]C, [label=below:$C'$]C'}}];
draw [name path=C--C',red] (C) -- (C');
path [name intersections={of=A--B and C--C',by=F}];
node [fill=red,inner sep=1pt,label=-45:$F$] at (F) {};

\tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{X}
```

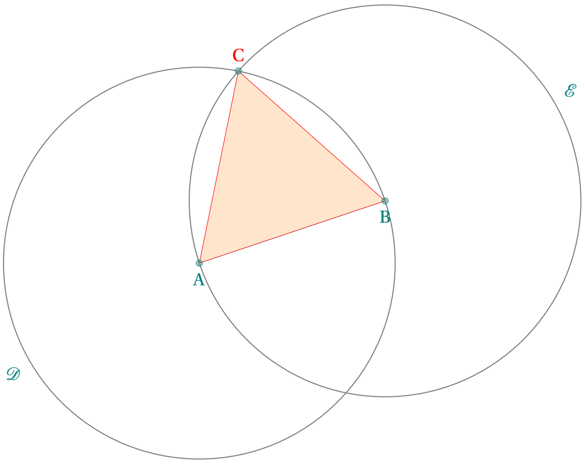
How to draw points :
如何画点:

```
\foreach \point in {A,B,C}
\fill [black,opacity=.5] (\point) circle (2pt);

 \tkzDrawPoints[fill=gray,opacity=.5](A,B,C)
```

3.2.2. Complete code with tkz-euclide
tkz-euclide 完整代码

We need to define colors
我们需要定义颜色

```
\colorlet{input}{red!80!black}
\colorlet{output}{red!70!black}
\colorlet{triangle}{orange!40}
```



```
\colorlet{input}{red!80!black}
\colorlet{output}{red!70!black}
\colorlet{triangle}{orange!40}
\begin{tikzpicture}[scale=1.25,thick,help lines/.style={thin,draw=black!50}]
\tkzDefPoint(0,0){A}
\tkzDefPoint(1.25+rand(),0.25+rand()){B}
\tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{X}

\tkzFillPolygon[triangle,opacity=.5](A,B,C)
\tkzDrawSegment[input](A,B)
\tkzDrawSegments[red](A,C B,C)
\tkzDrawCircles[help lines](A,B B,A)
\tkzDrawPoints[fill=gray,opacity=.5](A,B,C)

\tkzLabelPoints(A,B)
\tkzLabelCircle[below=12pt](A,B)(180){$\mathcal{D}$}
\tkzLabelCircle[above=12pt](B,A)(180){$\mathcal{E}$}
\tkzLabelPoint[above,red](C){$C$}

\end{tikzpicture}
```

3.2.3. Book I, 命题 II _几何原本_

Book I, Proposition II _Euclid’s Elements_

To place a straight line equal to a given straight line with one end at a given point.

Book I, 命题 II _几何原本_

画一条与给定线段相等的线段，其一端在给定点。

Explanation
说明
In the first part, we need to find the midpoint of the straight line AB. With Ti*k*Z we can use the calc library
在第一部分中，需要找到线段 AB 的中点。通过 Ti*k*Z 我们可以使用计算库

```
\coordinate [label=left:$A$] (A) at (0,0);
\coordinate [label=right:$B$] (B) at (1.25,0.25);
\draw (A) -- (B);
\node [fill=red,inner sep=1pt,label=below:$X$] (X) at ($ (A)!.5!(B) $) {};
```

With `tkz-euclide` we have a macro `\tkzDefMidPoint`, we get the point X with `\tkzGetPoint` but we don't need this point to get the next step.
使用 `tkz-euclide` 有一个宏 `\tkzDefMidPoint` ，用 `\tkzGetPoint` 得到点 X，但是不需要这个点来得到下一步。

```
\tkzDefPoints{0/0/A,0.75/0.25/B,1/1.5/C}
\tkzDefMidPoint(A,B) \tkzGetPoint{X}
```

Then we need to construct a triangle equilateral. It's easy with `tkz-euclide` . With TikZ you need some effort because you need to use the midpoint X to get the point D with trigonometry calculation.
然后需要构建一个等边三角形。用 `tkz-euclide` 很简单。使用 TikZ，需要付出一些努力，因为需要使用中点 X 来获得三角计算的点 D。

```
\node [fill=red,inner sep=1pt,label=below:$X$] (X) at ($ (A)!.5!(B) $) {};
\node [fill=red,inner sep=1pt,label=above:$D$] (D) at
($ (X) ! {sin(60)*2} ! 90:(B) $) {};
\draw (A) -- (D) -- (B);
```

```
\tkzDefTriangle[equilateral](A,B) \tkzGetPoint{D}
```

We can draw the triangle at the end of the picture with
可以在绘图的最后，画一个三角形

```
\tkzDrawPolygon{A,B,C}
```

We know how to draw the circle $\mathscr{H}$ around B through C and how to place the points E and F
我们知道如何以点 B 为圆心，过点 C 画圆 $\mathscr{H}$，以及如何放置点 E 和 F

```
\node (H) [label=135:$H$,draw,circle through=(C)] at (B) {};
\draw (D) -- ($ (D) ! 3.5 ! (B) $) coordinate [label=below:$F$] (F);
\draw (D) -- ($ (D) ! 2.5 ! (A) $) coordinate [label=below:$E$] (E);

\tkzDrawCircle(B,C)
\tkzDrawLines[add=0 and 2](D,A D,B)
```

We can place the points E and F at the end of the picture. We don't need them now.
可以将点 E 和 F 放在图形的末尾。现在不需要它们了。
Intersecting a Line and a Circle : here we search the intersection of the circle around B through C and the line DB. The infinite straight line DB intercepts the circle but with Ti*k*Z we need to extend the lines DB and that can be done using partway calculations. We get the point F and BF or DF intercepts the circle
一条线和一个圆相交: 这里搜索点 B 为圆心，过点 C 的圆和线 DB 的交点。直线 DB 与圆相交，但是使用 Ti*k*Z 需要延长直线 DB, 这可以使用部分计算来完成。BF 或 DF 与圆相交得到点 F

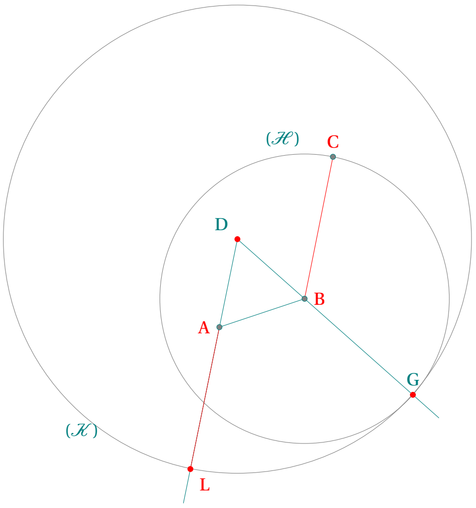
```
 \node (H) [label=135:$H$,draw,circle through=(C)] at (B) {};
\path let \p1 = ($ (B) - (C) $) in
  coordinate [label=left:$G$] (G) at ($ (B) ! veclen(\x1,\y1) ! (F) $);
\fill[red,opacity=.5] (G) circle (2pt);
```

Like the intersection of two circles, it's easy to find the intersection of a line and a circle with `tkz-euclide`. We don't need F
就像两个圆的交点一样，用 `tkz-euclide` 很容易找到一条线和一个圆的交点。不需要 F

```
\tkzInterLC(B,D)(B,C)\tkzGetFirstPoint{G}
```

There are no more difficulties. Here the final code with some simplications. We draw the circle $\mathcal{K}$ with center D and passing through G. It intersects the line AD at point L. AL = BC.
没有更多的困难。这是经过一些简化的最终代码。以点 D 为圆心，经过点 G，画圆 $\mathcal{K}$。它与直线 AD 在点 L 处相交。AL = BC。

```
\tkzDrawCircle(D,G)
\tkzInterLC(D,A)(D,G)\tkzGetSecondPoint{L}
```



```
\begin{tikzpicture}[scale=1.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(0.75,0.25){B}
\tkzDefPoint(1,1.5){C}
\tkzDefTriangle[equilateral](A,B)    \tkzGetPoint{D}
\tkzInterLC[near](D,B)(B,C)          \tkzGetSecondPoint{G}
\tkzInterLC[near](A,D)(D,G)          \tkzGetFirstPoint{L}
\tkzDrawCircles(B,C D,G)
\tkzDrawLines[add=0 and 2](D,A D,B)
\tkzDrawSegment(A,B)
\tkzDrawSegments[red](A,L B,C)
\tkzDrawPoints[red](D,L,G)
\tkzDrawPoints[fill=gray](A,B,C)
\tkzLabelPoints[left,red](A)
\tkzLabelPoints[below right,red](L)
\tkzLabelCircle[above](B,C)(20){$\mathcal{(H)}$}
\tkzLabelPoints[above left](D)
\tkzLabelPoints[above](G)
\tkzLabelPoints[above,red](C)
\tkzLabelPoints[right,red](B)
\tkzLabelCircle[below](D,G)(-90){$\mathcal{(K)}$}
\end{tikzpicture}
```

3.3. tkz-euclide 4 vs tkz-euclide 3

Now I am no longer a Mathematics teacher, and I only spend a few hours studying geometry. I wanted to avoid multiple complications by trying to make `tkz-euclide` independent of `tkz-base`. Thus was born `tkz-euclide` 4. The latter is a simplified version of its predecessor. The macros of `tkz-euclide 3` have been retained. The unit is now `cm`. If you need some macros from `tkz-base`, you may need to use the `\tkzInit`.
现在我已经不是数学老师了，只花几个小时研究几何。我想通过尝试使 `tkz-euclide` 独立于 `tkz-base` 来避免多种复杂性。就这样诞生了 `tkz-euclide` 4。后者是其前者的简化版本。保留了 `tkz-euclide 3` 的宏。长度单位现在是 `cm`。如果需要一些来自 `tkz-base` 的宏，可能需要使用`\tkzInit`。

3.4. tkz-euclide 5 vs tkz-euclide 4

Rien ne change pour l'utilisateur. La compilation doit être effectuée avec le moteur LuaLaTeX et les résultats sont plus précis et obtenus plus rapidement. Il suffit de charger `tkz-euclide` 5 comme ceci `\usepackage[lua{tkz-euclide}]`.
对用户来说没有什么变化。编译必须使用 LuaLaTeX 引擎进行，以便结果更准确，更快。只需 用`\usepackage[lua{tkz-euclide}]` 加载 `tkz-euclide` 5。

3.5. How to use the tkz-euclide package ?
如何使用 tkz-euclide 宏包？

3.5.1. Let's look at a classic example

一个经典的例子

In order to show the right way, we will see how to build an equilateral triangle. Several possibilities are open to us, we are going to follow the steps of Euclid.
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。

– First of all, you have to use a document class. The best choice to test your code is to create a single figure with the class **standalone**.

  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}
```

– Then load the **tkz-euclide** package:

– 然后载入 **tkz-euclide** 宏包：

```
\usepackage{tkz-euclide} or \usepackage[lua]{tkz-euclide}
```

  You don't need to load Ti*k*Z because the **tkz-euclide** package works on top of TikZ and loads it.

  由于 **tkz-euclide** 宏包是基于 Ti*k*Z 宏包开发的，会同时载入该宏包，因此，无需再次载入 Ti*k*Z 宏包

– Start the document and open a TikZ picture environment:

– 开始文档，并使用 tikzpicture 环境绘制欧氏几何图形：

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

– Now we define two fixed points:

– 定义两个已知点：

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

– Two points define two circles, let's use these circles:

– 使用这两个点定义两个圆，并使用这两个圆定义交点：

  circle with center A through B and circle with center B through A. These two circles have two points in common.

  (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，两个圆有两个公共点 A 和点 B。

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

  We can get the points of intersection with

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

  得到两个圆的交点，并命名为 C 和 D

– All the necessary points are obtained, we can move on to the final steps including the plots.

– 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)% The triangle
```

- Draw all points A, B, C and D:
- 绘制点 A、B、C 和 D：

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

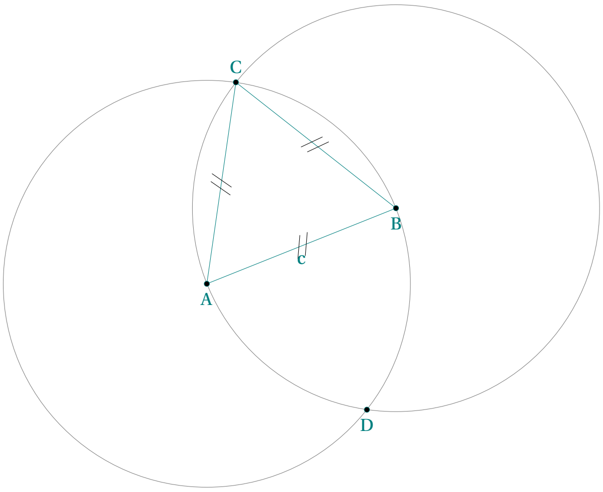
- The final step, we print labels to the points and use options for positioning:
- 绘制标注，在绘制标注时，可以为其指定位置参数。

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

- We finally close both environments
- 最后，结束各个环境

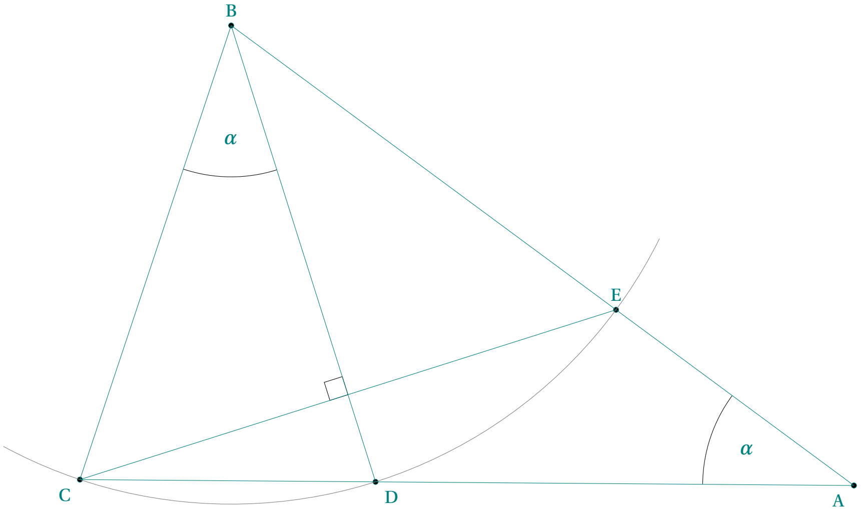
```
\end{tikzpicture}
\end{document}
```

- The complete code
- 完整代码



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

3.5.2. Part I: golden triangle 第 I 部分：黄金三角形



Let’s analyze the figure
其绘制过程如下：

1. CBD and DBE are isosceles triangles;
   CBD 和 DBE 都是等腰三角形；

2. BC = BE and (BD) is a bisector of the angle CBE;
   BC = BE，BD 是角 CBE 的角平分线。

3. From this we deduce that the CBD and DBE angles are equal and have the same measure α
   由此推出角 CBD 和角 DBE 相等，在此，记为 α。

$$\widehat{\mathrm{BAC}}+\widehat{\mathrm{ABC}}+\widehat{\mathrm{BCA}}=180^\circ \text{ in the triangle BAC}$$

$$3\alpha+\widehat{\mathrm{BCA}}=180^\circ \text{ in the triangle CBD}$$

   then

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

   or

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

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

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

   因此

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

   或

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

4. Finally

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

   从而可得

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

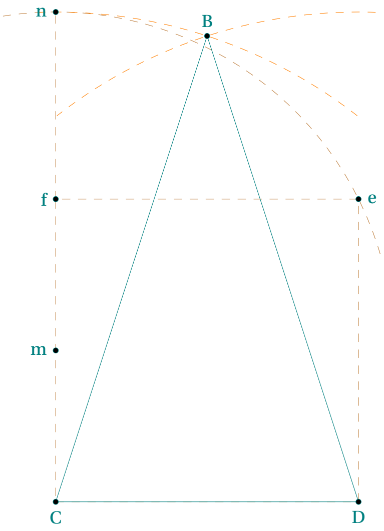
   the triangle CBD is a ”golden” triangle.

   所以，CBD 是一个“黄金”三角形。

How construct a golden triangle or an angle of 36°?
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. We place the fixed points C and D. \tkzDefPoint(0,0){C} and \tkzDefPoint(4,0){D}; 先使用\tkzDefPoint(0,0){C} 和\tkzDefPoint(4,0){D}定义两个已知点 C 和 D。
2. We construct a square CDef and we construct the midpoint m of [Cf]; 构造一个正方形 CDef，并且取 Cf 的中点 m。

   We can do all of this with a compass and a rule; 当然，可以使用圆规和直尺来完成所有这些工作。
3. Then we trace an arc with center m through e. This arc cross the line (Cf) at n; 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. Now the two arcs with center C and D and radius Cn define the point B. 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% from tikz
  \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
  \tkzDrawPolygon(B,...,D)
  \tkzDrawPoints(B,C,D,e,f,m,n)
  \tkzLabelPoints[above](B)
  \tkzLabelPoints[left](f,m,n)
  \tkzLabelPoints(C,D)
  \tkzLabelPoints[right](e)
\end{tikzpicture}
```

After building the golden triangle BCD, we build the point A by noticing that BD = DA. Then we get the point E and finally the point F. This is done with already intersections of defined objects (line and circle).
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。

3.5.3. Part II: two others methods with golden and euclid triangle
第 II 部分：黄金三角和欧几里德三角的另外两种方法

tkz-euclide knows how to define a ”golden” or ”euclide” triangle. We can define BCD and BCA like gold triangles.
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[golden](C,D)
  \tkzGetPoint{B}
  \tkzDefTriangle[golden](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(B,F,C)
  \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}
```

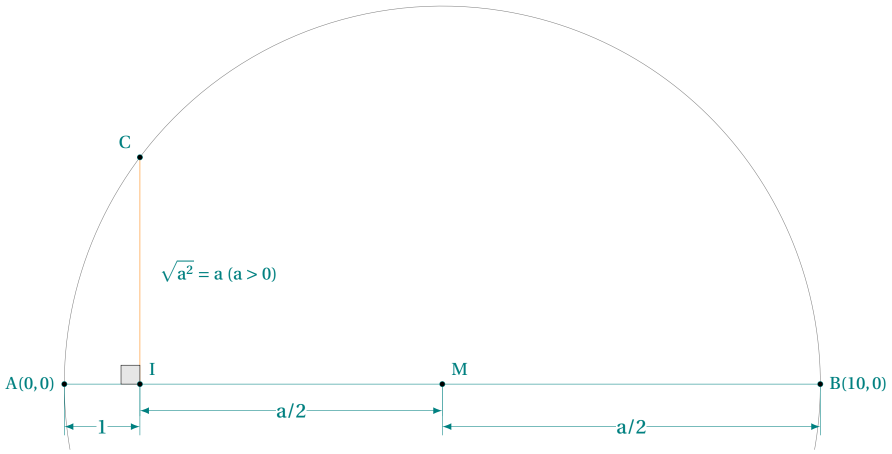
Here is a final method that uses rotations:
也可以使用坐标旋转的方法定义三角形:

```
\begin{tikzpicture}
\tkzDefPoint(0,0){C} % possible
% \tkzDefPoint[label=below:$C$](0,0){C}
% but don't do this
% 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}
% We get D and E with a rotation
% 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}
% To get A we use an intersection of lines
% 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}
% drawing
% 绘制
\tkzDrawArc[delta=10](B,C)(E)
\tkzDrawPolygon(C,B,D)
\tkzDrawSegments(D,A B,A C,E)
% angles
% 角度标记
\tkzMarkAngles(C,B,D E,A,D) %this is to draw the arcs
\tkzLabelAngles[pos=1.5](C,B,D E,A,D){$\alpha$}
\tkzMarkRightAngle(B,H,C)
\tkzDrawPoints(A,...,E)
% Label only now
% 仅实现标注
\tkzLabelPoints[below left](C,A)
```

```
\tkzLabelPoints[below right](D)
\tkzLabelPoints[above](B,E)
\end{tikzpicture}
```

3.5.4. Complete but minimal example
最小工作示例

A unit of length being chosen, the example shows how to obtain a segment of length $\sqrt{a}$ from a segment of length a, using a ruler and a compass.
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



```
\begin{tikzpicture}[scale=1,ra/.style={fill=gray!20}]
   % fixed points
   % 已知点
   \tkzDefPoint(0,0){A}
   \tkzDefPoint(1,0){I}
   % calculation
   % 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)              \tkzGetFirstPoint{C}
   % 绘图
   \tkzDrawSegment[style=orange](I,C)
   \tkzDrawArc(M,B)(A)
   \tkzDrawSegment[dim={$1$,-16pt,}](A,I)
   \tkzDrawSegment[dim={$a/2$,-10pt,}](I,M)
   \tkzDrawSegment[dim={$a/2$,-16pt,}](M,B)
   % 标记
   \tkzMarkRightAngle[ra](A,I,C)
   % 标注
   \tkzDrawPoints(I,A,B,C,M)
   \tkzLabelPoint[left](A){$A(0,0)$}
   \tkzLabelPoints[above right](I,M)
   \tkzLabelPoints[above left](C)
   \tkzLabelPoint[right](B){$B(10,0)$}
   \tkzLabelSegment[right=4pt](I,C){$\sqrt{a^2}=a \ (a>0)$}
\end{tikzpicture}
```

Comments 建议

- The Preamble
- 导言区

  Let us first look at the preamble. If you need it, you have to load **xcolor** before **tkz-euclide**, that is, before Ti*k*Z. Ti*k*Z may cause problems with the active characters, but... provides a library in its latest version that's supposed to solve these problems babel.

  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 Ti*k*Z 宏包之前载入。否则可能会与 Ti*k*Z 宏包冲突，但 Ti*k*Z 宏包提供的 babel 库能解决这些冲突。

```
\documentclass{standalone} % or another class % 或其它文档类
   % \usepackage{xcolor} % before tikz or tkz-euclide if necessary
   % \usepackage{xcolor} % 需要在 tikz 或 tkz-euclide 宏包之前载入
\usepackage{tkz-euclide} % no need to load TikZ % 不再需要显式载入 TikZ 宏包
   % \usetkzobj{all}  is no longer necessary
   % \usetkzobj{all}  不再需要该命令
   % \usetikzlibrary{babel}  if there are problems with the active characters
   % \usetikzlibrary{babel} 如果有冲突，则可以使用该库进行解决
```

  The following code consists of several parts:

  该代码可以分解为以下几个部分：

- Definition of fixed points: the first part includes the definitions of the points necessary for the construction, these are the fixed points. The macros \tkzInit and \tkzClip in most cases are not necessary.
- 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}
```

- The second part is dedicated to the creation of new points from the fixed points; a B point is placed at 10 cm from A. The middle of [AB] is defined by M and then the orthogonal line to the (AB) line is searched for at the I point. Then we look for the intersection of this line with the semi-circle of center M passing through A.
- 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}
```

- The third one includes the different drawings;
- 第 3 部分是图形绘制：

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

– Marking: the fourth is devoted to marking;

– 第 4 部分是绘制标记：

```
\tkzMarkRightAngle[ra](A,I,C)
```

– Labelling: the latter only deals with the placement of labels.

– 最后是布置标注：

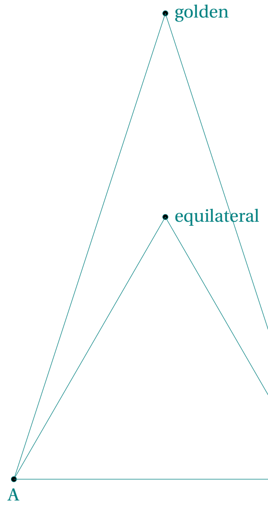
```
\tkzLabelPoint[left](A){$A(0,0)$}
\tkzLabelPoint[right](B){$B(10,0)$}
\tkzLabelSegment[right=4pt](I,C){$\sqrt{a^2}=a \ (a>0)$}
```

4. The Elements of tkz code tkz 的基本要素

To work with my package, you need to have notions of LaTeX as well as Ti*k*Z.
要使用这个宏宏，需要有 LaTeX 和 Ti*k*Z 的概念。
In this paragraph, we start looking at the "rules" and "symbols" used to create a figure with `tkz-euclide`.
接下来，讨论使用 `tkz-euclide` 绘制图形时的“规则”和“符号系统”

4.1. Objects and language 对象和语言

The primitive objects are points. You can refer to a point at any time using the name given when defining it. (it is possible to assign a different name later on).
`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于不同的名称)。
To get new points you will use macros. `tkz-euclide` macros have a name beginning with tkz. There are four main categories starting with: `\tkzDef...` `\tkzDraw...` `\tkzMark...` and `\tkzLabel...`. The used points are passed as parameters between parentheses while the created points are between braces.
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...`。使用的点作为参数在括号之间传递，而创建的点在大括号之间。
The code of the figures is placed in an environment **tikzpicture**
图形的代码放在 **tikzpicture** 环境中
Contrary to Ti*k*Z, you should not end a macro with ";". We thus lose the important notion which is the **path**. However, it is possible to place some code between the macros `tkz-euclide`.
与 Ti*k*Z 相反，宏不应该以“;”结尾。因此，我们失去了 **path** 这个重要的概念。但是，可以在宏 `tkz-euclide` 之间放置一些代码。
Among the first category, `\tkzDefPoint` allows you to define fixed points. It will be studied in detail later. Here we will see in detail the macro `\tkzDefTriangle`.
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详细讨论。在此，首先详细讨论`\tkzDefTriangle` 命令。
This macro makes it possible to associate to a pair of points a third point in order to define a certain triangle `\tkzDefTriangle(A,B)`. The obtained point is referenced `tkzPointResult` and it is possible to choose another reference with `\tkzGetPoint{C}` for example.
该宏可以将一对点与第三个点相关联，以定义某个三角形`\tkzDefTriangle(A,B)`。获得的点被引用为`tkzPointResult`，例如，可以使用`\tkzGetPoint{C}`选择另一个引用。
`\tkzDefTriangle[euclid](A,B) \tkzGetPoint{C}`
Parentheses are used to pass arguments. In `(A,B)` A and B are the points with which a third will be defined. However, in `{C}` we use braces to retrieve the new point.
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。
In order to choose a certain type of triangle among the following choices: `equilateral`, `isosceles right`, `half`, `pythagoras`, `school`, `golden or sublime`, `euclid`, `gold`, `cheops`... and `two angles` you just have to choose between hooks, for example:
三角形类型有：`equilateral`, `isosceles right`, `half`, `pythagoras`, `school`, `golden or sublime`, `euclid`, `gold`, `cheops`... `two angles`，只需要在挂钩之间做出选择，例如：



```
 \begin{tikzpicture}[scale=.5]
 \tkzDefPoints{0/0/A,8/0/B}
\foreach \tr in {golden, equilateral}
 {\tkzDefTriangle[\tr](A,B) \tkzGetPoint{C}
 \tkzDrawPoint(C)
 \tkzLabelPoint[right](C){\tr}
 \tkzDrawSegments(A,C C,B)}
 \tkzDrawPoints(A,B)
 \tkzDrawSegments(A,B)
 \tkzLabelPoints(A,B)
 \end{tikzpicture}
```

4.2. Notations and conventions 符号和约定

I deliberately chose to use the geometric French and personal conventions to describe the geometric objects represented. The objects defined and represented by **tkz-euclide** are points, lines and circles located in a plane. They are the primary objects of Euclidean geometry from which we will construct figures.
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，它们是欧氏几何中构成几何图形主要元素。
According to **Euclid**, these figures will only illustrate pure ideas produced by our brain. Thus a point has no dimension and therefore no real existence. In the same way the line has no width and therefore no existence in the real world. The objects that we are going to consider are only representations of ideal mathematical objects. **tkz-euclide** will follow the steps of the ancient Greeks to obtain geometrical constructions using the ruler and the compass.
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。
Here are the notations that will be used:
以下是使用的基本符号：

- The points are represented geometrically either by a small disc or by the intersection of two lines (two straight lines, a straight line and a circle or two circles). In this case, the point is represented by a cross.
- 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。在这种情况下，该点用十字交叉表示。

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

or else

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

The existence of a point being established, we can give it a label which will be a capital letter (with some exceptions) of the Latin alphabet such as A, B or C. For example:

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：【其实本说明中，作者有时用小写字母表示点。但不建议这样做。】

- O is a center for a circle, a rotation, etc.;O 是圆、旋转等的中心。;
- M defined a midpoint;M 定义了一个中点；
- H defined the foot of an altitude;H 定义了垂足；
- P′ is the image of P by a transformation ;P′ 是通过变换得到的 P 的对应点；

It is important to note that the reference name of a point in the code may be different from the label to designate it in the text. So we can define a point A and give it as label P. In particular the style will be different, point A will be labeled A.

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。

```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDrawPoints(A)
  \tkzLabelPoint(A){$P$}
\end{tikzpicture}
```

Exceptions: some points such as the middle of the sides of a triangle share a characteristic, so it is normal that their names also share a common character. We will designate these points by M_a, M_b and M_c or M_A, M_B and M_C.

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。

In the code, these points will be referred to as: M_A, M_B and 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。

Another exception relates to intermediate construction points which will not be labelled. They will often be designated by a lowercase letter in the code.

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。

- The line segments are designated by two points representing their ends in square brackets: [AB].
- 线段使用方括号中的两个点表示，如：[AB]。
- The straight lines are in Euclidean geometry defined by two points so A and B define the straight line (AB). We can also designate this stright line using the Greek alphabet and name it (δ) or (Δ). It is also possible to designate the straight line with lowercase letters such as d and d′.
- 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′。
- The semi-straight line is designated as follows [AB).
- 射线可表示为 [AB).
- Relation between the straight lines. Two perpendicular (AB) and (CD) lines will be written (AB) $\perp$ (CD) and if they are parallel we will write (AB) $/\!/$ (CD).
- 两直线间的关系：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) $/\!/$ (CD)。
- The lengths of the sides of triangle ABC are AB, AC and BC. The numbers are also designated by a lowercase letter so we will write: AB = c, AC = b and BC = a. The letter a is also used to represent an angle, and r is frequently used to represent a radius, d a diameter, l a length, d a distance.

- 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。
- Polygons are designated afterwards by their vertices so ABC is a triangle, EFGH a quadrilateral.
- 多边形用其顶点表示，如三角形表示为 ABC，四边形表示为 EFGH。
- Angles are generally measured in degrees (ex 60°) and in an equilateral ABC triangle we will write $\widehat{ABC} = \widehat{B} = 60°$.
- 角度的单位是度 (例如：60°)，对于等边三角形 ABC，可以表示为 $\widehat{ABC} = \widehat{B} = 60°$。
- The arcs are designated by their extremities. For example if A and B are two points of the same circle then $\widehat{AB}$.
- 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。
- Circles are noted either $\mathscr{C}$ if there is no possible confusion or $\mathscr{C}$ (O ; A) for a circle with center O and passing through the point A or $\mathscr{C}$ (O ; 1) for a circle with center O and radius 1 cm.
- 如果没有岐义，一个圆可以表示为 $\mathscr{C}$，或用 $\mathscr{C}$ (O ; A) 表示圆心在 O 点并通过点 A 的圆或用 $\mathscr{C}$ (O ; 1) 表示圆心在点 O 半径为 1 cm 的圆。
- Name of the particular lines of a triangle: I used the terms bisector, bisector out, mediator (sometimes called perpendicular bisectors), altitude, median and symmedian.
- 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) coordinates of the point A_1, (x_A,y_A) coordinates of the point A.
- (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。

4.3. Set, Calculate, Draw, Mark, Label 点集、计算、绘制、标记和标注

The title could have been: `Separation of Calculus and Drawings`
该标题的含义是：计算与绘制分离
When a document is prepared using the LaTeX system, the source code of the document can be divided into two parts: the document body and the preamble. Under this methodology, publications can be structured, styled and typeset with minimal effort. I propose a similar methodology for creating figures with `tkz-euclide`.
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。
The first part defines the fixed points, the second part allows the creation of new points. `Set and Calculate` are the two main parts. All that is left to do is to draw (or fill), mark and label. It is possible that `tkz-euclide` is insufficient for some of these latter actions but you can use Ti*k*Z
首先定义已知点，然后计算其他点。`Set and Calculate` 是两个主要部分。剩下要做的就是绘制 (或者填充)，标记，标注。对于后面的一些操作，`tkz-euclide` 功能可能不够用，但是可以使用 Ti*k*Z。
One last remark that I think is important, it is best to avoid introducing coordinates within a code as much as possible. I think that the coordinates should appear at the beginning of the code with the fixed points. Then the use of references is recommended. Most macros have the option `nodes` or `with nodes`.
我认为重要的最后一点是，最好尽可能避免在代码中引入坐标。我认为已知点的坐标应该出现在代码的开头。推荐使用参考资料：大多数宏都有选项 `nodes` 或 `with nodes`。
I also think it’s best to define the styles of the different objects from the beginning.
我还认为最好从一开始就确定不同对象的风格。

5. About this documentation and the examples 关于本文档和实例

It is obtained by compiling with ”lualatex”. I use a class `doc.cls` based on `scrartcl`.
它是通过用 ”lualatex” 编译得到的。使用了一个基于 `doc.cls` 的类，即 `scrartcl`。
Below the list of styles used in the docuimentation. To understand how to use the styles see the section 37
下面是本文件中使用的样式列表。要了解如何使用这些样式，请参见 37 部分。

```
\tkzSetUpColors[background=white,text=black]
\tkzSetUpCompass[color=orange, line width=.2pt,delta=10]
\tkzSetUpArc[color=gray,line width=.2pt]
\tkzSetUpPoint[size=2,color=teal]
\tkzSetUpLine[line width=.2pt,color=teal]
\tkzSetUpStyle[color=orange,line width=.2pt]{new}
\tikzset{every picture/.style={line width=.2pt}}
\tikzset{label angle style/.append style={color=teal,font=\footnotesize}}
\tikzset{label style/.append style={below,color=teal,font=\scriptsize}}
```

Some examples use predefined styles like
一些例子使用预定义的样式，如：

```
\tikzset{new/.style={color=orange,line width=.2pt}}
```

Part II.

Setting 设置

6. First step: fixed points 第一步：已知点

The first step in a geometric construction is to define the fixed points from which the figure will be constructed.
几何绘图的第一步是定义已知点，从这些已知点开始绘制图形。
The general idea is to avoid manipulating coordinates and to prefer to use the references of the points fixed in the first step or obtained using the tools provided by the package. Even if it's possible, I think it's a bad idea to work directly with coordinates. Preferable is to use named points.
一般的想法是避免操作坐标，而倾向于使用第一步中已知点的坐标，或者使用软件包提供的工具获得。即使有可能，我也认为直接用坐标工作也不是个好主意。最好的办法是使用命名的点。
`tkz-euclide` uses macros and vocabulary specific to geometric construction. It is of course possible to use the tools of Ti*k*Z but it seems more logical to me not to mix the different syntaxes.
`tkznameofpack` 使用的是几何结构特有的宏和词汇。当然也可以使用 Ti*k*Z 的工具，但对我来说，不混合不同的语法似乎更符合逻辑。
A point in `tkz-euclide` is a particular ”node” for Ti*k*Z. In the next section we will see how to define points using coordinates. The style of the points (color and shape) will not be discussed. You will find some indications in some examples; for more information you can read the following section 37.
`tkz-euclide` 中的一个点是 Ti*k*Z 的一个特定 ”node 节点”。在下一节，我们将看到如何用坐标来定义点。点的样式（颜色和形状）将不会被讨论。你会在一些例子中发现一些指示；更多信息你可以阅读下面的章节：37。

7. Definition of a point : \tkzDefPoint or \tkzDefPoints 定义一个点

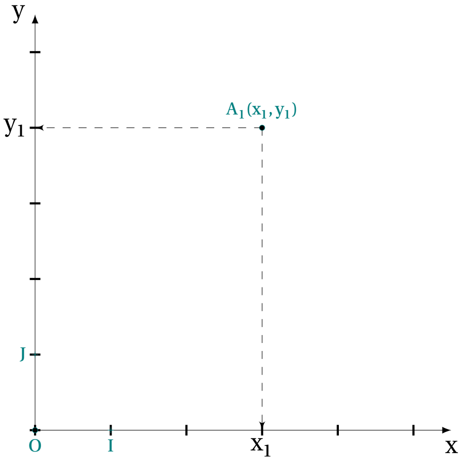
Points can be specified in any of the following ways:
可以通过如下方式定义一个点:

- Cartesian coordinates; 笛卡尔坐标;
- Polar coordinates; 极坐标;
- Named points; 直接命名一个点;
- Relative points. 相对位置.

A point is defined if it has a name linked to a unique pair of decimal numbers. Let (x,y) or (a:d) i.e. (x abscissa, y ordinate) or (a angle: d distance). This is possible because the plan has been provided with an orthonormed Cartesian coordinate system. The working axes are (ortho)normed with unity equal to 1 cm.
如果一个点有一个与一对唯一的十进制数字相联系的名称，那么这个点就被定义了。让 (x,y) 或 (a:d) 即 (x 横坐标，y 纵横坐标) 或 (a 角度：d 距离)。这种方法已经提供了一个正交的直角坐标系。坐标轴是（正交）规范的，其单位长度等于 1 cm。
The Cartesian coordinate (a,b) refers to the point a centimeters in the x-direction and b centimeters in the y-direction.
笛卡尔坐标 (a，b) 指的是在 x 轴方向上 a 厘米处和在 y 轴方向上 b 厘米处的点。
A point in polar coordinates requires an angle α, in degrees, and a distance d from the origin with a dimensional unit by default it's the `cm`.
极坐标中的点，需要以度为单位的角度 α，以及距原点的距离 d，默认情况下，其尺寸单位为 `cm`。
The `\tkzDefPoint` macro is used to define a point by assigning coordinates to it. This macro is based on `\coordinate`, a macro of Ti*k*Z. It can use Ti*k*Z-specific options such as `shift`. If calculations are required then the `xfp` package is chosen. We can use Cartesian or polar coordinates.
`\tkzDefPoint` 宏用于通过给点分配坐标来定义点。此宏基于`\coordinate`，Ti*k*Z 的宏。它可以使用特定于 Ti*k*Z 的选项，如 `shift`。如果需要计算，则选择 `xfp` 包。我们可以使用笛卡尔坐标或极坐标。

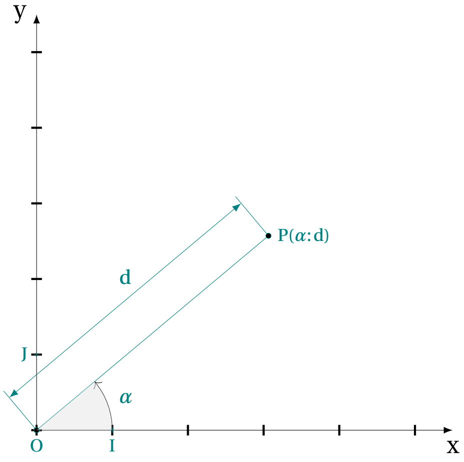
Cartesian coordinates 直角坐标

```
\begin{tikzpicture}[scale=1]
  \tkzInit[xmax=5,ymax=5]
  % necessary to limit
  % the size of the axes
  \tkzDrawX[>=latex]
  \tkzDrawY[>=latex]
  \tkzDefPoints{0/0/O,1/0/I,0/1/J}
  \tkzDefPoint(3,4){A}
  \tkzDrawPoints(O,A)
  \tkzLabelPoint[above](A){$A_1 (x_1,y_1)$}
  \tkzShowPointCoord[xlabel=$x_1$,
                     ylabel=$y_1$](A)
  \tkzLabelPoints(O,I)
  \tkzLabelPoints[left](J)
  \tkzDrawPoints[shape=cross](I,J)
\end{tikzpicture}
```



Polar coordinates 极坐标

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



7.1. Defining a named point \tkzDefPoint 定义命名点

\tkzDefPoint[⟨local options⟩](⟨x,y⟩){⟨ref⟩} or (⟨α:d⟩){⟨ref⟩}

arguments	default	definition
(x,y)	no default	x and y are two dimensions, by default in cm.
(α:d)	no default	α is an angle in degrees, d is a dimension
{ref}	no default	Reference assigned to the point: A, T_a ,P1 or P_1

The obligatory arguments of this macro are two dimensions expressed with decimals, in the first case they are two measures of length, in the second case they are a measure of length and the measure of an angle in degrees. Do not confuse the reference with the name of a point. The reference is used by calculations, but frequently, the name is identical to the reference.

options	default	definition
label	no default	allows you to place a label at a predefined distance
shift	no default	adds (x,y) or (α:d) to all coordinates

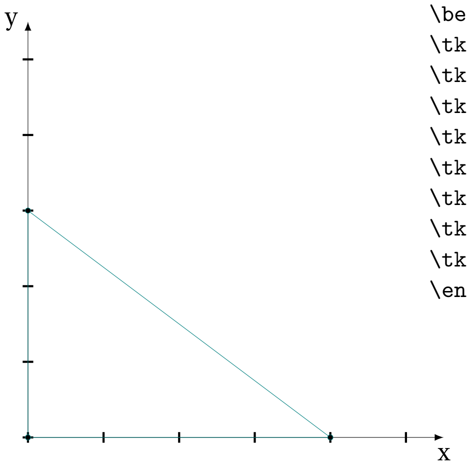
`\tkzDefPoint[`⟨命令选项⟩`](`⟨x,y⟩`){`⟨名称⟩`}` **or** `(`⟨α:d⟩`){`⟨名称⟩`}`

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.
(α: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。不要将引用与点的名称混淆。该引用由计算得到，但通常名称与引用相同。

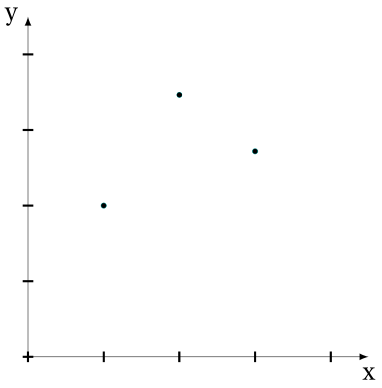
选项	默认值	含义
label	无	按预设的距离添加标注
shift	无	为 (x,y) 或 (α:d) 添加坐标偏移

7.1.1. Cartesian coordinates 笛卡尔坐标示例



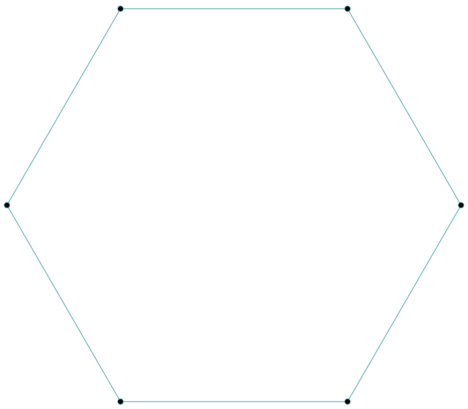
```
\begin{tikzpicture}
\tkzInit[xmax=5,ymax=5] % limits the size of the axes
\tkzDrawX[>=latex]
\tkzDrawY[>=latex]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(0,3){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

7.1.2. Calculations with xfp 使用 xfp 宏包实现计算



```
\begin{tikzpicture}[scale=1]
  \tkzInit[xmax=4,ymax=4]
  \tkzDrawX\tkzDrawY
  \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(O,B,A)
\end{tikzpicture}
```

7.1.3. Polar coordinates 极坐标示例

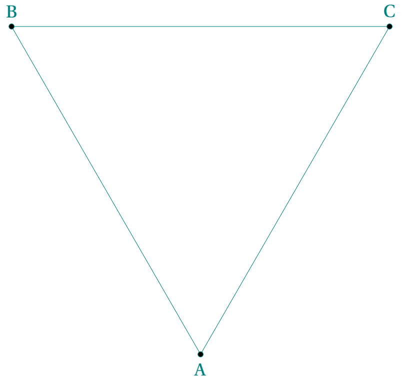


```
\begin{tikzpicture}
\foreach \an [count=\i] in {0,60,...,300}
 { \tkzDefPoint(\an:3){A_\i}}
\tkzDrawPolygon(A_1,A_...,A_6)
\tkzDrawPoints(A_1,A_...,A_6)
\end{tikzpicture}
```

7.1.4. Relative points 相对位置点（依赖点）

First, we can use the scope environment from Ti*k*Z. In the following example, we have a way to define an equilateral triangle.

首先，可以从 Ti*k*Z 使用 scope 环境。在下面的例子中，有一个定义等边三角形的方法。



```
\begin{tikzpicture}[scale=1]
 \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}
```

7.2. Point relative to another: \tkzDefShiftPoint 定义平移点

\tkzDefShiftPoint[⟨Point⟩](⟨x,y⟩){⟨ref⟩} or (⟨α:d⟩){⟨ref⟩}

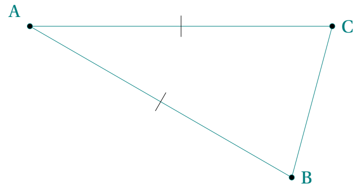
arguments	default	definition
(x,y)	no default	x and y are two dimensions, by default in cm.
(α:d)	no default	α is an angle in degrees, d is a dimension
{ref}	no default	Reference assigned to the point: A, T_a ,P1 or P_1
options	**default**	**definition**
[pt]	no default	\tkzDefShiftPoint[A](0:4){B}

\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.
(α:d)	无	α 是角度（度），d 是距离
选项	**默认值**	**含义**
[参考点]	无	例如：\tkzDefShiftPoint[A](0:4){B}

7.2.1. Isosceles triangle 等腰三角形

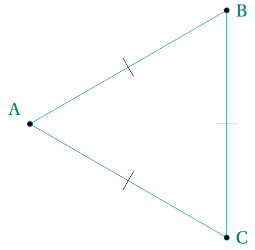
This macro allows you to place one point relative to another. This is equivalent to a translation. Here is how to construct an isosceles triangle with main vertex A and angle at vertex of 30°.
该命令允许基于一个点定义另一个点，等价于点的平移。下面的代码给出了一种通过点 A 和 30° 顶角等腰三角形的构造方法。



```
\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=|](A,B A,C)
 \tkzDrawPoints(A,B,C)
 \tkzLabelPoints[right](B,C)
 \tkzLabelPoints[above left](A)
\end{tikzpicture}
```

7.2.2. Equilateral triangle 构造等边三角形

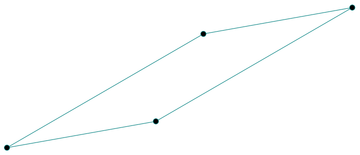
Let’s see how to get an equilateral triangle (there is much simpler)
下面的代码给出了一种极为简捷的等边三角形构造方法。



```
\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[right](B,C)
 \tkzLabelPoints[above left](A)
 \tkzMarkSegments[mark=|](A,B A,C B,C)
\end{tikzpicture}
```

7.2.3. Parallelogram 命令构造平行四边形

There’s a simpler way
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}
```

7.3. Definition of multiple points: \tkzDefPoints 定义点集

```
\tkzDefPoints[⟨local options⟩]{⟨x1/y1/n1,x2/y2/r2, ...⟩}
```

x_i and y_i are the coordinates of a referenced point r_i

arguments	default	example
$x_i/y_i/r_i$		\tkzDefPoints{0/0/O,2/2/A}

options	default	definition
shift	no default	Adds (x,y) or (α:d) to all coordinates

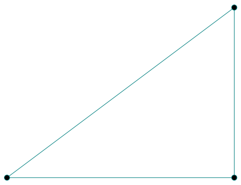
`\tkzDefPoints[⟨命令选项⟩]{⟨x_1/y_1/r_1,x_2/y_2/r_2,...⟩}`

x_i 和 y_i 是 r_i 点的 2 维坐标

参数	默认值	样例
$x_i/y_i/r_i$	无	\tkzDefPoints{0/0/O,2/2/A}

选项	默认值	含义
shift	无	为所有点添加 (x,y) 或 (α:d) 坐标偏移

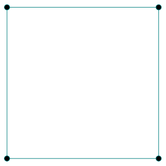
7.4. Create a triangle 构造三角形



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoints{0/0/A,4/0/B,4/3/C}
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

7.5. Create a square 构造正方形

Note here the syntax for drawing the polygon.
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,...,D)
\end{tikzpicture}
```

Part III.

Calculating 计算

Now that the fixed points are defined, we can with their references using macros from the package or macros that you will create get new points. The calculations may not be apparent but they are usually done by the package. You may need to use some mathematical constants, here is the list of constants defined by the package. You may need to use some mathematical constants, here is the list of constants defined by the package.
既然已经定义了已知点，就可以使用程序包中的宏或将要创建的宏来引用它们，从而获得新的点。这些计算可能不明显，通常可由软件包完成。可能需要使用一些数学常数，这里是由软件包定义的常数列表。【你可能需要使用一些数学常数，这里是由软件包定义的常数列表。此句原文重复】

8. Auxiliary tools 辅助工具

8.1. Constants 常量

tkz-euclide knows some constants, here is the list:
tkznameofpack 里一些常数（常量）列表：

```
\def\tkzPhi{1.618034}
\def\tkzInvPhi{0.618034}
\def\tkzSqrtPhi{1.27202}
\def\tkzSqrTwo{1.414213}
\def\tkzSqrThree{1.7320508}
\def\tkzSqrFive{2.2360679}
\def\tkzSqrTwobyTwo{0.7071065}
\def\tkzPi{3.1415926}
\def\tkzEuler{2.71828182}
```

8.2. New point by calculation 通过计算新建点

When a macro of **tkznameofpack** creates a new point, it is stored internally with the reference **tkzPointResult**. You can assign your own reference to it. This is done with the macro **\tkzGetPoint**. A new reference is created, your choice of reference must be placed between braces.
当 **tkznameofpack** 宏创建一个新点时，它与引用 **tkzPointResult** 一起存储在内部。可以给它分配自己的索引。这是通过宏**\tkzGetPoint** 完成的。创建了一个新的引用，选择的索引必须放在大括号中。

\tkzGetPoint{⟨ref⟩}

If the result is in **tkzPointResult**, you can access it with **\tkzGetPoint**.

arguments	default	example
ref	no default	**\tkzGetPoint{M}** see the next example

\tkzGetPoint{⟨ref⟩}

结果在 **tkzPointResult** 中，可以使用**\tkzGetPoint** 来访问它。

参数	默认值	示例
ref	no default	**\tkzGetPoint{M}** 请参见下一个示例

Sometimes you need to get two points. It's possible with
有时候需要得到两点。方法

\tkzGetPoints{⟨ref1⟩}{⟨ref2⟩}

The result is in `tkzPointFirstResult` and `tkzPointSecondResult`.

arguments	default	example
{ref1,ref2}	no default	\tkzGetPoints{M,N} It's the case with \tkzInterCC

\tkzGetPoints{⟨ref1⟩}{⟨ref2⟩}

结果在 `tkzPointFirstResult` 和 `tkzPointSecondResult` 中。

参数	默认值	示例
{ref1,ref2}	无	\tkzGetPoints{M,N} 这就是\tkzInterCC 的情况

If you need only the first or the second point you can also use :
如果只需要第一点或第二点，也可以使用:

\tkzGetFirstPoint{⟨ref1⟩}

arguments	default	example
ref1	no default	\tkzGetFirstPoint{M}

\tkzGetFirstPoint{⟨ref1⟩}

参数	默认值	示例
ref1	无	\tkzGetFirstPoint{M}

\tkzGetSecondPoint{⟨ref2⟩}

arguments	default	example
ref2	no default	\tkzGetSecondPoint{M}

\tkzGetSecondPoint{⟨ref2⟩}

参数	默认值	示例
ref2	无	\tkzGetSecondPoint{M}

Sometimes the results consist of a point and a dimension. You get the point with `\tkzGetPoint` and the dimension with `\tkzGetLength`.
有时结果由一个点和一个尺寸组成。使用`\tkzGetPoint` 获取点，使用`\tkzGetLength` 获取尺寸。

`\tkzGetLength{⟨name of a macro⟩}`

arguments	default	example
name of a macro	no default	\tkzGetLength{rAB} \rAB gives the length in cm

`\tkzGetLength{⟨name of a macro⟩}`

参数	默认值	示例
宏名	无	\tkzGetLength{rAB} \rAB 给出以厘米为单位的长度

9. Special points 特殊点

Here are some special points.
以下是一些特殊点。

9.1. Middle of a segment \tkzDefMidPoint 定义线段中点

It is a question of determining the middle of a segment.
确定线段中点的示例。

`\tkzDefMidPoint(⟨pt1,pt2⟩)`

The result is in `tkzPointResult`. We can access it with `\tkzGetPoint`.

arguments	default	definition
(pt1,pt2)	no default	pt1 and pt2 are two points

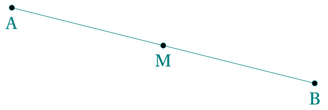
`\tkzDefMidPoint(⟨pt1,pt2⟩)`

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

9.1.1. Use of \tkzDefMidPoint 命令示例

Review the use of `\tkzDefPoint`.
`\tkzDefPoint` 的用法。



```
\begin{tikzpicture}[scale=1]
 \tkzDefPoint(2,3){A}
 \tkzDefPoint(6,2){B}
 \tkzDefMidPoint(A,B)
 \tkzGetPoint{M}
 \tkzDrawSegment(A,B)
 \tkzDrawPoints(A,B,M)
 \tkzLabelPoints[below](A,B,M)
\end{tikzpicture}
```

9.2. Golden ratio \tkzDefGoldenRatio 黄金比例

From Wikipedia : In mathematics, two quantities are in the golden ratio if their ratio is the same as the ratio of their sum to the larger of the two quantities. Expressed algebraically, for quantities a, b such as a > b > 0; a + b is to a as a is to b.
来自维基百科: 在数学中，如果两个量之比等于它们之和与两个量中较大者之比，那么这两个量就是黄金比例。用代数表示，对于数 a，b，a > b > 0；a + b 对 a 的比率与 a 对 b 的比率相同。
$\frac{a+b}{a} = \frac{a}{b} = \phi = \frac{1+\sqrt{5}}{2}$
One of the two solutions to the equation $x^2 - x - 1 = 0$ is the golden ratio ϕ, $\phi = \frac{1+\sqrt{5}}{2}$.
方程 $x^2 - x - 1 = 0$ 的两个解之一是黄金比例 ϕ, $\phi = \frac{1+\sqrt{5}}{2}$。

`\tkzDefGoldenRatio(⟨pt1,pt2⟩)`

arguments	default	example
(pt1,pt2)	no default	\tkzDefGoldenRatio(A,C) \tkzGetPoint{B}

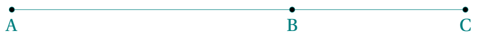
AB = a, BC = b and $\frac{AC}{AB} = \frac{AB}{BC} = \phi$

`\tkzDefGoldenRatio(⟨pt1,pt2⟩)`

参数	默认值	含义
(pt1,pt2)	无	\tkzDefGoldenRatio(A,C) \tkzGetPoint{B}

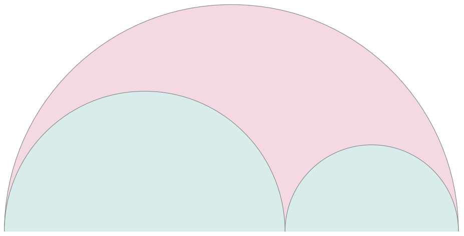
AB = a, BC = b and $\frac{AC}{AB} = \frac{AB}{BC} = \phi$

9.2.1. Use the golden ratio to divide a line segment 使用黄金分割比例来分割线段



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,6/0/C}
 \tkzDefMidPoint(A,C) \tkzGetPoint{I}
 %\tkzDefPointWith[linear,K=\tkzInvPhi](A,C)
 \tkzDefGoldenRatio(A,C) \tkzGetPoint{B}
 \tkzDrawSegments(A,C)
 \tkzDrawPoints(A,B,C)
 \tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

9.2.2. Golden arbelos 黄金阿贝罗



```
\begin{tikzpicture}[scale=.6]
\tkzDefPoints{0/0/A,10/0/B}
\tkzDefGoldenRatio(A,B)     \tkzGetPoint{C}
\tkzDefMidPoint(A,B)        \tkzGetPoint{O_1}
\tkzDefMidPoint(A,C)        \tkzGetPoint{O_2}
\tkzDefMidPoint(C,B)        \tkzGetPoint{O_3}
\tkzDrawSemiCircles[fill=purple!15](O_1,B)
\tkzDrawSemiCircles[fill=teal!15](O_2,C O_3,B)
\end{tikzpicture}
```

It is also possible to use the following macro.
也可以使用以下宏。

9.3. Barycentric coordinates with \tkzDefBarycentricPoint 重心坐标

pt_1, pt_2, ..., pt_n being n points, they define n vectors $\overrightarrow{v_1}$, $\overrightarrow{v_2}$, ..., $\overrightarrow{v_n}$ with the origin of the referential as the common endpoint. α_1, α_2, ...α_n are n numbers, the vector obtained by:

$$\frac{\alpha_1\overrightarrow{v_1}+\alpha_2\overrightarrow{v_2}+\cdots+\alpha_n\overrightarrow{v_n}}{\alpha_1+\alpha_2+\cdots+\alpha_n}$$

defines a single point.
设共有 pt_1, pt_2, ..., pt_n n 个点，则它们定义了 n 个向量 $\overrightarrow{v_1}$, $\overrightarrow{v_2}$, ..., $\overrightarrow{v_n}$。令 α_1,α_2,...α_n 是 n 常数, 因此可按下式得到一个新向量：

$$\frac{\alpha_1\overrightarrow{v_1}+\alpha_2\overrightarrow{v_2}+\cdots+\alpha_n\overrightarrow{v_n}}{\alpha_1+\alpha_2+\cdots+\alpha_n}$$

\tkzDefBarycentricPoint(⟨pt1=α_1,pt2=α_2,...⟩)

arguments	default	definition
(pt1=α_1,pt2=α_2,...)	no default	Each point has a assigned weight

You need at least two points. Result in `tkzPointResult`.

\tkzDefBarycentricPoint(⟨pt1=α_1,pt2=α_2,...⟩)

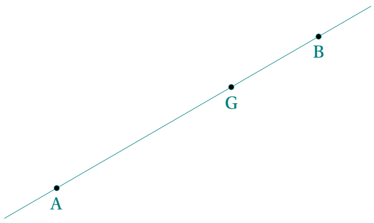
参数	默认值	含义
(pt1=α_1,pt2=α_2,...)	无	每个点的权重

注意：至少需要两个已知点，才能实现计算。

9.3.1. with two points 计算 2 个点的重心

In the following example, we obtain the barycenter of points A and B with coefficients 1 and 2, in other words:
下面的代码中，通过系数 1 和 2 得到了点 A 和点 B 的重心：

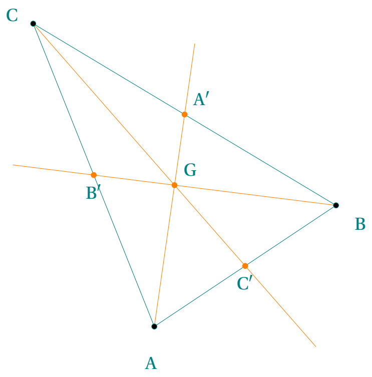
$$\overrightarrow{AI}=\frac{2}{3}\overrightarrow{AB}$$



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

9.3.2. with three points 计算 3 个点的重心

This time M is simply the center of gravity of the triangle.
下面的代码中，M 是三角形的重心。
For reasons of simplification and homogeneity, there is also `\tkzCentroid`.
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]
  \tkzDefPoints{2/1/A,5/3/B,0/6/C}
  \tkzDefBarycentricPoint(A=1,B=1,C=1)
  \tkzGetPoint{G}
  \tkzDefMidPoint(A,B)  \tkzGetPoint{C'}
  \tkzDefMidPoint(A,C)  \tkzGetPoint{B'}
  \tkzDefMidPoint(C,B)  \tkzGetPoint{A'}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawLines[add=0 and 1,new](A,G B,G C,G)
  \tkzDrawPoints[new](A',B',C',G)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoint[above right](G){$G$}
  \tkzAutoLabelPoints[center=G](A,B,C)
  \tkzLabelPoints[above right](A')
  \tkzLabelPoints[below](B',C')
\end{tikzpicture}
```

9.4. Internal and external Similitude Center 相似中心

The centers of the two homotheties in which two circles correspond are called external and internal centers of similitude. You can use `\tkzDefIntSimilitudeCenter` and `\tkzDefExtSimilitudeCenter` but the next macro is better.
两个圆对应的两个同调的中心叫做外相似中心和内相似中心。你可以用`\tkzDefIntSimilitudeCenter` 和 `\tkzDefExtSimilitudeCenter`，但是下一个宏更好。

```
\tkzDefSimilitudeCenter[⟨options⟩](⟨O,A⟩)(⟨O',B⟩)
```

arguments	example	explanation
(⟨pt1,pt2⟩)(⟨pt3,pt4⟩)	$(O,A)(O',B)$	$r = OA, r' = O'B$

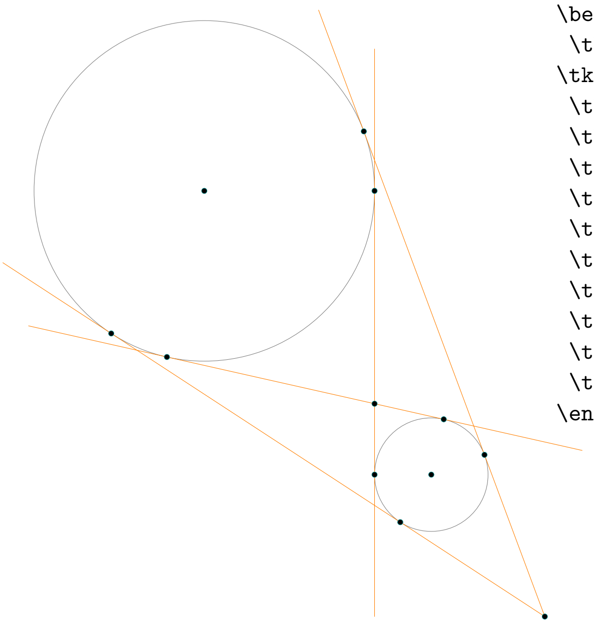
options	default	definition
ext	ext	external center
int	ext	internal center

\tkzDefSimilitudeCenter[⟨options⟩](⟨O,A⟩)(⟨O',B⟩)

参数	默认值	含义
(⟨pt1,pt2⟩)(⟨pt3,pt4⟩)	(O,A)(O′,B)	r = OA, r′ = O′B

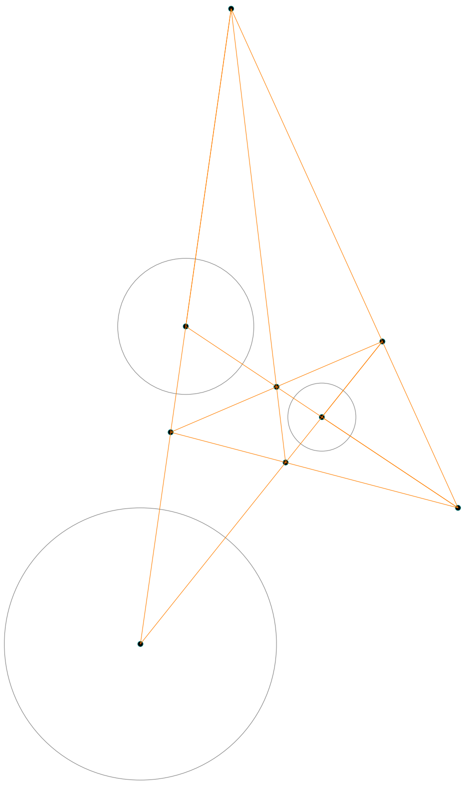
options	default	definition
ext	ext	外相似中心
int	ext	内相似中心

9.4.1. Internal and external with node



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoints{0/0/O,4/-5/A,3/0/B,5/-5/C}
\tkzDefSimilitudeCenter[int](O,B)(A,C)     \tkzGetPoint{I}
 \tkzDefSimilitudeCenter[ext](O,B)(A,C)    \tkzGetPoint{J}
 \tkzDefLine[tangent from = I](O,B)        \tkzGetPoints{D}{E}
 \tkzDefLine[tangent from = I](A,C)        \tkzGetPoints{D'}{E'}
 \tkzDefLine[tangent from = J](O,B)        \tkzGetPoints{F}{G}
 \tkzDefLine[tangent from = J](A,C)
 \tkzGetPoints{F'}{G'}
 \tkzDrawCircles(O,B A,C)
 \tkzDrawSegments[add = .5 and .5,new](D,D' E,E')
 \tkzDrawSegments[add= 0 and 0.25,new](J,F J,G)
 \tkzDrawPoints(O,A,I,J,D,E,F,G,D',E',F',G')
\end{tikzpicture}
```

9.4.2. D'Alembert Theorem 达朗贝尔定理

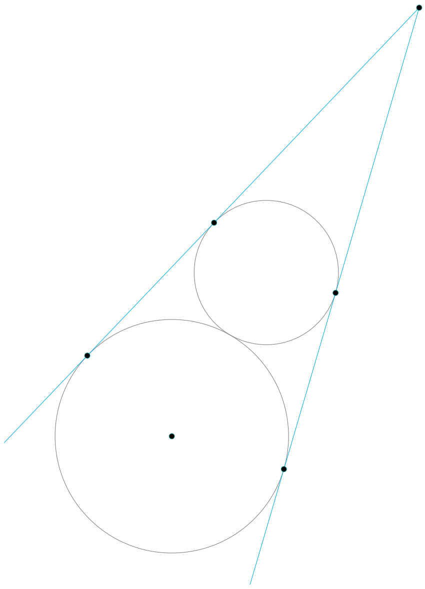


```
\begin{tikzpicture}[scale=.6,rotate=90]
\tkzDefPoints{0/0/A,3/0/a,7/-1/B,5.5/-1/b}
\tkzDefPoints{5/-4/C,4.25/-4/c}
\tkzDrawCircles(A,a B,b C,c)
\tkzDefExtSimilitudeCenter(A,a)(B,b) \tkzGetPoint{I}
\tkzDefExtSimilitudeCenter(A,a)(C,c) \tkzGetPoint{J}
\tkzDefExtSimilitudeCenter(C,c)(B,b) \tkzGetPoint{K}
\tkzDefIntSimilitudeCenter(A,a)(B,b) \tkzGetPoint{I'}
\tkzDefIntSimilitudeCenter(A,a)(C,c) \tkzGetPoint{J'}
\tkzDefIntSimilitudeCenter(C,c)(B,b) \tkzGetPoint{K'}
\tkzDrawPoints(A,B,C,I,J,K,I',J',K')
\tkzDrawSegments[new](I,K A,I A,J B,I B,K C,J C,K)
\tkzDrawSegments[new](I,J' I',J I',K)
\end{tikzpicture}
```

You can use `\tkzDefBarycentricPoint` to find a homothetic center

```
\tkzDefBarycentricPoint(O=\r,A=\R)      \tkzGetPoint{I}
\tkzDefBarycentricPoint(O={-\r},A=\R)   \tkzGetPoint{J}
```

9.4.3. Example with node



```
\begin{tikzpicture}[rotate=60,scale=.5]
 \tkzDefPoints{0/0/A,5/0/C}
 \tkzDefGoldenRatio(A,C) \tkzGetPoint{B}
 \tkzDefSimilitudeCenter(A,B)(C,B) \tkzGetPoint{J}
 \tkzDefTangent[from = J](A,B)   \tkzGetPoints{F}{G}
 \tkzDefTangent[from = J](C,B)    \tkzGetPoints{F'}{G'}
 \tkzDrawCircles(A,B C,B)
 \tkzDrawSegments[add= 0 and 0.25,cyan](J,F J,G)
 \tkzDrawPoints(A,J,F,G,F',G')
\end{tikzpicture}
```

9.5. Harmonic division with \tkzDefHarmonic 调和分割

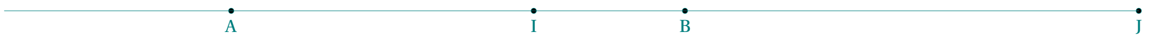
```
\tkzDefHarmonic[⟨options⟩](⟨pt1,pt2,pt3⟩) or (⟨pt1,pt2,k⟩)
```

options	default	definition
both	both	(⟨A,B,2⟩) we look for C and D such that $(A,B;C,D)=-1$ and CA=2CB
ext	both	(⟨A,B,C⟩) we look for D such that $(A,B;C,D)=-1$
int	both	(⟨A,B,D⟩) we look for C such that $(A,B;C,D)=-1$

```
\tkzDefHarmonic[⟨options⟩](⟨pt1,pt2,pt3⟩) or (⟨pt1,pt2,k⟩)
```

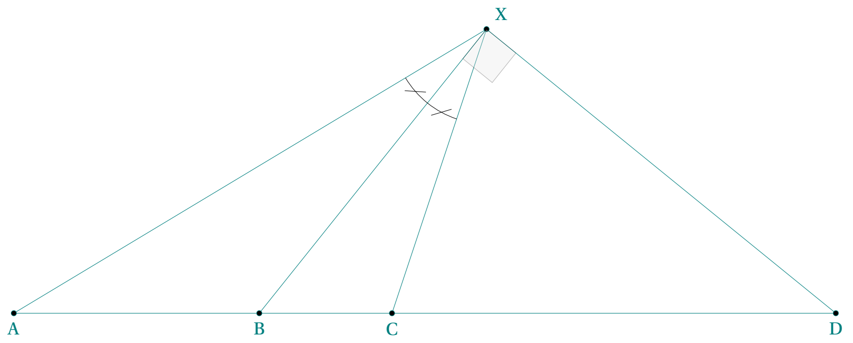
参数	默认值	含义
both	both	(⟨A,B,2⟩) 点 C 和点 D : $(A,B;C,D)=-1$ 及 CA=2CB
ext	both	(⟨A,B,C⟩) 点 D : $(A,B;C,D)=-1$
int	both	(⟨A,B,D⟩) 点 C : $(A,B;C,D)=-1$

9.5.1. options ext and int



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,6/0/B,4/0/C}
\tkzDefHarmonic[ext](A,B,C) \tkzGetPoint{J}
\tkzDefHarmonic[int](A,B,J) \tkzGetPoint{I}
\tkzDrawPoints(A,B,I,J)
\tkzDrawLine[add=.5 and 1](A,B)
\tkzLabelPoints(A,B,I,J)
\end{tikzpicture}
```

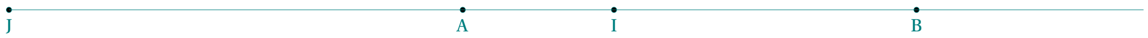
9.5.2. Bisector and harmonic division 二分法和调和分割



```
\begin{tikzpicture}[scale=1.25]
\tkzDefPoints{0/0/A,4/0/C,5/3/X}
\tkzDefLine[bisector](A,X,C) \tkzGetPoint{x}
\tkzInterLL(X,x)(A,C)        \tkzGetPoint{B}
\tkzDefHarmonic[ext](A,C,B)  \tkzGetPoint{D}
\tkzDrawPolygon(A,X,C)
\tkzDrawSegments(X,B C,D D,X)
\tkzDrawPoints(A,B,C,D,X)
\tkzMarkAngles[mark=s|](A,X,B B,X,C)
\tkzMarkRightAngle[size=.4,
                   fill=gray!20,
                   opacity=.3](B,X,D)
\tkzLabelPoints(A,B,C,D)
\tkzLabelPoints[above right](X)
\end{tikzpicture}
```

9.5.3. option both

both is the default option
both 是默认参数



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,6/0/B}
 \tkzDefHarmonic(A,B,{1/2})\tkzGetPoints{I}{J}
 \tkzDrawPoints(A,B,I,J)
 \tkzDrawLine[add=1 and .5](A,B)
 \tkzLabelPoints(A,B,I,J)
\end{tikzpicture}
```

9.6. Equidistant points with \tkzDefEquiPoints 等距点

\tkzDefEquiPoints[⟨local options⟩](⟨pt1,pt2⟩)

arguments	default	definition
(pt1,pt2)	no default	unordered list of two items

options	default	definition
dist	2 (cm)	half the distance between the two points
from=pt	no default	reference point
show	false	if true displays compass traces
/compass/delta	0	compass trace size

This macro makes it possible to obtain two points on a straight line equidistant from a given point.

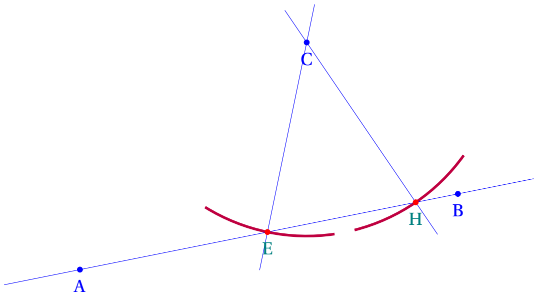
\tkzDefEquiPoints[⟨local options⟩](⟨pt1,pt2⟩)

参数	默认值	含义
(pt1,pt2)	无	两个点的无序列表

参数	默认值	含义
dist	2 (cm)	两点之间距离的一半
from=pt	无	参考点
show	假	如果为真，则显示圆规的痕迹
/compass/delta	0	圆规的尺寸

该宏可以在一条直线上获得与给定点等距的两个点。【即尺规作图中，以定长为半径，画圆与定直线相交，得到两个交点。】

9.6.1. Using \tkzDefEquiPoints with options 示例



```
\begin{tikzpicture}
  \tkzSetUpCompass[color=purple,line width=1pt]
  \tkzDefPoints{0/1/A,5/2/B,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}
```

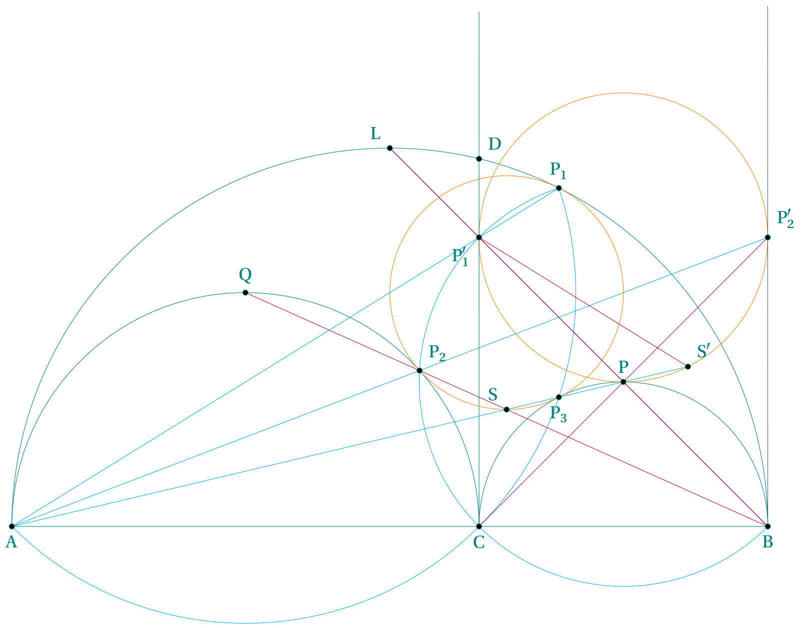
9.7. Middle of an arc 弧的中点

\tkzDefMidArc(⟨pt1,pt2,pt3⟩)

arguments	default	definition
pt1,pt2,pt3	no default	pt1 is the center, $\overset{\frown}{\mathrm{pt2pt3}}$ the arc

\tkzDefMidArc(⟨pt1,pt2,pt3⟩)

参数	默认值	含义
pt1,pt2,pt3	无	点 pt1 是中心，$\overset{\frown}{\mathrm{pt2pt3}}$ 是弧


L
D
P_1
P'_2
P'_1
Q
P_2
S'
P
S
P_3
A
C
B

```
\begin{tikzpicture}[scale=1]
 \tkzDefPoints{0/0/A,10/0/B}
 \tkzDefGoldenRatio(A,B)                             \tkzGetPoint{C}
 \tkzDefMidPoint(A,B)                                \tkzGetPoint{O_1}
 \tkzDefMidPoint(A,C)                                \tkzGetPoint{O_2}
 \tkzDefMidPoint(C,B)                                \tkzGetPoint{O_3}
 \tkzDefMidArc(O_3,B,C)                              \tkzGetPoint{P}
 \tkzDefMidArc(O_2,C,A)                              \tkzGetPoint{Q}
 \tkzDefMidArc(O_1,B,A)                              \tkzGetPoint{L}
 \tkzDefPointBy[rotation=center C angle 90](B)       \tkzGetPoint{c}
 \tkzInterCC[common=B](P,B)(O_1,B)                   \tkzGetFirstPoint{P_1}
 \tkzInterCC[common=C](P,C)(O_2,C)                   \tkzGetFirstPoint{P_2}
 \tkzInterCC[common=C](Q,C)(O_3,C)                   \tkzGetFirstPoint{P_3}
 \tkzInterLC[near](c,C)(O_1,A)                       \tkzGetFirstPoint{D}
 \tkzInterLL(A,P_1)(C,D)                             \tkzGetPoint{P_1'}
 \tkzDefPointBy[inversion = center A through D](P_2) \tkzGetPoint{P_2'}
 \tkzDefCircle[circum](P_3,P_2,P_1)                  \tkzGetPoint{O_4}
 \tkzInterLL(B,Q)(A,P)                               \tkzGetPoint{S}
 \tkzDefMidPoint(P_2',P_1')                          \tkzGetPoint{o}
 \tkzDefPointBy[inversion = center A through D](S)   \tkzGetPoint{S'}
 \tkzDrawArc[cyan,delta=0](Q,A)(P_1)
 \tkzDrawArc[cyan,delta=0](P,P_1)(B)
 \tkzDrawSemiCircles[teal](O_1,B O_2,C O_3,B)
 \tkzDrawCircles[new](o,P O_4,P_1)
 \tkzDrawSegments(A,B)
 \tkzDrawSegments[cyan](A,P_1 A,S' A,P_2')
 \tkzDrawSegments[purple](B,L C,P_2' B,Q B,L S',P_1')
 \tkzDrawLines[add=0 and .8](B,P_2')
 \tkzDrawLines[add=0 and .4](C,D)
 \tkzDrawPoints(A,B,C,P,Q,P_3,P_2,P_1,P_1',D,P_2',L,S,S')
 \tkzLabelPoints(A,B,C,P_3)
 \tkzLabelPoints[above](P,Q,P_1)
 \tkzLabelPoints[above right](P_2,P_2',D,S')
 \tkzLabelPoints[above left](L,S)
  \tkzLabelPoints[below left](P_1')
\end{tikzpicture}
```

10. Point on line or circle 直线或圆上的点

10.1. Point on a line with \tkzDefPointOnLine 直线上的点

\tkzDefPointOnLine[⟨local options⟩](⟨A,B⟩)

arguments	default	definition
pt1,pt2	no default	Two points to define a line

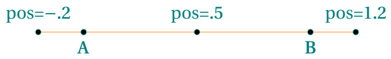
options	default	definition
pos=nb		nb is a decimal

\tkzDefPointOnLine[⟨local options⟩](⟨A,B⟩)

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

10.1.1. Use of option pos pos 选项



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,3/0/B}
\tkzDefPointOnLine[pos=1.2](A,B)\tkzGetPoint{P}
\tkzDefPointOnLine[pos=-0.2](A,B)\tkzGetPoint{R}
\tkzDefPointOnLine[pos=0.5](A,B) \tkzGetPoint{S}
\tkzDrawLine[new](A,B)
\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}
```

10.2. Point on a circle with \tkzDefPointOnCircle 圆上的点

The order of the arguments has changed: now it is center, angle and point or radius. I have added two options for working with radians which are `through in rad` and `R in rad`.
参数的顺序改变了: 现在是中心、角度、点或半径。添加了两个使用弧度的选项，分别是 **through in rad** 和 **R in rad**。

`\tkzDefPointOnCircle[⟨local options⟩]`

options	default	examples definition
through		through = center K1 angle 30 point B]
R		R = center K1 angle 30 radius **\rAp**
through in rad		through in rad= center K1 angle pi/4 point B]
R in rad		R in rad = center K1 angle pi/6 radius **\rAp**

The new order for arguments are : center, angle and point or radius.

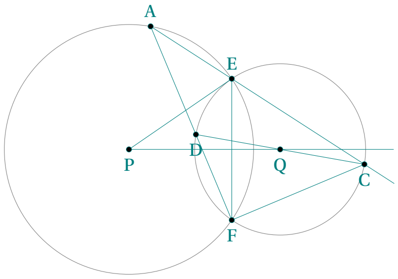
`\tkzDefPointOnCircle[⟨local options⟩]`

参数	默认值	示例定义
through		through=中心 K1 角度 30 点 B]
R		R =中心 K1 角度 30 半径**\rAp**
through in rad		通过半径=中心 K1 角度 pi/4 点 B]
R in rad		R in rad = center K1 angle pi/6 半径 **\rAp**

参数的新顺序是: 中心、角度、点或半径。

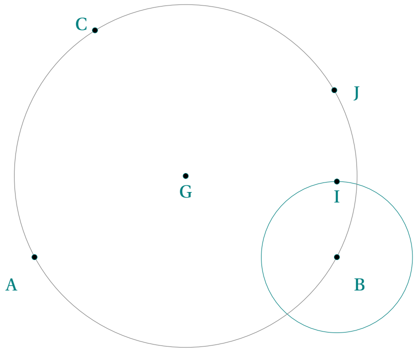
10.2.1. Altshiller's Theorem 阿尔特希勒定理

The two lines joining the points of intersection of two orthogonal circles to a point on one of the circles met the other circle in two diametricaly oposite points. Altshiller p 176
连接两个正交圆的交点与其中一个圆上的一点的两条直线，在两个直径相反的点上与另一个圆相交。阿尔特希勒 p 176



```
\begin{tikzpicture}[scale=.4]
\tkzDefPoints{0/0/P,5/0/Q,3/2/I}
\tkzDefCircle[orthogonal from=P](Q,I)
\tkzGetFirstPoint{E}
\tkzDrawCircles(P,E Q,E)
\tkzInterCC[common=E](P,E)(Q,E) \tkzGetFirstPoint{F}
\tkzDefPointOnCircle[through =  center P angle 80 point E]
 \tkzGetPoint{A}
\tkzInterLC[common=E](A,E)(Q,E)  \tkzGetFirstPoint{C}
\tkzInterLL(A,F)(C,Q)  \tkzGetPoint{D}
\tkzDrawLines[add=0 and .75](P,Q)
\tkzDrawLines[add=0 and 2](A,E)
\tkzDrawSegments(P,E E,F F,C A,F C,D)
\tkzDrawPoints(P,Q,E,F,A,C,D)
\tkzLabelPoints(P,Q,F,C,D)
\tkzLabelPoints[above](E,A)
\end{tikzpicture}
```

10.2.2. Use of \tkzDefPointOnCircle



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/0/B,0.8/3/C}
\tkzDefPointOnCircle[R = center B  angle 90 radius 1]
\tkzGetPoint{I}
\tkzDefCircle[circum](A,B,C)
\tkzGetPoints{G}{g}
\tkzDefPointOnCircle[through = center G  angle 30 point g]
\tkzGetPoint{J}
\tkzDefCircle[R](B,1) \tkzGetPoint{b}
\tkzDrawCircle[teal](B,b)
\tkzDrawCircle(G,J)
\tkzDrawPoints(A,B,C,G,I,J)
\tkzAutoLabelPoints[center=G](A,B,C,J)
\tkzLabelPoints[below](G,I)
\end{tikzpicture}
```

11. Special points relating to a triangle 三角形特殊点

11.1. Triangle center: \tkzDefTriangleCenter 三角形特殊点

```
\tkzDefTriangleCenter[⟨local options⟩](⟨A,B,C⟩)
```

This macro allows you to define the center of a triangle.. Be careful, the arguments are lists of three points. This macro is used in conjunction with `\tkzGetPoint` to get the center you are looking for.
You can use `tkzPointResult` if it is not necessary to keep the results.

arguments	default	example
(pt1,pt2,pt3)	no default	\tkzDefTriangleCenter[ortho](B,C,A)
options	default	definition
ortho	circum	intersection of the altitudes
orthic	circum	…
centroid	circum	intersection of the medians
median	circum	…
circum	circum	circle center circumscribed
in	circum	center of the circle inscribed in a triangle
in	circum	intersection of the bisectors
ex	circum	center of a circle exinscribed to a triangle
euler	circum	center of Euler's circle
gergonne	circum	defined with the Contact triangle
symmedian	circum	Lemoine's point or symmedian center or Grebe's point
lemoine	circum	…
grebe	circum	…
spieker	circum	Spieker circle center
nagel	circum	Nagel Center
mittenpunkt	circum	Or middlespoint
feuerbach	circum	Feuerbach Point

```
\tkzDefTriangleCenter[⟨local options⟩](⟨A,B,C⟩)
```

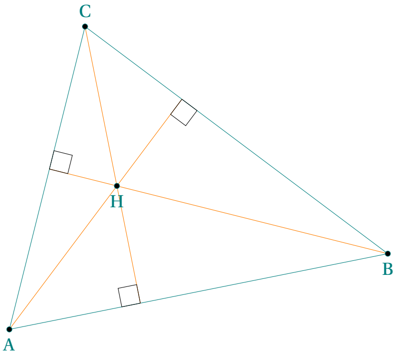
这个宏允许定义一个三角形的特殊点。请注意，参数是三个点的列表。这个宏与`\tkzGetPoint`一起使用，可以得到要找的特殊点。
如果不需要保留结果，可以使用`tkzPointResult`。

参数	默认值	含义
(pt1,pt2,pt3)	无	\tkzDefTriangleCenter[ortho](B,C,A)
选项	默认值	含义
ortho	circum	垂心，三条高的交点
orthic	circum	…
centroid	circum	重心，三条中线的交点
median	circum	…
circum	circum	外心，外接圆心
in	circum	内心，三角形内切圆的圆心
in	circum	平分线的交点
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
gergonne	circum	defined with the Contact triangle
symmedian	circum	陪位重心或对称中心或Grebe的点
lemoine	circum	…
grebe	circum	…
spieker	circum	中点三角形的内心
nagel	circum	奈格尔点
mittenpunkt	circum	三个旁切圆圆心和对应三边中点连线，三线共点。同时也是中点三角形的格高尼点。
feuerbach	circum	Feuerbach Point

【X(6) = SYMMEDIAN POINT (LEMOINE POINT, GREBE POINT) 陪位重心
AN、BM、CE 为三角形的中线，N’、M’、E’ 分别在 BC、CA、AB 上，若角 BAN’ =角 NAC，角 CBM=角 M’ BA，角 ACE=角 E’ CB，则 AN’、BM’、CE’ 三线共点。此点称为“陪位重心”。Trilinears a : b : c = sin A : sin B : sin C Barycentrics a2 : b2 : c2 X(6) is the point of concurrence of the symmedians (reflections of medians in corresponding angle bisectors); the point (x, y, z), given here in actual trilinear distances, that minimizes x2 + y2 + z2. Let A’B’C’ be the pedal triangle of an arbitrary point X, and let S(X) be the vector sum XA’ + XB’ + XC’. Then S(X) = (0 vector) if and only if X = X(6). The ”if” implication is equivalent to the well known fact that X(6) is the centroid of its pedal triangle, and the converse was proved by Barry Wolk (Hyacinthos 19, Dec. 23, 1999). X(6) is the radical trace of the 1st and 2nd Lemoine circles. (Peter J. C. Moses, 8/24/03)】
【中点三角形的内心 The Spieker center of a triangle ABC is the incenter S of the medial triangle A’B’C’ of the triangle ABC. The Spieker center is also the centroid of the perimeter of the original triangle ABC. The medial triangle A’B’C’ is formed by joining the midpoints of the sides of a triangle ABC.
三角形 ABC 中，D，E，F 分别为 BC，AC，AB 的中点，G，I 为三角形的重心与内心（1）过 D，E，F 分别作此三角形内切圆切点三角形对应三边的垂线，证明：它们交于一点 J，在此我们称 J 为三角形 ABC 的 Spieker 点。Spieker 点是三角形“线框”的重心，同时又是三个旁切圆的根心。容易看出 J 是中点三角形 DEF 的内心，而中点三角形与原三角形关于重心 G 是 1：2 位似的，由此（3）中 J、G、I 三点间的关系就不难予以说明。（2）若过 D，E，F 分别作对应的三角形 ABC 三个旁切圆切点所构成的三角形三边的垂线，证明：它们交于一点，分别记作 J1，J2，J3，在此我们称其为原三角形的“旁 Spieker 点”。（3）GJ=0.5IG（4）J 就是三角形 J1J2J3 的垂心，四个 Spieker 点恰好构成垂心组。（5）三角形 J1J2J3 的三条高的垂足为三角形 ABC 三边中点 D，E，F】
【欧几里得几何中，任一个三角形伴随有一个奈格尔点（Nagel）。平面内一个三角形 ABC 具有边长 a = BC，b = CA，和 c = AB，设 TA，TB，和 TC 分别是三旁切圆和三条边的切点。直线 ATA，BTB，CTC 共点 交于三角形 ABC 的奈格尔点 N。奈格尔点以十九世纪德国数学家 Christian Heinrich von Nagel 命名，他在 1836 年提到这个点。】
【The mittenpunkt or middlespoint of a triangle ABC is the point of concurrence of the lines from the excenters D, E, F through the corresponding triangle side midpoints G, H, N. The mittenpunkt of triangle ABC is the symmedian point of the excentral triangle DEF. The point was studied by Christian Heinrich von Nagel in 1836.】
参考资料：https://faculty.evansville.edu/ck6/encyclopedia/ETC.html，上千种与三角形相关的点。

11.1.1. Option ortho or orthic

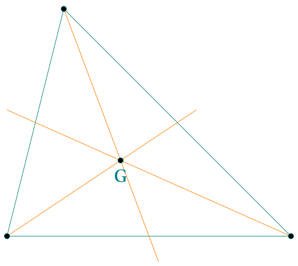
The intersection H of the three altitudes of a triangle is called the orthocenter.
三角形三条高的交点叫做垂心。



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,1){B}
  \tkzDefPoint(1,4){C}
  \tkzDefTriangleCenter[ortho](B,C,A)
  \tkzGetPoint{H}
  \tkzDefSpcTriangle[orthic,name=H](A,B,C){a,b,c}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawSegments[new](A,Ha B,Hb C,Hc)
  \tkzDrawPoints(A,B,C,H)
  \tkzLabelPoint(H){$H$}
  \tkzLabelPoints[below](A,B)
  \tkzLabelPoints[above](C)
 \tkzMarkRightAngles(A,Ha,B B,Hb,C C,Hc,A)
\end{tikzpicture}
```

11.1.2. Option centroid

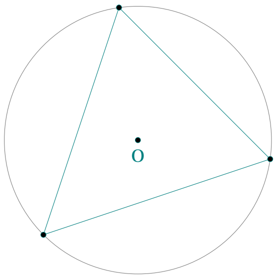
三角形三条中线的交点称为重心。



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/A,5/0/B,1/4/C}
  \tkzDefTriangleCenter[centroid](A,B,C)
  \tkzGetPoint{G}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawLines[add = 0 and 2/3,new](A,G B,G C,G)
  \tkzDrawPoints(A,B,C,G)
  \tkzLabelPoint(G){$G$}
\end{tikzpicture}
```

11.1.3. Option circum

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。

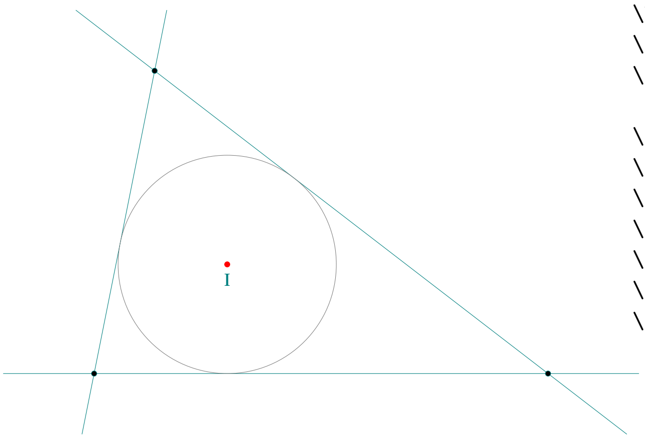


```
\begin{tikzpicture}
  \tkzDefPoints{0/1/A,3/2/B,1/4/C}
  \tkzDefTriangleCenter[circum](A,B,C)
  \tkzGetPoint{O}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawCircle(O,A)
  \tkzDrawPoints(A,B,C,O)
  \tkzLabelPoint(O){$O$}
\end{tikzpicture}
```

11.1.4. Option in

In geometry, the incircle or inscribed circle of a triangle is the largest circle contained in the triangle; it touches (is tangent to) the three sides. The center of the incircle is a triangle center called the triangle's incenter. The center of the incircle, called the incenter, can be found as the intersection of the three internal angle bisectors. The center of an excircle is the intersection of the internal bisector of one angle (at vertex A, for example) and the external bisectors of the other two. The center of this excircle is called the excenter relative to the vertex A, or the excenter of A. Because the internal bisector of an angle is perpendicular to its external bisector, it follows that the center of the incircle together with the three excircle centers form an orthocentric system.
(Article on Wikipedia)
在几何学中，三角形的内切圆是三角形中包含的最大圆；它接触 (相切于) 三条边。内切圆的圆心是一个三角形的中心，称为三角形的内心。内切圆的圆心叫做内心，它是三条内角平分线的交点。旁切圆的圆心是一个角的内平分线 (例如在顶点 A 处) 和另外两个角的外角平分线的交点。这个旁切圆的圆心称为相对于顶点 A 的旁，或 A 的旁心。因为一个角的内平分线垂直于它的外角平分线，所以它的内切圆圆心与三个旁切圆圆心一起构成一个垂心系统。
(参见 Wikipedia)

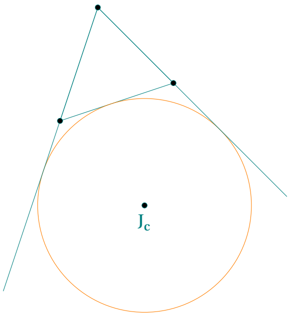
We get the center of the inscribed circle of the triangle. The result is of course in `tkzPointResult`. We can retrieve it with `\tkzGetPoint`.
三角形内切圆的圆心，结果在 `tkzPointResult` 中。可以用`\tkzGetPoint` 来索引它。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefTriangleCenter[in](A,B,C)
   \tkzGetPoint{I}
\tkzDrawLines(A,B B,C C,A)
\tkzDefCircle[in](A,B,C) \tkzGetPoints{I}{i}
\tkzDrawCircle(I,i)
\tkzDrawPoint[red](I)
\tkzDrawPoints(A,B,C)
\tkzLabelPoint(I){$I$}
\end{tikzpicture}
```

11.1.5. Option ex 旁心

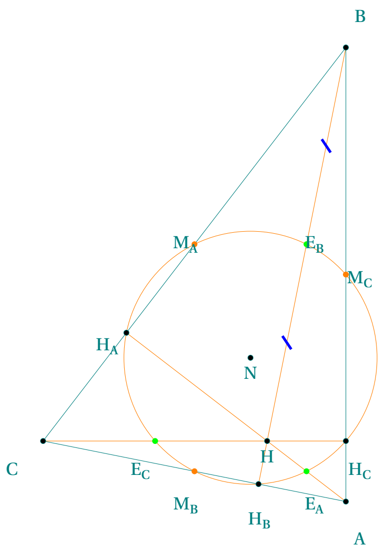
An excircle or escribed circle of the triangle is a circle lying outside the triangle, tangent to one of its sides and tangent to the extensions of the other two. Every triangle has three distinct excircles, each tangent to one of the triangle's sides.
(Article on Wikipedia)
三角形的旁切圆是位于三角形外部的圆，与三角形的一条边相切，并与其他两条边的延长线相切。每个三角形都有三个不同的旁切圆，每个旁切圆都与三角形的一条边相切。
(Article on Wikipedia)
We get the center of an inscribed circle of the triangle. The result is of course in **tkzPointResult**. We can retrieve it with **\tkzGetPoint**.
得到的旁心存储在 **tkzPointResult** 命令中，可以通过**\tkzGetPoint** 检索该点。



```
\begin{tikzpicture}[scale=.5]
  \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}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawCircle[new](J_c,Tc)
  \tkzDrawLines[add=1.5 and 0](A,C B,C)
  \tkzDrawPoints(A,B,C,J_c)
  \tkzLabelPoints(J_c)
\end{tikzpicture}
```

11.1.6. Option euler 欧拉点

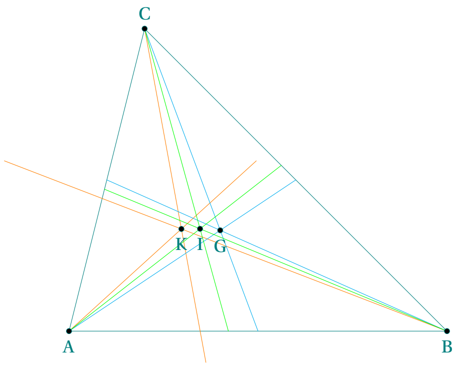
This macro allows to obtain the center of the circle of the nine points or euler's circle or Feuerbach's circle. The nine-point circle, also called Euler's circle or the Feuerbach circle, is the circle that passes through the perpendicular feet H_A, H_B, and H_C dropped from the vertices of any reference triangle ABC on the sides opposite them. Euler showed in 1765 that it also passes through the midpoints M_A, M_B, M_C of the sides of ABC. By Feuerbach's theorem, the nine-point circle also passes through the midpoints E_A, E_B, and E_C of the segments that join the vertices and the orthocenter H. These points are commonly referred to as the Euler points.
(https://mathworld.wolfram.com/Nine-PointCircle.html)
这个宏允许获得九点圆或欧拉圆或费尔巴哈圆的圆心。欧拉点是欧拉圆的圆心，欧拉圆又称九点圆或费尔巴哈圆。欧拉圆是经过三角形 ABC 三个顶点向对边作垂线形成的三个垂足 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 三条边的中点 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,rotate=90]
 \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(A,B,C)
 \tkzDrawCircle[new](N,E_A)
 \tkzDrawSegments[new](A,H_A B,H_B C,H_C)
 \tkzDrawPoints(A,B,C,N,H)
 \tkzDrawPoints[new](M_A,M_B,M_C)
 \tkzDrawPoints( 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}
```

11.1.7. Option **symmedian** 陪位重心

The point of concurrence K of the symmedians, sometimes also called the Lemoine point (in England and France) or the Grebe point (in Germany).
Weisstein, Eric W. ”Symmedian Point.” From MathWorld–A Wolfram Web Resource.
对称点 K，有时也称为 Lemoine 点（在英国和法国）或 Grebe 点（在德国）。
Weisstein, Eric W. ”Symmedian Point.” From MathWorld–A Wolfram Web Resource.

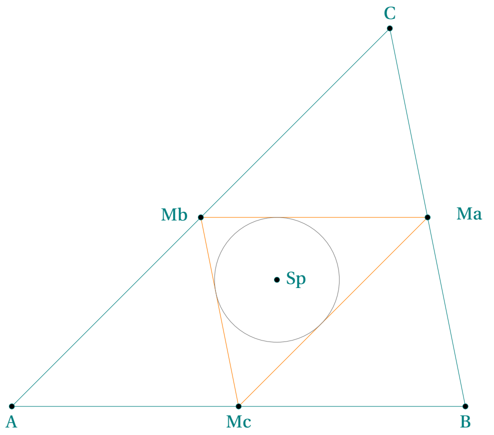


```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPoint(1,4){C}
  \tkzDefTriangleCenter[symmedian](A,B,C)\tkzGetPoint{K}
  \tkzDefTriangleCenter[median](A,B,C)\tkzGetPoint{G}
  \tkzDefTriangleCenter[in](A,B,C)\tkzGetPoint{I}
  \tkzDefSpcTriangle[centroid,name=M](A,B,C){a,b,c}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

11.1.8. Option **spieker**

The Spieker center is the center Sp of the Spieker circle, i.e., the incenter of the medial triangle of a reference triangle.
Weisstein, Eric W. ”Spieker Center.” From MathWorld–A Wolfram Web Resource.
Spieker 点是中点三角形的内心。

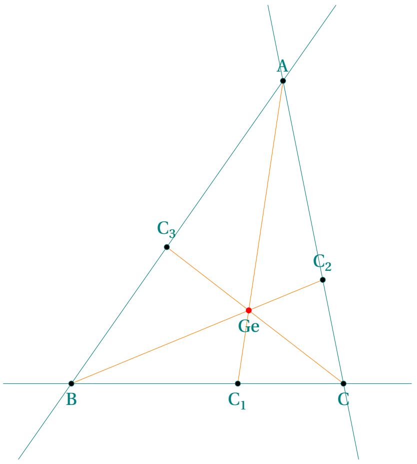
【中点三角形的内心 The Spieker center of a triangle ABC is the incenter S of the medial triangle A'B'C' of the triangle ABC. The Spieker center is also the centroid of the perimeter of the original triangle ABC. The medial triangle A'B'C' is formed by joining the midpoints of the sides of a triangle ABC.
三角形 ABC 中，D，E，F 分别为 BC，AC，AB 的中点，G，I 为三角形的重心与内心（1）过 D，E，F 分别作此三角形内切圆切点三角形对应三边的垂线，证明：它们交于一点 J，在此我们称 J 为三角形 ABC 的 Spieker 点。Spieker 点是三角形“线框”的重心，同时又是三个旁切圆的根心。容易看出 J 是中点三角形 DEF 的内心，而中点三角形与原三角形关于重心 G 是 1：2 位似的，由此（3）中 J、G、I 三点间的关系就不难予以说明。（2）若过 D，E，F 分别作对应的三角形 ABC 三个旁切圆切点所构成的三角形三边的垂线，证明：它们交于一点，分别记作 J1，J2，J3，在此我们称其为原三角形的“旁 Spieker 点”。（3）GJ=0.5IG（4）J 就是三角形 J1J2J3 的垂心，四个 Spieker 点恰好构成垂心组。（5）三角形 J1J2J3 的三条高的垂足为三角形 ABC 三边中点 D，E，F】



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,6/0/B,5/5/C}
 \tkzDefSpcTriangle[medial](A,B,C){Ma,Mb,Mc}
 \tkzDefTriangleCenter[centroid](A,B,C)
 \tkzGetPoint{G}
 \tkzDefTriangleCenter[spieker](A,B,C)
 \tkzGetPoint{Sp}
 \tkzDrawPolygon[](A,B,C)
 \tkzDrawPolygon[new](Ma,Mb,Mc)
 \tkzDefCircle[in](Ma,Mb,Mc) \tkzGetPoints{I}{i}
 \tkzDrawCircle(I,i)
 \tkzDrawPoints(B,C,A,Sp,Ma,Mb,Mc)
 \tkzAutoLabelPoints[center=G,dist=.3](Ma,Mb)
 \tkzLabelPoints[right](Sp)
 \tkzLabelPoints[below](A,B,Mc)
 \tkzLabelPoints[above](C)
\end{tikzpicture}
```

11.1.9. Option gergonne

The Gergonne Point is the point of concurrency which results from connecting the vertices of a triangle to the opposite points of tangency of the triangle's incircle. (Joseph Gergonne French mathematician)
Gergonne 点是指将三角形的顶点，和三角形内切圆与该顶点的对边的切点相连而产生的交点。(Joseph Gergonne 法国数学家)

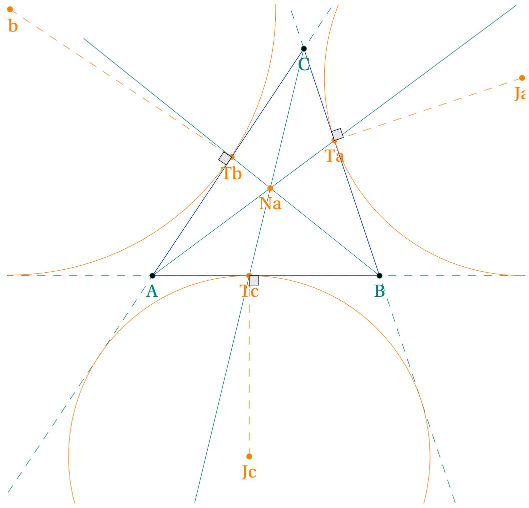


```
\begin{tikzpicture}
\tkzDefPoints{0/0/B,3.6/0/C,2.8/4/A}
\tkzDefTriangleCenter[gergonne](A,B,C)
\tkzGetPoint{Ge}
\tkzDefSpcTriangle[intouch](A,B,C){C_1,C_2,C_3}
\tkzDefCircle[in](A,B,C) \tkzGetPoints{I}{i}
\tkzDrawLines[add=.25 and .25,teal](A,B A,C B,C)
\tkzDrawSegments[new](A,C_1 B,C_2 C,C_3)
\tkzDrawPoints(A,...,C,C_1,C_2,C_3)
\tkzDrawPoints[red](Ge)
\tkzLabelPoints(B,C,C_1,Ge)
\tkzLabelPoints[above](A,C_2,C_3)
\end{tikzpicture}
```

11.1.10. Option nagel

Let Ta be the point at which the excircle with center Ja meets the side BC of a triangle ABC, and define Tb and Tc similarly. Then the lines ATa, BTb, and CTc concur in the Nagel point Na.

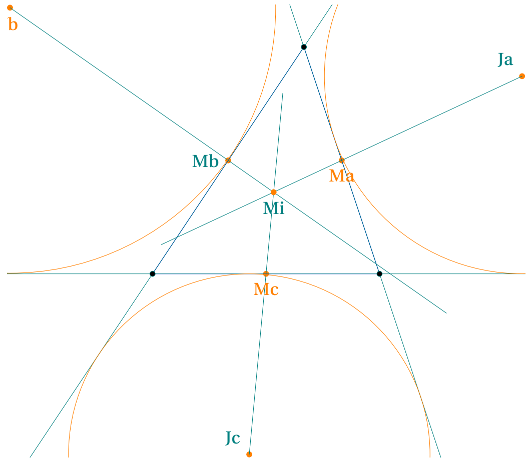
Weisstein, Eric W. ”Nagel point.” From MathWorld–A Wolfram Web Resource.
令 Ta、Tb 和 Tc 分别为旁切圆与三角形三条边的切点，连线 ATa、BTb 和 CTc，其交点称为 Nagel 点，俗称三角形的“界心”。



```
\begin{tikzpicture}[scale=.5]
\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}
\tkzDefTriangleCenter[nagel](A,B,C)
\tkzGetPoint{Na}
\tkzDrawPolygon[blue](A,B,C)
\tkzDrawLines[add=0 and 1](A,Ta B,Tb C,Tc)
\tkzDrawPoints[new](Ja,Jb,Jc,Ta,Tb,Tc)
\tkzClipBB
\tkzDrawLines[add=1 and 1,dashed](A,B B,C C,A)
\tkzDrawCircles[new](Ja,Ta Jb,Tb Jc,Tc)
\tkzDrawSegments[new,dashed](Ja,Ta Jb,Tb Jc,Tc)
\tkzDrawPoints(B,C,A)
\tkzDrawPoints[new](Na)
\tkzLabelPoints(B,C,A)
\tkzLabelPoints[new](Na)
\tkzLabelPoints[new](Ja,Jb,Jc,Ta,Tb,Tc)
\tkzMarkRightAngles[fill=gray!20](Ja,Ta,C
            Jb,Tb,A Jc,Tc,B)
\end{tikzpicture}
```

11.1.11. Option mittenpunkt

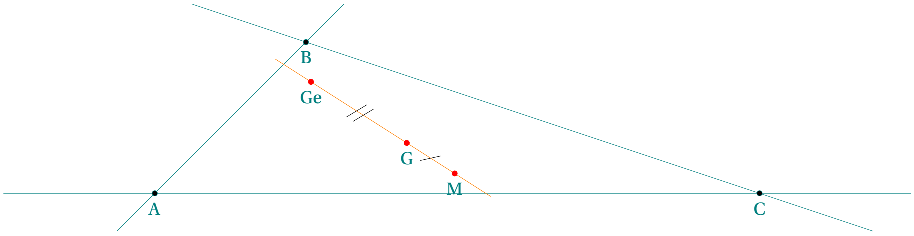
The mittenpunkt (also called the middlespoint) of a triangle ABC is the symmedian point of the excentral triangle, i.e., the point of concurrence M of the lines from the excenters through the corresponding triangle side midpoints.
Weisstein, Eric W. ”Mittenpunkt.” From MathWorld–A Wolfram Web Resource.
三个旁切圆圆心与对应边中点连线的交点。
三角形 ABC 的中间点 (也称为中点) 是外心三角形的对称中点，即从外心通过相应三角形边中点的直线的重合点 M。



```
\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[new](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[new](Ja,Ta Jb,Tb Jc,Tc)
 \tkzDrawPoints(B,C,A)
 \tkzDrawPoints[new](Mi)
 \tkzLabelPoints(Mi)
 \tkzLabelPoints[left](Mb)
 \tkzLabelPoints[new](Ma,Mc,Jb,Jc)
 \tkzLabelPoints[above left](Ja,Jc)
\end{tikzpicture}
```

11.1.12. Relation between gergonne, centroid and mittenpunkt

The Gergonne point Ge, triangle centroid G, and mittenpunkt M are collinear, with GeG/GM=2.
Gergonne 点 Ge、三角形重心 G、mittenpunkt 点 M 共线，并且 GeG/GM=2。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,2/2/B,8/0/C}
\tkzDefTriangleCenter[gergonne](A,B,C) \tkzGetPoint{Ge}
\tkzDefTriangleCenter[centroid](A,B,C)
\tkzGetPoint{G}
\tkzDefTriangleCenter[mittenpunkt](A,B,C)
\tkzGetPoint{M}
\tkzDrawLines[add=.25 and .25,teal](A,B A,C B,C)
\tkzDrawLines[add=.25 and .25,new](Ge,M)
\tkzDrawPoints(A,...,C)
\tkzDrawPoints[red,size=2](G,M,Ge)
\tkzLabelPoints(A,...,C,M,G,Ge)
\tkzMarkSegment[mark=s||](Ge,G)
\tkzMarkSegment[mark=s|](G,M)
\end{tikzpicture}
```

12. Definition of points by transformation 利用坐标变换定义点

These transformations are:
这些变换主要有：

- translation;
- homothety;
- orthogonal reflection or symmetry;
- central symmetry;
- orthogonal projection;
- rotation (degrees or radians);
- inversion with respect to a circle.
- 平移;
- 缩放;
- 轴对称;
- 中心对称;
- 正交投影;
- 旋转 (度或弧度);
- 关于圆的反演.

12.1. \tkzDefPointBy 通过变换定义一个点

The choice of transformations is made through the options. There are two macros, one for the transformation of a single point `\tkzDefPointBy` and the other for the transformation of a list of points `\tkzDefPointsBy`. By default the image of A is A′. For example, we'll write:
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)
```

The result is in **tkzPointResult**
结果保存于 **tkzPointResult** 命令中。

`\tkzDefPointBy[⟨local options⟩](⟨pt⟩)`

The argument is a simple existing point and its image is stored in **tkzPointResult**. If you want to keep this point then the macro **\tkzGetPoint{M}** allows you to assign the name `M` to the point.

arguments	definition	examples
pt	`existing point name`	(A)

options		examples
`translation`	`= from #1 to #2`	`[translation=from A to B](E)`
`homothety`	`= center #1 ratio #2`	`[homothety=center A ratio .5](E)`
`reflection`	`= over #1--#2`	`[reflection=over A--B](E)`
`symmetry`	`= center #1`	`[symmetry=center A](E)`
`projection`	`= onto #1--#2`	`[projection=onto A--B](E)`
`rotation`	`= center #1 angle #2`	`[rotation=center O angle 30](E)`
`rotation in rad`	`= center #1 angle #2`	`[rotation in rad=center O angle pi/3](E)`
`rotation with nodes`	`= center #1 from #2 to #3`	`[center O from A to B](E)`
`inversion`	`= center #1 through #2`	`[inversion =center O through A](E)`
`inversion negative`	`= center #1 through #2`	`...`

The image is only defined and not drawn.

\tkzDefPointBy[⟨ 命令选项 ⟩](⟨pt⟩)

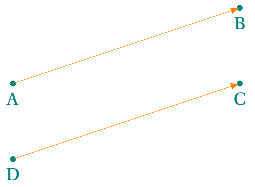
参数是一个已知点，结果存储于命令\tkzPointResult，可用\tkzGetPoint{M}命令保存该点，并为点命名。

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

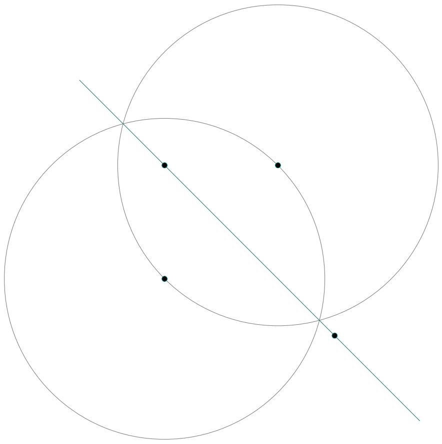
该命令仅定义一个点，并不绘制该点。

12.1.1. translation 平移示例



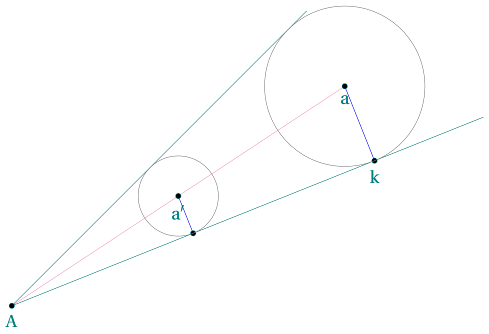
```
\begin{tikzpicture}[>=latex]
 \tkzDefPoints{0/0/A,3/1/B,3/0/C}
 \tkzDefPointBy[translation= from B to A](C)
 \tkzGetPoint{D}
 \tkzDrawPoints[teal](A,B,C,D)
 \tkzLabelPoints[color=teal](A,B,C,D)
 \tkzDrawSegments[orange,->](A,B D,C)
\end{tikzpicture}
```

12.1.2. reflection (orthogonal symmetry) 轴对称示例



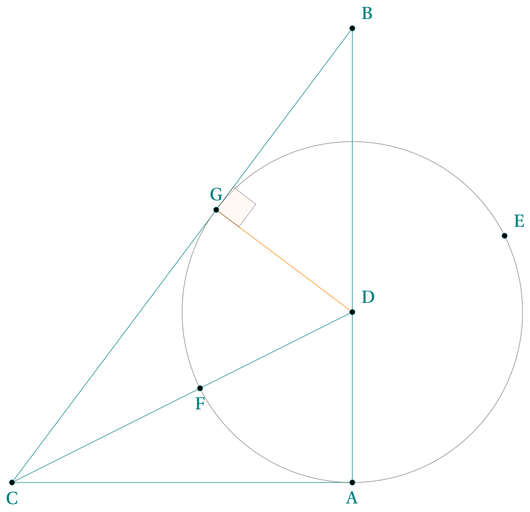
```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoints{-2/-2/A,-1/-1/C,-4/2/D,-4/0/O}
 \tkzDrawCircle(O,A)
 \tkzDefPointBy[reflection = over C--D](A)
 \tkzGetPoint{A'}
 \tkzDefPointBy[reflection = over C--D](O)
 \tkzGetPoint{O'}
 \tkzDrawCircle(O',A')
 \tkzDrawLine[add= .5 and .5](C,D)
 \tkzDrawPoints(C,D,O,O')
\end{tikzpicture}
```

12.1.3. homothety and projection



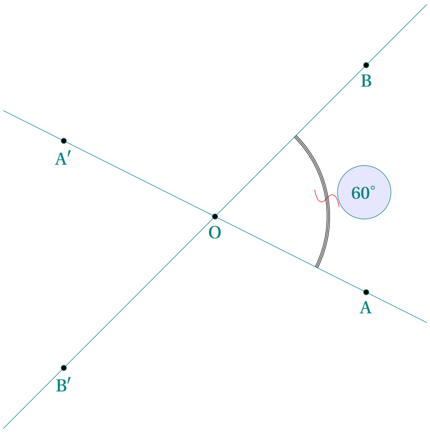
```
\begin{tikzpicture}
  \tkzDefPoints{0/1/A,5/3/B,3/4/C}
  \tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
  \tkzDrawLine[add=0 and 0,color=magenta!50 ](A,a)
  \tkzDefPointBy[homothety=center A ratio .5](a)
  \tkzGetPoint{a'}
  \tkzDefPointBy[projection = onto A--B](a')
  \tkzGetPoint{k'}
  \tkzDefPointBy[projection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

12.1.4. projection 投影示例



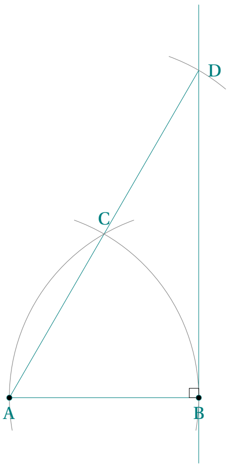
```
\begin{tikzpicture}[scale=1.5]
 \tkzDefPoints{0/0/A,0/4/B}
 \tkzDefTriangle[pythagore](B,A) \tkzGetPoint{C}
 \tkzDefLine[bisector](B,C,A) \tkzGetPoint{c}
 \tkzInterLL(C,c)(A,B)  \tkzGetPoint{D}
 \tkzDefPointBy[projection=onto B--C](D)
 \tkzGetPoint{G}
 \tkzInterLC(C,D)(D,A) \tkzGetPoints{E}{F}
 \tkzDrawPolygon(A,B,C)
 \tkzDrawSegment(C,D)
 \tkzDrawCircle(D,A)
 \tkzDrawSegment[new](D,G)
 \tkzMarkRightAngle[fill=orange!10,opacity=.4](D,G,B)
 \tkzDrawPoints(A,C,F) \tkzLabelPoints(A,C,F)
 \tkzDrawPoints(B,D,E,G)
 \tkzLabelPoints[above right](B,D,E)
  \tkzLabelPoints[above](G)
 \end{tikzpicture}
```

12.1.5. symmetry 中心对称示例



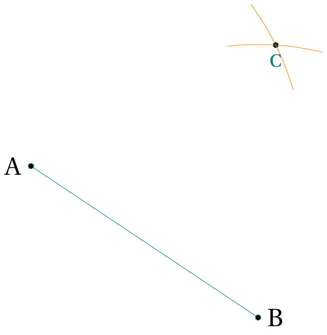
```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{2/-1/A,2/2/B,0/0/O}
  \tkzDefPointsBy[symmetry=center O](B,A){}
  \tkzDrawLine(A,A')
  \tkzDrawLine(B,B')
  \tkzMarkAngle[mark=s,arc=lll,
      size=1.5,mkcolor=red](A,O,B)
  \tkzLabelAngle[pos=2,circle,draw,
    fill=blue!10,font=\scriptsize](A,O,B){$60^{\circ}$}
  \tkzDrawPoints(A,B,O,A',B')
  \tkzLabelPoints(B,B')
  \tkzLabelPoints[below](A,O,A')
\end{tikzpicture}
```

12.1.6. rotation 旋转示例 (度)



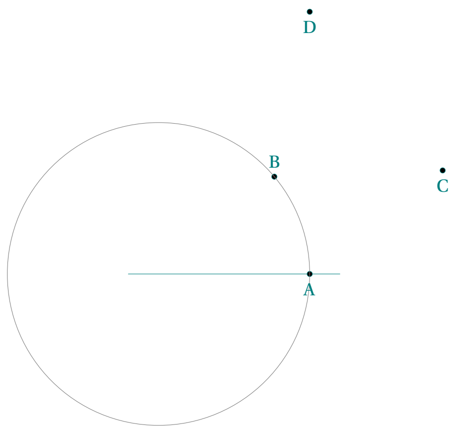
```
\begin{tikzpicture}[scale=0.5]
 \tkzDefPoints{0/0/A,5/0/B}
 \tkzDrawSegment(A,B)
 \tkzDefPointBy[rotation=center A angle 60](B)
 \tkzGetPoint{C}
 \tkzDefPointBy[symmetry=center C](A)
 \tkzGetPoint{D}
 \tkzDrawSegment(A,tkzPointResult)
 \tkzDrawLine(B,D)
 \tkzDrawArc(A,B)(C) \tkzDrawArc(B,C)(A)
 \tkzDrawArc(C,D)(D)
 \tkzMarkRightAngle(D,B,A)
 \tkzDrawPoints(A,B)
 \tkzLabelPoints(A,B)
 \tkzLabelPoints[above](C)
 \tkzLabelPoints[right](D)
\end{tikzpicture}
```

12.1.7. rotation in radian 旋转示例 (弧度)



```
\begin{tikzpicture}
  \tkzDefPoint["$A$" left](1,5){A}
  \tkzDefPoint["$B$" right](4,3){B}
  \tkzDefPointBy[rotation in rad= center A angle pi/3](B)
  \tkzGetPoint{C}
  \tkzDrawSegment(A,B)
  \tkzDrawPoints(A,B,C)
  \tkzCompass(A,C)
  \tkzCompass(B,C)
  \tkzLabelPoints(C)
\end{tikzpicture}
```

12.1.8. rotation with nodes 旋转示例（标注)



```
\begin{tikzpicture}
 \tkzDefPoint(0,0){O}
 \tkzDefPoint(0:2){A}
 \tkzDefPoint(40:2){B}
 \tkzDefPoint(20:4){C}
 \tkzDrawLine(O,A)
 \tkzDefPointBy[rotation with nodes%
             =center O from A to B](C)
 \tkzGetPoint{D}
\tkzDrawPoints(A,B,C,D)
\tkzDrawCircle(O,A)
\tkzLabelPoints(A,C,D)
\tkzLabelPoints[above](B)
\end{tikzpicture}
```

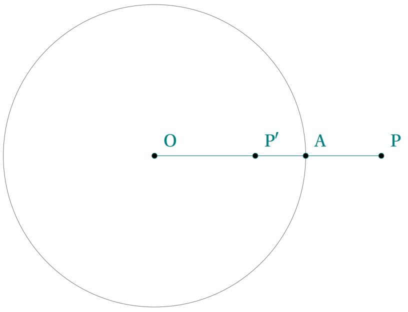
12.1.9. inversion 反演

Inversion is the process of transforming points to a corresponding set of points known as their inverse points. Two points P and P′ are said to be inverses with respect to an inversion circle having inversion center O and inversion radius k if P′ is the perpendicular foot of the altitude of OQP, where Q is a point on the circle such that OQ is perpendicular to PQ.
The quantity k^2 is known as the circle power (Coxeter 1969, p. 81). (https://mathworld.wolfram.com/Inversion.html)
反演是将点集转换为另一点集的过程，这些点称为它们的反演点。如果 P′ 是 OQP 高的垂足，那么两点 P 和 P′ 被认为是关于具有反演中心 O 和反演半径 k 的反演圆的反演，其中 Q 是圆上的一点，使得 OQ 垂直于 PQ。k^2 的量被称为圆幂 (Coxeter 1969，第 81 页)。(https://mathworld.wolfram.com/Inversion.html)
Some propositions :
一些命题:

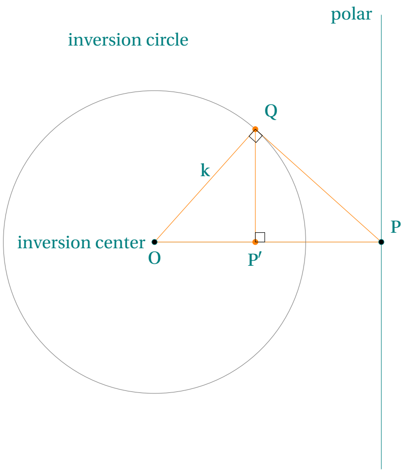
- The inverse of a circle (not through the center of inversion) is a circle.
- The inverse of a circle through the center of inversion is a line.
- The inverse of a line (not through the center of inversion) is a circle through the center of inversion.
- A circle orthogonal to the circle of inversion is its own inverse.
- A line through the center of inversion is its own inverse.
- Angles are preserved in inversion.
- 圆的反演 (不通过反演的中心) 是圆。
- 通过圆心的圆的反演是一条线。
- 直线的反演 (不通过反演中心) 是通过反演中心的圆。
- 与反演圆正交的圆是它自己的反演。
- 通过反演中心的直线是其自身的反演。
- 角度在反转中保持不变。

【设在平面内给定一点 O 和常数 k（k 不等于零），对于平面内任意一点 A，确定 A′，使 A′ 为直线 OA 上一点，并且有向线段 OA 与 OA′ 满足 OA・OA′=k，我们称这种变换是以 O 为反演中心，以 k 为反演幂的反演变换，简称反演。称 A′ 为 A 关于 O（r）的互为反演点。

当 k>0 时，有向线段 OA 与 OA′ 同向，A 与 A′ 在反演极同侧，这种反演变换称为正幂反演，亦叫双曲线式反演变换。
当 k<0 时，有向线段 OA 与 OA′ 反向，A 与 A′ 在反演极异侧，这种反演变换称为负幂反演，亦叫椭圆式反演变换。
在某一反演变换中相互对应的两个图形互为反演图形或反象】
Explanation:
解释:
Directly (Center O power=$k^2 = OA^2 = OP \times OP'$)

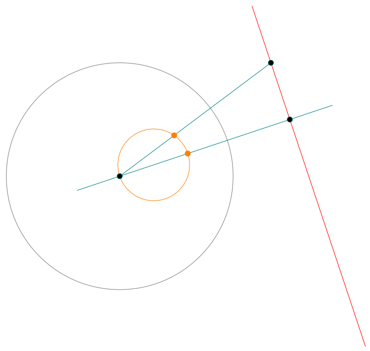


```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{4/0/A,6/0/P,0/0/O}
  \tkzDefPointBy[inversion = center O through A](P)
  \tkzGetPoint{P'}
  \tkzDrawSegments(O,P)
  \tkzDrawCircle(O,A)
  \tkzLabelPoints[above right,font=\scriptsize](O,A,P,P')
  \tkzDrawPoints(O,A,P,P')
\end{tikzpicture}
```



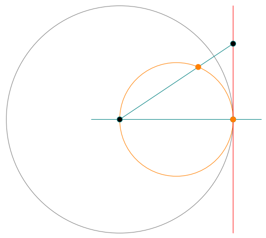
```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{4/0/A,6/0/P,0/0/O}
  \tkzDefLine[orthogonal=through P](O,P)
  \tkzGetPoint{L}
  \tkzDefLine[tangent from = P](O,A) \tkzGetPoints{R}{Q}
  \tkzDefPointBy[projection=onto O--A](Q) \tkzGetPoint{P'}
  \tkzDrawSegments(O,P O,A)
  \tkzDrawSegments[new](O,P O,Q P,Q Q,P')
  \tkzDrawCircle(O,A)
  \tkzDrawLines[add=1 and 0](P,L)
  \tkzLabelPoints[below,font=\scriptsize](O,P')
  \tkzLabelPoints[above right,font=\scriptsize](P,Q)
  \tkzDrawPoints(O,P) \tkzDrawPoints[new](Q,P')
  \tkzLabelSegment[above](O,Q){$k$}
  \tkzMarkRightAngles(A,P',Q P,Q,O)
  \tkzLabelCircle[above=.5cm,
      font=\scriptsize](O,A)(100){inversion circle}
  \tkzLabelPoint[left,font=\scriptsize](O){inversion center}
  \tkzLabelPoint[left,font=\scriptsize](L){polar}
\end{tikzpicture}
```

12.1.10. Inversion of lines ex 1 直线的反演例 1



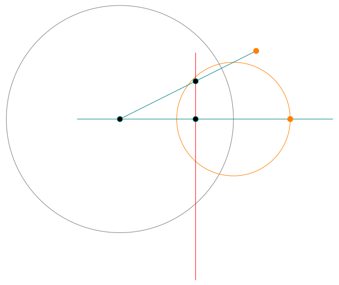
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/O,3/0/I,4/3/P,6/-3/Q}
\tkzDrawCircle(O,I)
\tkzDefPointBy[projection= onto P--Q](O) \tkzGetPoint{A}
\tkzDefPointBy[inversion = center O through I](A)
\tkzGetPoint{A'}
\tkzDefPointBy[inversion = center O through I](P)
\tkzGetPoint{P'}
\tkzDefCircle[diameter](O,A')\tkzGetPoint{o}
\tkzDrawCircle[new](o,A')
\tkzDrawLines[add=.25 and .25,red](P,Q)
\tkzDrawLines[add=.25 and .25](O,A)
\tkzDrawSegments(O,P)
\tkzDrawPoints(A,P,O) \tkzDrawPoints[new](A',P')
\end{tikzpicture}
```

12.1.11. inversion of lines ex 2 直线的反演例 2



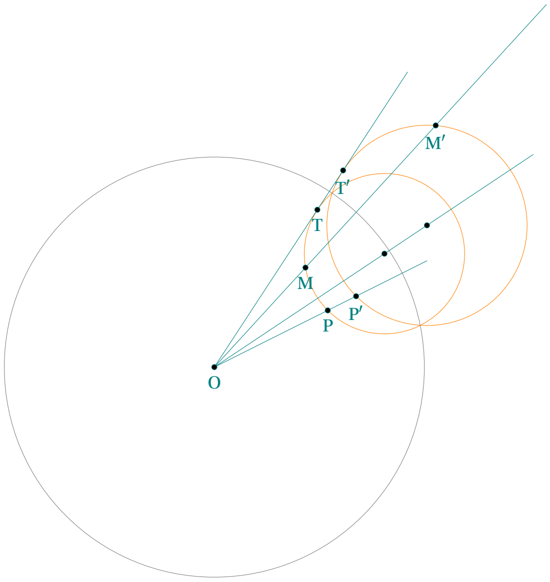
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/O,3/0/I,3/2/P,3/-2/Q}
\tkzDrawCircle(O,I)
\tkzDefPointBy[projection= onto P--Q](O) \tkzGetPoint{A}
\tkzDefPointBy[inversion = center O through I](A)
\tkzGetPoint{A'}
\tkzDefPointBy[inversion = center O through I](P)
\tkzGetPoint{P'}
\tkzDefCircle[diameter](O,A')\tkzGetPoint{o}
\tkzDrawCircle[new](o,A')
\tkzDrawLines[add=.25 and .25,red](P,Q)
\tkzDrawLines[add=.25 and .25](O,A)
\tkzDrawSegments(O,P)
\tkzDrawPoints(A,P,O) \tkzDrawPoints[new](A',P')
\end{tikzpicture}
```

12.1.12. inversion of lines ex 3 直线的反演例 3



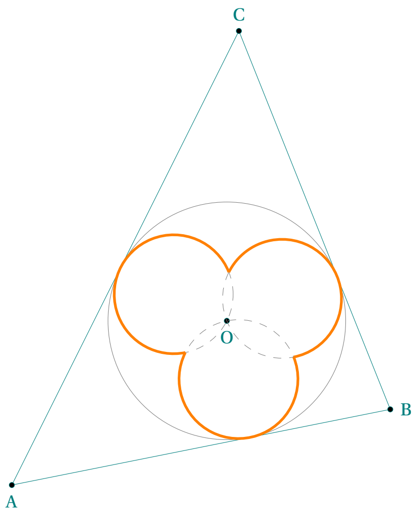
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/O,3/0/I,2/1/P,2/-2/Q}
\tkzDrawCircle(O,I)
\tkzDefPointBy[projection= onto P--Q](O) \tkzGetPoint{A}
\tkzDefPointBy[inversion = center O through I](A)
\tkzGetPoint{A'}
\tkzDefPointBy[inversion = center O through I](P)
\tkzGetPoint{P'}
\tkzDefCircle[diameter](O,A')
\tkzDrawCircle[new](I,A')
\tkzDrawLines[add=.25 and .75,red](P,Q)
\tkzDrawLines[add=.25 and .25](O,A')
\tkzDrawSegments(O,P')
\tkzDrawPoints(A,P,O) \tkzDrawPoints[new](A',P')
\end{tikzpicture}
```

12.1.13. inversion of circle and homothety 圆的反演和缩放



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/O,3/2/A,2/1/P}
\tkzDefLine[tangent from = O](A,P) \tkzGetPoints{T}{X}
\tkzDefPointsBy[homothety = center O%
                ratio 1.25](A,P,T){}
\tkzInterCC(A,P)(A',P') \tkzGetPoints{C}{D}
\tkzCalcLength(A,P)
\tkzGetLength{rAP}
\tkzDefPointOnCircle[R=  center A angle 190 radius \rAP]
\tkzGetPoint{M}
\tkzDefPointBy[inversion = center O through C](M)
\tkzGetPoint{M'}
\tkzDrawCircles[new](A,P A',P')
\tkzDrawCircle(O,C)
\tkzDrawLines[add=0 and .5](O,T' O,A' O,M' O,P')
\tkzDrawPoints(A,A',P,P',O,T,T',M,M')
\tkzLabelPoints(O,T,T',M,M')
\tkzLabelPoints[below](P,P')
\end{tikzpicture}
```

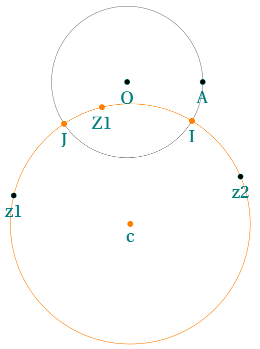
12.1.14. inversion of Triangle with respect to the Incircle 三角形相对于内切圆的反演



```
\begin{tikzpicture}[scale=1]
\tkzDefPoints{0/0/A,5/1/B,3/6/C}
\tkzDefTriangleCenter[in](A,B,C) \tkzGetPoint{O}
\tkzDefPointBy[projection= onto A--C](O) \tkzGetPoint{b}
\tkzDefPointBy[projection= onto A--C](O) \tkzGetPoint{b}
\tkzDefPointBy[projection= onto B--C](O) \tkzGetPoint{a}
\tkzDefPointBy[projection= onto A--B](O) \tkzGetPoint{c}
\tkzDefPointsBy[inversion = center O through b](a,b,c)%
                                     {Ia,Ib,Ic}
\tkzDefMidPoint(O,Ia) \tkzGetPoint{Ja}
\tkzDefMidPoint(O,Ib) \tkzGetPoint{Jb}
\tkzDefMidPoint(O,Ic) \tkzGetPoint{Jc}
\tkzInterCC(Ja,O)(Jb,O) \tkzGetPoints{O}{x}
\tkzInterCC(Ja,O)(Jc,O) \tkzGetPoints{y}{O}
\tkzInterCC(Jb,O)(Jc,O) \tkzGetPoints{O}{z}
\tkzDrawPolygon(A,B,C)
\tkzDrawCircle(O,b)\tkzDrawPoints(A,B,C,O)
\tkzDrawCircles[dashed,gray](Ja,y Jb,x Jc,z)
\tkzDrawArc[line width=1pt,orange,delta=0](Jb,x)(z)
\tkzDrawArc[line width=1pt,orange,delta=0](Jc,z)(y)
\tkzDrawArc[line width=1pt,orange,delta=0](Ja,y)(x)
\tkzLabelPoint[below](A){$A$}\tkzLabelPoint[above](C){$C$}
\tkzLabelPoint[right](B){$B$}\tkzLabelPoint[below](O){$O$}
\end{tikzpicture}
```

12.1.15. inversion: orthogonal circle with inversion circle inversion: 带正交圆的反演

The inversion circle itself, circles orthogonal to it, and lines through the inversion center are invariant under inversion. If the circle meets the reference circle, these invariant points of intersection are also on the inverse circle. See I and J in the next figure.
反演圆本身、与其正交的圆以及通过反演中心的线，在反演变换下是不变的。如果圆与参考圆相交，这些不变的交点也在反演圆上。见下图中的 I 和 J。

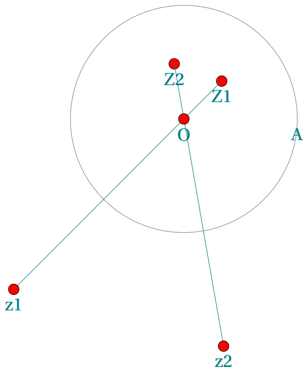


```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,0){O}\tkzDefPoint(1,0){A}
\tkzDefPoint(-1.5,-1.5){z1}
\tkzDefPoint(1.5,-1.25){z2}
\tkzDefCircle[orthogonal through=z1 and z2](O,A)
\tkzGetPoint{c}
\tkzDrawCircle[new](c,z1)
\tkzDefPointBy[inversion =  center O through A](z1)
\tkzGetPoint{Z1}
\tkzInterCC(O,A)(c,z1) \tkzGetPoints{I}{J}
\tkzDefPointBy[inversion =  center O through A](I)
\tkzGetPoint{I'}
\tkzDrawCircle(O,A)
\tkzDrawPoints(O,A,z1,z2)
\tkzDrawPoints[new](c,Z1,I,J)
\tkzLabelPoints(O,A,z1,z2,c,Z1,I,J)
\end{tikzpicture}
```

For a more complex example see Pappus 46.25
有关更复杂的示例，请参见 Pappus 46.25

12.1.16. inversion negative

It's an inversion followed by a symmetry of center O
这是一个圆心 O 对称的反演



```
\begin{tikzpicture}[scale=1.5]
  \tkzDefPoints{1/0/A,0/0/O}
  \tkzDefPoint(-1.5,-1.5){z1}
  \tkzDefPoint(0.35,-2){z2}
  \tkzDefPointBy[inversion negative = center O through A](z1)
  \tkzGetPoint{Z1}
  \tkzDefPointBy[inversion negative = 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[font=\scriptsize](O,A,z1,z2,Z1,Z2)
\end{tikzpicture}
```

12.2. Transformation of multiple points; \tkzDefPointsBy 多点变换

Variant of the previous macro for defining multiple images. You must give the names of the images as arguments, or indicate that the names of the images are formed from the names of the antecedents, leaving the argument empty.
用于定义多个图像的上一个宏的变体。必须将图像的名称作为参数提供，或者指示图像的名称由前置项的名称组成，将参数留空。

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

The images are B′ and C′.
图像分别为 B′ 和 C′。

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

The images are D and E.
图像分别为 D 和 E。

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

The image is B′.
图像为 B′。

```
\tkzDefPointsBy[⟨local options⟩](⟨list of points⟩){⟨list of points⟩}
```

arguments	examples
(⟨list of points⟩){⟨list of pts⟩}	(A,B){E,F} E,F images of A, B

If the list of images is empty then the name of the image is the name of the antecedent to which ” ’ ” is added.

options	examples
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
rotation with nodes = center #1 from #2 to #3	[center O from A to B](E){F}
inversion = center #1 through #2	[inversion = center O through A](E){F}
inversion negative = center #1 through #2	...

The points are only defined and not drawn.

```
\tkzDefPointsBy[⟨local options⟩](⟨list of points⟩){⟨list of points⟩}
```

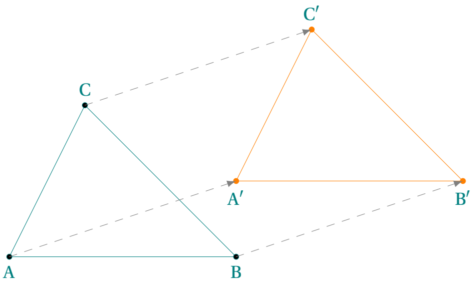
参数	示例
(⟨list of points⟩){⟨list of pts⟩}	(A,B){E,F} A, B 的图像为 E,F

如果图像列表为空，则图像的名称是前面加上” ’ ”。

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
rotation with nodes = center #1 from #2 to #3	[center O from A to B](E){F}
inversion = center #1 through #2	[inversion = center O through A](E){F}
inversion negative = center #1 through #2	...

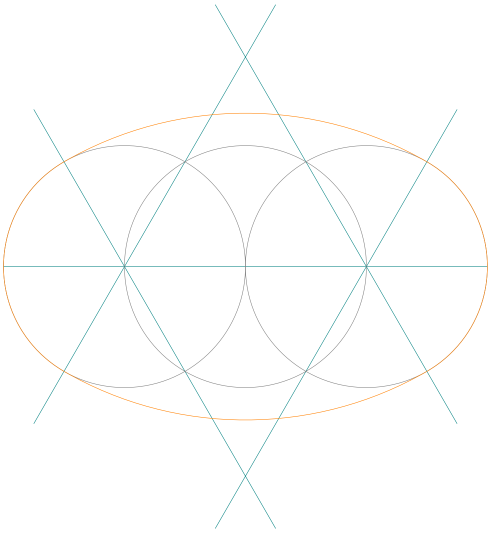
只定义点，不绘制点。

12.2.1. translation of multiple points 多点平移



```
\begin{tikzpicture}[>=latex]
 \tkzDefPoints{0/0/A,3/0/B,3/1/A',1/2/C}
 \tkzDefPointsBy[translation= from A to A'](B,C){}
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPolygon[new](A',B',C')
 \tkzDrawPoints(A,B,C)
 \tkzDrawPoints[new](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}
```

12.2.2. symmetry of multiple points: an oval 多点对称：椭圆



```
\begin{tikzpicture}[scale=0.4]
  \tkzDefPoint(-4,0){I}
  \tkzDefPoint(4,0){J}
  \tkzDefPoint(0,0){O}
  \tkzInterCC(J,O)(O,J) \tkzGetPoints{L}{H}
  \tkzInterCC(I,O)(O,I) \tkzGetPoints{K}{G}
  \tkzInterLL(I,K)(J,H) \tkzGetPoint{M}
  \tkzInterLL(I,G)(J,L) \tkzGetPoint{N}
  \tkzDefPointsBy[symmetry=center J](L,H){D,E}
  \tkzDefPointsBy[symmetry=center I](G,K){C,F}
  \begin{scope}[line style/.style = {very thin,teal}]
    \tkzDrawLines[add=1.5 and 1.5](I,K I,G J,H J,L)
    \tkzDrawLines[add=.5 and .5](I,J)
    \tkzDrawCircles(O,I I,O J,O)
    \tkzDrawArc[delta=0,orange](N,D)(C)
    \tkzDrawArc[delta=0,orange](M,F)(E)
    \tkzDrawArc[delta=0,orange](J,E)(D)
    \tkzDrawArc[delta=0,orange](I,C)(F)
  \end{scope}
\end{tikzpicture}
```

13. Defining points using a vector 通过向量定义点

13.1. \tkzDefPointWith 定义向量点

There are several possibilities to create points that meet certain vector conditions. This can be done with `\tkzDefPointWith`. The general principle is as follows, two points are passed as arguments, i.e. a vector. The different options allow to obtain a new point forming with the first point (with some exceptions) a collinear vector or a vector orthogonal to the first vector. Then the length is either proportional to that of the first one, or proportional to the unit. Since this point is only used temporarily, it does not have to be named immediately. The result is in `tkzPointResult`. The macro `\tkzGetPoint` allows you to retrieve the point and name it differently.
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。
There are options to define the distance between the given point and the obtained point. In the general case this distance is the distance between the 2 points given as arguments if the option is of the "normed" type then the distance between the given point and the obtained point is 1 cm. Then the K option allows to obtain multiples.

`\tkzDefPointWith(⟨pt1,pt2⟩)`

It is in fact the definition of a point meeting vectorial conditions.

arguments	definition	explanation
(pt1,pt2)	point couple	the result is a point in **tkzPointResult**

In what follows, it is assumed that the point is recovered by `\tkzGetPoint{C}`

options	example	explanation
orthogonal	[orthogonal](A,B)	$AC = AB$ and $\overrightarrow{AC}\perp\overrightarrow{AB}$
orthogonal normed	[orthogonal normed](A,B)	$AC = 1$ and $\overrightarrow{AC}\perp\overrightarrow{AB}$
linear	[linear](A,B)	$\overrightarrow{AC} = K\times\overrightarrow{AB}$
linear normed	[linear normed](A,B)	$AC = K$ and $\overrightarrow{AC} = k\times\overrightarrow{AB}$
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	$\overrightarrow{AC} = 2\times\overrightarrow{AB}$

`\tkzDefPointWith(⟨pt1,pt2⟩)`

是满足向量条件的点的定义。

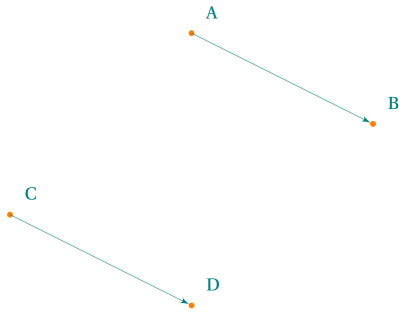
参数	含义	说明
(pt1,pt2)	点对	结果是保存于\tkzPointResult 命令

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	$\overrightarrow{AC} = 2\times\overrightarrow{AB}$

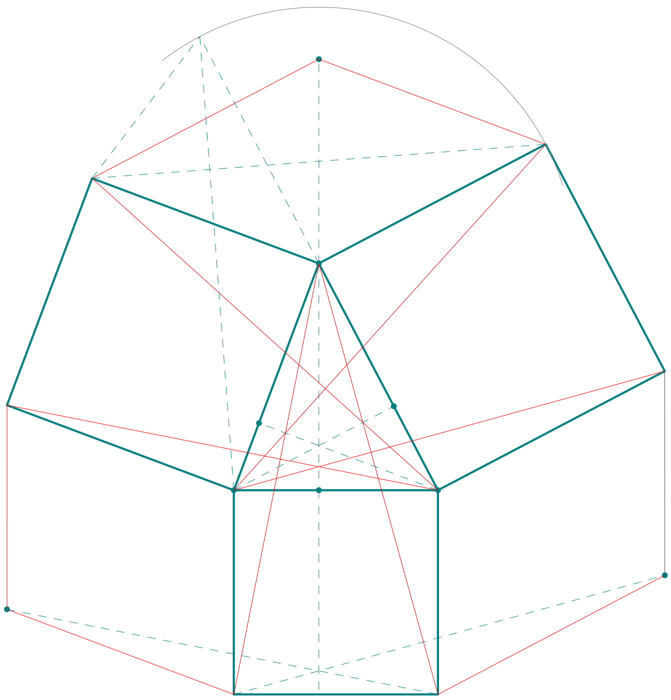
13.1.1. Option colinear at, simple example 简单示例

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



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

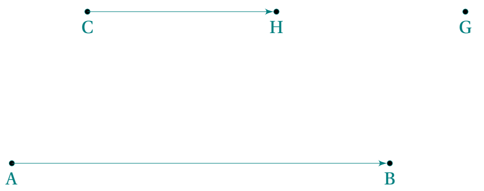
13.1.2. Option colinear at, complex example 复杂示例



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/B,3.6/0/C,1.5/4/A}
\tkzDefSpcTriangle[ortho](A,B,C){Ha,Hb,Hc}
\tkzDefTriangleCenter[ortho](A,B,C) \tkzGetPoint{H}
\tkzDefSquare(A,C) \tkzGetPoints{R}{S}
\tkzDefSquare(B,A) \tkzGetPoints{M}{N}
\tkzDefSquare(C,B) \tkzGetPoints{P}{Q}
\tkzDefPointWith[colinear= at M](A,S) \tkzGetPoint{A'}
\tkzDefPointWith[colinear= at P](B,N) \tkzGetPoint{B'}
\tkzDefPointWith[colinear= at Q](C,R) \tkzGetPoint{C'}
\tkzDefPointBy[projection=onto P--Q](Ha) \tkzGetPoint{Pa}
\tkzDrawPolygon[teal,thick](A,C,R,S)\tkzDrawPolygon[teal,thick](A,B,N,M)
\tkzDrawPolygon[teal,thick](C,B,P,Q)
\tkzDrawPoints[teal,size=2](A,B,C,Ha,Hb,Hc,A',B',C')
\tkzDrawSegments[ultra thin,red](M,A' A',S P,B' B',N Q,C' C',R B,S C,M C,N B,R A,P A,Q)
\tkzDrawSegments[ultra thin,teal, dashed](A,Ha B,Hb C,Hc)
\tkzDefPointBy[rotation=center A angle 90](S) \tkzGetPoint{S'}
\tkzDrawSegments[ultra thin,teal,dashed](B,S' A,S' A,A' M,S' B',Q P,C' M,S Ha,Pa)
\tkzDrawArc(A,S)(S')
\end{tikzpicture}
```

13.1.3. Option colinear at

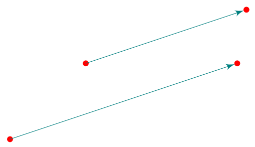
How to use K
带 K 示例



```
\begin{tikzpicture}[vect/.style={->,
               shorten >=1pt,>=latex'}]
  \tkzDefPoints{0/0/A,5/0/B,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}
```

13.1.4. Option colinear at

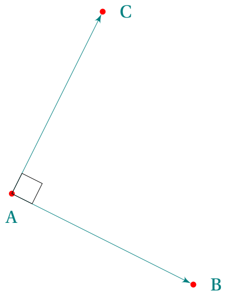
With $K = \frac{\sqrt{2}}{2}$



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

13.1.5. Option orthogonal

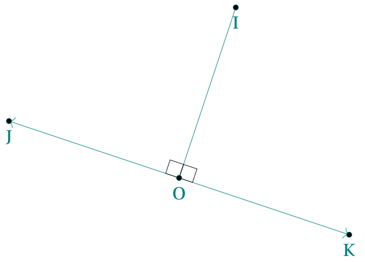
AB=AC since K = 1.



```
\begin{tikzpicture}[scale=1.2,
  vect/.style={->,shorten >=1pt,>=latex'}]
  \tkzDefPoints{2/3/A,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}
```

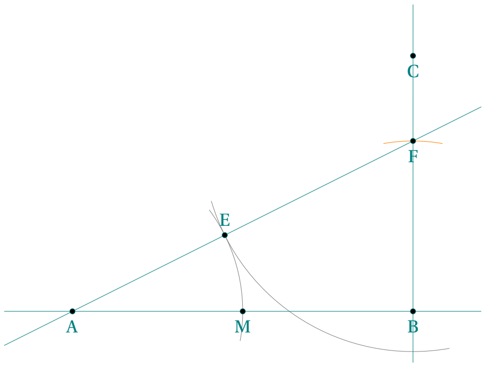
13.1.6. Option orthogonal

With $K = -1$ OK=OI since $|K| = 1$ then OI=OJ=OK.



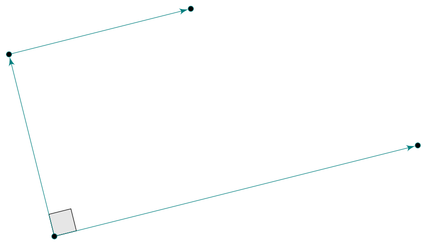
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{1/2/O,2/5/I}
  \tkzDefPointWith[orthogonal](O,I)
  \tkzGetPoint{J}
  \tkzDefPointWith[orthogonal,K=-1](O,I)
  \tkzGetPoint{K}
  \tkzDrawSegment(O,I)
  \tkzDrawSegments[->](O,J O,K)
  \tkzMarkRightAngles(I,O,J I,O,K)
  \tkzDrawPoints(O,I,J,K)
  \tkzLabelPoints(O,I,J,K)
\end{tikzpicture}
```

13.1.7. Option orthogonal more complicated example



```
\begin{tikzpicture}[scale=.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)
  \tkzLabelPoints[above](E)
\end{tikzpicture}
```

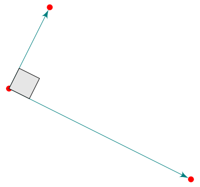
13.1.8. Options colinear and orthogonal



```
\begin{tikzpicture}[scale=1.2,
  vect/.style={->,shorten >=1pt,>=latex'}]
  \tkzDefPoints{2/1/A,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}
```

13.1.9. Option orthogonal normed

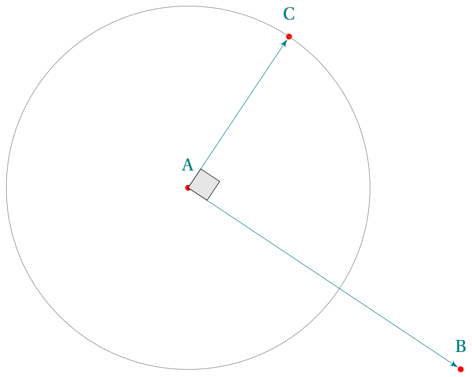
K = 1 AC = 1.



```
\begin{tikzpicture}[scale=1.2,
  vect/.style={->,shorten >=1pt,>=latex'}]
  \tkzDefPoints{2/3/A,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}
```

13.1.10. Option orthogonal normed and K=2

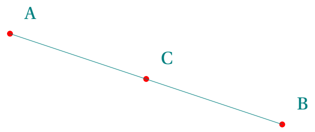
K = 2 therefore AC = 2.



```
\begin{tikzpicture}[scale=1.2,
   vect/.style={->,shorten >=1pt,>=latex'}]
  \tkzDefPoints{2/3/A,5/1/B}
  \tkzDefPointWith[orthogonal normed,K=2](A,B)
  \tkzGetPoint{C}
  \tkzDrawPoints[color=red](A,B,C)
  \tkzDefCircle[R](A,2) \tkzGetPoint{a}
  \tkzDrawCircle(A,a)
  \tkzDrawSegments[vect](A,B A,C)
  \tkzMarkRightAngle[fill=gray!20](B,A,C)
  \tkzLabelPoints[above=3pt](A,B,C)
\end{tikzpicture}
```

13.1.11. Option linear

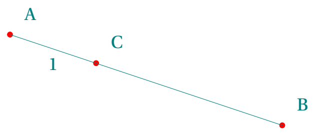
Here K = 0.5.
This amounts to applying a homothety or a multiplication of a vector by a real. Here is the middle of [AB].
这相当于应用同调或向量乘以实数。这里是线段 [AB] 的中点。



```
\begin{tikzpicture}[scale=1.2]
  \tkzDefPoints{1/3/A,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}
```

13.1.12. Option linear normed

In the following example AC = 1 and C belongs to (AB).



```
\begin{tikzpicture}[scale=1.2]
 \tkzDefPoints{1/3/A,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}
```

13.2. \tkzGetVectxy 获取向量坐标分量

Retrieving the coordinates of a vector.
获取一个向量的坐标。

\tkzGetVectxy(⟨A,B⟩){⟨text⟩}

Allows to obtain the coordinates of a vector.

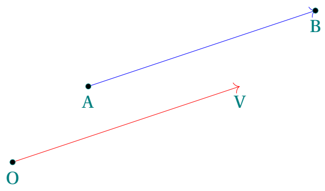
arguments	example	explanation
(point){name of macro}	\tkzGetVectxy(A,B){V}	\Vx,\Vy: coordinates of $\overrightarrow{AB}$

\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}	\Vx,\Vy 向量 $\overrightarrow{AB}$ 的坐标分量

13.2.1. Coordinate transfer with \tkzGetVectxy 实现坐标变换



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/O,1/1/A,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}
```

14. Straight lines 直线

绘制直线当然是必要的，但是在绘制直线之前，有必要能够定义某些特定的线，例如中线、角平分线、平行线甚至垂线。原理是确定直线上的两点。

14.1. Definition of straight lines 直线

```
\tkzDefLine[⟨local options⟩](⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)
```

The argument is a list of two or three points. Depending on the case, the macro defines one or two points necessary to obtain the line sought. Either the macro **\tkzGetPoint** or the macro **\tkzGetPoints** must be used. I used the term ”mediator” to designate the perpendicular bisector line at the middle of a line segment.

arguments	example	explanation
(⟨pt1,pt2⟩)	[mediator](⟨A,B⟩)	mediator of the segment [A,B]
(⟨pt1,pt2,pt3⟩)	[bisector](⟨A,B,C⟩)	bisector of $\widehat{ABC}$
(⟨pt1,pt2,pt3⟩)	[altitude](⟨A,B,C⟩)	altitude from B
(⟨pt1⟩)	[tangent at=A](⟨O⟩)	tangent at A on the circle center O
(⟨pt1,pt2⟩)	[tangent from=A](⟨O,B⟩)	circle center O through B

options	default	definition
mediator		perpendicular bisector of a line segment
perpendicular=through...	mediator	perpendicular to a straight line passing through a point
orthogonal=through...	mediator	see above
parallel=through...	mediator	parallel to a straight line passing through a point
bisector	mediator	bisector of an angle defined by three points
bisector out	mediator	exterior angle bisector
symmedian	mediator	symmedian from a vertex
altitude	mediator	altitude from avertex
euler	mediator	euler line of a triangle
tangent at	mediator	tangent at a point of a circle
tangent from	mediator	tangent from an exterior point
K	1	coefficient for the perpendicular line
normed	false	normalizes the created segment

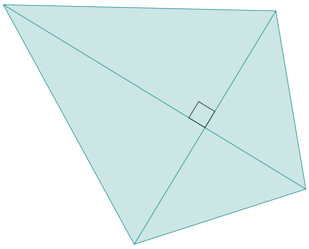
```
\tkzDefLine[⟨local options⟩](⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)
```

参数是 2 个或 3 个点的列表，根据具体问题，该命令得到 1 个或 2 个点，可以使用**\tkzGetPoint** 或**\tkzGetPoints** 命令保存并命名定义的点。用术语“mediator”来表示线段的垂直平分平分线。

参数	样例	说明
(⟨pt1,pt2⟩)	[mediator](⟨A,B⟩)	线段 [A,B] 的中垂线
(⟨pt1,pt2,pt3⟩)	[bisector](⟨A,B,C⟩)	角 $\widehat{ABC}$ 的平分线
(⟨pt1,pt2,pt3⟩)	[altitude](⟨A,B,C⟩)	过点 B 的垂线
(⟨pt1⟩)	[tangent at=A](⟨O⟩)	与圆 O 相切于点 A 处的切线
(⟨pt1,pt2⟩)	[tangent from=A](⟨O,B⟩)	过点 B 与圆 O 的切线

选项	默认值	含义
mediator		线段的垂直平分线
perpendicular=through…	mediator	通过指定点的垂线
orthogonal=through…	mediator	见上
parallel=through…	mediator	通过指定点的平行线
bisector	mediator	由三点定义的角的平分线，即内角角平分线
bisector out	mediator	外角角平分线
symmedian	mediator	顶点的对称线
altitude	mediator	垂线
euler	mediator	欧拉线
tangent at	mediator	过圆上一个的点切线
tangent from	mediator	过圆外一个的点切线
K	1	垂线的比例系数
normed	false	线段归一化

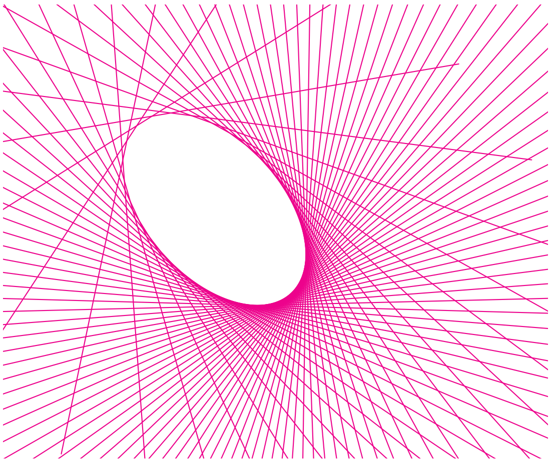
14.1.1. With mediator 中垂线



```
\begin{tikzpicture}[rotate=25]
 \tkzDefPoints{-2/0/A,1/2/B}
 \tkzDefLine[mediator](A,B)           \tkzGetPoints{C}{D}
 \tkzDefPointWith[linear,K=.75](C,D) \tkzGetPoint{D}
 \tkzDefMidPoint(A,B)                \tkzGetPoint{I}
 \tkzFillPolygon[color=teal!20](A,C,B,D)
 \tkzDrawSegments(A,B C,D)
 \tkzMarkRightAngle(B,I,C)
 \tkzDrawSegments(D,B D,A)
 \tkzDrawSegments(C,B C,A)
\end{tikzpicture}
```

14.1.2. An envelope with option mediator 包络

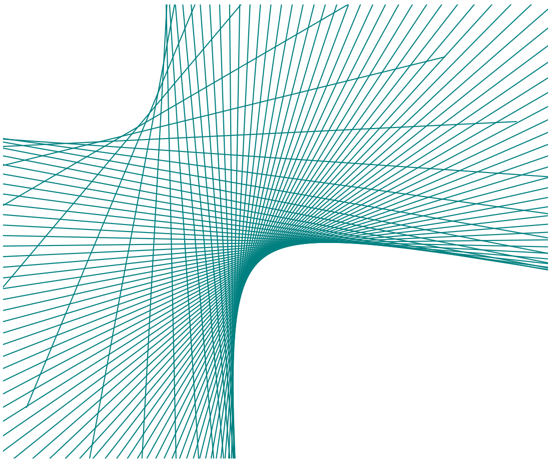
Based on a figure from O. Reboux with pst-eucl by D Rodriguez.
基于 D Rodriguez 用 pst-eucl 宏包绘制的 O. Reboux 设计的图形绘制。



```
\begin{tikzpicture}[scale=.6]
   % necessary
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6]
\tkzClip
\tkzSetUpLine[thin,color=magenta]
\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)
 \tkzGetPoints{x}{y}
 \tkzDrawLine[add= 3 and 3](x,y)}
\end{tikzpicture}
```

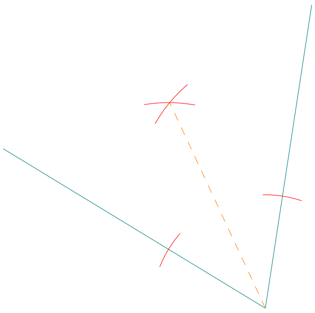
14.1.3. A parabola with option mediator 抛物线

Based on a figure from O. Reboux with pst-eucl by D Rodriguez. It is not necessary to name the two points that define the mediator.
基于 D Rodriguez 用 pst-eucl 宏包绘制的 O. Reboux 设计的图形绘制。对定义的垂直平分线的两个端点进行命名是不必要的。



```
\begin{tikzpicture}[scale=.6]
\tkzInit[xmin=-6,ymin=-4,xmax=6,ymax=6]
\tkzClip
\tkzSetUpLine[thin,color=teal]
\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)
 \tkzGetPoints{x}{y}
 \tkzDrawLine[add= 3 and 3](x,y)}
\end{tikzpicture}
```

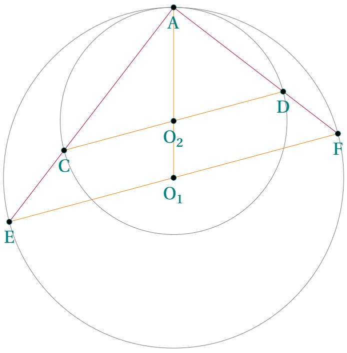
14.1.4. With options bisector and normed 角平分线



```
\begin{tikzpicture}[rotate=25,scale=.75]
 \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[new,dashed,add= 0 and 3](A,a)
\end{tikzpicture}
```

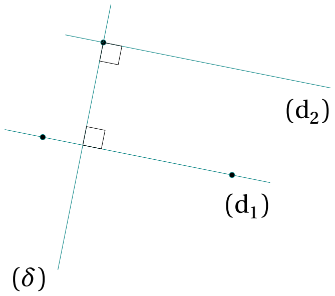
14.1.5. With option parallel=through

Archimedes' Book of Lemmas proposition 1
阿基米德的引理书命题 1



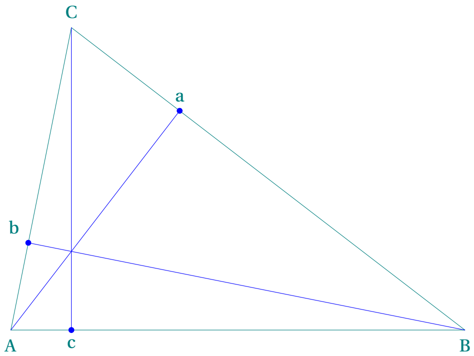
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O_1,0/1/O_2,0/3/A}
  \tkzDefPoint(15:3){F}
  \tkzInterLC(F,O_1)(O_1,A) \tkzGetSecondPoint{E}
  \tkzDefLine[parallel=through O_2](E,F) \tkzGetPoint{x}
  \tkzInterLC(x,O_2)(O_2,A) \tkzGetPoints{D}{C}
  \tkzDrawCircles(O_1,A O_2,A)
  \tkzDrawSegments[new](O_1,A E,F C,D)
  \tkzDrawSegments[purple](A,E A,F)
  \tkzDrawPoints(A,O_1,O_2,E,F,C,D)
  \tkzLabelPoints(A,O_1,O_2,E,F,C,D)
\end{tikzpicture}
```

14.1.6. With option orthogonal and parallel 平行线



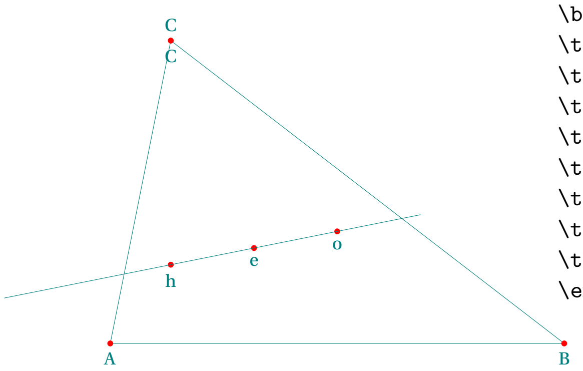
```
\begin{tikzpicture}
   \tkzDefPoints{-1.5/-0.25/A,1/-0.75/B,-0.7/1/C}
   \tkzDrawLine(A,B)
   \tkzLabelLine[pos=1.25,below left](A,B){$(d_1)$}
   \tkzDrawPoints(A,B,C)
   \tkzDefLine[orthogonal=through C](B,A) \tkzGetPoint{c}
   \tkzDrawLine(C,c)
   \tkzLabelLine[pos=1.25,left](C,c){$(\delta)$}
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

14.1.7. With option altitude 高



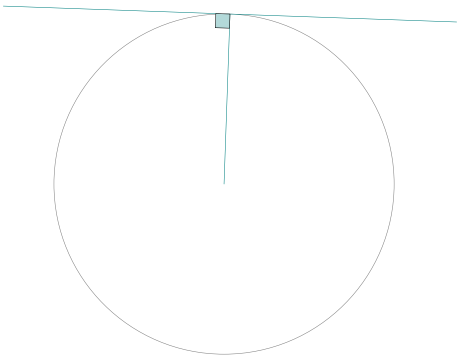
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefLine[altitude](A,B,C)     \tkzGetPoint{b}
\tkzDefLine[altitude](B,C,A)     \tkzGetPoint{c}
\tkzDefLine[altitude](B,A,C)     \tkzGetPoint{a}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints[blue](a,b,c)
\tkzDrawSegments[blue](A,a B,b C,c)
\tkzLabelPoints(A,B,c)
\tkzLabelPoints[above](C,a)
\tkzLabelPoints[above left](b)
\end{tikzpicture}
```

14.1.8. With option euler 欧拉线



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefLine[euler](A,B,C)              \tkzGetPoints{h}{e}
\tkzDefTriangleCenter[circum](A,B,C)  \tkzGetPoint{o}
\tkzDrawPolygon[teal](A,B,C)
\tkzDrawPoints[red](A,B,C,h,e,o)
\tkzDrawLine[add= 2 and 2](h,e)
\tkzLabelPoints(A,B,C,h,e,o)
\tkzLabelPoints[above](C)
\end{tikzpicture}
```

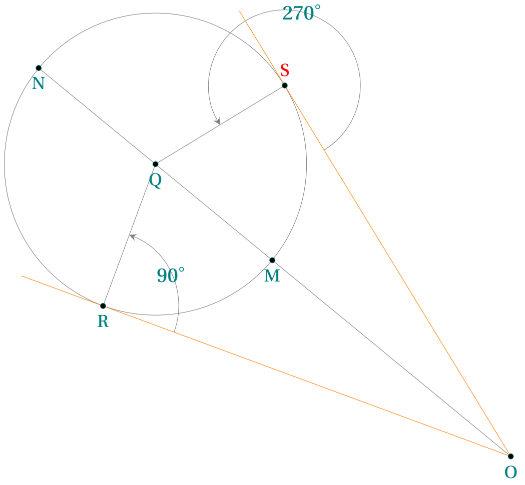
14.1.9. Tangent passing through a point on the circle **tangent at** 过圆上一点的切线



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(6,6){E}
  \tkzDefRandPointOn[circle=center O radius 3]
  \tkzGetPoint{A}
  \tkzDrawSegment(O,A)
  \tkzDrawCircle(O,A)
  \tkzDefLine[tangent at=A](O)
  \tkzGetPoint{h}
  \tkzDrawLine[add = 4 and 3](A,h)
  \tkzMarkRightAngle[fill=teal!30](O,A,h)
\end{tikzpicture}
```

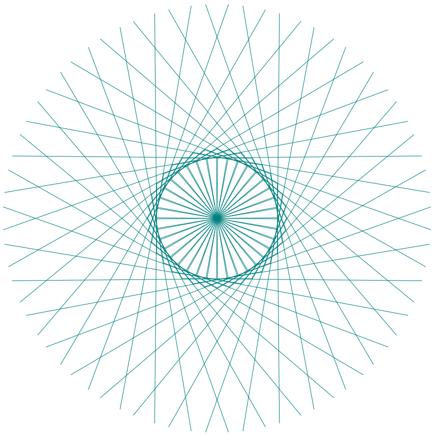
14.1.10. Choice of contact point with tangents passing through an external point option **tangent from** 过圆外一点的切线

The tangent is not drawn. With option `at`, a point of the tangent is given by `tkzPointResult`. With option `from` you get two points of the circle with `tkzFirstPointResult` and `tkzSecondPointResult`. You can choose between these two points by comparing the angles formed with the outer point, the contact point and the center. The two possible angles have different directions. Angle counterclockwise refers to `tkzFirstPointResult`.
不绘制切线。使用选项 `at`，切线的点由 `tkzPointResult` 给出。使用选项 `from` 可以获得圆的两个点，分别为 `tkzFirstPointResult` 和 `tkzSecondPointResult`。通过比较圆外的点、切点和圆心形成的角度，可以在这两点之间进行选择。这两个可能的角度具有不同的方向。`tkzFirstPointResult` 获得逆时针方向的点。



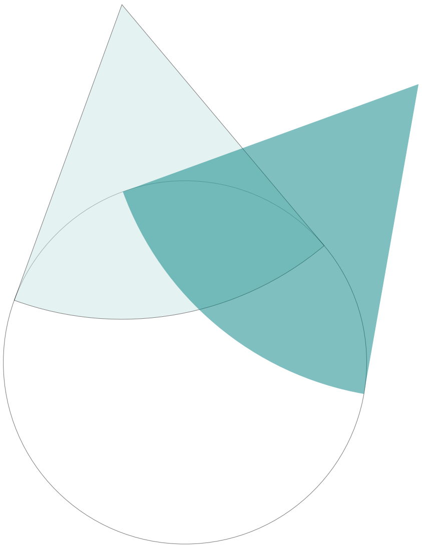
```
\begin{tikzpicture}[scale=1,rotate=-30]
\tkzDefPoints{0/0/Q,0/2/A,6/-1/O}
\tkzDefLine[tangent from = O](Q,A)  \tkzGetPoints{R}{S}
\tkzInterLC[near](O,Q)(Q,A)         \tkzGetPoints{M}{N}
\tkzDrawCircle(Q,M)
\tkzDrawSegments[new,add = 0 and .2](O,R O,S)
\tkzDrawSegments[gray](N,O R,Q S,Q)
\tkzDrawPoints(O,Q,R,S,M,N)
\tkzMarkAngle[gray,-stealth,size=1](O,R,Q)
\tkzFindAngle(O,R,Q)   \tkzGetAngle{an}
\tkzLabelAngle(O,R,Q){$\pgfmathprintnumber{\an}^\circ$}
\tkzMarkAngle[gray,-stealth,size=1](O,S,Q)
\tkzFindAngle(O,S,Q)   \tkzGetAngle{an}
\tkzLabelAngle(O,S,Q){$\pgfmathprintnumber{\an}^\circ$}
\tkzLabelPoints(Q,O,M,N,R)
\tkzLabelPoints[above,text=red](S)
\end{tikzpicture}
```

14.1.11. Example of tangents passing through an external point 过圆外指定点的切线示例



```
\begin{tikzpicture}[scale=.8]
\tkzDefPoints{0/0/c,1/0/d,3/0/a0}
\def\tkzRadius{1}
\tkzDrawCircle(c,d)
 \foreach \an in {0,10,...,350}{
  \tkzDefPointBy[rotation=center c angle \an](a0)
  \tkzGetPoint{a}
  \tkzDefLine[tangent from = a](c,d)
  \tkzGetPoints{e}{f}
  \tkzDrawLines(a,f a,e)
  \tkzDrawSegments(c,e c,f)}
\end{tikzpicture}
```

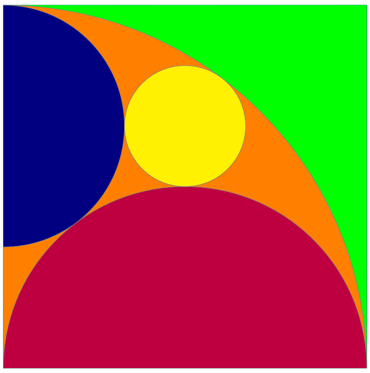
14.1.12. Example of Andrew Mertz



```
 \begin{tikzpicture}[scale=.6]
 \tkzDefPoint(100:8){A}\tkzDefPoint(50:8){B}
 \tkzDefPoint(0,0){C} \tkzDefPoint(0,-4){R}
 \tkzDrawCircle(C,R)
 \tkzDefLine[tangent from = A](C,R)  \tkzGetPoints{D}{E}
\tkzDefLine[tangent from = B](C,R)  \tkzGetPoints{F}{G}
 \tkzDrawSector[fill=teal!20,opacity=0.5](A,E)(D)
 \tkzFillSector[color=teal,opacity=0.5](B,G)(F)
 \end{tikzpicture}
```

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

14.1.13. Drawing a tangent option tangent from



```
\begin{tikzpicture}[scale=.6]
 \tkzDefPoint(0,0){B}
 \tkzDefPoint(0,8){A}
 \tkzDefSquare(A,B)
 \tkzGetPoints{C}{D}
 \tkzDrawPolygon(A,B,C,D)
 \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)
 \tkzDefLine[tangent from = B](F,A)
 \tkzInterLL(F,tkzSecondPointResult)(C,D)
 \tkzInterLL(A,tkzPointResult)(F,E)
 \tkzDrawCircle[fill = yellow](tkzPointResult,Q)
 \tkzDefPointBy[projection= onto B--A](tkzPointResult)
 \tkzDrawCircle[fill = blue!50!black](tkzPointResult,A)
\end{tikzpicture}
```

15. Triangles 定义与绘制三角形

15.1. Definition of triangles \tkzDefTriangle 定义三角形

The following macros will allow you to define or construct a triangle from `at least` two points.
三角形定义命令允许使用“至少”2 个点构造一个三角形。
At the moment, it is possible to define the following triangles:
可以按如下方式定义三角形：

- `two angles` determines a triangle with two angles; 已知 2 个角的三角形;
- `equilateral` determines an equilateral triangle;
- `isosceles right` determines an isoxsceles right triangle; 等边三角形;
- `half` determines a right-angled triangle such that the ratio of the measurements of the two adjacent sides to the right angle is equal to 2; 直角边与斜边之比等于 2 的直角三角形;
- `pythagore` determines a right-angled triangle whose side measurements are proportional to 3, 4 and 5; 勾股 3、4、5 直角三角形;
- `school` determines a right-angled triangle whose angles are 30, 60 and 90 degrees; 三个角分别是 30、60 和 90 的直角三角形;
- `golden` determines a right-angled triangle such that the ratio of the measurements on the two adjacent sides to the right angle is equal to $\Phi = 1.618034$, I chose ”golden triangle” as the denomination because it comes from the golden rectangle and I kept the denomination ”gold triangle” or ”Euclid’s triangle” for the isosceles triangle whose angles at the base are 72 degrees; 直角边与斜边比等于黄金分割比 $\Phi = 1.618034$ 的直角三角形，保留了等腰三角形中“黄金三角形”或“欧几里得三角形”，其底角为 72 度；
- `euclid` or `gold` for the gold triangle; in the previous version the option was ”euclide” with an ”e”. “欧几里得三角形”或“黄金三角形”，前一版本中，参数是带”e” 的”euclide”；
- `cheops` determines a third point such that the triangle is isosceles with side measurements proportional to 2, Φ and Φ. 直角边与斜边比等于黄金分割比 $\Phi = 1.618034$ 的直角三角形，

```
\tkzDefTriangle[⟨local options⟩](⟨A,B⟩)
```

The points are ordered because the triangle is constructed following the direct direction of the trigonometric circle. This macro is either used in partnership with `\tkzGetPoint` or by using `tkzPointResult` if it is not necessary to keep the name.

options	default	definition
two angles= #1 and #2	no defaut	triangle knowing two angles
equilateral	equilateral	equilateral triangle
half	equilateral	B rectangle $AB = 2BC$ AC hypothenuse
isosceles right	equilateral	isosceles right triangle
pythagore	equilateral	proportional to the pythagorean triangle 3-4-5
pythagoras	equilateral	same as above
egyptian	equilateral	same as above
school	equilateral	angles of 30, 60 and 90 degrees
gold	equilateral	B rectangle and $AB/AC = \Phi$
euclid	equilateral	angles of 72, 72 and 36 degrees, A is the apex
golden	equilateral	angles of 72, 72 and 36 degrees, C is the apex
sublime	equilateral	angles of 72, 72 and 36 degrees, C is the apex
cheops	equilateral	AC=BC, AC and BC are proportional to 2 and Φ.
swap	false	gives the symmetric point with respect to AB

`\tkzGetPoint` allows you to store the point otherwise `tkzPointResult` allows for immediate use.

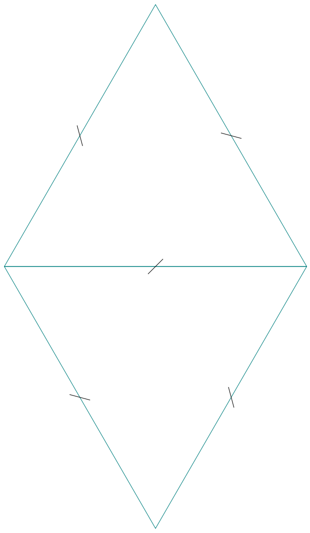
```
\tkzDefTriangle[⟨local options⟩](⟨A,B⟩)
```

这些点是有序的，因为三角形是按照三角圆的直接方向构造的。此宏可以与 `\tkzGetPoint` 一起使用，或者如果不命名，则使用 `tkzPointResult`。

选项	默认值	含义
two angles= #1 and #2	无	三角形两个已知角
equilateral	equilateral	等边三角形
half	equilateral	B 为直角 $AB = 2BC$ AC 斜边
isosceles right	equilateral	等腰直角三角形
pythagore	equilateral	与直角三角形三边比 3-4-5
pythagoras	equilateral	同上
egyptian	equilateral	同上
school	equilateral	30、60 和 90 度的三角形
gold	equilateral	B 为直角和 $AB/AC = /Phi$
euclid	equilateral	角度为 72、72 和 36 度，A 为顶点
golden	equilateral	角度为 72、72 和 36 度，C 为顶点
sublime	equilateral	角度为 72、72 和 36 度，C 为顶点
cheops	equilateral	AC=BC，AC 和 BC 与 2 和 Phi 成正比。
swap	false	给出相对于 AB 的对称点

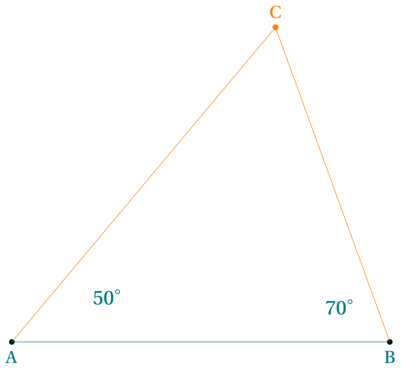
使用`\tkzGetPoint` 保存并命名得到的点，使用 `tkzPointResult` 命令使用得到的点，但不命名。。

15.1.1. Option equilateral



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefTriangle[equilateral](A,B)
  \tkzGetPoint{C}
  \tkzDrawPolygons(A,B,C)
  \tkzDefTriangle[equilateral](B,A)
  \tkzGetPoint{D}
  \tkzDrawPolygon(B,A,D)
  \tkzMarkSegments[mark=s|](A,B B,C A,C A,D B,D)
\end{tikzpicture}
```

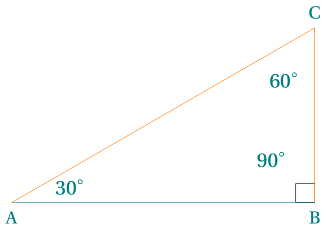
15.1.2. Option two angles



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,0){B}
\tkzDefTriangle[two angles = 50 and 70](A,B)
\tkzGetPoint{C}
\tkzDrawSegment(A,B)
\tkzDrawPoints(A,B)
\tkzLabelPoints(A,B)
\tkzDrawSegments[new](A,C B,C)
\tkzDrawPoints[new](C)
\tkzLabelPoints[above,new](C)
\tkzLabelAngle[pos=1.4](B,A,C){$50^\circ$}
\tkzLabelAngle[pos=0.8](C,B,A){$70^\circ$}
\end{tikzpicture}
```

15.1.3. Option school

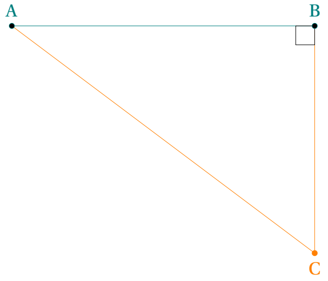
The angles are 30, 60 and 90 degrees.
三角形内角是 30 度、60 度和 90 度。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDefTriangle[school](A,B)
  \tkzGetPoint{C}
  \tkzMarkRightAngles(C,B,A)
  \tkzLabelAngle[pos=0.8](B,A,C){$30^\circ$}
  \tkzLabelAngle[pos=0.8](C,B,A){$90^\circ$}
  \tkzLabelAngle[pos=0.8](A,C,B){$60^\circ$}
  \tkzDrawSegments(A,B)
  \tkzDrawSegments[new](A,C B,C)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above](C)
\end{tikzpicture}
```

15.1.4. Option pythagore

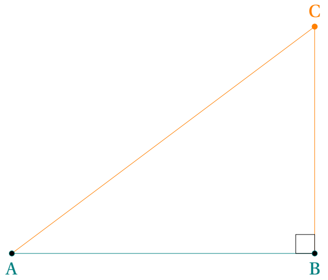
This triangle has sides whose lengths are proportional to 3, 4 and 5.
三角形的三边边长与 3、4 和 5 成正比。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDefTriangle[pythagore](A,B)
  \tkzGetPoint{C}
  \tkzDrawSegments(A,B)
  \tkzDrawSegments[new](A,C B,C)
  \tkzMarkRightAngles(A,B,C)
  \tkzDrawPoints[new](C)
  \tkzDrawPoints(A,B)
  \tkzLabelPoints[above](A,B)
  \tkzLabelPoints[new](C)
\end{tikzpicture}
```

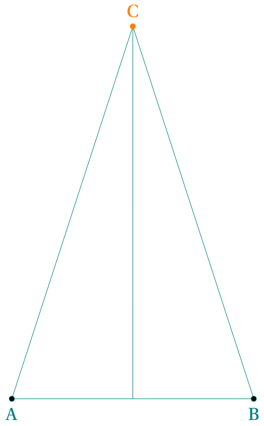
15.1.5. Option pythagore and swap

This triangle has sides whose lengths are proportional to 3, 4 and 5.
三角形的三边边长与 3、4 和 5 成正比。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDefTriangle[pythagore,swap](A,B)
  \tkzGetPoint{C}
  \tkzDrawSegments(A,B)
  \tkzDrawSegments[new](A,C B,C)
  \tkzMarkRightAngles(A,B,C)
  \tkzLabelPoint[above,new](C){$C$}
  \tkzDrawPoints[new](C)
  \tkzDrawPoints(A,B)
  \tkzLabelPoints(A,B)
\end{tikzpicture}
```

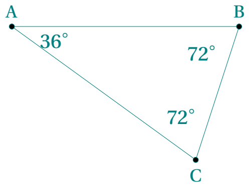
15.1.6. Option golden



```
\begin{tikzpicture}[scale=.8]
\tkzDefPoint(0,0){A} \tkzDefPoint(4,0){B}
\tkzDefTriangle[golden](A,B)\tkzGetPoint{C}
\tkzDefSpcTriangle[in,name=M](A,B,C){a,b,c}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B)
\tkzDrawSegment(C,Mc)
\tkzDrawPoints[new](C)
\tkzLabelPoints(A,B)
\tkzLabelPoints[above,new](C)
\end{tikzpicture}
```

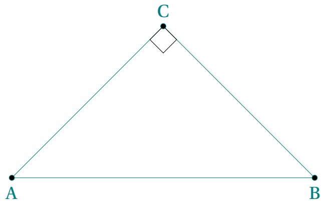
15.1.7. Option euclid

Euclid and **golden** are identical but the segment AB is a base in one and a side in the other.
Euclid 和 **golden** 是相同的，但是线段 AB 一个是底，另一个是边。



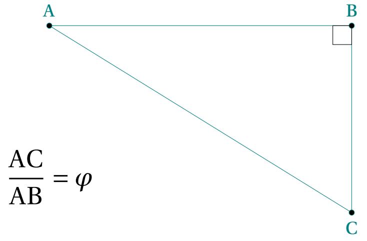
```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(0,0){A} \tkzDefPoint(4,0){B}
 \tkzDefTriangle[euclid](A,B)\tkzGetPoint{C}
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPoints(A,B,C)
 \tkzLabelPoints(C)
 \tkzLabelPoints[above](A,B)
 \tkzLabelAngle[pos=0.8](A,B,C){$72^\circ$}
 \tkzLabelAngle[pos=0.8](B,C,A){$72^\circ$}
 \tkzLabelAngle[pos=0.8](C,A,B){$36^\circ$}
\end{tikzpicture}
```

15.1.8. Option isosceles right



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefTriangle[isosceles right](A,B)
  \tkzGetPoint{C}
  \tkzDrawPolygons(A,B,C)
  \tkzDrawPoints(A,B,C)
  \tkzMarkRightAngles(A,C,B)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above](C)
\end{tikzpicture}
```

15.1.9. Option gold



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B}
 \tkzDefTriangle[gold](A,B)
 \tkzGetPoint{C}
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPoints(A,B,C)
 \tkzLabelPoints[above](A,B)
 \tkzLabelPoints[below](C)
 \tkzMarkRightAngle(A,B,C)
 \tkzText(0,-2){$\dfrac{AC}{AB}=\varphi$}
\end{tikzpicture}
```

15.2. Specific triangles with \tkzDefSpcTriangle 定义特殊三角形

The centers of some triangles have been defined in the ”points” section, here it is a question of determining the three vertices of specific triangles.
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[⟨local options⟩](⟨p1,p2,p3⟩){⟨r1,r2,r3⟩}
```

The order of the points is important! p1p2p3 defines a triangle then the result is a triangle whose vertices have as reference a combination with **name** and r1,r2, r3. If **name** is empty then the references are r1,r2 and r3.

options	default	definition
orthic	centroid	determined by endpoints of the altitudes ...
centroid or medial	centroid	intersection of the triangle's three triangle medians
in or incentral	centroid	determined with the angle bisectors
ex or excentral	centroid	determined with the excenters
extouch	centroid	formed by the points of tangency with the excircles
intouch or contact	centroid	formed by the points of tangency of the incircle each of the vertices
euler	centroid	formed by Euler points on the nine-point circle
symmedial	centroid	intersection points of the symmedians
tangential	centroid	formed by the lines tangent to the circumcircle
feuerbach	centroid	formed by the points of tangency of the nine-point ... circle with the excircles
name	empty	used to name the vertices

```
\tkzDefSpcTriangle[⟨local options⟩](⟨p1,p2,p3⟩){⟨r1,r2,r3⟩}
```

点的顺序很重要！p1p2p3 定义了一个三角形，则结果是一个顶点以 **name** 和 r1、r2、r3 的组合作为参考的三角形。如果 **name** 为空，则引用是 r1、r2 和 r3。

选项	默认值	含义
orthic	centroid	由高的垂足决定...
centroid or medial	centroid	三角形的三条三角形中线的交点
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		内切三角形
euler	centroid	欧拉三角形
symmedial		对称中线的交点
tangential	centroid	由外接圆的切线形成的，切线三角形
feuerbach	centroid	九点圆与旁切圆的切点构成的三角形
name	空	顶点命名前缀

15.2.1. How to name the vertices 如何命名顶点

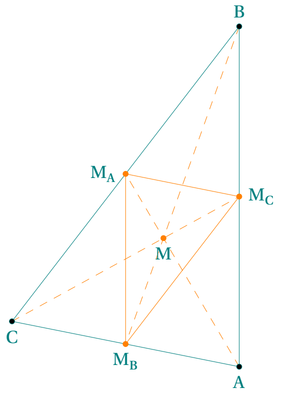
With `\tkzDefSpcTriangle[medial,name=M](A,B,C){_A,_B,_C}` you get three vertices named M_A, M_B and M_C.
使用 `\tkzDefSpcTriangle[medial,name=M](A,B,C){_A,_B,_C}` 可以得到三个顶点，分别命名为 M_A、M_B 和 M_C。

With \tkzDefSpcTriangle[medial](A,B,C){a,b,c} you get three vertices named and labeled a, b and c.
使用 \tkzDefSpcTriangle[medial](A,B,C){a,b,c} 可以获得三个顶点，它们被命名并标记为 a 、b 和 c。
Possible \tkzDefSpcTriangle[medial,name=M_](A,B,C){A,B,C} you get three vertices named M_A, M_B and M_C.
使用 \tkzDefSpcTriangle[medial,name=M_](A,B,C){A,B,C} 得到三个顶点，分别命名为 M_A、M_B 和 M_C。

15.3. Option medial or centroid

The geometric centroid of the polygon vertices of a triangle is the point G (sometimes also denoted M) which is also the intersection of the triangle’s three triangle medians. The point is therefore sometimes called the median point. The centroid is always in the interior of the triangle.
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交点，该点也称为重心，重心总是位于三角形内部。

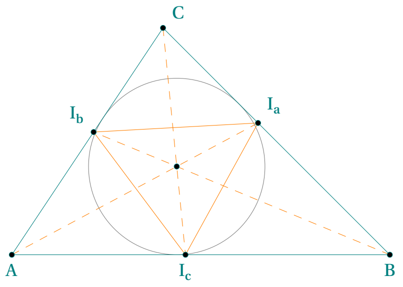
Weisstein, Eric W. ”Centroid triangle” From MathWorld–A Wolfram Web Resource.
In the following example, we obtain the Euler circle which passes through the previously defined points.
下面的例子中，通过预先定义的点，得到通过这些点的欧拉圆。



```
\begin{tikzpicture}[rotate=90,scale=.75]
 \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(A,B,C)
 \tkzDrawSegments[dashed,new](A,M_A B,M_B C,M_C)
 \tkzDrawPolygon[new](M_A,M_B,M_C)
 \tkzDrawPoints(A,B,C)
 \tkzDrawPoints[new](M,M_A,M_B,M_C)
 \tkzLabelPoints[above](B)
 \tkzLabelPoints[below](A,C,M_B)
 \tkzLabelPoints[right](M_C)
 \tkzLabelPoints[left](M_A)
 \tkzLabelPoints[font=\scriptsize](M)
\end{tikzpicture}
```

15.3.1. Option in or incentral

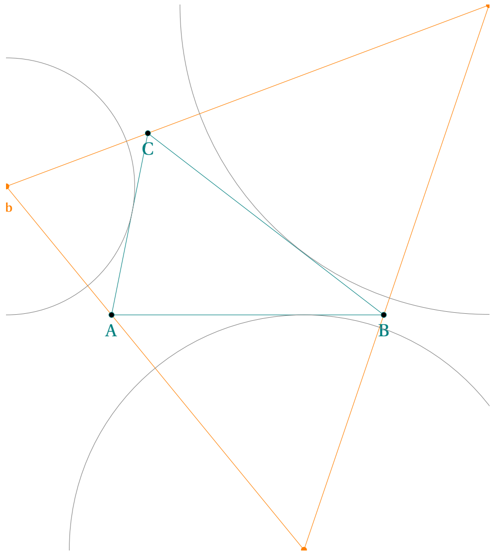
The incentral triangle is the triangle whose vertices are determined by the intersections of the reference triangle’s angle bisectors with the respective opposite sides.
内心三角形是由一个三角形的三个内角平分线与对边交点确定的三角形。
Weisstein, Eric W. ”Incentral triangle” From MathWorld–A Wolfram Web Resource.



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{ 0/0/A,5/0/B,2/3/C}
  \tkzDefSpcTriangle[in,name=I](A,B,C){_a,_b,_c}
  \tkzDefCircle[in](A,B,C) \tkzGetPoints{I}{a}
  \tkzDrawCircle(I,a)
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPolygon[new](I_a,I_b,I_c)
  \tkzDrawSegments[dashed,new](A,I_a B,I_b C,I_c)
  \tkzDrawPoints(A,B,C,I,I_a,I_b,I_c)
  \tkzLabelPoints[below](A,B,I_c)
  \tkzLabelPoints[above left](I_b)
  \tkzLabelPoints[above right](C,I_a)
\end{tikzpicture}
```

15.3.2. Option ex or excentral

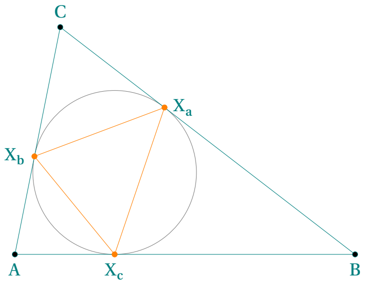
The excentral triangle of a triangle ABC is the triangle $J_aJ_bJ_c$ with vertices corresponding to the excenters of ABC.
旁心三角形是由一个三角形 ABC 的三个旁心 $J_aJ_bJ_c$ 构成的三角形。



```
\begin{tikzpicture}[scale=.6]
 \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(A,B,C)
 \tkzDrawPolygon[new](J_a,J_b,J_c)
 \tkzClipBB
 \tkzDrawPoints(A,B,C)
 \tkzDrawPoints[new](J_a,J_b,J_c)
 \tkzLabelPoints(A,B,C)
 \tkzLabelPoints[new](J_b,J_c)
 \tkzLabelPoints[new,above](J_a)
 \tkzDrawCircles[gray](J_a,T_a J_b,T_b J_c,T_c)
\end{tikzpicture}
```

15.3.3. Option intouch or contact

The contact triangle of a triangle ABC, also called the intouch triangle, is the triangle formed by the points of tangency of the incircle of ABC with ABC.
内接三角形是由三角形的内切圆的三个切点构成的三角形。
Weisstein, Eric W. "Contact triangle" From MathWorld–A Wolfram Web Resource.
We obtain the intersections of the bisectors with the sides.
可以得到平分线与边的交点。

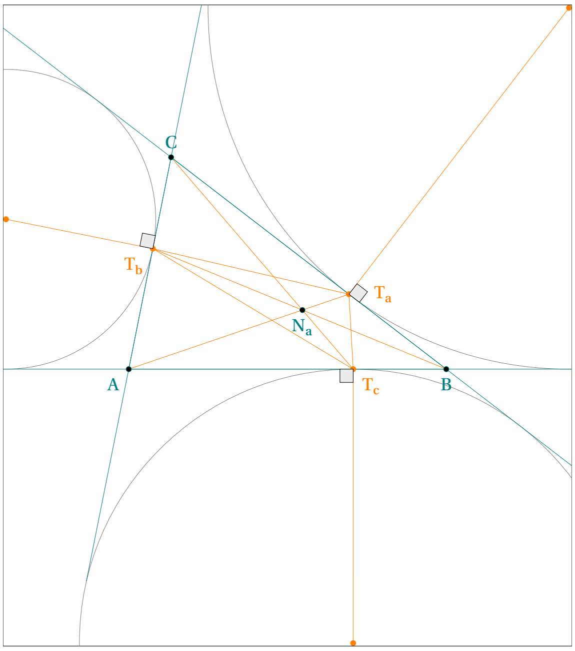


```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
 \tkzDefSpcTriangle[intouch,name=X](A,B,C){_a,_b,_c}
 \tkzInCenter(A,B,C)\tkzGetPoint{I}
 \tkzDefCircle[in](A,B,C) \tkzGetPoints{I}{i}
 \tkzDrawCircle(I,i)
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPolygon[new](X_a,X_b,X_c)
 \tkzDrawPoints(A,B,C)
 \tkzDrawPoints[new](X_a,X_b,X_c)
 \tkzLabelPoints[right](X_a)
 \tkzLabelPoints[left](X_b)
 \tkzLabelPoints[above](C)
 \tkzLabelPoints[below](A,B,X_c)
\end{tikzpicture}
```

15.3.4. Option extouch

The extouch triangle $T_aT_bT_c$ is the triangle formed by the points of tangency of a triangle ABC with its excircles J_a, J_b, and J_c. The points T_a, T_b, and T_c can also be constructed as the points which bisect the perimeter of $A_1A_2A_3$ starting at A, B, and C.
外切三角形 $T_aT_bT_c$ 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。T_a、T_b 和 T_c 这三个点也可以由从 A、B 和 C 开始 $A_1A_2A_3$ 的周长的中点得到。
Weisstein, Eric W. "Extouch triangle" From MathWorld–A Wolfram Web Resource.

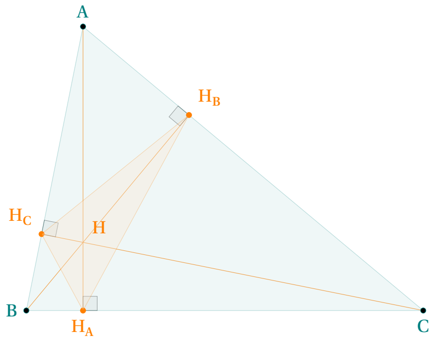
We obtain the points of contact of the exinscribed circles as well as the triangle formed by the centers of the exinscribed circles.
可以得到旁切圆的切点和由这三个切点构成的三角形。



```
\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[new](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[new](A,T_a B,T_b C,T_c)
\tkzDrawSegments[new](J_a,T_a J_b,T_b J_c,T_c)
\tkzDrawPolygon(A,B,C)
\tkzDrawPolygon[new](T_a,T_b,T_c)
\tkzDrawPoints(A,B,C,N_a)
\tkzDrawPoints[new](T_a,T_b,T_c)
\tkzLabelPoints[below left](A)
\tkzLabelPoints[below](N_a,B)
\tkzLabelPoints[above](C)
\tkzLabelPoints[new,below left](T_b)
\tkzLabelPoints[new,below right](T_c)
\tkzLabelPoints[new,right=6pt](T_a)
\tkzMarkRightAngles[fill=gray!15](J_a,T_a,B
 J_b,T_b,C J_c,T_c,A)
\end{tikzpicture}
```

15.3.5. Option orthic

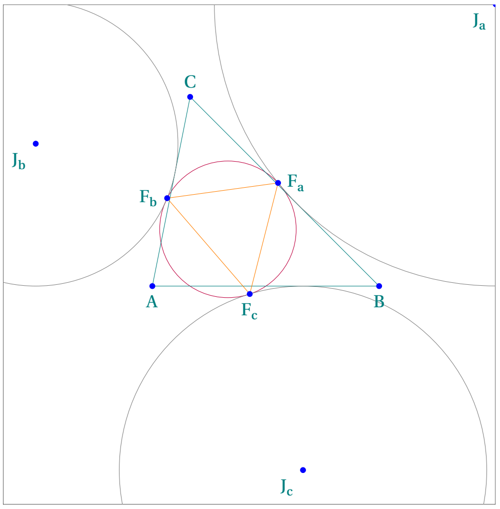
Given a triangle ABC, the triangle $H_AH_BH_C$ whose vertices are endpoints of the altitudes from each of the vertices of ABC is called the orthic triangle, or sometimes the altitude triangle. The three lines AH_A, BH_B, and CH_C are concurrent at the orthocenter H of ABC.
给定一个三角形 ABC，其顶点是从 ABC 的每个顶点开始的高的垂足的三角形 $H_AH_BH_C$ 称为 orthic 三角形，有时也称为 altitude 三角形。三条线 AH_A, BH_B, 和 CH_C 在 ABC 的垂心 H 处重合。



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{1/5/A,0/0/B,7/0/C}
 \tkzDefSpcTriangle[orthic](A,B,C){H_A,H_B,H_C}
 \tkzDefTriangleCenter[ortho](B,C,A)
 \tkzGetPoint{H}
 \tkzDefPointWith[orthogonal,normed](H_A,B)
 \tkzGetPoint{a}
 \tkzDrawSegments[new](A,H_A B,H_B C,H_C)
 \tkzMarkRightAngles[fill=gray!20,
         opacity=.5](A,H_A,C B,H_B,A C,H_C,A)
 \tkzDrawPolygon[fill=teal!20,opacity=.3](A,B,C)
 \tkzDrawPoints(A,B,C)
 \tkzDrawPoints[new](H_A,H_B,H_C)
 \tkzDrawPolygon[new,fill=orange!20,
                opacity=.3](H_A,H_B,H_C)
 \tkzLabelPoints(C)
 \tkzLabelPoints[left](B)
 \tkzLabelPoints[above](A)
 \tkzLabelPoints[new](H_A)
 \tkzLabelPoints[new,above left](H_C)
 \tkzLabelPoints[new,above right](H_B,H)
\end{tikzpicture}
```

15.3.6. Option feuerbach

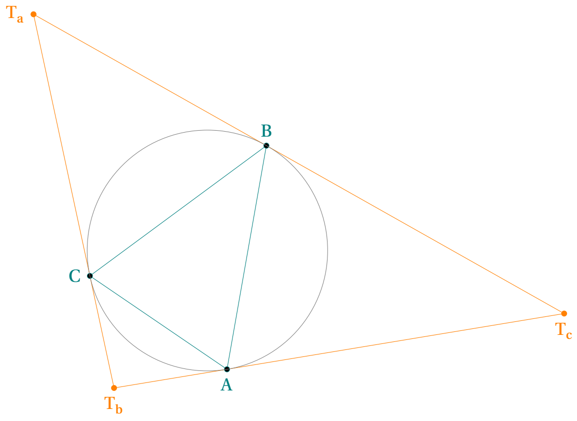
The Feuerbach triangle is the triangle formed by the three points of tangency of the nine-point circle with the excircles.
费尔巴哈三角形是由九点圆与三个旁切圆的 3 个切点构成的三角形。
Weisstein, Eric W. ”Feuerbach triangle” From MathWorld–A Wolfram Web Resource.
The points of tangency define the Feuerbach triangle.
切点定义了费尔巴哈三角形。



```
\begin{tikzpicture}[scale=1]
  \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}
  \tkzLabelPoints[below left](J_a,J_b,J_c)
  \tkzClipBB \tkzShowBB
  \tkzDrawCircle[purple](N,F_a)
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPolygon[new](F_a,F_b,F_c)
  \tkzDrawCircles[gray](J_a,F_a J_b,F_b J_c,F_c)
  \tkzDrawPoints[blue](J_a,J_b,J_c,%
          F_a,F_b,F_c,A,B,C)
  \tkzLabelPoints(A,B,F_c)
  \tkzLabelPoints[above](C)
  \tkzLabelPoints[right](F_a)
  \tkzLabelPoints[left](F_b)
\end{tikzpicture}
```

15.3.7. Option tangential

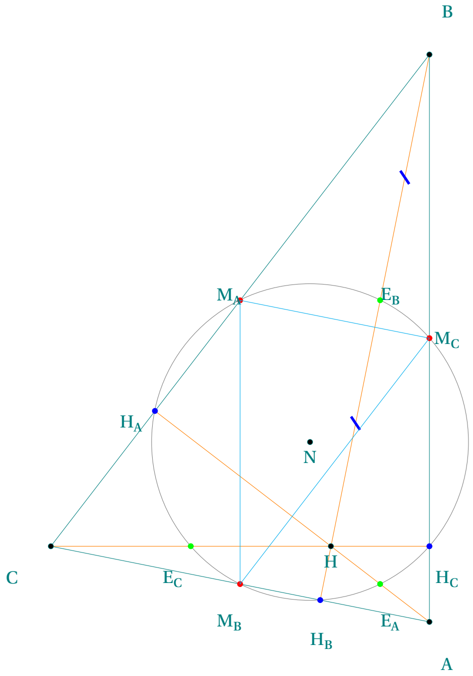
The tangential triangle is the triangle $T_aT_bT_c$ formed by the lines tangent to the circumcircle of a given triangle ABC at its vertices. It is therefore antipedal triangle of ABC with respect to the circumcenter O.
切向三角形是三角形 ABC 外接圆在三个顶点处的切线构成的三角形 $T_aT_bT_c$。它是相对于三角形 ABC 外心 O 的反三角形。
Weisstein, Eric W. "Tangential Triangle." From MathWorld–A Wolfram Web Resource.



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

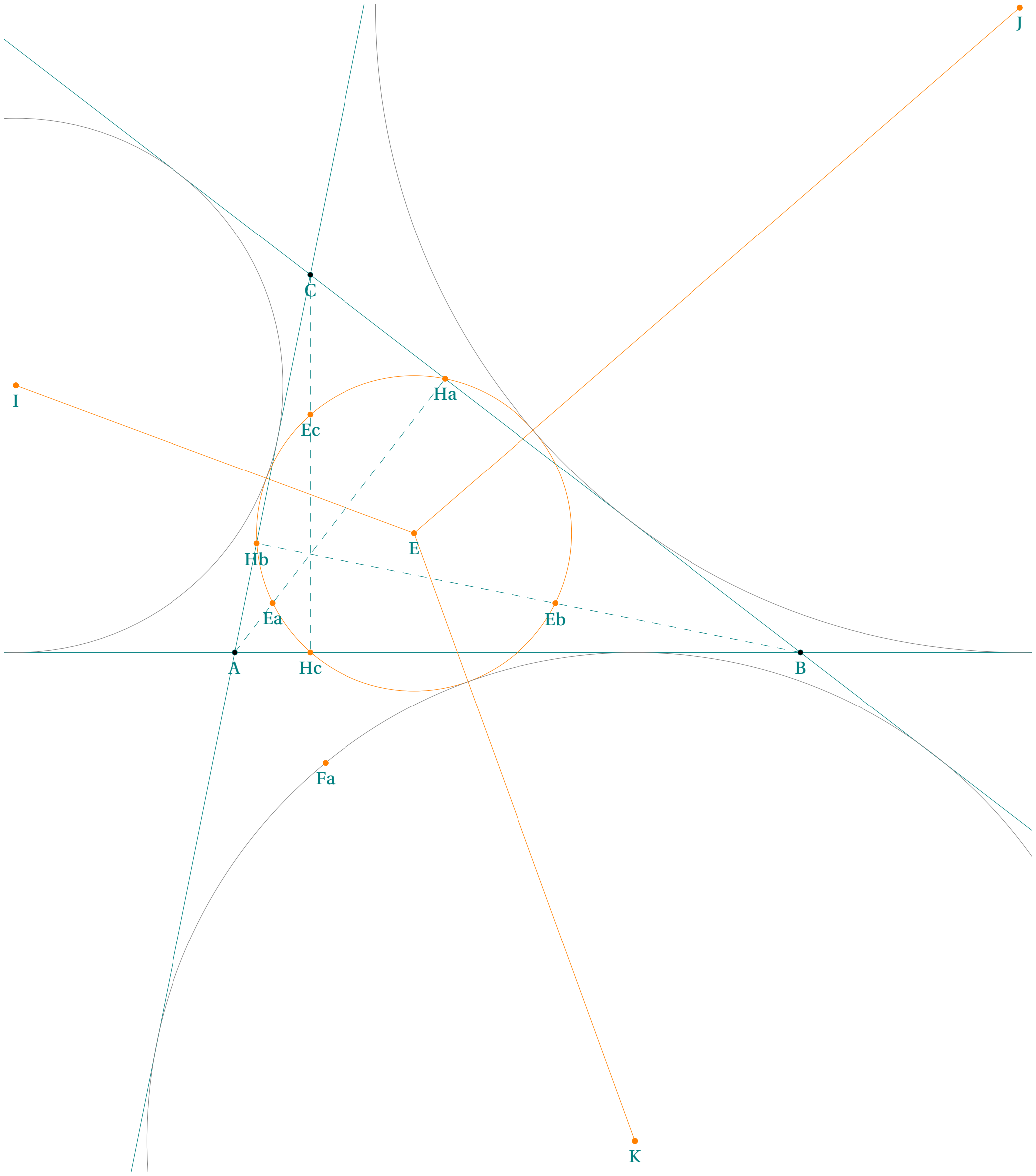
15.3.8. Option euler

The Euler triangle of a triangle ABC is the triangle $E_AE_BE_C$ whose vertices are the midpoints of the segments joining the orthocenter H with the respective vertices. The vertices of the triangle are known as the Euler points, and lie on the nine-point circle.
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 $E_AE_BE_C$，欧拉三角形的顶点是欧拉点，它们位于三角形的九点圆上。
Weisstein, Eric W. "Euler Triangle." From MathWorld–A Wolfram Web Resource.



```
\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(A,B,C)
 \tkzDrawCircle(N,E_A)
 \tkzDrawSegments[new](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=cyan](M_A,M_B,M_C)
\end{tikzpicture}
```

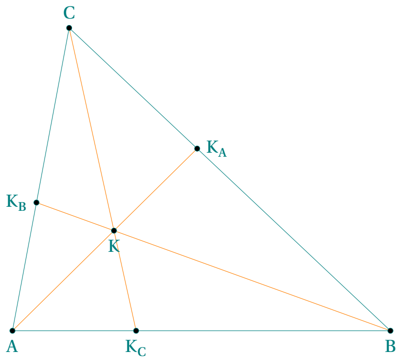
15.3.9. Option euler and Option orthic



```
\begin{tikzpicture}[scale=1.25]
  \tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
  \tkzDefSpcTriangle[euler,name=E](A,B,C){a,b,c}
  \tkzDefSpcTriangle[orthic,name=H](A,B,C){a,b,c}
  \tkzDefExCircle(A,B,C) \tkzGetPoints{I}{i}
  \tkzDefExCircle(C,A,B) \tkzGetPoints{J}{j}
  \tkzDefExCircle(B,C,A) \tkzGetPoints{K}{k}
  \tkzDrawPoints[orange](I,J,K)
  \tkzLabelPoints[font=\scriptsize](A,B,C,I,J,K)
  \tkzClipBB
  \tkzInterLC(I,C)(I,i) \tkzGetSecondPoint{Fc}
  \tkzInterLC(J,B)(J,j) \tkzGetSecondPoint{Fb}
  \tkzInterLC(K,A)(K,k) \tkzGetSecondPoint{Fa}
  \tkzDrawLines[add=1.5 and 1.5](A,B A,C B,C)
  \tkzDefCircle[euler](A,B,C) \tkzGetPoints{E}{e}
  \tkzDrawCircle[orange](E,e)
  \tkzDrawSegments[orange](E,I E,J E,K)
  \tkzDrawSegments[dashed](A,Ha B,Hb C,Hc)
  \tkzDrawCircles(J,j I,i K,k)
  \tkzDrawPoints(A,B,C)
  \tkzDrawPoints[orange](E,I,J,K,Ha,Hb,Hc,Ea,Eb,Ec,Fa,Fb,Fc)
  \tkzLabelPoints[font=\scriptsize](E,Ea,Eb,Ec,Ha,Hb,Hc,Fa,Fb,Fc)
\end{tikzpicture}
```

15.3.10. Option symmedial

The symmedial triangle$K_AK_BK_C$ is the triangle whose vertices are the intersection points of the symmedians with the reference triangle ABC.
对称三角形 $K_AK_BK_C$ 的顶点是与三角形 ABC 对称的交点。



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(5,0){B}
\tkzDefPoint(.75,4){C}
\tkzDefTriangleCenter[symmedian](A,B,C)\tkzGetPoint{K}
\tkzDefSpcTriangle[symmedial,name=K_](A,B,C){A,B,C}
\tkzDrawPolygon(A,B,C)
\tkzDrawSegments[new](A,K_A B,K_B C,K_C)
\tkzDrawPoints(A,B,C,K,K_A,K_B,K_C)
\tkzLabelPoints(A,B,K,K_C)
\tkzLabelPoints[above](C)
\tkzLabelPoints[right](K_A)
\tkzLabelPoints[left](K_B)
\end{tikzpicture}
```

15.4. Permutation of two points of a triangle 三角形两点的排列组合

\tkzPermute(⟨pt1,pt2,pt3⟩)

arguments	example	explanation
(pt1,pt2,pt3)	**\tkzPermute**(A,B,C)	A, $\widehat{B,A,C}$ are unchanged, B, C exchange their position

The triangle is unchanged.

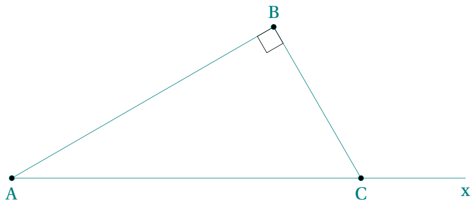
`\tkzPermute(⟨pt1,pt2,pt3⟩)`

参数	样例	说明
(pt1,pt2,pt3)	\tkzPermute(A,B,C)	B，C 交换位置，A，widehatB,A,C 保持不变

The triangle is unchanged.

15.4.1. Modification of the school triangle

This triangle is constructed from the segment [AB] on [A,x).
这个三角形是由 [AB] 在 [A,x) 上构建的。
If we want the segment [AC] to be on [A,x), we just have to swap B and C.
如果想让线段 [AC] 在 [A,x] 上，我们只需将 B 和 C 交换。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B,6/0/x}
  \tkzDefTriangle[school](A,B)
  \tkzGetPoint{C}
  \tkzPermute(A,B,C)
  \tkzDrawSegments(A,B C,x)
  \tkzDrawSegments(A,C B,C)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,C,x)
  \tkzLabelPoints[above](B)
  \tkzMarkRightAngles(C,B,A)
\end{tikzpicture}
```

Remark: Only the first point is unchanged. The order of the last two parameters is not important.
备注：只有第一点是不变的。后面两个参数的顺序并不重要。

16. Definition of polygons 定义多边形

16.1. Defining the points of a square 定义正方形

We have seen the definitions of some triangles. Let us look at the definitions of some quadrilaterals and regular polygons.
已经看到了一些三角形的定义。现在看看一些四边形和正多边形的定义。

`\tkzDefSquare(⟨pt1,pt2⟩)`

The square is defined in the forward direction. From two points, two more points are obtained such that the four taken in order form a square. The square is defined in the forward direction.
The results are in `tkzFirstPointResult` and `tkzSecondPointResult`.
We can rename them with `\tkzGetPoints`.

Arguments	example	explanation
(⟨pt1,pt2⟩)	\tkzDefSquare(⟨A,B⟩)	The square is defined in the direct direction.

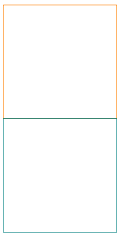
`\tkzDefSquare(⟨pt1,pt2⟩)`

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
(⟨pt1,pt2⟩)	\tkzDefSquare(⟨A,B⟩)	按指定的方向定义正方形

16.1.1. Using \tkzDefSquare with two points 通过两个点定义正方形

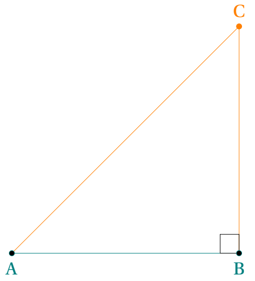
Note the inversion of the first two points and the result.
需要注意点的方向问题。



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

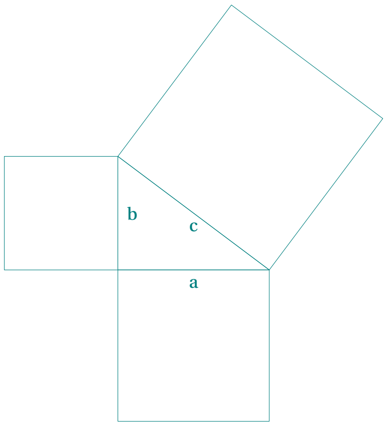
We may only need one point to draw an isosceles right-angled triangle so we use `\tkzGetFirstPoint` or `\tkzGetSecondPoint`.
可以使用`\tkzGetFirstPoint` 或`\tkzGetSecondPoint` 命令利用其中的 1 个点绘制等腰直角三角形。

16.1.2. Use of \tkzDefSquare to obtain an isosceles right-angled triangle 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}
  \tkzDrawSegment(A,B)
  \tkzDrawSegments[new](A,C B,C)
  \tkzMarkRightAngles(A,B,C)
  \tkzDrawPoints(A,B) \tkzDrawPoint[new](C)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[new,above](C)
\end{tikzpicture}
```

16.1.3. Pythagorean Theorem and \tkzDefSquare 绘制 Pythagorean 定理示意图



```
\begin{tikzpicture}[scale=.5]
\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}
\tkzDrawPolygon(A,B,C)
\tkzDrawPolygon(A,C,G,H)
\tkzDrawPolygon(C,B,I,J)
\tkzDrawPolygon(B,A,E,F)
\tkzLabelSegment(A,C){$a$}
\tkzLabelSegment[right](C,B){$b$}
\tkzLabelSegment[swap](A,B){$c$}
\end{tikzpicture}
```

16.2. Defining the points of a rectangle 定义矩形的点

.

`\tkzDefRectangle(⟨pt1,pt2⟩)`

The rectangle is defined in the forward direction. From two points, two more points are obtained such that the four taken in order form a rectangle. The two points passed in arguments are the ends of a diagonal of the rectangle. The sides are parallel to the axes.
The results are in `tkzFirstPointResult` and `tkzSecondPointResult`.
We can rename them with `\tkzGetPoints`.

Arguments	example	explanation
(⟨pt1,pt2⟩)	\tkzDefRectangle(⟨A,B⟩)	The rectangle is defined in the direct direction.

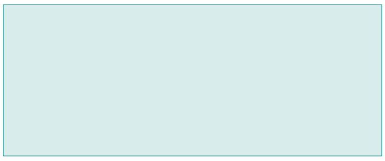
`\tkzDefRectangle(⟨pt1,pt2⟩)`

矩形被定义在向前的方向上。从两个点获得另外两个点，使得四个点按顺序形成一个矩形。参数中的两个点是矩形对角线的端点。边平行于轴线。【即输入矩形对角线上的两个端点，再获得另外两个端点。所得矩形的边是水平与竖直的】

结果在 tkzFirstPointResult 和 tkzSecondPointResult 中。
可以用\tkzGetPoints 来重命名它们。

Arguments	example	explanation
(⟨pt1,pt2⟩)	\tkzDefRectangle(⟨A,B⟩)	矩形是在直接方向上定义的。

16.2.1. Example of a rectangle definition



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,5/2/C}
\tkzDefRectangle(A,C) \tkzGetPoints{B}{D}
\tkzDrawPolygon[fill=teal!15](A,...,D)
\end{tikzpicture}
```

16.3. Definition of parallelogram 定义平行四边形

Defining the points of a parallelogram. It is a matter of completing three points in order to obtain a parallelogram.
定义平行四边形的点。从已知三个点得到一个平行四边形。

`\tkzDefParallelogram(⟨pt1,pt2,pt3⟩)`

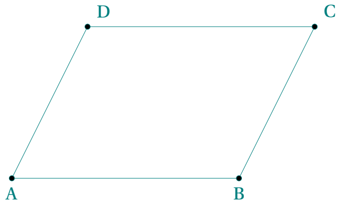
arguments	default	definition
(⟨pt1,pt2,pt3⟩)	no default	Three points are necessary

`\tkzDefParallelogram(⟨pt1,pt2,pt3⟩)`

参数	默认值	含义
(⟨pt1,pt2,pt3⟩)	无	必须 3 个顶点

From three points, another point is obtained such that the four taken in order form a parallelogram.
The result is in tkzPointResult.
We can rename it with the name \tkzGetPoint...
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\tkzPointResult 中。可使用\tkzGetPoint 命令保存并命名结果...。

16.3.1. Example of a parallelogram definition



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

16.4. The golden rectangle 黄金矩形

`\tkzDefGoldenRectangle(⟨point,point⟩)`

The macro determines a rectangle whose size ratio is the number Φ.
The created points are in **tkzFirstPointResult** and **tkzSecondPointResult**.
They can be obtained with the macro **\tkzGetPoints**. The following macro is used to draw the rectangle.

arguments	example	explanation
(⟨pt1,pt2⟩)	(⟨A,B⟩)	If C and D are created then $AB/BC = \Phi$.

`\tkzDefGoldenRectangle` or `\tkzDefGoldRectangle`

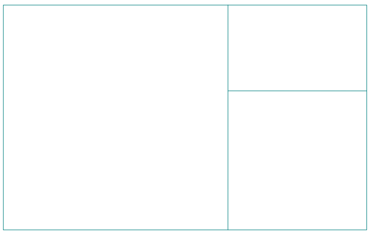
`\tkzDefGoldenRectangle(⟨point,point⟩)`

该宏确定一个矩形，其大小比是数字 Φ。
创建的点保存于 **tkzFirstPointResult** 和 **tkzSecondPointResult** 中。
它们可以通过宏**\tkzGetPoints** 获得。下面的宏用于绘制矩形。

参数	样例	说明
(⟨pt1,pt2⟩)	(⟨A,B⟩)	如果创建了 C 和 D，则 $AB/BC = \Phi$。

`\tkzDefGoldenRectangle` or `\tkzDefGoldRectangle`

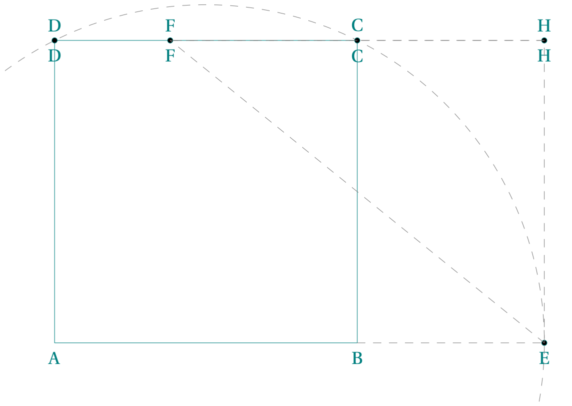
16.4.1. Golden Rectangles



```
\begin{tikzpicture}[scale=.6]
 \tkzDefPoint(0,0){A}      \tkzDefPoint(8,0){B}
 \tkzDefGoldRectangle(A,B) \tkzGetPoints{C}{D}
 \tkzDefGoldRectangle(B,C) \tkzGetPoints{E}{F}
 \tkzDefGoldRectangle(C,E) \tkzGetPoints{G}{H}
 \tkzDrawPolygon(A,B,C,D)
 \tkzDrawSegments(E,F G,H)
\end{tikzpicture}
```

16.4.2. Construction of the golden rectangle 黄金矩形示例

Without the previous macro here is how to get the golden rectangle.
不用前面的宏，得到黄金矩形的方法。



```
\begin{tikzpicture}[scale=.5]
\tkzDefPoint(0,0){A}
\tkzDefPoint(8,0){B}
\tkzDefMidPoint(A,B)
\tkzGetPoint{I}
\tkzDefSquare(A,B)\tkzGetPoints{C}{D}
\tkzInterLC(A,B)(I,C)\tkzGetPoints{G}{E}
\tkzDefPointWith[colinear= at C](E,B)
 \tkzGetPoint{F}
\tkzDefPointBy[projection=onto D--C ](E)
 \tkzGetPoint{H}
\tkzDrawArc[style=dashed](I,E)(D)
\tkzDrawPolygon(A,B,C,D)
\tkzDrawPoints(C,D,E,F,H)
\tkzLabelPoints(A,B,C,D,E,F,H)
\tkzLabelPoints[above](C,D,F,H)
\tkzDrawSegments[style=dashed,color=gray]%
(E,F C,F B,E F,H H,C E,H)
\end{tikzpicture}
```

16.5. Regular polygon 正多边形

\tkzDefRegPolygon[⟨local options⟩](⟨pt1,pt2⟩)

From the number of sides, depending on the options, this macro determines a regular polygon according to its center or one side.

arguments	example	explanation
(⟨pt1,pt2⟩)	(⟨O,A⟩)	with option "center", O is the center of the polygon.
(⟨pt1,pt2⟩)	(⟨A,B⟩)	with option "side", [AB] is a side.

options	default	example
name	P	The vertices are named P1,P2,…
sides	5	number of sides.
center	center	The first point is the center.
side	center	The two points are vertices.
Options TikZ	...	

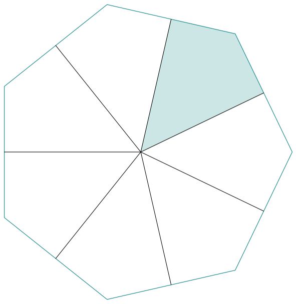
\tkzDefRegPolygon[⟨ 命令选项 ⟩](⟨pt1,pt2⟩)

根据选项中指定的边数，以指定的点为中心或是指定的边，定义一个正多边形。

参数	样例	说明
(⟨pt1,pt2⟩)	(⟨O,A⟩)	如果使用“center”选项，则 O 是多边形中心
(⟨pt1,pt2⟩)	(⟨A,B⟩)	如果使用“side”选项，[AB] 一条边

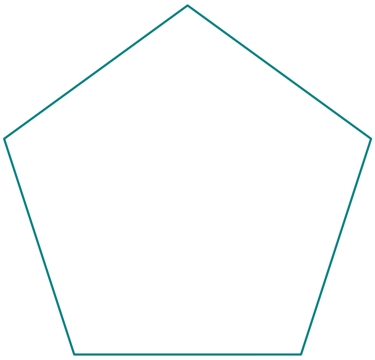
选项	默认值	样例
name	P	顶点命名为 P1，P2，…
sides	5	边数
center	center	第 1 个点是正多边形中心
side	center	指定的两个顶点构成一条边
TikZ 选项	…	

16.5.1. Option center



```
\begin{tikzpicture}
  \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)
  \tkzFillPolygon[teal!20](Q0,Q1,P2,Q2)
  \tkzDrawPolygon(P1,P...,P7)
  \foreach \j in {1,...,7} {%
  \tkzDrawSegment[black](P0,Q\j)}
\end{tikzpicture}
```

16.5.2. Option side



```
\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. Circles 圆

Among the following macros, one will allow you to draw a circle, which is not a real feat. To do this, you will need to know the center of the circle and either the radius of the circle or a point on the circumference. It seemed to me that the most frequent use was to draw a circle with a given center passing through a given point. This will be the default method, otherwise you will have to use the R option. There are a large number of special circles, for example the circle circumscribed by a triangle.
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。

- I have created a first macro \tkzDefCircle which allows, according to a particular circle, to retrieve its center and the measurement of the radius in cm. This recovery is done with the macros \tkzGetPoint and \tkzGetLength;
- then a macro \tkzDrawCircle;
- then a macro that allows you to color in a disc, but without drawing the circle \tkzFillCircle;
- sometimes, it is necessary for a drawing to be contained in a disk, this is the role assigned to \tkzClipCircle;
- it finally remains to be able to give a label to designate a circle and if several possibilities are offered, we will see here \tkzLabelCircle.
- \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. Characteristics of a circle: \tkzDefCircle 定义圆

This macro allows you to retrieve the characteristics (center and radius) of certain circles.
该命令用指定的圆心和半径定义一个圆。

```
\tkzDefCircle[⟨local options⟩](⟨A,B⟩) or (⟨A,B,C⟩)
```

Attention the arguments are lists of two or three points. This macro is either used in partnership with \tkzGetPoints to obtain the center and a point on the circle, or by using tkzFirstPointResult and tkzSecondPointResult if it is not necessary to keep the results. You can also use \tkzGetLength to get the radius.

arguments	example	explanation
(⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)	(⟨A,B⟩)	[AB] is radius A is the center

options	default	definition
R	circum	circle characterized by a center and a radius
diameter	circum	circle characterized by two points defining a diameter
circum	circum	circle circumscribed of a triangle
in	circum	incircle a triangle
ex	circum	excircle of a triangle
euler or nine	circum	Euler's Circle
spieker	circum	Spieker Circle
apollonius	circum	circle of Apollonius
orthogonal from	circum	[orthogonal from = A](O,M)
orthogonal through	circum	[orthogonal through = A and B](O,M)
K	1	coefficient used for a circle of Apollonius

> *In the following examples, I draw the circles with a macro not yet presented. You may only need the center and a point on the circle.*

```
\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`\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。

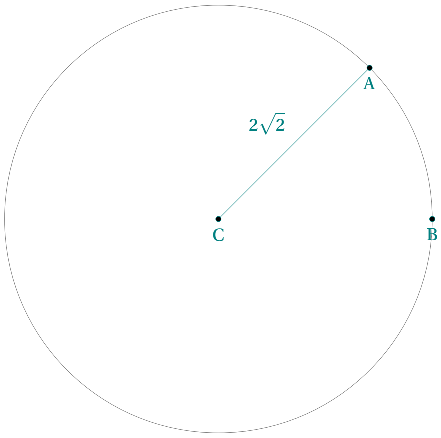
参数	样例	说明
(⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)	(⟨A,B⟩)	[AB] 是半径 A 圆心

选项	默认值	含义
R	circum	圆心和半径
diameter	circum	两点连线为直径
circum	circum	三角形的外接圆
in	circum	三角形的内切圆
ex	circum	三角形的旁切圆
euler or nine	circum	三角形的欧拉圆
spieker	circum	三角形的 Spieker 圆
apollonius	circum	Apollonius 圆
orthogonal from	circum	[orthogonal from = A](O,M)
orthogonal through	circum	[orthogonal through = A and B](O,M)
K	1	Apollonius 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. Example with option R

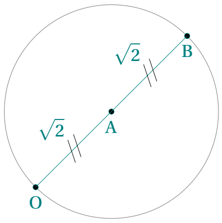
We obtain with the macro `\tkzGetPoint` a point of the circle which is the East pole.
使用宏`\tkzGetPoint` 获得圆的一个点，该点是圆与过圆心且水平的线与圆的交点（右边的点）。



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(3,3){C}
  \tkzDefPoint(5,5){A}
   \tkzCalcLength(A,C) \tkzGetLength{rAC}
  \tkzDefCircle[R](C,\rAC) \tkzGetPoint{B}
  \tkzDrawCircle(C,B)
  \tkzDrawSegment(C,A)
  \tkzLabelSegment[above left](C,A){$2\sqrt{2}$}
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,C,B)
\end{tikzpicture}
```

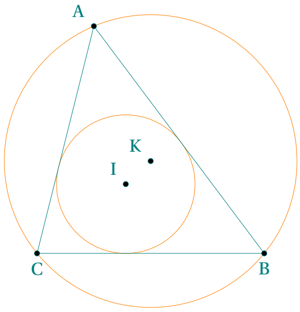
17.1.2. Example with option diameter 直径

It is simpler here to search directly for the middle of [AB]. The result is the center and if necessary
可以通过线段 [AB] 的中点确定圆心。



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,2){B}
  \tkzDefCircle[diameter](O,B) \tkzGetPoint{A}
  \tkzDrawCircle(A,B)
  \tkzDrawPoints(O,A,B)
  \tkzDrawSegment(O,B)
  \tkzLabelPoints(O,A,B)
  \tkzLabelSegment[above left](O,A){$\sqrt{2}$}
  \tkzLabelSegment[above left](A,B){$\sqrt{2}$}
  \tkzMarkSegments[mark=s||](O,A A,B)
\end{tikzpicture}
```

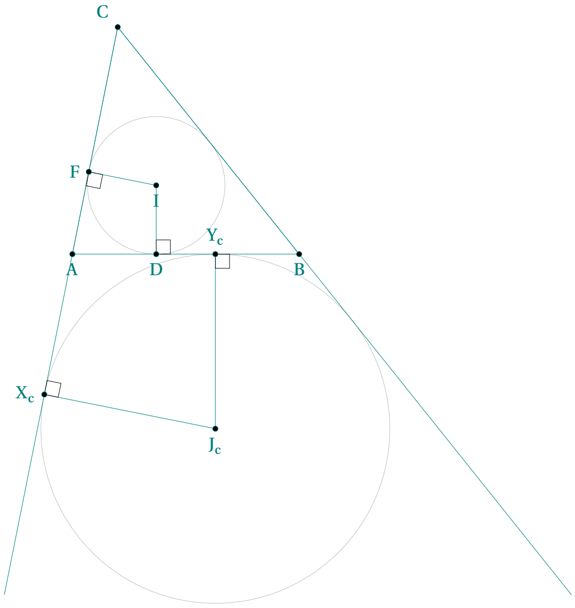
17.1.3. Circles inscribed and circumscribed for a given triangle 三角形的内切圆和外接圆示例



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(2,2){A}  \tkzDefPoint(5,-2){B}
 \tkzDefPoint(1,-2){C}
 \tkzDefCircle[in](A,B,C)
 \tkzGetPoints{I}{x}
 \tkzDefCircle[circum](A,B,C)
 \tkzGetPoint{K}
 \tkzDrawCircles[new](I,x K,A)
 \tkzLabelPoints[below](B,C)
 \tkzLabelPoints[above left](A,I,K)
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPoints(A,B,C,I,K)
\end{tikzpicture}
```

17.1.4. Example with option ex

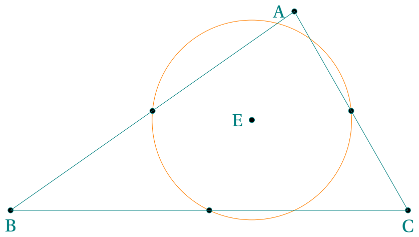
We want to define an excircle of a triangle relatively to point C
与顶点 C 对应的旁切圆。



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{ 0/0/A,4/0/B,0.8/4/C}
  \tkzDefCircle[ex](B,C,A)
  \tkzGetPoints{J_c}{h}
  \tkzDefPointBy[projection=onto A--C ](J_c)
  \tkzGetPoint{X_c}
  \tkzDefPointBy[projection=onto A--B ](J_c)
  \tkzGetPoint{Y_c}
  \tkzDefCircle[in](A,B,C)
  \tkzGetPoints{I}{y}
  \tkzDrawCircles[color=lightgray](J_c,h I,y)
  \tkzDefPointBy[projection=onto A--C ](I)
  \tkzGetPoint{F}
  \tkzDefPointBy[projection=onto A--B ](I)
  \tkzGetPoint{D}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawLines[add=0 and 1.5](C,A C,B)
  \tkzDrawSegments(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)
  \tkzLabelPoints[above](Y_c)
  \tkzLabelPoints[left](X_c)
  \tkzLabelPoints[above left](C)
  \tkzLabelPoints[left](F)
\end{tikzpicture}
```

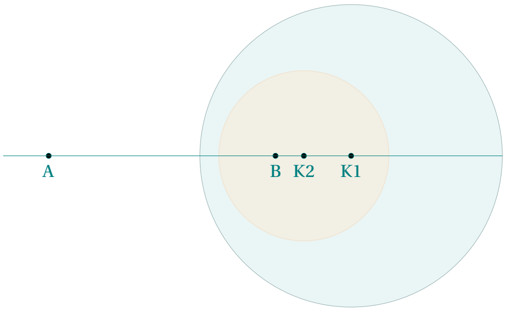
17.1.5. Euler's circle for a given triangle with option euler

We verify that this circle passes through the middle of each side.
验证欧拉圆会通过三角形三个边的中点。



```
\begin{tikzpicture}[scale=.75]
   \tkzDefPoint(5,3.5){A}
   \tkzDefPoint(0,0){B} \tkzDefPoint(7,0){C}
   \tkzDefCircle[euler](A,B,C)
   \tkzGetPoints{E}{e}
   \tkzDefSpcTriangle[medial](A,B,C){M_a,M_b,M_c}
   \tkzDrawCircle[new](E,e)
   \tkzDrawPoints(A,B,C,E,M_a,M_b,M_c)
   \tkzDrawPolygon(A,B,C)
   \tkzLabelPoints[below](B,C)
   \tkzLabelPoints[left](A,E)
\end{tikzpicture}
```

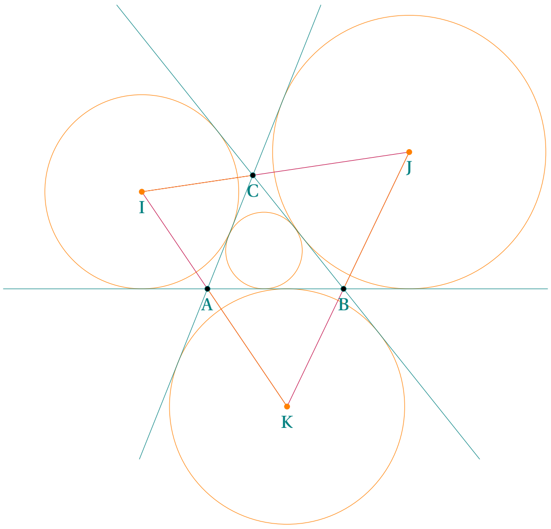
17.1.6. Apollonius circles for a given segment option apollonius



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefCircle[apollonius,K=2](A,B)
  \tkzGetPoints{K1}{x}
  \tkzDrawCircle[color = teal!50!black,
      fill=teal!20,opacity=.4](K1,x)
  \tkzDefCircle[apollonius,K=3](A,B)
  \tkzGetPoints{K2}{y}
  \tkzDrawCircle[color=orange!50,
      fill=orange!20,opacity=.4](K2,y)
  \tkzLabelPoints[below](A,B,K1,K2)
  \tkzDrawPoints(A,B,K1,K2)
  \tkzDrawLine[add=.2 and 1](A,B)
\end{tikzpicture}
```

17.1.7. Circles exinscribed to a given triangle option ex

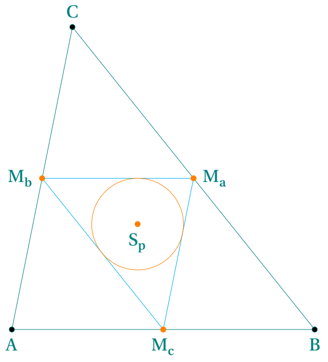
You can also get the center and the projection of it on one side of the triangle.
with `\tkzGetFirstPoint{Jb}` and `\tkzGetSecondPoint{Tb}`.
可以使用`\tkzGetFirstPoint{Jb}`和`\tkzGetSecondPoint{Tb}`命令，得到旁切圆圆心在边上的投影。



```
\begin{tikzpicture}[scale=.6]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(3,0){B}
  \tkzDefPoint(1,2.5){C}
  \tkzDefCircle[ex](A,B,C) \tkzGetPoints{I}{i}
  \tkzDefCircle[ex](C,A,B) \tkzGetPoints{J}{j}
  \tkzDefCircle[ex](B,C,A) \tkzGetPoints{K}{k}
  \tkzDefCircle[in](B,C,A) \tkzGetPoints{O}{o}
  \tkzDrawCircles[new](J,j I,i K,k O,o)
  \tkzDrawLines[add=1.5 and 1.5](A,B A,C B,C)
  \tkzDrawPolygon[purple](I,J,K)
  \tkzDrawSegments[new](A,K B,J C,I)
  \tkzDrawPoints(A,B,C)
  \tkzDrawPoints[new](I,J,K)
  \tkzLabelPoints(A,B,C,I,J,K)
\end{tikzpicture}
```

17.1.8. Spieker circle with option spieker

The incircle of the medial triangle $M_aM_bM_c$ is the Spieker circle:
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(A,B,C)
   \tkzDrawPolygon[cyan](M_a,M_b,M_c)
   \tkzDrawPoints(B,C,A)
   \tkzDefCircle[spieker](A,B,C)
   \tkzDrawPoints[new](M_a,M_b,M_c,S_p)
   \tkzDrawCircle[new](tkzFirstPointResult,tkzSecondPointResult)
   \tkzLabelPoints[right](M_a)
   \tkzLabelPoints[left](M_b)
   \tkzLabelPoints[below](A,B,M_c,S_p)
   \tkzLabelPoints[above](C)
\end{tikzpicture}
```

17.2. Projection of excenters 旁心投影

\tkzDefProjExcenter[⟨local options⟩](⟨A,B,C⟩)(⟨a,b,c⟩){⟨X,Y,Z⟩}

Each excenter has three projections on the sides of the triangle ABC. We can do this with one macro **\tkzDefProjExcenter[name=J](A,B,C)(a,b,c){Y,Z,X}**.

options	default	definition
name	no defaut	used to name the vertices

arguments	default	definition
(pt1=α_1,pt2=α_2,...)	no default	Each point has a assigned weight

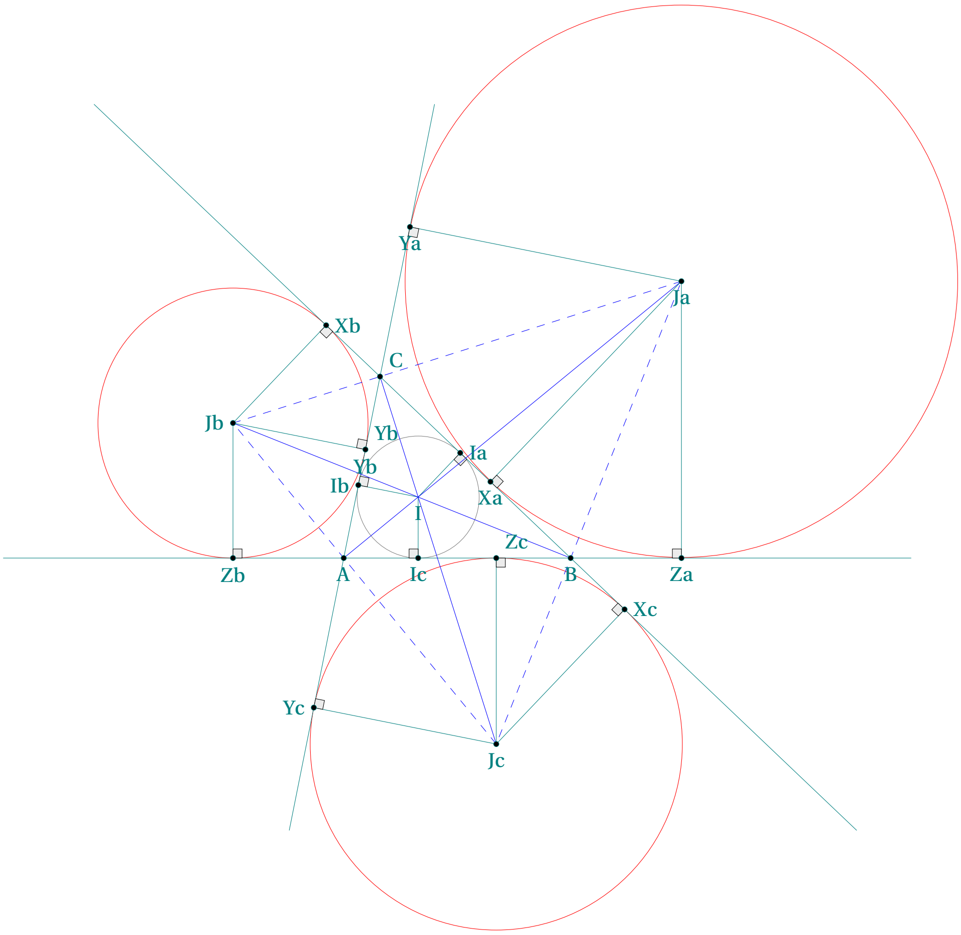
\tkzDefProjExcenter[⟨local options⟩](⟨A,B,C⟩)(⟨a,b,c⟩){⟨X,Y,Z⟩}

每个旁心在三角形 ABC 的边上有三个投影。我们可以用一个宏 **\tkzDefProjExcenter[name=J](A,B,C)(a,b,c){Y,Z,X}**.

选项	默认值	含义
name	无	用于命名顶点

arguments	default	definition
(pt1=α_1,pt2=α_2,...)	no default	Each point has a assigned weight

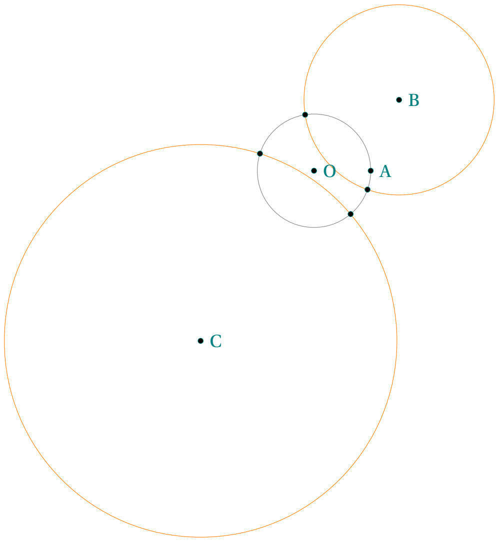
17.2.1. Excircles



```
\begin{tikzpicture}[scale=.6]
\tikzset{line style/.append style={line width=.2pt}}
\tikzset{label style/.append style={color=teal,font=\footnotesize}}
\tkzDefPoints{0/0/A,5/0/B,0.8/4/C}
\tkzDefSpcTriangle[excentral,name=J](A,B,C){a,b,c}
\tkzDefSpcTriangle[intouch,name=I](A,B,C){a,b,c}
\tkzDefProjExcenter[name=J](A,B,C)(a,b,c){X,Y,Z}
\tkzDefCircle[in](A,B,C)   \tkzGetPoint{I} \tkzGetSecondPoint{T}
\tkzDrawCircles[red](Ja,Xa Jb,Yb Jc,Zc)
\tkzDrawCircle(I,T)
\tkzDrawPolygon[dashed,color=blue](Ja,Jb,Jc)
\tkzDrawLines[add=1.5 and 1.5](A,C A,B B,C)
\tkzDrawSegments(Ja,Xa Ja,Ya Ja,Za
                 Jb,Xb Jb,Yb Jb,Zb
                 Jc,Xc Jc,Yc Jc,Zc
                 I,Ia I,Ib I,Ic)
\tkzMarkRightAngles[size=.2,fill=gray!15](Ja,Za,B Ja,Xa,B Ja,Ya,C Jb,Yb,C)
\tkzMarkRightAngles[size=.2,fill=gray!15](Jb,Zb,B Jb,Xb,C Jc,Yc,A Jc,Zc,B Jc,Xc,C I,Ia,B I,Ib,C I,Ic,A)
\tkzDrawSegments[blue](Jc,C Ja,A Jb,B)
\tkzDrawPoints(A,B,C,Xa,Xb,Xc,Ja,Jb,Jc,Ia,Ib,Ic,Ya,Yb,Yc,Za,Zb,Zc)
\tkzLabelPoints(A,Ya,Yb,Ja,I)
\tkzLabelPoints[left](Jb,Ib,Yc)
\tkzLabelPoints[below](Zb,Ic,Jc,B,Za,Xa)
\tkzLabelPoints[above right](C,Zc,Yb)
\tkzLabelPoints[right](Xb,Ia,Xc)
\end{tikzpicture}
```

17.2.2. Orthogonal from

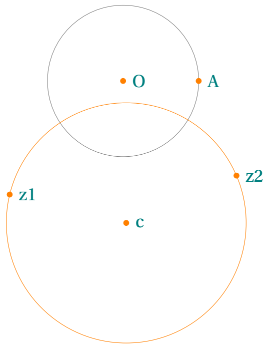
Orthogonal circle of given center. \tkzGetPoints{z1}{z2} gives two points of the circle.
给定中心的正交圆。\tkzGetPoints{z1}{z2} 给出了圆的两个点。



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

17.2.3. Orthogonal through

Orthogonal circle passing through two given points.
通过两个给定点的正交圆。



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){A}
  \tkzDrawCircle(O,A)
  \tkzDefPoint(-1.5,-1.5){z1}
  \tkzDefPoint(1.5,-1.25){z2}
  \tkzDefCircle[orthogonal through=z1 and z2](O,A)
   \tkzGetPoint{c}
  \tkzDrawCircle[new](tkzPointResult,z1)
  \tkzDrawPoints[new](O,A,z1,z2,c)
  \tkzLabelPoints[right](O,A,z1,z2,c)
\end{tikzpicture}
```

17.3. Definition of circle by transformation; \tkzDefCircleBy 变换得到圆

These transformations are:
这些变换有：

- translation;
- homothety;
- orthogonal reflection or symmetry;
- central symmetry;
- orthogonal projection;
- rotation (degrees);
- inversion.
- 平移;
- 中心投影;
- 正交反射或对称;
- 中心对称;
- 正交投影;
- 旋转 (度);
- 翻转.

The choice of transformations is made through the options. The macro is `\tkzDefCircleBy` and the other for the transformation of a list of points `\tkzDefCirclesBy`. For example, we'll write:
变换是通过选项确定的。宏是`\tkzDefCircleBy`，另一个是用于点的列表的变换`\tkzDefCirclesBy`。例如，我们会写：

```
\tkzDefCircleBy[translation= from A to A'](O,M)
```

O is the center and M is a point on the circle. The image is a circle. The new center is `tkzFirstPointResult` and `tkzSecondPointResult` is a point on the new circle. You can get the results with the macro `\tkzGetPoints`.
O 是圆心，M 是圆上的一点。图像是一个圆。新的圆心是`tkzFirstPointResult`, 而`tkzSecondPointResult` 是新圆上的一点。您可以使用宏`\tkzGetPoints` 获得结果。

`\tkzDefCircleBy[⟨local options⟩](⟨pt1,pt2⟩)`

The argument is a couple of points. The results is a couple of points. If you want to keep these points then the macro `\tkzGetPoints{O'}{M'}` allows you to assign the name `O'` to the center and `M'` to the point on the circle.

arguments	definition	examples
pt1,pt2	existing points	(O,M)

options		examples
translation	= from #1 to #2	[translation=from A to B](O,M)
homothety	= center #1 ratio #2	[homothety=center A ratio .5](O,M)
reflection	= over #1--#2	[reflection=over A--B](O,M)
symmetry	= center #1	[symmetry=center A](O,M)
projection	= onto #1--#2	[projection=onto A--B](O,M)
rotation	= center #1 angle #2	[rotation=center O angle 30](O,M)
inversion	= center #1 through #2	[inversion =center O through A](O,M)

The image is only defined and not drawn.

`\tkzDefCircleBy[⟨local options⟩](⟨pt1,pt2⟩)`

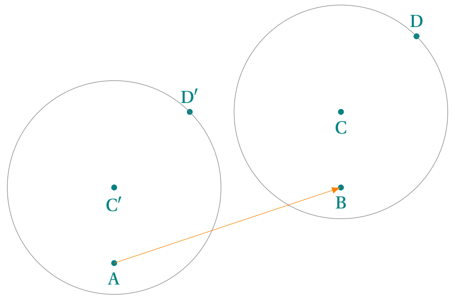
The argument is a couple of points. The results is a couple of points. If you want to keep these points then the macro `\tkzGetPoints{O'}{M'}` allows you to assign the name `O'` to the center and `M'` to the point on the circle. 参数是两个点。结果是两点。如果您想保留这些点，那么宏`\tkzGetPoints{O'}{M'}`允许将名称 `O'` 指定给圆心，将 `M'` 指定给圆上的点。

参数	样例	说明
pt1,pt2	存在的点	(O, M)

选项		含义
translation	= from #1 to #2	[translation=from A to B](O,M)
homothety	= center #1 ratio #2	[homothety=center A ratio .5](O,M)
reflection	= over #1--#2	[reflection=over A--B](O,M)
symmetry	= center #1	[symmetry=center A](O,M)
projection	= onto #1--#2	[projection=onto A--B](O,M)
rotation	= center #1 angle #2	[rotation=center O angle 30](O,M)
inversion	= center #1 through #2	[inversion =center O through A](O,M)

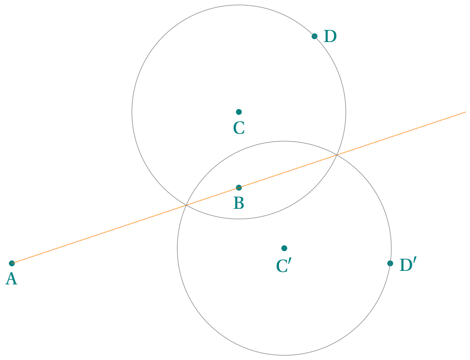
只定义图像，不绘制图像。

17.3.1. Translation 平移



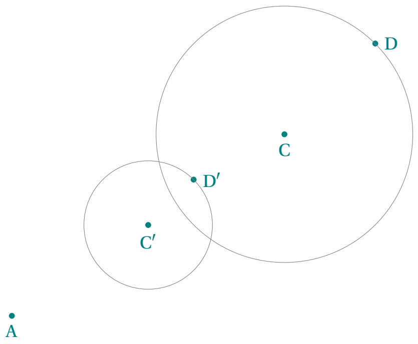
```
\begin{tikzpicture}[>=latex]
 \tkzDefPoint(0,0){A}  \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[translation= from B to A](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](A,B,C,D,C',D')
 \tkzDrawSegments[orange,->](A,B)
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,B,C,C')
 \tkzLabelPoints[color=teal,above](D,D')
\end{tikzpicture}
```

17.3.2. Reflection (orthogonal symmetry) 对称



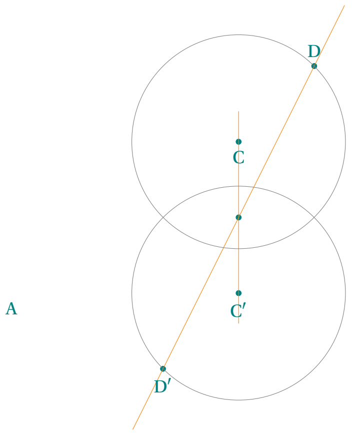
```
\begin{tikzpicture}[>=latex]
 \tkzDefPoint(0,0){A}  \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[reflection = over A--B](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](A,B,C,D,C',D')
 \tkzDrawLine[add =0 and 1][orange](A,B)
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,B,C,C')
 \tkzLabelPoints[color=teal,right](D,D')
\end{tikzpicture}
```

17.3.3. Homothety Homothety



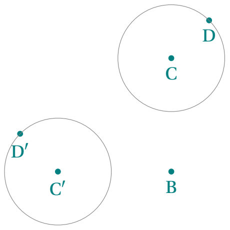
```
\begin{tikzpicture}[scale=1.2]
 \tkzDefPoint(0,0){A}   \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[homothety=center A ratio .5](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](A,C,D,C',D')
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,C,C')
 \tkzLabelPoints[color=teal,right](D,D')
\end{tikzpicture}
```

17.3.4. Symmetry 对称



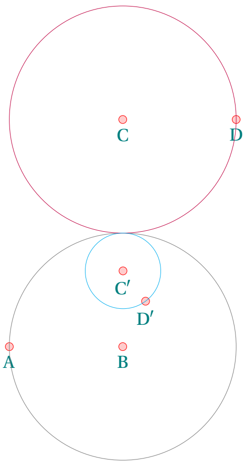
```
\begin{tikzpicture}[scale=1]
 \tkzDefPoint(0,0){A}   \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[symmetry=center B](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](B,C,D,C',D')
 \tkzDrawLines[orange](C,C' D,D')
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,C,C')
 \tkzLabelPoints[color=teal,above](D)
 \tkzLabelPoints[color=teal,below](D')
\end{tikzpicture}
```

17.3.5. Rotation 旋转



```
\begin{tikzpicture}[scale=0.5]
 \tkzDefPoint(3,-1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[rotation=center B angle 90](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](B,C,D,C',D')
 \tkzLabelPoints[color=teal](B,C,D,C',D')
 \tkzDrawCircles(C,D C',D')
\end{tikzpicture}
```

17.3.6. Inversion 翻折



```
\begin{tikzpicture}[scale=1.5]
\tkzSetUpPoint[size=3,color=red,fill=red!20]
\tkzSetUpStyle[color=purple,ultra thin]{st1}
\tkzSetUpStyle[color=cyan,ultra thin]{st2}
\tkzDefPoint(2,0){A} \tkzDefPoint(3,0){B}
\tkzDefPoint(3,2){C} \tkzDefPoint(4,2){D}
\tkzDefCircleBy[inversion = center B through A](C,D)
\tkzGetPoints{C'}{D'}
\tkzDrawPoints(A,B,C,D,C',D')
\tkzLabelPoints(A,B,C,D,C',D')
\tkzDrawCircles(B,A)
\tkzDrawCircles[st1](C,D)
\tkzDrawCircles[st2](C',D')
\end{tikzpicture}
```

18. Intersections 交点

It is possible to determine the coordinates of the points of intersection between two straight lines, a straight line and a circle, and two circles.
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。
The associated commands have no optional arguments and the user must determine the existence of the intersection points himself.
求交点的相关命令没有可选参数，用户必须确保交点存在。

18.1. Intersection of two straight lines \tkzInterLL 求两条直线的交点

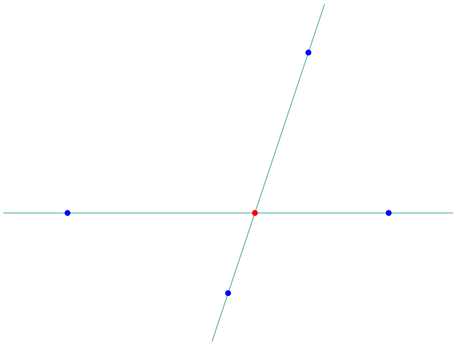
\tkzInterLL(⟨A,B⟩)(⟨C,D⟩)

Defines the intersection point **tkzPointResult** of the two lines (AB) and (CD). The known points are given in pairs (two per line) in brackets, and the resulting point can be retrieved with the macro **\tkzDefPoint**.

\tkzInterLL(⟨A,B⟩)(⟨C,D⟩)

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** 保存并命令交点。

18.1.1. Example of intersection between two straight lines 两直线交点示例



```
\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}
```

18.2. Intersection of a straight line and a circle \tkzInterLC 一条直线和一个圆的交点

As before, the line is defined by a couple of points. The circle is also defined by a couple:
直线可以由两个点定义，圆可以按如下方式进行定义：

- (O,C) which is a pair of points, the first is the center and the second is any point on the circle.
- (O,r) The r measure is the radius measure.
- (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* 可 *pt*。

\tkzInterLC[⟨options⟩](⟨A,B⟩)(⟨O,C⟩) or (⟨O,r⟩) or (⟨O,C,D⟩)

So the arguments are two couples.

options	default	definition
N	N	(O,C) determines the circle
R	N	(O, 1) unit 1 cm
with nodes	N	(O,C,D) CD is a radius
common=pt		pt is common point; tkzFirstPoint gives the other point
near		tkzFirstPoint is the closest point to the first point of the line

The macro defines the intersection points I and J of the line (AB) and the center circle O with radius r if they exist; otherwise, an error will be reported in the `.log` file. `with nodes` avoids you to calculate the radius which is the length of [CD]. If `common` and `near` are not used then `tkzFirstPoint` is the smallest angle (angle with `tkzSecondPoint` and the center of the circle).

\tkzInterLC[⟨命令选项⟩](⟨A,B⟩)(⟨O,C⟩) 或 (⟨O,r⟩) 或 (⟨O,C,D⟩)

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) 单位 cm
with nodes	N	(O,C,D)CD 是半径
common=pt		pt 是共同点；tkzFirstPoint 给出了另一个点
near		tkzFirstPoint 是距离直线的第一个点最近的点

定义直线 (AB) 与以点 O 圆心和 r 半径定义的圆的交点为 I 和 J，如果出现错误，则记录在`.log` 日志文件中。`with nodes` 避免您计算半径，即 [CD]。如果未使用 `common` 和 `near`，则 `tkzFirstPoint` 是最小的角度 (与 `tkzSecondPoint` 和圆心的角度)。

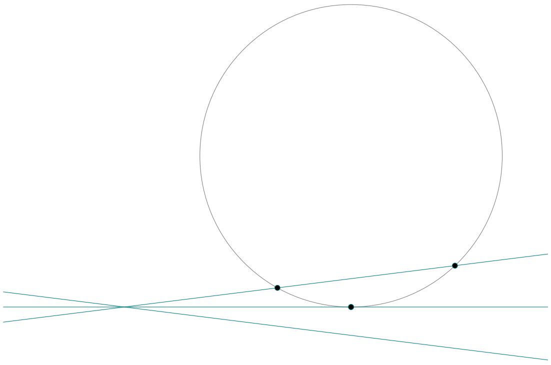
\tkzTestInterLC(⟨O,A⟩)(⟨O′,B⟩)

So the arguments are two couples which define a line and a circle with a center and a point on the circle. If there is a non empty intersection between these the line and the circle then the test `\iftkzFlagLC` gives true.

\tkzTestInterLC(⟨O,A⟩)(⟨O′,B⟩)

参数是两个：一条线，和一个圆（已知圆心和圆上的一个点）。如果在这些直线和圆之间有一个非空的交点，则命令 `\iftkzFlagLC` 给出结果为 true。

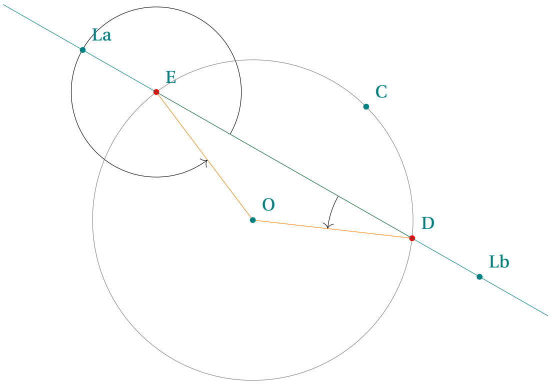
18.2.1. test line-circle intersection



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{% x   y   name
                 3   /4   /I,
                 3   /2   /P,
                 0   /2   /La,
                 8   /3   /Lb}
\tkzDrawCircle(I,P)
\foreach \i in {1,...,3}{%
   \coordinate  (Lb) at (8,\i);
   \tkzDrawLine(La,Lb)
   \tkzTestInterLC(La,Lb)(I,P)
    \iftkzFlagLC
    \tkzInterLC(La,Lb)(I,P)
    \tkzGetPoints{a}{b}
    \tkzDrawPoints(a,b)
    \fi
   }
\end{tikzpicture}
```

18.2.2. Line-circle intersection

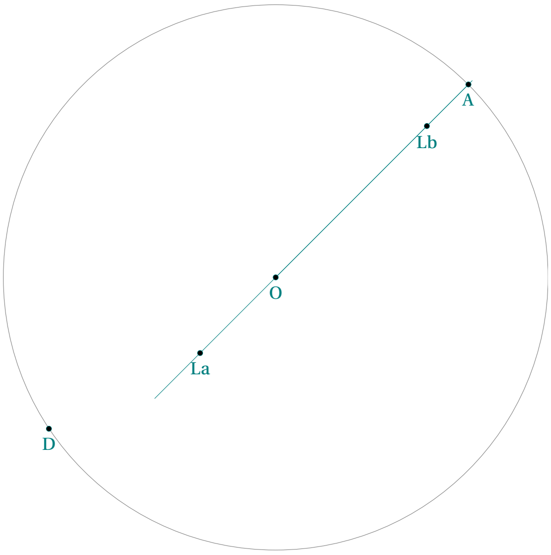
In the following example, the drawing of the circle uses two points and the intersection of the straight line and the circle uses two pairs of points. We will compare the angles $\widehat{D,E,O}$ and $\widehat{E,D,O}$. These angles are in opposite directions. **tkzFirstPoint** is assigned to the point that forms the angle with the smallest measure (counterclockwise direction). The counterclockwide angle $\widehat{D,E,O}$ has a measure equal to 360◦ minus the measure of $\widehat{O,E,D}$.
在以下示例中，圆的绘制使用两个点，直线和圆的交点使用两对点。比较角度 $\widehat{D,E,O}$ 和 $\widehat{E,D,O}$。这些角度方向相反。**tkzFirstPoint** 被分配给与最小度量值 (逆时针方向) 形成角度的点。逆时针方向角度 $\widehat{D,E,O}$ 的度量值等于 360◦ 减去 $\widehat{O,E,D}$ 的度量值。



```
\begin{tikzpicture}[scale=.75]
 \tkzInit[xmax=5,ymax=4]
 \tkzDefPoint(1,1){O}
 \tkzDefPoint(-2,4){La}
 \tkzDefPoint(5,0){Lb}
 \tkzDefPoint(3,3){C}
 \tkzInterLC(La,Lb)(O,C)  \tkzGetPoints{D}{E}
 \tkzMarkAngle[->,size=1.5](E,D,O)
 \tkzDrawPolygons[new](O,D,E)
 \tkzMarkAngle[->,size=1.5](D,E,O)
 \tkzDrawCircle(O,C)
 \tkzDrawPoints[color=teal](O,La,Lb,C)
 \tkzDrawPoints[color=red](D,E)
 \tkzDrawLine(La,Lb)
 \tkzLabelPoints[above right](O,La,Lb,C,D,E)
\end{tikzpicture}
```

18.2.3. Line passing through the center option common

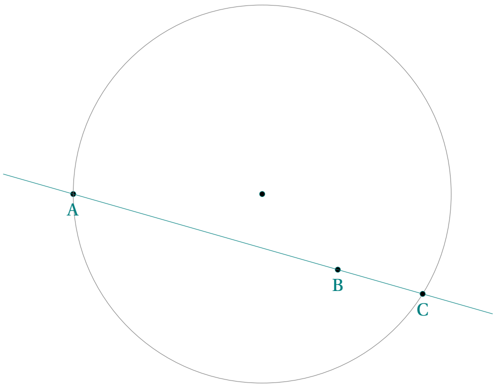
This case is special. You cannot compare the angles. In this case, the option **near** must be used. **tkzFirstPoint** is assigned to the point closest to the first point given for the line. Here we want A to be closest to Lb.
这个示例很特殊。不能比较角度的情况下，必须使用选项 **near**。**tkzFirstPoint** 被分配给最接近为直线指定的第一个点的点。这里希望 A 最接近 Lb。



```
\begin{tikzpicture}
\tkzDefPoints{% x   y   name
             0   /1   /D,
             6   /0   /B,
             3   /3   /O,
             2   /2   /La,
             5   /5   /Lb}
  \tkzDrawCircle(O,D)
  \tkzDrawLine(La,Lb)
  \tkzInterLC[near](Lb,La)(O,D)
  \tkzGetFirstPoint{A}
  \tkzDrawSegments(O,A)
  \tkzDrawPoints(O,D,A,La,Lb)
  \tkzLabelPoints(O,D,A,La,Lb)
\end{tikzpicture}
```

18.2.4. Line-circle intersection with option common

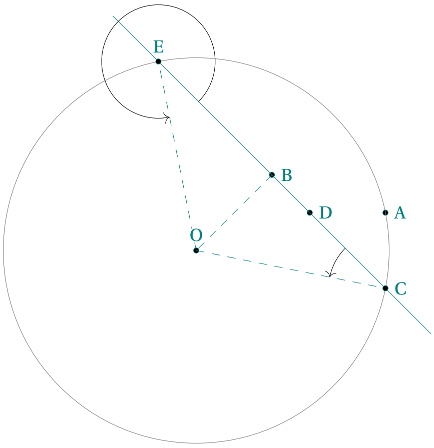
A special case that we often meet, a point of the line is on the circle and we are looking for the other common point.
经常遇到的一个特例，直线上的一点在圆上，要寻找另一个公共点。



```
\begin{tikzpicture}[scale=.5]
 \tkzDefPoints{0/0/O,-5/0/A,2/-2/B,0/5/D}
 \tkzInterLC[common=A](B,A)(O,D)
 \tkzGetFirstPoint{C}
 \tkzDrawPoints(O,A,B)
 \tkzDrawCircle(O,A)
 \tkzDrawLine(A,C)
 \tkzDrawPoint(C)
 \tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

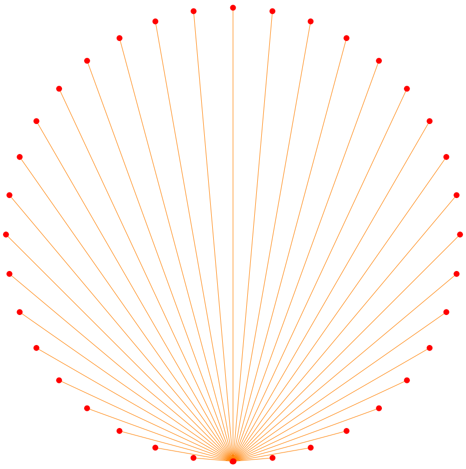
18.2.5. Line-circle intersection order of points 线圆交点顺序

The idea is to compare the angles formed with the first defining point of the line, a resultant point and the center of the circle. The first point is the one that corresponds to the smallest angle.
这个想法是比较与直线的第一个定义点、合成点和圆心形成的角度。第一个点是对应于最小角度的点。
As you can see $\widehat{BCO} < \widehat{BEO}$. To tell the truth,$\widehat{BEO}$ is counterclockwise.
结论 $\widehat{BCO} < \widehat{BEO}$。$\widehat{BEO}$ 是逆时针方向的。



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/O,5/1/A,2/2/B,3/1/D}
  \tkzInterLC[common=A](B,D)(O,A) \tkzGetPoints{C}{E}
  \tkzDrawPoints(O,A,B,D)
  \tkzDrawCircle(O,A) \tkzDrawLine(E,C)
  \tkzDrawSegments[dashed](B,O O,C)
  \tkzMarkAngle[->,size=1.5](B,C,O)
  \tkzDrawSegments[dashed](O,E)
  \tkzMarkAngle[->,size=1.5](B,E,O)
  \tkzDrawPoints(C,E)
  \tkzLabelPoints[above](O,E)
  \tkzLabelPoints[right](A,B,C,D)
\end{tikzpicture}
```

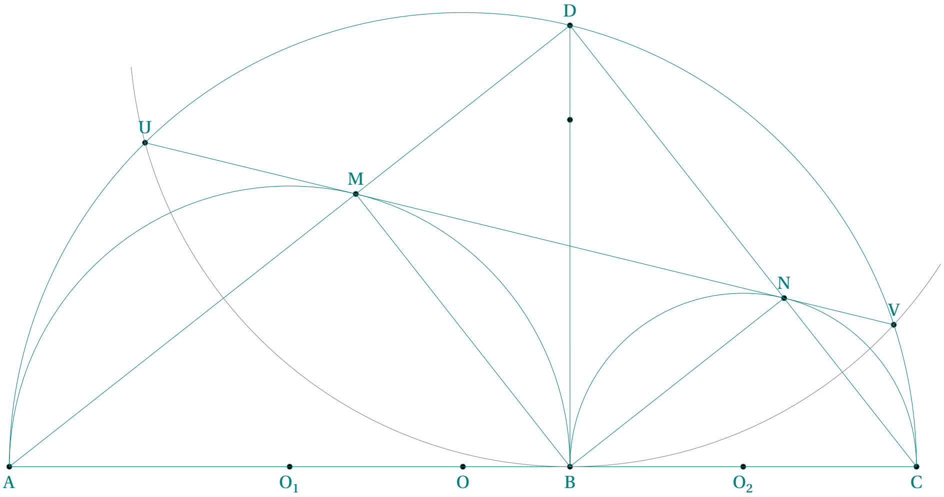
18.2.6. Example with \foreach



```
\begin{tikzpicture}[scale=3,rotate=180]
\tkzDefPoint(0,1){J}
\tkzDefPoint(0,0){O}
\foreach \i in {0,-5,-10,...,-90}{
 \tkzDefPoint({2.5*cos(\i*pi/180)},{1+2.5*sin(\i*pi/180)}){P}
 \tkzInterLC[R](P,J)(O,1)\tkzGetPoints{N}{M}
 \tkzDrawSegment[color=orange](J,N)
 \tkzDrawPoints[red](N)}
\foreach \i in {-90,-95,...,-175,-180}{
 \tkzDefPoint({2.5*cos(\i*pi/180)},{1+2.5*sin(\i*pi/180)}){P}
 \tkzInterLC[R](P,J)(O,1)\tkzGetPoints{N}{M}
 \tkzDrawSegment[color=orange](J,M)
 \tkzDrawPoints[red](M)}
\end{tikzpicture}
```

18.2.7. Line-circle intersection with option near

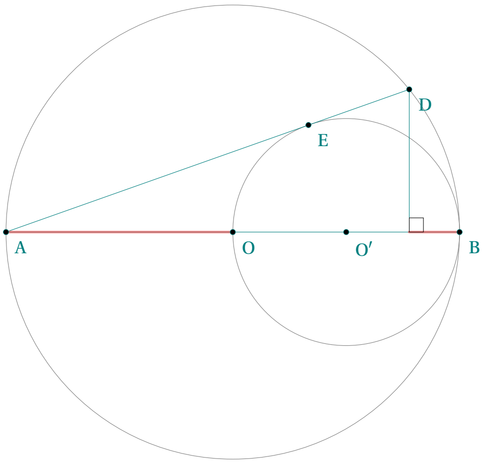
D is the point closest to b.
D 是最接近 b 的点。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,12/0/C}
  \tkzDefGoldenRatio(A,C)                          \tkzGetPoint{B}
  \tkzDefMidPoint(A,C)                             \tkzGetPoint{O}
  \tkzDefMidPoint(A,B)                             \tkzGetPoint{O_1}
  \tkzDefMidPoint(B,C)                             \tkzGetPoint{O_2}
  \tkzDefPointBy[rotation=center O_2 angle 90](C)  \tkzGetPoint{P}
  \tkzDefPointBy[rotation=center O_1 angle 90](B)  \tkzGetPoint{Q}
  \tkzDefPointBy[rotation=center B angle 90](C)    \tkzGetPoint{b}
  \tkzInterLC[near](b,B)(O,A)                      \tkzGetFirstPoint{D}
  \tkzInterCC(D,B)(O,C)                            \tkzGetPoints{V}{U}
  \tkzDefPointBy[projection=onto U--V](O_1)        \tkzGetPoint{M}
  \tkzDefPointBy[projection=onto U--V](O_2)        \tkzGetPoint{N}
  \tkzDrawPoints(A,B,C,O,O_1,O_2,D,U,V,M,N,b)
  \tkzDrawSemiCircles[teal](O,C O_1,B O_2,C)
  \tkzDrawSegments(A,C B,D U,V A,D C,D M,B B,N)
  \tkzDrawArc(D,U)(V)
  \tkzLabelPoints(A,B,C,O,O_1,O_2)
  \tkzLabelPoints[above](D,U,V,M,N)
\end{tikzpicture}
```

18.2.8. More complex example of a line-circle intersection

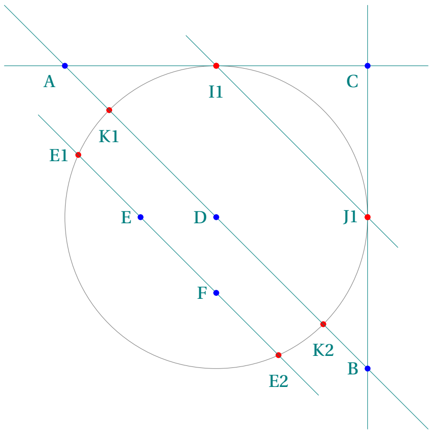
Figure from http://gogeometry.com/problem/p190_tangent_circle



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(0,0){A}
 \tkzDefPoint(8,0){B}
 \tkzDefMidPoint(A,B)               \tkzGetPoint{O}
 \tkzDefMidPoint(O,B)               \tkzGetPoint{O'}
 \tkzDefLine[tangent from=A](O',B) \tkzGetFirstPoint{E}
 \tkzInterLC(A,E)(O,B)              \tkzGetFirstPoint{D}
 \tkzDefPointBy[projection=onto A--B](D)
 \tkzGetPoint{F}
 \tkzDrawCircles(O,B O',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)
 \tkzMarkRightAngle(D,F,B)
 \tkzLabelPoints[below right](A,B,O,O',E,D)
\end{tikzpicture}
```

18.2.9. Circle defined by a center and a measure, and special cases 由圆心和度量定义的圆，以及特殊情况

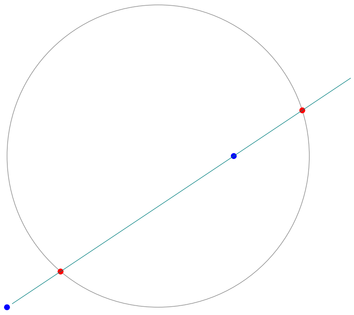
Let’s look at some special cases like straight lines tangent to the circle.
一些特殊情况，比如与圆相切的直线。



```
\begin{tikzpicture}[scale=.5]
 \tkzDefPoint(0,8){A}      \tkzDefPoint(8,0){B}
 \tkzDefPoint(8,8){C}      \tkzDefPoint(4,4){D}
 \tkzDefPoint(2,4){E}      \tkzDefPoint(4,2){F}
 \tkzDefPoint(8,4){G}
 \tkzInterLC(A,C)(D,G)     \tkzGetPoints{I1}{I2}
 \tkzInterLC(B,C)(D,G)     \tkzGetPoints{J1}{J2}
 \tkzInterLC[near](A,B)(D,G)  \tkzGetPoints{K1}{K2}
 \tkzInterLC(E,F)(D,G)     \tkzGetPoints{E1}{E2}
 \tkzDrawCircle(D,G)
 \tkzDrawPoints[color=red](I1,J1,K1,K2,E1,E2)
 \tkzDrawLines(A,B B,C A,C I2,J2 E1,E2)
 \tkzDrawPoints[color=blue](A,...,F)
 \tkzDrawPoints[color=red](I2,J2)
 \tkzLabelPoints[left](B,D,E,F)
 \tkzLabelPoints[below left](A,C)
 \tkzLabelPoints[below=4pt](I1,K1,K2,E2)
 \tkzLabelPoints[left](J1,E1)
\end{tikzpicture}
```

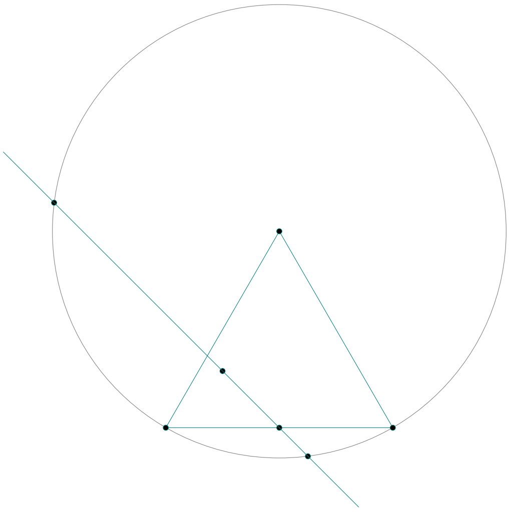
18.2.10. Calculation of radius 半径的计算

With **pgfmath** and \pgfmathsetmacro
The radius measurement may be the result of a calculation that is not done within the intersection macro, but before. A length can be calculated in several ways. It is possible of course, to use the module **pgfmath** and the macro \pgfmathsetmacro. In some cases, the results obtained are not precise enough, so the following calculation 0.0002 ÷ 0.0001 gives 1.98 with pgfmath while xfp will give 2.
半径测量可能是计算的结果，该计算不是在宏 intersection 中完成的，而是在之前完成的。有几种方法可以计算长度。当然有可能，使用模块 **pgfmath** 和宏\pgfmathsetmacro。在某些情况下，得到的结果不够精确，所以下面的计算 0.0002 ÷ 0.0001 用 pgfmath 给出 1.98 而 xfp 给出 2。
With **xfp** and \fpeval:



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

18.2.11. Option "with nodes"



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,4/0/B,1/1/D,2/0/E}
\tkzDefTriangle[equilateral](A,B)
\tkzGetPoint{C}
\tkzInterLC[with nodes](D,E)(C,A,B)
\tkzGetPoints{F}{G}
\tkzDrawCircle(C,A)
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,...,G)
\tkzDrawLine(F,G)
\end{tikzpicture}
```

18.3. Intersection of two circles \tkzInterCC 两圆交点

The most frequent case is that of two circles defined by their center and a point, but as before the option R allows to use the radius measurements.
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[⟨options⟩](⟨O,A⟩)(⟨O′,A′⟩) or (⟨O,r⟩)(⟨O′,r′⟩) or (⟨O,A,B⟩) (⟨O′,C,D⟩)

options	default	definition
N	N	OA and O′A′ are radii, O and O′ are the centers.
R	N	r and r′ are dimensions and measure the radii.
with nodes	N	in (A,A,C)(C,B,F) AC and BF give the radii.
common=pt		pt is common point; tkzFirstPoint gives the other point.

This macro defines the intersection point(s) I and J of the two center circles O and O′. If the two circles do not have a common point then the macro ends with an error that is not handled. If the centers are O and O′ and the intersections are A and B then the angles $\widehat{O,A,O'}$ and $\widehat{O,B,O'}$ are in opposite directions. **tkzFirstPoint** is assigned to the point that forms the ”clockwise” angle.

\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
common=pt		pt 是共同点；tkzFirstPoint 给出了另一点。

该宏定义了圆心为 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。该宏定义了两个圆心为 O 和 O′ 的交点 I 和 J。如果两个圆没有一个公共点，那么宏会以一个无法处理的错误结束。如果圆心是 O 和 O′ 并且交点是 A 和 B，那么角 $\widehat{O,A,O'}$ 和 $\widehat{O,B,O'}$ 在相反的方向。**tkzFirstPoint** 被分配给形成“顺时针”角度的点。

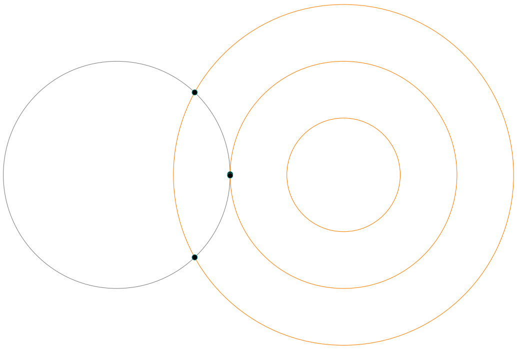
\tkzTestInterCC(⟨O,A⟩)(⟨O′,B⟩)

So the arguments are two couples which define two circles with a center and a point on the circle. If there is a non empty intersection between these two circles then the test \iftkzFlagCC gives true.

\tkzTestInterCC(⟨O,A⟩)(⟨O′,B⟩)

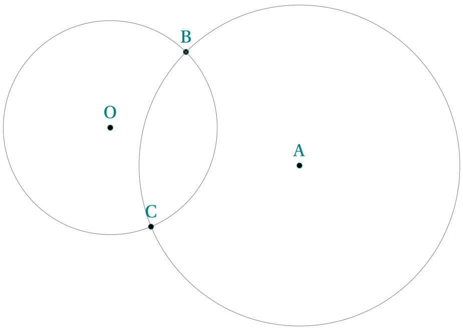
参数是两个：两个圆（圆心和圆上的点）。如果这两个圆之间有一个非空的交点，则 \iftkzFlagCC 给出结论 true。

18.3.1. test circle-circle intersection



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{% x    y    name
                 0    /0    /A,
                 2    /0    /B,
                 4    /0    /I,
                 1    /0    /P}
\tkzDrawCircle(A,B)
\foreach \i in {1,...,3}{%
   \coordinate  (P) at (\i,0);
\tkzDrawCircle[new](I,P)
   \tkzTestInterCC(A,B)(I,P)
    \iftkzFlagCC
    \tkzInterCC(A,B)(I,P)  \tkzGetPoints{a}{b}
    \tkzDrawPoints(a,b)
    \fi}
  \end{tikzpicture}
```

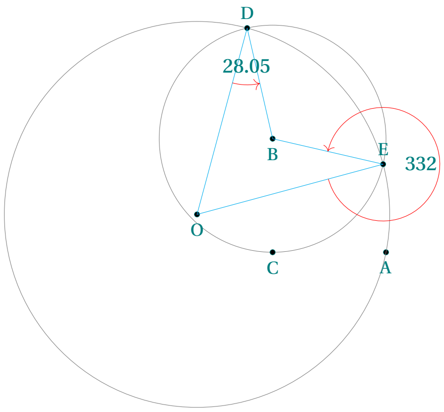
18.3.2. circle-circle intersection with common point.



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/O,5/-1/A,2/2/B}
  \tkzDrawPoints(O,A,B)
  \tkzDrawCircles(O,B A,B)
  \tkzInterCC[common=B](O,B)(A,B)\tkzGetFirstPoint{C}
  \tkzDrawPoint(C)
  \tkzLabelPoints[above](O,A,B,C)
\end{tikzpicture}
```

18.3.3. circle-circle intersection order of points. 两圆交点的次序

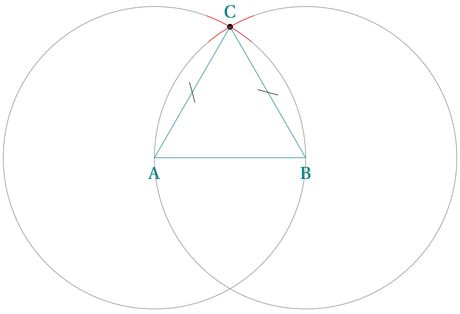
The idea is to compare the angles formed with the first center, a resultant point and the center of the second circle. The first point is the one that corresponds to the smallest angle.
设计是比较与第一个圆心、交点和第二个圆的中心形成的角度。第一个点是对应于最小角度的点。
As you can see $\widehat{ODB} < \widehat{OBE}$



```
\begin{tikzpicture}[scale=.5]
   \pgfkeys{/pgf/number format/.cd,fixed relative,precision=4}
  \tkzDefPoints{0/0/O,5/-1/A,2/2/B,2/-1/C}
  \tkzDrawPoints(O,A,B)
  \tkzDrawCircles(O,A B,C)
  \tkzInterCC(O,A)(B,C)\tkzGetPoints{D}{E}
  \tkzDrawPoints(C,D,E)
  \tkzLabelPoints(O,A,B,C)
  \tkzLabelPoints[above](D,E)
  \tkzDrawSegments[cyan](D,O D,B)
  \tkzMarkAngle[red,->,size=1.5](O,D,B)
  \tkzFindAngle(O,D,B)   \tkzGetAngle{an}
  \tkzLabelAngle(O,D,B){$ \pgfmathprintnumber{\an}$}
  \tkzDrawSegments[cyan](E,O E,B)
  \tkzMarkAngle[red,->,size=1.5](O,E,B)
  \tkzFindAngle(O,E,B)   \tkzGetAngle{an}
  \tkzLabelAngle(O,E,B){$ \pgfmathprintnumber{\an}$}
\end{tikzpicture}
```

18.3.4. Construction of an equilateral triangle. 等边三角形的构造。

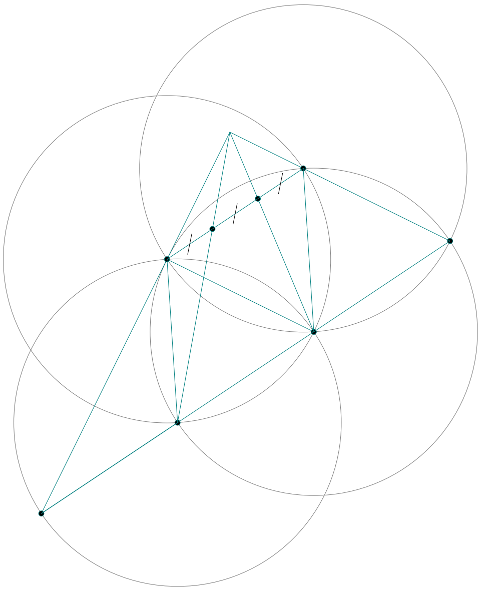
$\widehat{A,C,B}$ is a clockwise angle
$\widehat{A,C,B}$ 是顺时针角度



```
\begin{tikzpicture}[trim left=-1cm,scale=.5]
 \tkzDefPoint(1,1){A}
 \tkzDefPoint(5,1){B}
 \tkzInterCC(A,B)(B,A)\tkzGetPoints{C}{D}
 \tkzDrawPoint[color=black](C)
 \tkzDrawCircles(A,B 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}
```

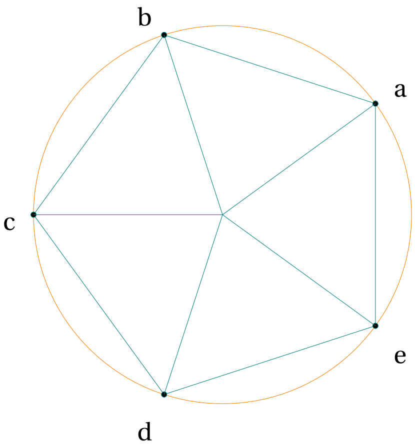
18.3.5. Segment trisection 三等分线段

The idea here is to divide a segment with a ruler and a compass into three segments of equal length.
想法是用直尺和圆规把一个线段分成三个等长的线段。



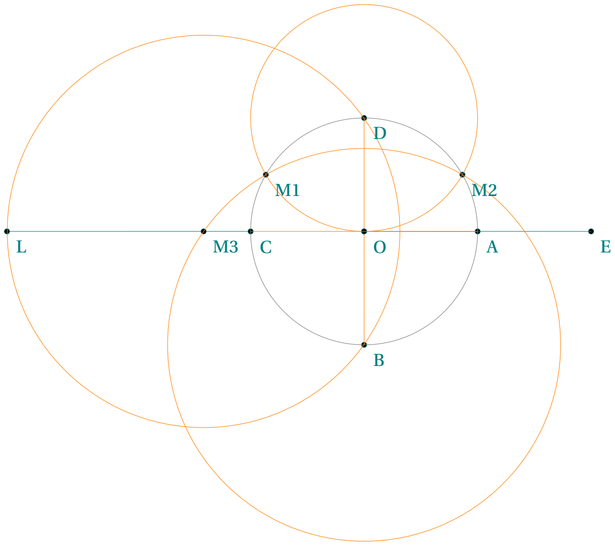
```
\begin{tikzpicture}[scale=.6]
 \tkzDefPoint(0,0){A}
 \tkzDefPoint(3,2){B}
 \tkzInterCC(A,B)(B,A)            \tkzGetSecondPoint{D}
 \tkzInterCC(D,B)(B,A)            \tkzGetPoints{A}{C}
 \tkzInterCC(D,B)(A,B)            \tkzGetPoints{E}{B}
 \tkzInterLC[common=D](C,D)(E,D)  \tkzGetFirstPoint{F}
 \tkzInterLL(A,F)(B,C)            \tkzGetPoint{O}
 \tkzInterLL(O,D)(A,B)            \tkzGetPoint{H}
 \tkzInterLL(O,E)(A,B)            \tkzGetPoint{G}
 \tkzDrawCircles(D,E A,B B,A E,A)
 \tkzDrawSegments[](O,F O,B O,D O,E)
 \tkzDrawPoints(A,...,H)
 \tkzDrawSegments(A,B B,D A,D A,E E,F C,F B,C)
 \tkzMarkSegments[mark=s|](A,G G,H H,B)
\end{tikzpicture}
```

18.3.6. With the option "with nodes"



```
\begin{tikzpicture}[scale=.5]
 \tkzDefPoints{0/0/A,0/5/B,5/0/C}
 \tkzDefPoint(54:5){F}
 \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}
 \tkzDrawCircle[new](A,C)
 \tkzDrawPoints(a,b,c,d,e)
 \tkzDrawPolygon(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(A,\vertex)
 }
\end{tikzpicture}
```

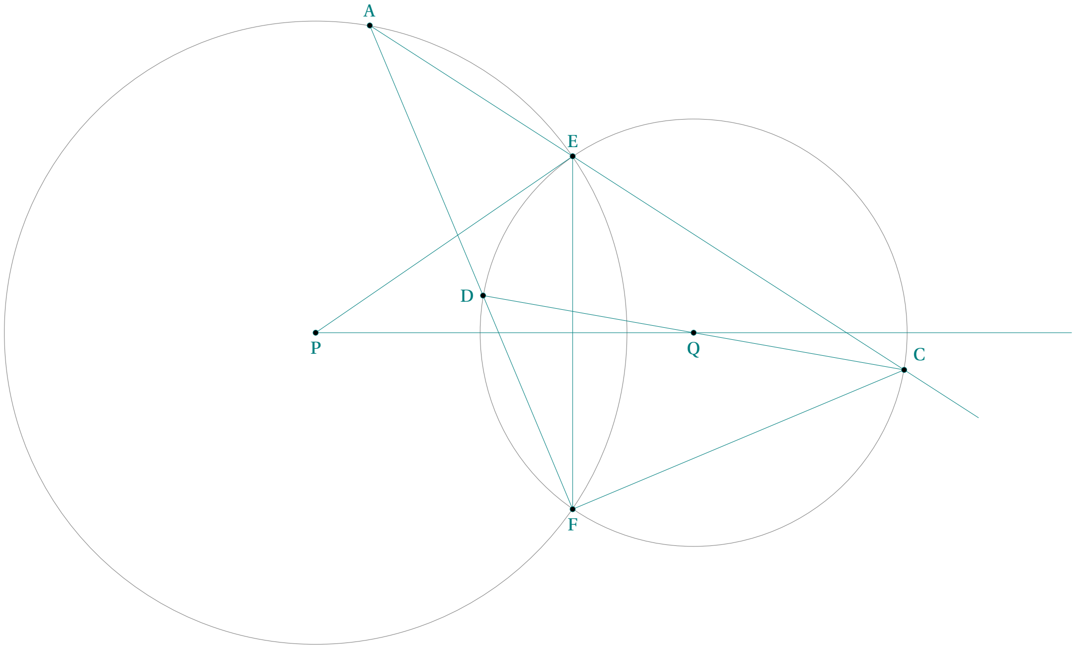
18.3.7. Mix of intersections



```
\begin{tikzpicture}[scale = .75]
  \tkzDefPoint(2,2){A}
  \tkzDefPoint(0,0){B}
  \tkzDefPoint(-2,2){C}
  \tkzDefPoint(0,4){D}
  \tkzDefPoint(4,2){E}
  \tkzCircumCenter(A,B,C)\tkzGetPoint{O}
  \tkzInterCC[R](O,2)(D,2) \tkzGetPoints{M1}{M2}
  \tkzInterCC(O,A)(D,O) \tkzGetPoints{1}{2}
  \tkzInterLC(A,E)(B,M1) \tkzGetSecondPoint{M3}
  \tkzInterLC(O,C)(M3,D) \tkzGetSecondPoint{L}
  \tkzDrawSegments(C,L)
  \tkzDrawPoints(A,B,C,D,E,M1,M2,M3,O,L)
  \tkzDrawSegments(O,E)
  \tkzDrawSegments[new](C,A D,B)
  \tkzDrawPoint(O)
  \tkzDrawCircles[new](M3,D B,M2 D,O)
  \tkzDrawCircle(O,A)
  \tkzLabelPoints[below right](A,B,C,D,E,M1,M2,M3,O,L)
\end{tikzpicture}
```

18.3.8. Altshiller-Court's theorem 阿尔特希勒-考特定理

The two lines joining the points of intersection of two orthogonal circles to a point on one of the circles met the other circle in two diametricaly oposite points. Altshiller p 176
连接两个正交圆的交点与其中一个圆上的一点的两条直线，在两个直径相反的点上与另一个圆相交。阿尔特希勒 p 176

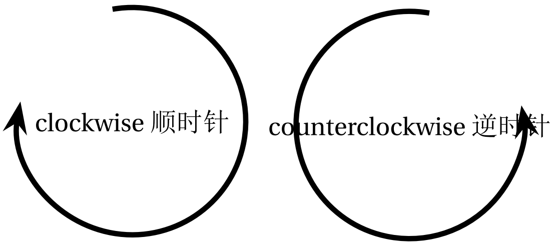


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/P,5/0/Q,3/2/I}
  \tkzDefCircle[orthogonal from=P](Q,I)
  \tkzGetFirstPoint{E}
  \tkzDrawCircles(P,E Q,E)
  \tkzInterCC[common=E](P,E)(Q,E) \tkzGetFirstPoint{F}
  \tkzDefPointOnCircle[through =  center P angle 80 point E]
  \tkzGetPoint{A}
  \tkzInterLC[common=E](A,E)(Q,E)  \tkzGetFirstPoint{C}
  \tkzInterLL(A,F)(C,Q)  \tkzGetPoint{D}
  \tkzDrawLines[add=0 and 1](P,Q)
  \tkzDrawLines[add=0 and 2](A,E)
  \tkzDrawSegments(P,E E,F F,C A,F C,D)
  \tkzDrawPoints(P,Q,E,F,A,C,D)
  \tkzLabelPoints(P,Q,F)
  \tkzLabelPoints[above](E,A)
  \tkzLabelPoints[left](D)
  \tkzLabelPoints[above right](C)
\end{tikzpicture}
```

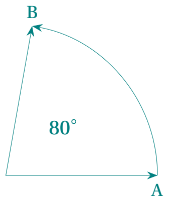
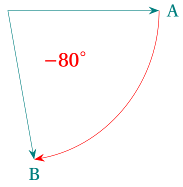
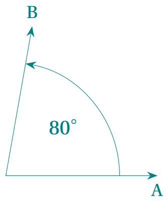
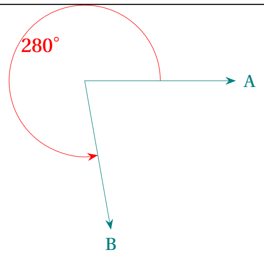
19. Angles 角

19.1. Definition and usage with tkz-euclide 定义和用法

In Euclidean geometry, an angle is the figure formed by two rays, called the sides of the angle, sharing a common endpoint, called the vertex of the angle.[Wikipedia]. A ray with **tkz-euclide** is defined by two points also each angle is defined with three points like $\widehat{AOB}$. The vertex O is the second point. Their order is important because it is assumed that the angle is specified in the direct order (counterclockwise). In trigonometry and mathematics in general, plane angles are conventionally measured counterclockwise, starting with 0° pointing directly to the right (or east), and 90° pointing straight up (or north)[Wikipedia]. Let us agree that an angle measured counterclockwise is positive.
在欧几里得几何中，角是由两条射线形成的图形，称为角的边，它们共用一个端点，称为角的顶点。【维基百科】。**tkz-euclide** 里的射线，由两个点定义；角度由三个点定义，如 $\widehat{AOB}$。顶点 O 是第二个点。它们的顺序很重要，因为假设角度是按直接顺序 (逆时针) 指定的。在一般的三角学和数学中，平面角通常是逆时针测量的，从 0° 直接指向右边 (或东方) 开始，90° 直接指向上方 (或北方)。规定逆时针测量的角度是正的。



`Angles` are involved in several macros like `\tkzDefPoint`,`\tkzDefPointBy[rotation = …]`, `\tkzDrawArc` and the next one `\tkzGetAngle`. With the exception of the last one, all these macros accept negative angles.
`Angles` 涉及几个宏，如`\tkzDefPoint`,`\tkzDefPointBy[rotation = …]`, `\tkzDrawArc`。除了最后一个，所有这些宏都接受负角度。

 Rotation 80° from (O,A) to (O,B) \tkzDefPointBy[rotation=center O angle 80]	 Rotation −80° from (O,A) to (O,B) \tkzDefPointBy[rotation=center O angle -80]
 \tkzFindAngle(A,O,B) gives 80	 \tkzFindAngle(A,O,B) gives 280°

As we can see, the −80° rotation defines a clockwise angle but the macro `\tkzFindAngle` recovers a counterclockwise angle.
正如我们所见，旋转 −80° 定义了一个顺时针方向的角度，但是宏`\tkzFindAngle` 依然以逆时针方向定义角度。

19.2. Recovering an angle \tkzGetAngle

`\tkzGetAngle(⟨name of macro⟩)`

Assigns the value in degree of an angle to a macro. The value is positive and between 0° and 360°. This macro retrieves `\tkzAngleResult` and stores the result in a new macro.

arguments	example	explanation
name of macro	`\tkzGetAngle{ang}`	`\ang` contains the value of the angle.

`\tkzGetAngle(⟨name of macro⟩)`

将角的角度值（以度为单位）指定给宏。该值为正值，介于 0° 和 360° 之间此宏检索`\tkzAngleResult`，并将结果存储在新宏中。

参数	示例	含义
name of macro	`\tkzGetAngle{ang}`	`\ang` 包含角度值。

This is an auxiliary macro that allows you to retrieve the result of the following macro `\tkzFindAngle`.
这是一个辅助宏，允许检索宏`\tkzFindAngle` 的结果。

19.3. Angle formed by three points 三个点形成的角

`\tkzFindAngle(⟨pt1,pt2,pt3⟩)`

The result is stored in a macro `\tkzAngleResult`.

arguments	example	explanation
(pt1,pt2,pt3)	`\tkzFindAngle(A,B,C)`	`\tkzAngleResult` gives the angle $(\overrightarrow{BA},\overrightarrow{BC})$

The measure is always positive and between 0° and 360°. With the usual conventions, a counterclockwise angle smaller than a straight angle has always a measure between 0° and 180°, while a clockwise angle smaller than a straight angle will have a measurement greater than 180°. `\tkzGetAngle` can retrieve the angle.

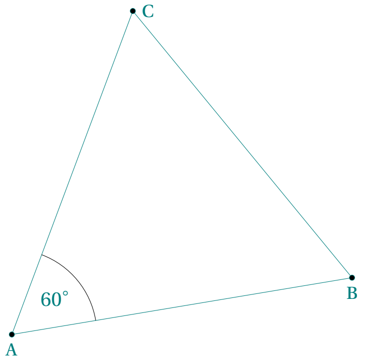
`\tkzFindAngle(⟨pt1,pt2,pt3⟩)`

结果存储在宏`\tkzAngleResult` 中。

参数	示例	含义
(pt1,pt2,pt3)	`\tkzFindAngle(A,B,C)`	`\tkzAngleResult` 给出角 $(\overrightarrow{BA},\overrightarrow{BC})$

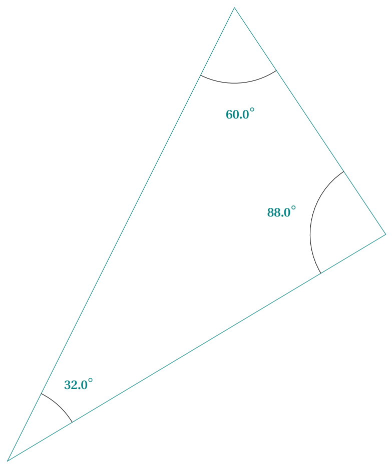
The measure is always positive and between 0° and 360°. With the usual conventions, a counterclockwise angle smaller than a straight angle has always a measure between 0° and 180°, while a clockwise angle smaller than a straight angle will have a measurement greater than 180°. `\tkzGetAngle` can retrieve the angle. 角度的测量值是 0° 和 360° 之间的一个整数。按照通常的惯例，小于直角的逆时针角的度量总是在 0° 和 180° 之间，而小于直角的顺时针角的度量将大于 180°。`\tkzGetAngle` 可以检索到该角度。。

19.3.1. Verification of angle measurement 验证角度测量



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

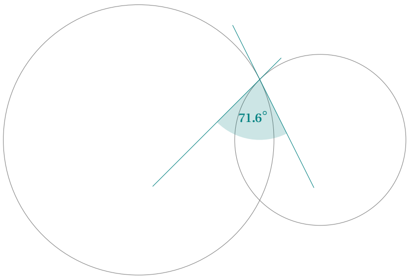
19.3.2. Determination of the three angles of a triangle 三角形的三个角的测定



```
\begin{tikzpicture}
\tikzset{label angle style/.append style={pos=1.4}}
\tkzDefPoints{0/0/a,5/3/b,3/6/c}
\tkzDrawPolygon(a,b,c)
\tkzFindAngle(c,b,a)\tkzGetAngle{angleCBA}
\pgfmathparse{round(1+\angleCBA)}
\let\angleCBA\pgfmathresult
\tkzFindAngle(a,c,b)\tkzGetAngle{angleACB}
\pgfmathparse{round(\angleACB)}
\let\angleACB\pgfmathresult
\tkzFindAngle(b,a,c)\tkzGetAngle{angleBAC}
\pgfmathparse{round(\angleBAC)}
\let\angleBAC\pgfmathresult
\tkzMarkAngle(c,b,a)
\tkzLabelAngle(c,b,a){\tiny $\angleCBA^\circ$}
\tkzMarkAngle(a,c,b)
\tkzLabelAngle(a,c,b){\tiny $\angleACB^\circ$}
\tkzMarkAngle(b,a,c)
\tkzLabelAngle(b,a,c){\tiny $\angleBAC^\circ$}
\end{tikzpicture}
```

19.3.3. Angle between two circles 两个圆之间的角度

We are looking for the angle formed by the tangents at a point of intersection
我们要找的是切线在交点处形成的角度



```
\begin{tikzpicture}[scale=.4]
\pgfkeys{/pgf/number format/.cd,%
          fixed,precision=1}
\tkzDefPoints{0/0/A,6/0/B,4/2/C}
\tkzDrawCircles(A,C B,C)
\tkzDefLine[tangent at=C](A) \tkzGetPoint{a}
\tkzDefPointsBy[symmetry = center C](a){d}
\tkzDefLine[tangent at=C](B) \tkzGetPoint{b}
\tkzDrawLines[add=1 and 4](a,C  C,b)
\tkzFillAngle[fill=teal,opacity=.2%
                        ,size=2](b,C,d)
\tkzFindAngle(b,C,d)\tkzGetAngle{bcd}
\tkzLabelAngle[pos=1.25](b,C,d){%
  \tiny $\pgfmathprintnumber{\bcd}^\circ$}
\end{tikzpicture}
```

19.4. Angle formed by a straight line with the horizontal axis \tkzFindSlopeAngle 直线与水平轴形成的角度

Much more interesting than the last one. The result is between -180 degrees and +180 degrees.
比上一个有趣多了。其结果是在-180 度和+180 度之间。

\tkzFindSlopeAngle(⟨A,B⟩)

Determines the slope of the straight line (AB). The result is stored in a macro **\tkzAngleResult**.

arguments	example	explanation
(pt1,pt2)	**\tkzFindSlopeAngle**(A,B)	

\tkzGetAngle can retrieve the result. If retrieval is not necessary, you can use **\tkzAngleResult**.

\tkzFindSlopeAngle(⟨A,B⟩)

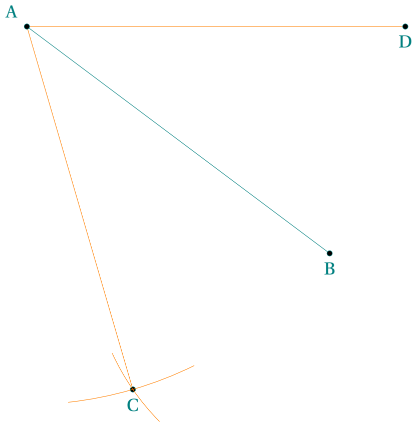
确定直线（AB）的斜率。结果存储在宏**\tkzAngleResult** 中。

参数	示例	含义
(pt1,pt2)	**\tkzFindSlopeAngle**(A,B)	

\tkzGetAngle 可以检索到结果。如果不需要检索，可以使用 **\tkzAngleResult**。

19.4.1. How to use \tkzFindSlopeAngle

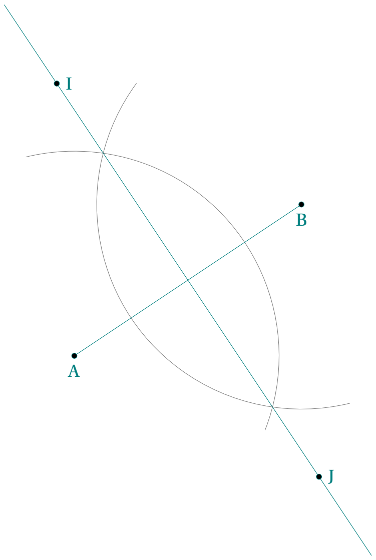
The point here is that (AB) is the bisector of $\widehat{CAD}$, such that the AD slope is zero. We recover the slope of (AB) and then rotate twice.
这里的重点是 (AB) 是 $\widehat{CAD}$ 的平分线，这样 AD 的斜率为零。得到 (AB) 的斜率，然后进行两次旋转。



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

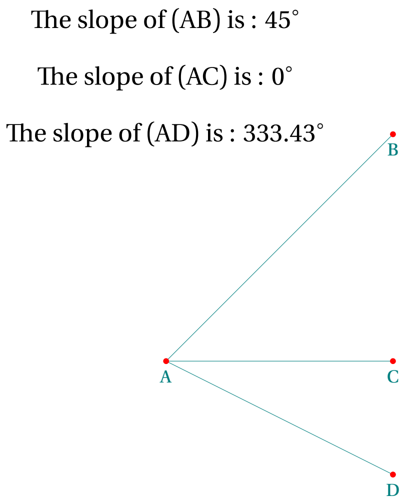
19.4.2. Use of \tkzFindSlopeAngle and \tkzGetAngle

Here is another version of the construction of a mediator
这是另外一个构造中点的方法



```
\begin{tikzpicture}
 \tkzInit
 \tkzDefPoint(0,0){A}        \tkzDefPoint(3,2){B}
 \tkzDefLine[mediator](A,B)  \tkzGetPoints{I}{J}
 \tkzCalcLength(A,B)         \tkzGetLength{dAB}
 \tkzFindSlopeAngle(A,B)     \tkzGetAngle{tkzangle}
 \begin{scope}[rotate=\tkzangle]
   \tkzSetUpArc[color=gray,line width=0.2pt,/tkzcompass/delta=10]
   \tkzDrawArc[R,arc](B,3/4*\dAB)(120,240)
   \tkzDrawArc[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}
```

19.4.3. Another use of \tkzFindSlopeAngle


The slope of (AB) is : 45°
The slope of (AC) is : 0°
The slope of (AD) is : 333.43°
B
A
C
D

```
\begin{tikzpicture}[scale=1.5]
  \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)
  \tkzFindSlopeAngle(A,B)\tkzGetAngle{SAB}
  \tkzFindSlopeAngle(A,C)\tkzGetAngle{SAC}
  \tkzFindSlopeAngle(A,D)\tkzGetAngle{SAD}
  \pgfkeys{/pgf/number format/.cd,fixed,precision=2}
  \tkzText(1,5){The slope of (AB) is :
     $\pgfmathprintnumber{\SAB}^\circ$}
  \tkzText(1,4.5){The slope of (AC) is :
     $\pgfmathprintnumber{\SAC}^\circ$}
  \tkzText(1,4){The slope of (AD) is :
     $\pgfmathprintnumber{\SAD}^\circ$}
\end{tikzpicture}
```

20. Random point definition

At the moment there are four possibilities:
可以使用以下四种方式定义随机点：

1. point in a rectangle;
2. on a segment;
3. on a straight line;
4. on a circle.

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

20.1. Obtaining random points 定义随机点

This is the new version that replaces `\tkzGetRandPointOn`.

`\tkzDefRandPointOn[⟨local options⟩]`

The result is a point with a random position that can be named with the macro `\tkzGetPoint`. It is possible to use `tkzPointResult` if it is not necessary to retain the results.

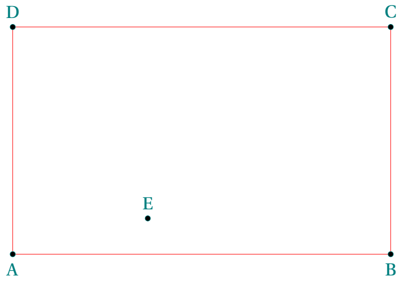
options	default	definition
`rectangle=pt1 and pt2`		`[rectangle=A and B]`
`segment= pt1--pt2`		`[segment=A--B]`
`line=pt1--pt2`		`[line=A--B]`
`circle =center pt1 radius dim`		`[circle = center A radius 2]`
`circle through=center pt1 through pt2`		`[circle through= center A through B]`
`disk through=center pt1 through pt2`		`[disk through=center A through B]`

`\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。

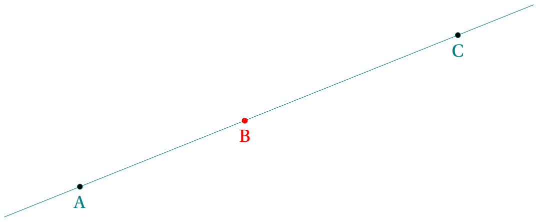
选项	默认值	含义
`rectangle=pt1 and pt2`		`[rectangle=A and B]`
`segment= pt1--pt2`		`[segment=A--B]`
`line=pt1--pt2`		`[line=A--B]`
`circle =center pt1 radius dim`		`[circle = center A radius 2 cm]`
`circle through=center pt1 through pt2`		`[circle through= center A through B]`
`disk through=center pt1 through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

20.1.1. Random point in a rectangle 矩形内的随机点



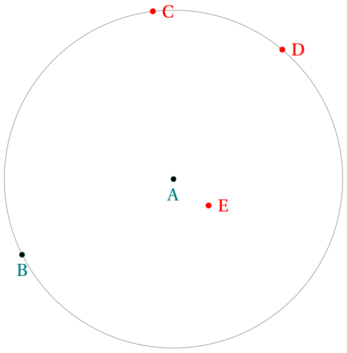
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,5/3/C}
  \tkzDefRandPointOn[rectangle = A and C]
  \tkzGetPoint{E}
  \tkzDefRectangle(A,C)\tkzGetPoints{B}{D}
  \tkzDrawPolygon[red](A,...,D)
  \tkzDrawPoints(A,...,E)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above](C,D,E)
\end{tikzpicture}
```

20.1.2. Random point on a segment or a line 线段上的随机点



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,5/2/C}
  \tkzDefRandPointOn[segment = A--C]\tkzGetPoint{B}
  \tkzDrawLine(A,C)
  \tkzDrawPoints(A,C) \tkzDrawPoint[red](B)
  \tkzLabelPoints(A,C) \tkzLabelPoints[red](B)
\end{tikzpicture}
```

20.1.3. Random point on a circle or a disk 圆周或圆内的随机点



```
\begin{tikzpicture}
\tkzDefPoints{3/2/A,1/1/B}
\tkzCalcLength(A,B) \tkzGetLength{rAB}
\tkzDefRandPointOn[circle = center A radius \rAB]
\tkzGetPoint{C}
\tkzDefRandPointOn[circle through= center A through B]
\tkzGetPoint{D}
\tkzDefRandPointOn[disk through=center A through B]
\tkzGetPoint{E}
\tkzDrawCircle(A,B)
\tkzDrawPoints(A,B)
\tkzLabelPoints(A,B)
\tkzDrawPoints[red](C,D,E)
\tkzLabelPoints[red,right](C,D,E)
\end{tikzpicture}
```

Part IV.

Drawing and Filling 绘制和填充

21. Drawing 绘制

tkz-euclide can draw 5 types of objects : point, line or line segment, circle, arc and sector.
tkz-euclide 可以绘制 5 种类型的对象: 点、直线或线段、圆、弧和扇形。

21.1. Draw a point or some points 绘制一个点或几个点

There are two possibilities : \tkzDrawPoint for a single point or \tkzDrawPoints for one or more points.
有两种方法:\tkzDrawPoint 用于单个点，\tkzDrawPoints 用于一个或多个点。

21.1.1. Drawing points \tkzDrawPoint

`\tkzDrawPoint[⟨local options⟩](⟨name⟩)`

arguments	default	definition
name of point	no default	Only one point name is accepted

The argument is required. The disc takes the color of the circle, but lighter. It is possible to change everything. The point is a node and therefore it is invariant if the drawing is modified by scaling.

options	default	definition
TikZ options		all TikZ options are valid.
shape	circle	Possible cross or cross out
size	6	6× \pgflinewidth
color	black	the default color can be changed

We can create other forms such as cross

`\tkzDrawPoint[⟨local options⟩](⟨name⟩)`

参数	示例	含义
点的名称	无	只能是一个点的名称

The argument is required. The disc takes the color of the circle, but lighter. It is possible to change everything. The point is a node and therefore it is invariant if the drawing is modified by scaling. 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
TikZ options		所有 TikZ 参数有效.
shape	circle	可以使用 cross 或 cross out
size	6	6× \pgflinewidth
color	black	默认颜色，可以修改

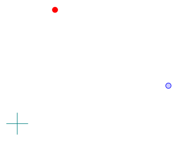
能够创建如 cross 的形式

By default, point style is defined like this :
默认情况下，point style 定义如下:

```
\tikzset{point style/.style = {%
         draw         = black,
         inner sep    = 0pt,
         shape        = circle,
         minimum size = 3 pt,
         fill         = black
                            }
      }
```

21.1.2. Example of point drawings

Note that `scale` does not affect the shape of the dots. Which is normal. Most of the time, we are satisfied with a single point shape that we can define from the beginning, either with a macro or by modifying a configuration file.
注意，缩放 `scale` 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=.5]
 \tkzDefPoint(1,3){A}
 \tkzDefPoint(4,1){B}
 \tkzDefPoint(0,0){O}
 \tkzDrawPoint[color=red](A)
 \tkzDrawPoint[fill=blue!20,draw=blue](B)
 \tkzDrawPoint[shape=cross,size=8pt,color=teal](O)
\end{tikzpicture}
```

It is possible to draw several points at once but this macro is a little slower than the previous one. Moreover, we have to make do with the same options for all the points.
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[⟨local options⟩](⟨liste⟩)
```

arguments	default	definition
points list	no default	example \tkzDrawPoints(A,B,C)

options	default	definition
shape	circle	Possible **cross** or **cross out**
size	6	6× **\pgflinewidth**
color	black	the default color can be changed

Beware of the final "s", an oversight leads to cascading errors if you try to draw multiple points. The options are the same as for the previous macro.

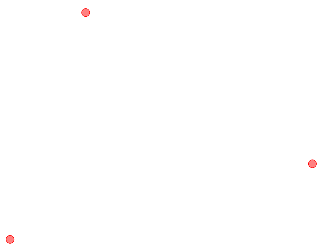
```
\tkzDrawPoints[⟨命令选项⟩](⟨点列表⟩)
```

参数	默认值	含义
点列表	无	例如：\tkzDrawPoints(A,B,C)，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× **\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。

21.1.3. Example



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

22. Drawing the lines 绘制直线

The following macros are simply used to draw, name lines.
下面的宏只是用来绘制直线，命名直线。

22.1. Draw a straight line

To draw a normal straight line, just give a couple of points. You can use the `add` option to extend the line (This option is due to **Mark Wibrow**, see the code below).
要绘制一条常规的直线，只需给出几个点。可以使用 `add` 选项来扩展该直线 (此选项源于 **Mark Wibrow**，参见下面的代码)。
The style of a line is by default :
默认情况下，直线的样式为:

```
\tikzset{line style/.style = {%
  line width = 0.6pt,
  color      = black,
  style      = solid,
  add        = {.2} and  {.2}%
 }}
```

with

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

You can modify this style with **\tkzSetUpLine** see 39.1
可以使用 **\tkzSetUpLine** 来修改此样式，参见 39.1

\tkzDrawLine[⟨local options⟩](⟨pt1,pt2⟩)

The arguments are a list of two points or three points. It would be possible, as for a half line, to create a style with \add.

options	default	definition
TikZ options		all TikZ options are valid.
add	0.2 and 0.2	add = kl and kr, …
…	…	allows the segment to be extended to the left and right.

add defines the length of the line passing through the points pt1 and pt2. Both numbers are percentages. The styles of TikZ are accessible for plots.

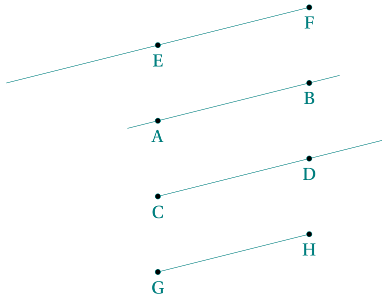
\tkzDrawLine[⟨local options⟩](⟨pt1,pt2⟩)

参数是两点或三点的列表。对于射线，可以使用 \add 创建样式。

选项	默认值	含义
TikZ 选项		所有 TikZ 选项有效.
add	0.2 and 0.2	add = kl and kr, …
…	…	允许线段向左和向右延伸。

add 定义通过点 pt1 和 pt2 的线段长度。两个数字都是百分比。TikZ 的样式可用于绘图。

22.1.1. Examples with add



```
\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}
```

It is possible to draw several lines, but with the same options.
可以画几条线，但是有相同的选项。

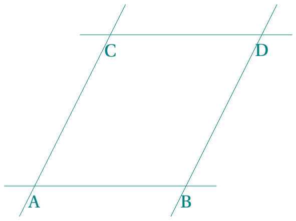
\tkzDrawLines[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)

Arguments are a list of pairs of points separated by spaces. The styles of TikZ are available for the draws.

\tkzDrawLines[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)

参数是由空格分隔的点对的列表。TikZ 的样式可供绘图使用。

22.1.2. Example with \tkzDrawLines



```
\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}
```

23. Drawing a segment 绘制线段

There is, of course, a macro to simply draw a segment.
当然，有一个宏可以简单地绘制一条线段。

23.1. Draw a segment \tkzDrawSegment 绘制一条线段

`\tkzDrawSegment[⟨local options⟩](⟨pt1,pt2⟩)`

The arguments are a list of two points. The styles of Ti*k*Z are available for the drawings.

argument	example	definition
(pt1,pt2)	(A,B)	draw the segment [A,B]

options	example	definition
TikZ options		all TikZ options are valid.
dim	no default	dim = {label,dim,option}, …
…	…	allows you to add dimensions to a figure.

This is of course equivalent to `\draw (A)--(B);`. You can also use the option `add`.

23.2. Draw a segment \tkzDrawSegment 绘制一条线段

`\tkzDrawSegment[⟨local options⟩](⟨pt1,pt2⟩)`

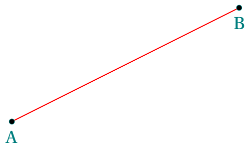
参数是两点的列表。Ti*k*Z 样式可用于绘图。

argument	example	definition
(pt1,pt2)	(A,B)	绘制线段 [A,B]

options	example	definition
TikZ 选项		所 TikZ 选项有效.
dim	无	dim = {label,dim,option}, …
…	…	允许向图形添加尺寸。

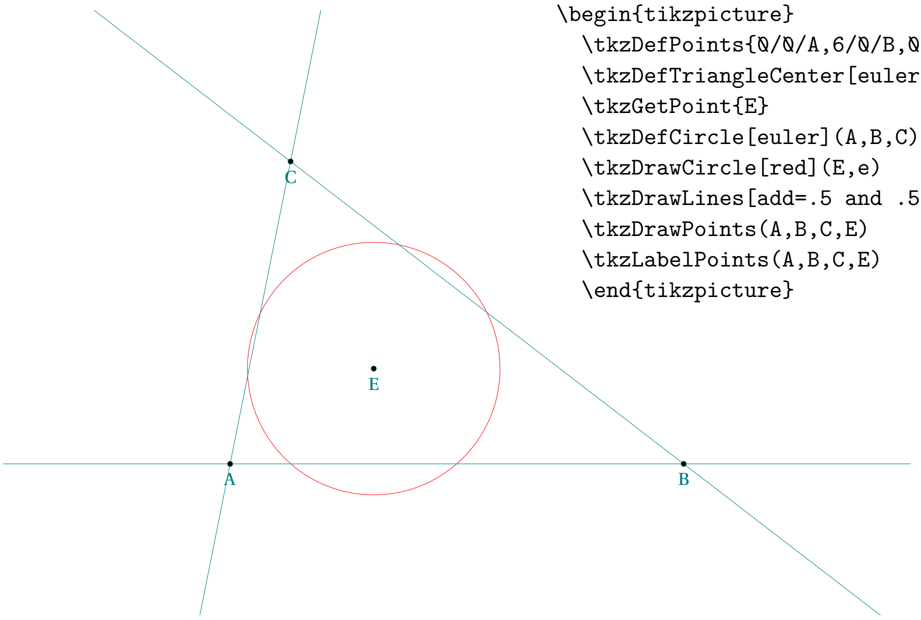
This is of course equivalent to `\draw (A)--(B);`. You can also use the option `add`. 这当然等同于`\draw (A)--(B);`. 也可以使用选项 `add`。

23.2.1. Example with point references 带有点参照的示例



```
\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}
```

23.2.2. Example of extending an segment with option add 使用选项 add 扩展段的示例

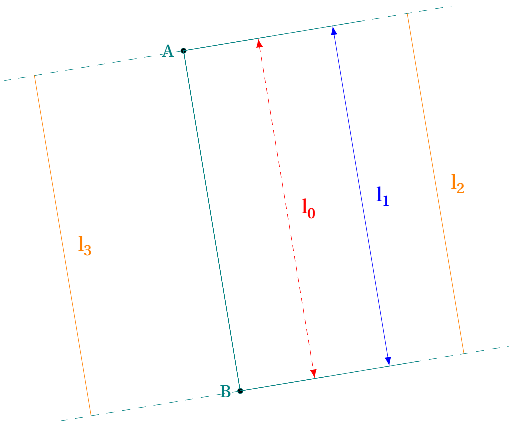


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
  \tkzDefTriangleCenter[euler](A,B,C)
  \tkzGetPoint{E}
  \tkzDefCircle[euler](A,B,C)\tkzGetPoints{E}{e}
  \tkzDrawCircle[red](E,e)
  \tkzDrawLines[add=.5 and .5](A,B A,C B,C)
  \tkzDrawPoints(A,B,C,E)
  \tkzLabelPoints(A,B,C,E)
  \end{tikzpicture}
```

23.2.3. Adding dimensions with option dim new code from Muzimuzhi Z 来自 Muzimuzhi Z 使用选项添加尺寸 dim 的新代码

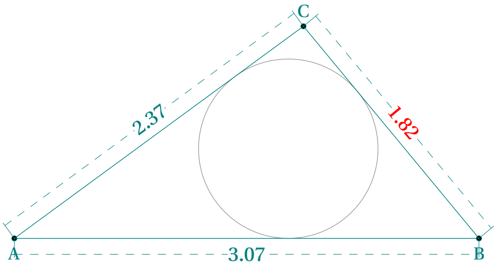
This code comes from an answer to this question on tex.stackexchange.com (change-color-and-style-of-dimension-lines-in-tkz-euclide). The code of **dim** is based on options of TikZ, you must add the units. You can use now two styles : dim style and dim fence style. You have several ways to use them. I'll let you look at the examples to see what you can do with these styles. 代码来自于 tex.stackexchange.com 对这个问题的一个回答 (在 tkz-euclide 中改变尺寸线的颜色和样式)。**dim** 的代码基于 TikZ 选项，必须添加单位。现在可以使用两种样式:dim style（暗淡样式）和 dim fence style（暗淡栅栏样式）。有几种方法来使用它们。看看这些示例，看看可以用这些样式做些什么。

```
\tikzset{dim style/.append style={dashed}} % append if you want to keep precedent style.
or
\begin{scope}[ dim style/.append style={orange},
    dim fence style/.style={dashed}]
```



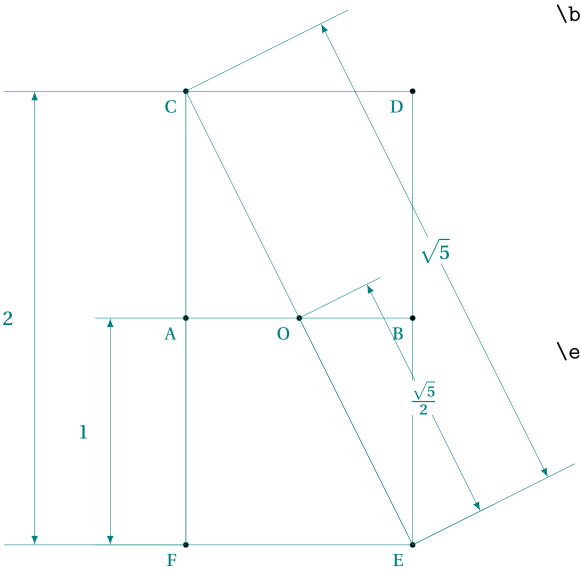
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/3/A, 1/-3/B}
  \tkzDrawPoints(A,B)
  \tkzDrawSegment[dim={\(l_0\),1cm,right=2mm},
    dim style/.append style={red,
    dash pattern={on 2pt off 2pt}}](A,B)
  \tkzDrawSegment[dim={\(l_1\),2cm,right=2mm},
    dim style/.append style={blue}](A,B)
  \begin{scope}[ dim style/.style={orange},
      dim fence style/.style={dashed}]
    \tkzDrawSegment[dim={\(l_2\),3cm,right=2mm}](A,B)
    \tkzDrawSegment[dim={\(l_3\),-2cm,right=2mm}](A,B)
  \end{scope}
  \tkzLabelPoints[left](A,B)
\end{tikzpicture}
```

23.2.4. Adding dimensions with option dim partI 添加尺寸



```
\begin{tikzpicture}[scale=2]
\pgfkeys{/pgf/number format/.cd,fixed,precision=2}
\tkzDefPoint(0,0){A}
\tkzDefPoint(3.07,0){B}
\tkzInterCC[R](A,2.37)(B,1.82)
\tkzGetPoints{C}{C'}
\tkzDefCircle[in](A,B,C) \tkzGetPoints{G}{g}
\tkzDrawCircle(G,g)
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\tkzCalcLength(A,B)\tkzGetLength{ABl}
\tkzCalcLength(B,C)\tkzGetLength{BCl}
\tkzCalcLength(A,C)\tkzGetLength{ACl}
\begin{scope}[dim style/.style={dashed,sloped,teal}]
  \tkzDrawSegment[dim={\pgfmathprintnumber\BCl,6pt,
                                          text=red}](C,B)
  \tkzDrawSegment[dim={\pgfmathprintnumber\ACl,6pt,}](A,C)
  \tkzDrawSegment[dim={\pgfmathprintnumber\ABl,-
6pt,}](A,B)
\end{scope}
\tkzLabelPoints(A,B) \tkzLabelPoints[above](C)
\end{tikzpicture}
```

23.2.5. Adding dimensions with option dim part II 添加尺寸



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O,-2/0/A,2/0/B,
                -2/4/C,2/4/D,2/-4/E,-2/-4/F}
  \tkzDrawPolygon(C,...,F)
  \tkzDrawSegments(A,B)
  \tkzDrawPoints(A,...,F,O)
  \tkzLabelPoints[below left](A,...,F,O)
  \tkzDrawSegment[dim={ $\sqrt{5}$,2cm,}](C,E)
  \tkzDrawSegment[dim={ $\frac{\sqrt{5}}{2}$,1cm,}](O,E)
  \tkzDrawSegment[dim={ $2$,2cm,left=8pt}](F,C)
  \tkzDrawSegment[dim={ $1$,1cm,left=8pt}](F,A)
\end{tikzpicture}
```

23.3. Drawing segments \tkzDrawSegments 绘制多条线段

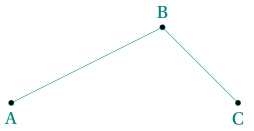
If the options are the same we can plot several segments with the same macro.
用相同的宏，可以以相同的选项，绘制几条线段。

`\tkzDrawSegments[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)`

The arguments are a two-point couple list. The styles of Ti*k*Z are available for the plots.

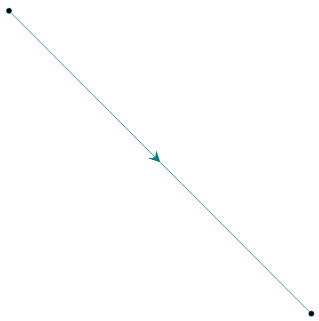
`\tkzDrawSegments[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)`

参数是两点列表。Ti*k*Z 样式可用于绘图。



```
\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}
```

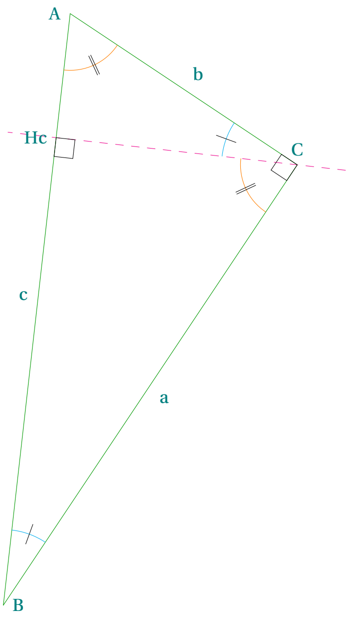
23.3.1. Place an arrow on segment 在线段上放置箭头



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

23.4. Drawing line segment of a triangle 绘制三角形的线段

23.4.1. How to draw Altitude 如何绘制高



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

23.5. Drawing a polygon 绘制多边形

\tkzDrawPolygon[⟨local options⟩](⟨points list⟩)

Just give a list of points and the macro plots the polygon using the Ti*k*Z options present. You can replace (A, B, C, D, E) by (A, ..., E) and (P_1, P_2, P_3, P_4, P_5) by (P_1, P..., P_5)

arguments	example	explanation
(⟨pt1,pt2,pt3,...⟩)	\tkzDrawPolygon[gray,dashed](A,B,C)	Drawing a triangle

options	default	example
Options TikZ	...	\tkzDrawPolygon[red,line width=2pt](A,B,C)

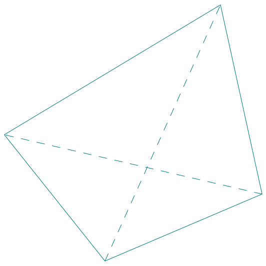
```
\tkzDrawPolygon[⟨local options⟩](⟨points list⟩)
```

只需给出一个点列表，宏就可以使用当前的 TikZ 选项绘制多边形。可以用 (A, ..., E) 代替 (A, B, C, D, E), 用 $(P_1, P..., P_5)$wafw $(P_1, P_2, P_3, P_4, P_5)$

参数	示例	含义
(⟨pt1,pt2,pt3,...⟩)	\tkzDrawPolygon[gray,dashed](A,B,C)	绘制三角形

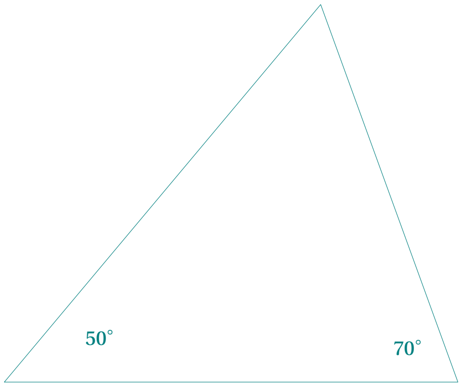
选项	默认值	含义
TikZ 选项	...	\tkzDrawPolygon[red,line width=2pt](A,B,C)

23.5.1. \tkzDrawPolygon



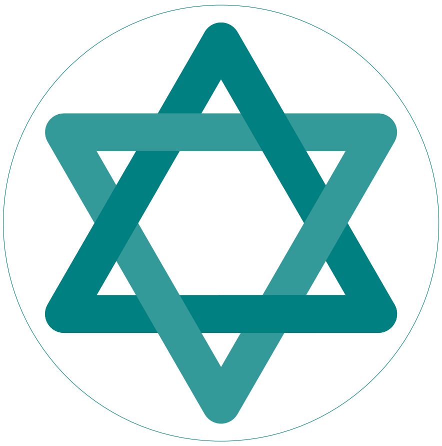
```
\begin{tikzpicture} [rotate=18,scale=1]
 \tkzDefPoints{0/0/A,2.25/0.2/B,2.5/2.75/C,-0.75/2/D}
 \tkzDrawPolygon(A,B,C,D)
 \tkzDrawSegments[style=dashed](A,C B,D)
\end{tikzpicture}
```

23.5.2. Option two angles



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(6,0){B}
\tkzDefTriangle[two angles = 50 and 70](A,B) \tkzGetPoint{C}
\tkzDrawPolygon(A,B,C)
\tkzLabelAngle[pos=1.4](B,A,C){$50^\circ$}
\tkzLabelAngle[pos=0.8](C,B,A){$70^\circ$}
\end{tikzpicture}
```

23.5.3. Style of line 直线的样式



```
\begin{tikzpicture}[scale=.6]
\tkzSetUpLine[line width=5mm,color=teal]
\tkzDefPoint(0,0){O}
\foreach \i in {0,...,5}{%
 \tkzDefPoint({30+60*\i}:4){p\i}}
\tkzDefMidPoint(p1,p3) \tkzGetPoint{m1}
\tkzDefMidPoint(p3,p5) \tkzGetPoint{m3}
\tkzDefMidPoint(p5,p1) \tkzGetPoint{m5}
\tkzDrawPolygon[line join=round](p1,p3,p5)
\tkzDrawPolygon[teal!80,
line join=round](p0,p2,p4)
\tkzDrawSegments(m1,p3 m3,p5 m5,p1)
\tkzDefCircle[R](O,4.8)\tkzGetPoint{o}
\tkzDrawCircle[teal](O,o)
\end{tikzpicture}
```

23.6. Drawing a polygonal chain 绘制多边形链

```
\tkzDrawPolySeg[⟨local options⟩](⟨points list⟩)
```

Just give a list of points and the macro plots the polygonal chain using the Ti*k*Z options present.

arguments	example	explanation
(⟨pt1,pt2,pt3,...⟩)	\tkzDrawPolySeg[gray,dashed](A,B,C)	Drawing a triangle

options	default	example
Options TikZ	...	\tkzDrawPolySeg[red,line width=2pt](A,B,C)

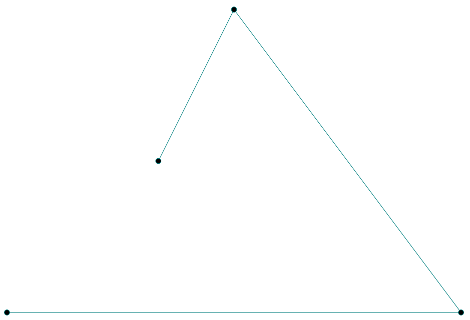
```
\tkzDrawPolySeg[⟨local options⟩](⟨points list⟩)
```

只需给出一个点的列表，宏就可以使用当前的 Ti*k*Z 选项绘制多边形链。

参数	示例	含义
(⟨pt1,pt2,pt3,...⟩)	\tkzDrawPolySeg[gray,dashed](A,B,C)	绘制三角形

选项	默认值	含义
TikZ 选项	...	\tkzDrawPolySeg[red,line width=2pt](A,B,C)

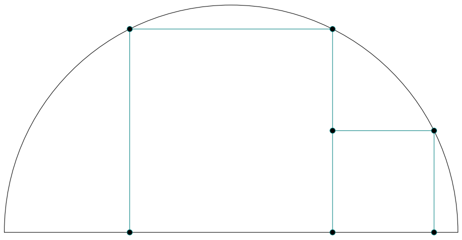
23.6.1. Polygonal chain 多边形链



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

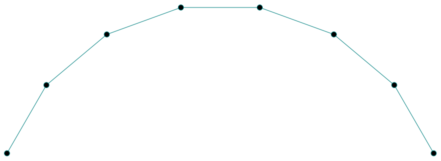
23.6.2. The idea is to inscribe two squares in a semi-circle. 半圆中的矩形示例

A Sangaku look! It is a question of proving that one can inscribe in a half-disc, two squares, and to determine the length of their respective sides according to the radius.
一个很好的示例! 这是一个可以在一个半圆上内接两个正方形，并根据半径确定其各自边长的问题。



```
\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(I,B)(A)
  \tkzDrawPolySeg(H,E,F,G)
  \tkzDrawPolySeg(J,K,L)
  \tkzDrawPoints(E,G,H,F,J,K,L)
\end{tikzpicture}
```

23.6.3. Polygonal chain: index notation 多边形链: 索引符号



```
\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}
```

24. Draw a circle with \tkzDrawCircle 绘制圆

24.1. Draw one circle 绘制个圆

```
\tkzDrawCircle[⟨local options⟩](⟨A,B⟩)
```

Attention you need only two points to define a radius. An additional option R is available to give a measure directly.

arguments	example	explanation
(⟨pt1,pt2⟩)	(⟨A,B⟩)	A center through B

Of course, you have to add all the styles of Ti*k*Z for the tracings...

```
\tkzDrawCircle[⟨local options⟩](⟨A,B⟩)
```

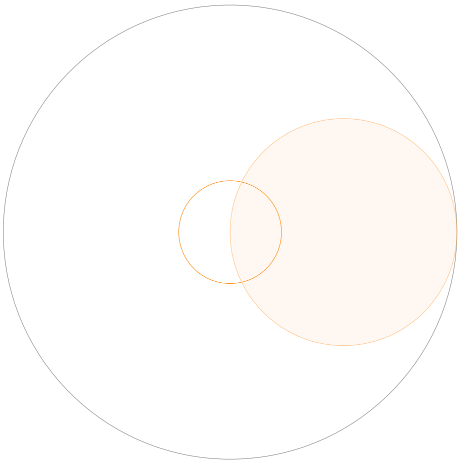
注意只需要两个点来定义半径。另外一个选项R可用于直接给出度量值。

参数	示例	含义
(⟨pt1,pt2⟩)	(⟨A,B⟩)	圆心 A，经过点 B

当然，必须添加所有的 Ti*k*Z 样式...

24.1.1. Circles and styles, draw a circle and color the disc 圆和样式，绘制一个圆并给圆着色

We'll see that it's possible to colour in a disc while tracing the circle.
将会看到，在绘制一个圆的同时，可以给圆着色。



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(3,0){A}
 % circle with center O and passing through A
  \tkzDrawCircle(O,A)
 % diameter circle $[OA]$
 \tkzDefCircle[diameter](O,A) \tkzGetPoint{I}
 \tkzDrawCircle[new,fill=orange!10,opacity=.5](I,A)
 % circle with center O and radius = exp(1) cm
  \edef\rayon{\fpeval{0.25*exp(1)}}
  \tkzDefCircle[R](O,\rayon) \tkzGetPoint{o}
   \tkzDrawCircle[color=orange](O,o)
\end{tikzpicture}
```

24.2. Drawing circles 绘制圆

\tkzDrawCircles[⟨local options⟩](⟨A,B C,D …⟩)

Attention, the arguments are lists of two points. The circles that can be drawn are the same as in the previous macro. An additional option **R** is available to give a measure directly.

arguments	example	explanation
(⟨pt1,pt2 pt3,pt4 ...⟩)	(⟨A,B C,D⟩)	List of two points

options	default	definition
through	through	circle with two points defining a radius

You do not need to use the default option **through**. Of course, you have to add all the styles of Ti*k*Z for the tracings...

\tkzDrawCircles[⟨local options⟩](⟨A,B C,D …⟩)

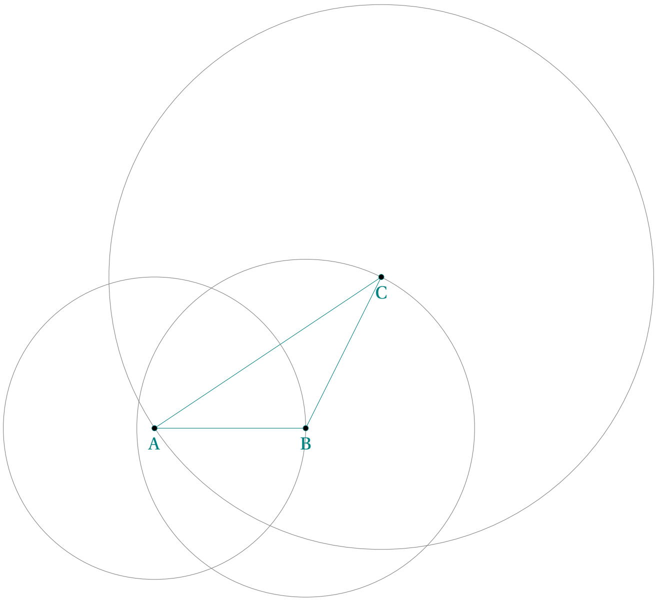
注意，参数是两点列表。可以绘制的圆与前面的宏中的相同。还有一个选项 **R** 可用于直接给出度量值。

参数	示例	含义
(⟨pt1,pt2 pt3,pt4 ...⟩)	(⟨A,B C,D⟩)	两点的列表

选项	默认值	含义
through	through	用两点定义半径的圆

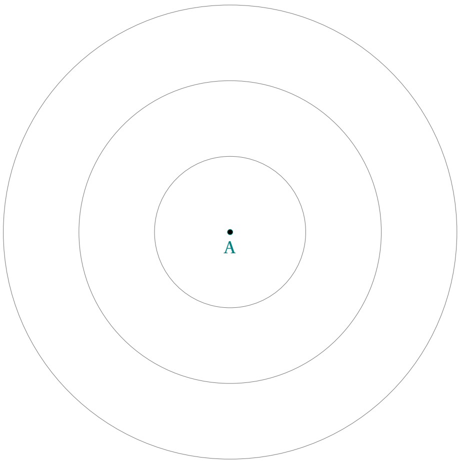
不需要使用默认选项 **through**。当然，必须添加所有的 Ti*k*Z 样式...

24.2.1. Circles defined by a triangle. 三角形定义的圆



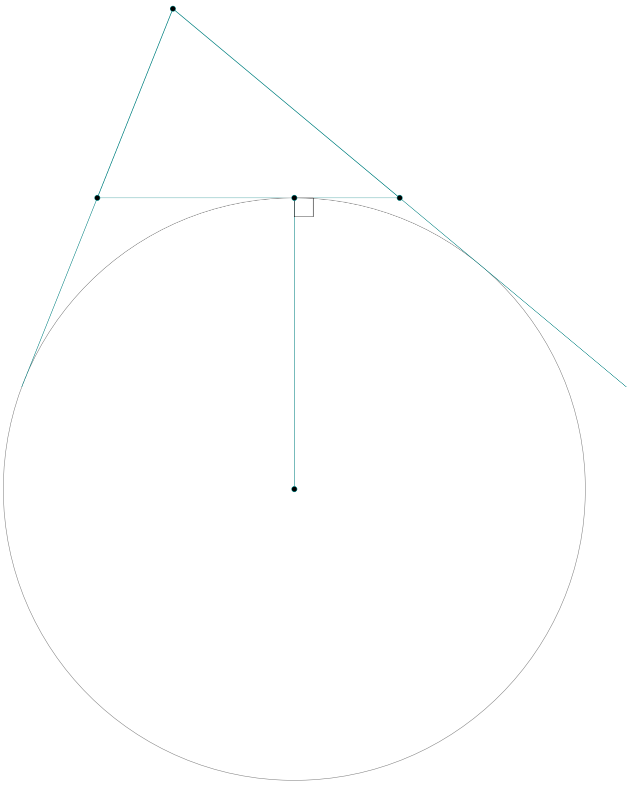
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,2/0/B,3/2/C}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawCircles(A,B B,C C,A)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

24.2.2. Concentric circles. 同心圆



```
\begin{tikzpicture}
   \tkzDefPoints{0/0/A,1/0/a,2/0/b,3/0/c}
   \tkzDrawCircles(A,a A,b A,c)
   \tkzDrawPoint(A)
   \tkzLabelPoints(A)
\end{tikzpicture}
```

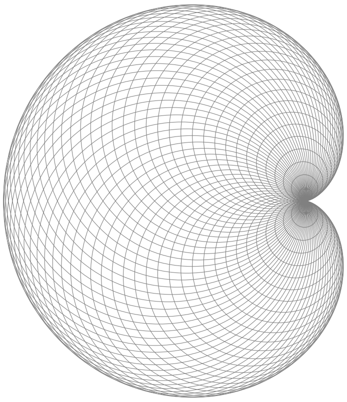
24.2.3. Exinscribed circles. 旁切圆



```
\begin{tikzpicture}[scale=1]
\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}
\tkzDrawCircle(J_c,T_c)
\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}
```

24.2.4. Cardioid 心形

Based on an idea by O. Reboux made with pst-eucl (Pstricks module) by D. Rodriguez.
基于 O. Reboux 的一个想法，用 D. Rodriguez 的 pst-eucl (Pstricks 模块) 制作。
Its name comes from the Greek *kardia (heart)*, in reference to its shape, and was given to it by Johan Castillon (Wikipedia).
它的名字来自希腊语 *kardia (heart)*，参考了它的形状，由 Johan Castillon(维基百科) 给它命名的。



```
\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}
```

24.3. Drawing semicircle 绘制半圆

\tkzDrawSemiCircle[⟨local options⟩](⟨O,A⟩)

arguments	example	explanation
(⟨pt1,pt2⟩)	(⟨O,A⟩)	OA= radius

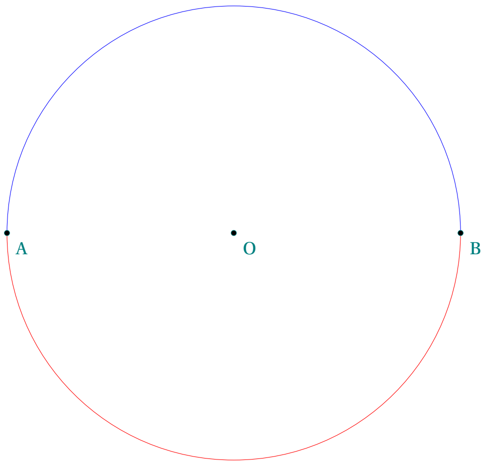
O center A extremity of the semicircle

\tkzDrawSemiCircle[⟨local options⟩](⟨O,A⟩)

参数	示例	含义
(⟨pt1,pt2⟩)	(⟨O,A⟩)	OA= radius

O center A extremity of the semicircle 点 O 为圆心，点 A 半圆的外端

24.3.1. Use of \tkzDrawSemiCircle



```
\begin{tikzpicture}
   \tkzDefPoint(0,0){A} \tkzDefPoint(6,0){B}
   \tkzDefMidPoint(A,B)  \tkzGetPoint{O}
   \tkzDrawSemiCircle[blue](O,B)
   \tkzDrawSemiCircle[red](O,A)
   \tkzDrawPoints(O,A,B)
   \tkzLabelPoints[below right](O,A,B)
 \end{tikzpicture}
```

24.4. Drawing semicircles 绘制多个半圆

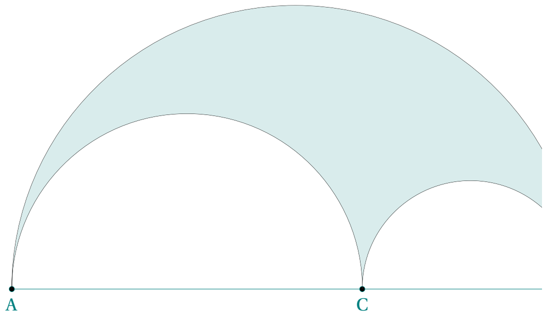
\tkzDrawSemiCircles[⟨local options⟩](⟨A,B C,D …⟩)

arguments	example	explanation
(⟨pt1,pt2 pt3,pt4 ...⟩)	(⟨A,B C,D⟩)	List of two points

\tkzDrawSemiCircles[⟨local options⟩](⟨A,B C,D …⟩)

参数	示例	含义
(⟨pt1,pt2 pt3,pt4 ...⟩)	(⟨A,B C,D⟩)	两点的序列

24.4.1. Use of \tkzDrawSemiCircles : Golden arbelos



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,10/0/B}
\tkzDefGoldenRatio(A,B) \tkzGetPoint{C}
\tkzDefMidPoint(A,B)                   \tkzGetPoint{O_0}
\tkzDefMidPoint(A,C)                   \tkzGetPoint{O_1}
\tkzDefMidPoint(C,B)                   \tkzGetPoint{O_2}
\tkzLabelPoints(A,B,C)
\tkzDrawSegment(A,B)
\tkzDrawPoints(A,B,C)
\begin{scope}[local bounding box = graph]
  \tkzDrawSemiCircles[color=black](O_0,B)
\end{scope}
\useasboundingbox (graph.south west) rectangle (graph.north east);
\tkzClipCircle[out](O_1,C)\tkzClipCircle[out](O_2,B)
\tkzDrawSemiCircles[draw=none,fill=teal!15](O_0,B)
\tkzDrawSemiCircles[color=black](O_1,C O_2,B)
\end{tikzpicture}
```

25. Drawing arcs 绘制圆弧

25.1. Macro: \tkzDrawArc

\tkzDrawArc[⟨local options⟩](⟨O,…⟩)(⟨…⟩)

This macro traces the arc of center O. Depending on the options, the arguments differ. It is a question of determining a starting point and an end point. Either the starting point is given, which is the simplest, or the radius of the arc is given. In the latter case, it is necessary to have two angles. Either the angles can be given directly, or nodes associated with the center can be given to determine them. The angles are in degrees.

options	default	definition
towards	towards	O is the center and the arc from A to (OB)
rotate	towards	the arc starts from A and the angle determines its length
R	towards	We give the radius and two angles
R with nodes	towards	We give the radius and two points
angles	towards	We give the radius and two points
delta	0	angle added on each side
reverse	false	inversion of the arc's path, interesting to inverse arrow

Of course, you have to add all the styles of Ti*k*Z for the tracings...

options	arguments	example
towards	(⟨pt,pt⟩)(⟨pt⟩)	\tkzDrawArc[delta=10](O,A)(B)
rotate	(⟨pt,pt⟩)(⟨an⟩)	\tkzDrawArc[rotate,color=red](O,A)(90)
R	(⟨pt,r⟩)(⟨an,an⟩)	\tkzDrawArc[R](O,2)(30,90)
R with nodes	(⟨pt,r⟩)(⟨pt,pt⟩)	\tkzDrawArc[R with nodes](O,2)(A,B)
angles	(⟨pt,pt⟩)(⟨an,an⟩)	\tkzDrawArc[angles](O,A)(0,90)

25.2. Macro: \tkzDrawArc

```
\tkzDrawArc[⟨local options⟩](⟨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。问题在于确定起点和终点。给了起点，这是最简单的，或者给圆弧半径。后一种情况，需要有两个角度。角度可以直接给出，也可以给出与中心相关联的节点来确定它们。角度以度为单位。

选项	默认值	含义
towards	towards	O 是圆心，并且圆弧从 A 到 (OB)
rotate	towards	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
reverse	false	圆弧路径的反转，有趣的是反转箭头

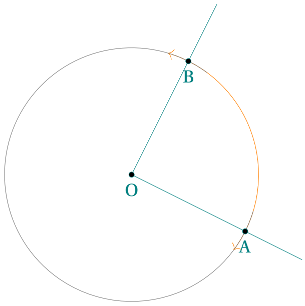
可以使用所有有效的 Ti*k*Z 样式。

options	arguments	example
towards	(⟨pt,pt⟩)(⟨pt⟩)	\tkzDrawArc[delta=10](O,A)(B)
rotate	(⟨pt,pt⟩)(⟨an⟩)	\tkzDrawArc[rotate,color=red](O,A)(90)
R	(⟨pt,r⟩)(⟨an,an⟩)	\tkzDrawArc[R](O,2)(30,90)
R with nodes	(⟨pt,r⟩)(⟨pt,pt⟩)	\tkzDrawArc[R with nodes](O,2)(A,B)
angles	(⟨pt,pt⟩)(⟨an,an⟩)	\tkzDrawArc[angles](O,A)(0,90)

Here are a few examples:
下面是几个例子：

25.2.1. Option towards

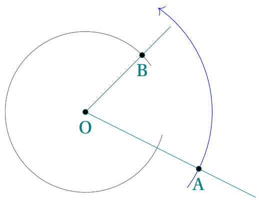
It's useless to put **towards**. In this first example the arc starts from A and goes to B. The arc going from B to A is different. The salient is obtained by going in the direct direction of the trigonometric circle.
towards 是默认选项。在第一个示例中，弧从点 A 开始，到点 B 结束。与从点 B 开始到点 A 结束的弧线是不同的。圆弧凸向由逆时针方向确定。



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPointBy[rotation= center O angle 90](A)
  \tkzGetPoint{B}
  \tkzDrawArc[color=orange,<->](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}
```

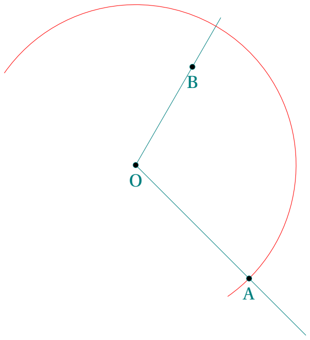
25.2.2. Option towards

In this one, the arc starts from A but stops on the right (OB).
该例中，圆弧从 A 开始，直到 (OB) 结束。



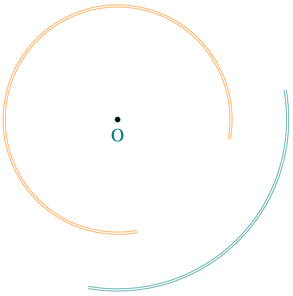
```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzDrawArc[color=blue,->](O,A)(B)
  \tkzDrawArc[color=gray](O,B)(A)
  \tkzDrawArc(O,B)(A)
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

25.2.3. Option rotate



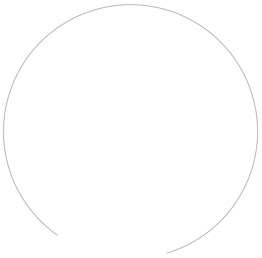
```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-2){A}
  \tkzDefPoint(60:2){B}
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawArc[rotate,color=red](O,A)(180)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

25.2.4. Option R



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoints{0/0/O}
  \tkzSetUpCompass[<->]
  \tkzDrawArc[R,color=teal,double](O,3)(270,360)
  \tkzDrawArc[R,color=orange,double](O,2)(0,270)
  \tkzDrawPoint(O)
  \tkzLabelPoint[below](O){$O$}
\end{tikzpicture}
```

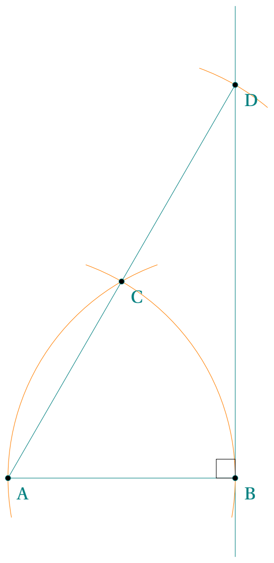
25.2.5. Option R with nodes



```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzCalcLength(B,A)\tkzGetLength{radius}
  \tkzDrawArc[R with nodes](B,\radius)(A,O)
\end{tikzpicture}
```

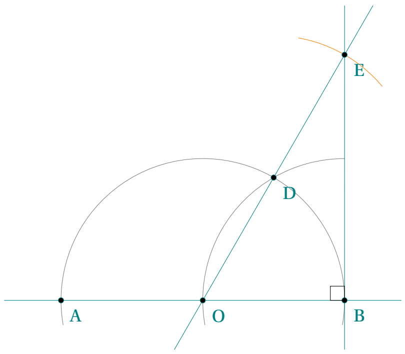
25.2.6. Option delta

This option allows a bit like `\tkzCompass` to place an arc and overflow on either side. delta is a measure in degrees. 该选项与`\tkzCompass` 结果类似，它能够延伸圆弧，`delta` 的单位是度。



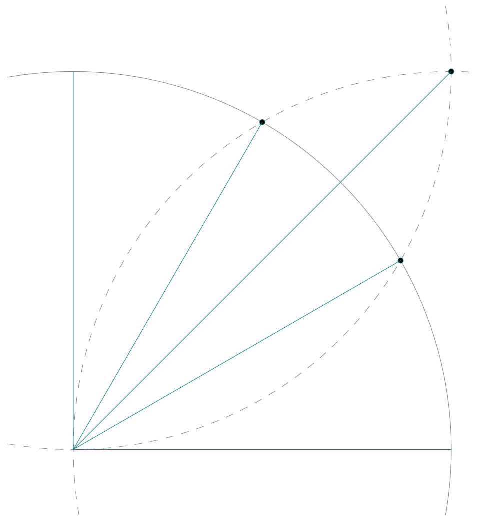
```
\begin{tikzpicture}
 \tkzDefPoint(0,0){A}
 \tkzDefPoint(3,0){B}
 \tkzDefPointBy[rotation= center A angle 60](B)
 \tkzGetPoint{C}
 \begin{scope}% style only local
   \tkzDefPointBy[symmetry= center C](A)
   \tkzGetPoint{D}
   \tkzDrawSegments(A,B A,D)
   \tkzDrawLine(B,D)
   \tkzSetUpCompass[color=orange]
   \tkzDrawArc[orange,delta=10](A,B)(C)
   \tkzDrawArc[orange,delta=10](B,C)(A)
   \tkzDrawArc[orange,delta=10](C,D)(D)
 \end{scope}

 \tkzDrawPoints(A,B,C,D)
 \tkzLabelPoints[below right](A,B,C,D)
 \tkzMarkRightAngle(D,B,A)
\end{tikzpicture}
```

25.2.7. Option angles: example 1



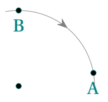
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPoint(2.5,0){O}
  \tkzDefPointBy[rotation=center O angle 60](B)
  \tkzGetPoint{D}
  \tkzDefPointBy[symmetry=center D](O)
  \tkzGetPoint{E}
  \begin{scope}
    \tkzDrawArc[angles](O,B)(0,180)
    \tkzDrawArc[angles,](B,O)(100,180)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

25.2.8. Option angles: example 2



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

25.2.9. Option reverse: inversion of the arrow



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/O,3/0/U}
  \tkzDefPoint(10:1){A}
  \tkzDefPoint(90:1){B}
  \tkzLabelPoints(A,B)
  \tkzDrawArc[reverse,tkz arrow={Stealth}](O,A)(B)
  \tkzDrawPoints(A,B,O)
\end{tikzpicture}
```

26. Drawing a sector or sectors 绘制扇形

26.1. \tkzDrawSector

Attention the arguments vary according to the options.

\tkzDrawSector[⟨local options⟩](⟨O,…⟩)(⟨…⟩)

options	default	definition
towards	towards	O is the center and the arc from A to (OB)
rotate	towards	the arc starts from A and the angle determines its length
R	towards	We give the radius and two angles
R with nodes	towards	We give the radius and two points

You have to add, of course, all the styles of TikZ for tracings...

options	arguments	example
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=teal](O,2)(30,90)
R with nodes	(⟨pt,r⟩)(⟨pt,pt⟩)	\tkzDrawSector[R with nodes](O,2)(A,B)

26.2. \tkzDrawSector

Attention the arguments vary according to the options.
注意: 根据参数的不同，选项也不同。

`\tkzDrawSector[⟨local options⟩](⟨O,...⟩)(⟨...⟩)`

选项	默认值	含义
towards	towards	O 是圆心 ，弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始，角度决定其长度
R	towards	给出半径和两个角度
R with nodes	towards	给出半径和两个点

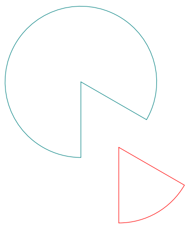
当然，必须添加所有的 TikZ 样式...

选项	默认值	含义
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=teal](O,2)(30,90)
R with nodes	(⟨pt,r⟩)(⟨pt,pt⟩)	\tkzDrawSector[R with nodes](O,2)(A,B)

Here are a few examples:
下面是一些示例:

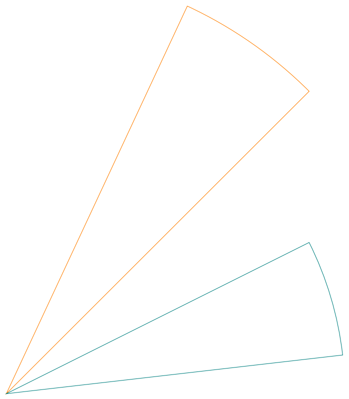
26.2.1. \tkzDrawSector and towards

There's no need to put **towards**. You can use **fill** as an option.
没有必要用 **towards**。可以使用 **fill** 作为选项。



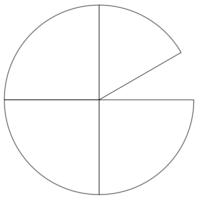
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(-30:1){A}
  \tkzDefPointBy[rotation = center O angle -60](A)
  \tkzDrawSector[teal](O,A)(tkzPointResult)
 \begin{scope}[shift={(-60:1)}]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(-30:1){A}
  \tkzDefPointBy[rotation = center O angle -60](A)
  \tkzDrawSector[red](O,tkzPointResult)(A)
  \end{scope}
\end{tikzpicture}
```

26.2.2. \tkzDrawSector and rotate



```
\begin{tikzpicture}[scale=2]
 \tkzDefPoints{0/0/O,2/2/A,2/1/B}
 \tkzDrawSector[rotate,orange](O,A)(20)
 \tkzDrawSector[rotate,teal](O,B)(-20)
\end{tikzpicture}
```

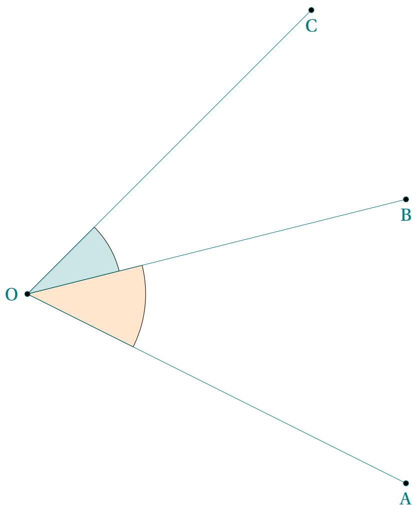
26.2.3. \tkzDrawSector and R



```
\begin{tikzpicture}[scale=1.25]
 \tkzDefPoint(0,0){O}
 \tkzDefPoint(2,-1){A}
 \tkzDrawSector[R](O,1)(30,90)
 \tkzDrawSector[R](O,1)(90,180)
 \tkzDrawSector[R](O,1)(180,270)
 \tkzDrawSector[R](O,1)(270,360)
\end{tikzpicture}
```

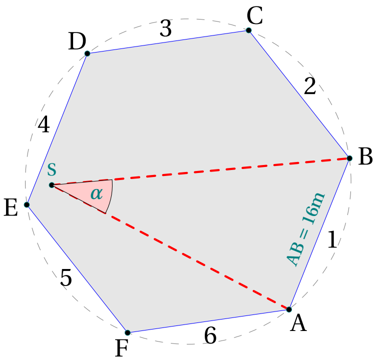
26.2.4. \tkzDrawSector and R with nodes

In this example I use the option `fill` but `\tkzFillSector` is possible.
在这个例子中，使用选项 `fill`，但 `\tkzFillSector` 也可以的。



```
\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=teal!20](O,1)(B,C)
 \tkzDrawSector[R with nodes,%
                fill=orange!20](O,1.25)(A,B)
\tkzDrawSegments(O,A O,B O,C)
\tkzDrawPoints(O,A,B,C)
\tkzLabelPoints(A,B,C)
\tkzLabelPoints[left](O)
\end{tikzpicture}
```

26.2.5. \tkzDrawSector and R with nodes



```
\begin{tikzpicture} [scale=.4]
 \tkzDefPoints{-1/-2/A,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){\(A B = 16m\)}
 \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){$S$}
  \tkzDrawSector[R with nodes,fill=red!20](S,2)(A,B)
  \tkzLabelAngle[pos=1.5](A,S,B){$\alpha$}
\end{tikzpicture}
```

26.3. Coloring a disc 给圆着色

This was possible with the macro \tkzDrawCircle, but disk tracing was mandatory, this is no longer the case.
这在使用宏 \tkzDrawCircle 时是可能的，但磁盘跟踪是强制性的，现在不再是这种情况。
【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。】【耿楠的译文】

\tkzFillCircle[⟨local options⟩](⟨A,B⟩)

options	default	definition
radius	radius	two points define a radius
R	radius	a point and the measurement of a radius

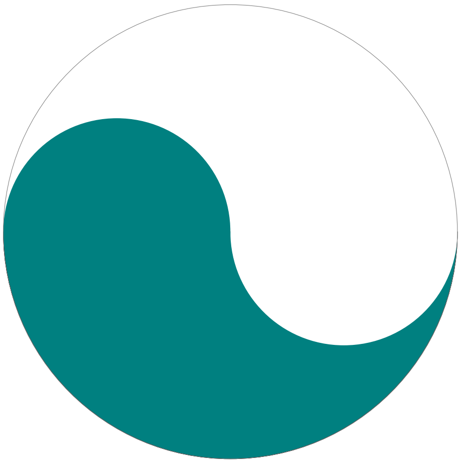
You don't need to put **radius** because that's the default option. Of course, you have to add all the styles of Ti*k*Z for the plots.

\tkzFillCircle[⟨ 命令选项 ⟩](⟨A,B⟩)

选项	默认值	含义
radius	radius	两个点定义半径
R	radius	一个点和半径的长度

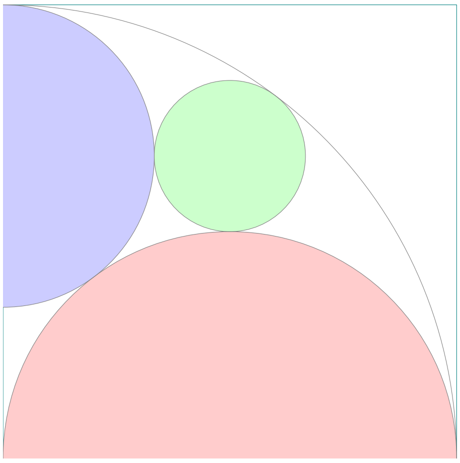
不需要输入 **radius**，因为这是默认选项。可以使用所有有效 Ti*k*Z 样式。

26.3.1. Yin and Yang 阴阳-太极



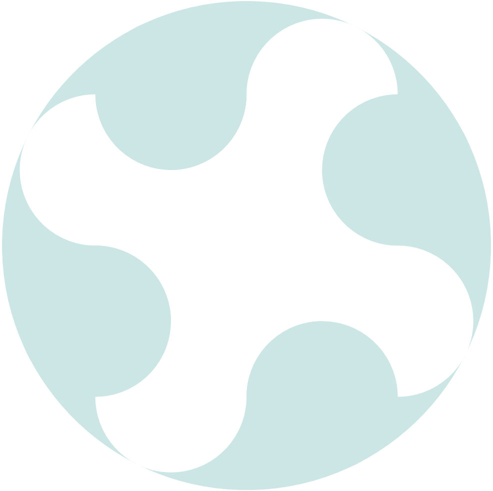
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(-4,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(-2,0){I}
  \tkzDefPoint(2,0){J}
  \tkzDrawSector[fill=teal](O,A)(B)
  \tkzFillCircle[fill=white](J,B)
  \tkzFillCircle[fill=teal](I,A)
  \tkzDrawCircle(O,A)
\end{tikzpicture}
```

26.3.2. From a sangaku



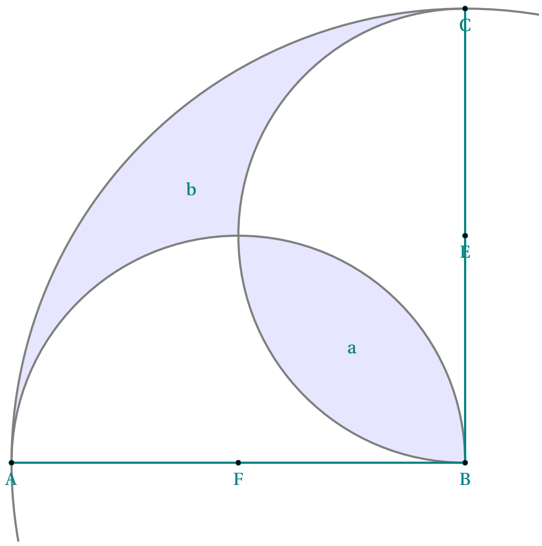
```
\begin{tikzpicture}
   \tkzDefPoint(0,0){B}  \tkzDefPoint(6,0){C}%
   \tkzDefSquare(B,C)    \tkzGetPoints{D}{A}
   \tkzClipPolygon(B,C,D,A)
   \tkzDefMidPoint(A,D)  \tkzGetPoint{F}
   \tkzDefMidPoint(B,C)  \tkzGetPoint{E}
   \tkzDefMidPoint(B,D)  \tkzGetPoint{Q}
   \tkzDefLine[tangent from = B](F,A) \tkzGetPoints{H}{G}
   \tkzInterLL(F,G)(C,D) \tkzGetPoint{J}
   \tkzInterLL(A,J)(F,E) \tkzGetPoint{K}
   \tkzDefPointBy[projection=onto B--A](K)
   \tkzGetPoint{M}
   \tkzDrawPolygon(A,B,C,D)
   \tkzFillCircle[red!20](E,B)
   \tkzFillCircle[blue!20](M,A)
   \tkzFillCircle[green!20](K,Q)
  \tkzDrawCircles(B,A M,A E,B K,Q)
\end{tikzpicture}
```

26.3.3. Clipping and filling part I 裁剪和填充第一部分



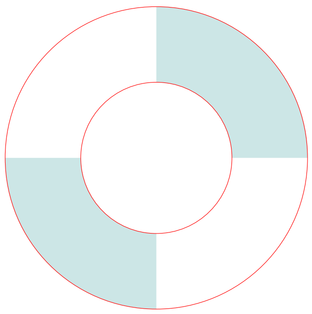
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/0/B,2/2/O,3/4/X,4/1/Y,1/0/Z,
              0/3/W,3/0/R,4/3/S,1/4/T,0/1/U}
\tkzDefSquare(A,B)\tkzGetPoints{C}{D}
\tkzDefPointWith[colinear normed=at X,K=1](O,X)
 \tkzGetPoint{F}
\begin{scope}
  \tkzFillCircle[fill=teal!20](O,F)
  \tkzFillPolygon[white](A,...,D)
  \tkzClipPolygon(A,...,D)
  \foreach \c/\t in {S/C,R/B,U/A,T/D}
  {\tkzFillCircle[teal!20](\c,\t)}
\end{scope}
\foreach \c/\t in {X/C,Y/B,Z/A,W/D}
{\tkzFillCircle[white](\c,\t)}
  \foreach \c/\t in {S/C,R/B,U/A,T/D}
  {\tkzFillCircle[teal!20](\c,\t)}
\end{tikzpicture}
```

26.3.4. Clipping and filling part II 裁剪和填充第二部分



```
\begin{tikzpicture}[scale=.75]
\tkzDefPoints{0/0/A,8/0/B,8/8/C,0/8/D}
\tkzDefMidPoint(A,B) \tkzGetPoint{F}
\tkzDefMidPoint(B,C) \tkzGetPoint{E}
\tkzDefMidPoint(D,B) \tkzGetPoint{I}
\tkzDefMidPoint(I,B) \tkzGetPoint{a}
\tkzInterLC(B,I)(B,C) \tkzGetSecondPoint{K}
\tkzDefMidPoint(I,K) \tkzGetPoint{b}
\begin{scope}
  \tkzFillSector[fill=blue!10](B,C)(A)
  \tkzDefMidPoint(A,B) \tkzGetPoint{x}
  \tkzDrawSemiCircle[fill=white](x,B)
  \tkzDefMidPoint(B,C) \tkzGetPoint{y}
  \tkzDrawSemiCircle[fill=white](y,C)
  \tkzClipCircle(E,B)
  \tkzClipCircle(F,B)
  \tkzFillCircle[fill=blue!10](B,A)
\end{scope}
\tkzDrawSemiCircle[thick](F,B)
\tkzDrawSemiCircle[thick](E,C)
\tkzDrawArc[thick](B,C)(A)
\tkzDrawSegments[thick](A,B B,C)
\tkzDrawPoints(A,B,C,E,F)
\tkzLabelPoints[centered](a,b)
\tkzLabelPoints(A,B,C,E,F)
\end{tikzpicture}
```

26.3.5. Clipping and filling part III 裁剪和填充第三部分



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A} \tkzDefPoint(1,0){B}
  \tkzDefPoint(2,0){C} \tkzDefPoint(-3,0){a}
  \tkzDefPoint(3,0){b}  \tkzDefPoint(0,3){c}
  \tkzDefPoint(0,-3){d}
\begin{scope}
 \tkzClipPolygon(a,b,c,d)
 \tkzFillCircle[teal!20](A,C)
\end{scope}
 \tkzFillCircle[white](A,B)
 \tkzDrawCircle[color=red](A,C)
 \tkzDrawCircle[color=red](A,B)
\end{tikzpicture}
```

26.4. Coloring a polygon 多边形着色

`\tkzFillPolygon[⟨local options⟩](⟨points list⟩)`

You can color by drawing the polygon, but in this case you color the inside of the polygon without drawing it.

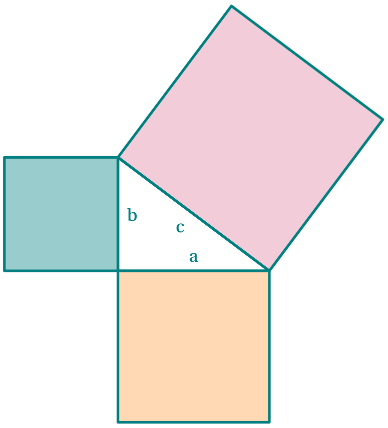
arguments	example	explanation
(⟨pt1,pt2,…⟩)	(⟨A,B,…⟩)	

\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
(⟨pt1,pt2,…⟩)	(⟨A,B,…⟩)	

26.4.1. \tkzFillPolygon 填充多边形



```
\begin{tikzpicture}[scale=.5]
   \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[color  =  orange!30   ](A,C,G,H)
   \tkzFillPolygon[color  =  teal!40  ](C,B,I,J)
   \tkzFillPolygon[color  =  purple!20](B,A,E,F)
   \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[above](C,A){$a$}
   \tkzLabelSegment[right](B,C){$b$}
   \tkzLabelSegment[below left](B,A){$c$}
\end{tikzpicture}
```

26.5. \tkzFillSector 填充扇形

Attention the arguments vary according to the options.
注意参数需要根据选项变化。

\tkzFillSector[⟨local options⟩](⟨O,…⟩)(⟨…⟩)

options	default	definition
towards	towards	O is the center and the arc from A to (OB)
rotate	towards	the arc starts from A and the angle determines its length
R	towards	We give the radius and two angles
R with nodes	towards	We give the radius and two points

Of course, you have to add all the styles of Ti*k*Z for the tracings...

options	arguments	example
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)(30,90)
R with nodes	(⟨pt,r⟩)(⟨pt,pt⟩)	\tkzFillSector[R with nodes](O,2)(A,B)

`\tkzFillSector[⟨命令选项⟩](⟨O,...⟩)(⟨...⟩)`

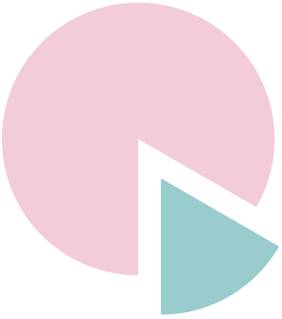
选项	默认值	含义
towards	towards	O 是圆心并且圆弧从 A 到 (OB)
rotate	towards	圆弧从 A 开始并且通过角度确定长度
R	towards	给定半径和两个角度
R with nodes	towards	给定半径和两个点

当然，可以使用所有有效的 Ti*k*Z 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)`

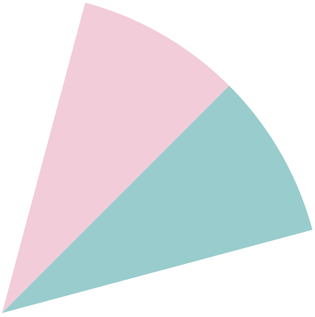
26.5.1. \tkzFillSector and towards

It is useless to put **towards** and you will notice that the contours are not drawn, only the surface is colored.
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=purple!20](O,A)(tkzPointResult)
  \begin{scope}[shift={(-60:1)}]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(-30:3){A}
  \tkzDefPointBy[rotation = center O angle -60](A)
  \tkzGetPoint{A'}
  \tkzFillSector[color=teal!40](O,A')(A)
    \end{scope}
\end{tikzpicture}
```

26.5.2. \tkzFillSector and rotate



```
\begin{tikzpicture}[scale=1.5]
 \tkzDefPoint(0,0){O} \tkzDefPoint(2,2){A}
 \tkzFillSector[rotate,color=purple!20](O,A)(30)
 \tkzFillSector[rotate,color=teal!40](O,A)(-30)
\end{tikzpicture}
```

26.6. Colour an angle: \tkzFillAngle 对角着色

The simplest operation
最简单的操作

```
\tkzFillAngle[⟨local options⟩](⟨A,O,B⟩)
```

O is the vertex of the angle. OA and OB are the sides. Attention the angle is determined by the order of the points.

options	default	definition
size	1	this option determines the radius of the coloured angular sector.

Of course, you have to add all the styles of Ti*k*Z, like the use of fill and shade...

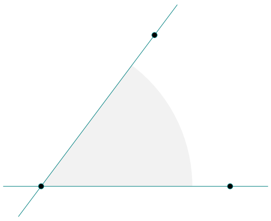
```
\tkzFillAngle[⟨ 命令选项 ⟩](⟨A,O,B⟩)
```

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

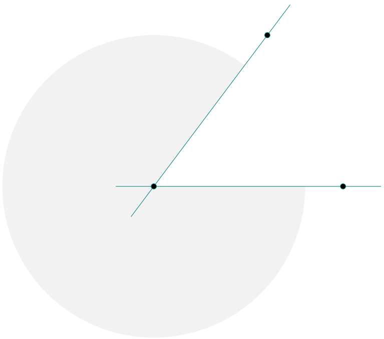
可以使用所有有效的 Ti*k*Z 样式，如 fill 和 shade 等。

26.6.1. Example with size

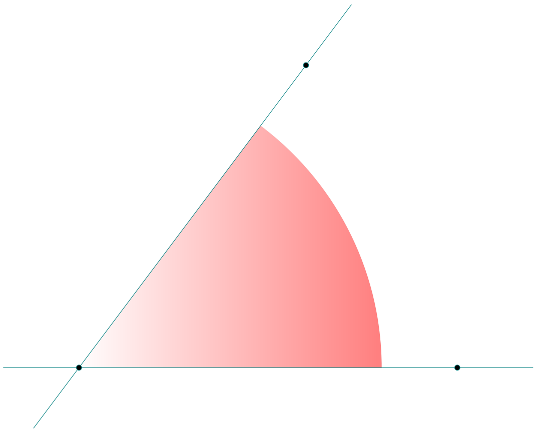


```
\begin{tikzpicture}
   \tkzInit
   \tkzDefPoints{0/0/O,2.5/0/A,1.5/2/B}
   \tkzFillAngle[size=2, fill=gray!10](A,O,B)
   \tkzDrawLines(O,A O,B)
   \tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

26.6.2. Changing the order of items 改变点的顺序



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



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

26.6.3. Multiples angles

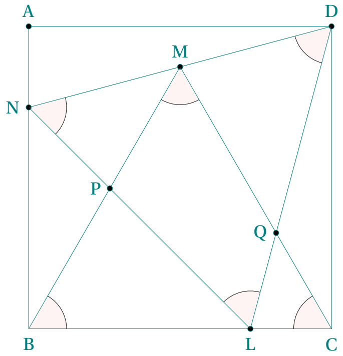
\tkzFillAngles[⟨local options⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.

With common options, there is a macro for multiple angles.

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

绘制多个角。

【原文上面的表格所处位置似乎不对，现调整到这里。】



```
\begin{tikzpicture}[scale=0.5]
  \tkzDefPoints{0/0/B,8/0/C,0/8/A,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}
```

27. Controlling Bounding Box ?????

From the **PgfManual** :"When you add the clip option, the current path is used for clipping subsequent drawings. Clipping never enlarges the clipping area. Thus, when you clip against a certain path and then clip again against another path, you clip against the intersection of both. The only way to enlarge the clipping path is to end the pgfscope in which the clipping was done. At the end of a pgfscope the clipping path that was in force at the beginning of the scope is reinstalled."
PgfManual:"?? ??????????????????? pgfscope?? pgfscope
???????????????????????"
First of all, you don't have to deal with Ti*k*Z the size of the bounding box. Early versions of tkz-euclide did not control the size of the bounding box, now with tkz-euclide 4 the size of the bounding box is limited.
??????? Ti*k*Z ???????tkz-euclide?????????????????tkz-euclide 4 ??????????
The initial bounding box after using the macro \tkzInit is defined by the rectangle based on the points $(0,0)$ and $(10,10)$. The \tkzInit macro allows this initial bounding box to be modified using the arguments (**xmin**, **xmax**, **ymin**, and **ymax**). Of course any external trace modifies the bounding box. Ti*k*Z maintains that bounding box. It is possible to influence this behavior either directly with commands or options in Ti*k*Z such as a command like \useasboundingbox or the option **use as bounding box**. A possible consequence is to reserve a box for a figure but the figure may overflow the box and spread over the main text. The following command \pgfresetboundingbox clears a bounding box and establishes a new one.
???\tkzInit???????????$(0,0)$?$(10,10)$??????\tkzInit???????(**xmin** ?**xmax** ?**ymin**? **ymax**)????????????????????????? Ti*k*Z
????????????? Ti*k*Z ??????????????????\useasboundingbox??? **use as bounding box**??
????\pgfresetboundingbox?????????????????

27.1. Utility of \tkzInit ????

However, it is sometimes necessary to control the size of what will be displayed. To do this, you need to have prepared the bounding box you are going to work in, this is the role of the macro \tkzInit. For some drawings, it is interesting to fix the extreme values (xmin,xmax,ymin and ymax) and to "clip" the definition rectangle in order to control the size of the figure as well as possible.
????????????????????? ????????????????????\tkzInit???????????????(xmin?xmax?ymin?ymax)??????????????????????????
The two macros that are useful for controlling the bounding box:
????????????:

- \tkzInit
- \tkzClip

To this, I added macros directly linked to the bounding box. You can now view it, backup it, restore it (see the section Bounding Box).
???????????????????????????????????(???????)?

27.2. \tkzInit

\tkzInit[⟨local options⟩]

options	default	definition
xmin	0	minimum value of the abscissae in cm
xmax	10	maximum value of the abscissae in cm
xstep	1	difference between two graduations in x
ymin	0	minimum y-axis value in cm
ymax	10	maximum y-axis value in cm
ystep	1	difference between two graduations in y

The role of \tkzInit is to define a orthogonal coordinates system and a rectangular part of the plane in which you will place your drawings using Cartesian coordinates. This macro allows you to define your working environment as with a calculator. With **tkz-euclide** 4 `\xstep` and `\ystep` are always 1. Logically it is no longer useful to use `\tkzInit`, except for an action like "Clipping Out".

`\tkzInit[⟨local options⟩]`

??	???	??
xmin	0	?????????????
xmax	10	?????????????
xstep	1	x???????
ymin	0	y???????????
ymax	10	y???????????
ystep	1	?y??????????

`\tkzInit`??????? orthogonal??????????????????????????????? ????????????????????? **tkznameofpack** 4 ???????? `\xstep`? `\ystep`???1????????? `\tkzInit`?????????? "Clipping Out"??????

27.3. \tkzClip

`\tkzClip[⟨local options⟩]`

The role of this macro is to make invisible what is outside the rectangle defined by (xmin ; ymin) and (xmax ; ymax).

options	default	definition
space	1	added value on the right, left, bottom and top of the background

The role of the **space** option is to enlarge the visible part of the drawing. This part becomes the rectangle defined by (xmin-space ; ymin-space) and (xmax+space ; ymax+space). **space** can be negative! The unit is cm and should not be specified.

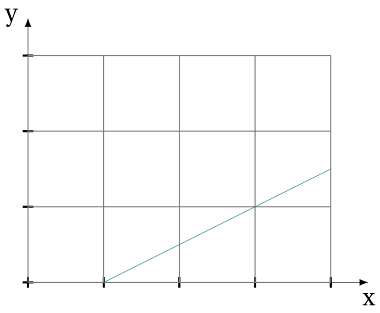
`\tkzClip[⟨local options⟩]`

????????(xmin ; ymin)?(xmax ; ymax)?????????????.

??	???	??
space	1	???????????????????

space ????????????????????????xmin-space ; ymin-space???xmax+space ; ymax+space??????? **space**?????! ???cm?

The role of this macro is to "clip" the initial rectangle so that only the paths contained in this rectangle are drawn. ??????? "??"??????????????????????



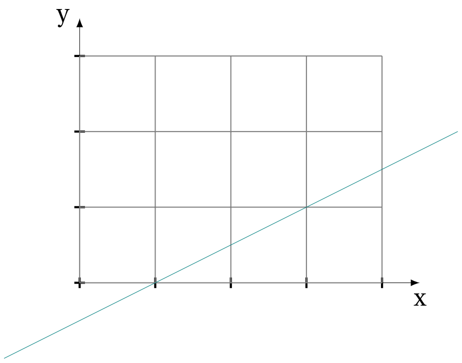
```
\begin{tikzpicture}
 \tkzInit[xmax=4, ymax=3]
 \tkzDefPoints{-1/-1/A,5/2/B}
 \tkzDrawX \tkzDrawY
 \tkzGrid
 \tkzClip
 \tkzDrawSegment(A,B)
\end{tikzpicture}
```

It is possible to add a bit of space
????????

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

27.4. `\tkzClip` and the option `space`

This option allows you to add some space around the ”clipped” rectangle.
???????? ”??”????????????



```
\begin{tikzpicture}
 \tkzInit[xmax=4, ymax=3]
 \tkzDefPoints{-1/-1/A,5/2/B}
 \tkzDrawX \tkzDrawY
 \tkzGrid
 \tkzClip[space=1]
 \tkzDrawSegment(A,B)
\end{tikzpicture}
```

The dimensions of the ”clipped” rectangle are `xmin-1`, `ymin-1`, `xmax+1` and `ymax+1`.
? ”?? ”??????? `xmin-1`, `ymin-1`, `xmax+1` ? `ymax+1`?

27.5. `tkzShowBB`

The simplest macro.
??????

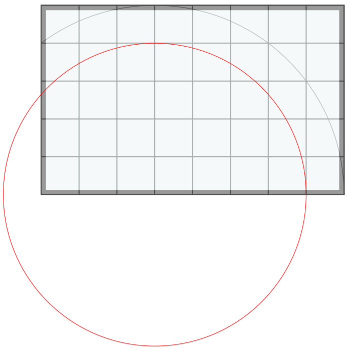
`\tkzShowBB[⟨local options⟩]`

This macro displays the bounding box. A rectangular frame surrounds the bounding box. This macro accepts Ti*k*Z options.

`\tkzShowBB[⟨local options⟩]`

??????????????????????????? Ti*k*Z ???

27.5.1. Example with \tkzShowBB



```
\begin{tikzpicture}[scale=.5]
  \tkzInit[ymax=5,xmax=8]
  \tkzGrid
  \tkzDefPoint(3,0){A}
   \begin{scope}
    \tkzClipBB
    \tkzDefCircle[R](A,5) \tkzGetPoint{a}
    \tkzDrawCircle(A,a)
     \tkzShowBB[line width = 4pt,fill=teal!10,opacity=.4]
   \end{scope}
\tkzDefCircle[R](A,4) \tkzGetPoint{b}
\tkzDrawCircle[red](A,b)
\end{tikzpicture}
```

27.6. tkzClipBB

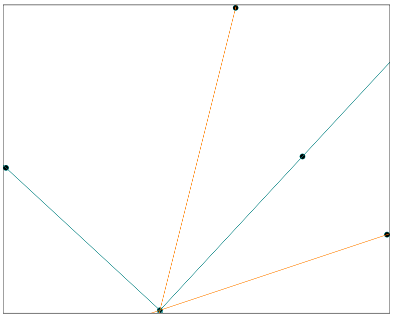
`\tkzClipBB`

The idea is to limit future constructions to the current bounding box.

`\tkzClipBB`

????????????????????????

27.6.1. Example with \tkzClipBB and the bisectors



```
\begin{tikzpicture}
\tkzInit[xmin=-3,xmax=6, ymin=-1,ymax=6]
\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}
\tkzDrawPoints(O,I,J,i,j)
\tkzClipBB
\tkzDrawLines[add = 1 and 2,color=orange](O,I O,J)
\tkzDrawLines[add = 1 and 2](O,i O,j)
\tkzShowBB
\end{tikzpicture}
```

28. Clipping different objects ???????

28.1. Clipping a polygon ???????

```
\tkzClipPolygon[⟨local options⟩](⟨points list⟩)
```

This macro makes it possible to contain the different plots in the designated polygon.

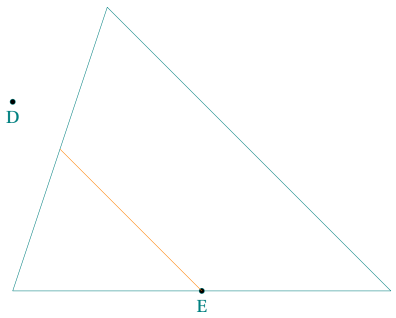
arguments	example	explanation
(⟨pt1,pt2,pt3,…⟩)	(⟨A,B,C⟩)	
options	default	definition
out		allows to clip the outside of the object

```
\tkzClipPolygon[⟨local options⟩](⟨points list⟩)
```

????????????????????????

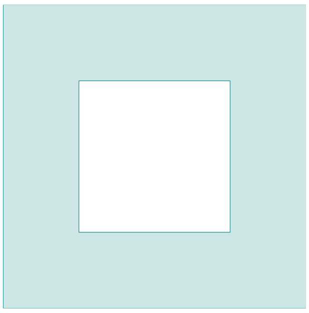
??	??	??
(⟨pt1,pt2,pt3,…⟩)	(⟨A,B,C⟩)	
??	???	??
out		?????????

28.1.1. \tkzClipPolygon



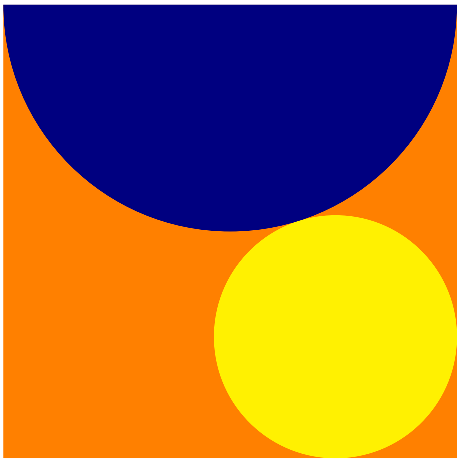
```
  \begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(1,3){C}
  \tkzDrawPolygon(A,B,C)
  \tkzDefPoint(0,2){D}
  \tkzDefPoint(2,0){E}
  \tkzDrawPoints(D,E)
  \tkzLabelPoints(D,E)
  \tkzClipPolygon(A,B,C)
  \tkzDrawLine[new](D,E)
\end{tikzpicture}
```

28.1.2. \tkzClipPolygon[out]



```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,0){P1}
\tkzDefPoint(4,0){P2}
\tkzDefPoint(4,4){P3}
\tkzDefPoint(0,4){P4}
\tkzDefPoint(1,1){Q1}
\tkzDefPoint(3,1){Q2}
\tkzDefPoint(3,3){Q3}
\tkzDefPoint(1,3){Q4}
\tkzDrawPolygon(P1,P2,P3,P4)
\begin{scope}
   \tkzClipPolygon[out](Q1,Q2,Q3,Q4)
  \tkzFillPolygon[teal!20](P1,P2,P3,P4)
\end{scope}
\tkzDrawPolygon(Q1,Q2,Q3,Q4)
\end{tikzpicture}
```

28.1.3. Example: use of "Clip" for Sangaku in a square



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
 \tkzDefSquare(A,B)   \tkzGetPoints{C}{D}
 \tkzDefPoint(4,8){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}
 \tkzClipPolygon(B,C,D,A)
 \tkzFillPolygon[color = orange](A,B,C,D)
 \tkzFillCircle[color = yellow](M,I)
 \tkzFillCircle[color = blue!50!black](F,D)
\end{tikzpicture}
```

28.2. Clipping a disc ???

\tkzClipCircle[⟨local options⟩](⟨A,B⟩)

arguments	example	explanation
(⟨A,B⟩)	(⟨A,B⟩)	AB radius

options	default	definition
out		allows to clip the outside of the object

It is not necessary to put **radius** because that is the default option.

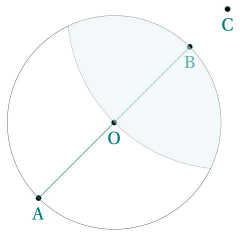
\tkzClipCircle[⟨local options⟩](⟨A,B⟩)

??	??	??
(⟨A,B⟩)	(⟨A,B⟩)	AB ??

??	???	??
out		?????????

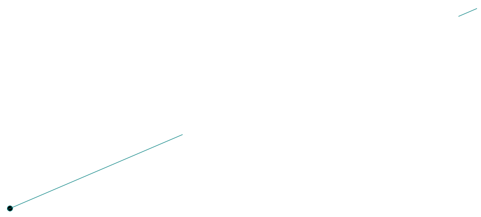
?????? **radius**??????????

28.2.1. Simple clip ????



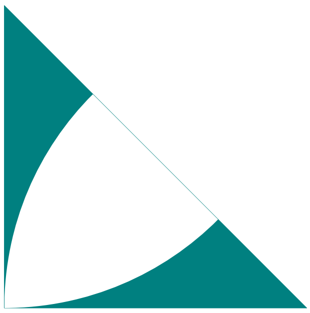
```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoint(0,0){A} \tkzDefPoint(2,2){O}
  \tkzDefPoint(4,4){B} \tkzDefPoint(5,5){C}
  \tkzDrawPoints(O,A,B,C)
  \tkzLabelPoints(O,A,B,C)
  \tkzDrawCircle(O,A)
  \tkzClipCircle(O,A)
  \tkzDrawLine(A,C)
  \tkzDrawCircle[fill=teal!10,opacity=.5](C,O)
\end{tikzpicture}
```

28.3. Clip out



```
\begin{tikzpicture}
  \tkzInit[xmin=-3,ymin=-2,xmax=4,ymax=3]
   \tkzDefPoint(0,0){O}
   \tkzDefPoint(-4,-2){A}
   \tkzDefPoint(3,1){B}
   \tkzDefCircle[R](O,2) \tkzGetPoint{o}
   \tkzDrawPoints(A,B) % to have a good bounding box
   \begin{scope}
    \tkzClipCircle[out](O,o)
    \tkzDrawLines(A,B)
   \end{scope}
\end{tikzpicture}
```

28.4. Intersection of disks ?????



```
\begin{tikzpicture}
\tkzDefPoints{0/0/O,4/0/A,0/4/B}
\tkzDrawPolygon[fill=teal](O,A,B)
\tkzClipPolygon(O,A,B)
\tkzClipCircle(A,O)
\tkzClipCircle(B,O)
\tkzFillPolygon[white](O,A,B)
\end{tikzpicture}
```

see a more complex example about clipping here : 46.6
???????????????????? 46.6?

28.5. Clipping a sector ????

☞ Attention the arguments vary according to the options.

```
\tkzClipSector[⟨local options⟩](⟨O,...⟩)(⟨...⟩)
```

options	default	definition
towards	towards	O is the center and the sector starts from A to (OB)
rotate	towards	The sector starts from A and the angle determines its amplitude.
R	towards	We give the radius and two angles

You have to add, of course, all the styles of TikZ for tracings...

options	arguments	example
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)(30,90)`

Attention the arguments vary according to the options.

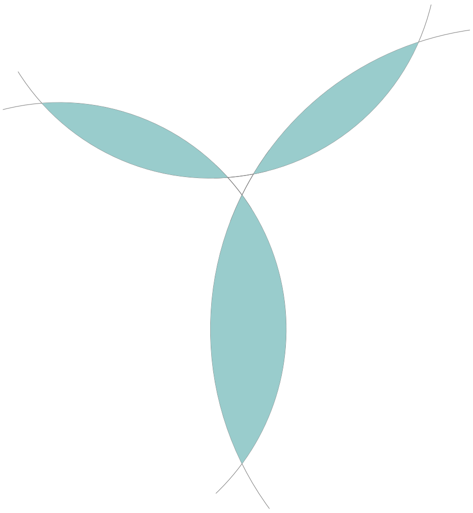
```
\tkzClipSector[⟨local options⟩](⟨O,...⟩)(⟨...⟩)
```

??	???	??
towards	towards	O???????A???(OB)?
rotate	towards	???A????????????
R	towards	?????????

?????? TikZ ???

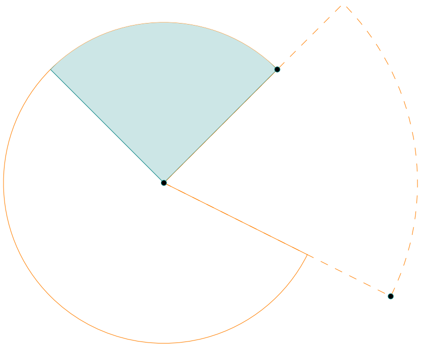
??	??	??
towards	(⟨pt,pt⟩)(⟨pt⟩)	`\tkzClipSector(O,A)(B)`
rotate	(⟨pt,pt⟩)(⟨angle⟩)	`\tkzClipSector[rotate](O,A)(90)`
R	(⟨pt,r⟩)(⟨angle 1,angle 2⟩)	`\tkzClipSector[R](O,2)(30,90)`

28.5.1. Example 1



```
\begin{tikzpicture}[scale=0.5]
\tkzDefPoint(0,0){a}
\tkzDefPoint(12,0){b}
\tkzDefPoint(4,10){c}
\tkzInterCC[R](a,6)(b,8)
\tkzGetFirstPoint{AB1} \tkzGetSecondPoint{AB2}
\tkzInterCC[R](a,6)(c,6)
\tkzGetFirstPoint{AC1} \tkzGetSecondPoint{AC2}
\tkzInterCC[R](b,8)(c,6)
\tkzGetFirstPoint{BC1} \tkzGetSecondPoint{BC2}
\tkzDrawArc(a,AB2)(AB1)
\tkzDrawArc(b,AB1)(AB2)
\tkzDrawArc(a,AC2)(AC1)
\tkzDrawArc(c,AC1)(AC2)
\tkzDrawArc(b,BC2)(BC1)
\tkzDrawArc(c,BC1)(BC2)
\begin{scope}
\tkzClipSector(b,BC2)(BC1)
\tkzFillSector[teal!40!white](c,BC1)(BC2)
\end{scope}
\begin{scope}
\tkzClipSector(a,AB2)(AB1)
\tkzFillSector[teal!40!white](b,AB1)(AB2)
\end{scope}
\begin{scope}
\tkzClipSector(a,AC2)(AC1)
\tkzFillSector[teal!40!white](c,AC1)(AC2)
\end{scope}
\end{tikzpicture}
```

28.5.2. Example 2



```
\begin{tikzpicture}[scale=1.5]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzDrawSector[new,dashed](O,A)(B)
  \tkzDrawSector[new](O,B)(A)
\begin{scope}
\tkzClipSector(O,B)(A)
\tkzDefSquare(O,B) \tkzGetPoints{B'}{O'}
\tkzDrawPolygon[color=teal,fill=teal!20](O,B,B',O')
\end{scope}
\tkzDrawPoints(A,B,O)
\end{tikzpicture}
```

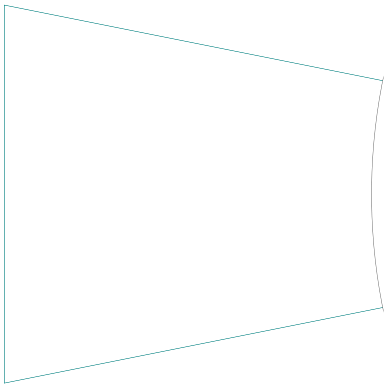
28.6. Options from *TikZ*: trim left or right ????? *TikZ*????????

See the **pgfmanual**
?? **pgfmanual**

28.7. *TikZ* Controls \pgfinterruptboundingbox and \endpgfinterruptboundingbox

This command temporarily interrupts the calculation of the box and configures a new box. See the **pgfmanual** ????????????????????????? ?? **pgfmanual**?

28.7.1. Example about contolling the bouding box ?????



```
\begin{tikzpicture}
\tkzDefPoint(0,5){A}\tkzDefPoint(5,4){B}
\tkzDefPoint(0,0){C}\tkzDefPoint(5,1){D}
\tkzDrawSegments(A,B C,D A,C)
\pgfinterruptboundingbox
   \tkzInterLL(A,B)(C,D)\tkzGetPoint{I}
\endpgfinterruptboundingbox
\tkzClipBB
\tkzDrawCircle(I,B)
\end{tikzpicture}
```

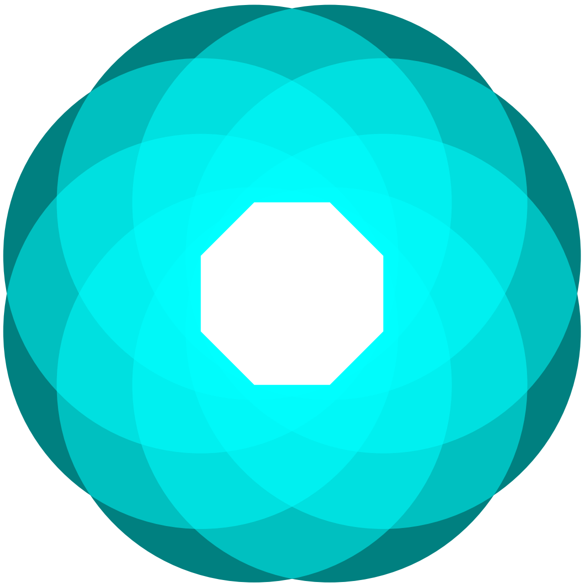
28.8. Reverse clip: tkzreverseclip ????

In order to use this option, a bounding box must be defined.
???????????????????

```
\tikzset{tkzreverseclip/.style={insert path={
   (current bounding box.south west) --(current bounding box.north west)
 --(current bounding box.north east) --  (current bounding box.south east)
 -- cycle} }}
```

28.8.1. Example with \tkzClipPolygon[out]

\tkzClipPolygon[out], \tkzClipCircle[out] use this option.



```
\begin{tikzpicture}[scale=1]
\tkzInit[xmin=-5,xmax=5,ymin=-4,ymax=6]
\tkzClip
\tkzDefPoints{-.5/0/P1,.5/0/P2}
\foreach \i [count=\j from 3] in {2,...,7}{%
    \tkzDefShiftPoint[P\i]({45*(\i-1)}:1){P\j}}
\tkzClipPolygon[out](P1,P...,P8)
\tkzCalcLength(P1,P5)\tkzGetLength{r}
\begin{scope}[blend group=screen]
  \foreach \i in {1,...,8}{%
   \tkzDefCircle[R](P\i,\r) \tkzGetPoint{x}
   \tkzFillCircle[color=teal](P\i,x)}
  \end{scope}
\end{tikzpicture}
```

Part V.

Marking 标记

28.9. Mark a segment \tkzMarkSegment 标记线段

```
\tkzMarkSegment[⟨local options⟩](⟨pt1,pt2⟩)
```

The macro allows you to place a mark on a segment.

options	default	definition
pos	.5	position of the mark
color	black	color of the mark
mark	none	choice of the mark
size	4pt	size of the mark

Possible marks are those provided by TikZ, but other marks have been created based on an idea by Yves Combe.

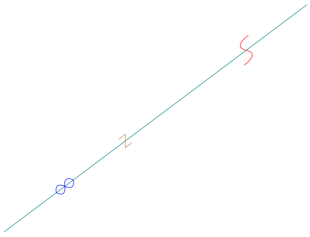
```
\tkzMarkSegment[⟨ 命令选项 ⟩](⟨pt1,pt2⟩)
```

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

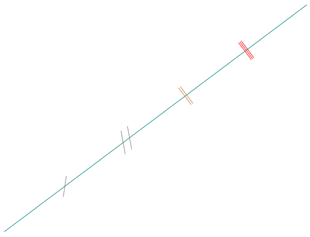
标记样式由 TikZ 提供，但也可使用基于 Yves Combe 方法的自定义标记样式。

28.9.1. Several marks 几种标记



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

28.9.2. Use of **mark**



```
\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=||](A,B)
  \tkzMarkSegment[color=red,pos=0.8,mark=|||](A,B)
\end{tikzpicture}
```

28.10. Marking segments \tkzMarkSegments 标记多条线段

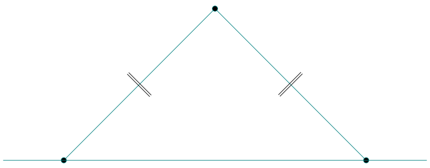
`\tkzMarkSegments[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)`

Arguments are a list of pairs of points separated by spaces. The styles of Ti*k*Z are available for plots.

`\tkzMarkSegments[⟨ 命令选项 ⟩](⟨pt1,pt2 pt3,pt4,…⟩)`

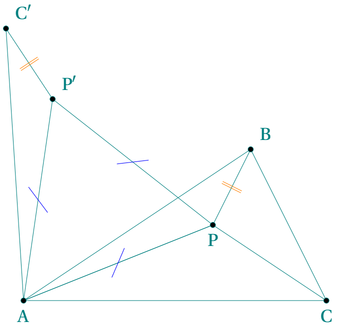
参数是用空格分隔的线段端点列表，每对端点用逗号分隔。可以使用所有有效 Ti*k*Z 样式。

28.10.1. Marks for an isosceles triangle 标记等腰三角形



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

28.11. Another marking 其它标记



```
 \begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){A}\tkzDefPoint(3,2){B}
  \tkzDefPoint(4,0){C}\tkzDefPoint(2.5,1){P}
  \tkzDrawPolygon(A,B,C)
  \tkzDefEquilateral(A,P) \tkzGetPoint{P'}
  \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}
```

28.12. Mark an arc \tkzMarkArc 标记弧

`\tkzMarkArc[⟨local options⟩](⟨pt1,pt2,pt3⟩)`

The macro allows you to place a mark on an arc. pt1 is the center, pt2 and pt3 are the endpoints of the arc.

options	default	definition
pos	.5	position of the mark
color	black	color of the mark
mark	none	choice of the mark
size	4pt	size of the mark

Possible marks are those provided by Ti*k*Z, but other marks have been created based on an idea by Yves Combe.

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

`\tkzMarkArc[⟨local options⟩](⟨pt1,pt2,pt3⟩)`

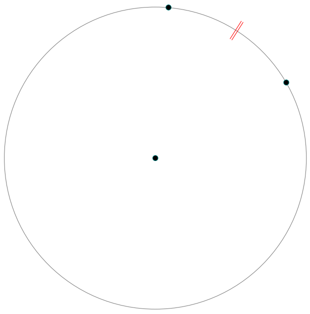
该宏允许在一个圆弧上放置一个标记。pt1 是中心，pt2 和 pt3 是圆弧的端点。

选项	默认值	含义
pos	.5	标记的位置
color	black	标记的颜色
mark	none	标记的选择
size	4pt	标记的大小

可能的标记是由 Ti*k*Z 提供的，其他标记是根据 Yves Combe 的想法创建的。

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

28.12.1. Several marks 几种标记



```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\pgfmathsetmacro\r{2}
\tkzDefPoint(30:\r){A}
\tkzDefPoint(85:\r){B}
\tkzDrawCircle(O,A)
\tkzMarkArc[color=red,mark=||](O,A,B)
\tkzDrawPoints(B,A,O)
\end{tikzpicture}
```

28.13. Mark an angle mark : \tkzMarkAngle 标记角

More delicate operation because there are many options. The symbols used for marking in addition to those of Ti*k*Z are defined in the file `tkz-lib-marks.tex` and designated by the following characters:
在 Ti*k*Z 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

`\tkzMarkAngle[⟨local options⟩](⟨A,O,B⟩)`

O is the vertex. Attention the arguments vary according to the options. Several markings are possible. You can simply draw an arc or add a mark on this arc. The style of the arc is chosen with the option **arc**, the radius of the arc is given by **mksize**, the arc can, of course, be colored.

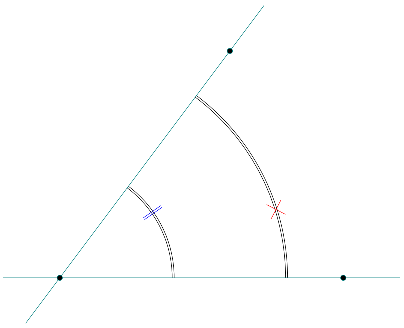
options	default	definition
arc	l	choice of l, ll and lll (single, double or triple).
size	1 (cm)	arc radius.
mark	none	choice of mark.
mksize	4pt	symbol size (mark).
mkcolor	black	symbol color (mark).
mkpos	0.5	position of the symbol on the arc.

`\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	l	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

28.13.1. Example with mark = x and with mark =||



```
\begin{tikzpicture}[scale=.75]
    \tkzDefPoints{0/0/O,5/0/A,3/4/B}
    \tkzMarkAngle[size = 4,mark = x,
                 arc=ll,mkcolor = red,mkpos=.33](A,O,B)
    \tkzMarkAngle[size = 2,mark = ||,
               arc=ll,mkcolor = blue,mkpos=.66](A,O,B)
    \tkzDrawLines(O,A O,B)
    \tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

`\tkzMarkAngles[⟨local options⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.`

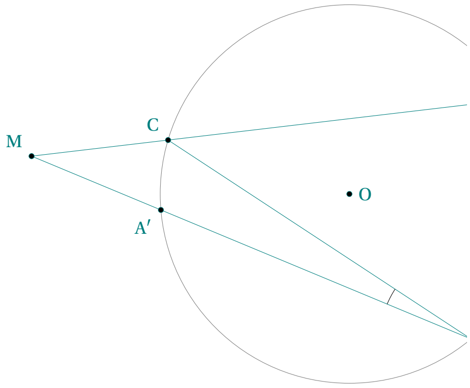
With common options, there is a macro for multiple angles.

`\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。

28.14. Problem to mark a small angle: Option veclen 标记小角

问题来自于 ”decorate”（装饰）动作，以及在**\tkzMarkAngle** 中使用的值解决方法是将宏 **\tkzMarkAngle** 包围起来。在下一个例子中，如果没有 ” 范围”，结果是：Latex 错误：尺寸太大。



```
\begin{tikzpicture}[scale=1]
  \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[common=B](M,B)(O,B) \tkzGetFirstPoint{C}
  \tkzInterLC[common=A](M,A)(O,A) \tkzGetFirstPoint{A'}
  \tkzDrawSegments(A,C M,A M,B A,B)
  \tkzDrawCircle(O,N)
  \begin{scope}[veclen]
    \tkzMarkAngle[mkpos=.2, size=1.2](C,A,M)
  \end{scope}
  \tkzDrawPoints(O, A, B, M, B, C, A')
  \tkzLabelPoints[right](O,A,B)
  \tkzLabelPoints[above left](M,C)
  \tkzLabelPoint[below left](A'){$A'$}
\end{tikzpicture}
```

28.15. Marking a right angle: \tkzMarkRightAngle 标记直角

\tkzMarkRightAngle[⟨local options⟩](⟨A,O,B⟩)

The **german** option allows you to change the style of the drawing. The option **size** allows to change the size of the drawing.

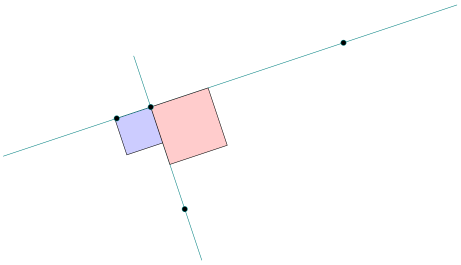
options	default	definition
german	normal	german arc with inner point.
size	0.2	side size.

\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。

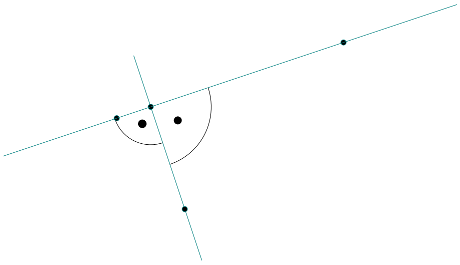
选项	默认值	含义
german	normal	带内点的圆弧
size	0.2	标记边的尺寸

28.15.1. Example of marking a right angle 直角标记



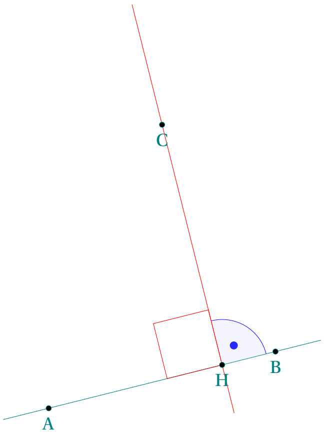
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
  \tkzDefPointBy[projection = onto B--A](P)  \tkzGetPoint{H}
  \tkzDrawLines[add=.5 and .5](P,H)
  \tkzMarkRightAngle[fill=blue!20,size=.5,draw](A,H,P)
  \tkzDrawLines[add=.5 and .5](A,B)
  \tkzMarkRightAngle[fill=red!20,size=.8](B,H,P)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

28.15.2. Example of marking a right angle, german style 使用 german 样式添加直角标记



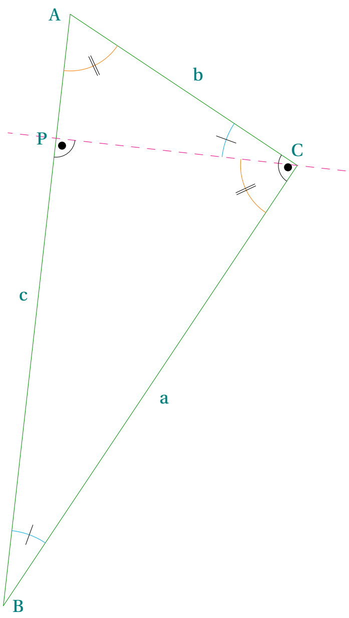
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
  \tkzDefPointBy[projection = onto B--A](P)  \tkzGetPoint{H}
  \tkzDrawLines[add=.5 and .5](P,H)
  \tkzMarkRightAngle[german,size=.5,draw](A,H,P)
  \tkzDrawPoints[](A,B,P,H)
  \tkzDrawLines[add=.5 and .5](A,B)
  \tkzMarkRightAngle[german,size=.8](P,H,B)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

28.15.3. Mix of styles 混合样式



```
\begin{tikzpicture}[scale=.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,H)
\end{tikzpicture}
```

28.15.4. Full example 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)
\tkzDefSpcTriangle[orthic](A,B,C){N,O,P}
\tkzDrawLine[dashed,color=magenta](C,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=1,color=cyan,mark=|](C,B,A)
\tkzMarkAngle[size=1,color=cyan,mark=|](A,C,P)
\tkzMarkAngle[size=0.75,color=orange,
    mark=||](P,C,B)
\tkzMarkAngle[size=0.75,color=orange,
   mark=||](B,A,C)
\tkzMarkRightAngle[german](A,C,B)
\tkzMarkRightAngle[german](B,P,C)
\end{tikzpicture}
```

28.16. \tkzMarkRightAngles 标记多个直角

\tkzMarkRightAngles[⟨local options⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.

With common options, there is a macro for multiple angles.

\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。

28.17. Angles Library 角库

If you prefer to use Ti*k*Z library **angles**, you can mark angles with the macro **\tkzPicAngle** and **\tkzPicRightAngle**.
如果喜欢使用 Ti*k*Z 的 **angles** 库， 可以用宏**\tkzPicAngle** 和 **\tkzPicRightAngle** 标记角度。

\tkzPicAngle[⟨tikz options⟩](⟨A,O,B⟩)

options	example	definition
tikz option	see below	drawing of the angle $\widehat{AOB}$.

`\tkzPicAngle[⟨tikz options⟩](⟨A,O,B⟩)`

选项	默认值	含义
tikz option	see below	绘制角 $\widehat{AOB}$.

`\tkzPicRightAngle[⟨tikz options⟩](⟨A,O,B⟩)`

options	example	definition
tikz option	see below	drawing of the right angle $\widehat{AOB}$.

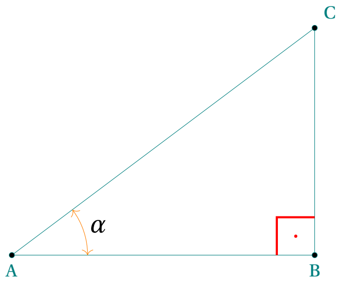
You need to know possible options of the **`angles`** *library*

`\tkzPicRightAngle[⟨tikz options⟩](⟨A,O,B⟩)`

选项	默认值	含义
tikz option	see below	绘制直角 $\widehat{AOB}$.

你需要知道 ***angles*** 库的可能选项

28.17.1. Angle with TikZ



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/0/B}
\tkzDefTriangle[right,swap](A,B) \tkzGetPoint{C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\tkzLabelPoints[below](B,A)
\tkzLabelPoints[above right](C)
\tkzPicAngle["$\alpha$",draw=orange,
             <->,angle eccentricity=1.2,
             angle radius=1cm](B,A,C)
\tkzPicRightAngle[draw,red,thick,
              angle eccentricity=.5,
              pic text=.](C,B,A)
\end{tikzpicture}
```

Part VI.

Labelling 标签

29. Labelling 标签

29.1. Label for a point 点的标签

It is possible to add several labels at the same point by using this macro several times.
通过多次使用该宏，可以在同一点添加多个标签。

```
\tkzLabelPoint[⟨local options⟩](⟨point⟩){⟨label⟩}
```

arguments	example	
point	\tkzLabelPoint(A){A_1}	
options	default	definition
TikZ options		colour, position etc.

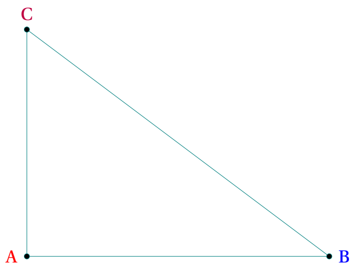
Optionally, we can use any style of TikZ, especially placement with above, right, dots...

```
\tkzLabelPoint[⟨local options⟩](⟨point⟩){⟨label⟩}
```

参数	示例	
point	\tkzLabelPoint(A){A_1}	
选项	默认值	含义
TikZ 选项		颜色、位置等。

可以选择使用任何风格的TikZ，特别是上面、右边、点的位置。

29.1.1. Example with \tkzLabelPoint



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(0,3){C}
  \tkzDrawSegments(A,B B,C C,A)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoint[left,red](A){$A$}
  \tkzLabelPoint[right,blue](B){$B$}
  \tkzLabelPoint[above,purple](C){$C$}
\end{tikzpicture}
```

29.1.2. Label and reference 标签和引用

The reference of a point is the object that allows to use the point, the label is the name of the point that will be displayed.
一个点的引用是允许使用该点的对象，标签是将被显示的该点的名称。

```
\begin{tikzpicture}
   \tkzDefPoint(2,0){A}
   \tkzDrawPoint(A)
   \tkzLabelPoint[above](A){$A_1$}
 \end{tikzpicture}
```

29.2. Add labels to points \tkzLabelPoints 标签多个点

It is possible to place several labels quickly when the point references are identical to the labels and when the labels are placed in the same way in relation to the points. By default, `below right` is chosen.
当点的引用与标签相同，并且标签相对于点的放置方式相同时，就有可能快速放置多个标签。默认情况下，选择的是 `below right`【右下角】。

\tkzLabelPoints[⟨local options⟩](⟨$A_1,A_2,\ldots$⟩)

arguments	example	result
list of points	\tkzLabelPoints(A,B,C)	Display of A, B and C

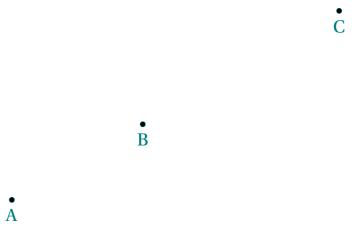
This macro reduces the number of lines of code, but it is not obvious that all points need the same label positioning.

\tkzLabelPoints[⟨local options⟩](⟨$A_1,A_2,\ldots$⟩)

参数	示例	结果
点列	\tkzLabelPoints(A,B,C)	显示 A, B , C

这个宏减少了代码行数，但并不明显，所有的点都需要相同的标签定位。

29.2.1. Example with \tkzLabelPoints



```
\begin{tikzpicture}
  \tkzDefPoint(2,3){A}
  \tkzDefShiftPoint[A](30:2){B}
  \tkzDefShiftPoint[A](30:5){C}
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

29.3. Automatic position of labels \tkzAutoLabelPoints 标签自动定位

The label of a point is placed in a direction defined by a center and a point `center`. The distance to the point is determined by a percentage of the distance between the center and the point. This percentage is given by `dist`.
一个点的标签被放置在一个由中心和一个点 `center` 定义的方向上。与该点的距离是由中心和该点之间的距离的百分比决定的。这个百分比由 `dist` 给出。

\tkzLabelPoints[⟨local options⟩](⟨$A_1,A_2,\ldots$⟩)

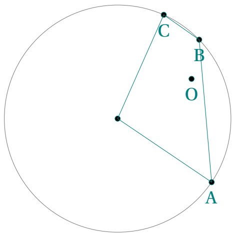
arguments	example	result
list of points	\tkzLabelPoint(A,B,C)	Display of A, B and C

\tkzLabelPoints[⟨local options⟩](⟨$A_1,A_2,\ldots$⟩)

参数	示例	结果
点列	\tkzLabelPoint(A,B,C)	显示 A, B , C

29.3.1. Label for points with \tkzAutoLabelPoints

Here the points are positioned relative to the center of gravity of A, B, C and O.
这里的点是相对于 A, B, C和O 的重心而定位的。



```
\begin{tikzpicture}[scale=1]
 \tkzDefPoint(2,1){O}
 \tkzDefRandPointOn[circle=center O radius 1.5]\tkzGetPoint{A}
 \tkzDefPointBy[rotation=center O angle 100](A)\tkzGetPoint{C}
 \tkzDefPointBy[rotation=center O angle 78](A)\tkzGetPoint{B}
 \tkzDrawCircle(O,A)
 \tkzDrawPoints(O,A,B,C)
 \tkzDrawSegments(C,B B,A A,O O,C)
 \tkzDefTriangleCenter[centroid](A,B,C) \tkzGetPoint{O}
 \tkzDrawPoint(tkzPointResult)
 \tkzLabelPoints(O,A,C,B)
\end{tikzpicture}
```

30. Label for a segment 线段的标签

\tkzLabelSegment[⟨local options⟩](⟨pt1,pt2⟩){⟨label⟩}

This macro allows you to place a label along a segment or a line. The options are those of TikZ for example `pos`.

argument	example	definition
label	`\tkzLabelSegment(A,B){5}`	label text
(pt1,pt2)	(A,B)	label along [AB]

options	default	definition
pos	.5	label's position

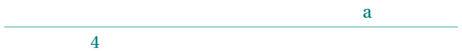
\tkzLabelSegment[⟨local options⟩](⟨pt1,pt2⟩){⟨label⟩}

这个宏允许给线段或直线放置一个标签。选项是那些 TikZ 的选项，例如 `pos`。

参数	示例	含义
label	`\tkzLabelSegment(A,B){5}`	标签文字
(pt1,pt2)	(A,B)	标签线段 [AB]

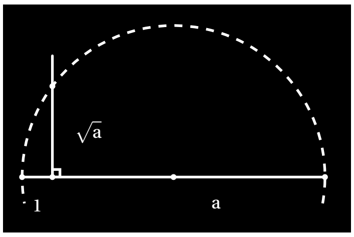
选项	默认值	含义
pos	.5	标签的位置

30.0.1. First example



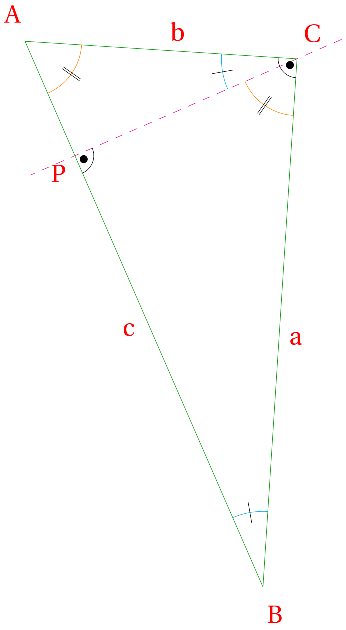
```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(6,0){B}
\tkzDrawSegment(A,B)
\tkzLabelSegment[above,pos=.8](A,B){$a$}
\tkzLabelSegment[below,pos=.2](A,B){$4$}
\end{tikzpicture}
```

30.0.2. Example : blackboard 示例：黑板



```
\tikzstyle{background rectangle}=[fill=black]
\begin{tikzpicture}[show background rectangle,scale=.4]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){I}
  \tkzDefPoint(10,0){A}
  \tkzDefPointWith[orthogonal normed,K=4](I,A)
   \tkzGetPoint{H}
  \tkzDefMidPoint(O,A) \tkzGetPoint{M}
  \tkzInterLC(I,H)(M,A)\tkzGetPoints{B}{C}
  \tkzDrawSegments[color=white,line width=1pt](I,H O,A)
  \tkzDrawPoints[color=white](O,I,A,B,M)
  \tkzMarkRightAngle[color=white,line width=1pt](A,I,B)
  \tkzDrawArc[color=white,line width=1pt,
             style=dashed](M,A)(O)
  \tkzLabelSegment[white,right=1ex,pos=.5](I,B){$\sqrt{a}$}
  \tkzLabelSegment[white,below=1ex,pos=.5](O,I){$1$}
  \tkzLabelSegment[pos=.6,white,below=1ex](I,A){$a$}
\end{tikzpicture}
```

30.0.3. Labels and option : swap



```
\begin{tikzpicture}[rotate=-60]
\tkzSetUpStyle[red,auto]{label style}
\tkzDefPoint(0,1){A}
\tkzDefPoint(2,4){C}
\tkzDefPointWith[orthogonal normed,K=7](C,A)
\tkzGetPoint{B}
\tkzDefSpcTriangle[orthic](A,B,C){N,O,P}
\tkzDefTriangleCenter[circum](A,B,C)
\tkzGetPoint{O}
\tkzDrawPolygon[green!60!black](A,B,C)
\tkzDrawLine[dashed,color=magenta](C,P)
\tkzLabelSegment(B,A){$c$}
\tkzLabelSegment[swap](B,C){$a$}
\tkzLabelSegment[swap](C,A){$b$}
\tkzMarkAngles[size=1,
     color=cyan,mark=|](C,B,A A,C,P)
\tkzMarkAngle[size=0.75,
     color=orange,mark=||](P,C,B)
\tkzMarkAngle[size=0.75,
      color=orange,mark=||](B,A,C)
\tkzMarkRightAngles[german](A,C,B B,P,C)
\tkzAutoLabelPoints[center = O,dist= .1](A,B,C)
 \tkzLabelPoint[below left](P){$P$}
 \end{tikzpicture}
```

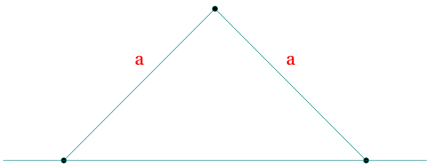
\tkzLabelSegments[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)

The arguments are a two-point couple list. The styles of TikZ are available for plotting.

\tkzLabelSegments[⟨local options⟩](⟨pt1,pt2 pt3,pt4 ...⟩)

参数是一个两点对的列表。绘图时可使用 TikZ 的样式。

30.0.4. Labels for an isosceles triangle 等腰三角形的标签



```
\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)
 \tkzLabelSegments[color=red,above=4pt](O,A A,B){$a$}
\end{tikzpicture}
```

31. Add labels on a straight line \tkzLabelLine 在一条直线上添加标签

`\tkzLabelLine[⟨local options⟩](⟨pt1,pt2⟩){⟨label⟩}`

arguments	default	definition
label		`\tkzLabelLine(A,B){$\Delta$}`

options	default	definition
pos	.5	pos is an option for TikZ, but essential in this case…

As an option, and in addition to the pos, you can use all styles of TikZ, especially the placement with **above**, **right**, …

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

参数	默认值	含义
label		`\tkzLabelLine(A,B){$\Delta$}`

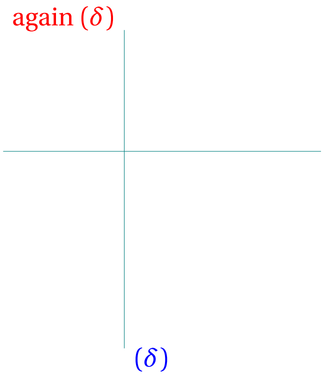
选项	默认值	含义
pos	.5	pos 是 TikZ 的一个选项

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。

31.0.1. Example with \tkzLabelLine

An important option is pos, it's the one that allows you to place the label along the right. The value of pos can be greater than 1 or negative.

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}
```

31.1. Label at an angle : \tkzLabelAngle 标注角

```
\tkzLabelAngle[⟨local options⟩](⟨A,O,B⟩)
```

There is only one option, dist (with or without unit), which can be replaced by the TikZ's pos option (without unit for the latter). By default, the value is in centimeters.

options	default	definition
pos	1	or dist, controls the distance from the top to the label.

It is possible to move the label with all TikZ options : rotate, shift, below, etc.

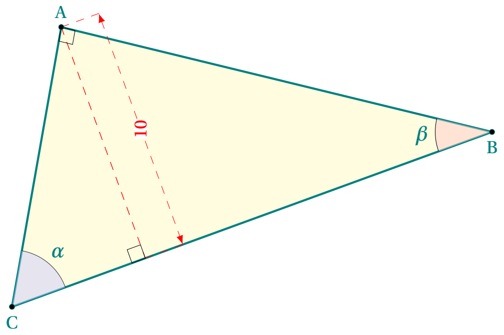
```
\tkzLabelAngle[⟨ 命令选项 ⟩](⟨A,O,B⟩)
```

该命令只有一个 dist 选项 (带或不带单位)，该选项可以被 Ti*k*Z 的 pos 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
pos	1	或是 dist，用于控制标注的距离

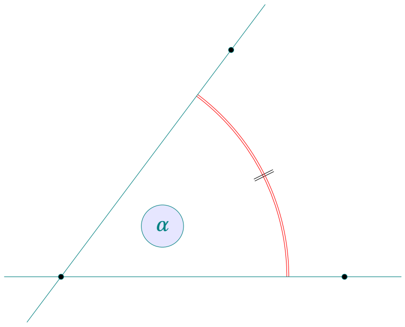
可以使用 Ti*k*Z 的 rotate、shift、below 等选项调整标注的位置。

31.1.1. Example author js bibra stackexchange



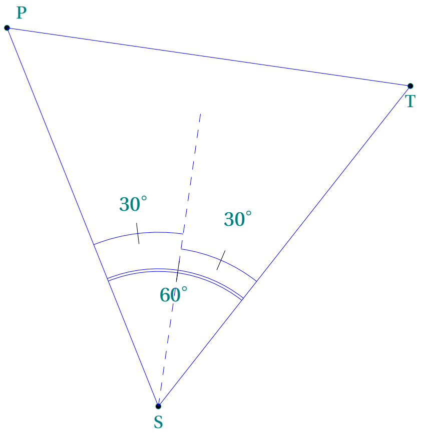
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(20:9){B}
  \tkzDefPoint(80:5){A}
  \tkzDefPointsBy[projection=onto B--C](A){a}
  \tkzDrawPolygon[thick,fill=yellow!15](A,B,C)
  \tkzDrawSegment[dashed, red](A,a)
  \tkzDrawSegment[style=red, dashed,
  dim={$10$,15pt,midway,font=\scriptsize,
   rotate=90}](A,a)
  \tkzMarkAngle(B,C,A)
  \tkzMarkRightAngle(A,a,C)
  \tkzMarkRightAngle(C,A,B)
  \tkzFillAngle[fill=blue!20, opacity=0.5](B,C,A)
  \tkzFillAngle[fill=red!20, opacity=0.5](A,B,C)
  \tkzLabelAngle[pos=1.25](A,B,C){$\beta$}
  \tkzLabelAngle[pos=1.25](B,C,A){$\alpha$}
  \tkzMarkAngle(A,B,C)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(B,C)
  \tkzLabelPoints[above](A)
\end{tikzpicture}
```

31.1.2. With pos



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O,5/0/A,3/4/B}
  \tkzMarkAngle[size = 4,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}
```

31.1.3. pos and \tkzLabelAngles



```
\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 = 1.8,mark = |,arc=ll,
                  color = blue](T,S,P)
  \tkzMarkAngle[size = 2.1,mark = |,arc=l,
                  color = blue](T,S,s)
  \tkzMarkAngle[size = 2.3,mark = |,arc=l,
                  color = blue](s,S,P)
 \tkzLabelAngle[pos = 1.5](T,S,P){$60^{\circ}$}%
 \tkzLabelAngles[pos = 2.7](T,S,s s,S,P){%
                          $30^{\circ}$}%
\end{tikzpicture}
```

```
\tkzLabelAngles[⟨local options⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.
```

With common options, there is a macro for multiple angles.

```
\tkzLabelAngles[⟨local options⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.
```

有了共同的选项，可以标注多角度。

It finally remains to be able to give a label to designate a circle and if several possibilities are offered, we will see here \tkzLabelCircle.
最后就是能够给一个圆指定一个标签，可以使用，\tkzLabelCircle。

31.2. Giving a label to a circle 标注一个圆

`\tkzLabelCircle[⟨tikz options⟩](⟨O,A⟩)(⟨angle⟩){⟨label⟩}`

options	default	definition
tikz options		circle O center through A

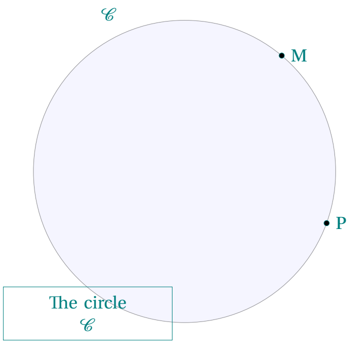
We can use the styles from TikZ. The label is created and therefore ”passed” between braces.

`\tkzLabelCircle[⟨tikz options⟩](⟨O,A⟩)(⟨angle⟩){⟨label⟩}`

选项	默认值	含义
tikz options		圆心 O，通过 A

可以使用来自 *TikZ* 的样式。标签被创建，因此在大括号之间 ”传递”。

31.2.1. Example



```
\begin{tikzpicture}
 \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!10,opacity=.4](O,M)
 \tkzLabelCircle[draw,
       text width=2cm,text centered,left=24pt](O,M)(-120)%
          {The circle\\ $\mathcal{C}$}
 \tkzDrawPoints(M,P)\tkzLabelPoints[right](M,P)
\end{tikzpicture}
```

32. Label for an arc 弧的标签

`\tkzLabelArc[⟨local options⟩](⟨pt1,pt2,pt3⟩){⟨label⟩}`

This macro allows you to place a label along an arc. The options are those of Ti*k*Z for example `pos`.

argument	example	definition
label	`\tkzLabelArc(A,B){5}`	label text
(pt1,pt2,pt3)	(O,A,B)	label along the arc $\widehat{AB}$

options	default	definition
pos	.5	label's position

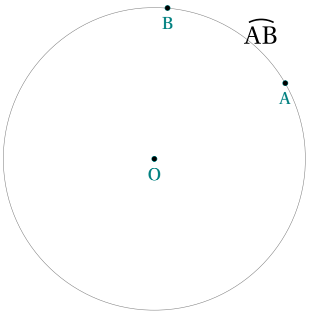
```
\tkzLabelArc[⟨local options⟩](⟨pt1,pt2,pt3⟩){⟨label⟩}
```

这个宏允许沿弧线放置一个标签。选项是 TikZ 的选项，例如 **pos**。

参数	示例	含义
label	**\tkzLabelArc(A,B){5}**	标签文本
(pt1,pt2,pt3)	(O,A,B)	沿着弧线 widearcAB 的标签

选项	默认值	含义
pos	.5	标签的位置

32.0.1. Label on arc



```
\begin{tikzpicture}
\tkzDefPoint(0,0){O}
\pgfmathsetmacro\r{2}
\tkzDefPoint(30:\r){A}
\tkzDefPoint(85:\r){B}
\tkzDrawCircle(O,A)
\tkzDrawPoints(B,A,O)
\tkzLabelArc[right=2pt](O,A,B){$\widearc{AB}$}
\tkzLabelPoints(A,B,O)
\end{tikzpicture}
```

Part VII.

Complements 补充

33. Using the compass 尺规标记

33.1. Main macro \tkzCompass 绘制尺规标记

`\tkzCompass[⟨local options⟩](⟨A,B⟩)`

This macro allows you to leave a compass trace, i.e. an arc at a designated point. The center must be indicated. Several specific options will modify the appearance of the arc as well as TikZ options such as style, color, line thickness etc.
You can define the length of the arc with the option `length` or the option `delta`.

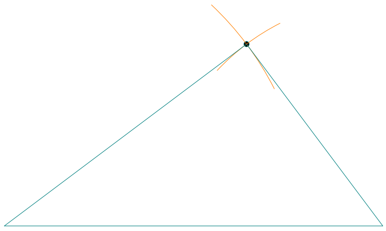
options	default	definition
delta	0 (deg)	Increases the angle of the arc symmetrically
length	1 (cm)	Changes the length (in cm)

`\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 Ti*k*Z 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。

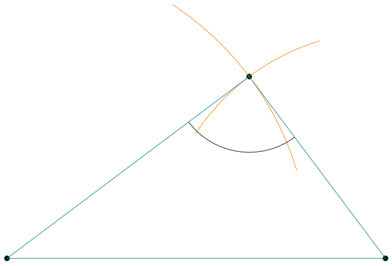
选项	默认值	含义
delta	0 (deg)	延伸长度 (度)
length	1 (cm)	圆弧长度 (cm)

33.1.1. Option length



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

33.1.2. Option delta



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

33.2. Multiple constructions \tkzCompasss 绘制多个尺规标记

```
\tkzCompasss[⟨local options⟩](⟨pt1,pt2 pt3,pt4,…⟩)
```

Attention the arguments are lists of two points. This saves a few lines of code.

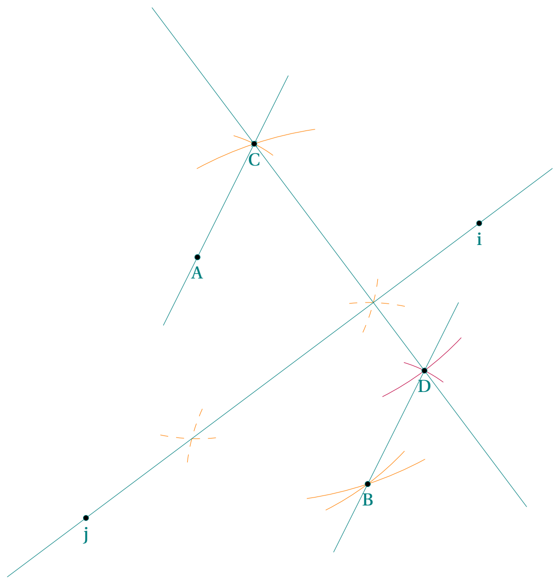
options	default	definition
delta	0	Modifies the angle of the arc by increasing it symmetrically
length	1	Changes the length

```
\tkzCompasss[⟨命令选项⟩](⟨pt1,pt2, pt3,pt4,…⟩)
```

注意：参数是点对列表。

选项	默认值	含义
delta	0	延伸角度
length	1	圆弧长度

33.2.1. Use \tkzCompasss



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

34. The Show 显示尺规标记

34.1. Show the constructions of some lines \tkzShowLine 显示直线尺规标记

```
\tkzShowLine[⟨local options⟩](⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)
```

These constructions concern mediatrices, perpendicular or parallel lines passing through a given point and bisectors. The arguments are therefore lists of two or three points. Several options allow the adjustment of the constructions. The idea of this macro comes from **Yves Combe**.

options	default	definition
mediator	mediator	displays the constructions of a mediator
perpendicular	mediator	constructions for a perpendicular
orthogonal	mediator	idem
bisector	mediator	constructions for a bisector
K	1	circle within a triangle
length	1	in cm, length of a arc
ratio	.5	arc length ratio
gap	2	placing the point of construction
size	1	radius of an arc (see bisector)

You have to add, of course, all the styles of Ti*k*Z for tracings…

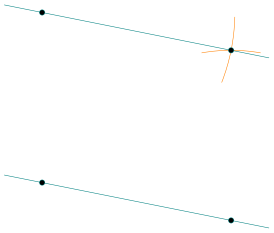
```
\tkzShowLine[⟨命令选项⟩](⟨pt1,pt2⟩) 或 (⟨pt1,pt2,pt3⟩)
```

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

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

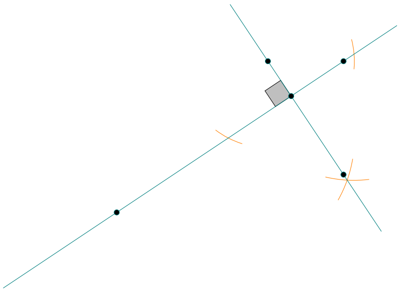
可以使用所有有效 Ti*k*Z 样式。

34.1.1. Example of \tkzShowLine and parallel



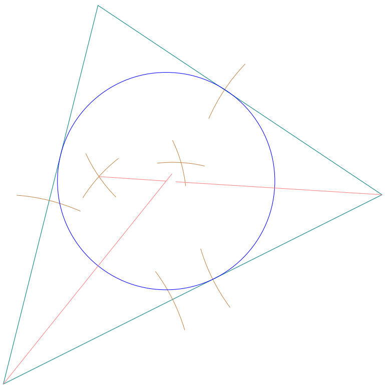
```
\begin{tikzpicture}
 \tkzDefPoints{-1.5/-0.25/A,1/-0.75/B,-1.5/2/C}
 \tkzDrawLine(A,B)
 \tkzDefLine[parallel=through C](A,B) \tkzGetPoint{c}
 \tkzShowLine[parallel=through C](A,B)
 \tkzDrawLine(C,c) \tkzDrawPoints(A,B,C,c)
\end{tikzpicture}
```

34.1.2. Example of \tkzShowLine and perpendicular



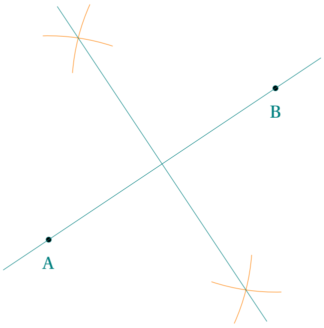
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A, 3/2/B, 2/2/C}
\tkzDefLine[perpendicular=through C,K=-.5](A,B) \tkzGetPoint{c}
\tkzShowLine[perpendicular=through C,K=-.5,gap=3](A,B)
\tkzDefPointBy[projection=onto A--B](c)\tkzGetPoint{h}
\tkzMarkRightAngle[fill=lightgray](A,h,C)
\tkzDrawLines[add=.5 and .5](A,B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

34.1.3. Example of \tkzShowLine and bisector



```
\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[color=blue,%
 line width=.2pt](I,H)
 \tkzDrawSegments[color=red!50](I,tkzPointResult)
 \tkzDrawLines[add=0 and -0.3,color=red!50](A,a B,b)
\end{tikzpicture}
```

34.1.4. Example of \tkzShowLine and mediator



```
\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}
```

34.2. Constructions of certain transformations \tkzShowTransformation 显示部分变换过程尺规标记

```
\tkzShowTransformation[⟨local options⟩](⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)
```

These constructions concern orthogonal symmetries, central symmetries, orthogonal projections and translations. Several options allow the adjustment of the constructions. The idea of this macro comes from **Yves Combe**.

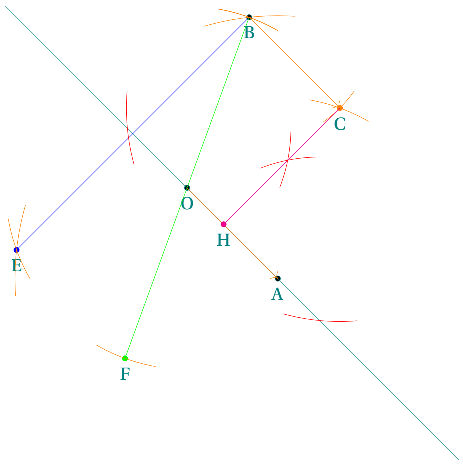
options	default	definition
reflection= over pt1--pt2	reflection	constructions of orthogonal symmetry
symmetry=center pt	reflection	constructions of central symmetry
projection=onto pt1--pt2	reflection	constructions of a projection
translation=from pt1 to pt2	reflection	constructions of a translation
K	1	circle within a triangle
length	1	arc length
ratio	.5	arc length ratio
gap	2	placing the point of construction
size	1	radius of an arc (see bisector)

```
\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)

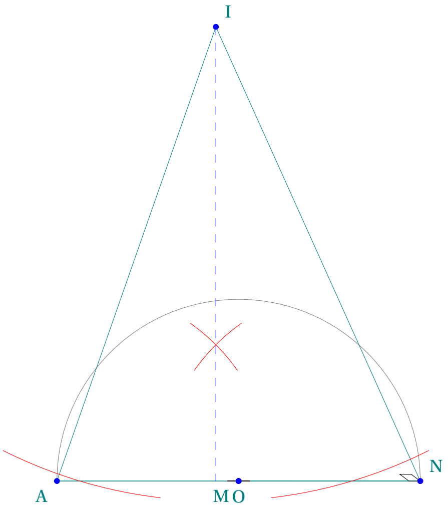
34.2.1. Example of the use of \tkzShowTransformation



```
\begin{tikzpicture}[scale=.6]
  \tkzDefPoint(0,0){O} \tkzDefPoint(2,-2){A}
  \tkzDefPoint(70:4){B} \tkzDrawPoints(A,O,B)
  \tkzLabelPoints(A,O,B)
  \tkzDrawLine[add= 2 and 2](O,A)
  \tkzDefPointBy[translation=from O to A](B)
  \tkzGetPoint{C}
  \tkzDrawPoint[color=orange](C)  \tkzLabelPoints(C)
  \tkzShowTransformation[translation=from O to A,%
           length=2](B)
  \tkzDrawSegments[->,color=orange](O,A B,C)
  \tkzDefPointBy[reflection=over O--A](B) \tkzGetPoint{E}
  \tkzDrawSegment[blue](B,E)
  \tkzDrawPoint[color=blue](E)\tkzLabelPoints(E)
  \tkzShowTransformation[reflection=over O--A,size=2](B)
  \tkzDefPointBy[symmetry=center O](B) \tkzGetPoint{F}
  \tkzDrawSegment[color=green](B,F)
  \tkzDrawPoint[color=green](F)\tkzLabelPoints(F)
  \tkzShowTransformation[symmetry=center O,%
                    length=2](B)
  \tkzDefPointBy[projection=onto O--A](C)
  \tkzGetPoint{H}
  \tkzDrawSegments[color=magenta](C,H)
  \tkzDrawPoint[color=magenta](H)\tkzLabelPoints(H)
  \tkzShowTransformation[projection=onto O--A,%
                      color=red,size=3,gap=-2](C)
\end{tikzpicture}
```

34.2.2. Another example of the use of \tkzShowTransformation

You’ll find this figure again, but without the construction features.
后续会再次使用这个示例，但不绘制尺规标记。



```
\begin{tikzpicture}[scale=.6]
  \tkzDefPoints{0/0/A,8/0/B,3.5/10/I}
  \tkzDefMidPoint(A,B) \tkzGetPoint{O}
  \tkzDefPointBy[projection=onto A--B](I)
     \tkzGetPoint{J}
  \tkzInterLC(I,A)(O,A)  \tkzGetPoints{M}{M'}
  \tkzInterLC(I,B)(O,A)  \tkzGetPoints{N}{N'}
  \tkzDefMidPoint(A,B) \tkzGetPoint{M}
  \tkzDrawSemiCircle(M,B)
  \tkzDrawSegments(I,A I,B A,B B,M A,N)
  \tkzMarkRightAngles(A,M,B A,N,B)
  \tkzDrawSegment[style=dashed,color=blue](I,J)
  \tkzShowTransformation[projection=onto A--B,
                color=red,size=3,gap=-3](I)
  \tkzDrawPoints[color=red](M,N)
  \tkzDrawPoints[color=blue](O,A,B,I,M)
  \tkzLabelPoints(O)
  \tkzLabelPoints[above right](N,I)
  \tkzLabelPoints[below left](M,A)
\end{tikzpicture}
```

35. Protractor 量角器

Based on an idea by Yves Combe, the following macro allows you to draw a protractor. The operating principle is even simpler. Just name a half-line (a ray). The protractor will be placed on the origin O, the direction of the half-line is given by A. The angle is measured in the direct direction of the trigonometric circle.
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。

35.1. The macro \tkzProtractor 绘制量角器

`\tkzProtractor[⟨local options⟩](⟨O,A⟩)`

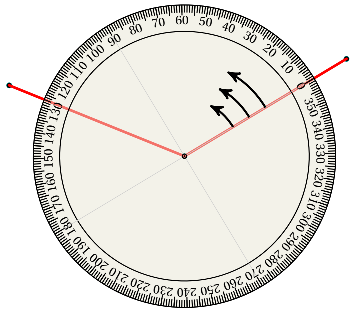
options	default	definition
lw	0.4 pt	line thickness
scale	1	ratio: adjusts the size of the protractor
return	false	trigonometric circle indirect

`\tkzProtractor[⟨命令选项⟩](⟨O,A⟩)`

选项	默认值	含义
lw	0.4 pt	线宽
scale	1	比例：用于调整量角器尺寸
return	false	反向测量圆

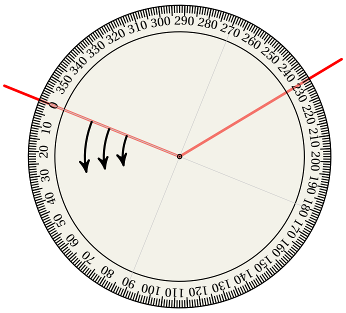
35.1.1. The circular protractor 正向圆量角器

Measuring in the forward direction
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}
```

35.1.2. The circular protractor, transparent and returned 圆形量角器，透明并返回



```
\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}
```

36. Miscellaneous tools and mathematical tools 杂项工具和数学工具

36.1. Duplicate a segment 复制线段

This involves constructing a segment on a given half-line of the same length as a given segment.
在给定的半直线上复制与给定线段长度相同的线段。

```
\tkzDuplicateSegment(⟨pt1,pt2⟩)(⟨pt3,pt4⟩){⟨pt5⟩}
```

This involves creating a segment on a given half-line of the same length as a given segment . It is in fact the definition of a point. `\tkzDuplicateSegment` is the new name of `\tkzDuplicateLen`.

arguments	example	explanation
(pt1,pt2)(pt3,pt4){pt5}	`\tkzDuplicateSegment(A,B)(E,F){C}`	AC=EF and C ∈ [AB)

The macro `\tkzDuplicateLength` is identical to this one.

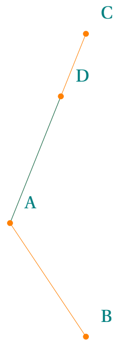
```
\tkzDuplicateSegment(⟨pt1,pt2⟩)(⟨pt3,pt4⟩)
```

这包括在给定的射线上创建一个与给定线段长度相同的线段。它实际上是一个点的定义。`\tkzDuplicateSegment` 是`\tkzDuplicateLen` 的新名称。

参数	样例	说明
(pt1,pt2)(pt3,pt4){pt5}	`\tkzDuplicateSegment(A,B)(E,F){C}`	AC=EF and C ∈ [AB)

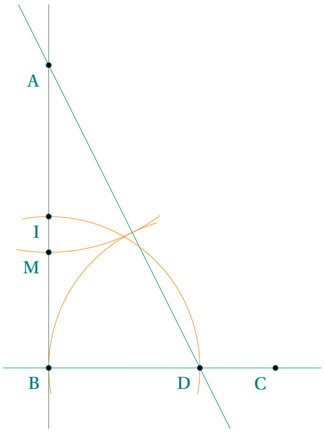
`\tkzDuplicateLength` 命令与该命令相同。

36.1.1. Use of\tkzDuplicateSegment



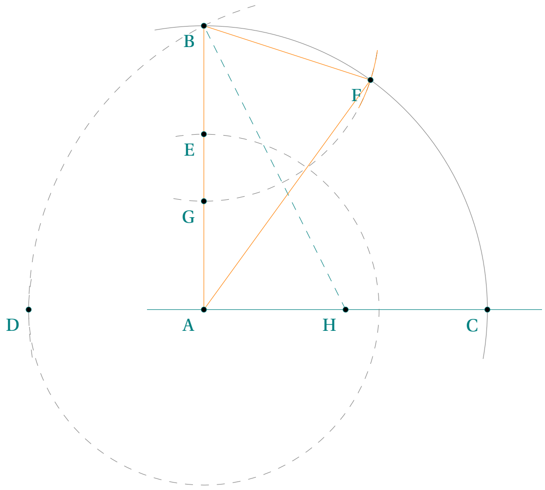
```
\begin{tikzpicture}[scale=.5]
 \tkzDefPoints{0/0/A,2/-3/B,2/5/C}
 \tkzDuplicateSegment(A,B)(A,C)
 \tkzGetPoint{D}
 \tkzDrawSegments[new](A,B A,C)
 \tkzDrawSegment[teal](A,D)
 \tkzDrawPoints[new](A,B,C,D)
 \tkzLabelPoints[above right=3pt](A,B,C,D)
\end{tikzpicture}
```

36.1.2. Proportion of gold with \tkzDuplicateSegment 黄金分割示例



```
\begin{tikzpicture}[rotate=-90,scale=.4]
 \tkzDefPoints{0/0/A,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)
 \tkzLabelPoints[below left](A,B,D,C,M,I)
\end{tikzpicture}
```

36.1.3. Golden triangle or sublime triangle 黄金三角形或 sublime 三角形



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/A,5/0/C,0/5/B}
  \tkzDefMidPoint(A,C)\tkzGetPoint{H}
  \tkzDuplicateSegment(H,B)(H,A)\tkzGetPoint{D}
  \tkzDuplicateSegment(A,D)(A,B)\tkzGetPoint{E}
  \tkzDuplicateSegment(A,D)(B,A)\tkzGetPoint{G}
  \tkzInterCC(A,C)(B,G)\tkzGetSecondPoint{F}
  \tkzDrawLine(A,C)
  \tkzDrawArc(A,C)(B)
  \begin{scope}[arc style/.style={color=gray,%
                              style=dashed}]
    \tkzDrawArc(H,B)(D)
    \tkzDrawArc(A,D)(B)
    \tkzDrawArc(B,G)(F)
  \end{scope}
  \tkzDrawSegment[dashed](H,B)
  \tkzCompass(B,F)
  \tkzDrawPolygon[new](A,B,F)
  \tkzDrawPoints(A,...,H)
  \tkzLabelPoints[below left](A,...,H)
\end{tikzpicture}
```

36.2. Segment length \tkzCalcLength 计算线段长度

There's an option in Ti*k*Z named **veclen**. This option is used to calculate AB if A and B are two points.
也可以用 Ti*k*Z 的 **veclen** 计算长度，该选项能够计算 A 点和 B 点间的距离 AB。
The only problem for me is that the version of Ti*k*Z is not accurate enough in some cases. My version uses the xfp package and is slower, but more accurate.
但 Ti*k*Z 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[⟨local options⟩](⟨pt1,pt2⟩)
```

You can store the result with the macro **\tkzGetLength** for example **\tkzGetLength{dAB}** defines the macro **\dAB**.

arguments	example	explanation
`(pt1,pt2){name of macro}`	`\tkzCalcLength(A,B)`	`\dAB gives AB in cm`

Only one option

options	default	example
cm	true	`\tkzCalcLength(A,B) After \tkzGetLength{dAB} \dAB gives AB in cm`

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

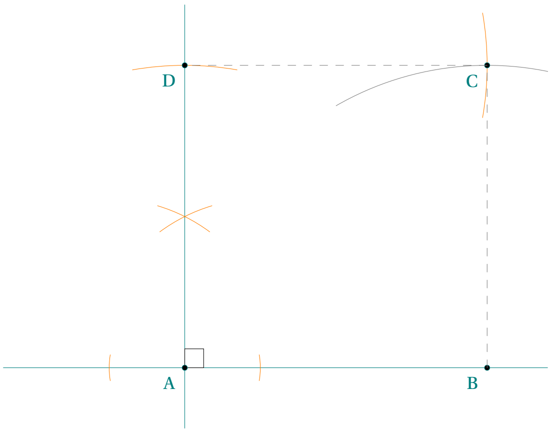
计算结果保存在指定的宏中。

参数	样例	说明
`(pt1,pt2){宏名称}`	`\tkzCalcLength(A,B){dAB}`	`\dAB` 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	`\tkzCalcLength[cm](A,B){dAB} \dAB` 得到 AB 的长度，单位是 `cm`

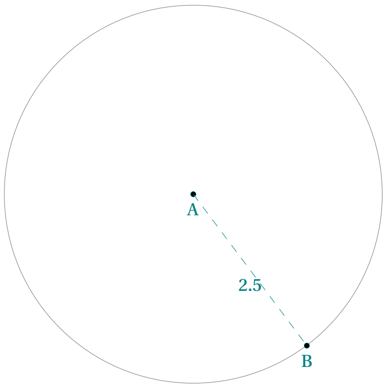
36.2.1. Compass square construction 绘制尺规标记



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

36.2.2. Example

The macro `\tkzDefCircle[radius](A,B)` defines the radius that we retrieve with `\tkzGetLength`, this result is in `cm`.
宏指令`\tkzDefCircle[radius](A,B)` 定义了用`\tkzGetLength` 检索到的半径，这个结果以 `cm` 为单位。



```
\begin{tikzpicture}[scale=.5]
 \tkzDefPoint(0,0){A}
 \tkzDefPoint(3,-4){B}
 \tkzDefMidPoint(A,B) \tkzGetPoint{M}
 \tkzCalcLength(M,B)\tkzGetLength{rAB}
 \tkzDrawCircle(A,B)
 \tkzDrawPoints(A,B)
 \tkzLabelPoints(A,B)
 \tkzDrawSegment[dashed](A,B)
 \tkzLabelSegment(A,B){$\pgfmathprintnumber{\rAB}$}
\end{tikzpicture}
```

36.3. Transformation from pt to cm or cm to pt 将 pt 转换为 cm

Not sure if this is necessary and it is only a division by 28.45274 and a multiplication by the same number. The macros are:
不能确定该命令是否有用，仅仅用 28.45274 进行了乘除运算。

`\tkzpttocm(⟨number⟩){⟨name of macro⟩}`

The result is stored in a macro.

arguments	example	explanation
(number){name of macro}	\tkzpttocm(120){len}	\len gives a number of tkznamecm

You'll have to use `\len` along with `cm`.

`\tkzpttocm(⟨nombre⟩){⟨ 宏名称 ⟩}`

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

需要使用`\len cm`【结果保存在宏中。】

36.4. Change of unit

`\tkzcmtopt(⟨number⟩){⟨name of macro⟩}`

The result is stored in a macro.

arguments	example	explanation
(number){name of macro}	\tkzcmtopt(5){len}	\len length in pts

The result can be used with `\len pt`

`\tkzcmtopt(⟨nombre⟩){⟨ 宏名称 ⟩}`

结果保存在宏中。

参数	样例	说明
(nombre){宏名称}	\tkzcmtopt(5){len}	\len 得到 pts 值

结果：\len *tkznamept*。

36.5. Get point coordinates 提取点的坐标分量

\tkzGetPointCoord(⟨A⟩){⟨name of macro⟩}

arguments	example	explanation
(point){name of macro}	\tkzGetPointCoord(A){A}	\Ax and \Ay give coordinates for A

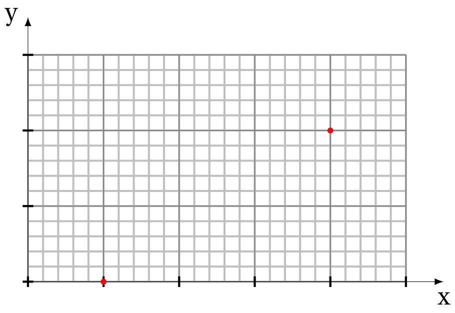
Stores in two macros the coordinates of a point. If the name of the macro is p*, then* \px *and* \py *give the coordinates of the chosen point with the cm as unit.*

\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

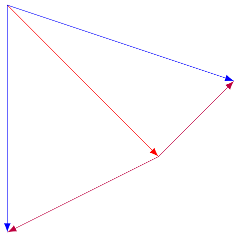
将点的坐标分量保存在两个宏中，如果宏名称是 p, 则将坐标分量保存在\px 和\py 宏中，单位是 cm。

36.5.1. Coordinate transfer with \tkzGetPointCoord



```
\begin{tikzpicture}
 \tkzInit[xmax=5,ymax=3]
 \tkzGrid[sub,orange]
 \tkzDrawX \tkzDrawY
 \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}
```

36.5.2. Sum of vectors with \tkzGetPointCoord 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[color=blue,->](a) -- ([shift=(b)]\cx,\cy) ;
  \draw[color=purple,->](b) -- ([shift=(b)]\cx,\cy) ;
  \tkzDrawSegment[->,blue](a,c)
  \tkzDrawSegment[->,purple](b,c)
\end{tikzpicture}
```

36.6. Swap labels of points 交换点的标签

\tkzSwapPoints(⟨pt1,pt2⟩)

arguments	example	explanation
(pt1,pt2)	\tkzSwapPoints(A,B)	now A has the coordinates of B

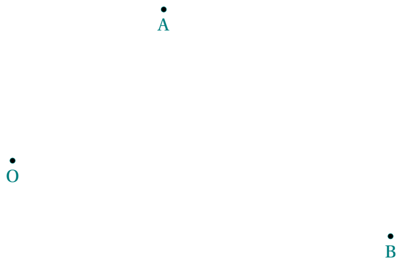
The points have exchanged their coordinates.

\tkzSwapPoints(⟨pt1,pt2⟩)

参数	示例	含义
(pt1,pt2)	\tkzSwapPoints(A,B)	现在 A 有 B 的坐标

The points have exchanged their coordinates.

36.6.1. Use of \tkzSwapPoints



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/O,5/-1/A,2/2/B}
   \tkzSwapPoints(A,B)
   \tkzDrawPoints(O,A,B)
   \tkzLabelPoints(O,A,B)
\end{tikzpicture}
```

36.7. Dot Product 点积

In Euclidean geometry, the dot product of the Cartesian coordinates of two vectors is widely used.
在欧几里得几何学中，两个向量的直角坐标的点积被广泛使用。

\tkzDotProduct(⟨pt1,pt2,pt3⟩)

The dot product of two vectors $\vec{u} = [a,b]$ and $\vec{v} = [a',b']$ is defined as: $\vec{u} \cdot \vec{v} = aa' + bb'$
$\vec{u} = \overrightarrow{pt1pt2}$ $\vec{v} = \overrightarrow{pt1pt3}$

arguments	example	explanation
(pt1,pt2,pt3)	\tkzDotProduct(A,B,C)	the result is $\overrightarrow{AB} \cdot \overrightarrow{AC}$

The result is a number that can be retrieved with ***\tkzGetResult****.*

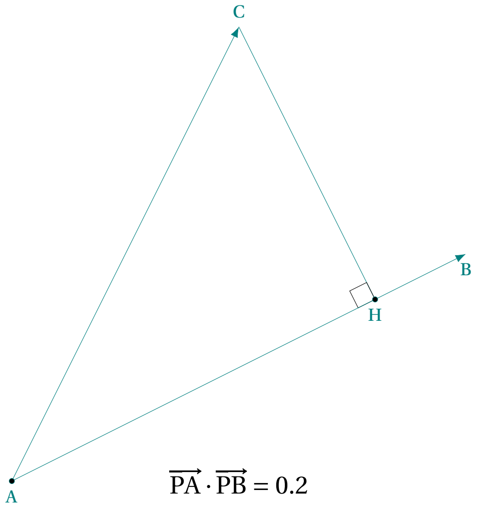
\tkzDotProduct(⟨pt1,pt2,pt3⟩)

两个向量 $\vec{u} = [a,b]$ 和 $\vec{v} = [a',b']$ 的点积定义为：$\vec{u} \cdot \vec{v} = aa' + bb'$
$\vec{u} = \overrightarrow{pt1pt2}$，$\vec{v} = \overrightarrow{pt1pt3}$

参数	示例	含义
(pt1,pt2,pt3)	\tkzDotProduct(A,B,C)	结果是 $\overrightarrow{AB} \cdot \overrightarrow{AC}$

结果是一个用***\tkzGetResult***保存的数字。

36.7.1. Simple example

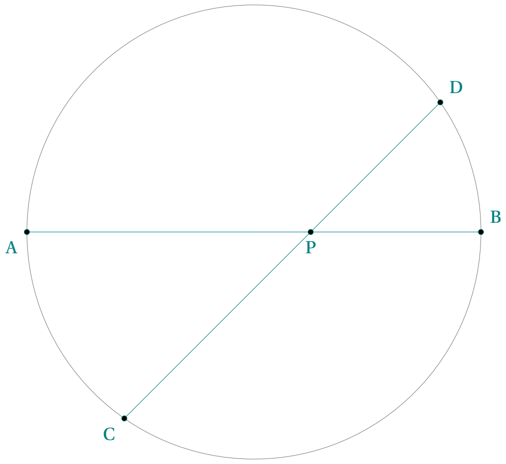


$$\overrightarrow{PA}\cdot\overrightarrow{PB}=0.2$$

$$PA\times PH=0.2$$

```
\begin{tikzpicture}
  \tkzDefPoints{-2/-3/A,4/0/B,1/3/C}
  \tkzDefPointBy[projection= onto A--B](C)
  \tkzGetPoint{H}
  \tkzDrawSegment(C,H)
  \tkzMarkRightAngle(C,H,A)
  \tkzDrawSegments[vector style](A,B A,C)
  \tkzDrawPoints(A,H) \tkzLabelPoints(A,B,H)
  \tkzLabelPoints[above](C)
  \tkzDotProduct(A,B,C) \tkzGetResult{pabc}
 % \pgfmathparse{round(10*\pabc)/10}
  \let\pabc\pgfmathresult
  \node at (1,-3) {$\overrightarrow{PA}\cdot \overrightarrow{PB}=\pa
  \tkzDotProduct(A,H,B) \tkzGetResult{phab}
 % \pgfmathparse{round(10*\phab)/10}
  \let\phab\pgfmathresult
  \node at (1,-4) {$PA \times PH = \phab $};
\end{tikzpicture}
```

36.7.2. Cocyclic points 共圆点



$$\overrightarrow{PA}\cdot\overrightarrow{PB}=\overrightarrow{PC}\cdot\overrightarrow{PD}$$

$$\overrightarrow{PA}\cdot\overrightarrow{PB}=-15.0$$

$$\overrightarrow{PC}\cdot\overrightarrow{PD}=-15.0$$

```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{1/2/O,5/2/B,2/2/P,3/3/Q}
  \tkzInterLC[common=B](O,B)(O,B) \tkzGetFirstPoint{A}
  \tkzInterLC[common=B](P,Q)(O,B) \tkzGetPoints{C}{D}
  \tkzDrawCircle(O,B)
  \tkzDrawSegments(A,B C,D)
  \tkzDrawPoints(A,B,C,D,P)
  \tkzLabelPoints(P)
  \tkzLabelPoints[below left](A,C)
  \tkzLabelPoints[above right](B,D)
  \tkzDotProduct(P,A,B) \tkzGetResult{pab}
  \pgfmathparse{round(10*\pab)/10}
  \let\pab\pgfmathresult
  \tkzDotProduct(P,C,D) \tkzGetResult{pcd}
  \pgfmathparse{round(10*\pcd)/10}
  \let\pcd\pgfmathresult
  \node at (1,-3) {%
  $\overrightarrow{PA}\cdot \overrightarrow{PB} =
   \overrightarrow{PC}\cdot \overrightarrow{PD}$};
    \node at (1,-4)%
    {$\overrightarrow{PA}\cdot \overrightarrow{PB} =\pab$};
 \node at (1,-5){%
 $\overrightarrow{PC}\cdot \overrightarrow{PD} =\pcd$};
\end{tikzpicture}
```

36.8. Power of a point with respect to a circle 点相对于圆的幂【圆幂】

\tkzPowerCircle(⟨pt1⟩)(⟨pt2,pt3⟩)

arguments	example	explanation
(pt1)(pt2,pt3)	\tkzPowerCircle(A)(O,M)	power of A with respect to the circle (O,A)

The result is a number that represents the power of a point with respect to a circle.

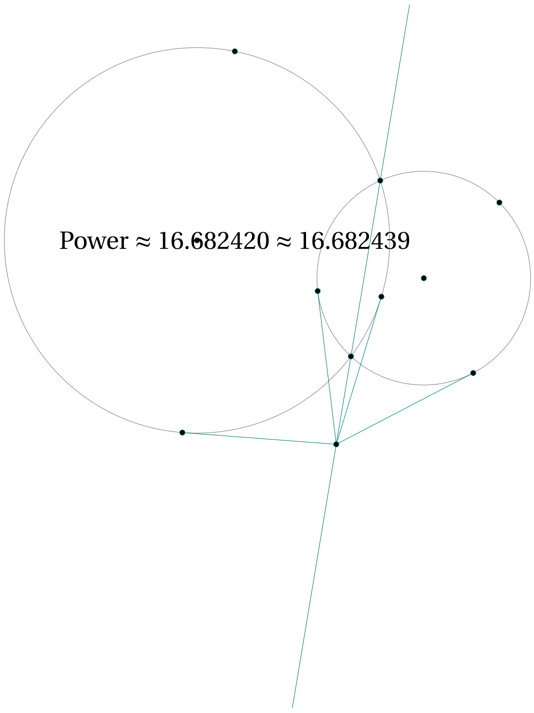
\tkzPowerCircle(⟨pt1⟩)(⟨pt2,pt3⟩)

参数	示例	含义
(pt1)(pt2,pt3)	\tkzPowerCircle(A)(O,M)	A 关于圆 (O, A) 的幂

结果是一个数字，表示一个点相对于圆的幂。

36.8.1. Power from the radical axis 根轴幂

In this example, the radical axis (EF) has been drawn. A point H has been chosen on (EF) and the power of the point H with respect to the circle of center A has been calculated as well as PS^2. You can check that the power of H with respect to the circle of center C as well as HS$'^2$, HT^2, HT$'^2$ give the same result.
在这个例子中，已经画出了根轴 (EF)。已经在 (EF) 上选择了点 H 并且已经计算了点 H 相对于圆心 A 的幂 PS^2. 可以检查 H 相对于圆心 C 以及 HS$'^2$、HT^2、HT$'^2$ 的幂给出相同的结果。



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{-1/0/A,0/5/B,5/-1/C,7/1/D}
  \tkzDrawCircles(A,B C,D)
  \tkzDefRadicalAxis(A,B)(C,D) \tkzGetPoints{E}{F}
  \tkzDrawLine[add=1 and 2](E,F)
  \tkzDefPointOnLine[pos=1.5](E,F) \tkzGetPoint{H}
  \tkzDefLine[tangent from = H](A,B)\tkzGetPoints{T}{T'}
  \tkzDefLine[tangent from = H](C,D)\tkzGetPoints{S}{S'}
  \tkzDrawSegments(H,T H,T' H,S H,S')
  \tkzDrawPoints(A,B,C,D,E,F,H,T,T',S,S')
  \tkzPowerCircle(H)(A,B) \tkzGetResult{pw}
  \tkzDotProduct(H,S,S) \tkzGetResult{phtt}
  \node {Power $\approx \pw \approx \phtt$};
\end{tikzpicture}
```

36.9. Radical axis 根轴

In geometry, the radical axis of two non-concentric circles is the set of points whose power with respect to the circles are equal. Here \tkzDefRadicalAxis(A,B)(C,D) gives the radical axis of the two circles $\mathscr{C}$(A,B) and $\mathscr{C}$(C,D).

在几何中，两个不同心的圆的根轴是它们相对于圆的幂相等的点的集合。这里`\tkzDefRadicalAxis(A,B)(C,D)`给出两个圆 $\mathscr{C}(A,B)$ 和 $\mathscr{C}(C,D)$ 的根轴。

\tkzDefRadicalAxis(⟨pt1,pt2⟩)(⟨pt3,pt4⟩)

arguments	example	explanation
(pt1,pt2)(pt3,pt4)	\tkzDefRadicalAxis(A,B)(C,D)	Two circles with centers A and C

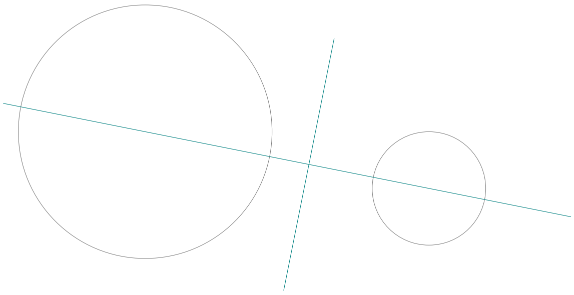
The result is two points of the radical axis.

\tkzDefRadicalAxis(⟨pt1,pt2⟩)(⟨pt3,pt4⟩)

参数	示例	含义
(pt1,pt2)(pt3,pt4)	\tkzDefRadicalAxis(A,B)(C,D)	两个圆心为 A 和 C 的圆

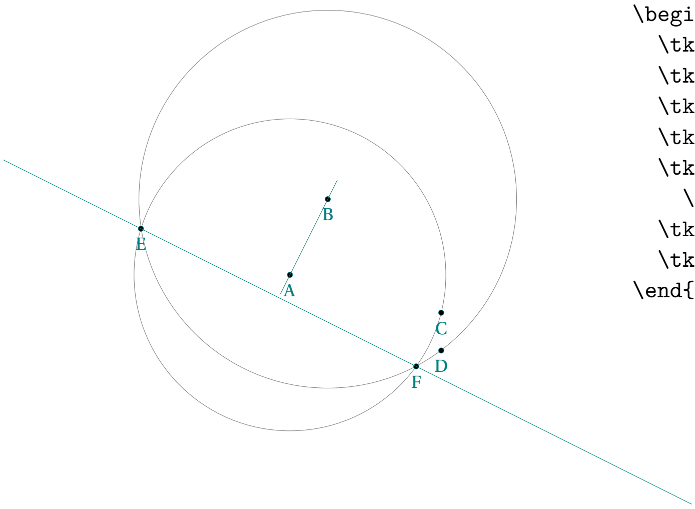
结果是根轴的两个点。

36.9.1. Two circles disjointed 两个圆相离



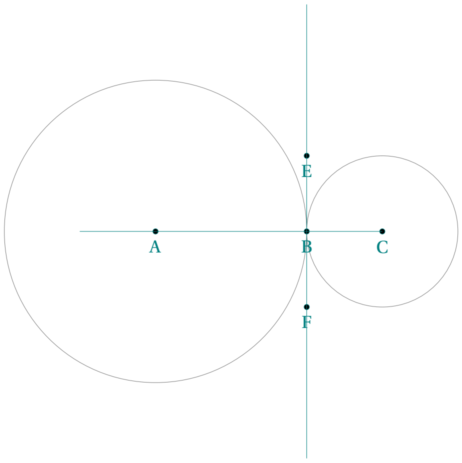
```
\begin{tikzpicture}[scale=.75]
    \tkzDefPoints{-1/0/A,0/2/B,4/-1/C,4/0/D}
    \tkzDrawCircles(A,B C,D)
    \tkzDefRadicalAxis(A,B)(C,D)
     \tkzGetPoints{E}{F}
    \tkzDrawLine[add=1 and 2](E,F)
    \tkzDrawLine[add=.5 and .5](A,C)
\end{tikzpicture}
```

36.10. Two intersecting circles 相交圆



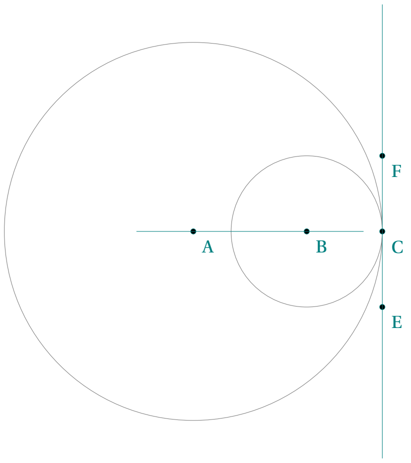
```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{-1/0/A,0/2/B,3/-1/C,3/-2/D}
  \tkzDrawCircles(A,C B,D)
  \tkzDefRadicalAxis(A,C)(B,D)
  \tkzGetPoints{E}{F}
  \tkzDrawPoints(A,B,C,D,E,F)
    \tkzLabelPoints(A,B,C,D,E,F)
  \tkzDrawLine[add=.5 and 1](E,F)
  \tkzDrawLine[add=.25 and .25](A,B)
\end{tikzpicture}
```

36.11. Two externally tangent circles 外切圆



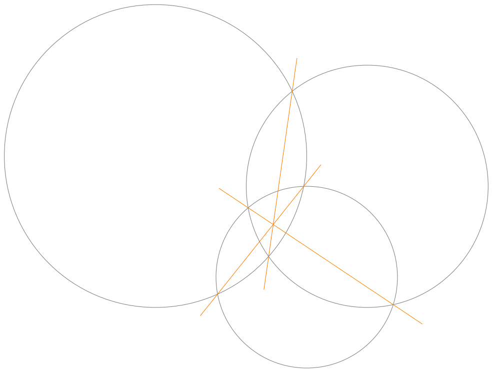
```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/A,4/0/B,6/0/C}
  \tkzDrawCircles(A,B C,B)
  \tkzDefRadicalAxis(A,B)(C,B)
  \tkzGetPoints{E}{F}
  \tkzDrawPoints(A,B,C,E,F)
    \tkzLabelPoints(A,B,C,E,F)
  \tkzDrawLine[add=1 and 1](E,F)
  \tkzDrawLine[add=.5 and .5](A,B)
\end{tikzpicture}
```

36.12. Two circles tangent internally 内切圆



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/A,3/0/B,5/0/C}
  \tkzDrawCircles(A,C B,C)
  \tkzDefRadicalAxis(A,C)(B,C)
  \tkzGetPoints{E}{F}
  \tkzDrawPoints(A,B,C,E,F)
  \tkzLabelPoints[below right](A,B,C,E,F)
  \tkzDrawLine[add=1 and 1](E,F)
  \tkzDrawLine[add=.5 and .5](A,B)
\end{tikzpicture}
```

36.12.1. Three circles 三圆相交



```
\begin{tikzpicture}[scale=.4]
  \tkzDefPoints{0/0/A,5/0/a,7/-1/B,3/-1/b,5/-
4/C,2/-4/c}
  \tkzDrawCircles(A,a B,b C,c)
  \tkzDefRadicalAxis(A,a)(B,b) \tkzGetPoints{i}{j}
  \tkzDefRadicalAxis(A,a)(C,c) \tkzGetPoints{k}{l}
  \tkzDefRadicalAxis(C,c)(B,b) \tkzGetPoints{m}{n}
  \tkzDrawLines[new](i,j k,l m,n)
\end{tikzpicture}
```

36.13. \tkzIsLinear, \tkzIsOrtho

\tkzIsLinear(⟨pt1,pt2,pt3⟩)

arguments	example	explanation
(pt1,pt2,pt3)	\tkzIsLinear(A,B,C)	A,B,C aligned ?

\tkzIsLinear *allows to test the alignment of the three points* pt1,pt2,pt3.

\tkzIsLinear(⟨pt1,pt2,pt3⟩)

参数	示例	含义
(pt1,pt2,pt3)	\tkzIsLinear(A,B,C)	A,B,C aligned ?

\tkzIsLinear 允许验证三个点 pt1 、pt2 、pt3 的共线情况。

\tkzIsOrtho(⟨pt1,pt2,pt3⟩)

arguments	example	explanation
(pt1,pt2,pt3)	\tkzIsOrtho(A,B,C)	(AB) ⊥ (AC) ?

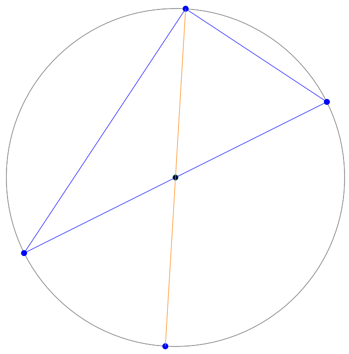
\tkzIsOrtho *allows to test the orthogonality of lines* (pt1pt2) *and* (pt1pt3).

\tkzIsOrtho(⟨pt1,pt2,pt3⟩)

参数	示例	含义
(pt1,pt2,pt3)	\tkzIsOrtho(A,B,C)	(AB) ⊥ (AC) ?

\tkzIsOrtho 验证两线 (pt1pt2) 和 (pt1pt3) 垂直的情况.

36.13.1. Use of \tkzIsOrtho and \tkzIsLinear



```
  \begin{tikzpicture}
  \tkzDefPoints{1/-2/A,5/0/B}
  \tkzDefCircle[diameter](A,B) \tkzGetPoint{O}
  \tkzDrawCircle(O,A)
  \tkzDefPointBy[rotation= center O angle 60](B)
  \tkzGetPoint{C}
  \tkzDefPointBy[rotation= center O angle 60](A)
  \tkzGetPoint{D}
  \tkzDrawCircle(O,A)
  \tkzDrawPoints(A,B,C,D,O)
  \tkzIsOrtho(C,A,B)
  \iftkzOrtho
    \tkzDrawPolygon[blue](A,B,C)
  \tkzDrawPoints[blue](A,B,C,D)
  \else
  \tkzDrawPoints[red](A,B,C,D)
  \fi
   \tkzIsLinear(O,C,D)
    \iftkzLinear
    \tkzDrawSegment[orange](C,D)
    \fi
\end{tikzpicture}
```

Part VIII.

Working with style 样式

37. Predefined styles 预定义样式

The way to proceed will depend on your use of the package. A method that seems to me to be correct is to use as much as possible predefined styles in order to separate the content from the form. This method will be the right one if you plan to create a document (like this documentation) with many figures. We will see how to define a global style for a document. We will see how to use a style locally.
如何进行将取决于您对该软件包的使用。在我看来，一个正确的方法是尽可能多地使用预定义的样式，以便将内容与形式分开。如果打算创建一个包含许多图形的文档 (如本文档), 这种方法将是正确的。下面介绍如何为一个文档定义一个全局样式。如何在本地使用一个样式。
The file `tkz-euclide.cfg` contains the predefined styles of the main objects. Among these the most important are points, lines, segments, circles, arcs and compass traces. If you always use the same styles and if you create many figures then it is interesting to create your own styles . To do this you need to know what features you can modify. It will be necessary to know some notions of Ti*k*Z.
文件 `tkz-euclide.cfg` 包含主对象的预定义样式。其中最重要的是点、线、线段、圆、弧和圆规轨迹。如果总是使用相同的样式，如果创造了许多图像，那么创造自己的样式是很有趣的。为此，需要知道可以修改哪些特征。有必要了解一些关于 Ti*k*Z 的概念。
The predefined styles are global styles. They exist before the creation of the figures. It is better to avoid changing them between two figures. On the other hand these styles can be modified in a figure temporarily. There the styles are defined locally and do not influence the other figures.
预定义样式是全局样式。它们在图像被创造出来之前就存在了。最好避免在两个图形之间改变它们。另一方面，这些样式可以在图形中临时修改。这些样式是在本地定义的，不会影响其他图形。
For the document you are reading here is how I defined the different styles.
对于正在阅读的文件，这里是我如何定义不同的样式。

```
\tkzSetUpColors[background=white,text=black]
\tkzSetUpPoint[size=2,color=teal]
\tkzSetUpLine[line width=.4pt,color=teal]
\tkzSetUpCompass[color=orange, line width=.4pt,delta=10]
\tkzSetUpArc[color=gray,line width=.4pt]
\tkzSetUpStyle[orange]{new}
```

The macro `\tkzSetUpColors` allows you to set the background color as well as the text color. If you don't use it, the colors of your document will be used as well as the fonts. Let's see how to define the styles of the main objects.
宏指令 `\tkzSetUpColors` 允许设置背景颜色，以及文本颜色。如果不使用它，文件的颜色和字体都将使用默认设置。看看如何定义主要对象的样式。

38. Points style 点的样式

This is how the points are defined :
这是对点的样式的定义：

```
\tikzset{point style/.style = {%
      draw         = \tkz@euc@pointcolor,
      inner sep    = 0pt,
      shape        = \tkz@euc@pointshape,
      minimum size = \tkz@euc@pointsize,
      fill         = \tkz@euc@pointcolor}}
```

It is of course possible to use `\tikzset` but you can use a macro provided by the package. You can use the macro `\tkzSetUpPoint` globally or locally,
Let's look at this possibility.
当然可以使用 `\tikzset`，但也可以使用软件包提供的宏。可以在全局或局部使用宏 `\tkzSetUpPoint`，看看这种可能性。

38.1. Use of \tkzSetUpPoint 设置点样式

```
\tkzSetUpPoint[⟨local options⟩]
```

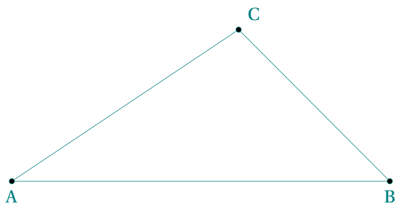
options	default	definition
color	black	point color
size	3	point size
fill	black!50	inside point color
shape	circle	point shape circle, cross or cross out

```
\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

38.1.1. Global style or local style

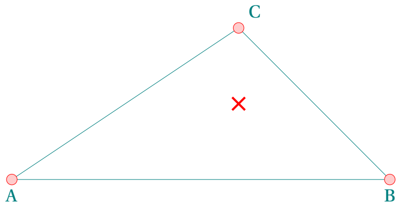
First of all here is a figure created with the styles of my documentation, then the style of the points is modified within the environment `tikzspicture`.
首先，这是一个用我的文档中的样式创建的图，然后在环境中修改点的样式 `tikzspicture`。
You can use the macro `\tkzSetUpPoint` globally or locally, If you place this macro in your preamble or before your first figure then the point style will be valid for all figures in your document. It will be possible to use another style locally by using this command within an environment `tikzpicture`.
Let's look at this possibility.
可以在全局或局部使用宏 `\tkzSetUpPoint`，如果把这个宏放在序言或第一个图之前，那么这个点样式将对文件中的所有图都有效。通过在环境中使用这个命令 `tikzpicture` 可以在本地使用另一种样式。让我们看看这种可能性。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,5/0/B,3/2/C,3/1/D}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above right](C)
\end{tikzpicture}
```

38.1.2. Local style 本地样式

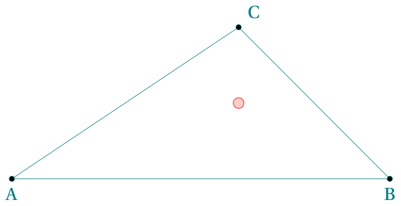
The style of the points is modified locally in the second figure
在第二个图中，点的样式被局部修改了



```
\begin{tikzpicture}
  \tkzSetUpPoint[size=4,color=red,fill=red!20]
  \tkzDefPoints{0/0/A,5/0/B,3/2/C,3/1/D}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
  \tkzDrawPoint[shape=cross out,thick](D)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above right](C)
\end{tikzpicture}
```

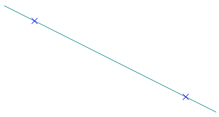
38.1.3. Style and scope

The points get back the initial style. Point D has a new style limited by the environment `scope`. It is also possible to use `{...}` or The points get back the initial style. Point D has a new style limited by the environment `scope`. It is also possible to use `{...}` or `\begingroup ... \endgroup`.
这些点恢复到初始样式。点 D 具有受环境限制的新样式 `scope`。也可以使用 `{...}` 或者点恢复到初始样式。点 D 具有受环境限制的新样式 `scope`。也可以使用 `{...}` 或 `\begingroup ... \endgroup`。



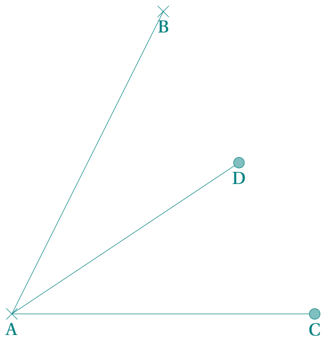
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,5/0/B,3/2/C,3/1/D}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
  \begin{scope}
    \tkzSetUpPoint[size=4,color=red,fill=red!20]
    \tkzDrawPoint(D)
  \end{scope}
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above right](C)
\end{tikzpicture}
```

38.1.4. Simple example with \tkzSetUpPoint



```
\begin{tikzpicture}
  \tkzSetUpPoint[shape = cross out,color=blue]
  \tkzDefPoint(2,1){A}
  \tkzDefPoint(4,0){B}
  \tkzDrawLine(A,B)
  \tkzDrawPoints(A,B)
\end{tikzpicture}
```

38.1.5. Use of \tkzSetUpPoint inside a group



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,2/4/B,4/0/C,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}
```

39. Lines style 线的样式

You have several possibilities to change the style of a line. You can modify the style of a line with `\tkzSetUpLine` or directly modify the style of the lines with `\tikzset{line style/.style = ... }`
有几种可能方法来改变线的样式。可以用 `\tkzSetUpLine` 来修改线的样式，或者直接用 `\tikzset{line style/.style = ..`
修改线的样式。
Reminder about `line width` : There are a number of predefined styles that provide more “natural” ways of setting the line width. You can also redefine these styles.
关于 `line width` 的提醒：有一些预定义的样式提供了更”自然”的线宽设置方式。也可以重新定义这些样式。

predefined style	value of `line width`
ultra thin	0.1 pt
very thin	0.2 pt
thin	0.4 pt
semithick	0.6 pt
thick	0.8 pt
very thick	1.2 pt
ultra thick	1.6 pt

预定义样式	`line width` 线宽
ultra thin 极细	0.1 pt
very thin 很细	0.2 pt
thin 细	0.4 pt
semithick 半粗	0.6 pt
thick 粗	0.8 pt
very thick 很粗	1.2 pt
ultra thick 极粗	1.6 pt

39.1. Use of \tkzSetUpLine

It is a macro that allows you to define the style of all the lines.
这个宏，允许定义所有线条的样式。

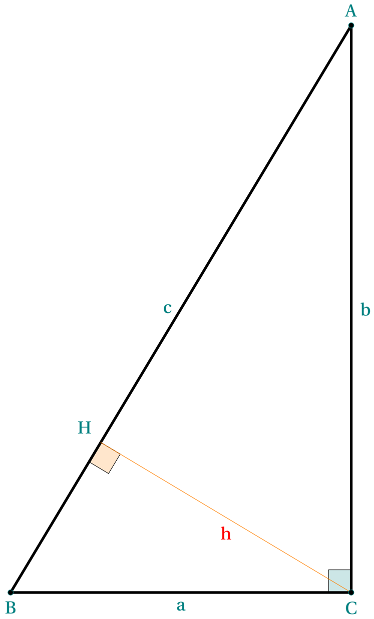
`\tkzSetUpLine[⟨local options⟩]`

options	default	definition
color	black	colour of the construction lines
line width	0.4pt	thickness of the construction lines
style	solid	style of construction lines
add	.2 and .2	changing the length of a line segment

`\tkzSetUpLine[⟨local options⟩]`

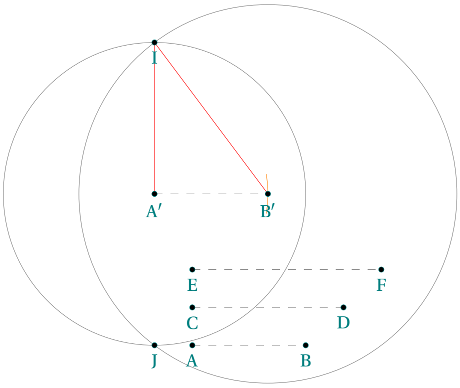
选项	默认值	含义
颜色	黑色	默认线的颜色
线宽	0.4pt	线宽
样式	solid	线的样式
延长	.2 和 .2	改变线段的长度

39.1.1. Change line width 改变线宽



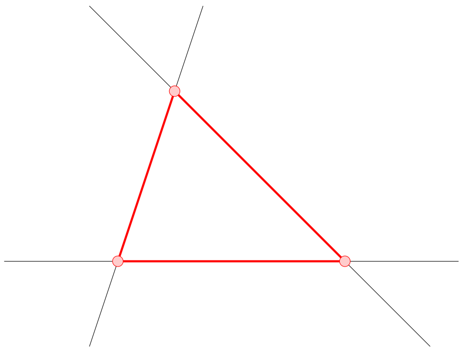
```
\begin{tikzpicture}[scale=.75]
\tkzSetUpLine[line width=1pt]
\begin{scope}[rotate=-90]
    \tkzDefPoints{0/6/A,10/0/B,10/6/C}
    \tkzDefPointBy[projection = onto B--A](C)
    \tkzGetPoint{H}
    \tkzMarkRightAngle[size=.4,
                       fill=teal!20](B,C,A)
    \tkzMarkRightAngle[size=.4,
                       fill=orange!20](B,H,C)
    \tkzDrawPolygon(A,B,C)
    \tkzDrawSegment[new](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}
```

39.1.2. Change style of line 改变线的样式



```
\begin{tikzpicture}[scale=.5]
\tikzset{line style/.style = {color = gray,
                              style=dashed}}
\tkzDefPoints{1/0/A,4/0/B,1/1/C,5/1/D}
\tkzDefPoints{1/2/E,6/2/F,0/4/A',3/4/B'}
\tkzCalcLength(C,D)
\tkzGetLength{rCD}
\tkzCalcLength(E,F)
\tkzGetLength{rEF}
\tkzInterCC[R](A',\rCD)(B',\rEF)
\tkzGetPoints{I}{J}
\tkzDrawLine(A',B')
\tkzCompass(A',B')
\tkzDrawSegments(A,B C,D E,F)
\tkzDefCircle[R](A',\rCD) \tkzGetPoint{a'}
\tkzDefCircle[R](B',\rEF)\tkzGetPoint{b'}
\tkzDrawCircles(A',a' B',b')
\begin{scope}
  \tkzSetUpLine[color=red]
  \tkzDrawSegments(A',I B',I)
\end{scope}
\tkzDrawPoints(A,B,C,D,E,F,A',B',I,J)
\tkzLabelPoints(A,B,C,D,E,F,A',B',I,J)
\end{tikzpicture}
```

39.1.3. Example 3: extend lines 延长线的长度



```
\begin{tikzpicture}[scale=.75]
\tkzSetUpLine[add=.5 and .5]
\tkzDefPoints{0/0/A,4/0/B,1/3/C}
\tkzDrawLines(A,B B,C A,C)
\tkzDrawPolygon[red,thick](A,B,C)
\tkzSetUpPoint[size=4,circle,color=red,fill=red!20]
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

40. Arc style 弧的样式

40.1. The macro \tkzSetUpArc

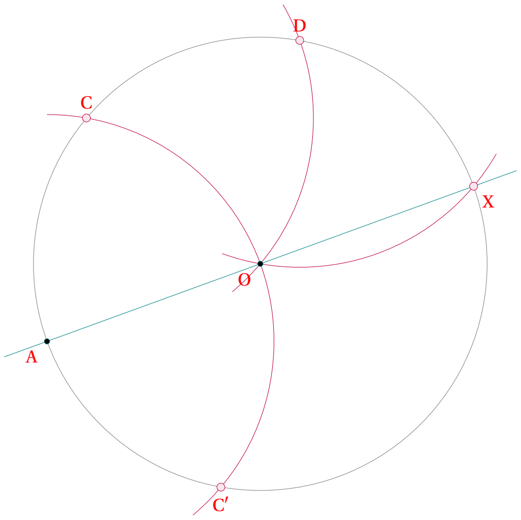
\tkzSetUpArc[⟨local options⟩]

options	default	definition
color	black	colour of the lines
line width	0.4pt	thickness of the lines
style	solid	style of construction lines

\tkzSetUpArc[⟨local options⟩]

选项	默认值	含义
颜色	黑色	线的颜色
线宽	0.4pt	线的宽度
样式	实线	默认样式

40.1.1. Use of \tkzSetUpArc



```
\begin{tikzpicture}
\def\r{3} \def\angle{200}
\tkzSetUpArc[delta=10,color=purple,line width=.2pt]
\tkzSetUpLabel[font=\scriptsize,red]
\tkzDefPoint(0,0){O}
\tkzDefPoint(\angle:\r){A}
\tkzInterCC(O,A)(A,O) \tkzGetPoints{C'}{C}
\tkzInterCC(O,A)(C,O) \tkzGetPoints{D'}{D}
\tkzInterCC(O,A)(D,O) \tkzGetPoints{X'}{X}
\tkzDrawCircle(O,A)
\tkzDrawArc(A,C')(C)
\tkzDrawArc(C,O)(D)
\tkzDrawArc(D,O)(X)
\tkzDrawLine[add=.1 and .1](A,X)
\tkzDrawPoints(O,A)
\tkzSetUpPoint[size=3,color=purple,fill=purple!10]
\tkzDrawPoints(C,C',D,X)
\tkzLabelPoints[below left](O,A)
\tkzLabelPoints[below](C')
\tkzLabelPoints[below right](X)
\tkzLabelPoints[above](C,D)
\end{tikzpicture}
```

41. Compass style, configuration macro \tkzSetUpCompass 设置尺规标记样式

The following macro will help to understand the construction of a figure by showing the compass traces necessary to obtain certain points.
下面的宏图将有助于理解图形的构造，显示获得某些点所需的圆规痕迹。

41.1. The macro \tkzSetUpCompass

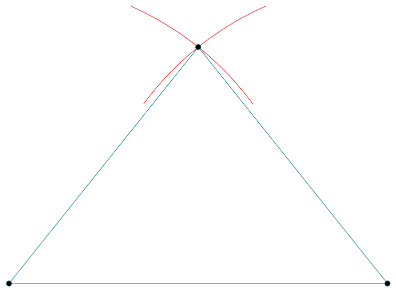
\tkzSetUpCompass[⟨local options⟩]

options	default	definition
color	black	colour of the construction lines
line width	0.4pt	thickness of the construction lines
style	solid	style of lines : solid, dashed,dotted,...
delta	0	changes the length of the arc

\tkzSetUpCompass[⟨命令选项⟩]

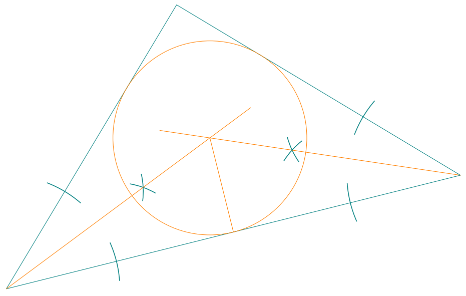
选项	默认值	含义
颜色	黑色	圆弧颜色
线宽	0.4pt	圆弧线宽
样式	实线	圆弧线型：实线，虚线，点线，...
delta	0	改变弧线的长度

41.1.1. Use of \tkzSetUpCompass



```
\begin{tikzpicture}
  \tkzSetUpCompass[color=red,delta=15]
  \tkzDefPoint(1,1){A}
  \tkzDefPoint(6,1){B}
  \tkzInterCC[R](A,4)(B,4) \tkzGetPoints{C}{D}
  \tkzCompass(A,C)
  \tkzCompass(B,C)
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

41.1.2. Use of \tkzSetUpCompass with \tkzShowLine



```
\begin{tikzpicture}[scale=.75]
\tkzSetUpStyle[bisector,size=2,gap=3]{showbi}
\tkzSetUpCompass[color=teal,line width=.3 pt]
\tkzDefPoints{0/1/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[new](I,H)
\tkzDrawSegments[new](I,H)
\tkzDrawLines[add=0 and .2,new](A,I B,I)
\end{tikzpicture}
```

42. Label style 标签样式

42.1. The macro \tkzSetUpLabel

The macro `\tkzSetUpLabel` is used to define the style of the point labels.
宏指令 `\tkzSetUpLabel` 用于定义点标签的样式。

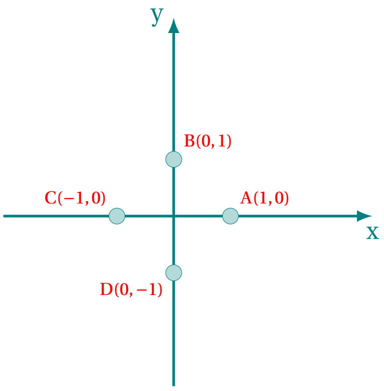
`\tkzSetUpStyle[⟨local options⟩]`

The options are the same as those of Ti*k*Z

`\tkzSetUpStyle[⟨local options⟩]`

选项与 Ti*k*Z 的选项相同

42.1.1. Use of \tkzSetUpLabel



```
\begin{tikzpicture}[scale=.75]
  \tkzSetUpLabel[font=\scriptsize,red]
  \tkzSetUpStyle[line width=1pt,teal]{XY}
  \tkzInit[xmin=-3,xmax=3,ymin=-3,ymax=3]
  \tkzDrawX[noticks,XY]
  \tkzDrawY[noticks,XY]
  \tkzDefPoints{1/0/A,0/1/B,-1/0/C,0/-1/D}
  \tkzDrawPoints[teal,fill=teal!30,size=6](A,...,D)
  \tkzLabelPoint[above right](A){$A(1,0)$}
  \tkzLabelPoint[above right](B){$B(0,1)$}
  \tkzLabelPoint[above left](C){$C(-1,0)$}
  \tkzLabelPoint[below left](D){$D(0,-1)$}
\end{tikzpicture}
```

43. Own style 自身风格

You can set your own style with \tkzSetUpStyle
可以用 \tkzSetUpStyle 设置自己的风格。

43.1. The macro \tkzSetUpStyle

\tkzSetUpStyle[⟨local options⟩]

The options are the same as those of Ti*k*Z

\tkzSetUpStyle[⟨local options⟩]

选项与 Ti*k*Z 的选项相同

43.1.1. Use of \tkzSetUpStyle

```
\begin{tikzpicture}
  \tkzSetUpStyle[color=blue!20!black,fill=blue!20]{mystyle}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(0,1){A}
  \tkzDrawPoints(O) % general style
  \tkzDrawPoints[mystyle,size=4](A) % my style
  \tkzLabelPoints(O,A)
\end{tikzpicture}
```

44. How to use arrows 如何使用箭头

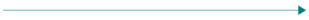
In some countries, arrows are used to indicate the parallelism of lines, to represent half-lines or the sides of an angle (rays).
在一些国家，箭头被用来表示线条的平行性、表示射线或一个角的边（射线）。
Here are some examples of how to place these arrows. tkz-euclide loads a library called arrows.meta.
下面是一些关于如何放置这些箭头的例子。tkz-euclide 加载一个名为 arrows.meta 的库。
\usetikzlibrary{arrows.meta}
This library is used to produce different styles of arrow heads. The next examples use some of them.
这个库被用来制作不同风格的箭头。下面的例子使用了其中的一些。

44.1. Arrows at endpoints on segment, ray or line 箭头位于线段、射线或直线的端点上

Stealth, **Triangle**, **To**, **Latex** and …which can be combined with **reversed**. That's easy to place an arrow at one or two endpoints.

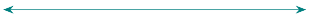
Stealth、**Triangle**、**To**、**Latex**、…可以与 tkznamereversed 组合。这很容易在一个或两个端点上放置一个箭头。

1. Triangle and Ray



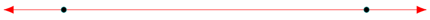
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[-Triangle](A,B)
\end{tikzpicture}
```

2. Stealth and Segment



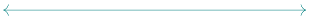
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[Stealth-Stealth](A,B)
\end{tikzpicture}
```

3. Latex and Line



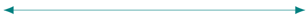
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawLine[red,Latex-Latex](A,B)
  \tkzDrawPoints(A,B)
\end{tikzpicture}
```

4. To and Segment



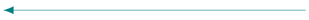
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[To-To](A,B)
\end{tikzpicture}
```

5. Latex and Segment



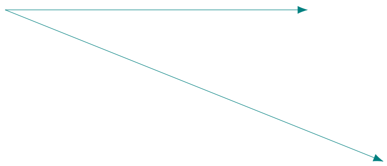
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[Latex-Latex](A,B)
\end{tikzpicture}
```

6. Latex and Ray



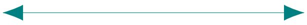
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[Latex-](A,B)
\end{tikzpicture}
```

7. Latex and Several rays



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B,5/-2/C}
 \tkzDrawSegments[-Latex](A,B A,C)
\end{tikzpicture}
```

44.1.1. Scaling an arrow head 缩放箭头



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B}
 \tkzDrawSegment[{Latex[scale=2]}-{Latex[scale=2]}](A,B)
\end{tikzpicture}
```

44.1.2. Using vector style 使用矢量样式

`\tikzset{vector style/.style={>=Latex,->}}`
You can redefine this style.

```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B}
\tkzDrawSegment[vector style](A,B)
\end{tikzpicture}
```

44.2. Arrows on middle point of a line segment 线段中间点上的箭头

Arrows on lines are used to indicate that those lines are parallel. It depends on the country, in France we prefer to indicate outside the figure that $(A,B)\parallel(D,C)$. The code is an adaptation of an answer by `muzimuzhi Z` on the site tex.stackexchange.com.
线条上的箭头用来表示这些线是平行的。这取决于国家习惯，在法国更喜欢在图外表示 $(A,B)\parallel(D,C)$。这个代码是由网站 tex.stackexchange.com. 上的 `muzimuzhi Z` 的答案改编的。

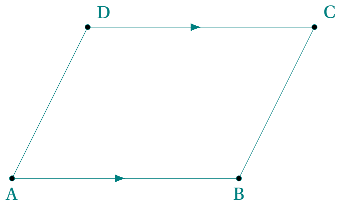
Syntax 语法:

- `tkz arrow` (`Latex` by default)
- `tkz arrow=<arrow end tip>`
- `tkz arrow=<arrow end tip> at <pos> (<pos> = .5 by default)`
- `tkz arrow={<arrow end tip>[<arrow options>] at <pos>}` option possible `scale`

Example usages 使用示例:

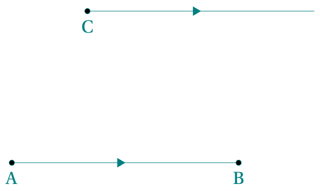
```
\tkzDrawSegment[tkz arrow=Stealth] (A,B)
\tkzDrawSegment[tkz arrow={To[scale=3] at .4}](A,B)
\tkzDrawSegment[tkz arrow={Latex[scale=5,blue] at .6}](A,B)
```

44.2.1. In a parallelogram 在平行四边形中



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

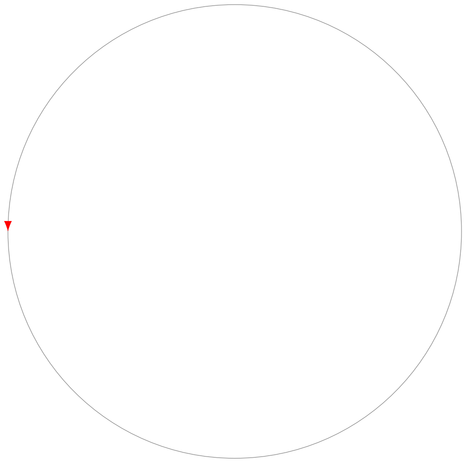
44.2.2. A line parallel to another one 平行于另一条直线的线



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,3/0/B,1/2/C}
 \tkzDefPointWith[colinear= at C](A,B)
 \tkzGetPoint{D}
 \tkzDrawSegments[tkz arrow=Triangle](A,B C,D)
 \tkzLabelPoints(A,B,C)
 \tkzDrawPoints(A,...,C)
\end{tikzpicture}
```

44.2.3. Arrow on a circle 圆上的箭头

It is possible to place an arrow on the first quarter of a circle. A rotation allows you to move the arrow.
可以在圆的四分之一处放置一个箭头。旋转可以移动箭头。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,3/0/B}
\begin{scope}[rotate=150]
 \tkzDrawCircle[tkz arrow={Latex[scale=2,red]}](A,B)
\end{scope}
\end{tikzpicture}
```

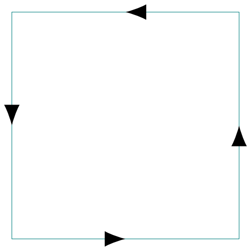
44.3. Arrows on all segments of a polygon 多边形所有线段上的箭头

Some users of my package have asked me to be able to place an arrow on each side of a polygon. I used a style proposed by Paul Gaborit on the site tex.stackexchange.com.
我的软件包的一些用户要求我能够在多边形的每一侧放置一个箭头。我使用了 Paul Gaborit 在网站 tex.stackexchange.com 上提出的一种样式

```
\tikzset{tkz arrows/.style=
{postaction={on each path={tkz arrow={Latex[scale=2,color=black]}}}}}
```

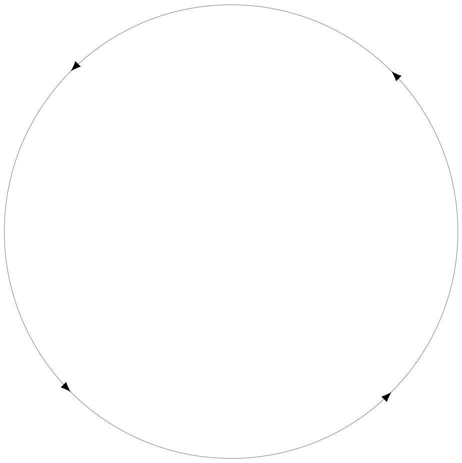
You can change this style. With `tkz arrows` you can an arrow on each segment of a polygon
可以改变这种风格。用 `tkz arrows`，可以在多边形的每一段上设置一个箭头。

44.3.1. Arrow on each segment with tkz arrows 多边形每条边上使用箭头



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,3/0/B}
 \tkzDefSquare(A,B) \tkzGetPoints{C}{D}
 \tkzDrawPolygon[tkz arrows](A,...,D)
\end{tikzpicture}
```

44.3.2. Using tkz arrows with a circle 圆上使用箭头



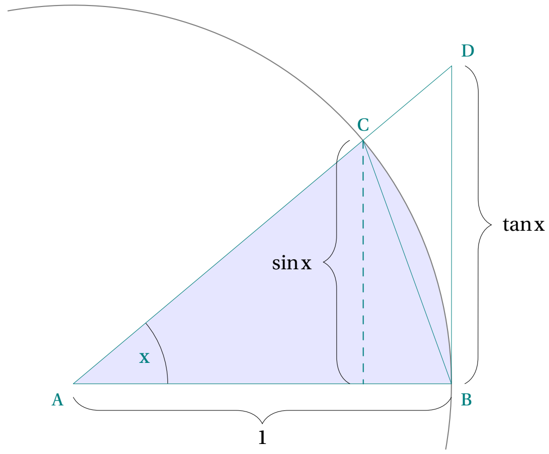
```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,3/0/B}
 \tkzDrawCircle[tkz arrows](A,B)
\end{tikzpicture}
```

Part IX.

Examples 示例

45. Different authors 不同作者

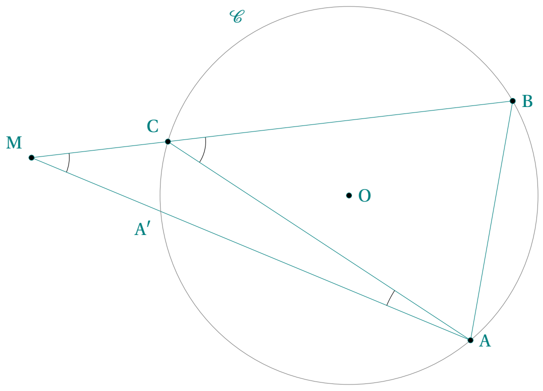
45.1. Code from Andrew Swan



```
\begin{tikzpicture}[scale=1.25]
\def\radius{4}
\def\angle{40}
\pgfmathsetmacro{\htan}{tan(\angle)}
\tkzDefPoint(0,0){A} \tkzDefPoint(0,\radius){F}
\tkzDefPoint(\radius,0){B}
\tkzDefPointBy[rotation= center A angle \angle](B)
\tkzGetPoint{C}
\tkzDefLine[perpendicular=through B,K=1](A,B)
\tkzGetPoint{b}
\tkzInterLL(A,C)(B,b) \tkzGetPoint{D}
\tkzDefLine[perpendicular=through C,K=-1](A,B)
\tkzGetPoint{c}
\tkzInterLL(C,c)(A,B) \tkzGetPoint{E}
\tkzDrawSector[fill=blue,opacity=0.1](A,B)(C)
\tkzDrawArc[thin](A,B)(F)
\tkzMarkAngle(B,A,C)
\tkzLabelAngle[pos=0.8](B,A,C){$x$}
\tkzDrawPolygon(A,B,D)
\tkzDrawSegments(C,B)
\tkzDrawSegments[dashed,thin](C,E)
\tkzLabelPoints[below left](A)
\tkzLabelPoints[below right](B)
\tkzLabelPoints[above](C)
\tkzLabelPoints[above right](D)
\begin{scope}[/pgf/decoration/raise=5pt]
\draw [decorate,decoration={brace,mirror,
   amplitude=10pt},xshift=0pt,yshift=-4pt]
(A) -- (B) node [black,midway,yshift=-20pt]
{\footnotesize $1$};
\draw [decorate,decoration={brace,amplitude=10pt},
       xshift=4pt,yshift=0pt]
(D) -- (B) node [black,midway,xshift=27pt]
{\footnotesize $\tan x$};
\draw [decorate,decoration={brace,amplitude=10pt},
       xshift=4pt,yshift=0pt]
(E) -- (C) node [black,midway,xshift=-27pt]
{\footnotesize $\sin x$};
\end{scope}
\end{tikzpicture}
```

45.2. Example: Dimitris Kapeta

You need in this example to use `mkpos=.2` with `\tkzMarkAngle` because the measure of $\widehat{CAM}$ is too small. Another possiblity is to use `\tkzFillAngle`.
本例中`\tkzMarkAngle` 要用参数 `mkpos=.2`， 因为 角度 $\widehat{CAM}$ 的测量值较小。



```
\begin{tikzpicture}[scale=1]
  \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[common=B](M,B)(O,B) \tkzGetFirstPoint{C}
  \tkzInterLC[common=A](M,A)(O,A) \tkzGetFirstPoint{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 A,B)
  \tkzDrawCircle(O,N)
  \tkzLabelCircle[above left](O,N)(120){%
                $\mathcal{C}$}
  \begin{scope}[veclen]
  \tkzMarkAngle[mkpos=.2, size=1.2](C,A,M)
  \end{scope}
  \tkzDrawPoints(O, A, B, M, B, C)
  \tkzLabelPoints[right](O,A,B)
  \tkzLabelPoints[above left](M,C)
  \tkzLabelPoint[below left](A'){$A'$}
\end{tikzpicture}
```

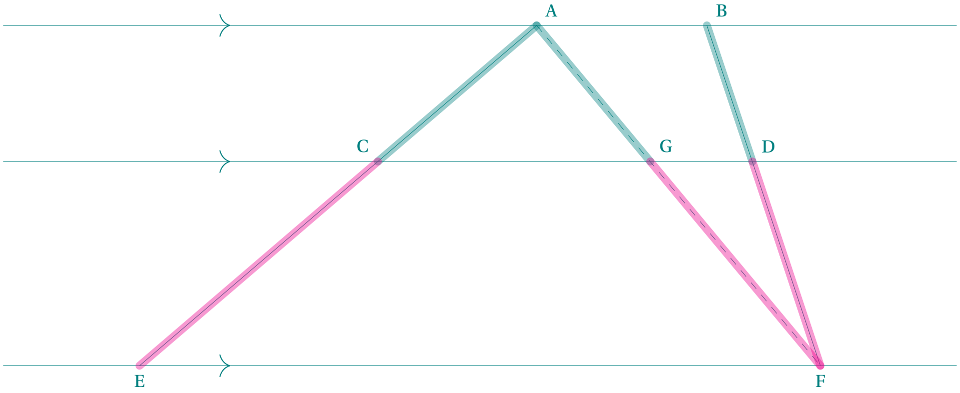
45.3. Example : John Kitzmiller

Prove that $\frac{\mathrm{AC}}{\mathrm{CE}} = \frac{\mathrm{BD}}{\mathrm{DF}}$.

证明 $\frac{\mathrm{AC}}{\mathrm{CE}} = \frac{\mathrm{BD}}{\mathrm{DF}}$.

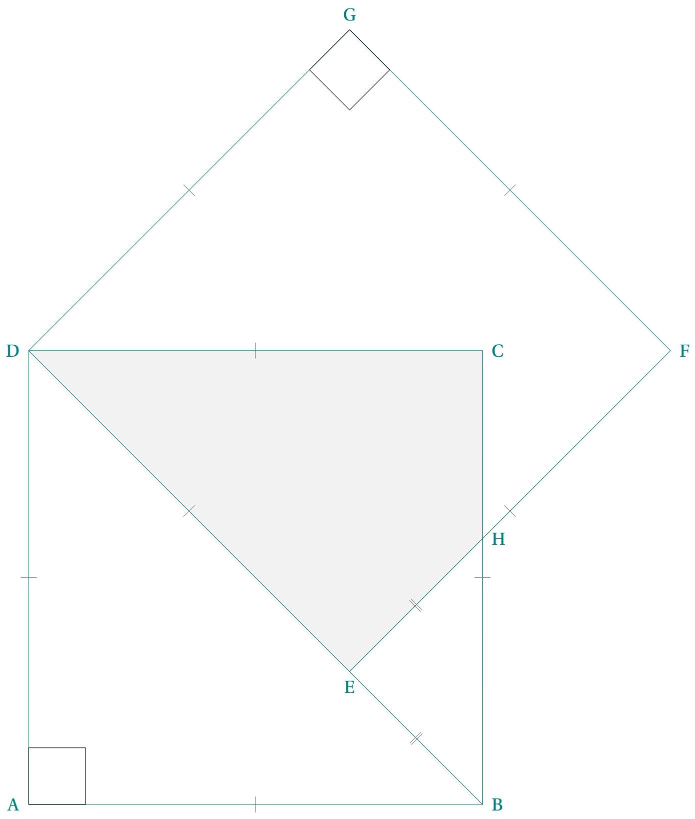
Another interesting example from John, you can see how to use some extra options like `decoration` and `postaction` from Ti*k*Z with **`tkz-euclide`**.

另一个来自 John 的有趣的例子, 可以看到来自 Ti*k*Z 的 **`tkz-euclide`** 使用额外的参数 `decoration` 和 `postaction`.



```
\begin{tikzpicture}[scale=1.5,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}
```

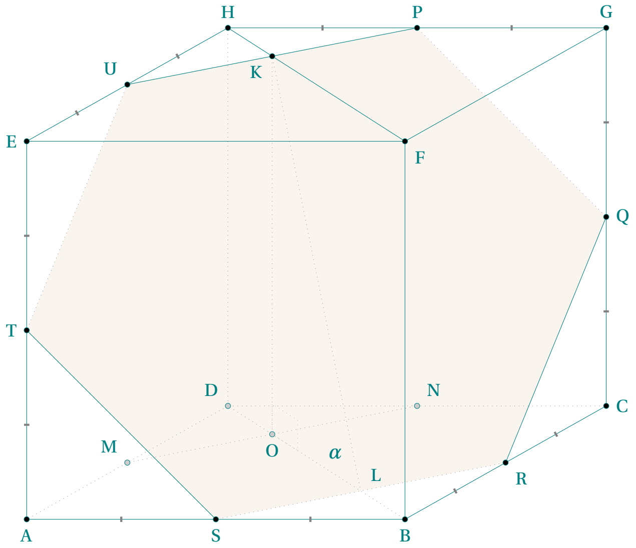
45.4. Example 1: from Indonesia



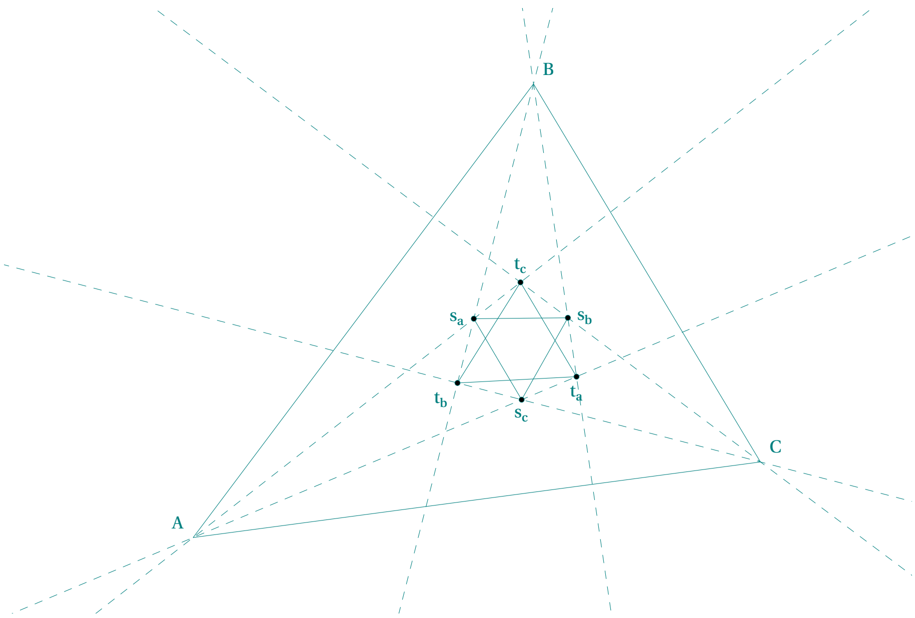
```
\begin{tikzpicture}[scale=3]
   \tkzDefPoints{0/0/A,2/0/B}
   \tkzDefSquare(A,B) \tkzGetPoints{C}{D}
   \tkzDefPointBy[rotation=center D angle 45](C)\tkzGetPoint{G}
   \tkzDefSquare(G,D)\tkzGetPoints{E}{F}
   \tkzInterLL(B,C)(E,F)\tkzGetPoint{H}
   \tkzFillPolygon[gray!10](D,E,H,C,D)
   \tkzDrawPolygon(A,...,D)\tkzDrawPolygon(D,...,G)
   \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}
```

45.5. Example 2: from Indonesia

```
  \begin{tikzpicture}[pol/.style={fill=brown!40,opacity=.2},
      seg/.style={tkzdotted,color=gray}, hidden pt/.style={fill=gray!40},
       mra/.style={color=gray!70,tkzdotted,/tkzrightangle/size=.2},scale=2]
  \tkzDefPoints{0/0/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}
```


H
P
G
U
K
E
F
Q
T
D
N
C
M
O
α
L
R
A
S
B

45.6. Illustration of the Morley theorem by Nicolas François 尼古拉斯•弗朗索瓦对莫利定理的阐释



```
\begin{tikzpicture}
  \tkzInit[ymin=-3,ymax=5,xmin=-5,xmax=7]
  \tkzClip
  \tkzDefPoints{-2.5/-2/A,2/4/B,5/-1/C}
  \tkzFindAngle(C,A,B) \tkzGetAngle{anglea}
  \tkzDefPointBy[rotation=center A angle 1*\anglea/3](C) \tkzGetPoint{TA1}
  \tkzDefPointBy[rotation=center A angle 2*\anglea/3](C) \tkzGetPoint{TA2}
  \tkzFindAngle(A,B,C) \tkzGetAngle{angleb}
  \tkzDefPointBy[rotation=center B angle 1*\angleb/3](A) \tkzGetPoint{TB1}
  \tkzDefPointBy[rotation=center B angle 2*\angleb/3](A) \tkzGetPoint{TB2}
  \tkzFindAngle(B,C,A) \tkzGetAngle{anglec}
  \tkzDefPointBy[rotation=center C angle 1*\anglec/3](B) \tkzGetPoint{TC1}
  \tkzDefPointBy[rotation=center C angle 2*\anglec/3](B) \tkzGetPoint{TC2}
  \tkzInterLL(A,TA1)(B,TB2) \tkzGetPoint{U1}
  \tkzInterLL(A,TA2)(B,TB1) \tkzGetPoint{V1}
  \tkzInterLL(B,TB1)(C,TC2) \tkzGetPoint{U2}
  \tkzInterLL(B,TB2)(C,TC1) \tkzGetPoint{V2}
  \tkzInterLL(C,TC1)(A,TA2) \tkzGetPoint{U3}
  \tkzInterLL(C,TC2)(A,TA1) \tkzGetPoint{V3}
  \tkzDrawPolygons(A,B,C U1,U2,U3 V1,V2,V3)
  \tkzDrawLines[add=2 and 2,very thin,dashed](A,TA1 B,TB1 C,TC1 A,TA2 B,TB2 C,TC2)
  \tkzDrawPoints(U1,U2,U3,V1,V2,V3)
  \tkzLabelPoint[left](V1){$s_a$} \tkzLabelPoint[right](V2){$s_b$}
  \tkzLabelPoint[below](V3){$s_c$} \tkzLabelPoint[above left](A){$A$}
  \tkzLabelPoints[above right](B,C) \tkzLabelPoint(U1){$t_a$}
  \tkzLabelPoint[below left](U2){$t_b$} \tkzLabelPoint[above](U3){$t_c$}
\end{tikzpicture}
```

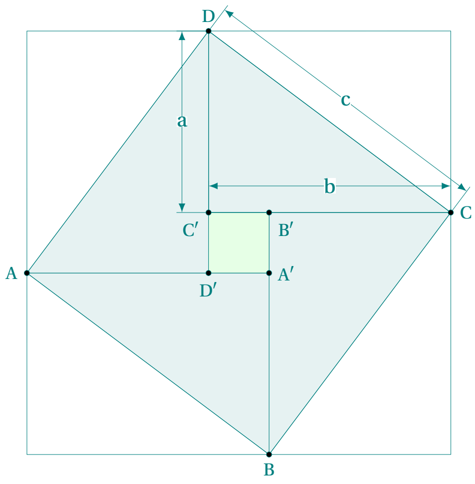
45.7. Gou gu theorem / Pythagorean Theorem by Zhao Shuang 勾股定理/毕氏定理 赵爽证法

Gou gu theorem / Pythagorean Theorem by Zhao Shuang

Pythagoras was not the first person who discovered this theorem around the world. Ancient China discovered this theorem much earlier than him. So there is another name for the Pythagorean theorem in China, the Gou-Gu theorem. Zhao Shuang was an ancient Chinese mathematician. He rediscovered the "Gou gu therorem", which is actually the Chinese version of the" Pythagorean theorem". Zhao Shuang used a method called the" cutting and compensation principle" , he created a picture of" Pythagorean Round Square" Below the figure used to illustrate the proof of the" Gou gu theorem." (code from Nan Geng)

勾股定理/毕氏定理 赵爽证法

毕达哥拉斯并不是全世界第一个发现这个定理的人。中国古代发现这个定理比他早得多。所以 *Pythagorean* 定理在中国还有另外一个名字，勾股定理。赵爽是中国古代数学家。他重新发现了“勾股定理”，其实就是中国版的“*Pythagorean* 定理”。赵爽使用了一种叫做”割补原理”的方法，他创造了一幅”勾股圆方图”，下面的图用来说明”勾股定理”的证明。(代码来自南庚)



```
\begin{tikzpicture}[scale=.8]
  \tkzDefPoint(0,0){A} \tkzDefPoint(4,0){A'}
  \tkzInterCC[R](A, 5)(A', 3)
  \tkzGetSecondPoint{B}
  \tkzDefSquare(A,B)   \tkzGetPoints{C}{D}
  \tkzCalcLength(A,A') \tkzGetLength{lA}
  \tkzCalcLength(A',B) \tkzGetLength{lB}
  \pgfmathparse{\lA-\lB}
  \tkzInterLC[R](A,A')(A',\pgfmathresult)
  \tkzGetFirstPoint{D'}
  \tkzDefSquare(D',A')\tkzGetPoints{B'}{C'}
  \tkzDefLine[orthogonal=through D](D,D')
   \tkzGetPoint{d}
  \tkzDefLine[orthogonal=through A](A,A')
   \tkzGetPoint{a}
  \tkzDefLine[orthogonal=through C](C,C')
   \tkzGetPoint{c}
  \tkzInterLL(D,d)(C,c) \tkzGetPoint{E}
  \tkzInterLL(D,d)(A,a) \tkzGetPoint{F}
  \tkzDefSquare(E,F)\tkzGetPoints{G}{H}
  \tkzDrawPolygons[fill=teal!10](A,B,A' B,C,B'
     C,D,C' A,D',D)
  \tkzDrawPolygons(A,B,C,D E,F,G,H)
  \tkzDrawPolygon[fill=green!10](A',B',C',D')
  \tkzDrawSegment[dim={$a$,-10pt,}](D,C')
  \tkzDrawSegment[dim={$b$,-10pt,}](C,C')
  \tkzDrawSegment[dim={$c$,-10pt,}](C,D)
  \tkzDrawPoints[size=2](A,B,C,D,A',B',C',D')
  \tkzLabelPoints[left](A)
  \tkzLabelPoints[below](B)
  \tkzLabelPoints[right](C)
  \tkzLabelPoints[above](D)
  \tkzLabelPoints[right](A')
  \tkzLabelPoints[below right](B')
  \tkzLabelPoints[below left](C')
  \tkzLabelPoints[below](D')
 \end{tikzpicture}
```

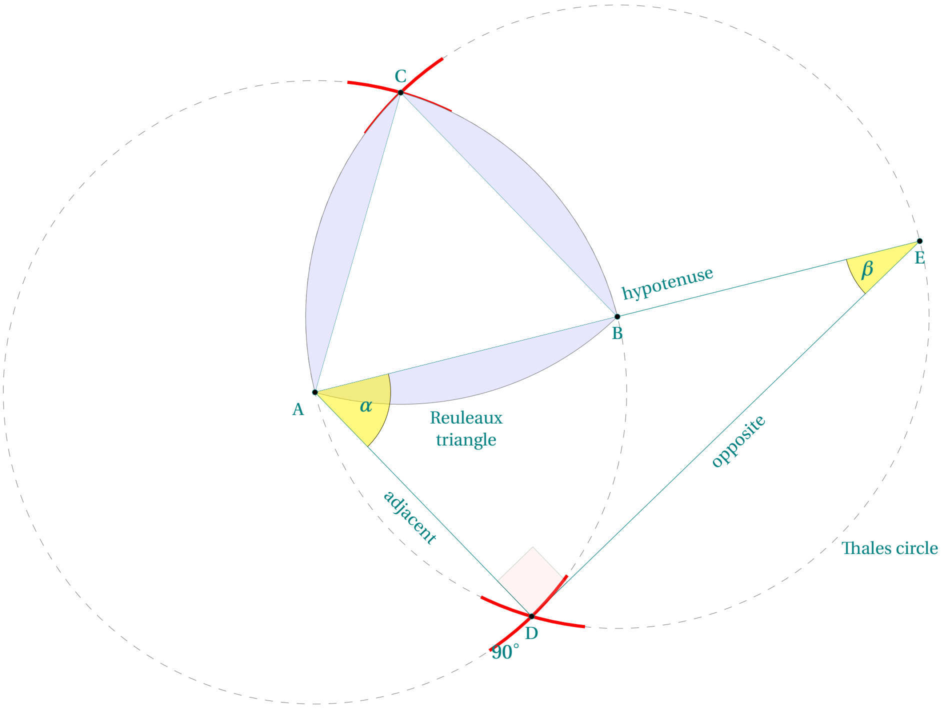
45.8. Reuleaux-Triangle 鲁洛三角形

Reuleaux-triangle by Stefan Kottwitz

A well-known classic field of mathematics is geometry. You may know Euclidean geometry from school, with constructions by compass and ruler. Math teachers may be very interested in drawing geometry constructions and explanations. Underlying constructions can help us with general drawings where we would need intersections and tangents of lines and circles, even if it does not look like geometry. So, here, we will remember school geometry drawings. We will use the tkz-euclide package, which works on top of TikZ. We will construct an equilateral triangle. Then we extend it to get a Reuleaux triangle, and add annotations. The code is fully explained in the LaTeX Cookbook, Chapter 10, Advanced Mathematics, Drawing geometry pictures. Stefan Kottwitz

斯特凡 • 科特维茨的鲁洛三角形

一个众所周知的数学经典领域是几何。你可能在学校学过欧几里得几何，用圆规和尺子作图。数学老师可能对画几何构造和解释很感兴趣。底层构造可以帮助我们绘制一般的图形，在这些图形中，需要直线和圆的交点和切线，即使它看起来不像几何图形。所以，在这里，大家将记住学校的几何图形。将使用 *tkz-euclide* 包，它工作在 *TikZ* 之上。将构建一个等边三角形。然后扩展它得到一个罗勒三角形，并添加注释。代码在 *LaTeX Cookbook*，第 *10* 章，高等数学，绘制几何图形中有完整的解释。斯特凡 • 科特维茨



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A} \tkzDefPoint(4,1){B}
  \tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{D}
  \tkzInterLC(A,B)(B,A) \tkzGetPoints{F}{E}
  \tkzDrawCircles[dashed](A,B B,A)
  \tkzDrawPolygons(A,B,C A,E,D)
  \tkzCompasss[color=red, very thick](A,C B,C A,D B,D)
  \begin{scope}
    \tkzSetUpArc[thick,delta=0]
    \tkzDrawArc[fill=blue!10](A,B)(C)
    \tkzDrawArc[fill=blue!10](B,C)(A)
    \tkzDrawArc[fill=blue!10](C,A)(B)
  \end{scope}
  \tkzMarkAngles(D,A,E A,E,D)
  \tkzFillAngles[fill=yellow,opacity=0.5](D,A,E A,E,D)
  \tkzMarkRightAngle[size=0.65,fill=red!20,opacity=0.2](A,D,E)
  \tkzLabelAngle[pos=0.7](D,A,E){$\alpha$}
  \tkzLabelAngle[pos=0.8](A,E,D){$\beta$}
  \tkzLabelAngle[pos=0.5,xshift=-1.4mm](A,D,D){$90^\circ$}
  \begin{scope}[font=\small]
    \tkzLabelSegment[below=0.6cm,align=center](A,B){Reuleaux\\triangle}
    \tkzLabelSegment[above right,sloped](A,E){hypotenuse}
    \tkzLabelSegment[below,sloped](D,E){opposite}
    \tkzLabelSegment[below,sloped](A,D){adjacent}
    \tkzLabelSegment[below right=4cm](A,E){Thales circle}
  \end{scope}
  \tkzLabelPoints[below left](A)
  \tkzLabelPoints(B,D)
  \tkzLabelPoint[above](C){$C$}
  \tkzLabelPoints(E)
  \tkzDrawPoints(A,...,E)

\end{tikzpicture}
```

46. Some interesting examples 有趣的例子

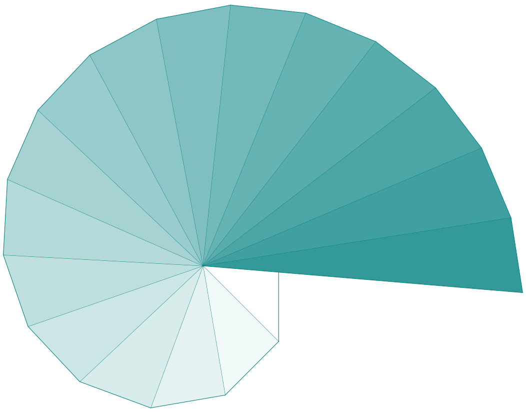
46.1. Square root of the integers 整数的平方根

Square root of the integers

How to get 1, $\sqrt{2}$, $\sqrt{3}$ *with a rule and a compass.*

整数的平方根

如何用尺规得到 1，$\sqrt{2}$，$\sqrt{3}$……。



```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){a0}
   \tkzDrawSegment(O,a0)
  \foreach \i [count=\j] in {0,...,16}{%
    \tkzDefPointWith[orthogonal normed](a\i,O)
    \tkzGetPoint{a\j}
       \pgfmathsetmacro{\c}{5*\i}
    \tkzDrawPolySeg[fill=teal!\c](a\i,a\j,O)}
 \end{tikzpicture}
```

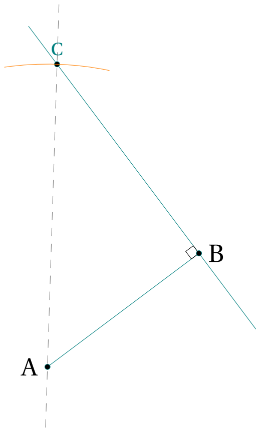
46.2. About right triangle 直角三角形

About right triangle

We have a segment [AB] *and we want to determine a point* C *such that* AC = 8 *cm and* ABC *is a right triangle in* B.

角三角形

有一条线段 [AB] 并且想确定一个点 C 使得 AC = 8 *cm* 并且 ABC 是以点 B 为直角顶点的一个直角三角形。



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

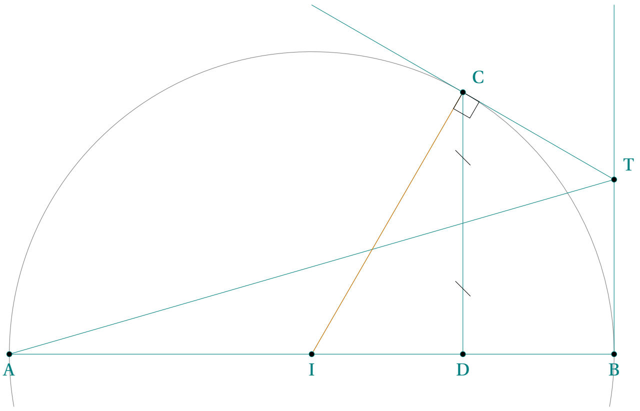
46.3. Archimedes 阿基米德

Archimedes

This is an ancient problem proved by the great Greek mathematician Archimedes . The figure below shows a semicircle, with diameter AB*. A tangent line is drawn and touches the semicircle at* B*. An other tangent line at a point,* C*, on the semicircle is drawn. We project the point* C *on the line segment* [AB] *on a point* D*. The two tangent lines intersect at the point* T*. Prove that the line* (AT) *bisects* (CD)

阿基米德

这是由伟大的希腊数学家阿基米德证明的一个古老问题。下图显示了一个直径为AB的半圆。画一条切线，切点为B。在半圆上的点C处画另一条切线。将点C投影到线段[AB]上的点D。两条切线相交于点T。证明直线(AT)平分线段(CD)



```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){A}\tkzDefPoint(6,0){D}
  \tkzDefPoint(8,0){B}\tkzDefPoint(4,0){I}
  \tkzDefLine[orthogonal=through D](A,D)
  \tkzInterLC[R](D,tkzPointResult)(I,4) \tkzGetSecondPoint{C}
  \tkzDefLine[orthogonal=through C](I,C)    \tkzGetPoint{c}
  \tkzDefLine[orthogonal=through B](A,B)    \tkzGetPoint{b}
  \tkzInterLL(C,c)(B,b) \tkzGetPoint{T}
  \tkzInterLL(A,T)(C,D) \tkzGetPoint{P}
  \tkzDrawArc(I,B)(A)
  \tkzDrawSegments(A,B A,T C,D I,C) \tkzDrawSegment[new](I,C)
  \tkzDrawLine[add = 1 and 0](C,T)   \tkzDrawLine[add = 0 and 1](B,T)
  \tkzMarkRightAngle(I,C,T)
  \tkzDrawPoints(A,B,I,D,C,T)
  \tkzLabelPoints(A,B,I,D)  \tkzLabelPoints[above right](C,T)
  \tkzMarkSegment[pos=.25,mark=s|](C,D) \tkzMarkSegment[pos=.75,mark=s|](C,D)
\end{tikzpicture}
```

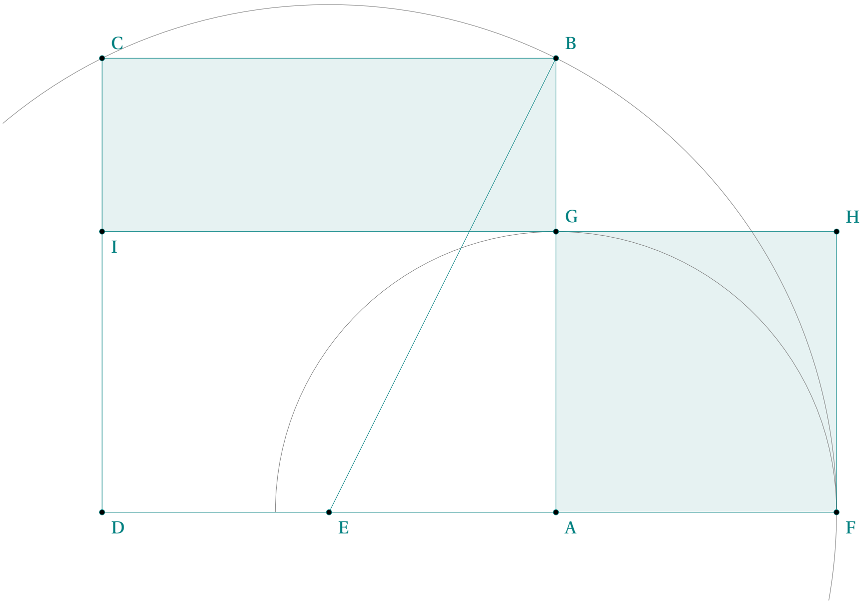
46.3.1. Square and rectangle of same area; Golden section 相同面积的正方形和长方形；黄金分割

Book II, proposition XI _Euclid’s Elements_

To construct Square and rectangle of same area.

欧几里得的《几何原本》第二册，命题 Xi

构造相同面积的正方形和长方形。



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(0,0){D} \tkzDefPoint(8,0){A}
 \tkzDefSquare(D,A) \tkzGetPoints{B}{C}
 \tkzDefMidPoint(D,A) \tkzGetPoint{E}
 \tkzInterLC(D,A)(E,B)\tkzGetSecondPoint{F}
 \tkzInterLC[near](B,A)(A,F)\tkzGetFirstPoint{G}
 \tkzDefSquare(A,F)\tkzGetFirstPoint{H}
 \tkzInterLL(C,D)(H,G)\tkzGetPoint{I}
 \tkzFillPolygon[teal!10](I,G,B,C)
 \tkzFillPolygon[teal!10](A,F,H,G)
 \tkzDrawArc[angles](E,B)(0,120)
 \tkzDrawSemiCircle(A,F)
 \tkzDrawSegments(A,F E,B H,I F,H)
 \tkzDrawPolygons(A,B,C,D)
 \tkzDrawPoints(A,...,I)
 \tkzLabelPoints[below right](A,E,D,F,I)
 \tkzLabelPoints[above right](C,B,G,H)
\end{tikzpicture}
```

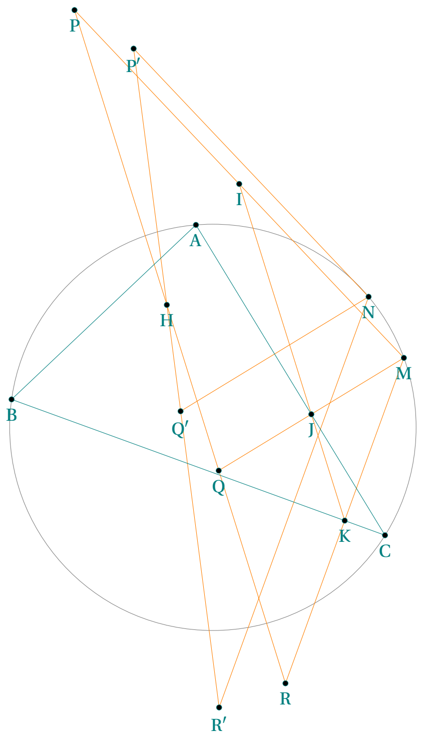
46.3.2. Steiner Line and Simson Line 斯坦纳线和西姆森线

Steiner Line and Simson Line

Consider the triangle ABC and a point M on its circumcircle. The projections of M on the sides of the triangle are on a line (Steiner Line), The three closest points to M on lines AB, AC, and BC are collinear. It's the Simson Line.

斯坦纳线和西姆森线

考虑三角形 *ABC* 和其外接圆上的点 *M*。*M* 在三角形各边上的投影在一条直线上 (斯坦纳线)，直线 *AB*、*AC* 和 *BC* 上离 *M* 最近的三个点共线（西姆森线）。



```
\begin{tikzpicture}[scale=.75,rotate=-20]
  \tkzDefPoint(0,0){B}
  \tkzDefPoint(2,4){A} \tkzDefPoint(7,0){C}
  \tkzDefCircle[circum](A,B,C)
  \tkzGetPoint{O}
  \tkzDrawCircle(O,A)
  \tkzCalcLength(O,A)
  \tkzGetLength{rOA}
  \tkzDefShiftPoint[O](40:\rOA){M}
  \tkzDefShiftPoint[O](60:\rOA){N}
  \tkzDefTriangleCenter[orthic](A,B,C)
  \tkzGetPoint{H}
  \tkzDefSpcTriangle[orthic,name=H](A,B,C){a,b,c}
  \tkzDefPointsBy[reflection=over A--B](M,N){P,P'}
  \tkzDefPointsBy[reflection=over A--C](M,N){Q,Q'}
  \tkzDefPointsBy[reflection=over C--B](M,N){R,R'}
  \tkzDefMidPoint(M,P)\tkzGetPoint{I}
  \tkzDefMidPoint(M,Q)\tkzGetPoint{J}
  \tkzDefMidPoint(M,R)\tkzGetPoint{K}
  \tkzDrawSegments[new](P,R M,P M,Q M,R N,P'%
   N,Q' N,R' P',R' I,K)
  \tkzDrawPolygons(A,B,C)
  \tkzDrawPoints(A,B,C,H,M,N,P,Q,R,P',Q',R',I,J,K)
  \tkzLabelPoints(A,B,C,H,M,N,P,Q,R,P',Q',R',I,J,K)
\end{tikzpicture}
```

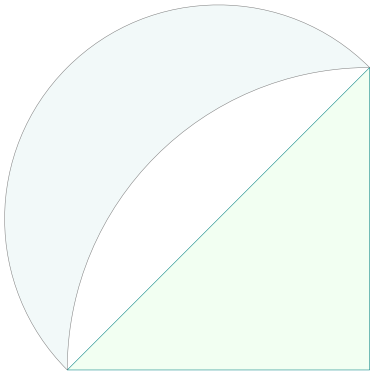
46.4. Lune of Hippocrates 希波克拉底月牙

Lune of Hippocrates

From wikipedia : In geometry, the lune of Hippocrates, named after Hippocrates of Chios, is a lune bounded by arcs of two circles, the smaller of which has as its diameter a chord spanning a right angle on the larger circle.In the first figure, the area of the lune is equal to the area of the triangle ABC. Hippocrates of Chios (ancient Greek mathematician,)

希波克拉底月牙

来自维基百科:在几何学中，希波克拉底月牙，以希波克拉底的希波克拉底命名，是由两个圆的弧线围成的月牙，其中较小的一个圆的直径是一条弦，它与较大的圆成直角。在第一幅图中，月牙的面积等于三角形 *ABC* 的面积。希波克拉底 *(*古希腊数学家*)*



```
\begin{tikzpicture}
 \tkzInit[xmin=-2,xmax=5,ymin=-1,ymax=6]
 \tkzClip % allows you to define a bounding box
   % large enough
  \tkzDefPoint(0,0){A}\tkzDefPoint(4,0){B}
  \tkzDefSquare(A,B)
  \tkzGetFirstPoint{C}
  \tkzDrawPolygon[fill=green!5](A,B,C)
   \begin{scope}
     \tkzClipCircle[out](B,A)
     \tkzDefMidPoint(C,A) \tkzGetPoint{M}
     \tkzDrawSemiCircle[fill=teal!5](M,C)
   \end{scope}
   \tkzDrawArc[delta=0](B,C)(A)
\end{tikzpicture}
```

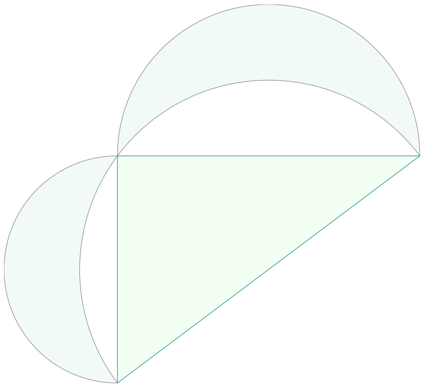
46.5. Lunes of Hasan Ibn al-Haytham 哈桑・伊本・海赛姆的月牙

Lunes of Hasan Ibn al-Haytham

From wikipedia : the Arab mathematician Hasan Ibn al-Haytham (Latinized name Alhazen) showed that two lunes, formed on the two sides of a right triangle, whose outer boundaries are semicircles and whose inner boundaries are formed by the circumcircle of the triangle, then the areas of these two lunes added together are equal to the area of the triangle. The lunes formed in this way from a right triangle are known as the lunes of Alhazen.

哈桑・伊本・海赛姆的月牙

来自维基百科：阿拉伯数学家哈桑・伊本・海瑟姆（拉丁名称 *Alhazen*）表明，在直角三角形的两侧形成两个月牙，其外边界是半圆，其内边界由三角形的圆周形成，那么这两个月牙的面积加在一起等于三角形的面积。以这种方式从直角三角形形成的月牙被称为阿尔哈森的月牙。



```
\begin{tikzpicture}[scale=.5,rotate=180]
  \tkzInit[xmin=-1,xmax=11,ymin=-4,ymax=7]
  \tkzClip
  \tkzDefPoints{0/0/A,8/0/B}
  \tkzDefTriangle[pythagore,swap](A,B)
  \tkzGetPoint{C}
  \tkzDrawPolygon[fill=green!5](A,B,C)
  \tkzDefMidPoint(C,A) \tkzGetPoint{I}
  \begin{scope}
    \tkzClipCircle[out](I,A)
    \tkzDefMidPoint(B,A) \tkzGetPoint{x}
    \tkzDrawSemiCircle[fill=teal!5](x,A)
    \tkzDefMidPoint(B,C) \tkzGetPoint{y}
    \tkzDrawSemiCircle[fill=teal!5](y,B)
  \end{scope}
  \tkzSetUpCompass[/tkzcompass/delta=0]
      \tkzDefMidPoint(C,A) \tkzGetPoint{z}
  \tkzDrawSemiCircle(z,A)
\end{tikzpicture}
```

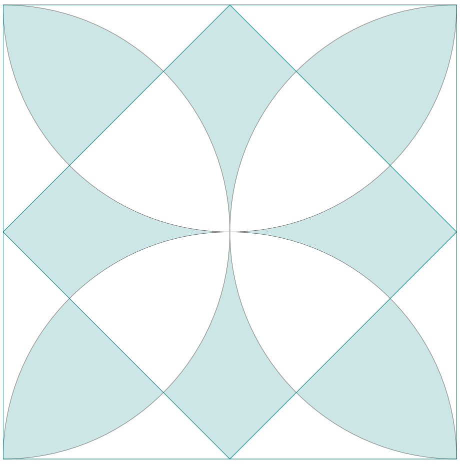
46.6. About clipping circles 关于剪裁圆

About clipping circles

The problem is the management of the bounding box. First you have to define a rectangle in which the figure will be inserted. This is done with the first two lines.

关于剪裁圆

问题是边界框的管理。首先，必须定义一个矩形，在其中插入图形。这是通过前两行完成的。



```
\begin{tikzpicture}
  \tkzInit[xmin=0,xmax=6,ymin=0,ymax=6]
  \tkzClip
  \tkzDefPoints{0/0/A, 6/0/B}
  \tkzDefSquare(A,B)      \tkzGetPoints{C}{D}
  \tkzDefMidPoint(A,B)        \tkzGetPoint{M}
  \tkzDefMidPoint(A,D)        \tkzGetPoint{N}
  \tkzDefMidPoint(B,C)        \tkzGetPoint{O}
  \tkzDefMidPoint(C,D)        \tkzGetPoint{P}
 \begin{scope}
  \tkzClipCircle[out](M,B) \tkzClipCircle[out](P,D)
  \tkzFillPolygon[teal!20](M,N,P,O)
 \end{scope}
 \begin{scope}
   \tkzClipCircle[out](N,A) \tkzClipCircle[out](O,C)
   \tkzFillPolygon[teal!20](M,N,P,O)
 \end{scope}
 \begin{scope}
   \tkzClipCircle(P,C) \tkzClipCircle(N,A)
   \tkzFillPolygon[teal!20](N,P,D)
 \end{scope}
 \begin{scope}
     \tkzClipCircle(O,C) \tkzClipCircle(P,C)
     \tkzFillPolygon[teal!20](P,C,O)
 \end{scope}
 \begin{scope}
     \tkzClipCircle(M,B)  \tkzClipCircle(O,B)
     \tkzFillPolygon[teal!20](O,B,M)
 \end{scope}
 \begin{scope}
     \tkzClipCircle(N,A) \tkzClipCircle(M,A)
     \tkzFillPolygon[teal!20](A,M,N)
 \end{scope}
 \tkzDrawSemiCircles(M,B N,A O,C P,D)
 \tkzDrawPolygons(A,B,C,D M,N,P,O)
 \end{tikzpicture}
```

46.7. Similar isosceles triangles 相似等腰三角形

Similar isosceles triangles

The following is from the excellent site ***Descartes et les Mathématiques****. I did not modify the text and I am only the author of the programming of the figures.* *http://debart.pagesperso-orange.fr/seconde/triangle.html*

相似等腰三角形

以下内容来自优秀网站 ***Descartes et les Mathématiques***。我没有修改文本，我只是图形编程的作者。*http://debart.pagesperso-orange.fr/seconde/triangle.html*

Bibliography:
参考文献：

- Géométrie au Bac - Tangente, special issue no. 8 - Exercise 11, page 11
- 巴克的几何学–《切线》，第 8 期特刊–练习 11，第 11 页
- Elisabeth Busser and Gilles Cohen: 200 nouveaux problèmes du ”Monde” - POLE 2007 (200 new problems of ”Le Monde”)
- Elisabeth Busser 和 Gilles Cohen：《世界报》的 200 个新问题 - POLE 2007
- Affaire de logique n° 364 - Le Monde February 17, 2004
- 第 364 号逻辑案例 -《世界报》2004 年 2 月 17 日

Two statements were proposed, one by the magazine *Tangente* and the other by *Le Monde*.
Tangente 杂志，Le Monde 杂志，提出了两个声明。

Editor of the magazine ”Tangente”: Two similar isosceles triangles AXB and BYC are constructed with main vertices X and Y, such that A, B and C are aligned and that these triangles are ”indirect”. Let α be the angle at vertex $\widehat{AXB} = \widehat{BYC}$. We then construct a third isosceles triangle XZY similar to the first two, with main vertex Z and ”indirect”. We ask to demonstrate that point Z belongs to the straight line (AC).
”Tangente ” 杂志的编辑：两个相似的等腰三角形 AXB 和 BYC 的公共顶点是 X 和 Y，这样 A、B 和 C 是共线的，这些三角形是 ” 间接 ” 的。在顶点 $\widehat{AXB} = \widehat{BYC}$ 的角度为 alpha。然后构造第三个等腰三角形 XZY 与前两个相似，主顶点 Z 和 ” 间接”。要求证明点 Z 在直线 (AC) 上。

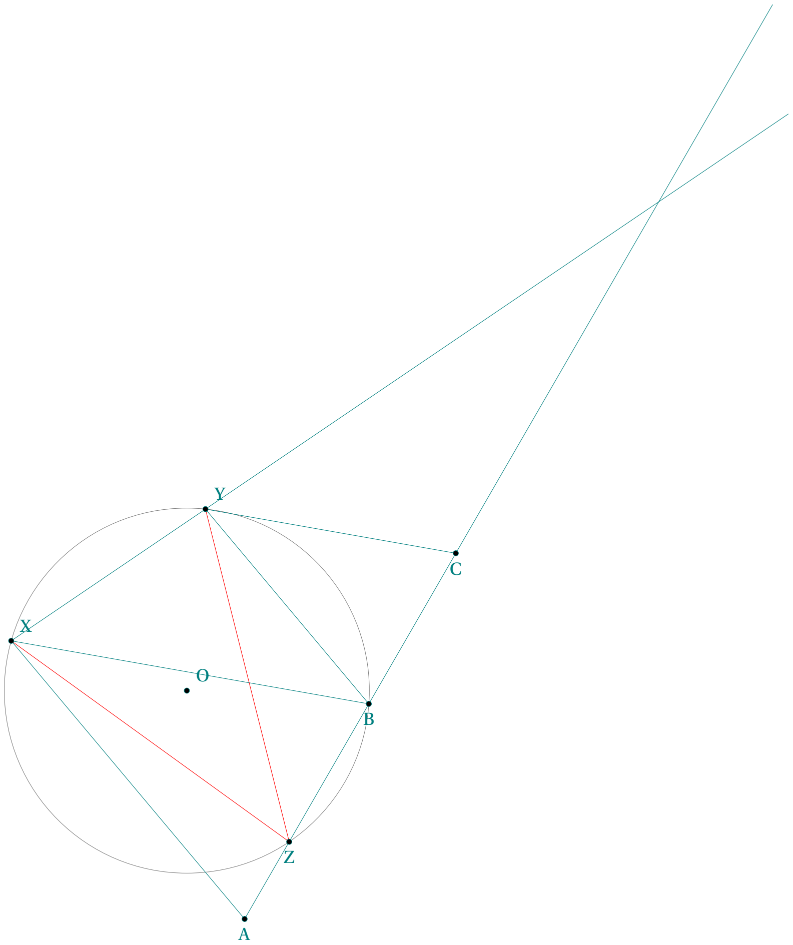
Editor of ”Le Monde”: We construct two similar isosceles triangles AXB and BYC with principal vertices X and Y, such that A, B and C are aligned and that these triangles are ”indirect”. Let α be the angle at vertex $\widehat{AXB} = \widehat{BYC}$. The point Z of the line segment [AC] is equidistant from the two vertices X and Y.
At what angle does he see these two vertices?

世界报编辑: 用主顶点 X 和 Y 构造两个相似的等腰三角形 AXB 和 BYC 使得 A 和 B 共线，并且这些三角形是“间接的”。设顶点 $\widehat{AXB} = \widehat{BYC}$ 的角度为 alpha。线段 [AC] 的点 Z 与两个顶点 X 和 Y 的距离相等。
从什么角度看这两个顶点？

The constructions and their associated codes are on the next two pages, but you can search before looking. The programming respects (it seems to me ...) my reasoning in both cases.

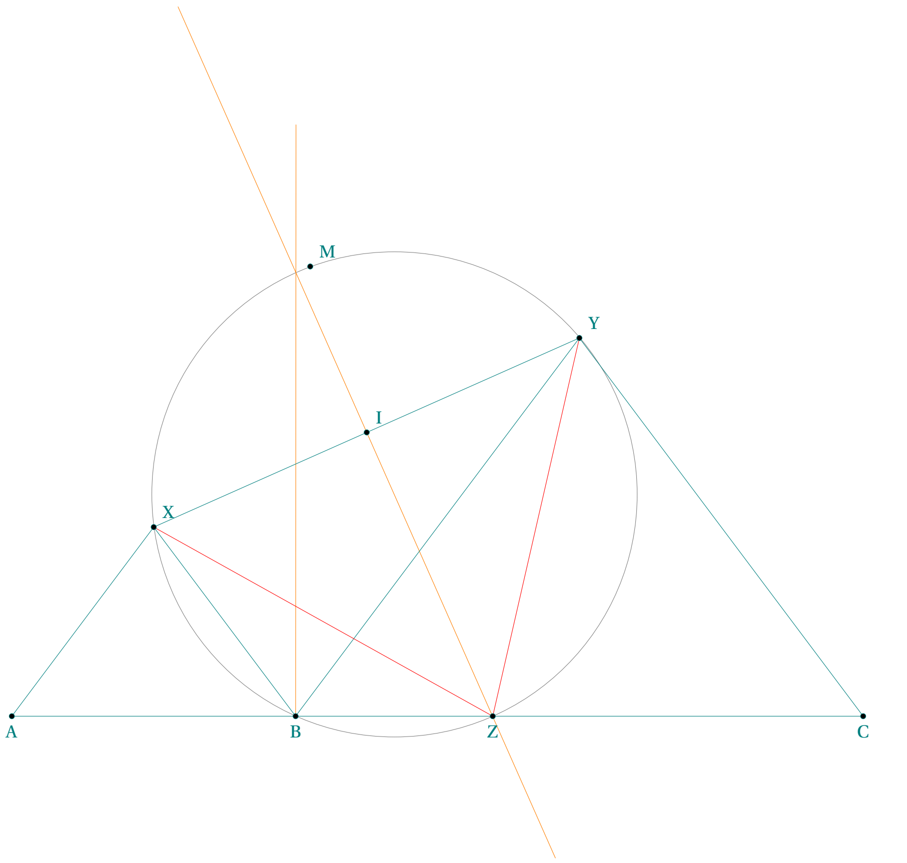
这些结构和它们的相关代码在接下来的两页上，但你可以在阅读之前进行搜索。在这两种情况下，编程反应了（在我看来......）我的推理。

46.8. Revised version of "Tangente" 修订版



```
\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}
```

46.9. "Le Monde" version《世界报》版本



```
\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)             \tkzGetFirstPoint{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}
```

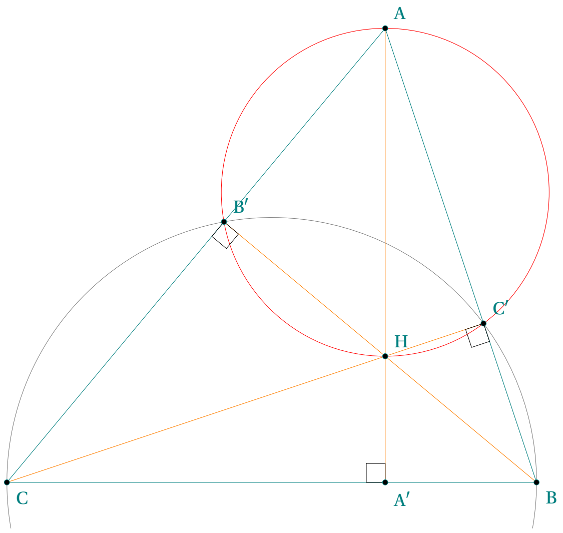
46.10. Triangle altitudes 三角形的高

Triangle altitudes

From Wikipedia : The following is again from the excellent site ***Descartes et les Mathématiques*** *(Descartes and the Mathematics).* *http://debart.pagesperso-orange.fr/geoplan/geometrie_triangle.html*. *The three altitudes of a triangle intersect at the same H-point.*

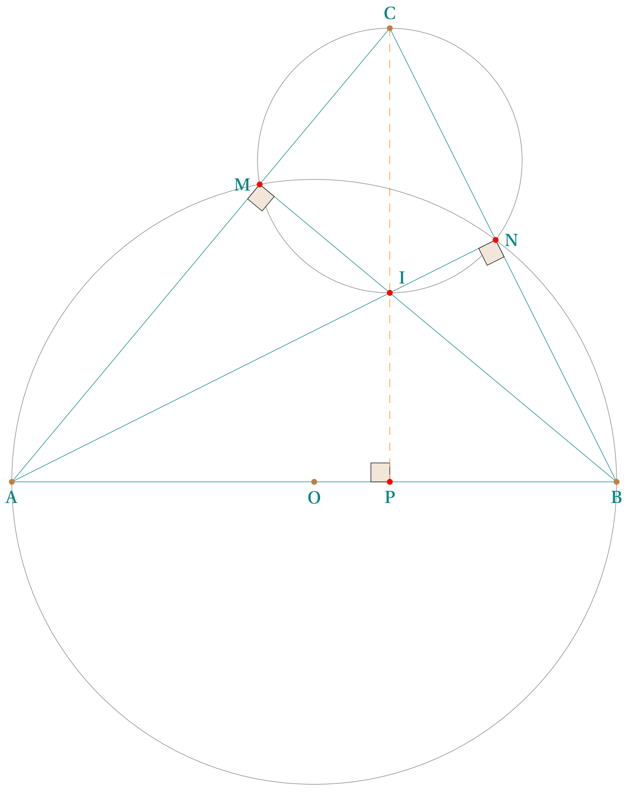
Triangle altitudes 三角形的高

来自维基百科: 以下同样来自优秀的网站 (笛卡尔和数学)。http://debart.pagesperso-orange.fr/geoplan/geometrie_triangle.html. 三角形的三条高相交于同一个点 H。



```
\begin{tikzpicture}
   \tkzDefPoint(0,0){C} \tkzDefPoint(7,0){B}
   \tkzDefPoint(5,6){A}
   \tkzDefMidPoint(C,B) \tkzGetPoint{I}
   \tkzInterLC(A,C)(I,B)
   \tkzGetFirstPoint{B'}
   \tkzInterLC(A,B)(I,B)
   \tkzGetSecondPoint{C'}
   \tkzInterLL(B,B')(C,C') \tkzGetPoint{H}
   \tkzInterLL(A,H)(C,B) \tkzGetPoint{A'}
   \tkzDefCircle[circum](A,B',C') \tkzGetPoint{O}
   \tkzDrawArc(I,B)(C)
   \tkzDrawPolygon(A,B,C)
   \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[above right](A,B',C',H)
   \tkzLabelPoints[below right](B,C,A')
\end{tikzpicture}
```

46.11. Altitudes - other construction 高：其他构造方法



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
\tkzDefPoint(5,6){C}
\tkzDefMidPoint(A,B)\tkzGetPoint{O}
\tkzDefPointBy[projection=onto A--B](C) \tkzGetPoint{P}
\tkzInterLC[common=A](C,A)(O,A)
\tkzGetFirstPoint{M}
\tkzInterLC(C,B)(O,A)
\tkzGetSecondPoint{N}
\tkzInterLL(B,M)(A,N)\tkzGetPoint{I}
\tkzDefCircle[diameter](A,B)\tkzGetPoint{x}
\tkzDefCircle[diameter](I,C)\tkzGetPoint{y}
\tkzDrawCircles(x,A y,C)
\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}
```

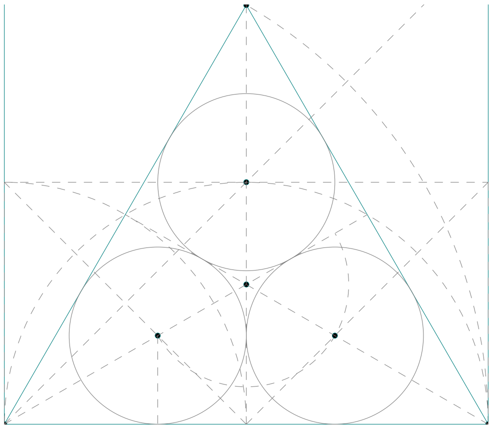
46.12. Three circles in an Equilateral Triangle 等边三角形中的三个圆

Three circles in an Equilateral Triangle

From Wikipedia : In geometry, the Malfatti circles are three circles inside a given triangle such that each circle is tangent to the other two and to two sides of the triangle. They are named after Gian Francesco Malfatti, who made early studies of the problem of constructing these circles in the mistaken belief that they would have the largest possible total area of any three disjoint circles within the triangle. Below is a study of a particular case with an equilateral triangle and three identical circles.

等边三角形中的三个圆

来自维基百科：在几何学中，马尔法蒂圆是给定三角形内的三个圆，每个圆都与其他两个圆和三角形的两条边相切。他们是以 *Gian Francesco Malfatti* 的名字命名的，他早期研究了构建这些圆的问题，错误地认为这些圆在三角形内任何三个不相交的圆中具有最大可能的总面积。下面是对一个等边三角形和三个相同圆的特殊情况的研究。



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

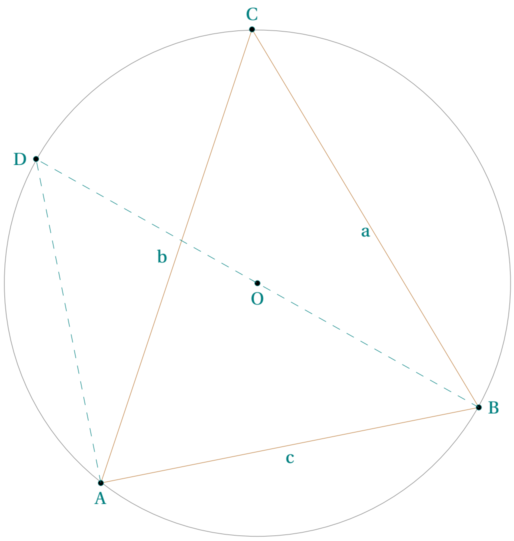
46.13. Law of sines 正弦定理

Law of sines

From wikipedia : *In trigonometry, the law of sines, sine law, sine formula, or sine rule is an equation relating the lengths of the sides of a triangle (any shape) to the sines of its angles.*

正弦定理

来自维基百科：在三角学中，正弦定理（正弦法则、正弦公式或正弦规则）是一个与三角形（任何形状）的边长和其对角的正弦有关的等式。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,5/1/B,2/6/C}
\tkzDefTriangleCenter[circum](A,B,C)
 \tkzGetPoint{O}
\tkzDefPointBy[symmetry= center O](B)
 \tkzGetPoint{D}
\tkzDrawPolygon[color=brown](A,B,C)
\tkzDrawCircle(O,A)
\tkzDrawPoints(A,B,C,D,O)
\tkzDrawSegments[dashed](B,D A,D)
\tkzLabelPoint[left](D){$D$}
\tkzLabelPoint[below](A){$A$}
\tkzLabelPoint[above](C){$C$}
\tkzLabelPoint[right](B){$B$}
\tkzLabelPoint[below](O){$O$}
\tkzLabelSegment(B,C){$a$}
\tkzLabelSegment[left](A,C){$b$}
\tkzLabelSegment(A,B){$c$}
\end{tikzpicture}
```

In the triangle ABC

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} \tag{1}$$

$$\widehat{C} = \widehat{D}$$

$$\frac{c}{2R} = \sin D = \sin C \tag{2}$$

Then

$$\frac{c}{\sin C} = 2R$$

46.14. Flower of Life 生命之花

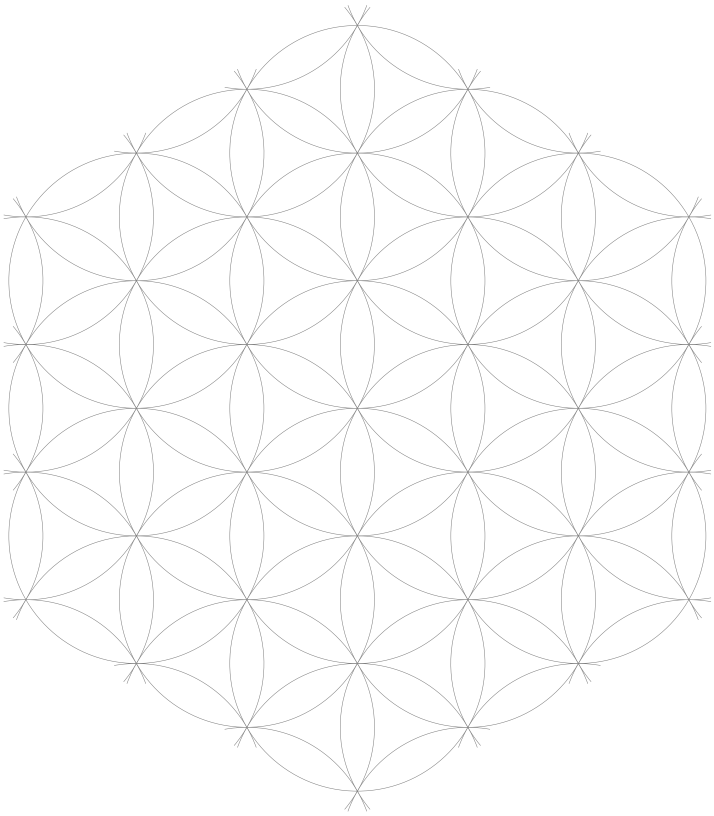
Book IV, proposition XI _Euclid's Elements_

Sacred geometry can be described as a belief system attributing a religious or cultural value to many of the fundamental forms of space and time. According to this belief system, the basic patterns of existence are perceived as sacred because in contemplating them one is contemplating the origin of all things. By studying the nature of these forms and their relationship to each other, one may seek to gain insight into the scientific, philosophical, psychological, aesthetic and mystical laws of the universe. The Flower of Life is considered to be a symbol of sacred geometry, said to contain ancient, religious value depicting the fundamental forms of space and time. In this sense, it is a visual expression of the connections life weaves through all mankind, believed by some to contain a type of Akashic Record of basic information of all living things.

Book IV, proposition XI _Euclid's Elements_

神圣的几何学可以被描述为一种信仰体系，它将宗教或文化价值赋予许多基本的空间和时间形式。根据这一信仰体系，存在的基本模式被认为是神圣的，因为在思考这些模式时，人们在思考万物的起源。通过研究这些形式的性质及其相互之间的关系，人们可以寻求对宇宙的科学、哲学、心理学、美学和神秘法则的洞察力。生命之花被认为是一个神圣的几何符号，据说包含了古老的宗教价值，描述了空间和时间的基本形式。从这个意义上说，它是生命在全人类中编织的联系的视觉表达，一些人认为它包含了一种所有生物的基本信息的阿卡西记录。

One of the beautiful arrangements of circles found at the Temple of Osiris at Abydos, Egypt (Rawles 1997).
Weisstein, Eric W. "Flower of Life." From MathWorld–A Wolfram Web Resource.
http://mathworld.wolfram.com/FlowerofLife.html
在埃及阿比多斯的奥西里斯神庙发现的美丽的圆圈排列之一（罗尔斯，1997 年）。
Weisstein, Eric W. " 生命之花"。来自 MathWorld–一个 Wolfram 网络资源。
http://mathworld.wolfram.com/FlowerofLife.html



```
\begin{tikzpicture}[scale=.75]
  \tkzSetUpLine[line width=2pt,color=teal!80!black]
  \tkzSetUpCompass[line width=2pt,color=teal!80!black]
   \tkzDefPoint(0,0){O}  \tkzDefPoint(2.25,0){A}
   \tkzDrawCircle(O,A)
\foreach \i in {0,...,5}{
   \tkzDefPointBy[rotation= center O angle 30+60*\i](A)\tkzGetPoint{a\i}
   \tkzDefPointBy[rotation= center {a\i} angle  120](O)\tkzGetPoint{b\i}
   \tkzDefPointBy[rotation= center {a\i} angle  180](O)\tkzGetPoint{c\i}
   \tkzDefPointBy[rotation= center {c\i} angle  120](a\i)\tkzGetPoint{d\i}
   \tkzDefPointBy[rotation= center {c\i} angle   60](d\i)\tkzGetPoint{f\i}
   \tkzDefPointBy[rotation= center {d\i} angle   60](b\i)\tkzGetPoint{e\i}
   \tkzDefPointBy[rotation= center {f\i} angle   60](d\i)\tkzGetPoint{g\i}
   \tkzDefPointBy[rotation= center {d\i} angle   60](e\i)\tkzGetPoint{h\i}
   \tkzDefPointBy[rotation= center {e\i} angle  180](b\i)\tkzGetPoint{k\i}
   \tkzDrawCircle(a\i,O)
   \tkzDrawCircle(b\i,a\i)
   \tkzDrawCircle(c\i,a\i)
   \tkzDrawArc[rotate](f\i,d\i)(-120)
   \tkzDrawArc[rotate](e\i,d\i)(180)
   \tkzDrawArc[rotate](d\i,f\i)(180)
   \tkzDrawArc[rotate](g\i,f\i)(60)
   \tkzDrawArc[rotate](h\i,d\i)(60)
   \tkzDrawArc[rotate](k\i,e\i)(60)
}
   \tkzClipCircle(O,f0)
\end{tikzpicture}
```

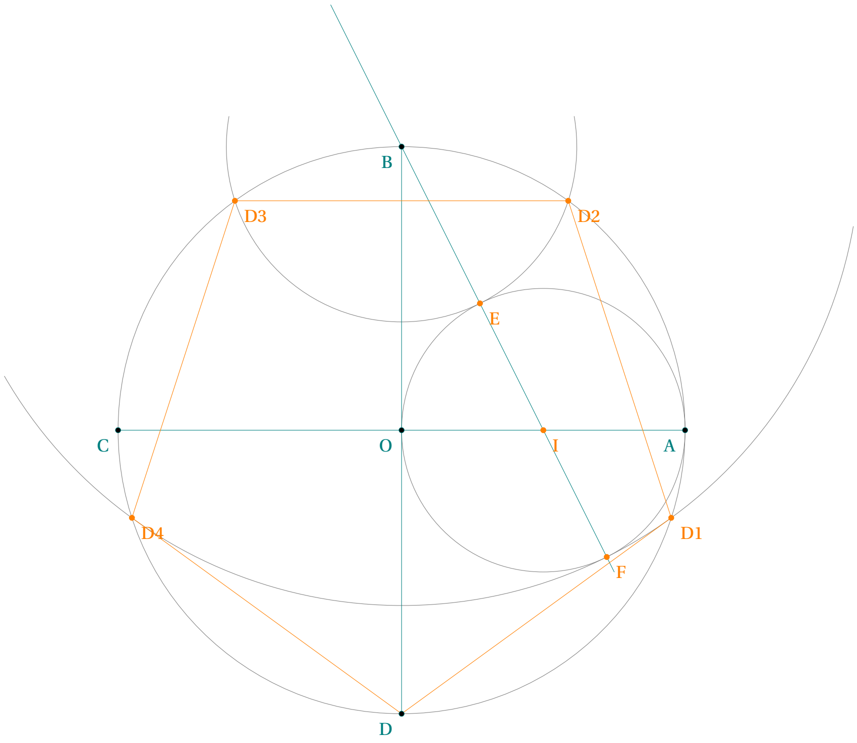
46.15. Pentagon in a circle 圆内接五边形

Book IV, proposition XI _Euclid's Elements_

To inscribe an equilateral and equiangular pentagon in a given circle.

Book IV, proposition XI _Euclid's Elements_

在给定的圆内画上一个正（等边和等角的）五边形。



```
\begin{tikzpicture}[scale=.75]
   \tkzDefPoint(0,0){O}
   \tkzDefPoint(5,0){A}
   \tkzDefPoint(0,5){B}
   \tkzDefPoint(-5,0){C}
   \tkzDefPoint(0,-5){D}
   \tkzDefMidPoint(A,O)              \tkzGetPoint{I}
   \tkzInterLC(I,B)(I,A)            \tkzGetPoints{F}{E}
   \tkzInterCC(O,C)(B,E)            \tkzGetPoints{D3}{D2}
   \tkzInterCC(O,C)(B,F)            \tkzGetPoints{D4}{D1}
   \tkzDrawArc[angles](B,E)(180,360)
   \tkzDrawArc[angles](B,F)(220,340)
   \tkzDrawLine[add=.5 and .5](B,I)
   \tkzDrawCircle(O,A)
   \tkzDefCircle[diameter](O,A)     \tkzGetPoint{x}
   \tkzDrawCircle(x,A)
   \tkzDrawSegments(B,D C,A)
   \tkzDrawPolygon[new](D,D1,D2,D3,D4)
   \tkzDrawPoints(A,...,D,O)
   \tkzDrawPoints[new](E,F,I,D1,D2,D4,D3)
   \tkzLabelPoints[below left](A,...,D,O)
   \tkzLabelPoints[new,below right](I,E,F,D1,D2,D4,D3)
\end{tikzpicture}
```

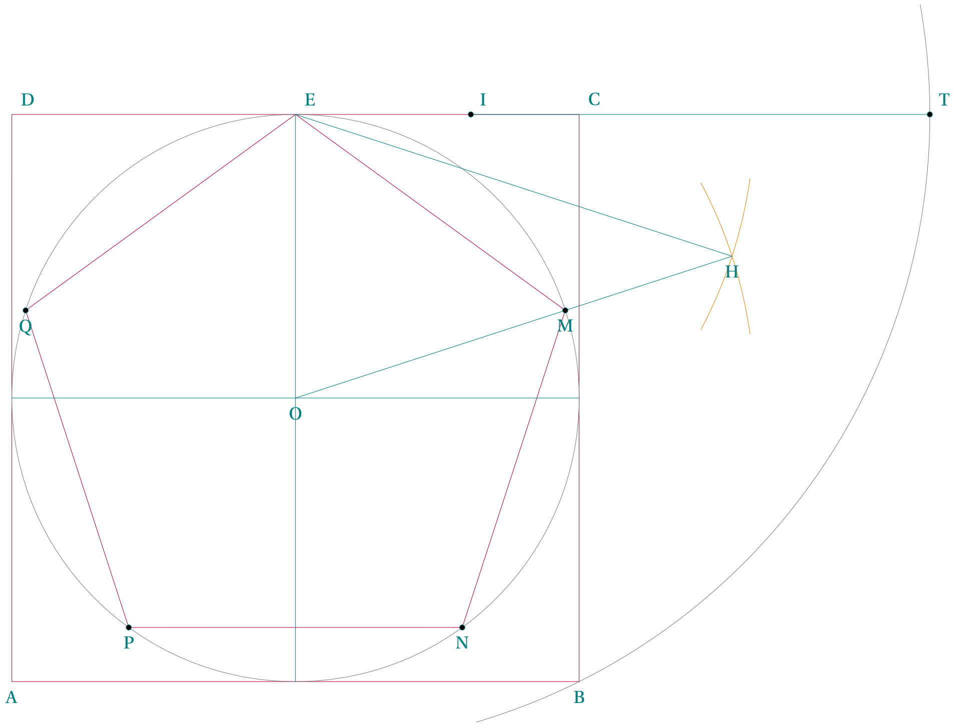
46.16. Pentagon in a square 正方形内的正五边形

Pentagon in a square

: *To inscribe an equilateral and equiangular pentagon in a given square.*

正方形内的正五边形

: 在一个给定的正方形中画上一个正（等边和等角的）五边形。



```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O,-5/-5/A,5/-5/B}
  \tkzDefSquare(A,B)   \tkzGetPoints{C}{D}
  \tkzDefMidPoint(A,B) \tkzGetPoint{F}
  \tkzDefMidPoint(C,D) \tkzGetPoint{E}
  \tkzDefMidPoint(B,C) \tkzGetPoint{G}
  \tkzDefMidPoint(A,D) \tkzGetPoint{K}
  \tkzInterLC(D,C)(E,B)                    \tkzGetSecondPoint{T}
  \tkzDefMidPoint(D,T)                     \tkzGetPoint{I}
  \tkzInterCC[with nodes](O,D,I)(E,D,I)    \tkzGetSecondPoint{H}
  \tkzInterLC(O,H)(O,E)                    \tkzGetSecondPoint{M}
  \tkzInterCC(O,E)(E,M)                    \tkzGetFirstPoint{Q}
  \tkzInterCC[with nodes](O,O,E)(Q,E,M)    \tkzGetFirstPoint{P}
  \tkzInterCC[with nodes](O,O,E)(P,E,M)    \tkzGetFirstPoint{N}
  \tkzCompasss(O,H E,H)
  \tkzDrawArc(E,B)(T)
  \tkzDrawPolygons[purple](A,B,C,D M,E,Q,P,N)
  \tkzDrawCircle(O,E)
  \tkzDrawSegments(T,I O,H E,H E,F G,K)
  \tkzDrawPoints(T,M,Q,P,N,I)
  \tkzLabelPoints(A,B,O,N,P,Q,M,H)
  \tkzLabelPoints[above right](C,D,E,I,T)
\end{tikzpicture}
```

46.17. Hexagon Inscribed 内接正六边形

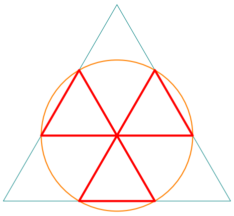
Hexagon Inscribed

To inscribe a regular hexagon in a given equilateral triangle perfectly inside it (no boarders).

内接正六边形

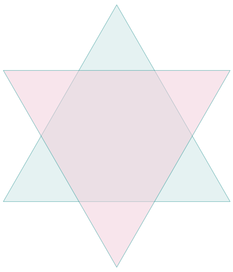
在一个给定的等边三角形内完美地画一个正六边形（没有边框）。

46.17.1. Hexagon Inscribed version 1 内接正六边形版本 1



```
 \begin{tikzpicture}[scale=.5]
  \pgfmathsetmacro{\c}{6}
  \tkzDefPoints{0/0/A,\c/0/B}
  \tkzDefTriangle[equilateral](A,B)\tkzGetPoint{C}
  \tkzDefTriangleCenter[centroid](A,B,C)
  \tkzGetPoint{I}
  \tkzDefPointBy[homothety=center A ratio 1./3](B)
  \tkzGetPoint{c1}
  \tkzInterLC(B,C)(I,c1) \tkzGetPoints{a1}{a2}
  \tkzInterLC(A,C)(I,c1) \tkzGetPoints{b1}{b2}
  \tkzInterLC(A,B)(I,c1) \tkzGetPoints{c1}{c2}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawCircle[thin,orange](I,c1)
  \tkzDrawPolygon[red,thick](a2,a1,b2,b1,c2,c1)
\end{tikzpicture}
```

46.17.2. Hexagon Inscribed version 2 内接正六边形版本 2

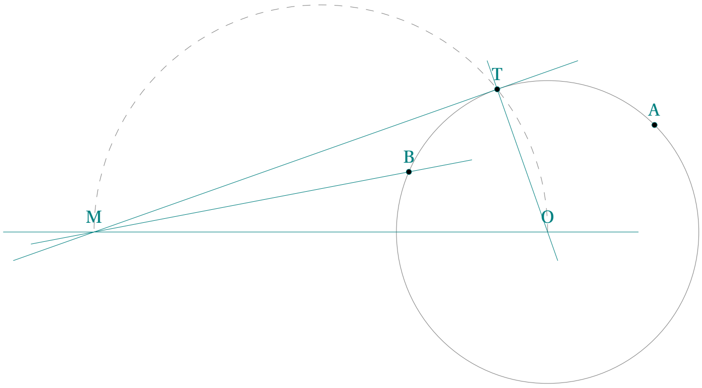


```
\begin{tikzpicture}[scale=.5]
 \pgfmathsetmacro{\c}{6}
 \tkzDefPoints{0/0/A,\c/0/B}
 \tkzDefTriangle[equilateral](A,B)\tkzGetPoint{C}
 \tkzDefTriangleCenter[centroid](A,B,C)
 \tkzGetPoint{I}
 \tkzDefPointsBy[rotation= center I%
                 angle 60](A,B,C){a,b,c}
 \tkzDrawPolygon[fill=teal!20,opacity=.5](A,B,C)
 \tkzDrawPolygon[fill=purple!20,opacity=.5](a,b,c)
\end{tikzpicture}
```

46.18. Power of a point with respect to a circle 一个点相对于一个圆的幂

Power of a point with respect to a circle

$\overline{MA} \times \overline{MB} = MT^2 = MO^2 - OT^2$



```
\begin{tikzpicture}
 \pgfmathsetmacro{\r}{2}%
 \pgfmathsetmacro{\xO}{6}%
 \pgfmathsetmacro{\xE}{\xO-\r}%
 \tkzDefPoints{0/0/M,\xO/0/O,\xE/0/E}
 \tkzDefCircle[diameter](M,O)
 \tkzGetPoint{I}
 \tkzInterCC(I,O)(O,E) \tkzGetPoints{T}{T'}
 \tkzDefShiftPoint[O](45:2){B}
 \tkzInterLC(M,B)(O,E) \tkzGetPoints{A}{B}
 \tkzDrawCircle(O,E)
 \tkzDrawSemiCircle[dashed](I,O)
 \tkzDrawLine(M,O)
 \tkzDrawLines(M,T O,T M,B)
 \tkzDrawPoints(A,B,T)
 \tkzLabelPoints[above](A,B,O,M,T)
\end{tikzpicture}
```

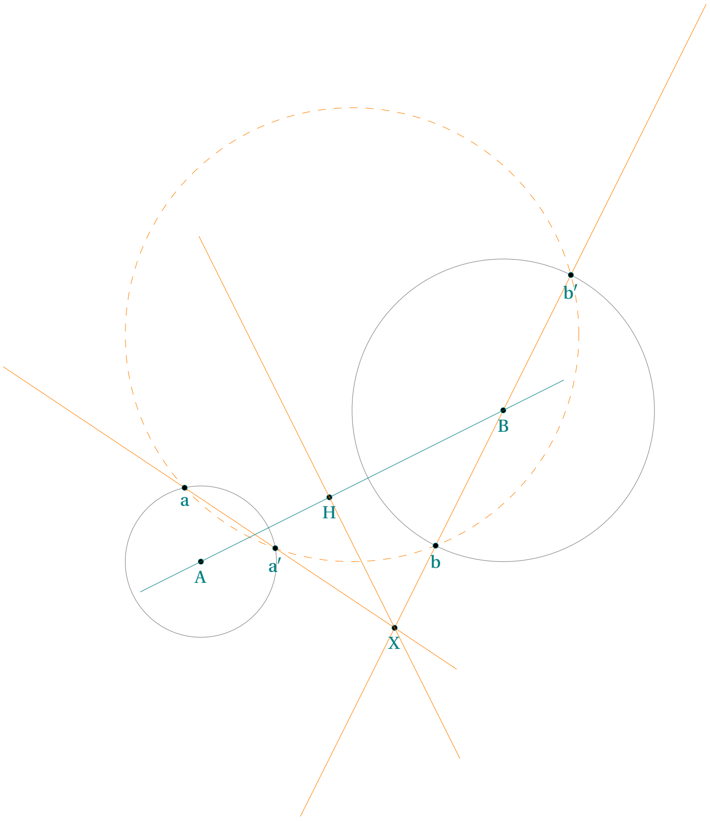
46.19. Radical axis of two non-concentric circles 两个非同心圆的根轴

Radical axis of two non-concentric circles

From Wikipedia : *In geometry, the radical axis of two non-concentric circles is the set of points whose power with respect to the circles are equal. For this reason the radical axis is also called the power line or power bisector of the two circles. The notation radical axis was used by the French mathematician M. Chasles as axe radical.*

两个非同心圆的根轴

来自维基百科：在几何学中，两个非同心圆的根轴是相对于圆的幂相等的点的集合。由于这个原因，根轴也被称为两个圆的幂线或幂平分线。法国数学家 *M.Chasles* 将根轴的符号称为轴根 *axe radical*。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B,2/3/K}
\tkzDefCircle[R](A,1)\tkzGetPoint{a}
\tkzDefCircle[R](B,2)\tkzGetPoint{b}
\tkzDefCircle[R](K,3)\tkzGetPoint{k}
\tkzDrawCircles(A,a B,b)
\tkzDrawCircle[dashed,new](K,k)
\tkzInterCC(A,a)(K,k) \tkzGetPoints{a}{a'}
\tkzInterCC(B,b)(K,k) \tkzGetPoints{b}{b'}
\tkzDrawLines[new,add=2 and 2](a,a')
\tkzDrawLines[new,add=1 and 1](b,b')
\tkzInterLL(a,a')(b,b') \tkzGetPoint{X}
\tkzDefPointBy[projection= onto A--B](X) \tkzGetPoint{H}
\tkzDrawPoints(A,B,H,X,a,b,a',b')
\tkzDrawLine(A,B)
\tkzDrawLine[add= 1 and 2,new](X,H)
\tkzLabelPoints(A,B,H,X,a,b,a',b')
\end{tikzpicture}
```

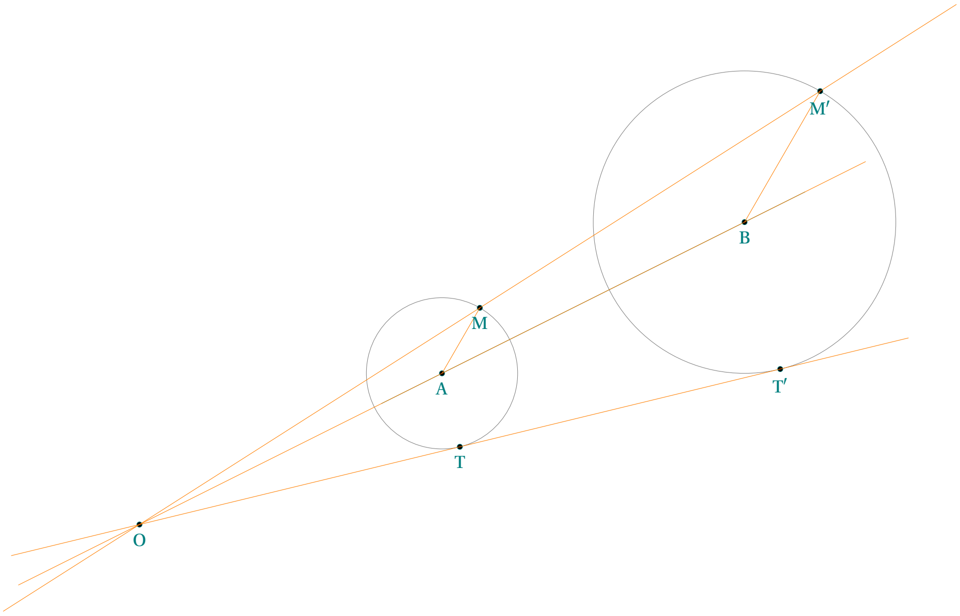
46.20. External homothetic center 外同调中心

External homothetic center

From Wikipedia : *Given two nonconcentric circles, draw radii parallel and in the same direction. Then the line joining the extremities of the radii passes through a fixed point on the line of centers which divides that line externally in the ratio of radii. This point is called the external homothetic center, or external center of similitude (Johnson 1929, pp. 19-20 and 41).*

External homothetic center 外相似中心

来自维基百科：给出两个非同心圆，画出平行且方向一致的半径。然后，连接半径两端的线通过中心线上的一个固定点，该固定点以半径之比在外部划分该线。这个点被称为外部同调中心，或外相似中心（*Johnson 1929*，第 *19-20* 和 *41* 页）。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B,2/3/K}
\tkzDefCircle[R](A,1)\tkzGetPoint{a}
\tkzDefCircle[R](B,2)\tkzGetPoint{b}
\tkzDrawCircles(A,a B,b)
\tkzDrawLine(A,B)
\tkzDefShiftPoint[A](60:1){M}
\tkzDefShiftPoint[B](60:2){M'}
\tkzInterLL(A,B)(M,M') \tkzGetPoint{O}
\tkzDefLine[tangent from = O](B,M') \tkzGetPoints{X}{T'}
\tkzDefLine[tangent from = O](A,M) \tkzGetPoints{X}{T}
\tkzDrawPoints(A,B,O,T,T',M,M')
\tkzDrawLines[new](O,B O,T' O,M')
\tkzDrawSegments[new](A,M B,M')
\tkzLabelPoints(A,B,O,T,T',M,M')
\end{tikzpicture}
```

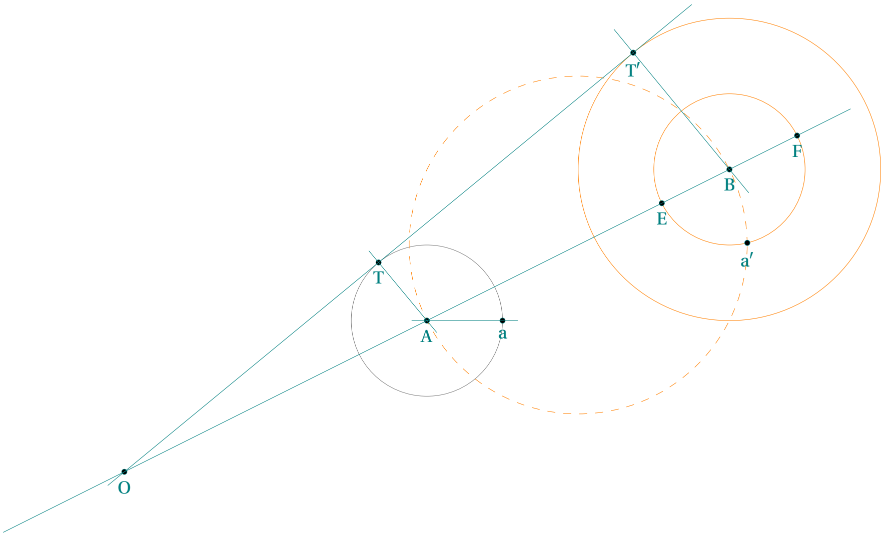
46.21. Tangent lines to two circles 两个圆的切线

Tangent lines to two circles

For two circles, there are generally four distinct lines that are tangent to both if the two circles are outside each other. For two of these, the external tangent lines, the circles fall on the same side of the line; the external tangent lines intersect in the external homothetic center

两个圆的切线

对于两个圆来说，如果两个圆都在对方的外面，一般有四条不同的线与两个圆相切。对于其中的两条线，即外切线，两个圆落在线的同一侧；外切线相交于外相似中心



```
\begin{tikzpicture}
 \pgfmathsetmacro{\r}{1}%
 \pgfmathsetmacro{\R}{2}%
 \pgfmathsetmacro{\rt}{\R-\r}%
 \tkzDefPoints{0/0/A,4/2/B,2/3/K}
 \tkzDefMidPoint(A,B) \tkzGetPoint{I}
 \tkzInterLC[R](A,B)(B,\rt) \tkzGetPoints{E}{F}
 \tkzInterCC(I,B)(B,F) \tkzGetPoints{a}{a'}
 \tkzInterLC[R](B,a)(B,\R) \tkzGetPoints{X'}{T'}
 \tkzDefLine[tangent at=T'](B) \tkzGetPoint{h}
 \tkzInterLL(T',h)(A,B) \tkzGetPoint{O}
 \tkzInterLC[R](O,T')(A,\r) \tkzGetPoints{T}{T}
 \tkzDefCircle[R](A,\r)  \tkzGetPoint{a}
 \tkzDefCircle[R](B,\R)  \tkzGetPoint{b}
 \tkzDefCircle[R](B,\rt)  \tkzGetPoint{c}
 \tkzDrawCircles(A,a)
 \tkzDrawCircles[orange](B,b B,c)
 \tkzDrawCircle[orange,dashed](I,B)
 \tkzDrawPoints(O,A,B,a,a',E,F,T',T)
 \tkzDrawLines(O,B A,a B,T' A,T)
 \tkzDrawLines[add= 1 and 8](T',h)
 \tkzLabelPoints(O,A,B,a,a',E,F,T,T')
\end{tikzpicture}
```

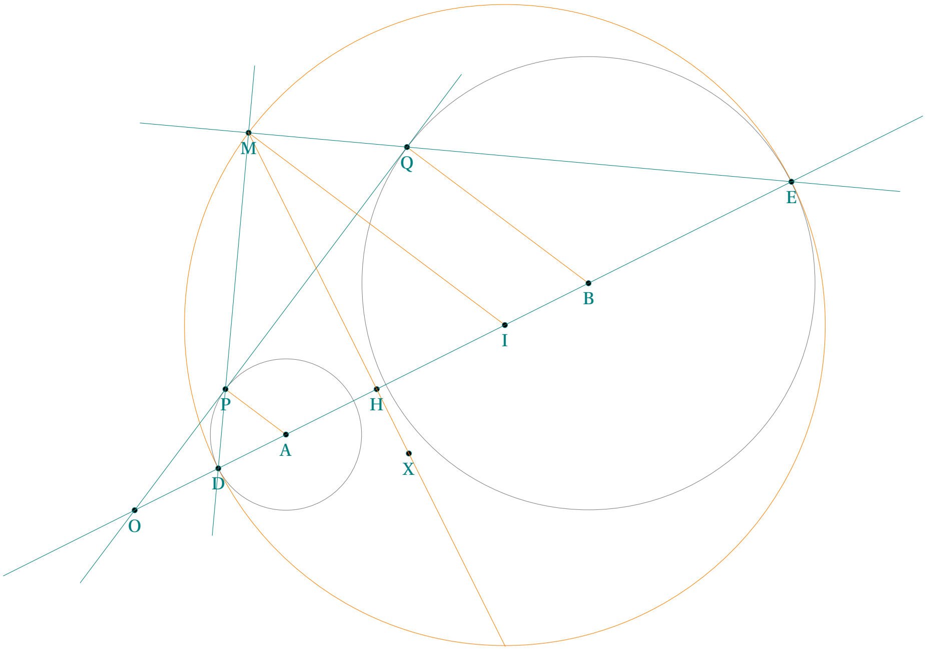
46.22. Tangent lines to two circles with radical axis 两个圆的切线与基轴的切线

Tangent lines to two circles with radical axis

As soon as two circles are not concentric, we can construct their radical axis, the set of points of equal power with respect to the two circles. We know that the radical axis is a line orthogonal to the line of the centers. Note that if we specify P *and* Q *as the points of contact of one of the common exterior tangents with the two circles and* D *and* E *as the points of the circles outside [AB], then (DP) and (EQ) intersect on the radical axis of the two circles. We will show that this property is always true and that it allows us to construct common tangents, even when the circles have the same radius.*

两个圆的切线与基轴的切线

只要两个圆不是同心的，就可以构建它们的根轴，即相对于两个圆的等幂的点的集合。大家知道，根轴是一条与圆心线正交的直线。请注意，如果指定 P 和 Q 为两圆的一条共同外切线的切点，D 和 E 为 *[AB]* 外的圆的点，那么 *(DP)* 和 *(EQ)* 相交于两圆的根轴。我们将证明这一属性始终是真实的，它允许我们构建共同的切线，即使是在圆的半径相同的情况下。



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B,2/3/K}
\tkzDefCircle[R](A,1) \tkzGetPoint{a}
\tkzDefCircle[R](B,3) \tkzGetPoint{b}
\tkzInterCC[R](A,1)(K,3) \tkzGetPoints{a}{a'}
\tkzInterCC[R](B,3)(K,3) \tkzGetPoints{b}{b'}
\tkzInterLL(a,a')(b,b')  \tkzGetPoint{X}
\tkzDefPointBy[projection= onto A--B](X) \tkzGetPoint{H}
\tkzGetPoint{C}
\tkzInterLC[R](A,B)(B,3) \tkzGetPoints{b1}{E}
\tkzInterLC[R](A,B)(A,1) \tkzGetPoints{D}{a2}
\tkzDefMidPoint(D,E) \tkzGetPoint{I}
\tkzDrawCircle[orange](I,D)
\tkzInterLC(X,H)(I,D) \tkzGetPoints{M}{M'}
\tkzInterLC(M,D)(A,D) \tkzGetPoints{P}{P'}
\tkzInterLC(M,E)(B,E) \tkzGetPoints{Q'}{Q}
\tkzInterLL(P,Q)(A,B) \tkzGetPoint{O}
\tkzDrawCircles(A,a B,b)
\tkzDrawSegments[orange](A,P I,M B,Q)
\tkzDrawPoints(A,B,D,E,M,I,O,P,Q,X,H)
\tkzDrawLines(O,E M,D M,E O,Q)
\tkzDrawLine[add= 3 and 4,orange](X,H)
\tkzLabelPoints(A,B,D,E,M,I,O,P,Q,X,H)
\end{tikzpicture}
```

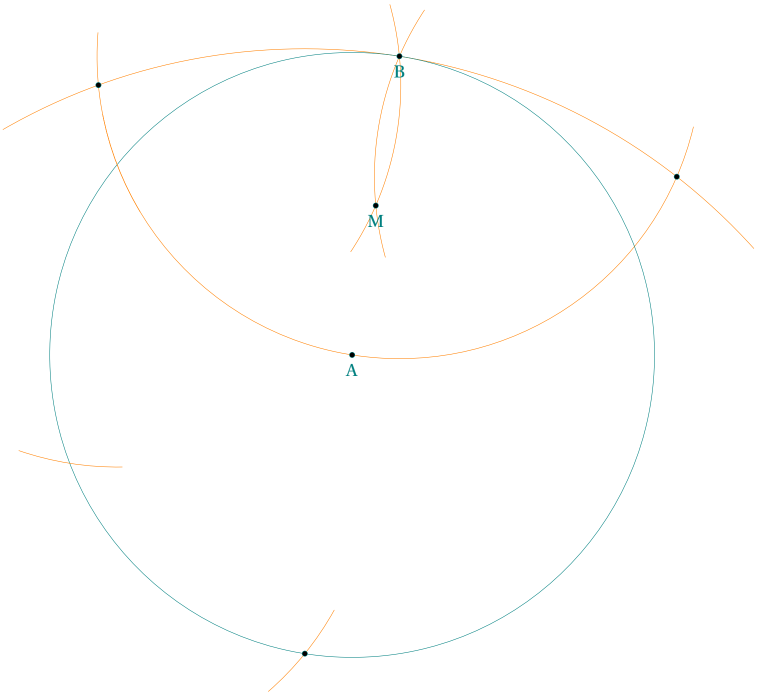
46.23. Middle of a segment with a compass 带圆规的段的中间

Tangent lines to two circles with radical axis

This example involves determining the middle of a segment, using only a compass.

带圆规的段的中间

这个例子涉及到确定一个段的中点，只使用圆规。



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefRandPointOn[circle= center A radius 4]    \tkzGetPoint{B}
\tkzDefPointBy[rotation= center A angle 180](B)  \tkzGetPoint{C}
\tkzInterCC(A,B)(B,A)                            \tkzGetPoints{I}{I'}
\tkzInterCC(A,I)(I,A)                            \tkzGetPoints{J}{B}
\tkzInterCC(B,A)(C,B)                            \tkzGetPoints{D}{E}
\tkzInterCC(D,B)(E,B)                            \tkzGetPoints{M}{M'}
\tkzSetUpArc[color=orange,style=solid,delta=10]
\tkzDrawArc(C,D)(E)
\tkzDrawArc(B,E)(D)
\tkzDrawCircle[color=teal,line width=.2pt](A,B)
\tkzDrawArc(D,B)(M)
\tkzDrawArc(E,M)(B)
\tkzCompasss[color=orange,style=solid](B,I I,J J,C)
\tkzDrawPoints(A,B,C,D,E,M)
\tkzLabelPoints(A,B,M)
\end{tikzpicture}
```

46.24. Definition of a circle _Apollonius_ 圆的定义

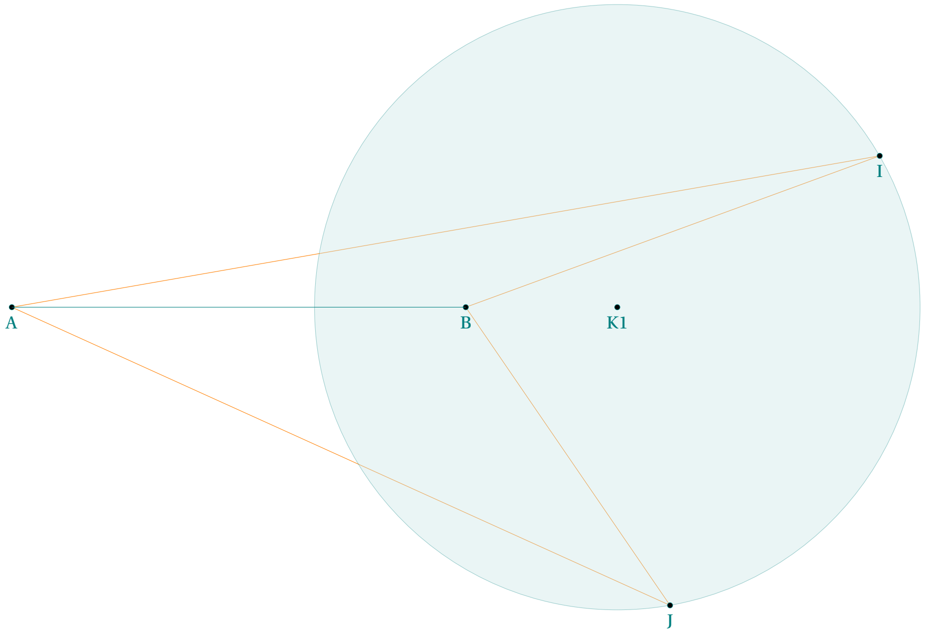
Definition of a circle _Apollonius_

From Wikipedia : *Apollonius showed that a circle can be defined as the set of points in a plane that have a specified ratio of distances to two fixed points, known as foci. This Apollonian circle is the basis of the Apollonius pursuit problem. ... The solutions to this problem are sometimes called the circles of Apollonius.*

阿波罗尼斯圆的定义

来自维基百科：阿波罗尼斯表明，圆可以被定义为平面内与两个固定点（称为焦点）有特定距离比的点的集合。这个阿波罗尼斯圆是阿波罗尼斯追及问题的基础。... 这个问题的解决方案有时被称为阿波罗尼斯的圆。

Explanation
解释
A circle is the set of points in a plane that are equidistant from a given point O. The distance r from the center is called the radius, and the point O is called the center. It is the simplest definition but it is not the only one. Apollonius of Perga gives another definition : The set of all points whose distances from two fixed points are in a constant ratio is a circle.
圆是平面内与给定点 O 等距离的点的集合，离中心的距离 r 称为半径，点 O 称为中心。这是最简单的定义，但它不是唯一的定义。Perga 的 Apollonius 给出了另一个定义：所有与两个固定点的距离成恒定比例的点的集合是一个圆。
With `tkz-euclide` is easy to show you the last definition
使用 `tkz-euclide` 很容易展示最后的定义



```
\begin{tikzpicture}[scale=1.5]
    % Firstly we defined two fixed point.
    % The figure depends of these points and the ratio K
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
    % tkz-euclide.sty knows about the apollonius's circle
    % with K=2 we search some points like  I such as IA=2 x IB
\tkzDefCircle[apollonius,K=2](A,B) \tkzGetPoints{K1}{k}
\tkzDefPointOnCircle[through=  center K1 angle 30 point k]
\tkzGetPoint{I}
\tkzDefPointOnCircle[through= center K1 angle 280  point k]
\tkzGetPoint{J}
\tkzDrawSegments[new](A,I I,B A,J J,B)
\tkzDrawCircle[color = teal,fill=teal!20,opacity=.4](K1,k)
\tkzDrawPoints(A,B,K1,I,J)
\tkzDrawSegment(A,B)
\tkzLabelPoints[below,font=\scriptsize](A,B,K1,I,J)
\end{tikzpicture}
```

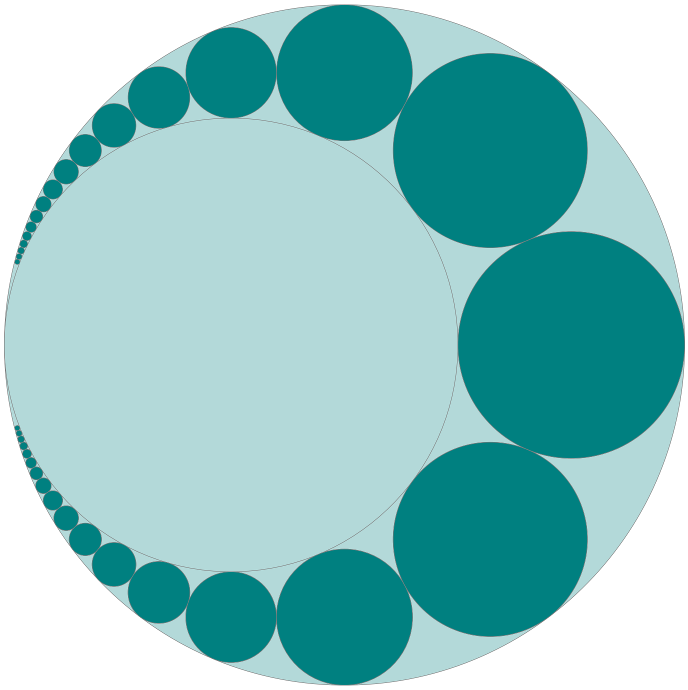
46.25. Application of Inversion : **Pappus chain** 反转的应用

Pappus chain

From Wikipedia *In geometry, the Pappus chain is a ring of circles between two tangent circles investigated by Pappus of Alexandria in the 3rd century AD.*

Pappus chain 帕普斯链

来自维基百科 在几何学中，帕普斯链是两个相切圆之间的圆环，公元 *3* 世纪亚历山大的帕普斯对此进行了研究。



```
\begin{tikzpicture}[ultra thin]
  \pgfmathsetmacro{\xB}{6}%
  \pgfmathsetmacro{\xC}{9}%
  \pgfmathsetmacro{\xD}{(\xC*\xC)/\xB}%
  \pgfmathsetmacro{\xJ}{(\xC+\xD)/2}%
  \pgfmathsetmacro{\r}{\xD-\xJ}%
  \pgfmathsetmacro{\nc}{16}%
  \tkzDefPoints{0/0/A,\xB/0/B,\xC/0/C,\xD/0/D}
  \tkzDefCircle[diameter](A,C) \tkzGetPoint{x}
  \tkzDrawCircle[fill=teal!30](x,C)
  \tkzDefCircle[diameter](A,B) \tkzGetPoint{y}
  \tkzDrawCircle[fill=teal!30](y,B)
  \foreach \i in {-\nc,...,0,...,\nc}
  {\tkzDefPoint(\xJ,2*\r*\i){J}
   \tkzDefPoint(\xJ,2*\r*\i-\r){H}
   \tkzDefCircleBy[inversion = center A through C](J,H)
   \tkzDrawCircle[fill=teal](tkzFirstPointResult,tkzSecondPointResult)}
\end{tikzpicture}
```

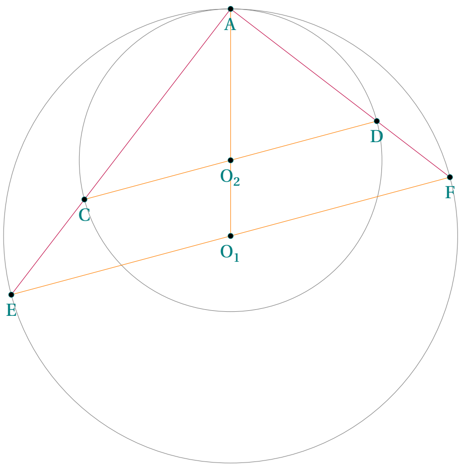
46.26. Book of lemmas proposition 1 Archimedes 阿基米德《公理书》命题 1

Book of lemmas proposition 1 Archimedes

If two circles touch at A, *and if* [CD], [EF] *be parallel diameters in them,* A, C *and* E *are aligned.*

阿基米德《公理书》命题 1

如果两个圆内切于点 A，并且直径 [CD]，[EF] 平行，则 A，C 和 E 共线。



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/O_1,0/1/O_2,0/3/A}
  \tkzDefPoint(15:3){F}
  \tkzInterLC(F,O_1)(O_1,A) \tkzGetSecondPoint{E}
  \tkzDefLine[parallel=through O_2](E,F)
  \tkzGetPoint{x}
  \tkzInterLC(x,O_2)(O_2,A) \tkzGetPoints{D}{C}
  \tkzDrawCircles(O_1,A O_2,A)
  \tkzDrawSegments[new](O_1,A E,F C,D)
  \tkzDrawSegments[purple](A,E A,F)
  \tkzDrawPoints(A,O_1,O_2,E,F,C,D)
  \tkzLabelPoints(A,O_1,O_2,E,F,C,D)
\end{tikzpicture}
```

(CD) ∥ (EF) (AO_1) is secant to these two lines so $\widehat{A0_2C} = \widehat{A0_1E}$.
Since the triangles AO_2C and AO_1E are isosceles the angles at the base are equal $\widehat{AC0_2} = \widehat{AE0_1} = \widehat{CA0_2} = \widehat{EA0_1}$.
Thus A,C and E are aligned

46.27. Book of lemmas proposition 6 Archimedes 阿基米德《公理书》命题 6

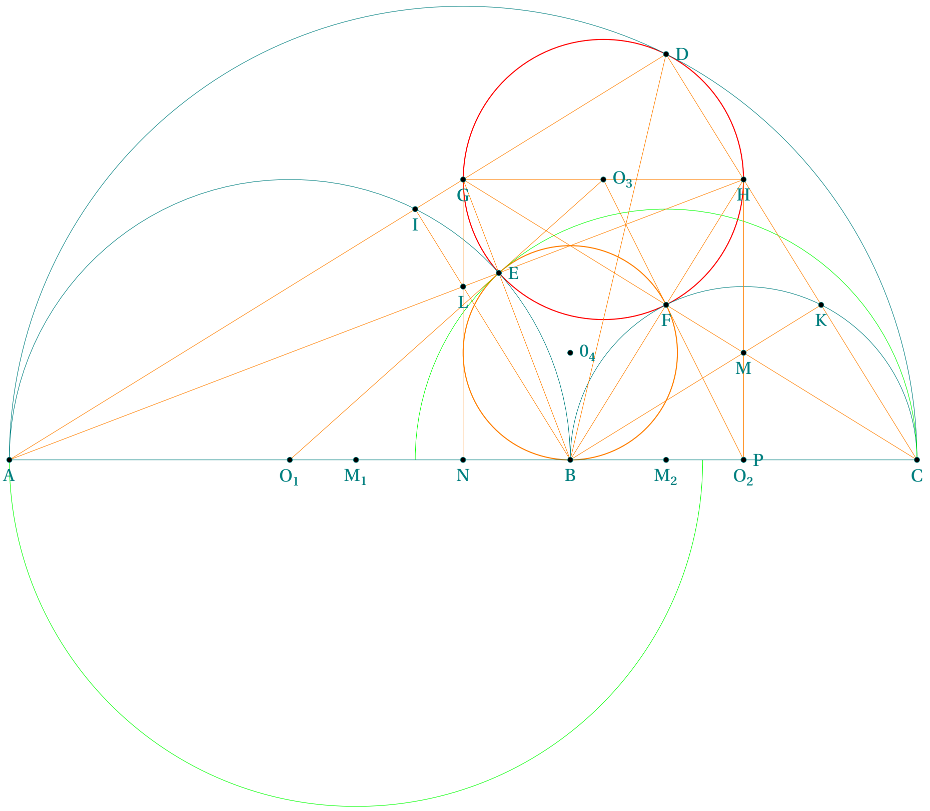
Book of lemmas proposition 6 Archimedes

Let AC*, the diameter of a semicircle, be divided at* B *so that* $AC/AB = \phi$ *or in any ratio. Describe semicircles within the first semicircle and on* AB, BC *as diameters, and suppose a circle drawn touching the all three semicircles. If* GH *be the diameter of this circle, to find relation between* GH *and* AC.

阿基米德《公理书》命题 6

一个半圆的直径 AC，在 B 处被分割，使 AC/AB = phi 或其他比例。在第一个半圆内，以 AB、BC 为直径，并假设画一个圆，接触所有三个半圆。如果 GH 是这个圆的直径，找出 GH 和 AC 之间的关系。

```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,12/0/C}
\tkzDefGoldenRatio(A,C)                  \tkzGetPoint{B}
\tkzDefMidPoint(A,C)                     \tkzGetPoint{O}
\tkzDefMidPoint(A,B)                     \tkzGetPoint{O_1}
\tkzDefMidPoint(B,C)                     \tkzGetPoint{O_2}
\tkzDefExtSimilitudeCenter(O_1,A)(O_2,B) \tkzGetPoint{M_0}
\tkzDefIntSimilitudeCenter(O,A)(O_1,A)   \tkzGetPoint{M_1}
\tkzDefIntSimilitudeCenter(O,C)(O_2,C)   \tkzGetPoint{M_2}
\tkzInterCC(O_1,A)(M_2,C)                \tkzGetFirstPoint{E}
\tkzInterCC(O_2,C)(M_1,A)                \tkzGetSecondPoint{F}
\tkzInterCC(O,A)(M_0,B)                  \tkzGetFirstPoint{D}
\tkzInterLL(O_1,E)(O_2,F)                \tkzGetPoint{O_3}
\tkzDefCircle[circum](E,F,B)             \tkzGetPoint{0_4}
\tkzInterLC(A,D)(O_1,A)                  \tkzGetFirstPoint{I}
\tkzInterLC(C,D)(O_2,B)                  \tkzGetSecondPoint{K}
\tkzInterLC[common=D](A,D)(O_3,D)        \tkzGetFirstPoint{G}
\tkzInterLC[common=D](C,D)(O_3,D)        \tkzGetFirstPoint{H}
\tkzInterLL(C,G)(B,K)                    \tkzGetPoint{M}
\tkzInterLL(A,H)(B,I)                    \tkzGetPoint{L}
\tkzInterLL(L,G)(A,C)                    \tkzGetPoint{N}
\tkzInterLL(M,H)(A,C)                    \tkzGetPoint{P}
\tkzDrawCircles[red,thin](O_3,F)
\tkzDrawCircles[new,thin](0_4,B)
\tkzDrawSemiCircles[teal](O,C O_1,B O_2,C)
\tkzDrawSemiCircles[green](M_2,C)
\tkzDrawSemiCircles[green,swap](M_1,A)
\tkzDrawSegment(A,C)
\tkzDrawSegments[new](O_1,O_3 O_2,O_3)
\tkzDrawSegments[new,very thin](B,H C,G A,H G,N H,P)
\tkzDrawSegments[new,very thin](B,D A,D C,D G,H I,B K,B B,G)
\tkzDrawPoints(A,B,C,M_1,M_2,E,O_3,F,D,0_4,O_1,O_2,I,K,G,H,L,P,N,M)
\tkzLabelPoints[font=\scriptsize](A,B,C,M_1,M_2,F,O_1,O_2,I,K,G,H,L,M,N)
\tkzLabelPoints[font=\scriptsize,right](E,O_3,D,0_4,P)
\end{tikzpicture}
```



Let GH be the diameter of the circle which is parallel to AC, and let the circle touch the semicircles on AC, AB, BC in D, E, F respectively.

设GH为平行于AC的圆的直径，并设该圆分别在D、E、F中与AC、AB、BC上的半圆相切。

Then, by Prop. 1 A,G and D are aligned, ainsi que D, H and C.

For a like reason A E and H are aligned, C F and Gare aligned, as also are B E and G, B F and H.

然后，根据提议1，A、G和D是共线的，同样D、H和C也是。

由于同样的原因，A、E和H是共线的，C、F和G是共线的，B、E和G，B、F和H也是一样。

Let (AD) meet the semicircle on [AC] at I, and let (BD) meet the semicircle on [BC] in K. Join CI, CK meeting AE, BF in L, M, and let GL, HM produced meet AB in N, P respectively.

让(AD)在I处与[AC]的半圆相交，让(BD)在K处与[BC]的半圆相交。让CI、CK在L、M中与AE、BF相交，让GL、HM在N、P中分别与AB相交。

Now, in the triangle AGB, the perpendiculars from A, C on the opposite sides meet in L. Therefore by the properties of triangles, (GN) is perpendicular to (AC). Similarly (HP) is perpendicular to (BC).

Again, since the angles at I, K, D are right, (CK) is parallel to (AD), and (CI) to (BD).

在三角形AGB中，来自A、C对边的垂直线在L中相交。因此根据三角形的特性，(GN)与(AC)垂直。

同样地，(HP)也与(BC)垂直。

同样，由于I、K、D的构成直角，(CK)与(AD)平行，而(CI)与(BD)平行。

Therefore

$$\frac{AB}{BC}=\frac{AL}{LH}=\frac{AN}{NP} \quad \text{and} \quad \frac{BC}{AB}=\frac{CM}{MG}=\frac{PC}{NP}$$

hence

$$\frac{AN}{NP}=\frac{NP}{PC} \quad \text{so} \quad NP^2=AN\times PC$$

Now suppose that B divides [AC] according to the divine proportion that is :

现在假设B按照神圣的比例分割[AC]，即：

$$\phi = \frac{\mathrm{AB}}{\mathrm{BC}} = \frac{\mathrm{AC}}{\mathrm{AB}} \quad \text{then} \quad \mathrm{AN} = \phi \mathrm{NPand} \quad \mathrm{NP} = \phi \mathrm{PC}$$

We have

$$\mathrm{AC} = \mathrm{AN} + \mathrm{NP} + \mathrm{PC} \quad \text{either} \quad \mathrm{AB} + \mathrm{BC} == \mathrm{AN} + \mathrm{NP} + \mathrm{PC} \quad \text{or} \quad (\phi + 1)\mathrm{BC} = \mathrm{AN} + \mathrm{NP} + \mathrm{PC}$$

we get

$$(\phi + 1)\mathrm{BC} = \phi \mathrm{NP} + \mathrm{NP} + \mathrm{PC} = (\phi + 1)\mathrm{NP} + \mathrm{PC} = \phi(\phi + 1)\mathrm{PC} + \mathrm{PC} = \phi^2 + \phi + 1)\mathrm{PC}$$

as

$$\phi^2 = \phi + 1 \quad \text{then} \quad (\phi + 1)\mathrm{BC} = 2(\phi + 1)\mathrm{PC} \quad \text{i.e.} \quad \mathrm{BC} = 2\mathrm{PC}$$

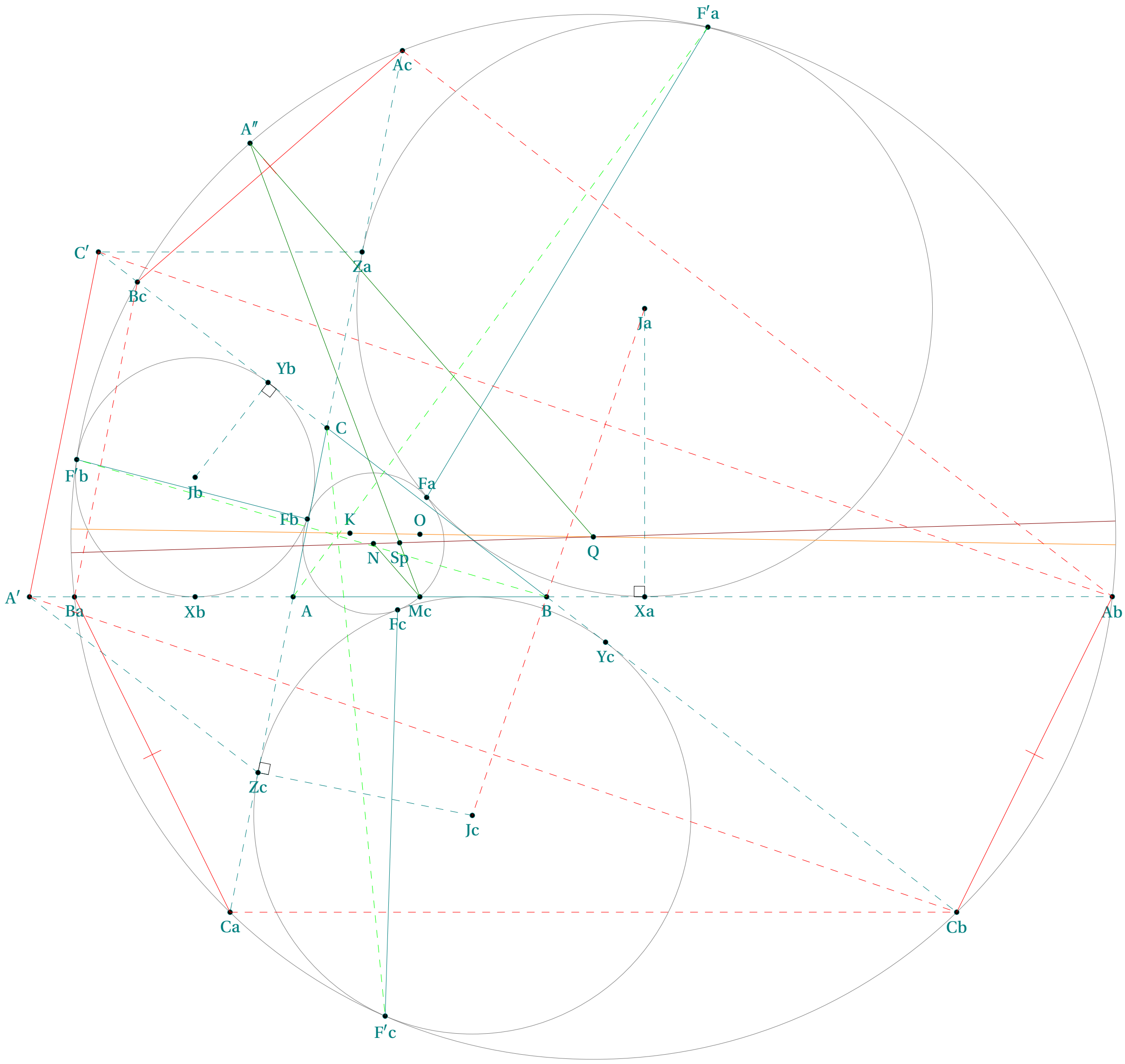
That is, p is the middle of the segment BC.
就是说，p 是线段 BC 的中点。
Part of the proof from https://www.cut-the-knot.org

46.28. "The" Circle of APOLLONIUS

The Apollonius circle of a triangle _Apollonius_

The circle which touches all three excircles of a triangle and encompasses them is often known as ”the” Apollonius circle (Kimberling 1998, p. 102)

Explanation
解释
The purpose of the first examples was to show the simplicity with which we could recreate these propositions. With TikZ you need to do calculations and use trigonometry while with `tkz-euclide` you only need to build simple objects
第一个例子的目的是显示可以简单地重新创建这些命题。使用 TikZ，需要进行计算和使用三角学，而使用 `tkz-euclide`，只需要建立简单的对象。
But don’t forget that behind or far above `tkz-euclide` there is TikZ. I’m only creating an interface between TikZ and the user of my package.
但别忘了，在 `tkz-euclide` 的背后或远处，还有 TikZ。我只是在 TikZ 和宏包的用户之间建立一个接口。
The last example is very complex and it is to show you all that we can do with `tkz-euclide`.
最后一个例子非常复杂，是为了展示能用 `tkz-euclide` 做的一切。


F′a
Ac
A″
C′
Za
Bc
Ja
Yb
C
F′b
Jb
Fa
Fb
K
O
N
Sp
Q
A′
Ba
Xb
A
Fc
Mc
B
Xa
Ab
Yc
Zc
Jc
Ca
Cb
F′c

```
\begin{tikzpicture}[scale=.6]
\tkzDefPoints{0/0/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[ortho](A,B,C)        \tkzGetPoint{H}
\tkzDefSpcTriangle[excentral,name=J](A,B,C){a,b,c}
\tkzDefSpcTriangle[centroid,name=M](A,B,C){a,b,c}
\tkzDefCircle[in](Ma,Mb,Mc)                \tkzGetPoint{Sp}   % Sp Spieker center
\tkzDefProjExcenter[name=J](A,B,C)(a,b,c){Y,Z,X}
\tkzDefLine[parallel=through Za](A,B)      \tkzGetPoint{Xc}
\tkzInterLL(Za,Xc)(C,B)                    \tkzGetPoint{C'}
\tkzDefLine[parallel=through Zc](B,C)      \tkzGetPoint{Ya}
\tkzInterLL(Zc,Ya)(A,B)                    \tkzGetPoint{A'}
\tkzDefPointBy[reflection= over Ja--Jc](C')\tkzGetPoint{Ab}
\tkzDefPointBy[reflection= over Ja--Jc](A')\tkzGetPoint{Cb}
\tkzInterLL(K,O)(N,Sp)                     \tkzGetPoint{Q}
\tkzInterLC(A,B)(Q,Cb)                     \tkzGetFirstPoint{Ba}
\tkzInterLC(A,C)(Q,Cb)                     \tkzGetPoints{Ac}{Ca}
\tkzInterLC(B,C')(Q,Cb)                    \tkzGetFirstPoint{Bc}
\tkzInterLC[next to=Ja](Ja,Q)(Q,Cb)        \tkzGetFirstPoint{F'a}
\tkzInterLC[next to=Jc](Jc,Q)(Q,Cb)        \tkzGetFirstPoint{F'c}
\tkzInterLC[next to=Jb](Jb,Q)(Q,Cb)        \tkzGetFirstPoint{F'b}
\tkzInterLC[common=F'a](Sp,F'a)(Ja,F'a)    \tkzGetFirstPoint{Fa}
\tkzInterLC[common=F'b](Sp,F'b)(Jb,F'b)    \tkzGetFirstPoint{Fb}
\tkzInterLC[common=F'c](Sp,F'c)(Jc,F'c)    \tkzGetFirstPoint{Fc}
\tkzInterLC(Mc,Sp)(Q,Cb)                   \tkzGetFirstPoint{A''}
\tkzDefCircle[euler](A,B,C)                \tkzGetPoints{E}{e}
\tkzDefCircle[ex](C,A,B)                   \tkzGetPoints{Ea}{a}
\tkzDefCircle[ex](A,B,C)                   \tkzGetPoints{Eb}{b}
\tkzDefCircle[ex](B,C,A)                   \tkzGetPoints{Ec}{c}
% Calculations are done, now you can draw, mark and label
\tkzDrawCircles(Q,Cb E,e)%
\tkzDrawCircles(Eb,b Ea,a Ec,c)
\tkzDrawPolygon(A,B,C)
\tkzDrawSegments[dashed](A,A' C,C' A',Zc Za,C' B,Cb B,Ab A,Ca)
\tkzDrawSegments[dashed](C,Ac Ja,Xa Jb,Yb Jc,Zc)
\begin{scope}
   \tkzClipCircle(Q,Cb) % We limit the drawing of the lines
   \tkzDrawLine[add=5 and 12,orange](K,O)
   \tkzDrawLine[add=12 and 28,red!50!black](N,Sp)
\end{scope}
\tkzDrawPoints(A,B,C,K,Ja,Jb,Jc,Q,N,O,Sp,Mc,Xa,Xb,Yb,Yc,Za,Zc)
\tkzDrawPoints(A',C',A'',Ab,Cb,Bc,Ca,Ac,Ba,Fa,Fb,Fc,F'a,F'b,F'c)
\tkzLabelPoints(Ja,Jb,Jc,Q,Xa,Xb,Za,Zc,Ab,Cb,Bc,Ca,Ac,Ba,F'b)
\tkzLabelPoints[above](O,K,F'a,Fa,A'')
\tkzLabelPoints[below](B,F'c,Yc,N,Sp,Fc,Mc)
\tkzLabelPoints[left](A',C',Fb)
\tkzLabelPoints[right](C)
\tkzLabelPoints[below right](A)
\tkzLabelPoints[above right](Yb)
\tkzDrawSegments(Fc,F'c Fb,F'b Fa,F'a)
\tkzDrawSegments[color=green!50!black](Mc,N Mc,A'' A'',Q)
\tkzDrawSegments[color=red,dashed](Ac,Ab Ca,Cb Ba,Bc Ja,Jc A',Cb C',Ab)
\tkzDrawSegments[color=red](Cb,Ab Bc,Ac Ba,Ca A',C')
\tkzMarkSegments[color=red,mark=|](Cb,Ab Bc,Ac Ba,Ca)
\tkzMarkRightAngles(Jc,Zc,A Ja,Xa,B Jb,Yb,C)
\tkzDrawSegments[green,dashed](A,F'a B,F'b C,F'c)
\end{tikzpicture}
```

Part X.

FAQ 问答

47. FAQ ????

47.1. Most common errors ??????

For the moment, I'm basing myself on my own, because having changed syntax several times, I've made a number of mistakes. This section is going to be expanded. With version 4.05 new problems may appear. ???4.05????????????????

- The mistake I make most often is to forget to put an "s" in the macro used to draw more than one object: like `\tkzDrawSegment(s)` or `\tkzDrawCircle(s)` ok like in this example `\tkzDrawPoint(A,B)` when you need `\tkzDrawPoints(A,B)`;

- ?????????????????????????? "s"??`\tkzDrawSegment(s)`?`\tkzDrawCircle(s)` ?????????????`\tkzDrawPoints(A,B)`?????????? "s" `\tkzDrawPoint(A,B)`?

- Don't forget that since version 4 the unit is obligatorily the "cm" it is thus necessary to withdraw the unit like here `\tkzDrawCircle[R](O,3cm)` which becomes `\tkzDrawCircle[R](O,3)`. The traditional options of `TikZ` keep their units example `below right = 12pt` on the other hand one will write `size=1.2` to position an arc in `\tkzMarkAngle`;

- ???????4????????? "cm"???????????????? `\tkzDrawCircle[R](O,3cm)` ??? `\tkzDrawCircle[R](O,3)`?TikZ??????????????? `below right = 12pt`?????????? `\tkzMarkAngle`?? `size=1.2`????????

- The following error still happens to me from time to time. A point that is created has its name in brackets while a point that is used either as an option or as a parameter has its name in braces. Example `\tkzGetPoint(A)` When defining an object, use braces and not brackets, so write: `\tkzGetPoint{A}`;

- ??`\tkzGetPoint(A)` ??????????????????????????? `\tkzGetPoint{A}`?

- The changes in obtaining the points of intersection between lines and circles sometimes exchange the solutions, this leads either to a bad figure or to an error.

- ???????????????????????????????????

- `\tkzGetPoint{A}` in place of `\tkzGetFirstPoint{A}`. When a macro gives two points as results, either we retrieve these points using `\tkzGetPoints{A}{B}`, or we retrieve only one of the two points, using `\tkzGetFirstPoint{A}` or `\tkzGetSecondPoint{A}`. These two points can be used with the reference `tkzFirstPointResult` or `tkzSecondPointResult`. It is possible that a third point is given as `tkzPointResult`;

- `\tkzGetPoint{A}`??`\tkzGetFirstPoint{A}`????????????????????`\tkzGetPoints{A}{B}`??????????`\tkzGetFirstPoint{A}`? `tkzPointResult`??????

- Mixing options and arguments; all macros that use a circle need to know the radius of the circle. If the radius is given by a measure then the option includes a `R`.

- ??? R?

- The angles are given in degrees, more rarely in radians.

- ???????????????

– If an error occurs in a calculation when passing parameters, then it is better to make these calculations before calling the macro.

– ?????????????????????????????????

– Do not mix the syntax of `pgfmath` and `xfp`. I've often chosen `xfp` but if you prefer pgfmath then do your calculations before passing parameters.

– ?????? `pgfmath` ? `xfp` ????????? `xfp`??????? pgfmath???????????????

– Error "dimension too large" : In some cases, this error occurs. One way to avoid it is to use the "**`veclen`**" option. When this option is used in an scope, the "veclen" function is replaced by a function dependent on "xfp". Do not use intersection macros in this scope. For example, an error occurs if you use the macro `\tkzDrawArc` with too small an angle. The error is produced by the `decoration` library when you want to place a mark on an arc. Even if the mark is absent, the error is still present.

– ?? "????"????????????????????????????"**`veclen`**"??????????????????"veclen "???????? "xfp "??????? ????????????????????????????`\tkzDrawAr` `decoration`????????????????????

Index