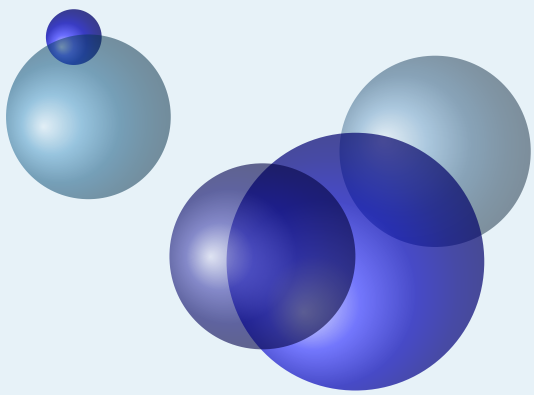
tkz-euclide 5.10c

AlterMundus

tkz-euclide v5

Géométrie euclidienne



Alain Matthes

29 avril 2024 Documentation V.5.10c

http://altermundus.fr

tkz-euclide

Alain Matthes

AlterMundus

☞ **tkz-euclide** passe en version 5 avec la possibilité d'effectuer une partie des calculs en utilisant `lua`. Consultez les sections `News` et `Lua` pour plus d'informations.

tkz-euclide est un ensemble de macros pratiques pour dessiner dans un plan (objet bidimensionnel fondamental) avec un système de coordonnées cartésiennes. Il gère les situations les plus classiques en géométrie euclidienne. **tkz-euclide** est construit sur la base de PGF et de son front-end associé Ti*k*Z et est un package de dessin (La)TeX-friendly. L'objectif est de fournir une interface utilisateur de haut niveau pour construire des graphiques relativement simplement. L'idée est de vous permettre de suivre étape par étape une construction qui serait faite à la main aussi naturellement que possible.

☞ Tout d'abord, je voudrais remercier **Till Tantau** pour le magnifique paquetage LaTeX, à savoir Ti*k*Z.

☞ Remerciements : J'ai reçu de précieux conseils, remarques, corrections et exemples de la part de
`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` pour son travail sur un rapporteur, `Paul Gaborit`, `Laurent Van Deik` pour toutes ses corrections, remarques et questions et `Muzimuzhi Z` pour le code concernant l'option "dim". Un grand merci à `Chetan Shirore` et `Dr. Ajit Kumar` car leur travail sur les nombres complexes dans leur package `luamaths` m'a beaucoup aidé.

☞ Je tiens également à remercier Eric Weisstein, créateur de MathWorld : MathWorld.

☞ Vous trouverez quelques exemples sur mon site : altermundus.fr. en cours de construction !

Veuillez signaler les fautes de frappe ou tout autre commentaire sur cette documentation à l'adresse suivante : Alain Matthes.

Table des matières

Première partie

Étude générale : une étude brève mais complète

Actualités et compatibilité

0.1. With 5.10 version

- Documentation en français ajoutée
- Option `mini` ajoutée. Vous pouvez utiliser cette option avec le package `tkz-elements`. Seuls les modules nécessaires pour les tracés seront chargés. Cette option est actuellement intéressante que si vous utilisez `tkz-elements`.

0.2. Avec la version 5.06

- Correction d'un bug avec la macro `\tkzLabelAngle` et l'option "angle"
- Ajout de `\tkzSetUpCircle`
- Correction de quelques fautes de frappe

0.3. Avec la version 5.05

Correction de la documentation dans l'exemple complet mais minimal.

0.4. Avec la version 5.03

- Correction d'un bug dans la macro `\tkzDefBarycentricPointTwo` du fichier tkz-obj-lua-points-spc.tex;
- Ajout de la macro `\tkzDrawEllipse`;
- Suppression des macros `\tkzDrawSectorAngles` et `\tkzDrawSectorRwithNodesAngles`.

Voici la correction de votre texte :

0.5. Avec la version 5.0

Avec la version 4, certains changements ont été apportés pour rendre la syntaxe plus homogène, et surtout pour distinguer la définition et la recherche de coordonnées du reste, c'est-à-dire le dessin, le marquage et l'étiquetage. Maintenant, les macros de définition sont isolées, ce qui facilitera l'introduction d'une phase de calcul de coordonnées en utilisant Lua.

- Enfin, j'ai ajouté l'option `lua` pour le package **`tkz-euclide`**. Cela permet d'effectuer les calculs pour les fonctions principales en utilisant Lua (voir 1). La syntaxe reste inchangée. Rien ne change pour l'utilisateur à l'exception de la compilation qui doit se faire à l'aide de `LuaLaTeX`;
- L'option `xfp` est devenue `veclen`, voir 26.14.

Les modifications incluent la correction de la ponctuation, l'ajustement de la formulation pour une meilleure clarté et la correction de quelques erreurs typographiques mineures.

1. Travailler avec lua

1.0.1. Option lua

Vous pouvez désormais utiliser l'option lua avec la version 5 de **tkz-euclide**. Il vous suffit d'écrire dans votre préambule
usepackage[lua]{tkz-euclide}. Évidemment, vous devrez compiler avec LuaLaTeX. Rien ne change pour la syntaxe.
Sans l'option, vous pouvez utiliser **tkz-euclide** avec le code proposé de la version 4.25.
Cette version n'est pas encore finalisée bien que la documentation que vous lisez actuellement ait été compilée avec cette option.
Quelques informations sur la méthode utilisée et les résultats obtenus. En ce qui concerne la méthode, j'ai envisagé deux possibilités. La première consistait simplement à remplacer partout où je le pouvais les calculs effectués par xfp ou parfois par lua. C'est ainsi que je suis passé de fp à xfp et maintenant à lua. La deuxième possibilité, plus ambitieuse, aurait été d'associer à chaque point un nombre complexe et de faire les calculs sur les complexes avec lua. Malheureusement, pour cela, je dois utiliser des bibliothèques pour lesquelles je ne connais pas la licence.
Sinon, les résultats sont bons. Cette documentation avec LualaTeX et xfp se compile en 47s, tandis qu'avec lua, cela ne prend que 30s pour 236 pages.
Un autre document de 61 pages est compilé en 16s avec pdflaTeX et xfp et en 13s avec LualaTeX et xfp.
Cette documentation se compile avec \usepackage{tkz-base} et \usepackage[lua]{tkz-euclide} mais je n'ai pas testé toutes les interactions en détail.

1.0.2. Option mini

Lorsque vous utilisez tkz-elements uniquement pour déterminer les points de vos figures, il n'est pas nécessaire de charger tous les modules de **tkz-euclide**. Dans ce cas, en utilisant l'option mini \usepackage[mini]{tkz-euclide}, vous ne chargerez que les modules nécessaires aux tracés.

2. Installation

tkz-euclide est sur le serveur du CTAN [1]. Si vous souhaitez tester une version bêta, il vous suffit de placer les fichiers suivants dans un dossier texmf que votre système peut trouver. Vous devrez vérifier plusieurs points :

— Le dossier **tkz-euclide** doit être situé sur un chemin reconnu par **latex**.

— **tkz-euclide** utilise xfp.

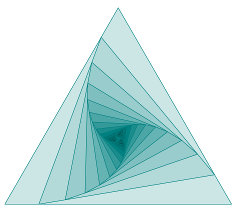
— Vous devez avoir **PGF** installé sur votre ordinateur. **tkz-euclide** utilise plusieurs bibliothèques de Ti*k*Z

1. **tkz-euclide** fait partie de TeXLive et **tlmgr** vous permet de les installer. Ce package fait également partie de MiKTeX sous Windows.

angles,
arrows,
arrows.meta,
calc,
decorations,
decorations.markings,
decorations.pathreplacing,
decorations.shapes,
decorations.text,
decorations.pathmorphing,
intersections,
math,
plotmarks,
positioning,
quotes,
shapes.misc,
through

— Cette documentation et tous les exemples ont été obtenus avec **lualatex**, mais **pdflatex** ou **xelatex** devraient également convenir.

3. Présentation et aperçu



```
\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}
```

3.1. Pourquoi tkz-euclide?

Mon objectif initial était de fournir aux autres enseignants de mathématiques et à moi-même un outil pour créer rapidement des figures de géométrie euclidienne sans investir trop d'efforts dans l'apprentissage d'un nouveau langage de programmation. Bien sûr, **tkz-euclide** s'adresse aux enseignants de mathématiques qui utilisent LaTeX et permet de créer facilement des dessins corrects grâce à LaTeX.
Il est apparu que la méthode la plus simple était de reproduire celle utilisée pour obtenir une construction à la main. Pour décrire une construction, il faut bien sûr définir les objets mais aussi les actions que vous effectuez. Il me semblait que la syntaxe proche du langage des mathématiciens et de leurs élèves serait plus facilement compréhensible; de plus, il me semblait également que cette syntaxe devrait être proche de celle de LaTeX. Les objets, bien sûr, sont des points, des segments, des droites, des triangles, des polygones et des cercles. Quant aux actions, j'en ai considéré cinq comme suffisantes, à savoir : définir, créer, dessiner, marquer et étiqueter.
La syntaxe est peut-être un peu verbeuse mais je crois qu'elle est facilement accessible. En conséquence, les étudiants comme les enseignants ont pu accéder facilement à cet outil.

3.2. TikZ vs tkz-euclide

J'adore programmer avec Ti*k*Z, et sans Ti*k*Z, je n'aurais jamais eu l'idée de créer **tkz-euclide**, mais n'oubliez jamais qu'il y a Ti*k*Z derrière, et qu'il est toujours possible d'insérer du code de Ti*k*Z. **tkz-euclide** ne vous empêche pas d'utiliser Ti*k*Z. Cela dit, je ne pense pas que mélanger les syntaxes soit une bonne chose.

Il n'est pas nécessaire de comparer TikZ et **tkz-euclide**. Ce dernier ne s'adresse pas au même public que TikZ. Le premier vous permet de faire beaucoup de choses, le second se contente de dessiner des figures géométriques. Le premier peut faire tout ce que le second fait, mais le second fera plus facilement ce que vous voulez.
Le but principal est de définir des points pour créer des figures géométriques. **tkz-euclide** vous permet de dessiner les objets essentiels de la géométrie euclidienne à partir de ces points, mais cela peut être insuffisant pour certaines actions comme la coloration des surfaces. Dans ce cas, vous devrez utiliser TikZ, ce qui est toujours possible.
Voici quelques comparaisons entre **TikZ** et **tkz-euclide** 4. Pour cela, je vais utiliser les exemples de géométrie du manuel PGF. Les deux outils euclidiens les plus importants utilisés par les premiers Grecs pour construire différentes formes géométriques et angles étaient un compas et une règle. Mon idée est de vous permettre de suivre étape par étape une construction qui serait faite à la main (avec un compas et une règle) aussi naturellement que possible.

3.2.1. Book I, proposition I _Euclid's Elements_

Livre I, proposition _Les éléments d'Euclide_

Construire un triangle équilatéral sur une droite finie donnée.

Explanation :
Le quatrième tutoriel du *PgfManual* est consacré aux constructions géométriques. *T. Tantau* propose d'obtenir le dessin avec son magnifique outil TikZ. Je propose ici la même construction avec *tkz-elements*. La couleur du code TikZ est green!50!black et celle de *tkz-elements* est rouge.

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

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

Comment obtenir la droite AB ? Pour obtenir cette droite, nous utilisons deux points fixes.

```
\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){$B$}
```

On veut tracer un cercle autour des points A et B dont le rayon est donné par la longueur de la droite AB.

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

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

L'intersection des cercles $\mathcal{D}$ et $\mathcal{E}$.

```
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}
```

Comment dessiner des points :

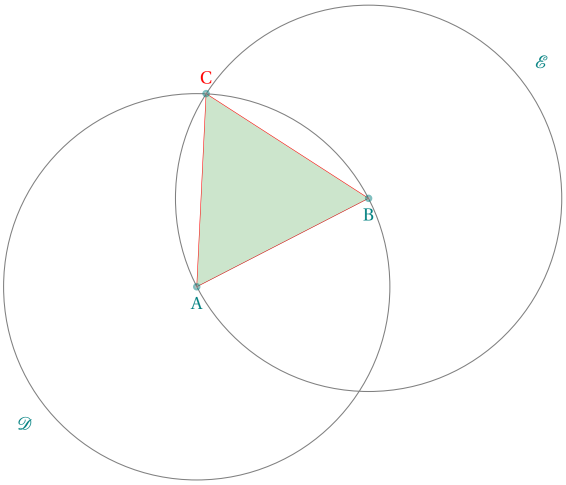
```
\foreach \point in {A,B,C}
\fill [black,opacity=.5] (\point) circle (2pt);
```

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

3.2.2. Complete code with tkz-euclide

Nous devons définir les couleurs

```
\colorlet{input}{red!80!black}
\colorlet{output}{red!70!black}
\colorlet{triangle}{green!50!black!40}
```




```
\colorlet{input}{red!80!black}
\colorlet{output}{red!70!black}
\colorlet{triangle}{green!50!black!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. Livre I, Proposition II _Les Éléments d'Euclide_

Livre I, Proposition II _Les Éléments d'Euclide_

Placer une droite égale à une droite donnée dont l'une des extrémités se trouve en un point donné.

Explication
Dans la première partie, nous devons trouver le point médian de la droite AB. Avec Ti*k*Z nous pouvons utiliser la bibliothèque calc

```
\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) $) {};
```

Avec `tkz-euclide` nous avons une macro `\tkzDefMidPoint`, nous obtenons le point X avec `\tkzGetPoint` mais nous n'avons pas besoin de ce point pour passer à l'étape suivante.

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

Ensuite, nous devons construire un triangle équilatéral. C'est facile avec `tkz-euclide` . Avec TikZ, il faut faire un effort car il faut utiliser le point médian X pour obtenir le point D en calculant la trigonométrie.

```
\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}
```

Nous pouvons dessiner le triangle à l'extrémité de l'image avec

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

Nous savons comment tracer le cercle $\mathcal{H}$ autour de B en passant par C et comment placer les points E et 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)
```

Nous pouvons placer les points E et F à la fin de l'image. Nous n'en avons pas besoin pour l'instant.
Intersection d'une droite et d'un cercle : nous cherchons ici l'intersection du cercle autour de B passant par C et de la droite DB. La droite infinie DB intercepte le cercle mais avec Ti*k*Z nous devons prolonger les droites DB et cela peut être fait en utilisant des calculs partiels. Nous obtenons le point F et BF ou DF intercepte le cercle.

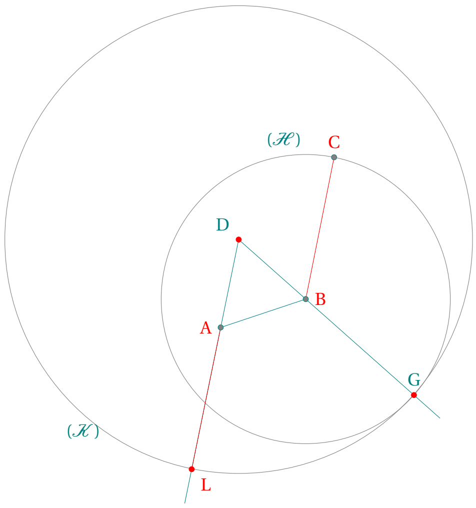
```
 \node (H) [label=135:$H$,draw,circle through=(C)] at (B) {};
\path let \p1 = ($ (B) - (C) $) in
  coordinate [label=left:$G$] (G) at ($ (B) ! veclen(\x1,\y1) ! (F) $);
\fill[red,opacity=.5] (G) circle (2pt);
```

Comme pour l'intersection de deux cercles, il est facile de trouver l'intersection d'une droite et d'un cercle avec `tkz-euclide`. Nous n'avons pas besoin de F

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

Il n'y a plus de difficultés. Voici le code final avec quelques simplifications. On trace le cercle $\mathcal{K}$ de centre D et passant par G. Il coupe la droite AD au point L. 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

Aujourd'hui, je ne suis plus professeur de mathématiques et je ne consacre que quelques heures à l'étude de la géométrie. J'ai voulu éviter de multiples complications en essayant de rendre `tkz-euclide` indépendant de `tkz-base`. C'est ainsi qu'est né `tkz-euclide` 4. Ce dernier est une version simplifiée de son prédécesseur. Les macros de `tkz-euclide 3` ont été conservées. L'unité est désormais `cm`. Si vous avez besoin de certaines macros de `tkz-base`, vous devrez peut-être utiliser `\tkzInit`.

3.4. tkz-euclide 5 vs tkz-euclide 4

Rien ne change pour l'utilisateur. La compilation doit être effectuée à l'aide du moteur LuaLaTeX, et les résultats sont plus précis et obtenus plus rapidement. Il suffit de charger `tkznameofpack` comme ceci `\usepackage[lua]{tkz-euclide}`.

3.5. Comment utiliser le paquet tkz-euclide ?

3.5.1. Regardons un exemple classique

Afin de montrer la bonne méthode, nous allons voir comment construire un triangle équilatéral. Plusieurs possibilités s'offrent à nous, nous allons suivre les étapes d'Euclide.

— Tout d'abord, vous devez utiliser une classe de document. Le meilleur choix pour tester votre code est de créer une seule figure avec la classe `standalone`.
`\documentclass{standalone}`

— Chargez ensuite le paquet `tkz-euclide` :
`\usepackage{tkz-euclide} ou \usepackage[lua]{tkz-euclide}`
Il n'est pas nécessaire de charger Ti*k*Z car le paquet `tkz-euclide` fonctionne au-dessus de TikZ et le charge.

— Démarrer le document et ouvrir un environnement d'images TikZ :
`\begin{document}`
`\begin{tikzpicture}`

— Nous définissons maintenant deux points fixes :

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

— Deux points définissent deux cercles, utilisons ces cercles :
cercle de centre A à B et cercle de centre B à A. Ces deux cercles ont deux points communs.
`\tkzInterCC(A,B)(B,A)`
Nous pouvons obtenir les points d'intersection avec
`\tkzGetPoints{C}{D}`

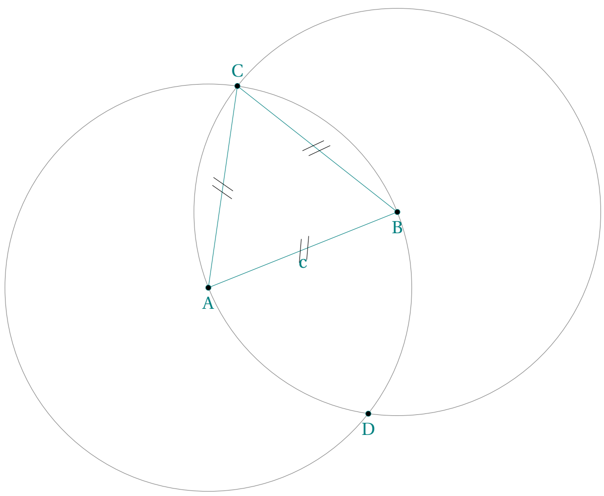
— Tous les points nécessaires étant obtenus, nous pouvons passer aux étapes finales, y compris les tracés.
`\tkzDrawCircles[gray,dashed](A,B B,A)`
`\tkzDrawPolygon(A,B,C)% The triangle`

— Dessiner tous les points A, B, C et D :
`\tkzDrawPoints(A,...,D)`

— Dans la dernière étape, nous imprimons des étiquettes sur les points et utilisons les options de positionnement suivantes
`\tkzLabelSegments[swap](A,B){$c$}`
`\tkzLabelPoints(A,B,D)`
`\tkzLabelPoints[above](C)`

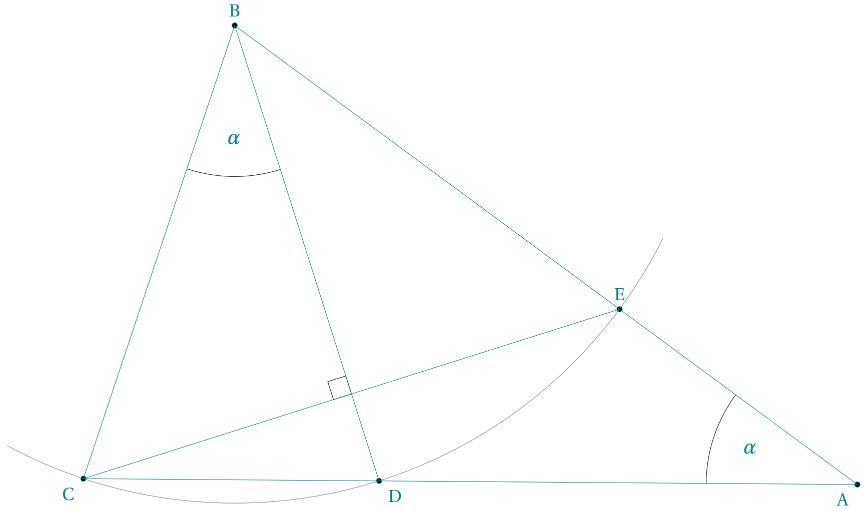
— Nous fermons finalement les deux environnements
`\end{tikzpicture}`
`\end{document}`

— Le code complet




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

3.5.2. Partie I : triangle d'or



Analysons la figure

1. CBD et DBE sont des triangles isocèles;
2. BC = BE et (BD) iest la bissectrice de l'angle CBE;
3. From this we deduce that the CBD and DBE angles are equal and have the same measure α :

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

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

   puis

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

   ou

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

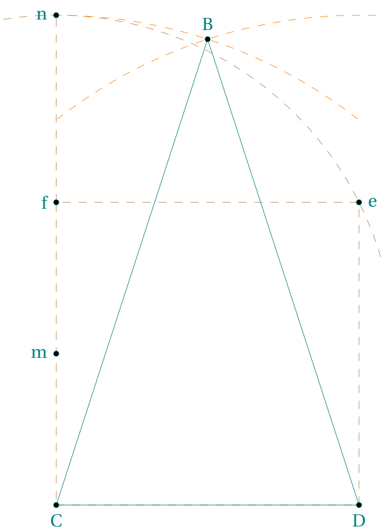
4. Finally

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

   le triangle CBD est un triangle "d'or".

Comment construire un triangle d'or ou un angle de 36° ?

1. Nous plaçons les points fixes C et D. Nous plaçons les points fixes C, D et D de la manière suivante : `\tkzDefPoint(0,0){C}` et `\tkzDefPoint(4,0){D}`;
2. On construit un carré CDef et on construit le milieu m de [Cf] ;
   Nous pouvons faire tout cela à l'aide d'un compas et d'une règle;
3. On trace alors un arc de centre m passant par e. Cet arc croise la droite (Cf) en n;
4. Les deux arcs de centre C et D et de rayon Cn définissent le point 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}
```

Après avoir construit le triangle d'or BCD, on construit le point A en remarquant que BD = DA. On obtient ensuite le point E et enfin le point F. Cela se fait déjà avec des intersections d'objets définis (ligne et cercle).

3.5.3. Partie II : deux autres méthodes avec le triangle d'or et le triangle d'euclide

tkz-euclide sait comment définir un triangle "d'or" ou "euclide". On peut définir BCD et BCA comme des triangles d'or.

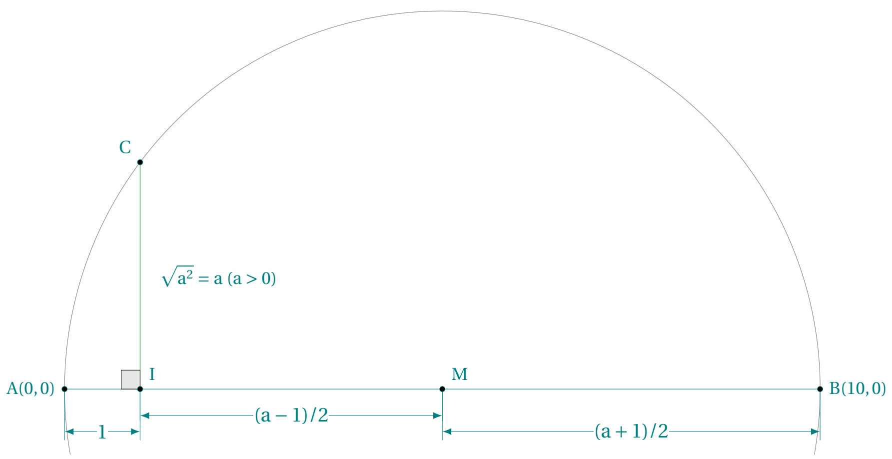
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(4,0){D}
  \tkzDefTriangle[golden](C,D)
  \tkzGetPoint{B}
  \tkzDefTriangle[golden](B,C)
  \tkzGetPoint{A}
  \tkzInterLC(B,A)(B,D) \tkzGetSecondPoint{E}
  \tkzInterLL(B,D)(C,E) \tkzGetPoint{F}
  \tkzDrawPoints(C,D,B)
  \tkzDrawPolygon(B,...,D)
  \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}
```

Voici une dernière méthode qui utilise les rotations :

```
\begin{tikzpicture}
\tkzDefPoint(0,0){C} % possible
% \tkzDefPoint[label=below:$C$](0,0){C}
% but don't do this
\tkzDefPoint(2,6){B}
% We get D and E with a rotation
\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
\tkzInterLL(B,E)(C,D) \tkzGetPoint{A}
\tkzInterLL(C,E)(B,D) \tkzGetPoint{H}
% drawing
\tkzDrawArc[delta=10](B,C)(E)
\tkzDrawPolygon(C,B,D)
\tkzDrawSegments(D,A B,A C,E)
% angles
\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
\tkzLabelPoints[below left](C,A)
\tkzLabelPoints[below right](D)
\tkzLabelPoints[above](B,E)
\end{tikzpicture}
```

3.5.4. Exemple complet mais minimal

Une unité de longueur étant choisie, l'exemple montre comment obtenir un segment de longueur $\sqrt{a}$ à partir d'un segment de longueur a, à l'aide d'une règle et d'un compas.
IB = a, AI = 1



```
\begin{tikzpicture}[scale=1,ra/.style={fill=gray!20}]
   % fixed points
   \tkzDefPoint(0,0){A}
   \tkzDefPoint(1,0){I}
   % calculation
   \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}
   \tkzDrawSegment[style=green!50!black](I,C)
   \tkzDrawArc(M,B)(A)
   \tkzDrawSegment[dim={$1$,-16pt,}](A,I)
   \tkzDrawSegment[dim={$(a-1)/2$,-10pt,}](I,M)
   \tkzDrawSegment[dim={$(a+1)/2$,-16pt,}](M,B)
   \tkzMarkRightAngle[ra](A,I,C)
   \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

— Le préambule
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. Examinons d'abord le préambule. Si vous en avez besoin, vous devez charger **xcolor** avant **tkz-euclide**, c'est-à-dire avant Ti*k*Z. Ti*k*Z peut causer des problèmes avec les caractères actifs, mais... fournit dans sa dernière version une bibliothèque censée résoudre ces problèmes : babel.

```
\documentclass{standalone} % or another class
   % \usepackage{xcolor} % before tikz or tkz-euclide if necessary
\usepackage{tkz-euclide} % no need to load TikZ
   % \usetkzobj{all}  is no longer necessary
   % \usetikzlibrary{babel}  if there are problems with the active characters
```

Le code suivant se compose de plusieurs parties :

— Définition des points fixes : la première partie comprend les définitions des points nécessaires à la construction, ce sont les points fixes. Les macros **\tkzInit** et **\tkzClip** ne sont pas nécessaires dans la plupart des cas.

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

— La deuxième partie est dédiée à la création de nouveaux points à partir des points fixes ; un point B est placé à 10 cm de A. Le milieu de [AB] est défini par M, puis la ligne orthogonale à la ligne (AB) est recherchée au point I. Ensuite, nous cherchons l'intersection de cette ligne avec le demi-cercle de centre M passant par A.

```
\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}
```

— La troisième comprend les différents dessins;

```
\tkzDrawSegment[style=green!50!black](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)
```

— Marquage : la quatrième partie est consacrée au marquage;

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

— Étiquetage : ce dernier point ne concerne que l'emplacement des étiquettes.

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

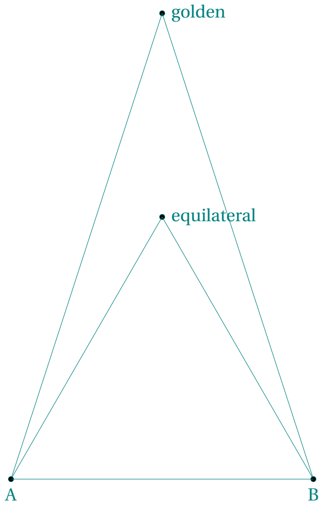
4. Les éléments du code de tkz

Pour travailler avec mon package, vous devez avoir des notions de LaTeX ainsi que de Ti*k*Z.
Dans ce paragraphe, nous commençons à examiner les "règles" et les "symboles" utilisés pour créer une figure avec **tkz-euclide**.

4.1. Objets et langage

Les objets primitifs sont des points. Vous pouvez vous référer à un point à tout moment en utilisant le nom donné lors de sa définition (il est possible d'attribuer un nom différent plus tard).
Pour obtenir de nouveaux points, vous utiliserez des macros. Les macros **tkz-euclide** ont un nom qui commence par `tkz`. Il existe quatre catégories principales commençant par : `\tkzDef...`, `\tkzDraw...`, `\tkzMark...` et `\tkzLabel...`. Les points utilisés sont passés en paramètres entre parenthèses tandis que les points créés sont entre accolades.
Le code des figures est placé dans un environnement `tikzpicture`.
Contrairement à Ti*k*Z, vous ne devez pas terminer une macro par ";". Nous perdons ainsi la notion importante qui est le `chemin` (*path*). Cependant, il est possible de placer du code entre les macros **tkz-euclide**.
Parmi la première catégorie, `\tkzDefPoint` vous permet de définir des points fixes. Cela sera étudié en détail plus tard. Ici, nous verrons en détail la macro `\tkzDefTriangle`.
Cette macro permet d'associer à une paire de points un troisième point afin de définir un certain triangle `\tkzDefTriangle(A,B)`. Le point obtenu est référencé par `tkzPointResult` et il est possible de choisir une autre référence avec
`\tkzGetPoint{C}` par exemple.
`\tkzDefTriangle[euclid](A,B) \tkzGetPoint{C}`
Les parenthèses sont utilisées pour passer des arguments. Dans `(A,B)`, A et B sont les points avec lesquels un troisième sera défini. Cependant, dans `{C}`, nous utilisons des accolades pour récupérer le nouveau point.

Pour choisir un certain type de triangle parmi les choix suivants : `equilateral`, `isosceles right`, `half`, `pythagoras`, `school`, `golden or sublime`, `euclid`, `gold`, `cheops`... et `two angles`, il suffit de choisir entre crochets, par exemple :




```
\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 et conventions

J'ai délibérément choisi d'utiliser les conventions géométriques françaises et personnelles pour décrire les objets géométriques représentés. Les objets définis et représentés par **tkz-euclide** sont des points, des droites et des cercles situés dans un plan. Ce sont les objets principaux de la géométrie euclidienne à partir desquels nous construirons des figures.
Selon `Euclide`, ces figures ne feront qu'illustrer des idées pures produites par notre cerveau. Ainsi, un point n'a pas de dimension et donc aucune existence réelle. De même, la ligne n'a pas de largeur et donc aucune existence dans le monde réel. Les objets que nous allons considérer ne sont que des représentations d'objets mathématiques idéaux. **tkz-euclide** suivra les étapes des anciens Grecs pour obtenir des constructions géométriques à l'aide de la règle et du compas.
Voici les notations qui seront utilisées :

— Les points sont représentés géométriquement soit par un petit disque, soit par l'intersection de deux lignes (deux droites, une droite et un cercle ou deux cercles). Dans ce cas, le point est représenté par une croix.


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

ou bien


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

L'existence d'un point étant établie, on peut lui donner une étiquette qui sera une lettre majuscule (sauf exceptions) de l'alphabet latin telle que A, B ou C. Par exemple :
— O est le centre d'un cercle, d'une rotation, etc ;
— M a défini un point médian ;
— H a défini le pied d'une hauteur ;

— P' est l'image de P par une transformation ;

Il est important de noter que le nom de référence d'un point dans le code peut être différent de l'étiquette qui le désigne dans le texte. Ainsi, nous pouvons définir un point A et lui donner l'étiquette P. En particulier, le style sera différent, le point A sera étiqueté A.


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

Exceptions : certains points comme le milieu des côtés d'un triangle partagent une caractéristique, il est donc normal que leurs noms partagent également un caractère commun. Nous désignerons ces points par M_a, M_b et M_c ou M_A, M_B et M_C.

Dans le code, ces points seront désignés par les termes suivants : M_A, M_B et M_C : M_A, M_B et M_C.

Une autre exception concerne les points de construction intermédiaires qui ne seront pas étiquetés. Ils sont souvent désignés par une lettre minuscule dans le code.

— Les segments de droite sont désignés par deux points représentant leurs extrémités entre crochets : [AB].

— En géométrie euclidienne, les droites sont définies par deux points. Ainsi, A et B définissent la droite (AB). On peut aussi désigner cette droite en utilisant l'alphabet grec et la nommer (δ) ou (Δ). Il est également possible de désigner la droite par des lettres minuscules telles que d et d'.

— La demi-droite est désignée comme suit [AB).

— Relation entre les droites. Deux droites perpendiculaires (AB) et (CD) seront écrites $(AB) \perp (CD)$ et si elles sont parallèles nous écrirons $(AB) \mathbin{/\!/} (CD)$.

— Les longueurs des côtés du triangle ABC sont AB, AC et BC. Les nombres sont également désignés par une lettre minuscule, nous écrirons donc : $AB = c$, $AC = b$ et $BC = a$. La lettre a est également utilisée pour représenter un angle, et r est fréquemment utilisé pour représenter un rayon, d un diamètre, l une longueur, d une distance.

— Les polygones sont ensuite désignés par leurs sommets. Ainsi, ABC est un triangle, EFGH un quadrilatère.

— Les angles sont généralement mesurés en degrés (ex 60°) et dans un triangle équilatéral ABC on écrira $\widehat{ABC} = \widehat{B} = 60^\circ$.

— Les arcs sont désignés par leurs extrémités. Par exemple, si A et B sont deux points du même cercle, alors widearcAB.

— Les cercles sont notés $\mathscr{C}$ s'il n'y a pas de confusion possible ou $\mathscr{C}(O\,;\,A)$ pour un cercle de centre O et passant par le point A ou $\mathscr{C}(O\,;\,1)$ pour un cercle de centre O et de rayon 1 cm.

— Nom des lignes particulières d'un triangle : J'ai utilisé les termes bissectrice, bissectrice estérieure, médiatrice , hauteur, médiane et symédiane.

— (x_1,y_1) coordonnées du point A_1, (x_A,y_A) coordonnées du point A.

4.3. Définir, calculer, dessiner, marquer, étiqueter

Le titre aurait pu être : `Séparation du calcul et des dessins`

Lorsqu'un document est préparé à l'aide du système LaTeX le code source du document peut être divisé en deux parties : le corps du document et le préambule. Grâce à cette méthodologie, les publications peuvent être structurées, stylisées et composées avec un minimum d'effort. Je propose une méthodologie similaire pour la création de figures avec **tkz-euclide**.

La première partie définit les points fixes, la seconde permet de créer de nouveaux points. `Set et Calculate` sont les deux parties principales. Il ne reste plus qu'à dessiner (ou remplir), marquer et étiqueter. Il est possible que **tkz-euclide** soit insuffisant pour certaines de ces dernières actions, mais vous pouvez utiliser Ti*k*Z

Une dernière remarque qui me semble importante, il est préférable d'éviter autant que possible d'introduire des coordonnées dans un code. Je pense que les coordonnées devraient apparaître au début du code avec les points fixes. L'utilisation de références est alors recommandée. La plupart des macros ont l'option `nœuds` ou `avec nœuds`.

Je pense également qu'il est préférable de définir les styles des différents objets dès le début.

5. À propos de cette documentation et des exemples

Il est obtenu en compilant avec "lualatex". J'utilise une classe `doc.cls` basée sur `scrartcl`.
Ci-dessous la liste des styles utilisés dans la documentation. Pour comprendre comment utiliser les styles, voir la section 35

```
\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}}
```

Certains exemples utilisent des styles prédéfinis tels que

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

Deuxième partie

Réglage

6. Première étape : points fixes

La première étape dans une construction géométrique consiste à définir les points fixes à partir desquels la figure sera construite.
L'idée générale est d'éviter de manipuler les coordonnées et de préférer utiliser les références des points fixes dans la première étape ou obtenues à l'aide des outils fournis par le package. Même si c'est possible, je pense que c'est une mauvaise idée de travailler directement avec les coordonnées. Il est préférable d'utiliser des points nommés.
tkz-euclide utilise des macros et un vocabulaire spécifiques à la construction géométrique. Il est bien sûr possible d'utiliser les outils de Ti*k*Z, mais il me semble plus logique de ne pas mélanger les différentes syntaxes.
Un point dans **tkz-euclide** est un "nœud" particulier pour Ti*k*Z. Dans la section suivante, nous verrons comment définir des points en utilisant des coordonnées. Le style des points (couleur et forme) ne sera pas discuté. Vous trouverez quelques indications dans certains exemples ; pour plus d'informations, vous pouvez lire la section suivante 35.

7. Définition d'un point : \tkzDefPoint ou \tkzDefPoints

Les points peuvent être spécifiés de l'une des manières suivantes :
- Coordonnées cartésiennes ;
- Coordonnées polaires ;
- Points nommés ;
- Points relatifs.

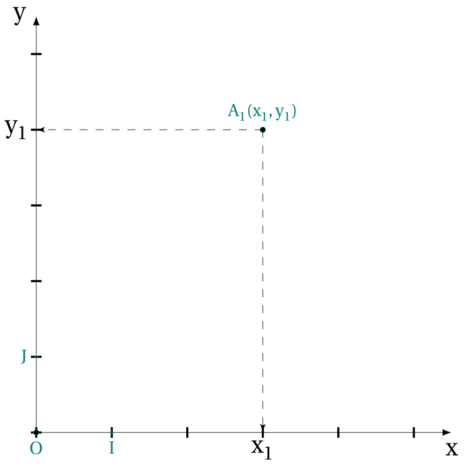
Un point est défini s'il possède un nom lié à une paire unique de nombres décimaux. Soit (x, y) ou $(a: d)$, c'est-à-dire (x abscisse, y ordonnée) ou (a angle : d distance). Cela est possible car le plan a été pourvu d'un système de coordonnées cartésiennes orthonormé. Les axes de travail sont (ortho)normés avec une unité égale à 1 cm.
Les coordonnées cartésiennes (a, b) se réfèrent au point a centimètres dans la direction de l'axe des x et b centimètres dans la direction de l'axe des y.
Un point en coordonnées polaires nécessite un angle α, en degrés, et une distance d depuis l'origine avec une unité dimensionnelle par défaut, qui est le `cm`.
La macro `\tkzDefPoint` est utilisée pour définir un point en lui attribuant des coordonnées. Cette macro est basée sur `\coordinate`, une macro de Ti*k*Z. Elle peut utiliser des options spécifiques à Ti*k*Z telles que **shift**. Si des calculs sont nécessaires, le package `xfp` est choisi. Nous pouvons utiliser des coordonnées cartésiennes ou polaires.

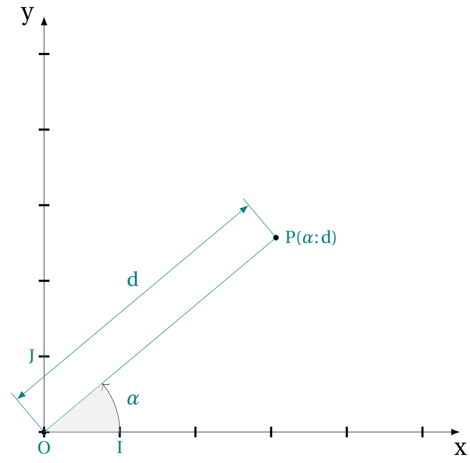
Cartesian coordinates

```
\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

```
\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. Définition d'un point nommé \tkzDefPoint

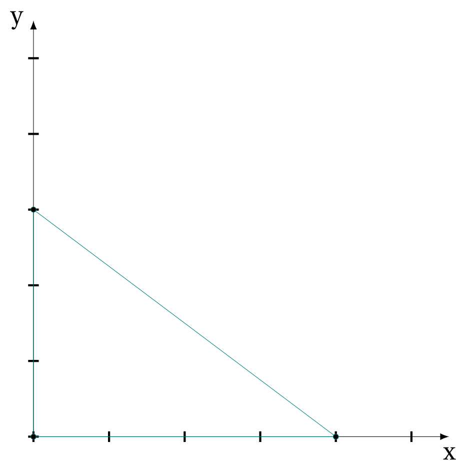
\tkzDefPoint[⟨options locales⟩](⟨x,y⟩){⟨ref⟩} or (⟨α:d⟩){⟨ref⟩}

arguments	défaut	définition
(x,y)	pas de défaut	x et y sont deux dimensions, par défaut en cm.
(α:d)	pas de défaut	α is an angle in degrees, d est une dimension
{ref}	pas de défaut	Référence attribuée au point : A, T_a ,P1 or P_1

Les arguments obligatoires de cette macro sont deux dimensions exprimées en décimales, dans le premier cas il s'agit de deux mesures de longueur, dans le second cas il s'agit d'une mesure de longueur et de la mesure d'un angle en degrés. Ne confondez pas la référence avec le nom d'un point. La référence est utilisée par les calculs, mais souvent, le nom est identique à la référence.

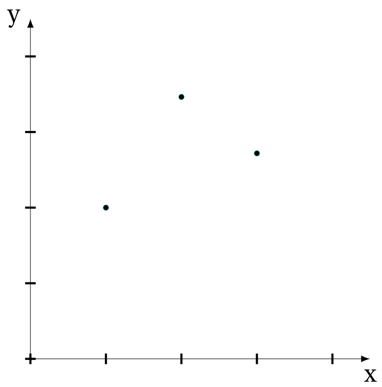
options	default	definition
label	pas de défaut	allows you to place a label at a predefined distance
shift	pas de défaut	adds (x,y) or (α:d) to all coordinates

7.1.1. Coordonnées cartésiennes



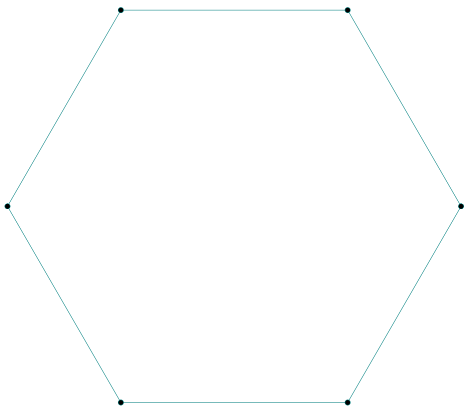

```
\begin{tikzpicture}
\tkzInit[xmax=5,ymax=5] % limits the size of the axes
\tkzDrawX[>=latex]
\tkzDrawY[>=latex]
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
\tkzDefPoint(0,3){C}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints(A,B,C)
\end{tikzpicture}
```

7.1.2. Calculs avec xfp




```
\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}
```

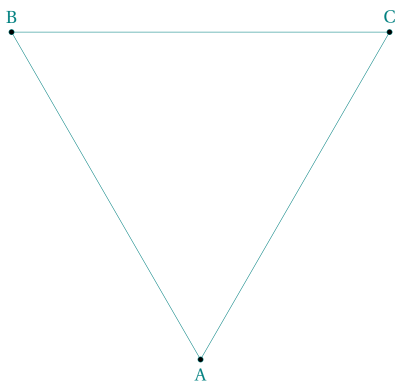
7.1.3. Coordonnées polaires



```
\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. Points relatifs

Tout d'abord, nous pouvons utiliser l'environnement `scope` de Ti*k*Z. Dans l'exemple suivant, nous avons un moyen de définir un triangle équilatéral.




```
\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}
```

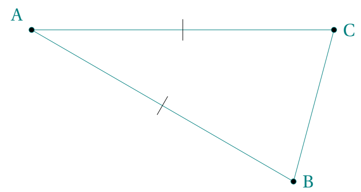
7.2. Point relatif à un autre : \tkzDefShiftPoint

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

arguments	défaut	définition
(x,y)	pas de défaut	x et y sont deux dimensions, par défaut en cm.
(α:d)	pas de défaut	α est un angle en degrés, d est une dimension
{ref}	pas de défaut	Reference assigned to the point: A, T_a ,P1 or P_1
options	**default**	**definition**
[pt]	pas de défaut	\tkzDefShiftPoint[A](0:4){B}

7.2.1. Triangle isocèle

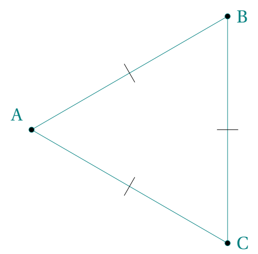
Cette macro permet de placer un point par rapport à un autre. Cela équivaut à une translation. Voici comment construire un triangle isocèle dont le sommet principal est A et l'angle au sommet 30°.




```
\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. Triangle équilatéral

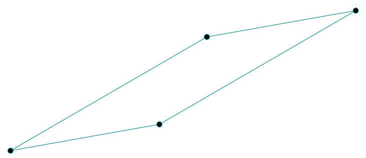
Voyons comment obtenir un triangle équilatéral (il y a beaucoup plus simple)




```
\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. Parallélogramme

Il existe un moyen plus simple



```
\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. Définition des points multiples : \tkzDefPoints

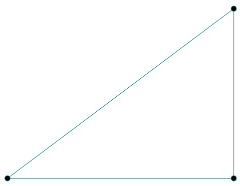
`\tkzDefPoints[⟨options locales⟩]{⟨x1/y1/n1,x2/y2/r2, ...⟩}`

x_i et y_i sont les coordonnées d'un point de référence r_i.

arguments	défaut	exemple
$x_i/y_i/r_i$		`\tkzDefPoints{0/0/O,2/2/A}`

options	défaut	définition
shift	pas de défaut	Adds (x, y) or $(\alpha{:}d)$ to all coordinates

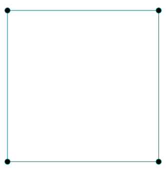
7.4. Créer un triangle



```
\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. Créer un carré

Notez ici la syntaxe pour dessiner le polygone.



```
\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}
```

Troisième partie

Calculs

Maintenant que les points fixes sont définis, nous pouvons, avec leurs références en utilisant des macros du package ou des macros que vous créerez, obtenir de nouveaux points. Les calculs peuvent ne pas être évidents, mais ils sont généralement effectués par le package. Vous pouvez avoir besoin d'utiliser certaines constantes mathématiques, voici la liste des constantes définies par le package.

8. Outils auxiliaires

8.1. Constantes

tkz-euclide connaît certaines constantes, voici la liste :

```
\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. Nouveau point par calcul

Lorsqu'une macro de **tkznameofpack** crée un nouveau point, celui-ci est stocké en interne avec la référence **tkzPointResult**. Vous pouvez lui attribuer votre propre référence. Cela se fait avec la macro **\tkzGetPoint**. Une nouvelle référence est créée, votre choix de référence doit être placé entre accolades.

\tkzGetPoint{⟨ref⟩}

Si le résultat se trouve dans **tkzPointResult**, vous pouvez y accéder avec **\tkzGetPoint**.

arguments	défaut	exemple	
ref	pas de défaut	\tkzGetPoint{M}	voir l'exemple suivant

Il faut parfois obtenir deux points. C'est possible avec

\tkzGetPoints{⟨ref1⟩}{⟨ref2⟩}

Le résultat se trouve dans **tkzPointFirstResult** et **tkzPointSecondResult**.

arguments	défaut	exemple	
{ref1,ref2}	pas de défaut	\tkzGetPoints{M,N}	C'est le cas de \tkzInterCC

Si vous n'avez besoin que du premier ou du deuxième point, vous pouvez également utiliser :

\tkzGetFirstPoint{⟨ref1⟩}

arguments	défaut	exemple
ref1	pas de défaut	\tkzGetFirstPoint{M}

\tkzGetSecondPoint{⟨ref2⟩}

arguments	défaut	exemple
ref2	pas de défaut	\tkzGetSecondPoint{M}

Parfois, les résultats consistent en un point et une dimension. Vous obtenez le point avec `\tkzGetPoint` et la dimension avec `\tkzGetLength`.

\tkzGetLength{⟨name of a macro⟩}

arguments	défaut	exemple
name of a macro	pas de défaut	\tkzGetLength{rAB} \rAB gives the length in cm

9. Points particuliers

Voici quelques points particuliers.

9.1. Milieu d'un segment \tkzDefMidPoint

Il s'agit de déterminer le milieu d'un segment.

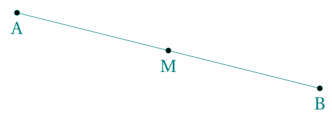
\tkzDefMidPoint(⟨pt1,pt2⟩)

Le résultat se trouve dans `tkzPointResult`. Nous pouvons y accéder avec `\tkzGetPoint`.

arguments, défaut et définition		
(pt1,pt2)	pas de défaut	pt1 et pt2 sont deux points

9.1.1. Utilisation de \tkzDefMidPoint

Réexaminer l'utilisation des `\tkzDefPoint`.




```
\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

D'après Wikipédia : En mathématiques, deux quantités sont au nombre d'or si leur rapport est le même que le rapport de leur somme à la plus grande des deux quantités. Exprimé algébriquement, pour des quantités a, b telles que $a > b > 0$; $a + b$ est à a ce que a est à b.

$\frac{a+b}{a} = \frac{a}{b} = \phi = \frac{1+\sqrt{5}}{2}$

L'une des deux solutions de l'équation $x^2 - x - 1 = 0$ est le nombre d'or. est le nombre d'or ϕ, $\phi = \frac{1+\sqrt{5}}{2}$.

`\tkzDefGoldenRatio(⟨pt1,pt2⟩)`

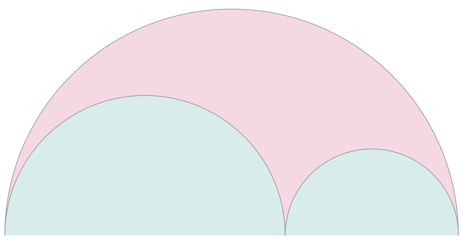
arguments	défaut	exemple
`(pt1,pt2)`	`pas de défaut`	`\tkzDefGoldenRatio(A,C) \tkzGetPoint{B}`

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

9.2.1. Utiliser le nombre d'or pour diviser un segment de droite


```
\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



```
\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}
```

Il est également possible d'utiliser la macro suivante.

9.3. Coordonnées barycentriques avec \tkzDefBarycentricPoint

pt_1, pt_2, …, pt_n étant n points, ils définissent n vecteurs $\vec{v_1}$, $\vec{v_2}$, …, $\vec{v_n}$ avec l'origine du repère comme point commun. α_1, α_2, …α_n sont n nombres, le vecteur obtenu par :

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

définit un unique point.

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

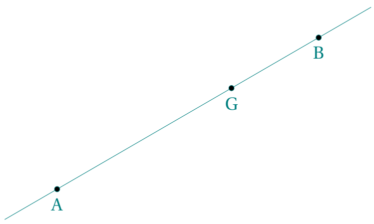
arguments, défaut et définition		
`(pt1=`α_1`,pt2=`α_2`,...)`	`pas de défaut`	`Each point has a assigned weight`

Vous avez besoin d'au moins deux points. Result in `tkzPointResult`.

9.3.1. avec deux points

Dans l'exemple suivant, on obtient le barycentre des points A et B avec des coefficients 1 et 2, autrement dit :

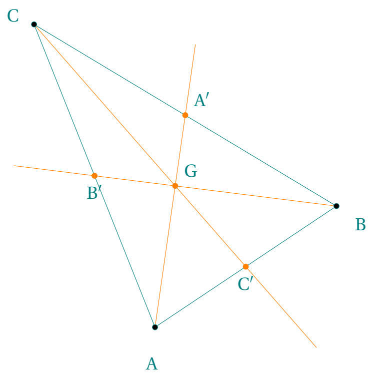
$$\overrightarrow{AI} = \frac{2}{3}\overrightarrow{AB}$$




```
\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. avec trois points

Cette fois, M est simplement le centre de gravité du triangle.
Pour des raisons de simplification et d'homogénéité, il existe également `\tkzCentroid`.




```
\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. Centre de similitude interne et externe

Les centres des deux homothéties dans lesquelles correspondent deux cercles sont appelés centres de similitude externe et interne. Vous pouvez utiliser `\tkzDefIntSimilitudeCenter` et `\tkzDefExtSimilitudeCenter` mais la macro suivante est meilleure.

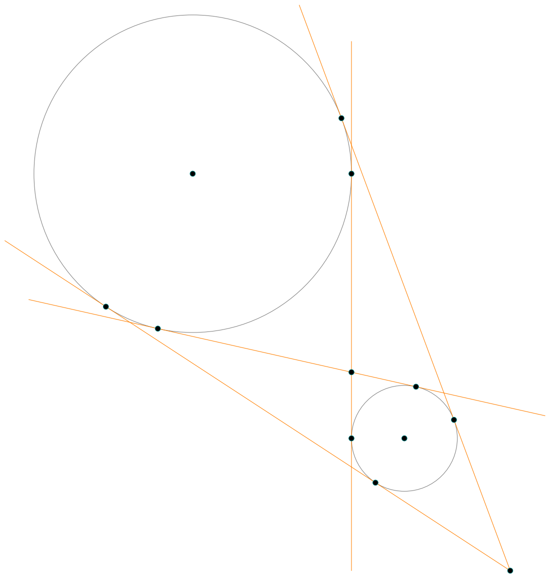
`\tkzDefSimilitudeCenter[⟨options⟩](⟨O,A⟩)(⟨O',B⟩)`

arguments, exemples et explications

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

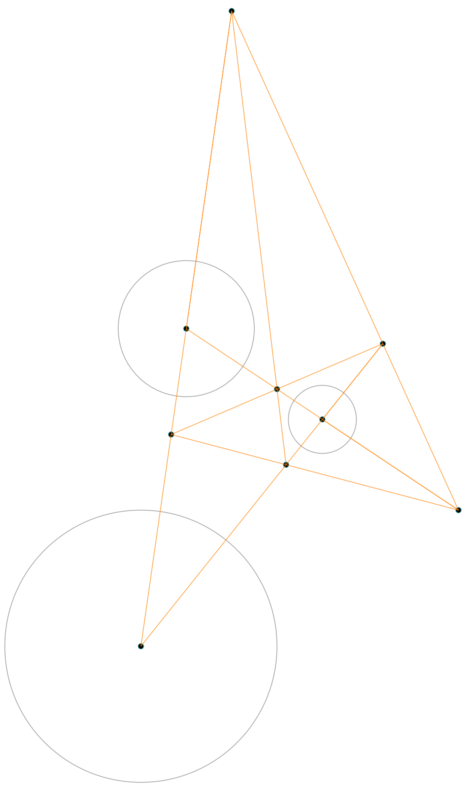
options	défaut	définition
ext	ext	external center
int	ext	internal center

9.4.1. Interne et externe avec node



```
\begin{tikzpicture}[scale=.7]
 \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

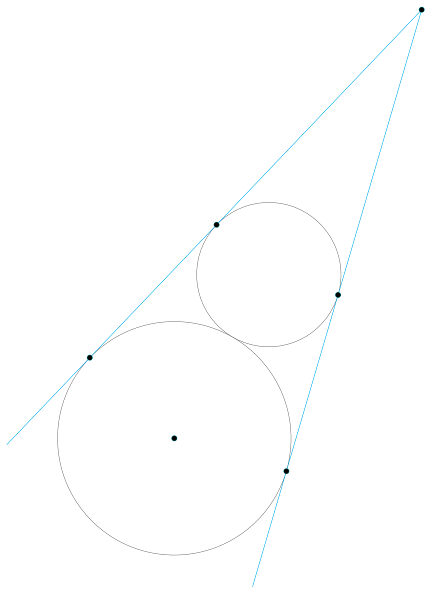


```
\begin{tikzpicture}[scale=.6,rotate=90]
\tkzDefPoints{0/0/A,3/0/a,7/-1/B,5.5/-1/b}
\tkzDefPoints{5/-4/C,4.25/-4/c}
\tkzDrawCircles(A,a B,b C,c)
\tkzDefExtSimilitudeCenter(A,a)(B,b) \tkzGetPoint{I}
\tkzDefExtSimilitudeCenter(A,a)(C,c) \tkzGetPoint{J}
\tkzDefExtSimilitudeCenter(C,c)(B,b) \tkzGetPoint{K}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

Vous pouvez utiliser `\tkzDefBarycentricPoint` pour trouver un centre homothétique

```
\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. Division harmonique avec \tkzDefHarmonic

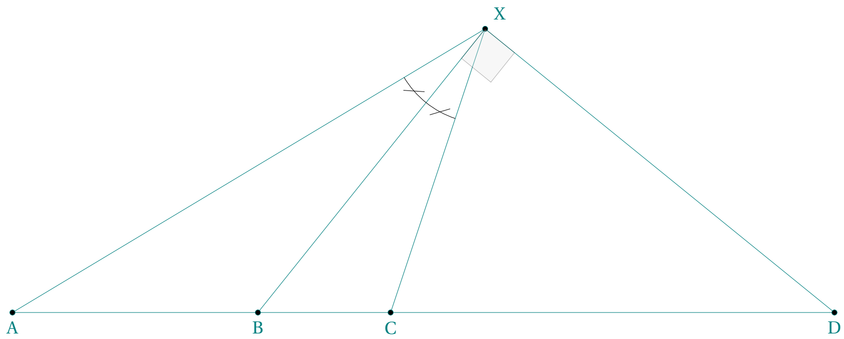
\tkzDefHarmonic[⟨options⟩](⟨pt1,pt2,pt3⟩) or (⟨pt1,pt2,k⟩)

options	défaut	définition
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 $(A,B;C,D)=-1$

9.5.1. options ext et 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. Bissectrice et division harmonique




```
\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` est l'option par défaut


```
\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. Points équidistants avec \tkzDefEquiPoints

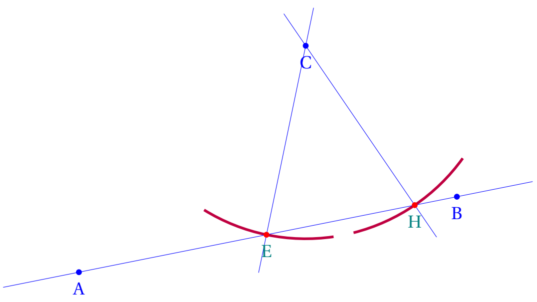
\tkzDefEquiPoints[⟨options locales⟩](⟨pt1,pt2⟩)

arguments	défault	définition
(pt1,pt2)	pas de défaut	liste non ordonnée de deux éléments

options	défaut	définition
dist	2 (cm)	la moitié de la distance entre les deux points
from=pt	pas de défaut	point de référence
show	false	si vrai, affiche les traces du compas
/compass/delta	0	taille du tracé du compas

Cette macro permet d'obtenir deux points sur une droite équidistante d'un point donné.

9.6.1. Using \tkzDefEquiPoints with options

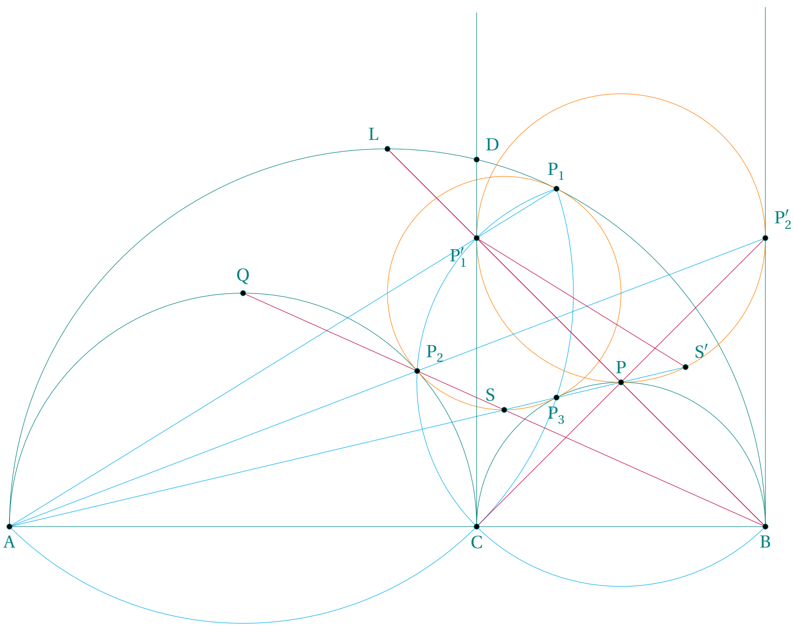



```
\begin{tikzpicture}
  \tkzSetUpCompass[color=purple,line width=1pt]
  \tkzDefPoints{0/1/A,5/2/B,3/4/C}
  \tkzDefEquiPoints[from=C,dist=1,show,
      /tkzcompass/delta=20](A,B)
   \tkzGetPoints{E}{H}
   \tkzDrawLines[color=blue](C,E C,H A,B)
   \tkzDrawPoints[color=blue](A,B,C)
   \tkzDrawPoints[color=red](E,H)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

9.7. Milieu d'un arc

\tkzDefMidArc(⟨pt1,pt2,pt3⟩)

arguments	défault	définition
pt1,pt2,pt3	pas de défaut	pt1 est le centre, $\overset{\frown}{pt2pt3}$ l'arc


L
D
P1
P'2
Q
P'1
P2
S'
P
S
P3
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 sur une ligne ou un cercle

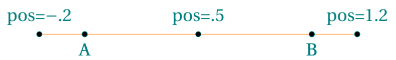
10.1. Point sur une ligne avec \tkzDefPointOnLine

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

arguments	défault	définition
pt1,pt2	pas de défaut	Deux points pour définir une ligne

options	default	definition
pos=nb		nb est une décimale

10.1.1. Utilisation de l'option pos




```
\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 sur un cercle avec \tkzDefPointOnCircle

L'ordre des arguments a changé : il s'agit maintenant du centre, de l'angle et du point ou du rayon. J'ai ajouté deux options pour travailler avec les radians qui sont `travers en rad` et `R en rad`.

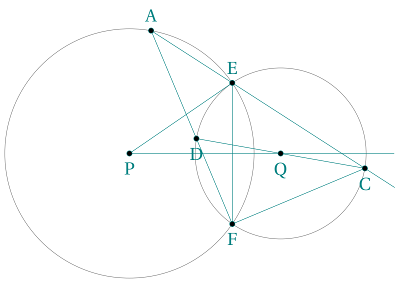
`\tkzDefPointOnCircle[⟨options locales⟩]`

options	défaut	exemples définition
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

Le nouvel ordre des arguments est le suivant : centre, angle et point ou rayon.

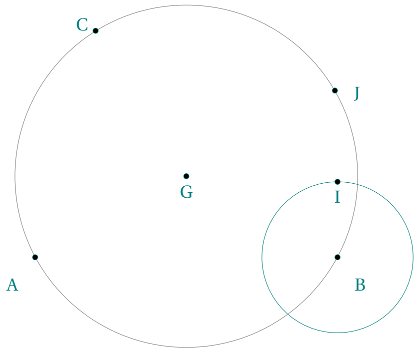
10.2.1. Altshiller's Theorem

Les deux lignes joignant les points d'intersection de deux cercles orthogonaux à un point de l'un des cercles rencontrent l'autre cercle en deux points diamétralement opposés. Altshiller 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. Utilisation de \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. Points spéciaux liés à un triangle

11.1. Centre du triangle : \tkzDefTriangleCenter

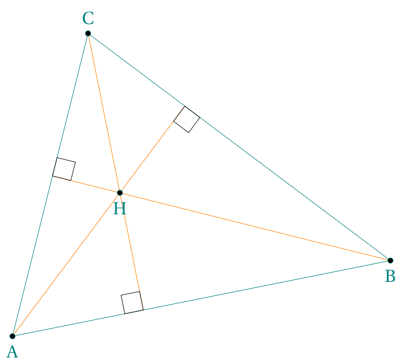
`\tkzDefTriangleCenter[⟨options locales⟩](⟨A,B,C⟩)`

Cette macro vous permet de définir le centre d'un triangle. Faites attention, les arguments sont des listes de trois points. Cette macro est utilisée en conjonction avec `\tkzGetPoint` pour obtenir le centre recherché. Vous pouvez utiliser `tkzPointResult` s'il n'est pas nécessaire de conserver les résultats.

arguments	défaut	exemple
(pt1,pt2,pt3)	pas de défaut	\tkzDefTriangleCenter[ortho](B,C,A)
options	**défaut**	**définition**
ortho	circum	intersection des hauteurs
orthic	circum	…
centroid	circum	intersection des médianes
median	circum	…
circum	circum	centre du cercle circonscrit
in	circum	centre du cercle inscrit dans un triangle
in	circum	intersection des bissectrices
ex	circum	centre d'un cercle exinscrit à un triangle
euler	circum	centre du cercle d'Euler
gergonne	circum	défini avec le triangle de contact
symmedian	circum	Point de Lemoine ou centre symédian ou point de Grebe
lemoine	circum	…
grebe	circum	…
spieker	circum	Centre de cercle Spieker
nagel	circum	Le centre Nagel
mittenpunkt	circum	Ou middlespoint
feuerbach	circum	Point de Feuerbach

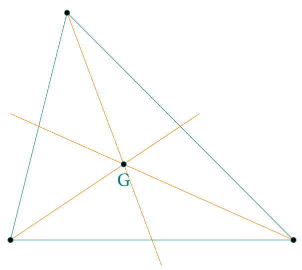
11.1.1. Option ortho or orthic

L'intersection H des trois altitudes d'un triangle est appelée orthocentre.



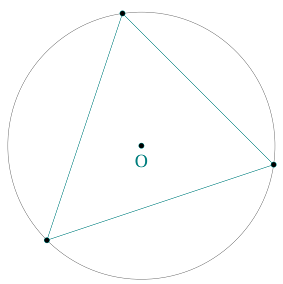

```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,1){B}
  \tkzDefPoint(1,4){C}
  \tkzDefTriangleCenter[ortho](B,C,A)
  \tkzGetPoint{H}
  \tkzDefSpcTriangle[orthic,name=H](A,B,C){a,b,c}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawSegments[new](A,Ha B,Hb C,Hc)
  \tkzDrawPoints(A,B,C,H)
  \tkzLabelPoint(H){$H$}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

11.1.2. Option centroid




```
\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

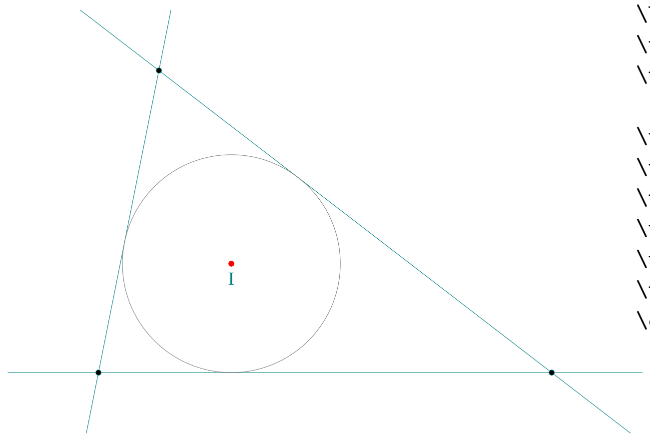



```
 \begin{tikzpicture}
  \tkzDefPoints{0/1/A,3/2/B,1/4/C}
  \tkzDefTriangleCenter[circum](A,B,C)
  \tkzGetPoint{O}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawCircle(O,A)
  \tkzDrawPoints(A,B,C,O)
  \tkzLabelPoint(O){$O$}
\end{tikzpicture}
```

11.1.4. Option in

En géométrie, le cercle inscrit ou cercle inscriptible d'un triangle est le plus grand cercle contenu dans le triangle ; il touche (est tangent à) les trois côtés. Le centre du cercle inscrit est un centre de triangle appelé le centre du triangle. Le centre du cercle inscrit, appelé le centre du cercle inscrit, peut être trouvé comme l'intersection des trois bissectrices internes des angles. Le centre d'un excercle est l'intersection de la bissectrice interne d'un angle (au sommet A, par exemple) et des bissectrices externes des deux autres. Le centre de cet excercle est appelé l'excentre par rapport au sommet A, ou l'excentre de A. Parce que la bissectrice interne d'un angle est perpendiculaire à sa bissectrice externe, il s'ensuit que le centre du cercle inscrit ainsi que les trois centres des excercles forment un système orthocentrique.
(Article sur Wikipedia)

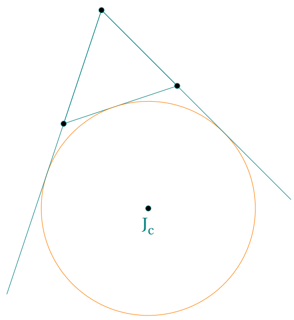
Nous obtenons le centre du cercle inscrit du triangle. Le résultat est bien sûr dans **tkzPointResult**. Nous pouvons le récupérer avec **\tkzGetPoint**.




```
\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

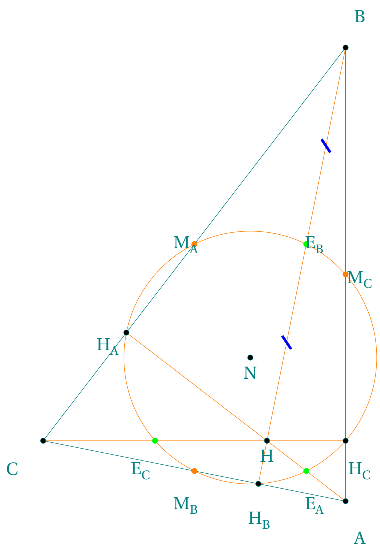
Un excercle ou cercle exinscrit du triangle est un cercle situé à l'extérieur du triangle, tangent à l'un de ses côtés et tangent aux extensions des deux autres. Chaque triangle a trois excercles distincts, chacun tangent à l'un des côtés du triangle.
(Article sur Wikipedia)
Nous obtenons le centre d'un cercle inscrit du triangle. Le résultat est bien sûr dans `tkzPointResult`. Nous pouvons le récupérer avec `\tkzGetPoint`.




```
\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

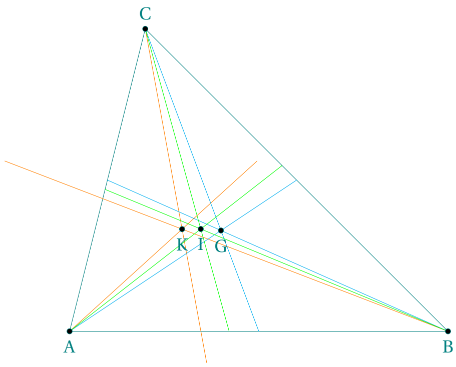
Cette macro permet d'obtenir le centre du cercle des neuf points ou cercle d'Euler ou cercle de Feuerbach. Le cercle des neuf points, également appelé cercle d'Euler ou cercle de Feuerbach, est le cercle qui passe par les pieds perpendiculaires H_A, H_B et H_C abaissés depuis les sommets de n'importe quel triangle de référence ABC sur les côtés opposés à eux. Euler a montré en 1765 qu'il passe également par les milieux M_A, M_B et M_C des côtés de ABC. Selon le théorème de Feuerbach, le cercle des neuf points passe également par les milieux E_A, E_B et E_C des segments joignant les sommets et l'orthocentre H. Ces points sont couramment appelés les points d'Euler.
(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

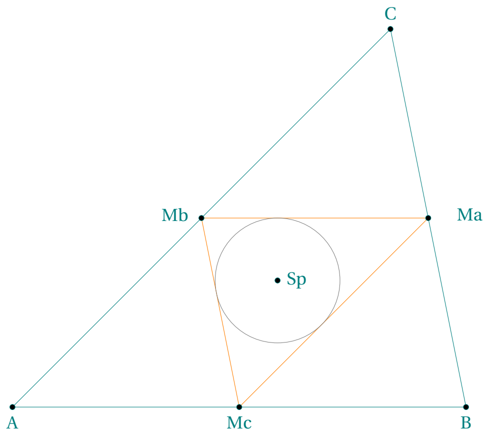
Le point de concurrence K des symmédiennes, parfois également appelé point de Lemoine (en Angleterre et en France) ou point de Grebe (en Allemagne).
Weisstein, Eric W. "Symmedian Point." From MathWorld–A Wolfram Web Resource.




```
\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

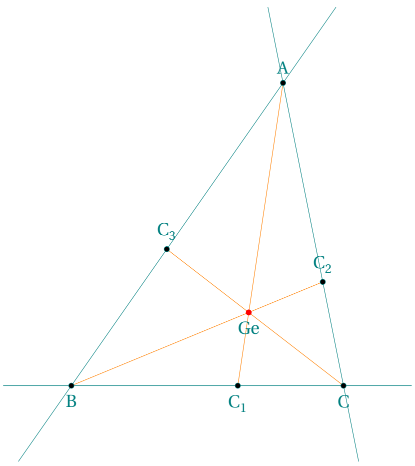
Le centre de Spieker est le centre Sp du cercle de Spieker, c'est-à-dire le centre du cercle inscrit du triangle médian d'un triangle de référence.
Source : Weisstein, Eric W. "Spieker Center." From MathWorld–A Wolfram Web Resource.




```
\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

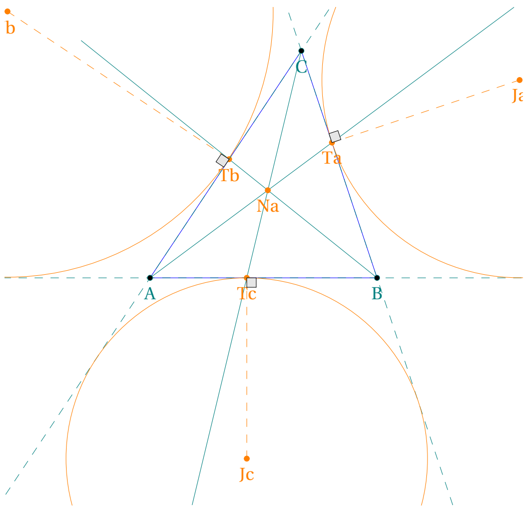
Le point de Gergonne est le point de concurrence qui résulte de la connexion des sommets d'un triangle aux points opposés de tangence du cercle inscrit du triangle. (Joseph Gergonne, mathématicien français)




```
\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

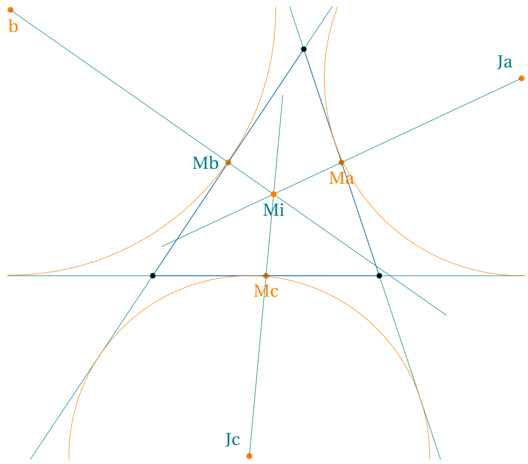
Soit Ta le point où le cercle exinscrit avec le centre Ja rencontre le côté BC d'un triangle ABC, et définissons Tb et Tc de manière similaire. Alors les droites ATa, BTb et CTc concourent au point de Nagel Na.
Source : Weisstein, Eric W. "Nagel point." From MathWorld–A Wolfram Web Resource.




```
\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}
```

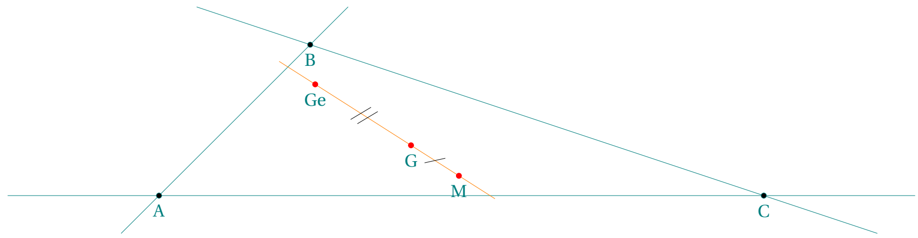
11.1.11. Option mittenpunkt

Le mittenpunkt (également appelé le point des milieux) d'un triangle ABC est le point symédian du triangle excentral, c'est-à-dire le point de concurrence M des droites des excentres passant par les milieux des côtés correspondants du triangle.
Source : Weisstein, Eric W. "Mittenpunkt." From MathWorld–A Wolfram Web Resource.




```
\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}
```

The statement provided indicates that the Gergonne point Ge, the centroid G, and the mittenpunkt M of a triangle are collinear, and their distances follow a specific ratio, namely 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. Définition des points par transformation

Ces transformations sont les suivantes :
- translation;
- homothétie;
- réflexion orthogonale ou symétrie;
- symétrie centrale;
- projection orthogonale;

— rotation (en degrés ou en radians);
— inversion par rapport à un cercle.

12.1. \tkzDefPointBy

Le choix des transformations se fait par le biais des options. Il existe deux macros, l'une pour la transformation d'un seul point \tkzDefPointBy et l'autre pour la transformation d'une liste de points \tkzDefPointsBy. Par défaut l'image de A est A′. Par exemple, on écrira :

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

Le résultat est dans **tkzPointResult**

\tkzDefPointBy[⟨options locales⟩](⟨pt⟩)

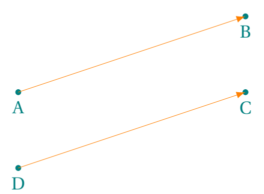
L'argument est un simple point existant et son image est stockée dans **tkzPointResult**. Si vous souhaitez conserver ce point, la macro **\tkzGetPoint{M}** vous permet d'attribuer le nom M au point.

arguments, définitions et exemples		
pt	nom du point existant	(A)

options		exemples
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	...

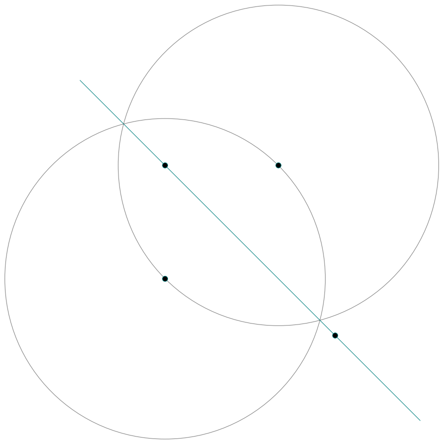
L'image est seulement définie et non dessinée.

12.1.1. translation



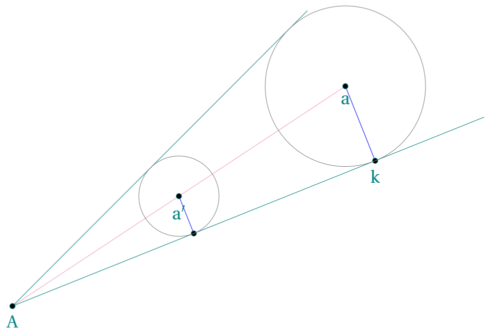

```
\begin{tikzpicture}[>=latex]
 \tkzDefPoints{0/0/A,3/1/B,3/0/C}
 \tkzDefPointBy[translation= from B to A](C)
 \tkzGetPoint{D}
 \tkzDrawPoints[teal](A,B,C,D)
 \tkzLabelPoints[color=teal](A,B,C,D)
 \tkzDrawSegments[orange,->](A,B D,C)
\end{tikzpicture}
```

12.1.2. réflexion (orthogonal symmetry)



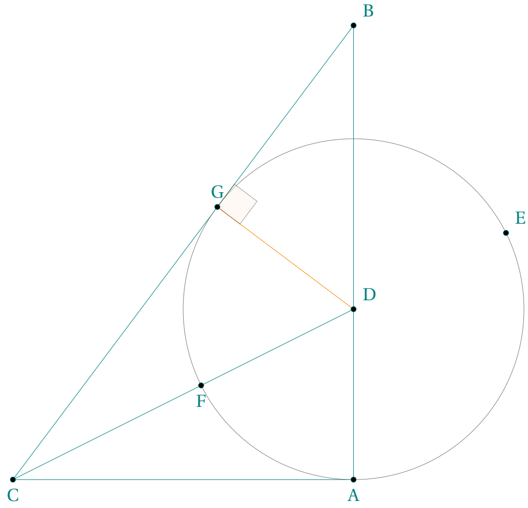
```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoints{-2/-2/A,-1/-1/C,-4/2/D,-4/0/O}
 \tkzDrawCircle(O,A)
 \tkzDefPointBy[reflection = over C--D](A)
 \tkzGetPoint{A'}
 \tkzDefPointBy[reflection = over C--D](O)
 \tkzGetPoint{O'}
 \tkzDrawCircle(O',A')
 \tkzDrawLine[add= .5 and .5](C,D)
 \tkzDrawPoints(C,D,O,O')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

12.1.3. homothétie and projection



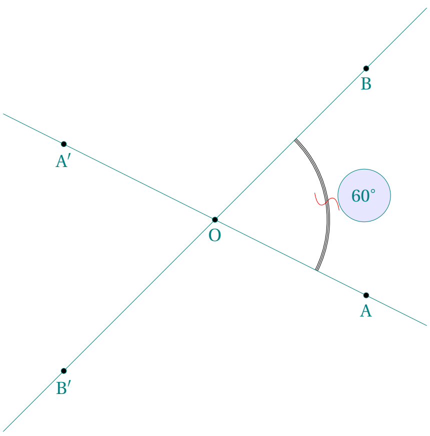

```
\begin{tikzpicture}
  \tkzDefPoints{0/1/A,5/3/B,3/4/C}
  \tkzDefLine[bisector](B,A,C) \tkzGetPoint{a}
  \tkzDrawLine[add=0 and 0,color=magenta!50 ](A,a)
  \tkzDefPointBy[homothety=center A ratio .5](a)
  \tkzGetPoint{a'}
  \tkzDefPointBy[projection = onto A--B](a')
  \tkzGetPoint{k'}
  \tkzDefPointBy[projection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



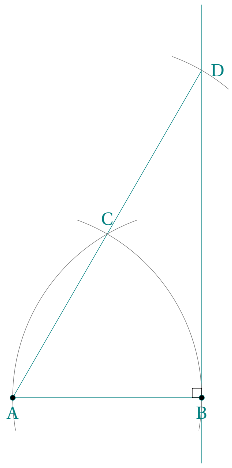

```
\begin{tikzpicture}[scale=1.5]
 \tkzDefPoints{0/0/A,0/4/B}
 \tkzDefTriangle[pythagore](B,A) \tkzGetPoint{C}
 \tkzDefLine[bisector](B,C,A) \tkzGetPoint{c}
 \tkzInterLL(C,c)(A,B)  \tkzGetPoint{D}
 \tkzDefPointBy[projection=onto B--C](D)
 \tkzGetPoint{G}
 \tkzInterLC(C,D)(D,A) \tkzGetPoints{E}{F}
 \tkzDrawPolygon(A,B,C)
 \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. symétrie



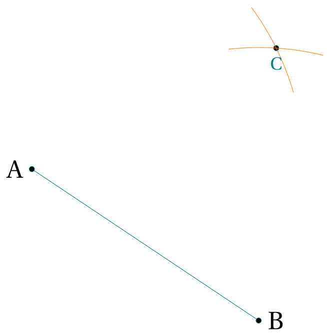

```
\begin{tikzpicture}[scale=1]
  \tkzDefPoints{2/-1/A,2/2/B,0/0/O}
  \tkzDefPointsBy[symmetry=center O](B,A){}
  \tkzDrawLine(A,A')
  \tkzDrawLine(B,B')
  \tkzMarkAngle[mark=s,arc=lll,
      size=1.5,mkcolor=red](A,O,B)
  \tkzLabelAngle[pos=2,circle,draw,
    fill=blue!10,font=\scriptsize](A,O,B){$60^{\circ}$}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

12.1.6. rotation



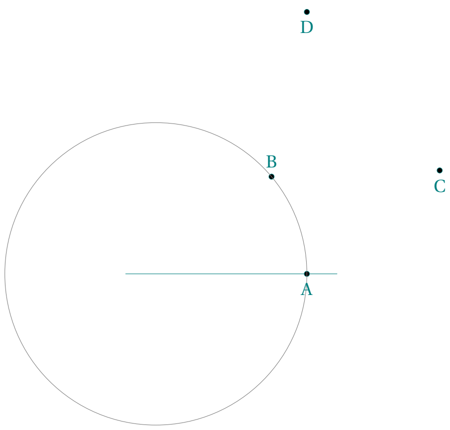

```
\begin{tikzpicture}[scale=0.5]
 \tkzDefPoints{0/0/A,5/0/B}
 \tkzDrawSegment(A,B)
 \tkzDefPointBy[rotation=center A angle 60](B)
 \tkzGetPoint{C}
 \tkzDefPointBy[symmetry=center C](A)
 \tkzGetPoint{D}
 \tkzDrawSegment(A,tkzPointResult)
 \tkzDrawLine(B,D)
 \tkzDrawArc(A,B)(C) \tkzDrawArc(B,C)(A)
 \tkzDrawArc(C,D)(D)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

12.1.7. rotation en radian




```
\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 avec des nodes




```
\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}
```

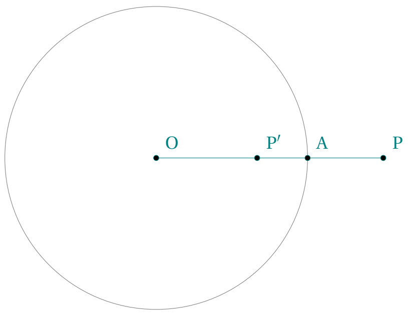
12.1.9. inversion

L'inversion est le processus de transformation de points en un ensemble correspondant de points connus sous le nom de points inverses. Deux points P et P′ sont dits inverses par rapport à un cercle d'inversion dont le centre d'inversion est O et le rayon d'inversion k si P′ est le pied perpendiculaire de l'altitude de OQP, où Q est un point

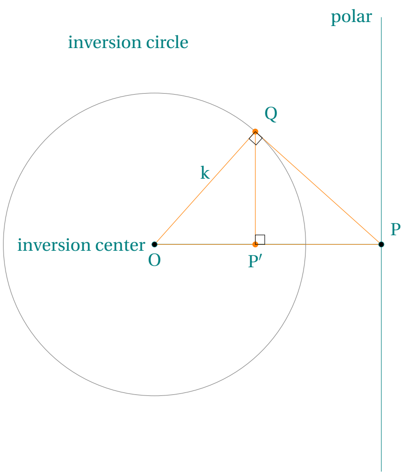
du cercle tel que OQ est perpendiculaire à PQ.
The quantity k^2 is known as the circle power (Coxeter 1969, p. 81).
(https://mathworld.wolfram.com/Inversion.html)
Voici quelques propositions :

— L'inverse d'un cercle (non passant par le centre d'inversion) est un cercle.
— L'inverse d'un cercle passant par le centre d'inversion est une droite.
— L'inverse d'une droite (non passant par le centre d'inversion) est un cercle passant par le centre d'inversion.
— Un cercle orthogonal au cercle d'inversion est son propre inverse.
— Une droite passant par le centre d'inversion est son propre inverse.
— Les angles sont préservés dans l'inversion.

Explication :
Directement (Centre O power=$k^2 = OA^2 = OP \times OP'$)

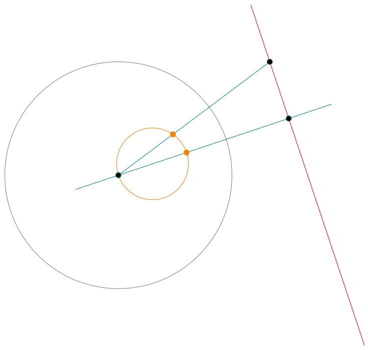



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{4/0/A,6/0/P,0/0/O}
  \tkzDefPointBy[inversion = center O through A](P)
  \tkzGetPoint{P'}
  \tkzDrawSegments(O,P)
  \tkzDrawCircle(O,A)
  \tkzLabelPoints[above right,font=\scriptsize](O,A,P,P')
  \tkzDrawPoints(O,A,P,P')
\end{tikzpicture}
```



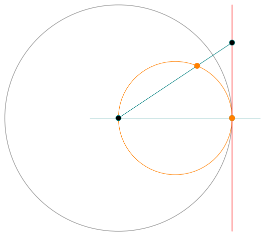

```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{4/0/A,6/0/P,0/0/O}
  \tkzDefLine[orthogonal=through P](O,P)
  \tkzGetPoint{L}
  \tkzDefLine[tangent from = P](O,A) \tkzGetPoints{R}{Q}
  \tkzDefPointBy[projection=onto O--A](Q) \tkzGetPoint{P'}
  \tkzDrawSegments(O,P O,A)
  \tkzDrawSegments[new](O,P O,Q 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



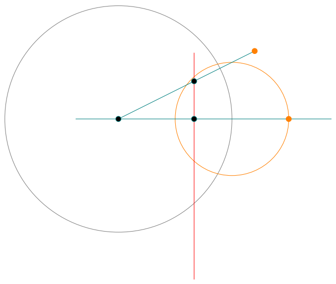
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/O,3/0/I,4/3/P,6/-3/Q}
\tkzDrawCircle(O,I)
\tkzDefPointBy[projection= onto P--Q](O) \tkzGetPoint{A}
\tkzDefPointBy[inversion = center O through I](A)
\tkzGetPoint{A'}
\tkzDefPointBy[inversion = center O through I](P)
\tkzGetPoint{P'}
\tkzDefCircle[diameter](O,A')\tkzGetPoint{o}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



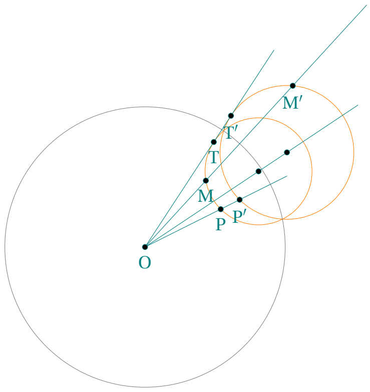
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/O,3/0/I,3/2/P,3/-2/Q}
\tkzDrawCircle(O,I)
\tkzDefPointBy[projection= onto P--Q](O) \tkzGetPoint{A}
\tkzDefPointBy[inversion = center O through I](A)
\tkzGetPoint{A'}
\tkzDefPointBy[inversion = center O through I](P)
\tkzGetPoint{P'}
\tkzDefCircle[diameter](O,A')\tkzGetPoint{o}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



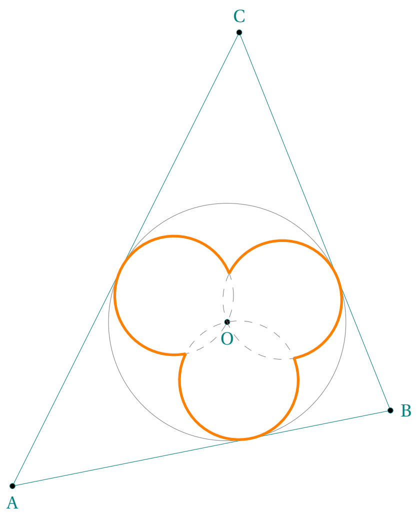
```
\begin{tikzpicture}[scale=.5]
\tkzDefPoints{0/0/O,3/0/I,2/1/P,2/-2/Q}
\tkzDrawCircle(O,I)
\tkzDefPointBy[projection= onto P--Q](O) \tkzGetPoint{A}
\tkzDefPointBy[inversion = center O through I](A)
\tkzGetPoint{A'}
\tkzDefPointBy[inversion = center O through I](P)
\tkzGetPoint{P'}
\tkzDefCircle[diameter](O,A')
\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




```
\begin{tikzpicture}[scale=.5]
\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}
```

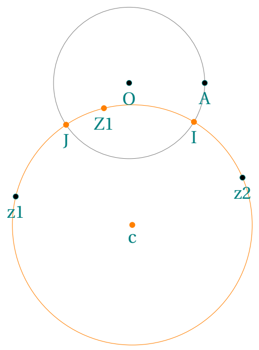
12.1.14. inversion du triangle par rapport à l'Incircle




```
\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: cercle orthogonal avec cercle d'inversion

Le cercle d'inversion lui-même, les cercles qui lui sont orthogonaux et les lignes passant par le centre de l'inversion sont invariants dans l'inversion. Si le cercle rencontre le cercle de référence, ces points d'intersection invariants se trouvent également sur le cercle inverse. Voir I et J dans la figure suivante.

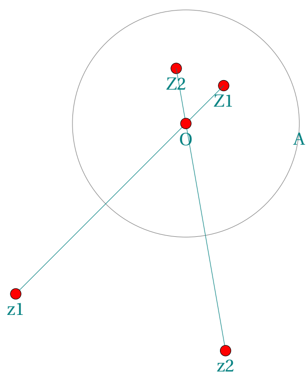



```
\begin{tikzpicture}[scale=1]
\tkzDefPoint(0,0){O}\tkzDefPoint(1,0){A}
\tkzDefPoint(-1.5,-1.5){z1}
\tkzDefPoint(1.5,-1.25){z2}
\tkzDefCircle[orthogonal through=z1 and z2](O,A)
\tkzGetPoint{c}
\tkzDrawCircle[new](c,z1)
\tkzDefPointBy[inversion =  center O through A](z1)
\tkzGetPoint{Z1}
\tkzInterCC(O,A)(c,z1)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

Pour un exemple plus complexe, voir **Pappus** 44.25

12.1.16. inversion negative

It's an inversion followed by a symmetry of center O




```
\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 de plusieurs points ; \tkzDefPointsBy

Variante de la macro précédente pour définir plusieurs images. Vous devez donner les noms des images comme arguments, ou indiquer que les noms des images sont formés à partir des noms des antécédents, en laissant l'argument vide.

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

Les images sont B′ and C′.

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

Les images sont D and E.

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

L'image est B′.

\tkzDefPointsBy[⟨options locales⟩](⟨liste des points⟩){⟨liste des points⟩}

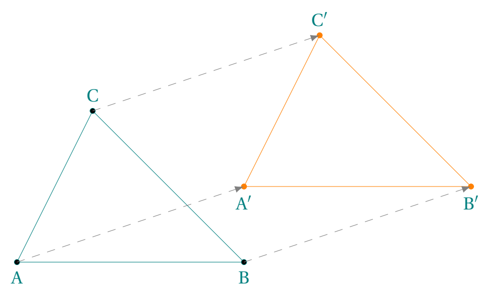
arguments	examples
(⟨liste des points⟩){⟨liste des pts⟩}	(A,B){E,F} E,F images de A, B

Si la liste des images est vide, le nom de l'image est le nom de l'antécédent auquel " ' " est ajouté.

options	exemples
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	...

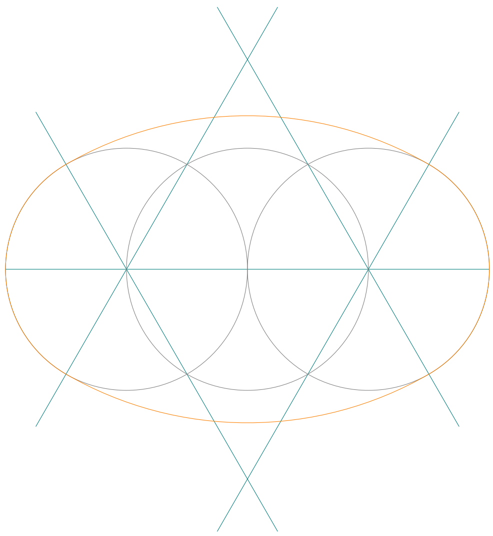
Les points sont seulement définis et non dessinés.

12.2.1. translation of multiple points




```
\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



```
\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}
```

12.3. \tkzDefPointWith

Il existe plusieurs possibilités de créer des points qui répondent à certaines conditions vectorielles. Cela peut être fait avec `\tkzDefPointWith`. Le principe général est le suivant : deux points sont passés en argument, c'est-à-dire un vecteur. Les différentes options permettent d'obtenir un nouveau point formant avec le premier point (à quelques exceptions près) un vecteur colinéaire ou un vecteur orthogonal au premier vecteur. Ensuite, la longueur est soit proportionnelle à celle du premier, soit proportionnelle à l'unité. Comme ce point n'est utilisé que temporairement, il n'a pas besoin d'être nommé immédiatement. Le résultat est dans **tkzPointResult**. La macro `\tkzGetPoint` permet de récupérer le point et de le nommer différemment.
Il existe des options pour définir la distance entre le point donné et le point obtenu. Dans le cas général, cette distance est la distance entre les 2 points donnés en argument. Si l'option est de type "normed", alors la distance entre le point donné et le point obtenu est de 1 cm. Ensuite, l'option K permet d'obtenir des multiples de cette distance.

`\tkzDefPointWith(⟨pt1,pt2⟩)`

Il s'agit en fait de la définition d'un point répondant à des conditions vectorielles.

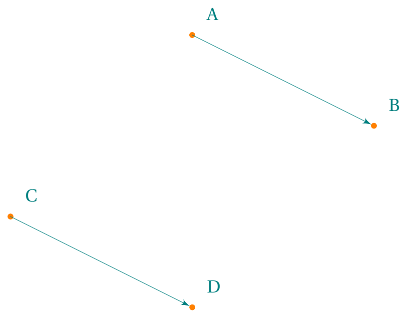
arguments	définition	explication
`(pt1,pt2)`	`couple de points`	`le résultat est un point dans` **tkzPointResult**

Dans ce qui suit, on suppose que le point est récupéré par `\tkzGetPoint{C}`

options	exemple	explication
`orthogonal`	`[orthogonal](A,B)`	$AC = AB$ et $\overrightarrow{AC} \perp \overrightarrow{AB}$
`orthogonal normed`	`[orthogonal normed](A,B)`	$AC = 1$ et $\overrightarrow{AC} \perp \overrightarrow{AB}$
`linear`	`[linear](A,B)`	$\overrightarrow{AC} = K \times \overrightarrow{AB}$
`linear normed`	`[linear normed](A,B)`	$AC = K$ et $\overrightarrow{AC} = k \times \overrightarrow{AB}$
`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`	$\overrightarrow{AC} = 2 \times \overrightarrow{AB}$

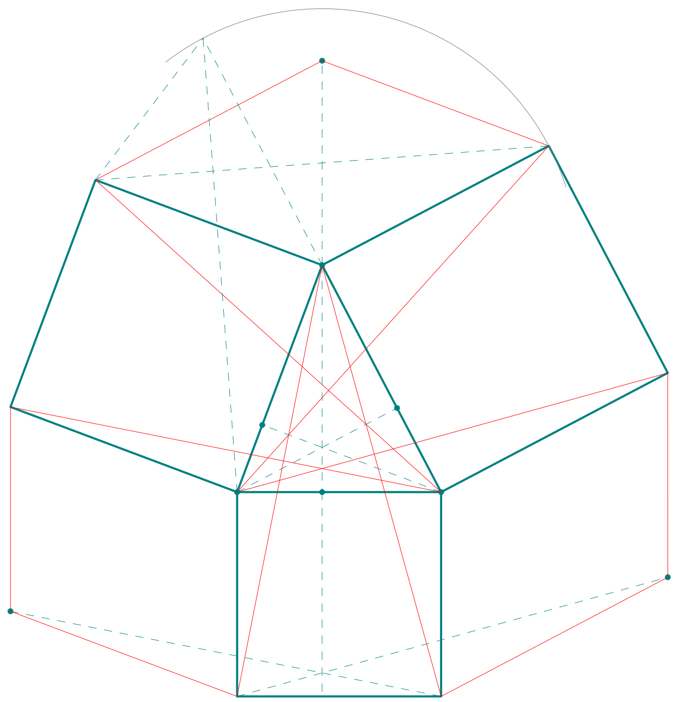
12.3.1. Option colinear at, simple exemple

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




```
\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}
```

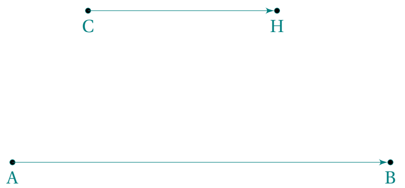
12.3.2. Option colinear at, exemple complexe



```
\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}
```

12.3.3. Option colinear at

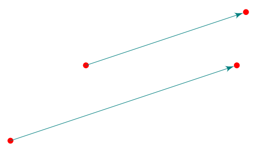
Comment l'utiliser K




```
\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}
```

12.3.4. Option colinear at

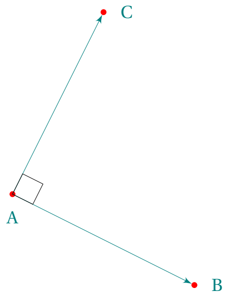
Avec $K = \frac{\sqrt{2}}{2}$



```
\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}
```

12.3.5. Option orthogonal

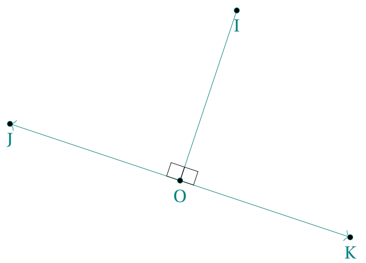
AB=AC car K = 1.




```
\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}
```

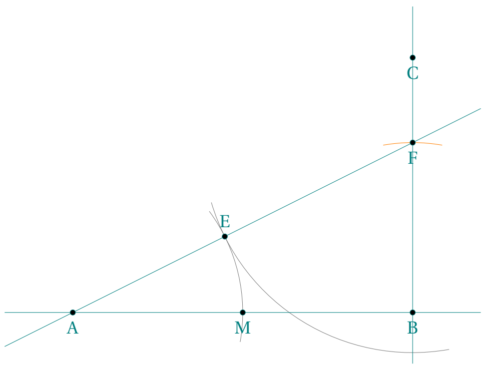
12.3.6. Option **orthogonal**

Avec K = −1 OK=OI car |K| = 1 puis OI=OJ=OK.



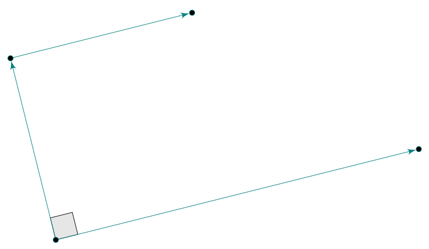

```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{1/2/O,2/5/I}
  \tkzDefPointWith[orthogonal](O,I)
  \tkzGetPoint{J}
  \tkzDefPointWith[orthogonal,K=-1](O,I)
  \tkzGetPoint{K}
  \tkzDrawSegment(O,I)
  \tkzDrawSegments[->](O,J O,K)
  \tkzMarkRightAngles(I,O,J I,O,K)
  \tkzDrawPoints(O,I,J,K)
  \tkzLabelPoints(O,I,J,K)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

12.3.7. Option **orthogonal** exemple plus complexe




```
\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}
```

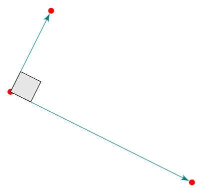
12.3.8. Options colinear et orthogonal



```
\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}
```

12.3.9. Option orthogonal normed

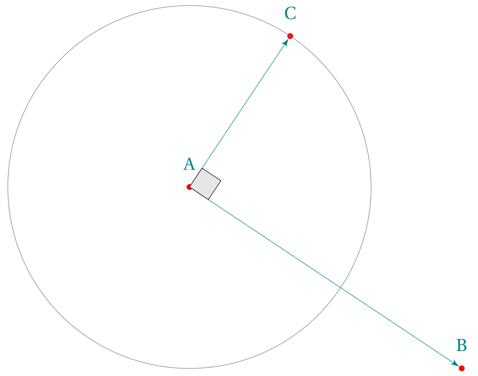
K = 1 AC = 1.



```
\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}
```

12.3.10. Option orthogonal normed et K=2

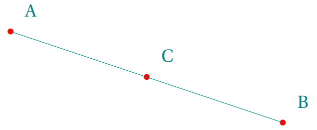
K = 2 therefore AC = 2.




```
\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}
```

12.3.11. Option linear

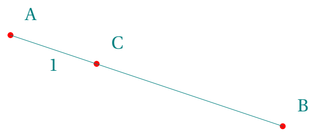
Ici K = 0.5.
Cela revient à appliquer une homothétie ou une multiplication d'un vecteur par un réel. Voici le milieu de [AB].




```
\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}
```

12.3.12. Option linear normed

Dans l'exemple suivant AC = 1 et C appartient à (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}
```

12.4. \tkzGetVectxy

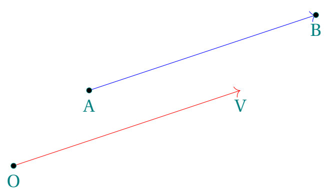
Récupération des coordonnées d'un vecteur.

\tkzGetVectxy(⟨A,B⟩){⟨text⟩}

Allows to obtain the coordinates of a vector.

arguments, exemples et explications		
(point){name of macro}	\tkzGetVectxy(A,B){V}	\Vx,\Vy: coordonnées de $\overrightarrow{AB}$

12.4.1. Coordonnées de transfert\tkzGetVectxy




```
\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}
```

13. Lignes droites

Il est bien sûr essentiel de tracer des lignes droites, mais avant de pouvoir le faire, il est nécessaire de pouvoir définir certaines lignes particulières telles que les médiatrices, les bissectrices, les parallèles ou même les perpendiculaires. Le principe est de déterminer deux points sur la ligne droite.

13.1. Définition des lignes droites

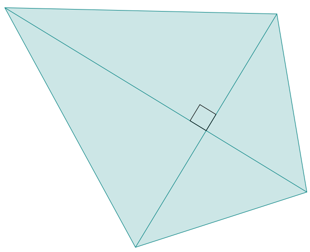
\tkzDefLine[⟨options locales⟩](⟨pt1,pt2⟩) ou (⟨pt1,pt2,pt3⟩)

L'argument est une liste de deux ou trois points. Selon le cas, la macro définit un ou deux points nécessaires pour obtenir la ligne recherchée. Soit la macro **\tkzGetPoint** soit la macro **\tkzGetPoints** doit être utilisée. J'ai utilisé le terme "médiatrice" pour désigner la ligne perpendiculaire bissectrice au milieu d'un segment de droite.

arguments, exemples et explications		
(⟨pt1,pt2⟩)	[mediator](⟨A,B⟩)	médiatrice du segment [A,B]
(⟨pt1,pt2,pt3⟩)	[bisector](⟨A,B,C⟩)	bissectrice de $\widehat{ABC}$
(⟨pt1,pt2,pt3⟩)	[altitude](⟨A,B,C⟩)	hauteur de B
(⟨pt1⟩)	[tangent at=A](⟨O⟩)	tangente à A sur le cercle de centre O
(⟨pt1,pt2⟩)	[tangent from=A](⟨O,B⟩)	centre du cercle O à travers B

options	défaut	définition
mediator		bissectrice perpendiculaire d'un segment de droite
perpendicular=through...	mediator	perpendiculaire à une droite passant par un point
orthogonal=through...	mediator	voir ci-dessus
parallel=through...	mediator	parallèle à une droite passant par un point
bisector	mediator	bissectrice d'un angle défini par trois points
bisector out	mediator	bissectrice extérieure
symmedian	mediator	symédiane depuis un sommet
altitude	mediator	hauteur depuis un sommet
euler	mediator	Ligne d'Euler d'un triangle
tangent at	mediator	tangente en un point du cercle
tangent from	mediator	tangent from an exterior point
K	1	coefficient de la droite perpendiculaire
normed	false	normalise le segment créé

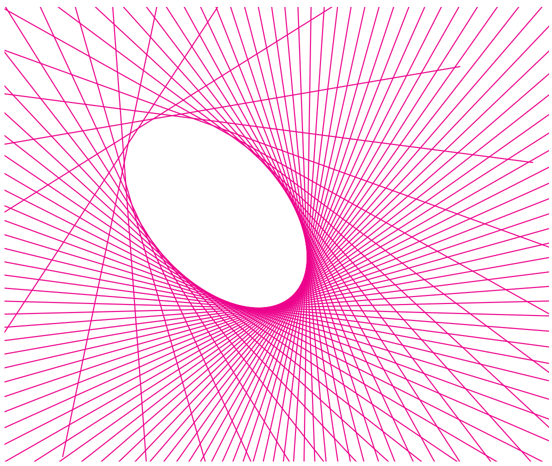
13.1.1. With mediator



```
\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}
```

13.1.2. Une enveloppe avec option mediator

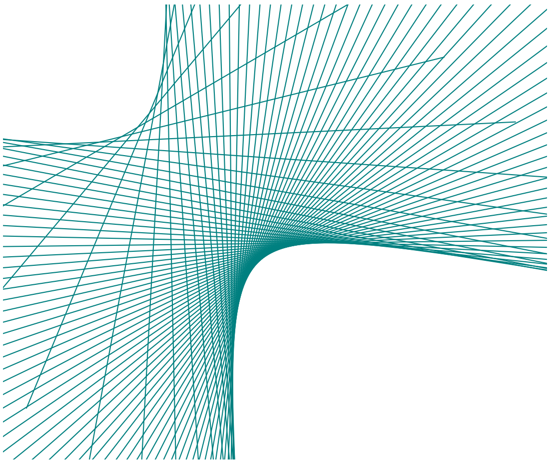
Basé sur une figure de O. Reboux avec pst-eucl par D Rodriguez.



```
\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}
```

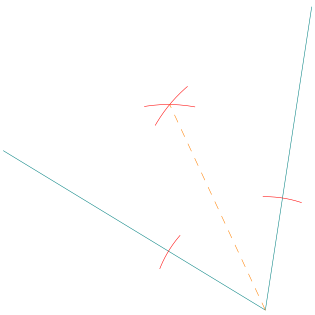
13.1.3. Avec option mediator

Basé sur une figure de O. Reboux avec pst-eucl par D Rodriguez. Il n'est pas nécessaire de nommer les deux points qui définissent la médiatrice.



```
\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}
```

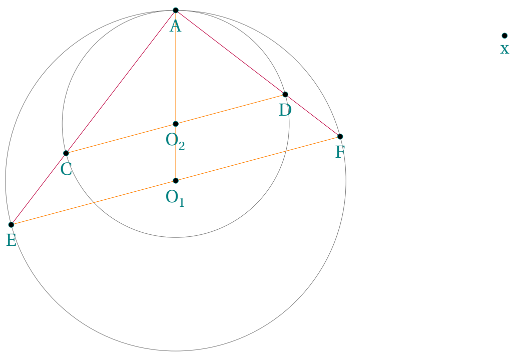
13.1.4. Avec les options bisector et normed



```
\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}
```

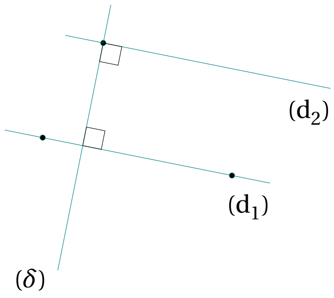
13.1.5. Avec option parallel=through

Archimedes' Book of Lemmas proposition 1



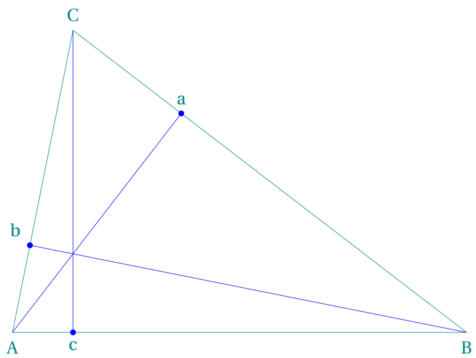

```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O_1,0/1/O_2,0/3/A}
  \tkzDefPoint(15:3){F}
   \tkzDefPointBy[symmetry=center O_1](F) \tkzGetPoint{E}
   \tkzDefLine[parallel=through O_2](E,F) \tkzGetPoint{x}
   \tkzInterLC(x,O_2)(O_2,A) \tkzGetPoints{D}{C}
   \tkzDrawCircles(O_1,A O_2,A)
   \tkzDrawSegments[orange](O_1,A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

13.1.6. Avec option orthogonal et parallel



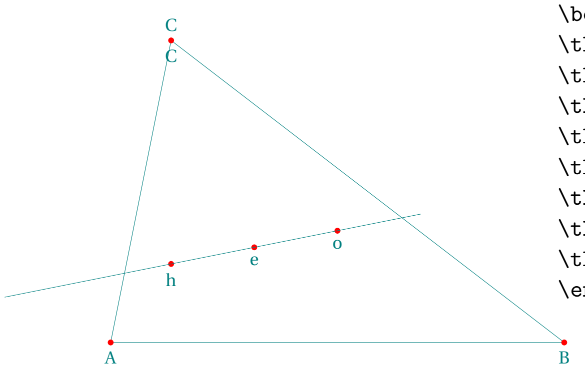

```
\begin{tikzpicture}
   \tkzDefPoints{-1.5/-0.25/A,1/-0.75/B,-0.7/1/C}
   \tkzDrawLine(A,B)
   \tkzLabelLine[pos=1.25,below left](A,B){$(d_1)$}
   \tkzDrawPoints(A,B,C)
   \tkzDefLine[orthogonal=through C](B,A) \tkzGetPoint{c}
   \tkzDrawLine(C,c)
   \tkzLabelLine[pos=1.25,left](C,c){$(\delta)$}
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

13.1.7. Avec option altitude



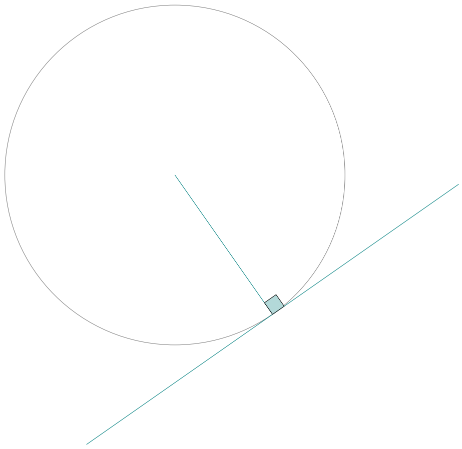

```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
\tkzDefLine[altitude](A,B,C)     \tkzGetPoint{b}
\tkzDefLine[altitude](B,C,A)     \tkzGetPoint{c}
\tkzDefLine[altitude](B,A,C)     \tkzGetPoint{a}
\tkzDrawPolygon(A,B,C)
\tkzDrawPoints[blue](a,b,c)
\tkzDrawSegments[blue](A,a B,b C,c)
\tkzLabelPoints(A,B,c)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

13.1.8. Avec option euler




```
\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}
```

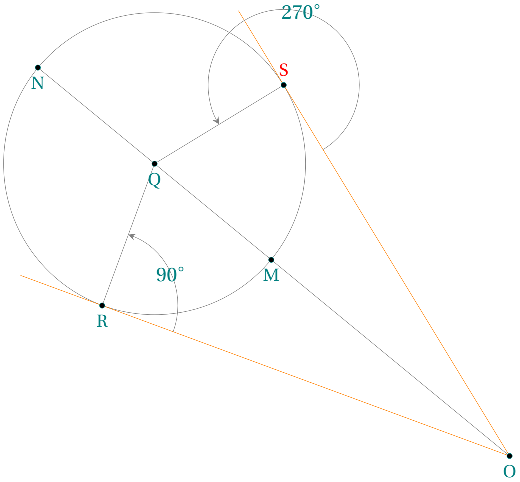
13.1.9. Tangente passant par un point du cercle **tangent at**



```
\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}
```

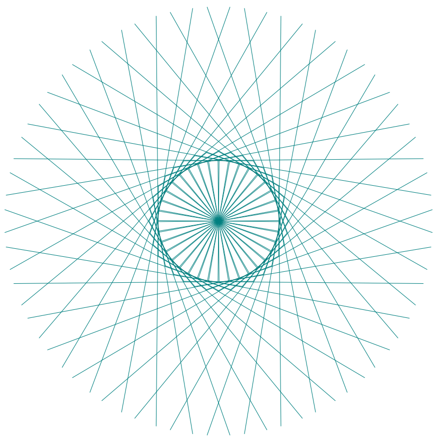
13.1.10. Choix du point de contact avec les tangentes passant par un point externe option **tangent from**

La tangente n'est pas tracée. Avec l'option **at**, un point de la tangente est donné par **tkzPointResult**. Avec l'option **from**, vous obtenez deux points du cercle avec **tkzFirstPointResult** et **tkzSecondPointResult**. Vous pouvez choisir entre ces deux points en comparant les angles formés avec le point extérieur, le point de contact et le centre. Les deux angles possibles ont des directions différentes. L'angle trigonométrique est associé à **tkzFirstPointResult**.



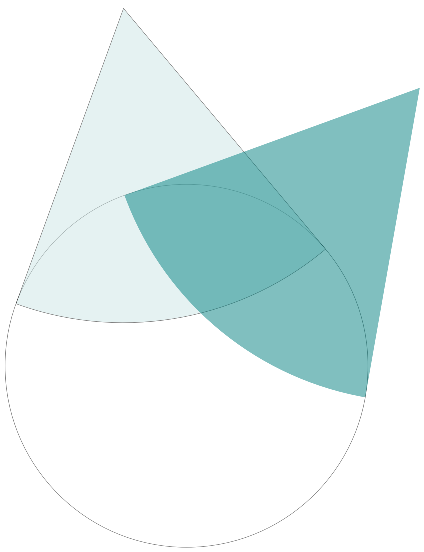

```
\begin{tikzpicture}[scale=1,rotate=-30]
\tkzDefPoints{0/0/Q,0/2/A,6/-1/O}
\tkzDefLine[tangent from = O](Q,A)  \tkzGetPoints{R}{S}
\tkzInterLC[near](O,Q)(Q,A)         \tkzGetPoints{M}{N}
\tkzDrawCircle(Q,M)
\tkzDrawSegments[new,add = 0 and .2](O,R O,S)
\tkzDrawSegments[gray](N,O R,Q S,Q)
\tkzDrawPoints(O,Q,R,S,M,N)
\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}
```

13.1.11. Exemple de tangentes passant par un point extérieur



```
\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}
```

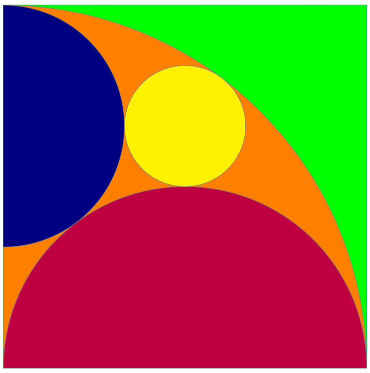
13.1.12. Exemple de Andrew Mertz



```
 \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/

13.1.13. Option de dessin d'une tangente `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}
```

13.2. Définition des triangles `\tkzDefTriangle`

Les macros suivantes vous permettront de définir ou de construire un triangle à partir d'au moins deux points. Pour le moment, il est possible de définir les triangles suivants :

- **`two angles`** détermine un triangle avec deux angles donnés;
- **`equilateral`** détermine un triangle équilatéral;
- **`isosceles right`** détermine un triangle isocèle rectangle;
- **`half`** détermine un triangle rectangle tel que le rapport des mesures des deux côtés adjacents à l'angle droit soit égal à 2;
- **`pythagore`** détermine un triangle rectangle dont les mesures des côtés sont proportionnelles à 3, 4 et 5;
- **`school`** détermine un triangle rectangle dont les angles sont de 30, 60 et 90 degrés;
- **`golden`** détermine un triangle rectangle tel que le rapport des mesures sur les deux côtés adjacents à l'angle droit soit égal à $\Phi = 1.618034$; j'ai choisi "triangle doré" comme dénomination car il provient du rectangle d'or et j'ai gardé la dénomination "triangle d'or" ou "triangle d'Euclide" pour le triangle isocèle dont les angles à la base sont de 72 degrés;
- **`euclid`** ou **`gold`** pour le triangle d'or;
- **`cheops`** détermine un troisième point de telle sorte que le triangle soit isocèle avec des mesures de côtés proportionnelles à 2, Φ et Φ.

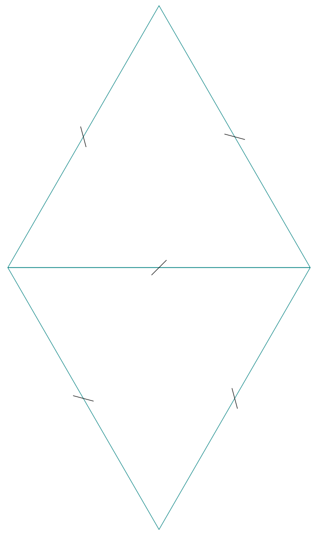
`\tkzDefTriangle[⟨options locales⟩](⟨A,B⟩)`

Les points sont ordonnés car le triangle est construit suivant la direction directe du cercle trigonométrique. Cette macro est utilisée soit en partenariat avec **`\tkzGetPoint`**, soit en utilisant **`tkzPointResult`** s'il n'est pas nécessaire de conserver le nom.

options	défaut	définition
two angles= #1 and #2	no default	triangle connaissant deux angles
equilateral	equilateral	triangle équilatéral
half	equilateral	B rectangle $AB = 2BC$ AC hypothénuse
isosceles right	equilateral	triangle droit isocèle
pythagore	equilateral	proportionnel au triangle pythagoricien 3-4-5
pythagoras	equilateral	Idem que ci-dessus
egyptian	equilateral	Idem que ci-dessus
school	equilateral	angles de 30, 60 et 90 degrés
gold	equilateral	B rectangle et $AB/AC = \Phi$
euclid	equilateral	angles de 72, 72 et 36 degrés, A is the apex
golden	equilateral	angles de 72, 72 et 36 degrés, C is the apex
sublime	equilateral	angles de 72, 72 et 36 degrés, C is the apex
cheops	equilateral	AC=BC, AC et BC sont proportionnels à 2 et Φ.
swap	false	donne le point symétrique par rapport à AB

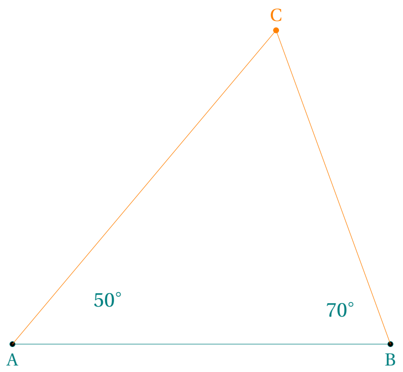
\tkzGetPoint permet de stocker le point, tandis que **tkzPointResult** *permet de l'utiliser immédiatement.*

13.2.1. 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}
```

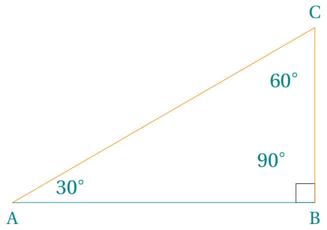
13.2.2. Option two angles




```
\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}
```

13.2.3. Option school

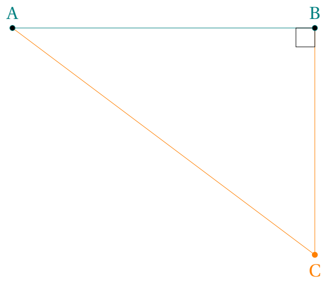
The angles are 30, 60 and 90 degrees.




```
\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}
```

13.2.4. Option `pythagore`

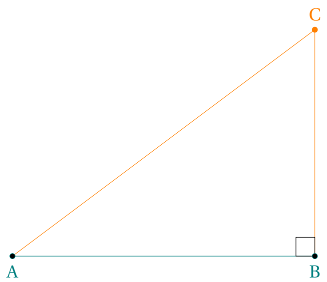
This triangle has sides whose lengths are proportional to 3, 4 and 5.




```
\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}
```

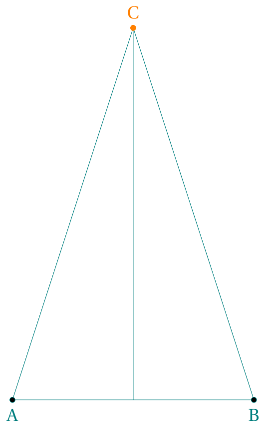
13.2.5. Option `pythagore` et `swap`

Ce triangle a des côtés dont les longueurs sont proportionnelles à 3, 4 et 5.




```
\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}
```

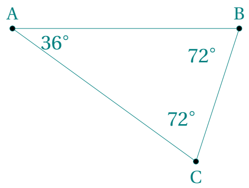
13.2.6. Option golden




```
\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}
```

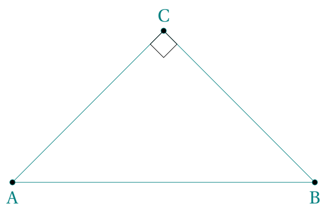
13.2.7. Option euclid

Euclid et golden sont identiques mais le segment AB est une base dans l'un et un côté dans l'autre.



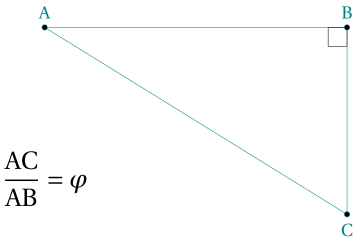

```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(0,0){A} \tkzDefPoint(4,0){B}
 \tkzDefTriangle[euclid](A,B)\tkzGetPoint{C}
 \tkzDrawPolygon(A,B,C)
 \tkzDrawPoints(A,B,C)
 \tkzLabelPoints(C)
 \tkzLabelPoints[above](A,B)
 \tkzLabelAngle[pos=0.8](A,B,C){$72^\circ$}
 \tkzLabelAngle[pos=0.8](B,C,A){$72^\circ$}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

13.2.8. 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}
```

13.2.9. 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}
```

13.3. Triangles spécifiques avec \tkzDefSpcTriangle

Les centres de certains triangles ont été définis dans la section "points", ici il s'agit de déterminer les trois sommets de triangles spécifiques.

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

L'ordre des points est important! p1p2p3 définit un triangle, puis le résultat est un triangle dont les sommets ont comme référence une combinaison avec **name** et r1, r2, r3. Si **name** est vide, alors les références sont r1, r2 et r3.

options	défaut	définition
orthic	centroid	déterminée par les points extrêmes des altitudes ...
centroid or medial	centroid	intersection des trois médianes du triangle
in or incentral	centroid	déterminée à l'aide des bissectrices
ex or excentral	centroid	déterminé avec l'excentriques
extouch	centroid	formé par les points de tangence avec les excircles
intouch or contact	centroid	formed by the points of tangency of the incircle chacun des sommets
euler	centroid	formé par les points d'Euler sur le cercle à neuf points
symmedial	centroid	points d'intersection des symedians
tangential	centroid	formé par les lignes tangentes au cercle circonscrit
feuerbach	centroid	formé par les points de tangence des neuf points ... cercle avec les cercles exinscrits
name	empty	utilisé pour nommer les sommets

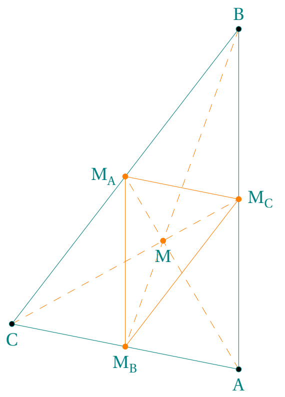
13.3.1. Comment nommer les sommets

Avec `\tkzDefSpcTriangle[medial,name=M](A,B,C){_A,_B,_C}`, vous obtenez trois sommets nommés M_A, M_B et M_C.
Avec `\tkzDefSpcTriangle[medial](A,B,C){a,b,c}`, vous obtenez trois sommets nommés et étiquetés a, b et c.
Possible `\tkzDefSpcTriangle[medial,name=M_](A,B,C){A,B,C}`, vous obtenez trois sommets nommés M_A, M_B et M_C.

13.4. Option medial ou centroid

Le centroïde géométrique des sommets d'un triangle est le point G (parfois aussi noté M) qui est également l'intersection des trois médiatrices du triangle. Le point est donc parfois appelé point médian. Le centroïde est toujours à l'intérieur du triangle.

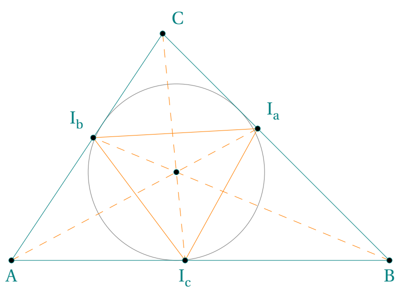
Weisstein, Eric W. "Centroid triangle" From MathWorld–A Wolfram Web Resource.
Dans l'exemple suivant, nous obtenons le cercle d'Euler qui passe par les points définis précédemment.




```
\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}
```

13.4.1. Option `in` ou `incentral`

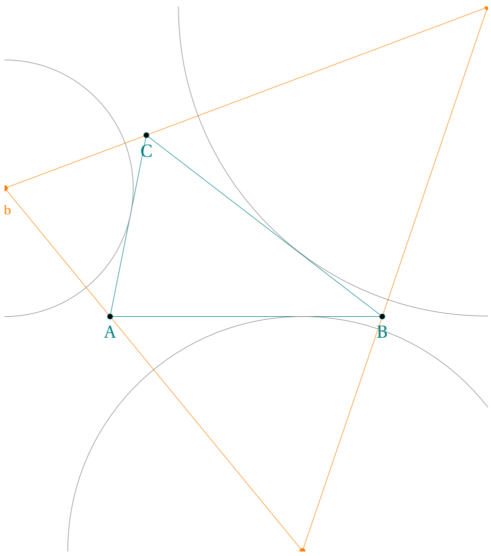
Le triangle incisif est le triangle dont les sommets sont déterminés par les intersections des bissectrices du triangle de référence avec les côtés opposés. côtés opposés respectifs.
Weisstein, Eric W. "Incentral triangle" From MathWorld–A Wolfram Web Resource.




```
\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}
```

13.4.2. Option `ex` ou `excentral`

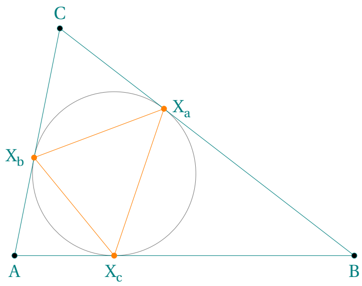
Le triangle excentrique d'un triangle ABC est le triangle $J_aJ_bJ_c$ dont les sommets correspondent aux excentriques de ABC.




```
\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}
```

13.4.3. Option `intouch` ou `contact`

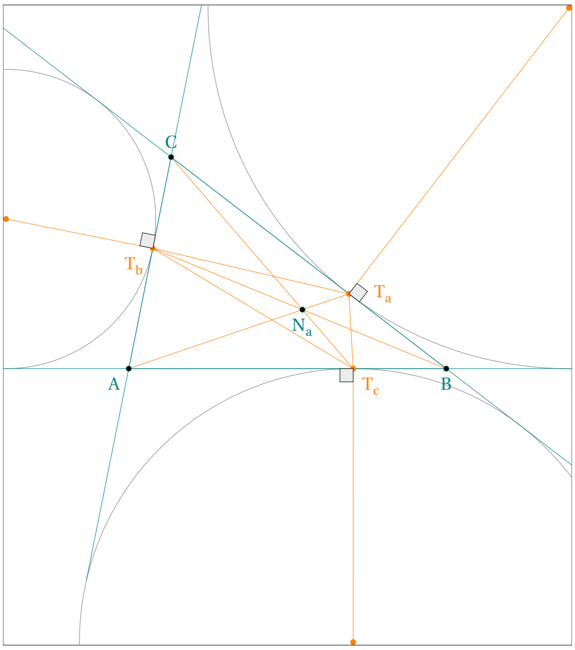
Le triangle de contact d'un triangle ABC, également appelé triangle intouch, est le triangle formé par les points de tangence de l'incircle de ABC avec ABC.
Weisstein, Eric W. "Contact triangle" From MathWorld–A Wolfram Web Resource.
We obtain the intersections of the bisectors with the sides.




```
\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}
```

13.4.4. Option `extouch`

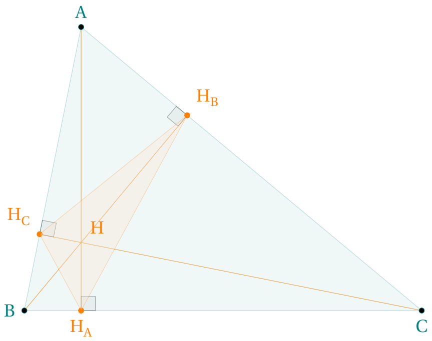
Le triangle exinscrit $T_aT_bT_c$ est le triangle formé par les points de tangence d'un triangle ABC avec ses cercles exinscrits J_a, J_b et J_c. Les points T_a, T_b et T_c peuvent également être construits comme les points qui bisectent le périmètre de $A_1A_2A_3$ en commençant par A, B et C.
Weisstein, Eric W. "Extouch triangle" From MathWorld–A Wolfram Web Resource.
Nous obtenons les points de contact des cercles exinscrits ainsi que le triangle formé par les centres des cercles exinscrits.




```
\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}
```

13.4.5. Option orthic

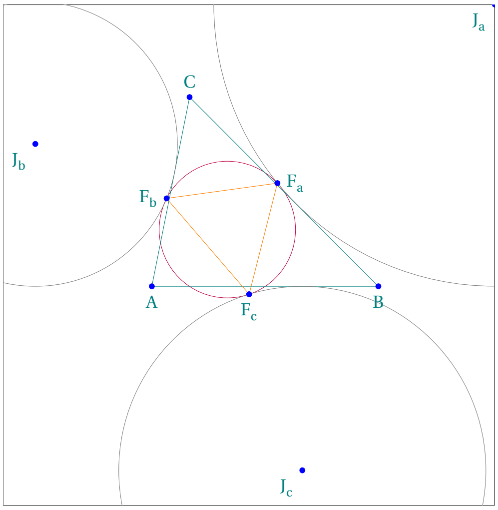
Given a triangle ABC, the triangle $H_AH_BH_C$ whose vertices are endpoints of the altitudes from each of the vertices of ABC is called the orthic triangle, or sometimes the altitude triangle. The three lines AH_A, BH_B, and CH_C are concurrent at the orthocenter H of ABC.




```
\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}
```

13.4.6. Option feuerbach

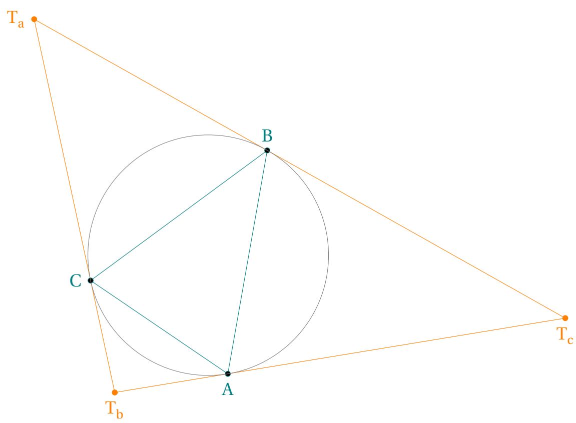
Le triangle de Feuerbach est le triangle formé par les trois points de tangence du cercle de neuf points avec les excircles. Weisstein, Eric W. "Feuerbach triangle" From MathWorld–A Wolfram Web Resource.
Les points de tangence définissent le triangle de Feuerbach.




```
\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}
```

13.4.7. Option tangential

Le triangle tangent est le triangle $T_aT_bT_c$ formé par les droites tangentes au cercle d'un triangle donné ABC en ses sommets. C'est donc le triangle antipédal de ABC par rapport au circoncentre O.
Weisstein, Eric W. "Tangential Triangle." From MathWorld–A Wolfram Web Resource.

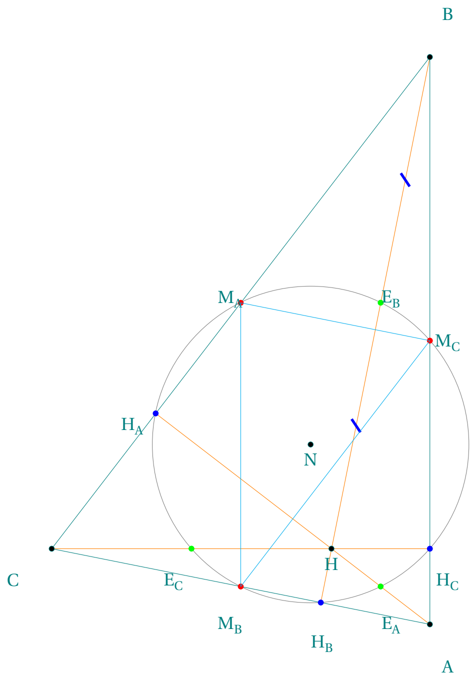



```
\begin{tikzpicture}[scale=.5,rotate=80]
  \tkzDefPoints{0/0/A,6/0/B,1.8/4/C}
  \tkzDefSpcTriangle[tangential,
    name=T](A,B,C){_a,_b,_c}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPolygon[new](T_a,T_b,T_c)
  \tkzDrawPoints(A,B,C)
  \tkzDrawPoints[new](T_a,T_b,T_c)
  \tkzDefCircle[circum](A,B,C)
  \tkzGetPoint{O}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

13.4.8. Option euler

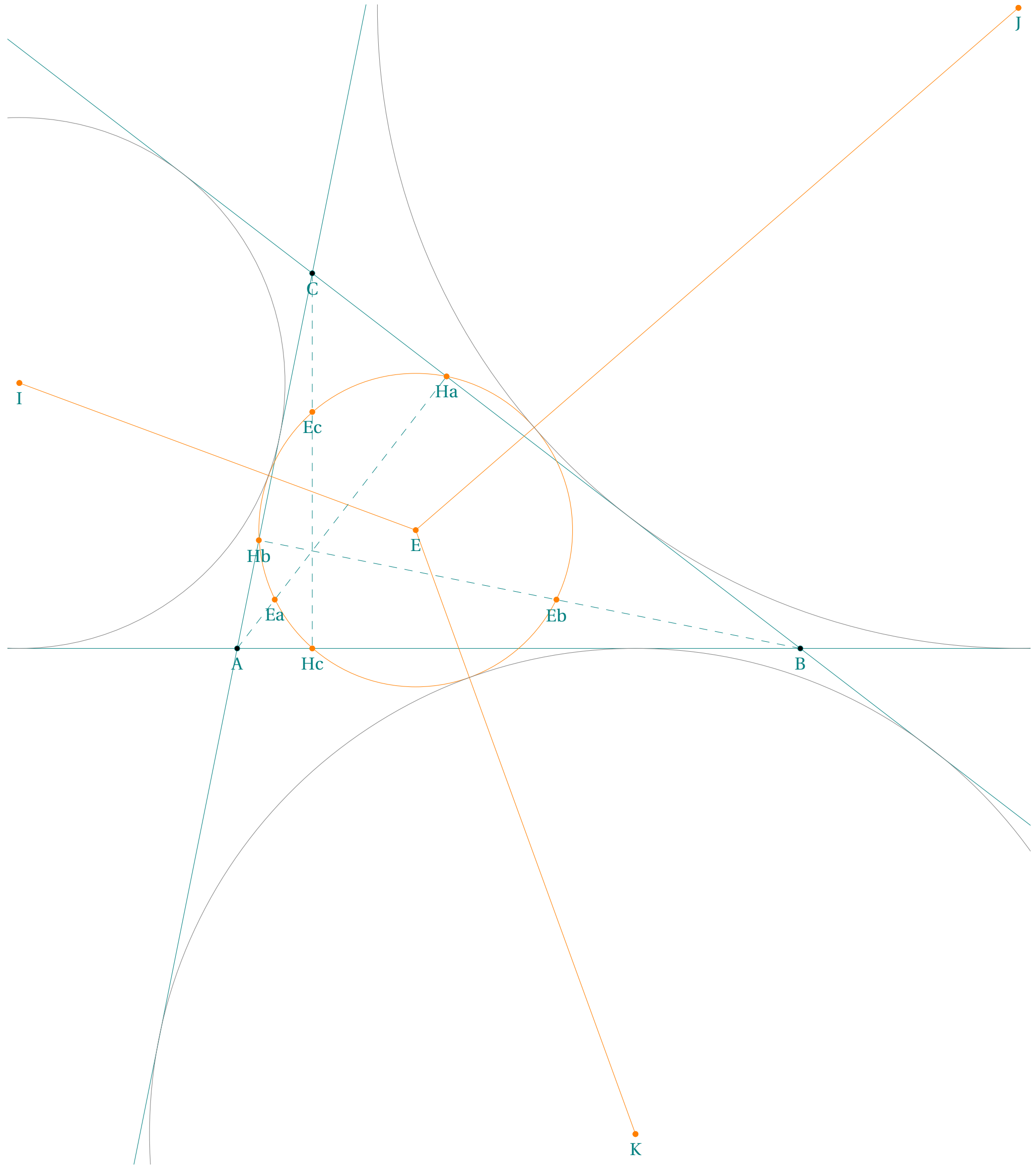
Le triangle d'Euler d'un triangle ABC est le triangle $E_AE_BE_C$ dont les sommets sont les milieux des segments joignant l'orthocentre H aux sommets respectifs. Les sommets du triangle sont appelés points d'Euler et se trouvent

sur le cercle à neuf points.
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}
```

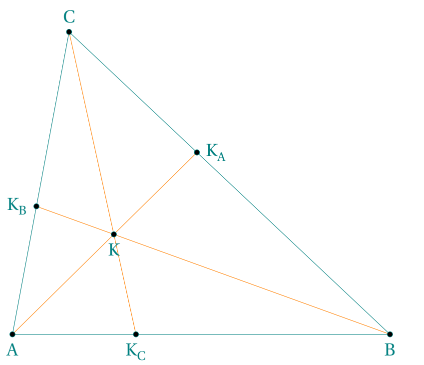
13.4.9. Option euler et option orthic



```
\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}
```

13.4.10. Option `symmedial`

Le triangle symédial$K_AK_BK_C$ est le triangle dont les sommets sont les points d'intersection des symédians avec le triangle de référence ABC. Le triangle symédial$K_AK_BK_C$ est le triangle dont les sommets sont les points d'intersection des symédians avec le triangle de référence ABC.




```
\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}
```

13.5. Permutation de deux points d'un triangle

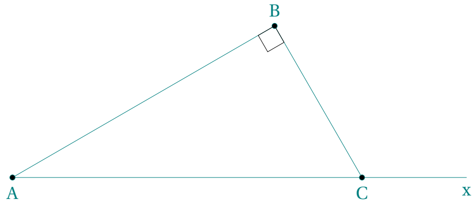
`\tkzPermute(⟨pt1,pt2,pt3⟩)`

arguments	example	explanation
(pt1,pt2,pt3)	`\tkzPermute(A,B,C)`	A, $\widehat{B,A,C}$ sont inchangés, B, C échangent leur position

Le triangle est inchangé.

13.5.1. Modification of the school triangle

Ce triangle est construit à partir du segment [AB] sur [A, x).
Si l'on veut que le segment [AC] soit sur [A, x), il suffit d'intervertir B et C.




```
\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}
```

Remarque : Seul le premier point est inchangé. L'ordre des deux derniers paramètres n'est pas important.

13.6. Définition des points d'un carré

Nous avons vu les définitions de certains triangles. Regardons maintenant les définitions de certains quadrilatères et polygones réguliers.

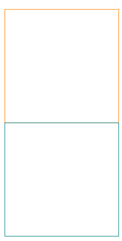
`\tkzDefSquare(⟨pt1,pt2⟩)`

Le carré est défini dans le sens direct. À partir de deux points, deux autres points sont obtenus de telle sorte que les quatre pris dans l'ordre forment un carré. Le carré est défini dans le sens direct.
Les résultats sont dans **tkzFirstPointResult** et **tkzSecondPointResult**.
Nous pouvons les renommer avec **\tkzGetPoints**.

Arguments, exemples et explications		
(⟨pt1,pt2⟩)	**\tkzDefSquare(⟨A,B⟩)**	Le carré est défini dans la direction directe.

13.6.1. Utilisation de \tkzDefSquare avec deux points

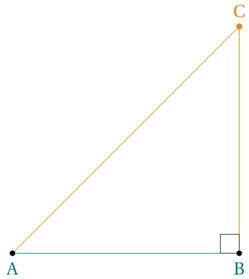
Notez l'inversion des deux premiers points et le résultat.



```
\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}
```

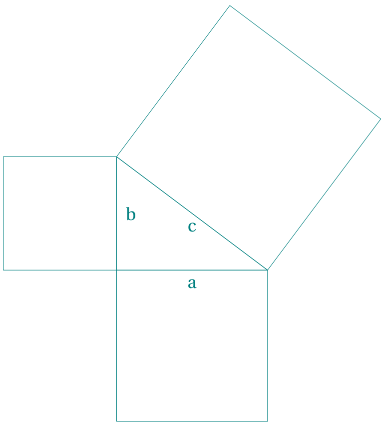
Il se peut que nous n'ayons besoin que d'un seul point pour dessiner un triangle rectangle isocèle. **\tkzGetFirstPoint** or **\tkzGetSecondPoint**.

13.6.2. Utilisation de \tkzDefSquare pour obtenir un triangle rectangle isocèle




```
\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}
```

13.6.3. Théorème de Pythagore et \tkzDefSquare




```
\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}
```

13.7. Définir les points d'un rectangle

\tkzDefRectangle(⟨pt1,pt2⟩)

Le rectangle est défini dans le sens direct. À partir de deux points, deux autres points sont obtenus de telle sorte que les quatre pris dans l'ordre forment un rectangle. Les deux points passés en argument sont les extrémités d'une diagonale du rectangle. Les côtés sont parallèles aux axes.
Les résultats sont dans **tkzFirstPointResult** et **tkzSecondPointResult**.
Nous pouvons les renommer avec **\tkzGetPoints**.

Arguments, exemples et explications		
(⟨pt1,pt2⟩)	**\tkzDefRectangle(⟨A,B⟩)**	Le rectangle est défini dans la direction d

13.7.1. Exemple de définition d'un rectangle

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

13.8. Définition du parallélogramme

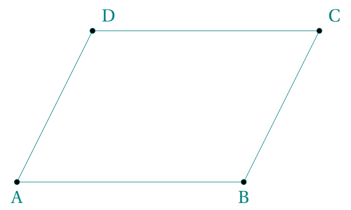
Définition des points d'un parallélogramme. Il s'agit de compléter trois points pour obtenir un parallélogramme.

`\tkzDefParallelogram(⟨pt1,pt2,pt3⟩)`

arguments	défault	définition
(⟨pt1,pt2,pt3⟩)	pas de défaut	Trois points sont nécessaires

A partir de trois points, on obtient un autre point tel que les quatre pris dans l'ordre forment un parallélogramme. The result is in **tkzPointResult**.
Nous pouvons le renommer avec le nom `\tkzGetPoint`...

13.8.1. Exemple de définition d'un parallélogramme




```
\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}
```

13.9. Le rectangle d'or

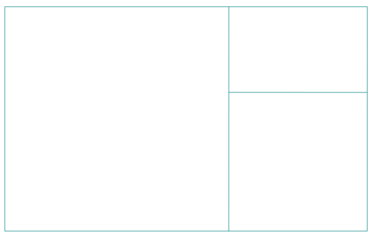
`\tkzDefGoldenRectangle(⟨point,point⟩)`

La macro détermine un rectangle dont le rapport de taille est le nombre Φ. La macro détermine un rectangle dont le rapport de taille est le nombre Φ. Les points créés se trouvent dans **tkzFirstPointResult** et **tkzSecondPointResult**. Ils peuvent être obtenus avec la macro **`\tkzGetPoints`**. La macro suivante est utilisée pour dessiner le rectangle.

arguments, exemples et explications		
(⟨pt1,pt2⟩)	(⟨A,B⟩)	Si C et D sont créés, alors $AB/BC = \Phi$.

`\tkzDefGoldenRectangle` ou `\tkzDefGoldRectangle`

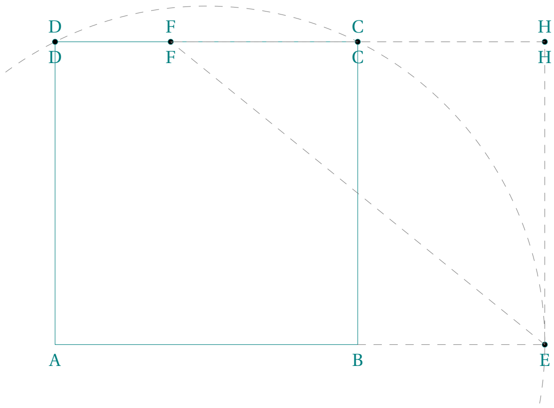
13.9.1. Rectangles d'or



```
\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}
```

13.9.2. Construction du rectangle d'or

Sans la macro précédente, voici comment obtenir le rectangle d'or.




```
\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}
```

13.10. Polygone régulier

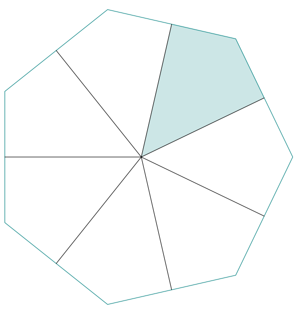
\tkzDefRegPolygon[⟨options locales⟩](⟨pt1,pt2⟩)

A partir du nombre de côtés, en fonction des options, cette macro détermine un polygone régulier en fonction de son centre ou d'un côté.

arguments	exemple	explication
(⟨pt1,pt2⟩)	(⟨O,A⟩)	avec l'option "center", O est le centre du polygone.
(⟨pt1,pt2⟩)	(⟨A,B⟩)	avec l'option "side", [AB] est un côté.

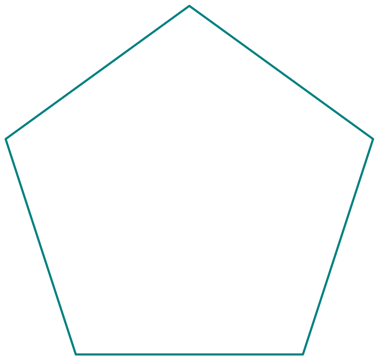
options	défaut	exemple
name	P	Les sommets sont nommés P1,P2,…
sides	5	nombre de côtés.
center	center	Le premier point est le centre.
side	center	Les deux points sont des sommets.
Options TikZ	...	

13.10.1. 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}
```

13.10.2. 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}
```

14. Cercles

Parmi les macros suivantes, l'une d'entre elles vous permettra de dessiner un cercle, ce qui n'est pas une prouesse en soi. Pour ce faire, vous devrez connaître le centre du cercle et soit le rayon du cercle, soit un point sur le cercle. Il m'a semblé que l'utilisation la plus fréquente était de dessiner un cercle avec un centre donné passant par un point donné. Ce sera la méthode par défaut, sinon vous devrez utiliser l'option `R`. Il existe un grand nombre de cercles spéciaux, par exemple le cercle circonscrit à un triangle.

— J'ai créé une première macro `\tkzDefCircle` qui permet, en fonction d'un cercle particulier, de récupérer son centre et la mesure du rayon en cm. Cette récupération se fait avec les macros `\tkzGetPoint` et `\tkzGetLength`;
— ensuite une macro `\tkzDrawCircle`;
— puis une macro qui vous permet de colorier un disque, mais sans dessiner le cercle `\tkzFillCircle`;
— parfois, il est nécessaire qu'un dessin soit contenu dans un disque, c'est le rôle attribué à `\tkzClipCircle`;
— il reste enfin à pouvoir donner un label pour désigner un cercle et si plusieurs possibilités sont offertes, nous verrons ici `\tkzLabelCircle`.

14.1. Caractéristiques d'un cercle : \tkzDefCircle

Cette macro vous permet de récupérer les caractéristiques (centre et rayon) de certains cercles.

`\tkzDefCircle[⟨options locales⟩](⟨A,B⟩) ou (⟨A,B,C⟩)`

Attention, les arguments sont des listes de deux ou trois points. Cette macro est utilisée soit en partenariat avec `\tkzGetPoints` pour obtenir le centre et un point sur le cercle, soit en utilisant `tkzFirstPointResult` et `tkzSecondPointResult` s'il n'est pas nécessaire de conserver les résultats. Vous pouvez également utiliser `\tkzGetLength` pour obtenir le rayon.

arguments, exemples et explications		
(⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)	(⟨A,B⟩)	[AB] est le rayon; A est le c

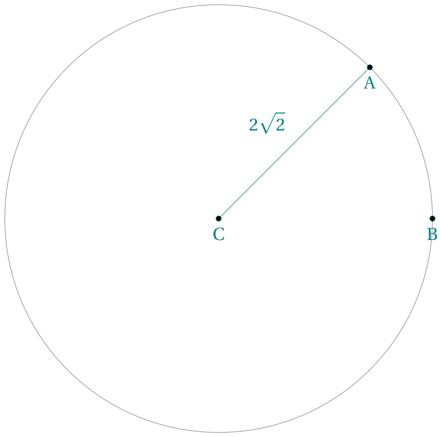
options	défaut	définition
R	circum	cercle caractérisé par un centre et un rayon
diameter	circum	cercle caractérisé par deux points définissant un diamètre
circum	circum	cercle circonscrit
in	circum	cercle inscrit
ex	circum	cercle exinscrit
euler or nine	circum	cercle d'Euler
spieker	circum	cercle de Spieker
apollonius	circum	cercle d'Apollonius
orthogonal from	circum	[orthogonal from = A](O,M)
orthogonal through	circum	[orthogonal through = A and B](O,M)
K	1	coefficient utilisé pour un cercle d'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.

Dans les exemples suivants, je dessine les cercles à l'aide d'une macro qui n'a pas encore été présentée. Vous pouvez n'avoir besoin que du centre et d'un point sur le cercle.

14.1.1. Exemple avec option R

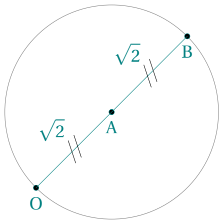
On obtient avec la macro `\tkzGetPoint` un point du cercle qui est le pôle Est.




```
\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}
```

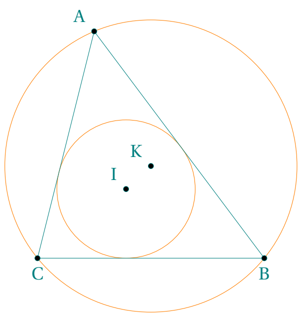
14.1.2. Exemple avec l'option diameter

Il est plus simple ici de rechercher directement le milieu de [AB]. Le résultat est le centre et si nécessaire




```
\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}
```

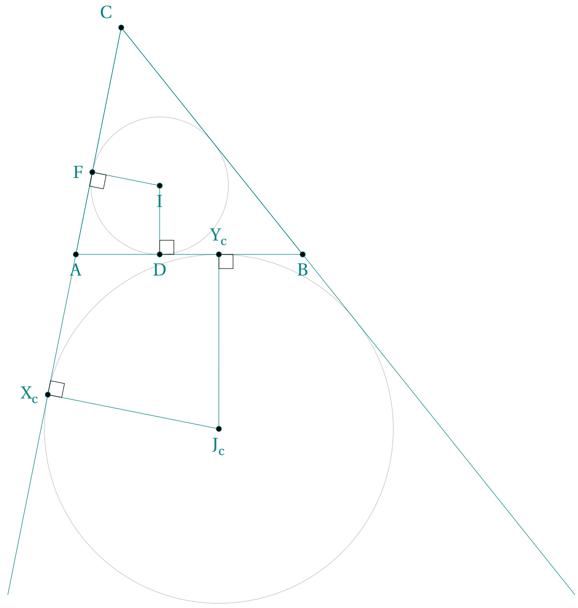
14.1.3. Cercles inscrits et circonscrits à un triangle donné




```
\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}
```

14.1.4. Exemple avec option ex

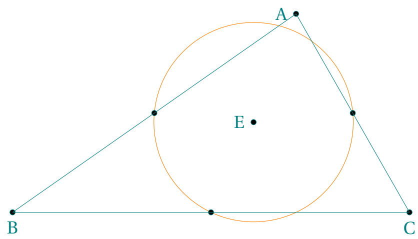
Nous voulons définir un excircle d'un triangle par rapport au point C




```
\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)
  \tkzMarkRightAngles(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}
```

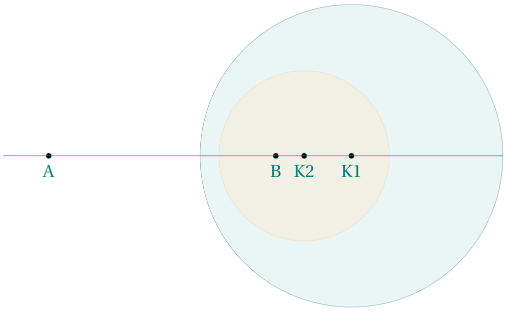
14.1.5. Cercle d'Euler pour un triangle donné avec option `euler`

Nous vérifions que ce cercle passe par le milieu de chaque côté.




```
\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}
```

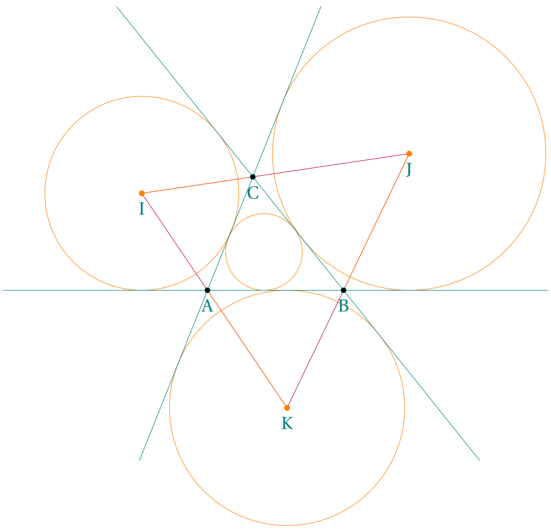
14.1.6. Cercles d'Apollonius pour une option de segment donné apollonius




```
\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}
```

14.1.7. Cercles exinscrits à un triangle donné option ex

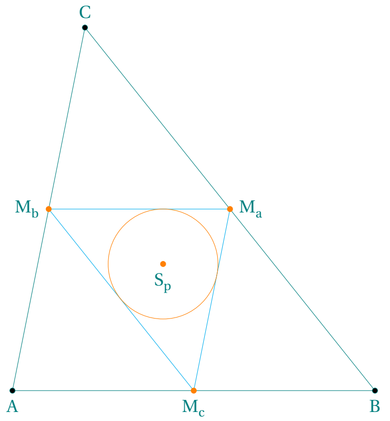
Vous pouvez également obtenir le centre et sa projection sur l'un des côtés du triangle. avec `\tkzGetFirstPoint{Jb}` et `\tkzGetSecondPoint{Tb}`.




```
\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}
```

14.1.8. Cercle de Spieker avec l'option spieker

Le cercle inscrit du triangle médian $M_aM_bM_c$ est le cercle de Spieker :




```
\begin{tikzpicture}[scale=1.2]
  \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}
```

14.2. Projection des excentres

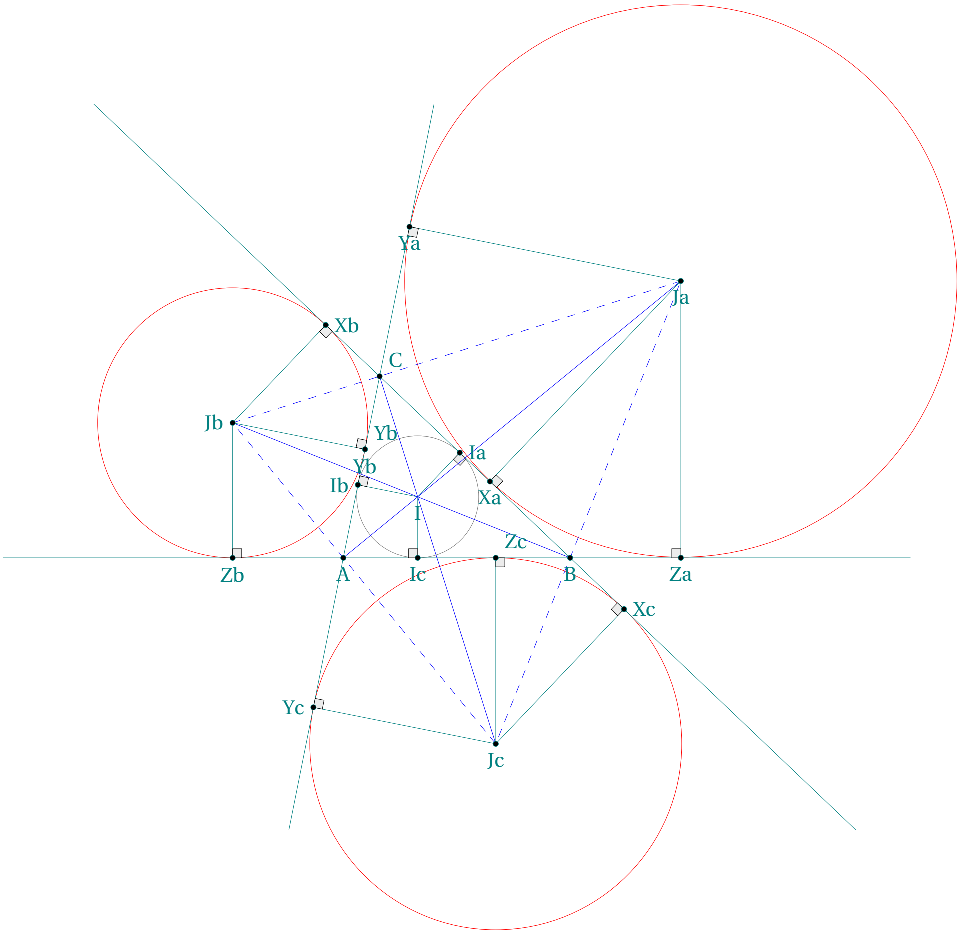
`\tkzDefProjExcenter[⟨options locales⟩](⟨A,B,C⟩)(⟨a,b,c⟩){⟨X,Y,Z⟩}`

Chaque excentrique a trois projections sur les côtés du triangle ABC. Nous pouvons faire cela avec une macro **`\tkzDefProjExcenter[name=J](A,B,C)(a,b,c){Y,Z,X}`**.

options	défaut	définition
`name`	`no defaut`	`used to name the vertices`

arguments	default	definition
`(pt1=`α_1`,pt2=`α_2`,…)`	`pas de défaut`	`Chaque point est affecté d'un poids`

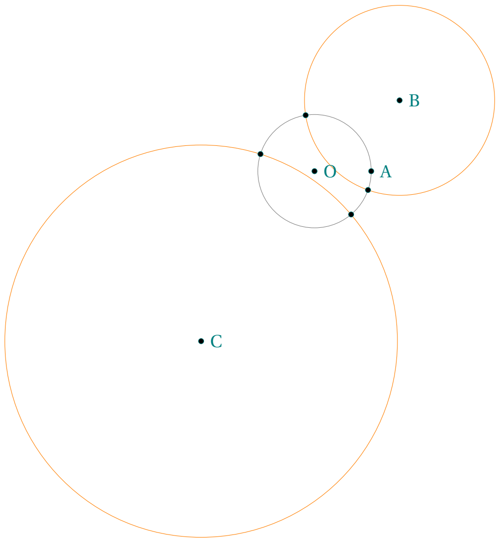
14.2.1. Cercles exinscrits



```
\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)
\tkzMarkRightAngles[size=.2,fill=gray!15](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}
```

14.2.2. Orthogonal from

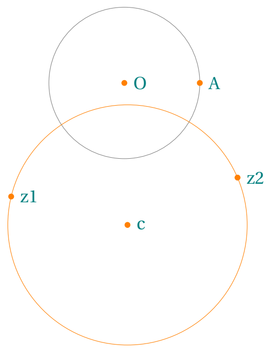
Cercle orthogonal de centre donné. `\tkzGetPoints{z1}{z2}` donne deux points du cercle.




```
\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}
```

14.2.3. Orthogonal through

Cercle orthogonal passant par deux points donnés.




```
\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}
```

14.3. Définition du cercle par transformation ; \tkzDefCircleBy

Ces transformations sont les suivantes :
- translation (translation) ;
- homothety (homothétie) ;
- orthogonal reflection or symmetry (réflexion orthogonale ou symétrie) ;
- central symmetry (symétrie centrale) ;
- orthogonal projection (projection orthogonale) ;
- rotation (degrees) (rotation (degrés)) ;
- inversion (inversion).

Le choix des transformations se fait à travers les options. La macro est `\tkzDefCircleBy` et les autres pour la transformation d'une liste de points `\tkzDefCirclesBy`. Par exemple, nous écrirons :

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

O est le centre et M est un point sur le cercle. L'image est un cercle. Le nouveau centre est **tkzFirstPointResult** et **tkzSecondPointResult** est un point sur le nouveau cercle. Vous pouvez obtenir les résultats avec la macro `\tkzGetPoints`.

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

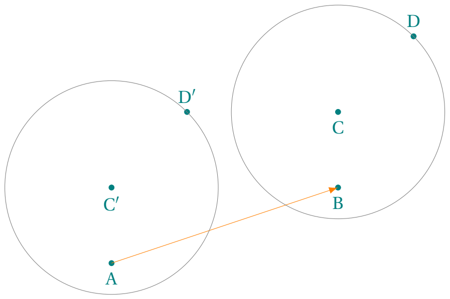
L'argument est un couple de points. Le résultat est un couple de points. Si vous souhaitez conserver ces points, alors la macro `\tkzGetPoints{O'}{M'}` vous permet d'assigner le nom `O'` au centre et `M'` au point sur le cercle.

arguments, définitions et exemples		
pt1,pt2	points existants	(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)

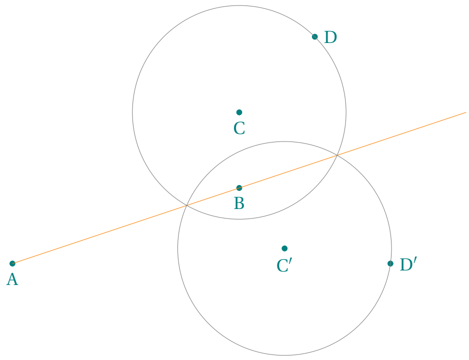
L'image est seulement définie et non dessinée.

14.3.1. Translation



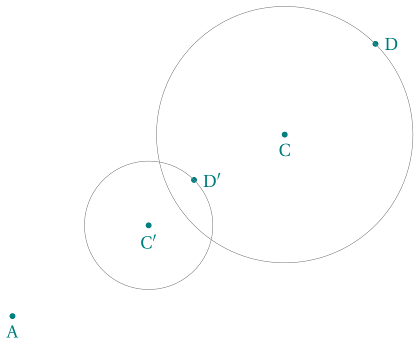

```
\begin{tikzpicture}[>=latex]
 \tkzDefPoint(0,0){A}  \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[translation= from B to A](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](A,B,C,D,C',D')
 \tkzDrawSegments[orange,->](A,B)
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,B,C,C')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

14.3.2. Reflection (orthogonal symmetry)



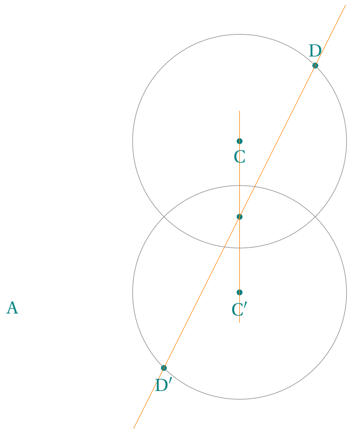

```
\begin{tikzpicture}[>=latex]
 \tkzDefPoint(0,0){A}  \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[reflection = over A--B](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](A,B,C,D,C',D')
 \tkzDrawLine[add =0 and 1][orange](A,B)
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,B,C,C')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

14.3.3. Homothety



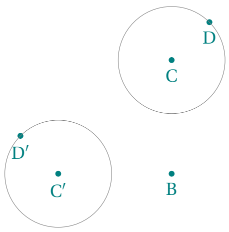

```
\begin{tikzpicture}[scale=1.2]
 \tkzDefPoint(0,0){A}   \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[homothety=center A ratio .5](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](A,C,D,C',D')
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,C,C')
 \tkzLabelPoints[color=teal,right](D,D')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

14.3.4. Symmetry



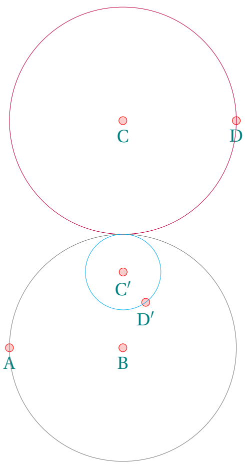

```
\begin{tikzpicture}[scale=1]
 \tkzDefPoint(0,0){A}   \tkzDefPoint(3,1){B}
 \tkzDefPoint(3,2){C}   \tkzDefPoint(4,3){D}
 \tkzDefCircleBy[symmetry=center B](C,D)
 \tkzGetPoints{C'}{D'}
 \tkzDrawPoints[teal](B,C,D,C',D')
 \tkzDrawLines[orange](C,C' D,D')
 \tkzDrawCircles(C,D C',D')
 \tkzLabelPoints[color=teal](A,C,C')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

14.3.5. Rotation




```
\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}
```

14.3.6. Inversion




```
\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}
```

15. Intersections

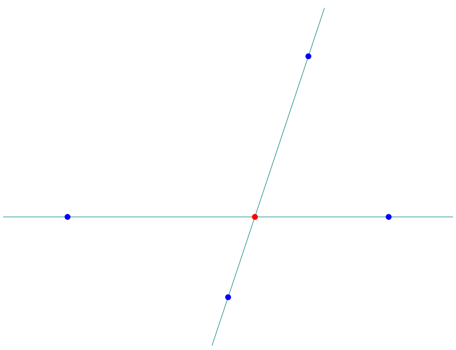
Il est possible de déterminer les coordonnées des points d'intersection entre deux droites, une droite et un cercle, et deux cercles.
Les commandes associées n'ont pas d'arguments optionnels et l'utilisateur doit déterminer lui-même l'existence des points d'intersection.

15.1. Intersection de deux droites \tkzInterLL

\tkzInterLL(⟨A, B⟩)(⟨C, D⟩)

Définit le point d'intersection `tkzPointResult` des deux droites (AB) et (CD). Les points connus sont donnés par paires (deux par droite) entre crochets, et le point résultant peut être récupéré avec la macro `\tkzDefPoint`.

15.1.1. Exemple d'intersection entre deux droites



```
\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}
```

15.2. Intersection d'une droite et d'un cercle \tkzInterLC

Comme précédemment, la droite est définie par un couple de points. Le cercle est également défini par un couple :

— (O, C) qui est un couple de points, le premier est le centre et le deuxième est n'importe quel point sur le cercle.

— (O, r) La mesure r est la mesure du rayon.

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

Les arguments concernent donc deux couples.

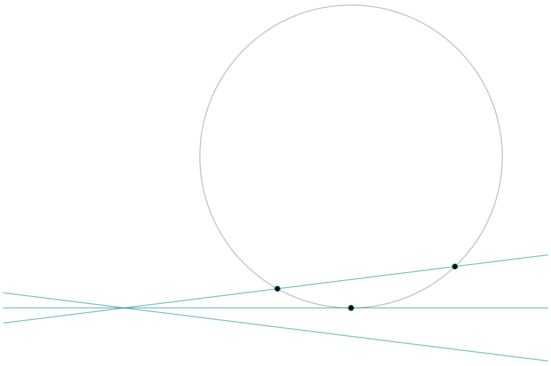
options	défaut	définition
N	N	(O,C) détermine le cercle
R	N	(O, 1) unit 1 cm
with nodes	N	(O,C,D) CD est un rayon
common=pt		pt est un point commun; tkzFirstPoint donne l'autre point
near		tkzFirstPoint sera le plus proche du premier point de la ligne

La macro définit les points d'intersection I et J de la droite (AB) et du cercle central O de rayon r s'ils existent; sinon, une erreur sera signalée dans le fichier `.log`. **avec nœuds** vous évite de calculer le rayon qui est la longueur de [CD]. Si `common` et `near` ne sont pas utilisés, `tkzFirstPoint` est le plus petit angle (angle entre `tkzSecondPoint` et le centre du cercle).

\tkzTestInterLC(⟨O, A⟩)(⟨O′, B⟩)

Les arguments sont donc deux couples qui définissent une ligne et un cercle avec un centre et un point sur le cercle. S'il existe une intersection non vide entre la droite et le cercle, le test `\iftkzFlagLC` donne true.

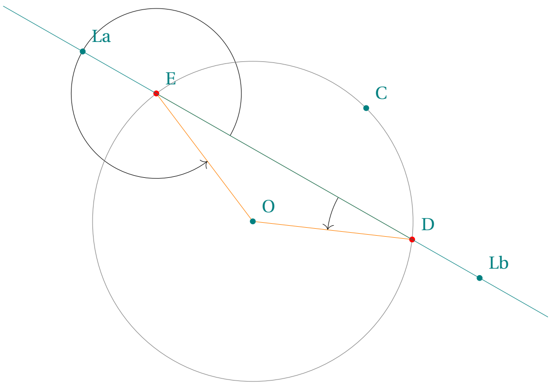
15.2.1. test d'intersection ligne-cercle



```
\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}
```

15.2.2. Intersection ligne-cercle

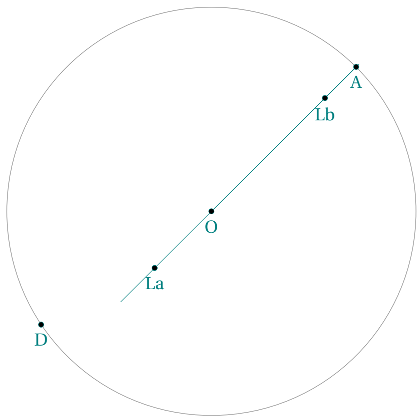
Dans l'exemple suivant, le dessin du cercle utilise deux points et l'intersection de la droite et du cercle utilise deux paires de points. Nous allons comparer les angles $\widehat{D,E,O}$ et $\widehat{E,D,O}$. Ces angles sont dans des directions opposées. **tkzFirstPoint** est assigné au point qui forme l'angle avec la plus petite mesure (dans le sens antihoraire). L'angle dans le sens antihoraire $\widehat{D,E,O}$ a une mesure égale à 360° moins la mesure de $\widehat{O,E,D}$.




```
\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}
```

15.2.3. Droite passant par le centre option `common`

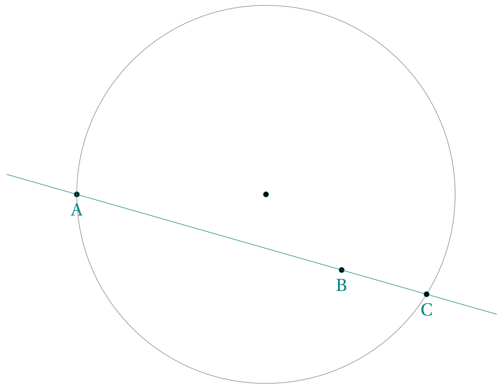
Ce cas est spécial. Vous ne pouvez pas comparer les angles. L'option `near` doit être utilisée.
tkzFirstPoint est assigné au point le plus proche du premier point donné pour la droite. Ici, nous voulons que A soit plus proche de Lb.




```
\begin{tikzpicture}[scale=.75]
\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}
```

15.2.4. Intersection de cercles linéaires avec l'option common

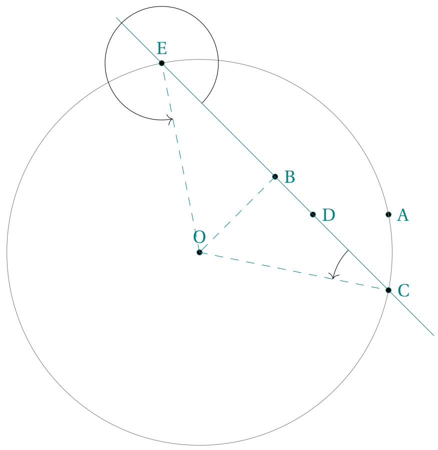
Un cas particulier que l'on rencontre souvent, un point de la droite est sur le cercle et l'on cherche l'autre point commun.




```
\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}
```

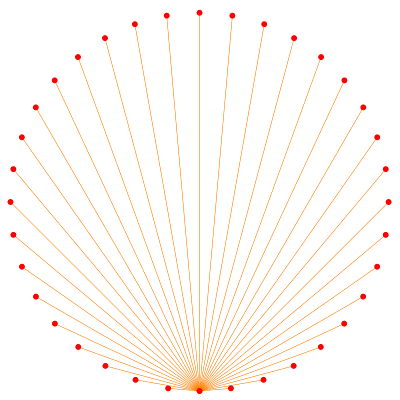
15.2.5. Ordre d'intersection des points du cercle

Il s'agit de comparer les angles formés avec le premier point de définition de la droite, un point résultant et le centre du cercle. Le premier point est celui qui correspond au plus petit angle.
Comme vous pouvez le constater $\widehat{BCO} < \widehat{BEO}$. Pour dire la vérité,$\widehat{BEO}$ est dans le sens inverse des aiguilles d'une montre.




```
\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}
```

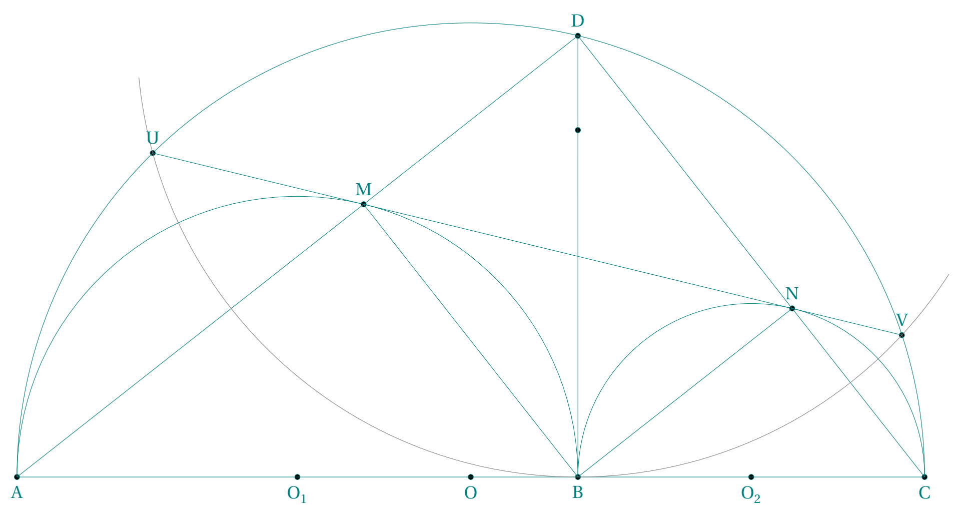
15.2.6. Exemple avec \foreach



```
\begin{tikzpicture}[scale=2.5,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}
```

15.2.7. Intersection de cercles linéaires avec l'option **near**

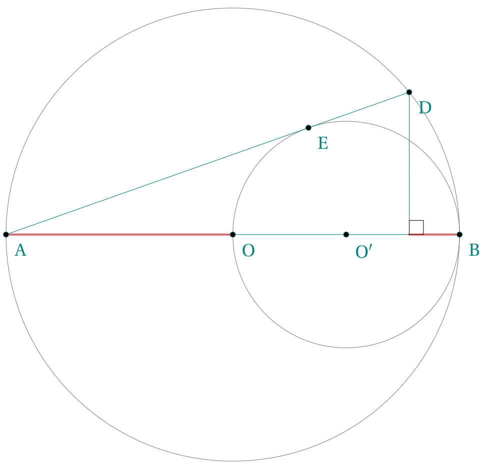
D est le point le plus proche de b.



```
\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}
```

15.2.8. Exemple plus complexe d'intersection ligne-cercle

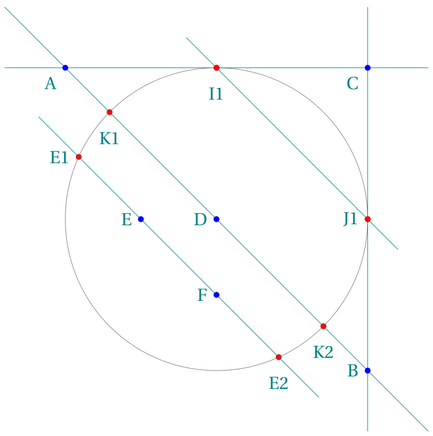
Figure de http://gogeometry.com/problem/p190_tangent_circle




```
\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}
```

15.2.9. Cercle défini par un centre et une mesure, et cas particuliers

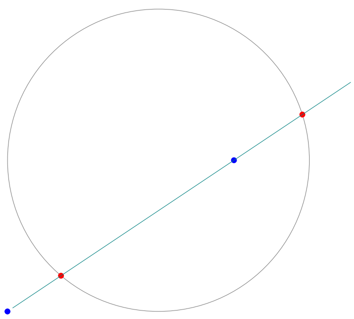
Examinons quelques cas particuliers, comme les droites tangentes au cercle.




```
\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}
```

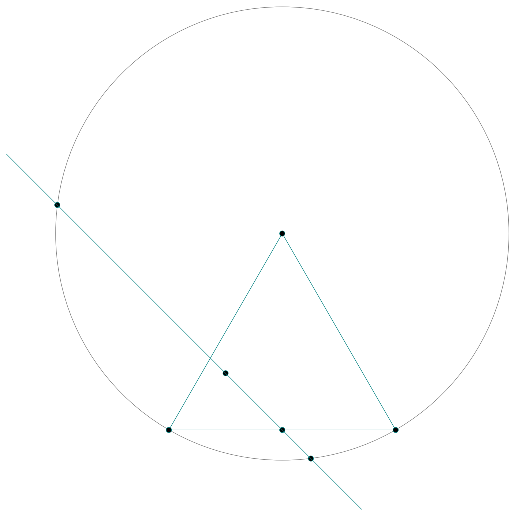
15.2.10. Calcul du rayon

Avec **pgfmath** et `\pgfmathsetmacro`
La mesure du rayon peut être le résultat d'un calcul qui n'est pas effectué dans la macro d'intersection, mais avant. Une longueur peut être calculée de plusieurs manières. Il est bien sûr possible d'utiliser le module **pgfmath** et la macro `\pgfmathsetmacro`. Dans certains cas, les résultats obtenus ne sont pas assez précis, ainsi le calcul suivant $0.0002 \div 0.0001$ donne 1.98 avec pgfmath alors que xfp donnera 2.
Avec **xfp** et `\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}
```

15.2.11. 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}
```

15.3. Intersection de deux cercles \tkzInterCC

Le cas le plus fréquent est celui de deux cercles définis par leur centre et un point, mais comme précédemment l'option `R` permet d'utiliser les mesures de rayon.

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

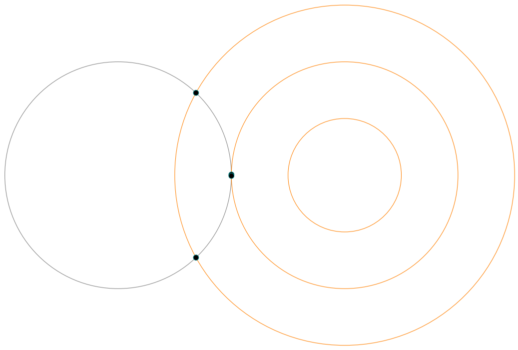
options	default	definition
N	N	OA and O′A′ are radii, O and O′ are the centers.
R	N	r and r′ are dimensions and measure the radii.
with nodes	N	in (A,A,C)(C,B,F) AC and BF give the radii.
common=pt		pt is common point; tkzFirstPoint gives the other point.

Correction : Cette macro définit le(s) point(s) d'intersection I et J des deux cercles centrés sur O et O′. Si les deux cercles n'ont pas de point commun, la macro se termine par une erreur qui n'est pas traitée. Si les centres sont O et O′ et que les intersections sont A et B, alors les angles $\widehat{O,A,O'}$ et $\widehat{O,B,O'}$ sont dans des directions opposées. **tkzFirstPoint** est attribué au point qui forme l'angle "dans le sens des aiguilles d'une montre".

`\tkzTestInterCC(⟨O,A⟩)(⟨O′,B⟩)`

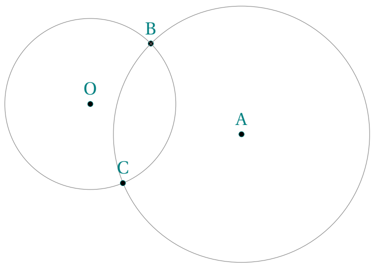
Les arguments sont donc deux couples qui définissent deux cercles avec un centre et un point sur le cercle. S'il existe une intersection non vide entre ces deux cercles, le test **\iftkzFlagCC** donne true.

15.3.1. Test d'intersection des cercles



```
\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}
```

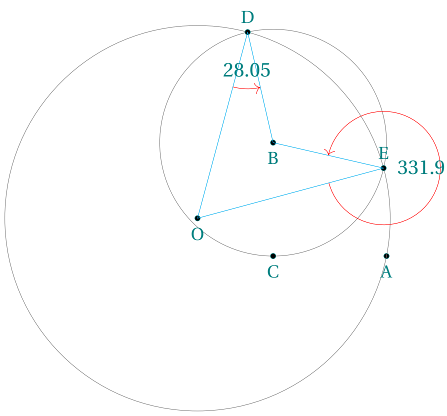
15.3.2. Intersection cercle-cercle avec un point commun.




```
\begin{tikzpicture}[scale=.4]
  \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}
```

15.3.3. Ordre d'intersection des points du cercle.

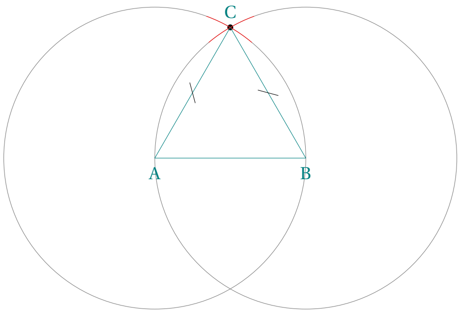
Il s'agit de comparer les angles formés avec le premier centre, un point résultant et le centre du deuxième cercle. Le premier point est celui qui correspond au plus petit angle.
Comme vous pouvez le constater, le premier point est celui qui correspond à l'angle le plus petit. $\widehat{ODB} < \widehat{OBE}$




```
\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}
```

15.3.4. Construction d'un triangle équilatéral.

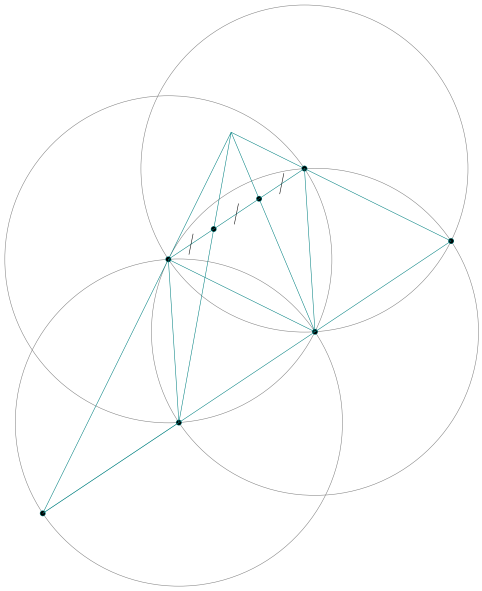
$\widehat{A,C,B}$ is a clockwise angle




```
\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}
```

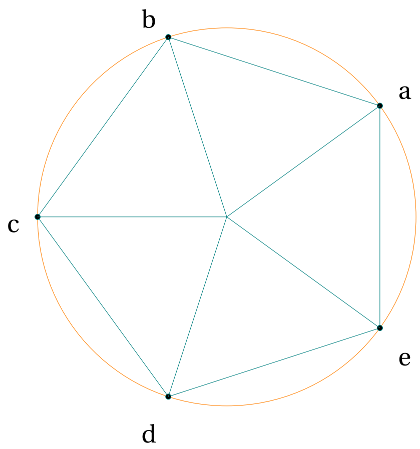
15.3.5. Segment trisection

The idea here is to divide a segment with a ruler and a compass into three segments of equal length.



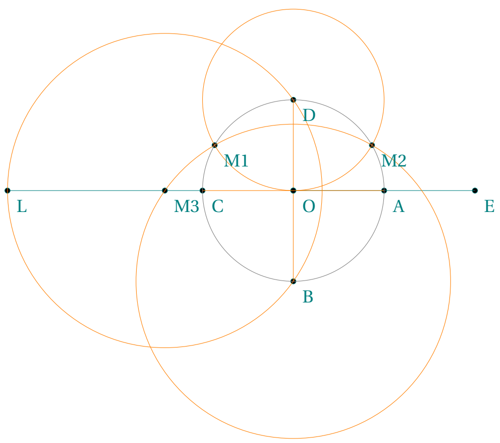
```
\begin{tikzpicture}[scale=.6]
 \tkzDefPoint(0,0){A}
 \tkzDefPoint(3,2){B}
 \tkzInterCC(A,B)(B,A)
 \tkzGetSecondPoint{D}
 \tkzInterCC(D,B)(B,A)          \tkzGetPoints{A}{C}
 \tkzInterCC(D,B)(A,B)          \tkzGetPoints{E}{B}
 \tkzInterLC[common=D](C,D)(E,D)\tkzGetFirstPoint{F}
 \tkzInterLL(A,F)(B,C)          \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}
```

15.3.6. 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}
```

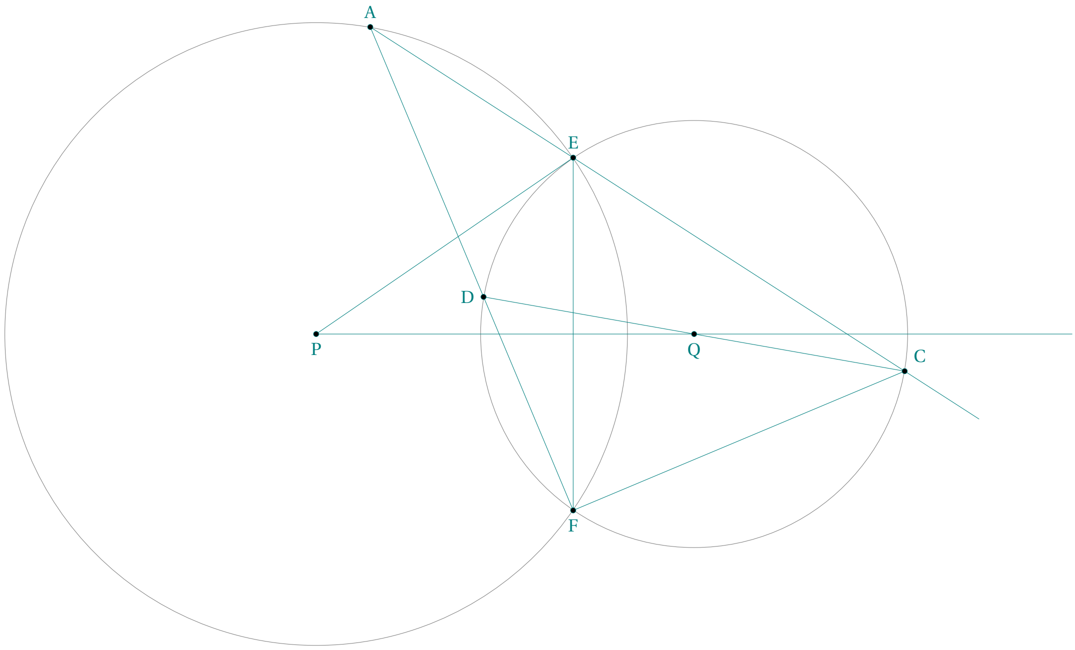
15.3.7. Mix of intersections




```
\begin{tikzpicture}[scale = .6]
  \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}
```

15.3.8. Théorème d'Altshiller-Court

Les deux lignes joignant les points d'intersection de deux cercles orthogonaux à un point de l'un des cercles rencontrent l'autre cercle en deux points diamétralement opposés. Altshiller p 176

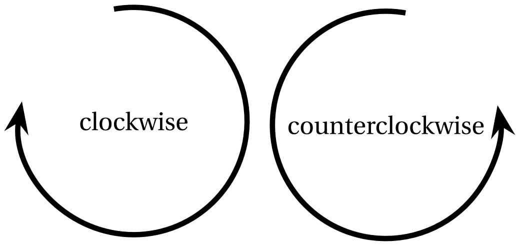



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/P,5/0/Q,3/2/I}
  \tkzDefCircle[orthogonal from=P](Q,I)
  \tkzGetFirstPoint{E}
  \tkzDrawCircles(P,E Q,E)
  \tkzInterCC[common=E](P,E)(Q,E) \tkzGetFirstPoint{F}
  \tkzDefPointOnCircle[through =  center P angle 80 point E]
  \tkzGetPoint{A}
  \tkzInterLC[common=E](A,E)(Q,E)  \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}
```

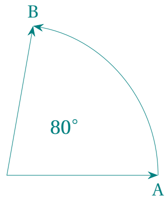
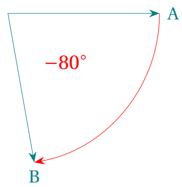
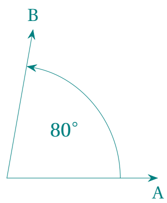
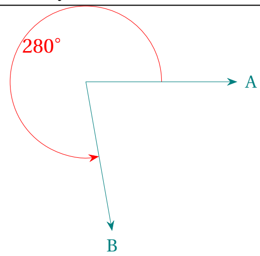
16. Angles

16.1. Définition et utilisation avec tkz-euclide

En géométrie euclidienne, un angle est la figure formée par deux demi-droires, appelés côtés de l'angle, partageant une extrémité commune, appelée sommet de l'angle [Wikipedia]. Un rayon avec `tkz-euclide` est défini par deux points et chaque angle est défini par trois points comme $\widehat{AOB}$. Le sommet O est le deuxième point. Leur ordre est important car on suppose que l'angle est spécifié dans l'ordre direct (dans le sens inverse des aiguilles d'une montre). En trigonométrie et en mathématiques en général, les angles plans sont conventionnellement mesurés dans le sens inverse des aiguilles d'une montre, en commençant par 0° pointant directement vers la droite (ou l'est), et 90° pointant directement vers le haut (ou le nord) [Wikipedia]. Convenons qu'un angle mesuré dans le sens inverse des aiguilles d'une montre est positif.



Angles sont impliqués dans plusieurs macros comme `\tkzDefPoint`,`\tkzDefPointBy[rotation = …]`, `\tkzDrawArc` et la suivante `\tkzGetAngle`. À l'exception de la dernière, toutes ces macros acceptent les angles négatifs.

 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°

Comme on peut le voir, la rotation -80° définit un angle dans le sens des aiguilles d'une montre mais la macro `\tkzFindAngle` récupère un angle dans le sens inverse des aiguilles d'une montre.

16.2. Récupération d'un angle \tkzGetAngle

`\tkzGetAngle(⟨name of macro⟩)`

Attribue la valeur en degré d'un angle à une macro. La valeur est positive et comprise entre 0° et 360°. Cette macro récupère `\tkzAngleResult` et stocke le résultat dans une nouvelle macro.

arguments, exemples et explications		
nom de la macro	`\tkzGetAngle{ang}`	`\ang` contient la valeur de l'angle.

This is an auxiliary macro that allows you to retrieve the result of the following macro `\tkzFindAngle`.

16.3. Angle formé par trois points

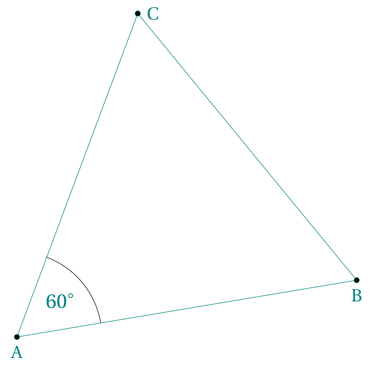
`\tkzFindAngle(⟨pt1,pt2,pt3⟩)`

Le résultat est stocké dans une macro `\tkzAngleResult`.

arguments	exemple	explication
(pt1,pt2,pt3)	`\tkzFindAngle(A,B,C)`	`\tkzAngleResult` donne l'angle $(\overrightarrow{BA},\overrightarrow{BC})$

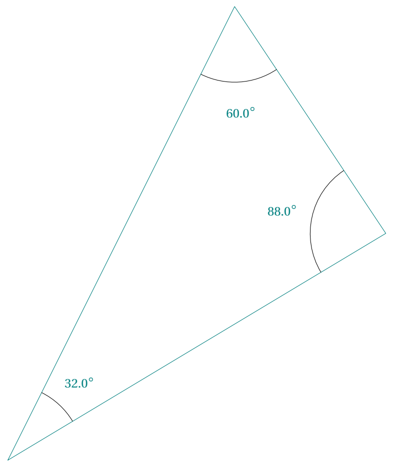
La mesure est toujours positive et comprise entre 0° et 360°. Avec les conventions habituelles, un angle dans le sens inverse des aiguilles d'une montre plus petit qu'un angle droit a toujours une mesure comprise entre 0° et 180°, tandis qu'un angle dans le sens des aiguilles d'une montre plus petit qu'un angle droit aura une mesure supérieure à 180°. **\tkzGetAngle** peut récupérer l'angle.

16.3.1. Verification of angle measurement




```
\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}
```

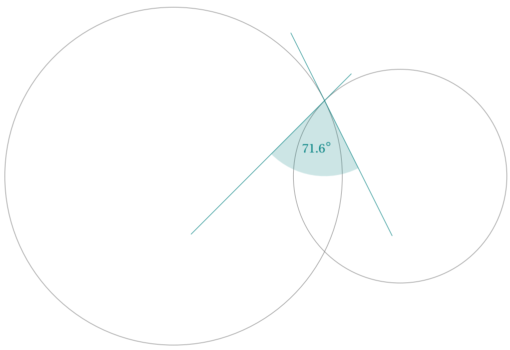
16.3.2. Détermination des trois angles d'un triangle




```
\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}
```

16.3.3. Angle entre deux cercles

Nous recherchons l'angle formé par les tangentes en un point d'intersection




```
\begin{tikzpicture}[scale=.5]
\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}
```

16.4. Angle formé par une droite et l'axe horizontal \tkzFindSlopeAngle

Beaucoup plus intéressant que le précédent. Le résultat se situe entre -180 degrés et +180 degrés.

\tkzFindSlopeAngle(⟨A,B⟩)

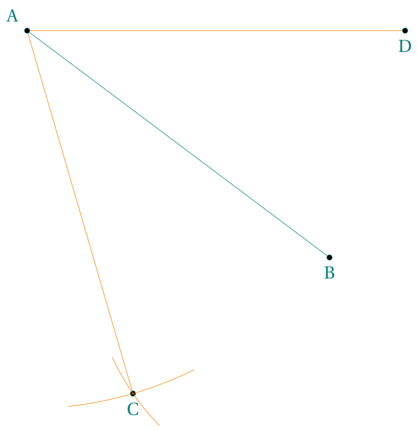
Détermine la pente de la ligne droite (AB). Le résultat est stocké dans une macro **\tkzAngleResult**.

arguments, exemples et explications	
(pt1,pt2)	**\tkzFindSlopeAngle**(A,B)

\tkzGetAngle récupére le résultat. Si l'extraction n'est pas nécessaire, vous pouvez utiliser **\tkzAngleResult**.

16.4.1. Comment utiliser \tkzFindSlopeAngle

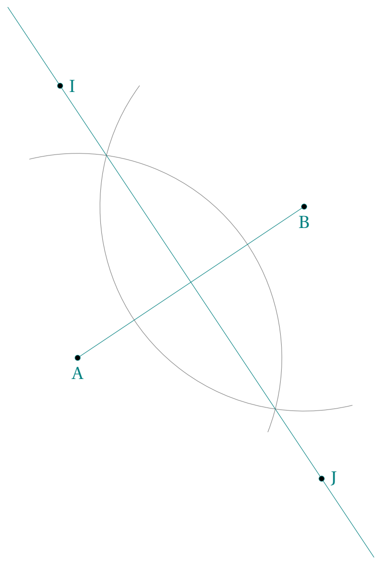
L'intérêt est que (AB) est la bissectrice de $\widehat{CAD}$, de sorte que la pente de AD est nulle. Nous retrouvons la pente de (AB) et effectuons deux rotations.




```
\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}
```

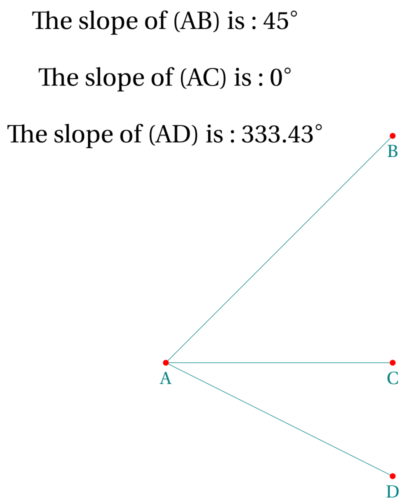
16.4.2. Utilisation de \tkzFindSlopeAngle et \tkzGetAngle

Voici une autre version de la construction d'un médiateur




```
\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}
```

16.4.3. Une autre utilisation de \tkzFindSlopeAngle




```
\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}
```

17. Définition du point aléatoire

À l'heure actuelle, il y a quatre possibilités :

1. point dans un rectangle ;
2. sur un segment ;
3. sur une ligne droite ;
4. sur un cercle.

17.1. Obtention de points aléatoires

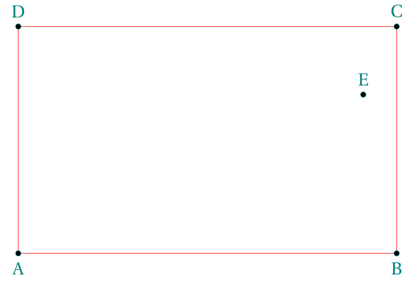
Il s'agit de la nouvelle version qui remplace `\tkzGetRandPointOn`.

`\tkzDefRandPointOn[⟨options locales⟩]`

Le résultat est un point avec une position aléatoire qui peut être nommé avec la macro `\tkzGetPoint`. Il est possible d'utiliser `tkzPointResult` s'il n'est pas nécessaire de conserver les résultats.

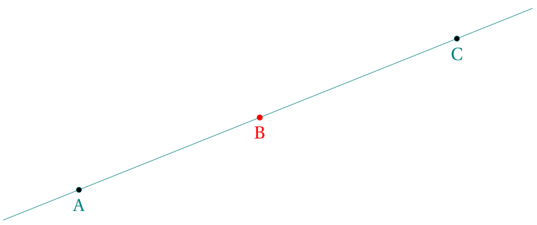
options	défaut	définition
rectangle=pt1 and pt2		[rectangle=A and B]
segment= pt1--pt2		[segment=A--B]
line=pt1--pt2		[line=A--B]
circle =center pt1 radius dim		[circle = center A radius 2]
circle through=center pt1 through pt2		[circle through= center A through B]
disk through=center pt1 through pt2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

17.1.1. Point aléatoire dans un rectangle



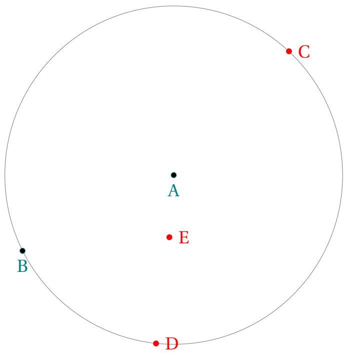

```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,5/3/C}
  \tkzDefRandPointOn[rectangle = A and C]
  \tkzGetPoint{E}
  \tkzDefRectangle(A,C)\tkzGetPoints{B}{D}
  \tkzDrawPolygon[red](A,...,D)
  \tkzDrawPoints(A,...,E)
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above](C,D,E)
\end{tikzpicture}
```

17.1.2. Point aléatoire sur un segment ou une ligne




```
\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}
```

17.1.3. Point aléatoire sur un cercle ou un disque




```
\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}
```

Quatrième partie

Dessin et remplissage

18. Tracé

tkz-euclide peut dessiner 5 types d'objets : point, ligne ou segment de ligne, cercle, arc et secteur.

18.1. Tracer un ou plusieurs points

Il existe deux possibilités : `\tkzDrawPoint` pour un seul point ou `\tkzDrawPoints` pour un ou plusieurs points.

18.1.1. Tracé de point \tkzDrawPoint

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

arguments	défault	définition
name of point	pas de défaut	Un seul nom de point est accepté

L'argument est nécessaire. Le disque prend la couleur du cercle, mais en plus clair. Il est possible de tout changer. Le point est un noeud et il est donc invariant si le dessin est modifié par une mise à l'échelle.

options	default	definition
TikZ options		toutes les options TikZ sont valables.
shape	circle	Possible **cross** ou **cross out**
size	6	6× `\pgflinewidth`
color	black	la couleur par défaut peut être modifiée

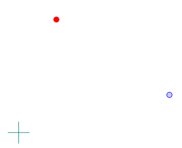
Nous pouvons créer d'autres formes telles que **cross**

Par défaut, **point style** est défini comme suit :

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

18.1.2. Exemple de tracés de points

Notez que **scale** n'affecte pas la forme des points. Ce qui est normal. La plupart du temps, on se contente d'une forme de point unique que l'on peut définir dès le départ, soit avec une macro, soit en modifiant un fichier de configuration.



```
\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}
```

Il est possible de dessiner plusieurs points à la fois mais cette macro est un peu plus lente que la précédente. De plus, il faut se contenter des mêmes options pour tous les points.

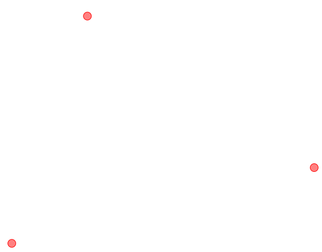
`\tkzDrawPoints[⟨options locales⟩](⟨liste⟩)`

arguments	default	definition
points list	pas de défaut	example `\tkzDrawPoints(A,B,C)`

options	par défaut	définition
shape	circle	Possible **cross** ou **cross out**
size	6	6× `\pgflinewidth`
color	black	la couleur par défaut peut être modifiée

Attention au "s" final, un oubli entraîne des erreurs en cascade si vous essayez de dessiner plusieurs points. Les options sont les mêmes que pour la macro précédente.

18.1.3. 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}
```

19. Tracer les lignes

Les macros suivantes sont simplement utilisées pour dessiner, nommer des lignes.

19.1. Tracer une ligne droite

Pour tracer une ligne droite normale, il suffit de donner quelques points. Vous pouvez utiliser l'option **add** pour prolonger la ligne (Cette option est due à `Mark Wibrow`, voir le code ci-dessous).
Le style d'une ligne est par défaut :

```
\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}}}
```

Vous pouvez modifier ce style avec `\tkzSetUpLine` Veuillez consulter la section 37.1 pour plus d'informations.

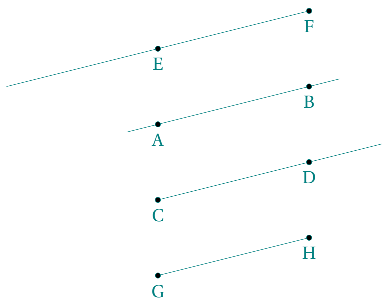
```
\tkzDrawLine[⟨options locales⟩](⟨pt1,pt2⟩)
```

Les arguments sont une liste de deux points ou de trois points. Il serait possible, comme pour une demi-ligne, de créer un style avec `\add`.

options	défaut	définition
TikZ options		toutes les options de TikZ sont valables.
add	0.2 and 0.2	add = kl and kr, …
…	…	permet d'étendre le segment à gauche et à droite.

`add` définit la longueur de la ligne passant par les points pt1 et pt2. Les deux nombres sont des pourcentages. Les styles de TikZ sont accessibles pour les tracés.

19.1.1. Exemples avec add



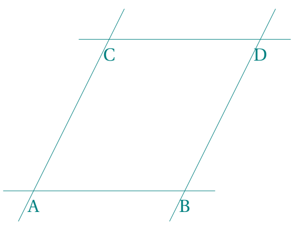

```
\begin{tikzpicture}
 \tkzInit[xmin=-2,xmax=3,ymin=-2.25,ymax=2.25]
 \tkzClip[space=.25]
 \tkzDefPoint(0,0){A} \tkzDefPoint(2,0.5){B}
 \tkzDefPoint(0,-1){C}\tkzDefPoint(2,-0.5){D}
 \tkzDefPoint(0,1){E} \tkzDefPoint(2,1.5){F}
 \tkzDefPoint(0,-2){G} \tkzDefPoint(2,-1.5){H}
 \tkzDrawLine(A,B)    \tkzDrawLine[add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

Il est possible de dessiner plusieurs lignes, mais avec les mêmes options.

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

Les arguments sont une liste de paires de points séparés par des espaces. Les styles de TikZ sont disponibles pour les tirages.

19.1.2. Exemple avec \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}
```

20. Dessiner un segment

Il existe bien sûr une macro permettant de dessiner simplement un segment.

20.1. Draw a segment \tkzDrawSegment

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

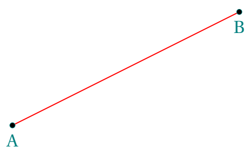
The arguments are a list of two points. The styles of Ti*k*Z are available for the drawings.

argument	exemple	définition
(pt1,pt2)	(A,B)	dessiner le segment [A,B]

options	exemple	définition
TikZ options		all TikZ options are valid.
dim	pas de défaut	dim = {label,dim,option}, …
…	…	allows you to add dimensions to a figure.

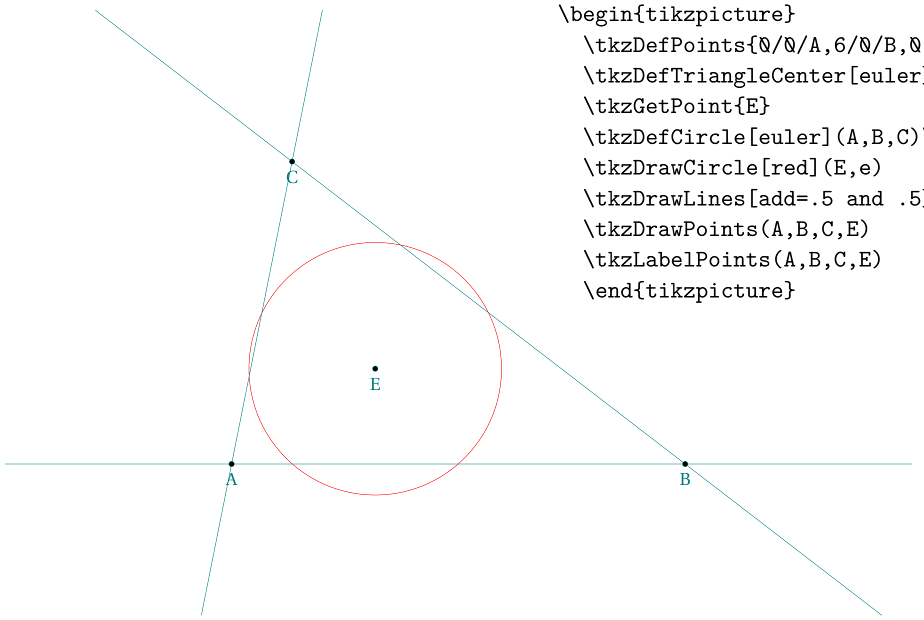
Ceci est bien sûr équivalent à `\draw (A)--(B);`. Vous pouvez également utiliser l'option `add`.

20.1.1. Exemple avec références ponctuelles




```
\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}
```

20.1.2. Exemple d'extension d'un segment avec option add

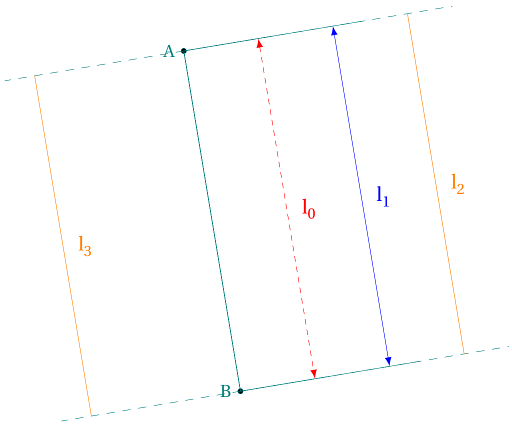



```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,6/0/B,0.8/4/C}
  \tkzDefTriangleCenter[euler](A,B,C)
  \tkzGetPoint{E}
  \tkzDefCircle[euler](A,B,C)\tkzGetPoints{E}{e}
  \tkzDrawCircle[red](E,e)
  \tkzDrawLines[add=.5 and .5](A,B A,C B,C)
  \tkzDrawPoints(A,B,C,E)
  \tkzLabelPoints(A,B,C,E)
  \end{tikzpicture}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

20.1.3. Ajouter des dimensions avec l'option dim nouveau code de Muzimuzhi Z

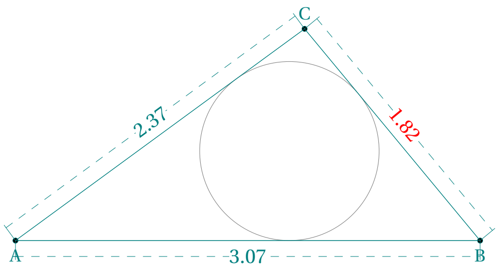
Ce code provient d'une réponse à cette question sur tex.stackexchange.com (change-color-and-style-of-dimension-lines-in-tkz-euclide). Le code de `dim` est basé sur les options de TikZ, vous devez ajouter les unités. Vous pouvez maintenant utiliser deux styles : `dim style` et `dim fence style`. Il y a plusieurs façons de les utiliser. Je vous laisse regarder les exemples pour voir ce que vous pouvez faire avec ces styles.

```
\tikzset{dim style/.append style={dashed}}
% append si vous souhaitez conserver le style précédent.
ou bien
\begin{scope}[ dim style/.append style={orange},
    dim fence style/.style={dashed}]
```



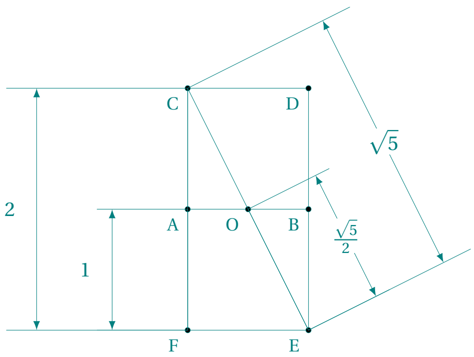

```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/3/A, 1/-3/B}
  \tkzDrawPoints(A,B)
  \tkzDrawSegment[dim={\(l_0\),1cm,right=2mm},
    dim style/.append style={red,
    dash pattern={on 2pt off 2pt}}](A,B)
  \tkzDrawSegment[dim={\(l_1\),2cm,right=2mm},
    dim style/.append style={blue}](A,B)
  \begin{scope}[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

20.1.4. Ajout de dimensions avec l'option **dim** part I




```
\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}
```

20.1.5. Ajout de dimensions avec option dim part II



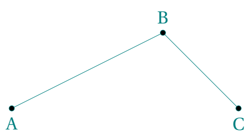

```
\begin{tikzpicture}[scale=.4]
  \tkzDefPoints{0/0/O,-2/0/A,2/0/B,
               -2/4/C,2/4/D,2/-4/E,-2/-4/F}
  \tkzDrawPolygon(C,...,F)
  \tkzDrawSegments(A,B)
  \tkzDrawPoints(A,...,F,O)
  \tkzLabelPoints[below left](A,...,F,O)
  \tkzDrawSegment[dim={ $\sqrt{5}$,2cm,}](C,E)
  \tkzDrawSegment[dim={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

20.2. Dessiner des segments \tkzDrawSegments

Si les options sont les mêmes, nous pouvons tracer plusieurs segments avec la même macro.

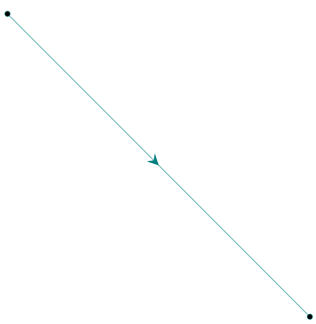
\tkzDrawSegments[⟨options locales⟩](⟨pt1,pt2 pt3,pt4 ...⟩)

Les arguments sont une liste de couples à deux points. Les styles de Ti*k*Z sont disponibles pour les tracés.




```
\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}
```

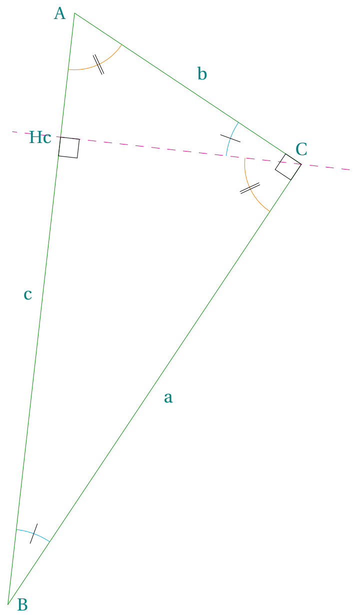
20.2.1. Placer une flèche sur le segment



```
\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}
```

20.3. Dessiner un segment de droite dans un triangle

20.3.1. Comment dessiner une hauteur




```
\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}
```

20.4. Tracer un polygone

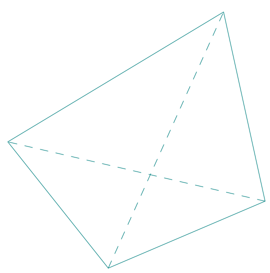
`\tkzDrawPolygon[⟨options locales⟩](⟨points list⟩)`

Il suffit de donner une liste de points et la macro trace le polygone en utilisant les options Ti*k*Z présentes. Vous pouvez remplacer (A, B, C, D, E) par (A, ..., E) et (P_1, P_2, P_3, P_4, P_5) par (P_1, P..., P_5).

arguments, exemples et explications		
`(⟨pt1,pt2,pt3,...⟩)`	`\tkzDrawPolygon[gray,dashed](A,B,C)`	Drawing a triangle

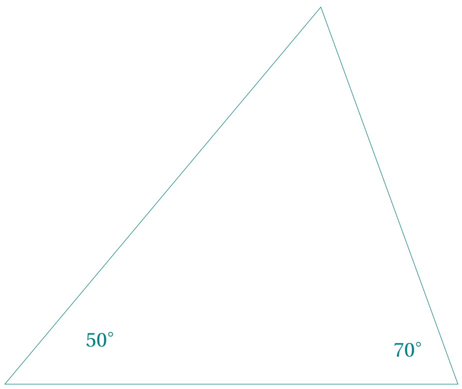
options	défaut	exemple
Options TikZ	...	`\tkzDrawPolygon[red,line width=2pt](A,B,C)`

20.4.1. \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}
```

20.4.2. 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}
```

20.4.3. Style de ligne

```
\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}
```

20.5. Tracer une chaîne polygonale

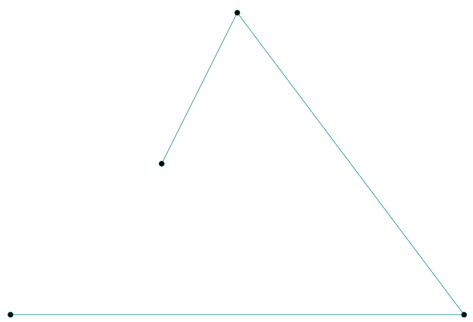
\tkzDrawPolySeg[⟨options locales⟩](⟨liste de points⟩)

Il suffit de donner une liste de points et la macro trace la chaîne polygonale en utilisant les options Ti*k*Z présentes.

arguments, exemples et explications		
(⟨pt1,pt2,pt3,...⟩)	`\tkzDrawPolySeg[gray,dashed](A,B,C)`	Drawing a triangle

options	défaut	exemple
Options TikZ	...	`\tkzDrawPolySeg[red,line width=2pt](A,B,C)`

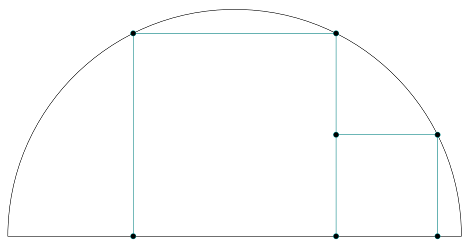
20.5.1. Chaîne polygonale



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

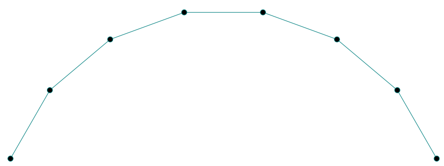
20.5.2. Il s'agit d'inscrire deux carrés dans un demi-cercle.

Un aspect Sangaku ! Il s'agit de prouver que l'on peut inscrire dans un demi-disque, deux carrés, et de déterminer la longueur de leurs côtés respectifs en fonction du rayon.



```
\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}
```

20.5.3. Chaîne polygonale : notation de l'indice



```
\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}
```

21. Tracer un cercle avec \tkzDrawCircle

21.1. Tracer un cercle

`\tkzDrawCircle[⟨options locales⟩](⟨A,B⟩)`

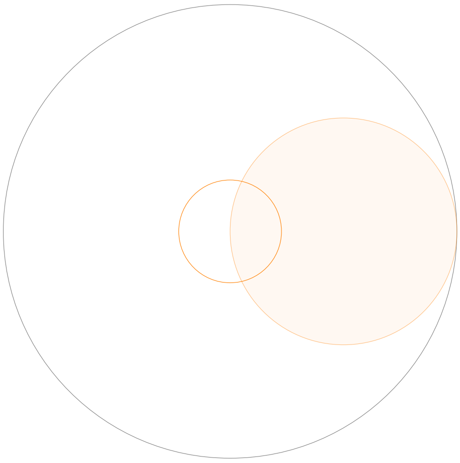
Attention, il suffit de deux points pour définir un rayon. Une option supplémentaire `R` est disponible pour donner une mesure directement.

arguments, exemples et explications		
(⟨pt1,pt2⟩)	(⟨A,B⟩)	A center through B

Bien sûr, il faut ajouter tous les styles de Ti*k*Z pour les tracés...

21.1.1. Cercles et styles, dessiner un cercle et colorier le disque

Nous verrons qu'il est possible de colorier un disque tout en traçant le cercle.



```
\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}
```

21.2. Tracer des cercles

`\tkzDrawCircles[⟨options locales⟩](⟨A,B C,D …⟩)`

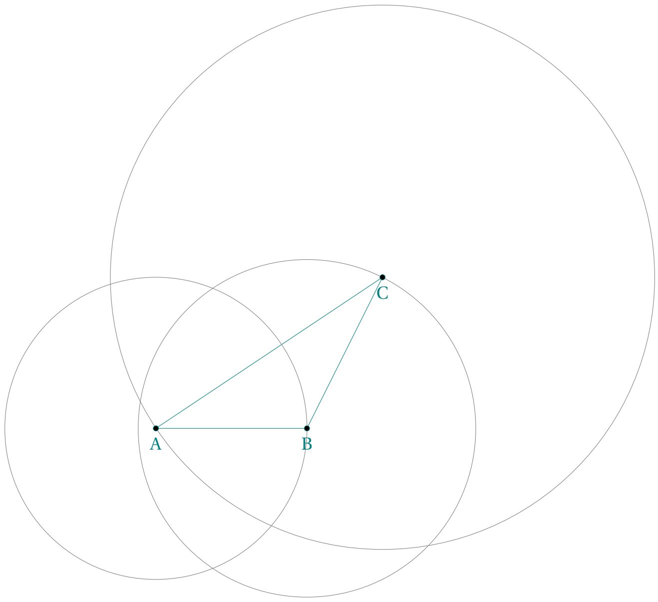
Attention, les arguments sont des listes de deux points. Les cercles qui peuvent être dessinés sont les mêmes que dans la macro précédente. Une option supplémentaire **R** est disponible pour donner une mesure directement.

arguments, exemples et explications		
(⟨pt1,pt2 pt3,pt4 ...⟩)	(⟨A,B C,D⟩)	Liste de deux points

options	défaut	définition
through	through	cercle avec deux points définissant un rayon

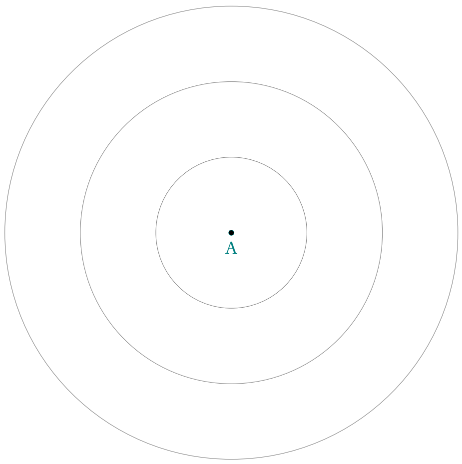
Il n'est pas nécessaire d'utiliser l'option par défaut **through**. Bien sûr, il faut ajouter tous les styles de Ti*k*Z pour les tracés...

21.2.1. Cercles définis par un triangle.



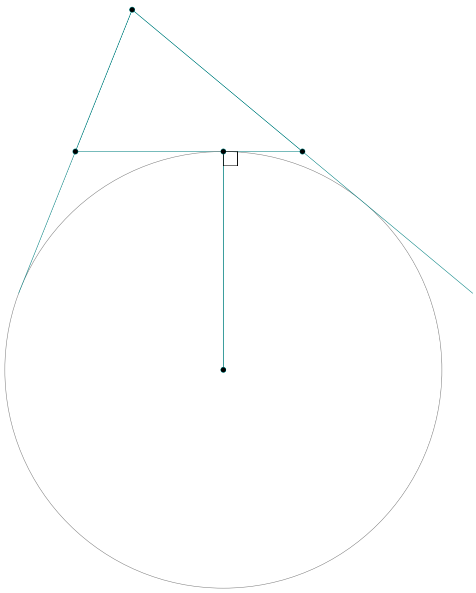

```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,2/0/B,3/2/C}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawCircles(A,B B,C C,A)
  \tkzDrawPoints(A,B,C)
  \tkzLabelPoints(A,B,C)
\end{tikzpicture}
```

21.2.2. Concentric circles.




```
\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}
```

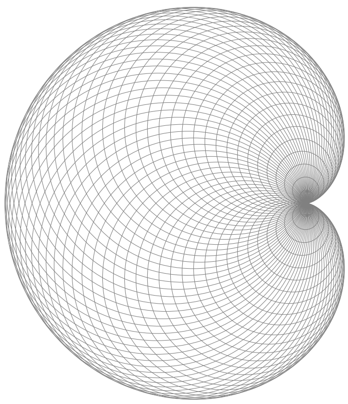
21.2.3. Cercles exinscrits.



```
\begin{tikzpicture}[scale=.75]
\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}
```

21.2.4. Cardioïde

Basé sur une idée de O. Reboux réalisée avec pst-eucl (module Pstricks) de D. Rodriguez.
Son nom vient du grec *kardia (cœur)*, en référence à sa forme, et lui a été donné par Johan Castillon (Wikipedia).



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

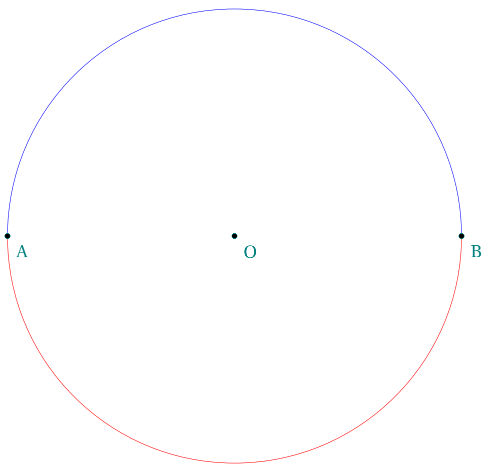
21.3. Tracer un demi-cercle

`\tkzDrawSemiCircle[⟨options locales⟩](⟨O,A⟩)`

arguments, exemples et explications		
(⟨pt1,pt2⟩)	(⟨O,A⟩)	OA= radius

O centre A extrémité du demi-cercle

21.3.1. Use of \tkzDrawSemiCircle



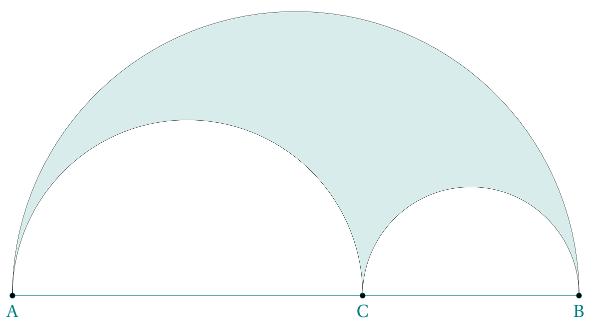

```
\begin{tikzpicture}
   \tkzDefPoint(0,0){A} \tkzDefPoint(6,0){B}
   \tkzDefMidPoint(A,B)  \tkzGetPoint{O}
   \tkzDrawSemiCircle[blue](O,B)
   \tkzDrawSemiCircle[red](O,A)
   \tkzDrawPoints(O,A,B)
   \tkzLabelPoints[below right](O,A,B)
 \end{tikzpicture}
```

21.4. Tracer des demi-cercles

`\tkzDrawSemiCircles[⟨options locales⟩](⟨A,B C,D …⟩)`

arguments, exemples et explications		
(⟨pt1,pt2 pt3,pt4 ...⟩)	(⟨A,B C,D⟩)	Liste de deux points

21.4.1. Utilisation de \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}
```

22. Tracer une ellipse avec \tkzDrawEllipse

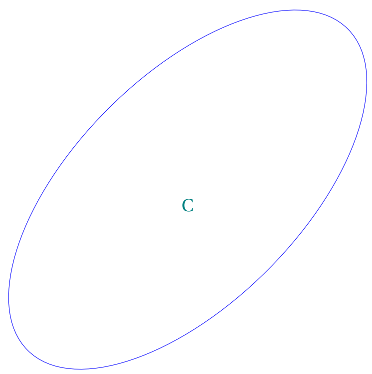
22.1. Draw an ellipse

`\tkzDrawEllipse[⟨options locales⟩](⟨C,a,b,An⟩)`

arguments, exemples et explications		
(⟨C,a,b,An⟩)	(⟨C,4,2,45⟩)	C centre ; 4 and 2 longueur des demi-axes 45 slope of main axis

Bien sûr, il faut ajouter tous les styles de Ti*k*Z pour les tracés...

22.1.1. Exemple de tracé d'une ellipse




```
\begin{tikzpicture}[scale=.75]
   \tkzDefPoint(0,4){C}
   \tkzDrawEllipse[blue](C,4,2,45)
   \tkzLabelPoints(C)
\end{tikzpicture}
```

23. Tracer des arcs

23.1. Macro: \tkzDrawArc

`\tkzDrawArc[⟨options locales⟩](⟨O,...⟩)(⟨...⟩)`

Cette macro trace l'arc de centre O. En fonction des options, les arguments diffèrent. Il s'agit de déterminer un point de départ et un point d'arrivée. Soit on donne le point de départ, ce qui est le plus simple, soit on donne le rayon de l'arc. Dans ce dernier cas, il est nécessaire d'avoir deux angles. On peut soit donner directement les angles, soit donner les nœuds associés au centre pour les déterminer. Les angles sont exprimés en degrés.

options	défaut	définition
`towards`	`towards`	`O est le centre de l'arc de A vers (OB)`
`rotate`	`towards`	`L'arc part de A et l'angle détermine sa longueur`
`R`	`towards`	`Nous donnons le rayon et deux angles`
`R with nodes`	`towards`	`Nous donnons le rayon et deux points`
`angles`	`towards`	`Nous donnons le rayon et deux points`
`delta`	`0`	`Angle ajouté de chaque côté`
`reverse`	`false`	`Inversion de la trajectoire de l'arc, ex. inverser une flèche`

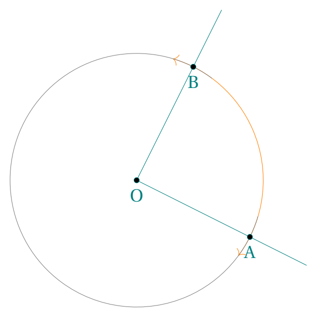
Bien sûr, il faut ajouter tous les styles de Ti*k*Z pour les tracés...

options	arguments	exemple
`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)`

Voici quelques exemples :

23.1.1. Option `towards`

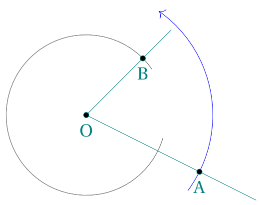
There is no need for **`towards`**. In this first example, the arc goes from A to B. The arc from B to A is different. The salient is obtained by going in the direct direction of the trigonometric circle.




```
\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}
```

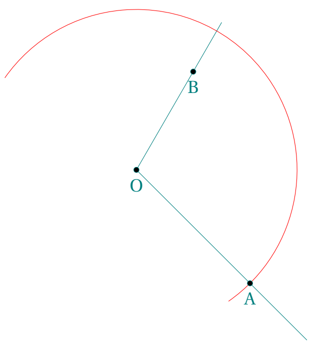
23.1.2. Option `towards`

Dans celui-ci, l'arc part de A mais s'arrête à droite (OB).



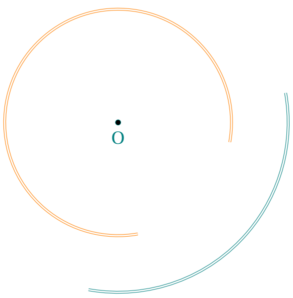

```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-1){A}
  \tkzDefPoint(1,1){B}
  \tkzDrawArc[color=blue,->](O,A)(B)
  \tkzDrawArc[color=gray](O,B)(A)
  \tkzDrawArc(O,B)(A)
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

23.1.3. Option rotate



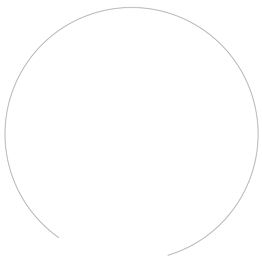

```
\begin{tikzpicture}[scale=0.75]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(2,-2){A}
  \tkzDefPoint(60:2){B}
  \tkzDrawLines[add = 0 and .5](O,A O,B)
  \tkzDrawArc[rotate,color=red](O,A)(180)
  \tkzDrawPoints(O,A,B)
  \tkzLabelPoints[below](O,A,B)
\end{tikzpicture}
```

23.1.4. 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}
```

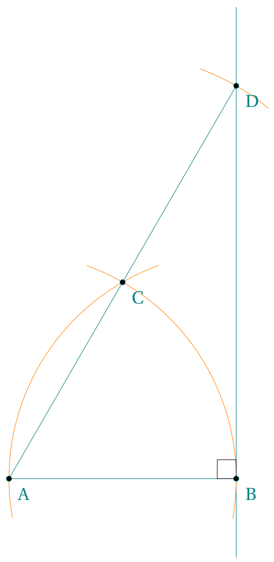
23.1.5. Option R with nodes



```
\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}
```

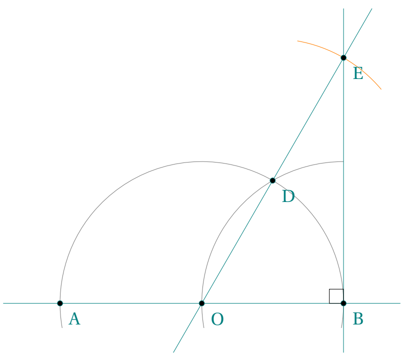
23.1.6. Option delta

Cette option permet, un peu comme `\tkzCompass`, de placer un arc et un débordement de chaque côté. delta est une mesure en degrés.



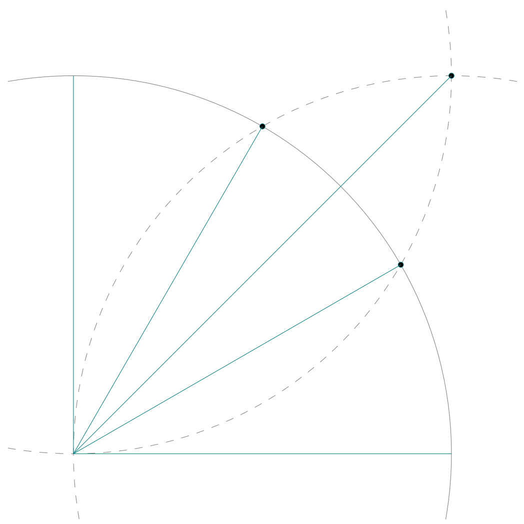

```
\begin{tikzpicture}
 \tkzDefPoint(0,0){A}
 \tkzDefPoint(3,0){B}
 \tkzDefPointBy[rotation= center A angle 60](B)
 \tkzGetPoint{C}
 \begin{scope}% style only local
   \tkzDefPointBy[symmetry= center C](A)
   \tkzGetPoint{D}
   \tkzDrawSegments(A,B A,D)
   \tkzDrawLine(B,D)
   \tkzSetUpCompass[color=orange]
   \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}
```

23.1.7. Option **angles**: exemple 1



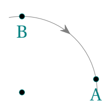

```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(5,0){B}
  \tkzDefPoint(2.5,0){O}
  \tkzDefPointBy[rotation=center O angle 60](B)
  \tkzGetPoint{D}
  \tkzDefPointBy[symmetry=center D](O)
  \tkzGetPoint{E}
  \begin{scope}
    \tkzDrawArc[angles](O,B)(0,180)
    \tkzDrawArc[angles,](B,O)(100,180)
    \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}
```

23.1.8. Option angles: exemple 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}
```

23.1.9. Option reverse: inversion de la flèche




```
\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}
```

24. Tracer un ou plusieurs secteurs

24.1. \tkzDrawSector

Attention les arguments varient en fonction des options.

`\tkzDrawSector[⟨options locales⟩](⟨O,...⟩)(⟨...⟩)`

options	défaut	définition
towards	towards	O est le centre et l'arc de A vers (OB)
rotate	towards	l'arc part de A et l'angle détermine sa longueur
R	towards	Nous donnons le rayon et deux angles
R with nodes	towards	Nous donnons le rayon et deux points

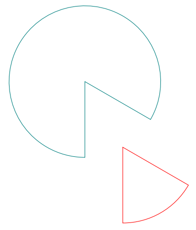
Il faut ajouter, bien sûr, tous les styles de TikZ pour les tracés...

options	arguments	exemple
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)

Voici quelques exemples :

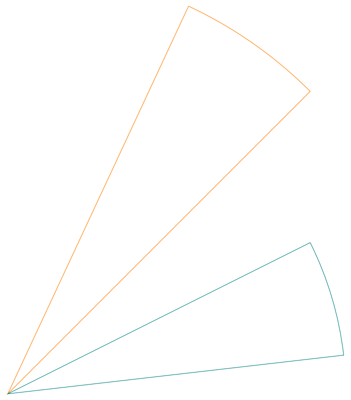
24.1.1. \tkzDrawSector et towards

Il n'est pas nécessaire de mettre `towards`. Vous pouvez utiliser `fill` en option.



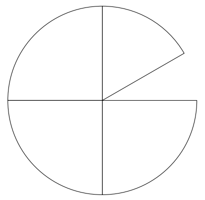
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(-30:1){A}
  \tkzDefPointBy[rotation = center O angle -60](A)
  \tkzDrawSector[teal](O,A)(tkzPointResult)
 \begin{scope}[shift={(-60:1)}]
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(-30:1){A}
  \tkzDefPointBy[rotation = center O angle -60](A)
  \tkzDrawSector[red](O,tkzPointResult)(A)
  \end{scope}
\end{tikzpicture}
```

24.1.2. \tkzDrawSector et 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}
```

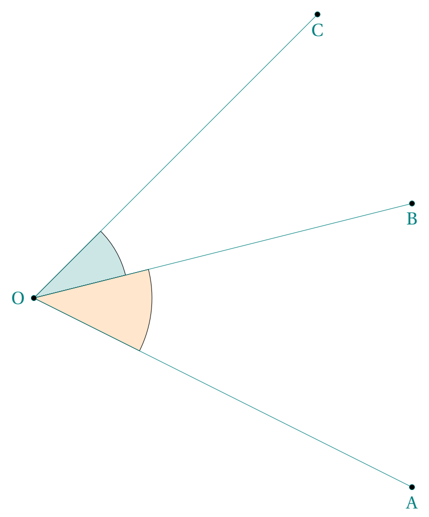
24.1.3. \tkzDrawSector et R



```
\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}
```

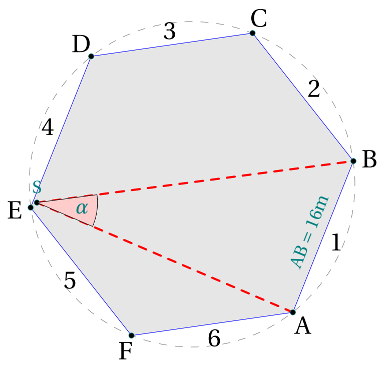
24.1.4. \tkzDrawSector et R with nodes

Dans cet exemple, j'utilise l'option `fill` mais `\tkzFillSector` est possible.




```
\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}
```

24.1.5. \tkzDrawSector et 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}
```

24.2. Coloration d'un disque

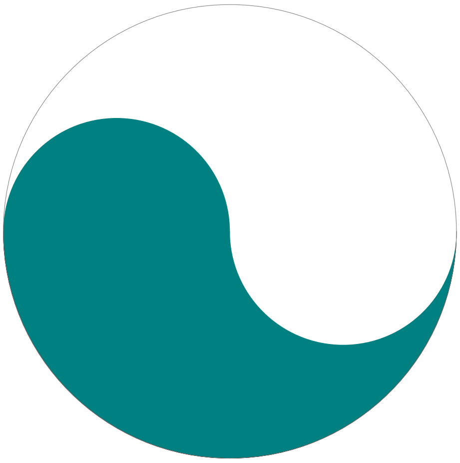
Cela était possible avec la macro `\tkzDrawCircle`, mais le traçage du disque était obligatoire, ce qui n'est plus le cas.

`\tkzFillCircle[⟨options locales⟩](⟨A,B⟩)`

options	défaut	définition
radius	radius	two points define a radius
R	radius	a point and the measurement of a radius

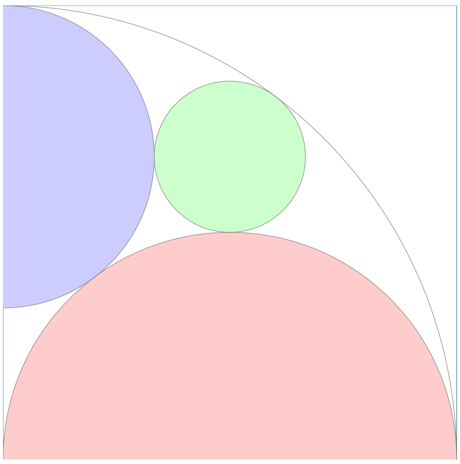
Il n'est pas nécessaire de mettre **radius** car c'est l'option par défaut. Bien entendu, vous devez ajouter tous les styles de Ti*k*Z pour les tracés.

24.2.1. Yin and Yang



```
\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}
```

24.2.2. D'un sangaku

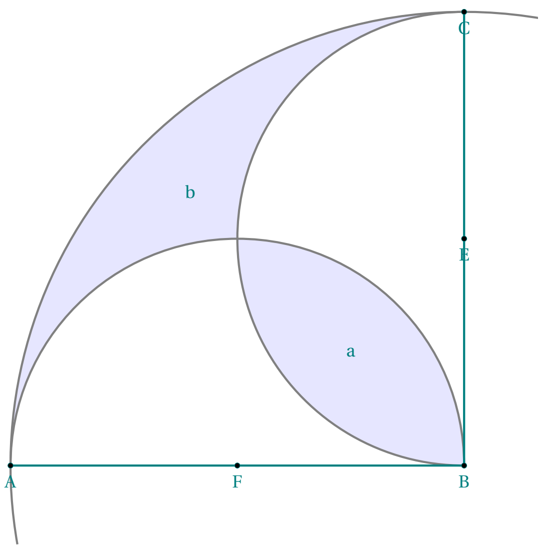


```
\begin{tikzpicture}
   \tkzDefPoint(0,0){B}  \tkzDefPoint(6,0){C}%
   \tkzDefSquare(B,C)    \tkzGetPoints{D}{A}
   \tkzClipPolygon(B,C,D,A)
   \tkzDefMidPoint(A,D)  \tkzGetPoint{F}
   \tkzDefMidPoint(B,C)  \tkzGetPoint{E}
   \tkzDefMidPoint(B,D)  \tkzGetPoint{Q}
   \tkzDefLine[tangent from = B](F,A)
   \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}
```

24.2.3. Découpage et remplissage part I

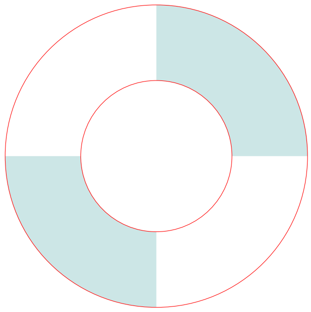

```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/0/B,2/2/O,3/4/X,4/1/Y,1/0/Z,
              0/3/W,3/0/R,4/3/S,1/4/T,0/1/U}
\tkzDefSquare(A,B)\tkzGetPoints{C}{D}
\tkzDefPointWith[colinear normed=at X,K=1](O,X)
 \tkzGetPoint{F}
\begin{scope}
  \tkzFillCircle[fill=teal!20](O,F)
  \tkzFillPolygon[white](A,...,D)
  \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}
```

24.2.4. Découpage et remplissage part II




```
\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}
```

24.2.5. Découpage et remplissage part III



```
\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}
```

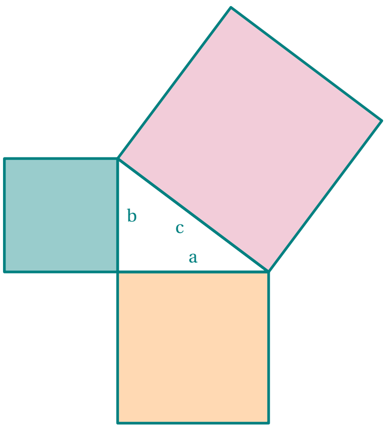
24.3. Coloration d'un polygone

\tkzFillPolygon[⟨options locales⟩](⟨liste de points⟩)

You can color by drawing the polygon, but in this case you color the inside of the polygon without drawing it.

arguments, exemples et explications	
(⟨pt1,pt2,...⟩)	(⟨A,B,...⟩)

24.3.1. \tkzFillPolygon




```
\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}
```

24.4. \tkzFillSector

Attention les arguments varient en fonction des options.

`\tkzFillSector[⟨options locales⟩](⟨O,...⟩)(⟨...⟩)`

options	défaut	définition
towards	towards	O est le centre et l'arc de A vers (OB)
rotate	towards	l'arc part de A et l'angle détermine sa longueur
R	towards	Nous donnons le rayon et deux angles
R with nodes	towards	Nous donnons le rayon et deux points

Bien sûr, il faut ajouter tous les styles de TikZ pour les tracés...

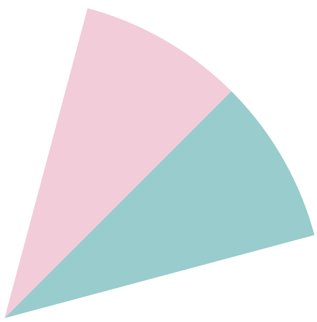
options	arguments	exemple
towards	(⟨pt,pt⟩)(⟨pt⟩)	`\tkzFillSector(O,A)(B)`
rotate	(⟨pt,pt⟩)(⟨an⟩)	`\tkzFillSector[rotate,color=red](O,A)(90)`
R	(⟨pt,r⟩)(⟨an,an⟩)	`\tkzFillSector[R,color=blue](O,2)(30,90)`
R with nodes	(⟨pt,r⟩)(⟨pt,pt⟩)	`\tkzFillSector[R with nodes](O,2)(A,B)`

24.4.1. \tkzFillSector et towards

Il est inutile de mettre **towards** et vous remarquerez que les contours ne sont pas dessinés, seule la surface est colorée.

```
\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}
```

24.4.2. \tkzFillSector et 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}
```

24.5. Colorer un angle : \tkzFillAngle

L'opération la plus simple

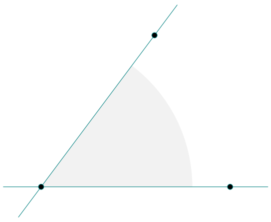
`\tkzFillAngle[⟨options locales⟩](⟨A,O,B⟩)`

O est le sommet de l'angle. OA et OB sont les côtés. Attention l'angle est déterminé par l'ordre des points.

options	défaut	définition
size	1	cette option détermine le rayon du secteur angulaire coloré.

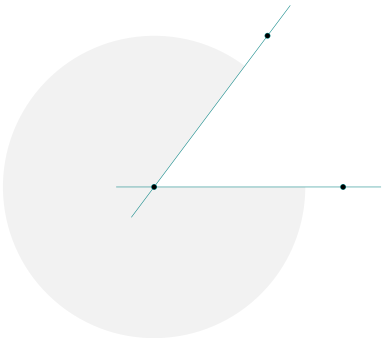
Bien sûr, il faut ajouter tous les styles de Ti*k*Z, comme l'utilisation du remplissage et de l'ombrage...

24.5.1. Exemple avec size

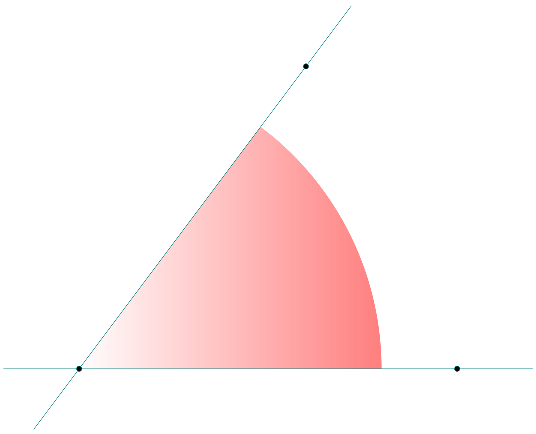


```
\begin{tikzpicture}
   \tkzInit
   \tkzDefPoints{0/0/O,2.5/0/A,1.5/2/B}
   \tkzFillAngle[size=2, fill=gray!10](A,O,B)
   \tkzDrawLines(O,A O,B)
   \tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

24.5.2. Modifier l'ordre des éléments



```
\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}
```

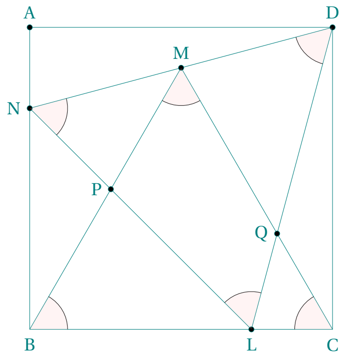


```
\begin{tikzpicture}
   \tkzInit
   \tkzDefPoints{0/0/O,5/0/A,3/4/B}
   % Don't forget {} to get, () to use
   \tkzFillAngle[size=4,left color=white,
                 right color=red!50](A,O,B)
   \tkzDrawLines(O,A O,B)
   \tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

`\tkzFillAngles[⟨options locales⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.`

Avec des options communes, il existe une macro pour des angles multiples.

24.5.3. Multiples angles




```
\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}
```

25. Contrôle de la Bounding Box

Traduction :
Depuis le `Manuel Pgf` : ”Lorsque vous ajoutez l’option clip, le chemin actuel est utilisé pour la découpe des tracés ultérieurs. Le `clipping` n’agrandit jamais la zone de découpe. Ainsi, lorsque vous `clipper` par rapport à un certain chemin, puis `clipper` à nouveau par rapport à un autre chemin, vous `clipper` par rapport à l’intersection des deux. La seule façon d’agrandir le chemin de découpe est de terminer l’environnement pgfscope dans lequel la découpe a été effectuée. À la fin d’un pgfscope, le chemin de découpe qui était en vigueur au début de l’environnement est réinstallé.”
Tout d’abord, vous n’avez pas à vous occuper avec Ti*k*Z de la taille de la `bounding box`. Les premières versions de `tkz-euclide` ne contrôlaient pas la taille de la `bounding box`, maintenant avec `tkz-euclide` 4 la taille de la `bounding box` est limitée.
La `bounding box` initiale après l’utilisation de la macro `\tkzInit` est définie par le rectangle basé sur les points $(0,0)$ et $(10,10)$. La macro `\tkzInit` permet de modifier cette `bounding box` initiale en utilisant les arguments (**`xmin`**, **`xmax`**, **`ymin`** et **`ymax`**). Bien sûr, toute trace externe modifie la bounding box. Ti*k*Z maintient cette “bounding box”. Il est possible d’influencer ce comportement, soit directement avec des commandes, soit avec des options de Ti*k*Z telles qu’une commande comme `\useasboundingbox` ou l’option **`use as bounding box`**. Une conséquence possible est de réserver une boîte pour une figure mais la figure peut déborder de la boîte et se répandre sur le texte principal. La commande suivante `\pgfresetboundingbox` efface une `bounding box` et en établit une nouvelle.

25.1. Utilité de \tkzInit

Cependant, il est parfois nécessaire de contrôler la taille de ce qui sera affiché. Pour ce faire, vous devez avoir préparé la `bounding box` dans laquelle vous allez travailler, c’est le rôle de la macro `\tkzInit`. Pour certains dessins, il est intéressant de fixer les valeurs extrêmes (xmin, xmax, ymin et ymax) et de `délimiter` le rectangle de définition afin de contrôler au mieux la taille de la figure.
Les deux macros qui sont utiles pour contrôler la `bounding box` :
— `\tkzInit`
— `\tkzClip`

A cela, j’ai ajouté des macros directement liées à la `bounding box`. Vous pouvez maintenant la visualiser, la sauvegarder, la restaurer (voir l’onglet section `bounding box`).

25.2. \tkzInit

\tkzInit[⟨options locales⟩]

options	default	definition
xmin	0	valeur minimale de l'abscisse en cm
xmax	10	valeur maximale de l'abscisse en cm
xstep	1	différence entre deux graduations en x
ymin	0	valeur minimale de l'axe des y en cm
ymax	10	valeur maximale de l'axe des y en cm
ystep	1	différence entre deux graduations en y

Le rôle de `\tkzInit` est de définir un système de coordonnées orthogonal et une partie rectangulaire du plan dans laquelle vous placerez vos dessins en utilisant des coordonnées cartésiennes. Cette macro vous permet de définir votre environnement de travail comme avec une calculatrice. Avec `tkz-euclide` 4, `\xstep` et `\ystep` sont toujours égaux à 1. Logiquement, il n'est plus nécessaire d'utiliser `\tkzInit`, sauf pour une action comme `Clipping Out`.

25.3. \tkzClip

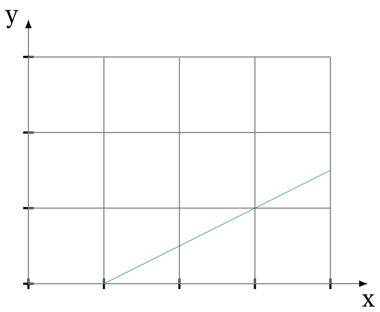
\tkzClip[⟨options locales⟩]

Le rôle de cette macro est de rendre invisible ce qui se trouve en dehors du rectangle défini par (xmin; ymin) et (xmax; ymax).

options	défaut	définition
space	1	valeur ajoutée à droite, à gauche, en bas et en haut de l'arrière-plan

Le rôle de l'option `space` est d'agrandir la partie visible du dessin. Cette partie devient le rectangle défini par (xmin-espace; ymin-espace) et (xmax+espace; ymax+espace). `space` peut être négatif! L'unité est le cm et ne doit pas être spécifiée.

Le rôle de cette macro est de "découper" le rectangle initial afin que seuls les chemins contenus dans ce rectangle soient dessinés.



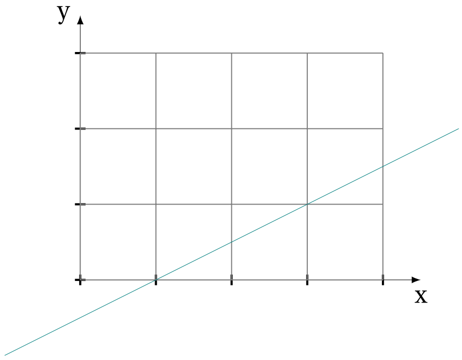

```
\begin{tikzpicture}
 \tkzInit[xmax=4, ymax=3]
 \tkzDefPoints{-1/-1/A,5/2/B}
 \tkzDrawX \tkzDrawY
 \tkzGrid
 \tkzClip
 \tkzDrawSegment(A,B)
\end{tikzpicture}
```

Il est possible d'ajouter un peu d'espace

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

25.4. \tkzClip et l'option space

Cette option vous permet d'ajouter de l'espace autour du rectangle "découpé".




```
\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}
```

Les dimensions du rectangle coupé sont les suivantes **xmin-1**, **ymin-1**, **xmax+1** et **ymax+1**.

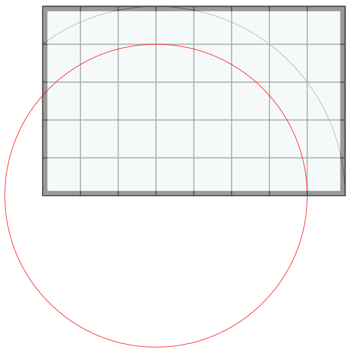
25.5. tkzShowBB

La macro la plus simple.

`\tkzShowBB[⟨options locales⟩]`

Cette macro affiche la `bounding box`. Un cadre rectangulaire entoure la `bounding box`. Cette macro accepte les options de TikZ.

25.5.1. Exemple avec \tkzShowBB



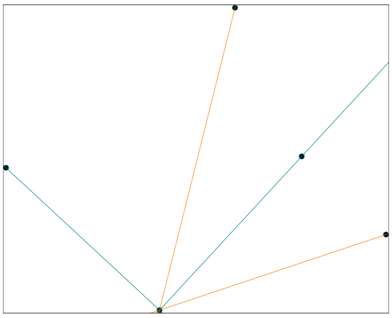
```
\begin{tikzpicture}[scale=.5]
  \tkzInit[ymax=5,xmax=8]
  \tkzGrid
  \tkzDefPoint(3,0){A}
   \begin{scope}
    \tkzClipBB
    \tkzDefCircle[R](A,5) \tkzGetPoint{a}
    \tkzDrawCircle(A,a)
     \tkzShowBB[line width = 4pt,fill=teal!10,%
               opacity=.4]
   \end{scope}
\tkzDefCircle[R](A,4) \tkzGetPoint{b}
\tkzDrawCircle[red](A,b)
\end{tikzpicture}
```

25.6. tkzClipBB

`\tkzClipBB`

L'idée est de limiter les constructions futures au périmètre actuel.

25.6.1. Exemple avec \tkzClipBB et les bissectrices



```
\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}
```

26. Découpage de différents objets

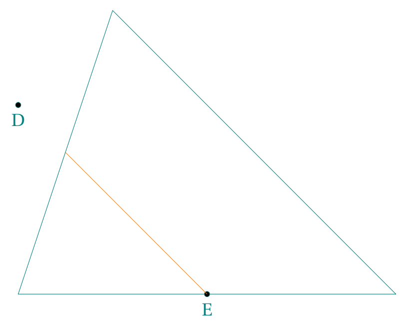
26.1. Découpage d' un polygone

\tkzClipPolygon[⟨options locales⟩](⟨liste de points⟩)

Cette macro permet de contenir les différentes parcelles dans le polygone désigné.

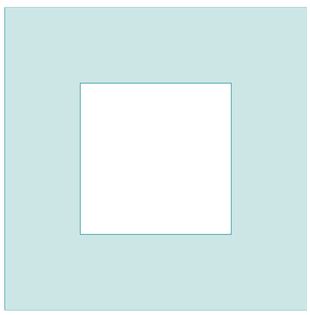
arguments, exemples et explications		
(⟨pt1,pt2,pt3,...⟩)	(⟨A,B,C⟩)	
options	défaut	définition
out		permet de découper l'extérieur de l'objet

26.1.1. \tkzClipPolygon



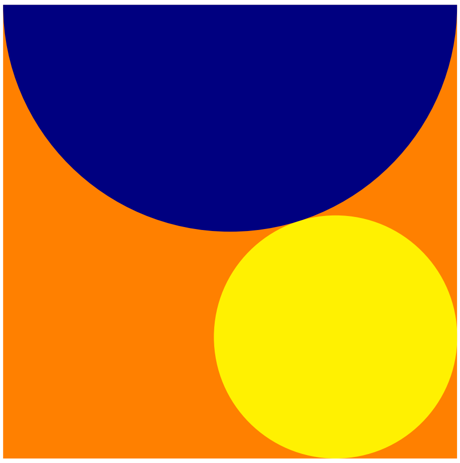

```
  \begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(4,0){B}
  \tkzDefPoint(1,3){C}
  \tkzDrawPolygon(A,B,C)
  \tkzDefPoint(0,2){D}
  \tkzDefPoint(2,0){E}
  \tkzDrawPoints(D,E)
  \tkzLabelPoints(D,E)
  \tkzClipPolygon(A,B,C)
  \tkzDrawLine[new](D,E)
\end{tikzpicture}
```

26.1.2. \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}
```

26.1.3. Exemple : utilisation de "Clip" pour Sangaku dans un carré



```
\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}
```

26.2. Découpage d'un disque

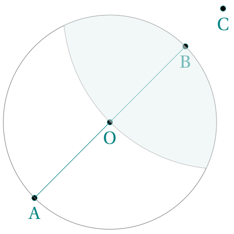
`\tkzClipCircle[⟨options locales⟩](⟨A,B⟩)`

arguments, exemples et explications		
(⟨A,B⟩)	(⟨A,B⟩)	AB rayon

options	défaut	définition
out		permet de découper l'extérieur de l'objet

Il n'est pas nécessaire de mettre **radius** car c'est l'option par défaut.

26.2.1. Simple découpage

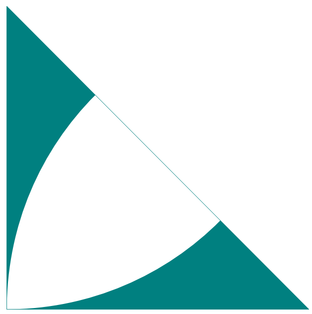



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoint(0,0){A} \tkzDefPoint(2,2){O}
  \tkzDefPoint(4,4){B} \tkzDefPoint(5,5){C}
  \tkzDrawPoints(O,A,B,C)
  \tkzLabelPoints(O,A,B,C)
  \tkzDrawCircle(O,A)
  \tkzClipCircle(O,A)
  \tkzDrawLine(A,C)
  \tkzDrawCircle[fill=teal!10,opacity=.5](C,O)
\end{tikzpicture}
```

26.3. 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}
```

26.4. Intersection de disques



```
\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}
```

Voir un exemple plus complexe sur le découpage ici : 44.6

26.5. Découpage d' un secteur

Attention the arguments vary according to the options.

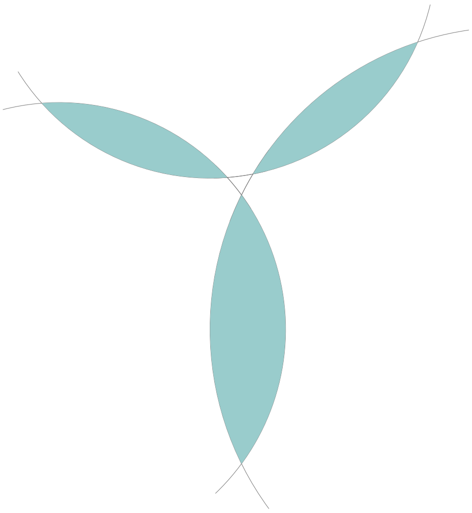
`\tkzClipSector[⟨options locales⟩](⟨O,...⟩)(⟨...⟩)`

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 Ti*k*Z for tracings...

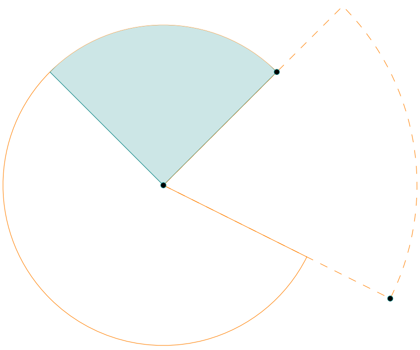
options	arguments	example
towards	(⟨pt,pt⟩)(⟨pt⟩)	`\tkzClipSector(O,A)(B)`
rotate	(⟨pt,pt⟩)(⟨angle⟩)	`\tkzClipSector[rotate](O,A)(90)`
R	(⟨pt,r⟩)(⟨angle 1,angle 2⟩)	`\tkzClipSector[R](O,2)(30,90)`

26.5.1. 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}
```

26.5.2. 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}
```

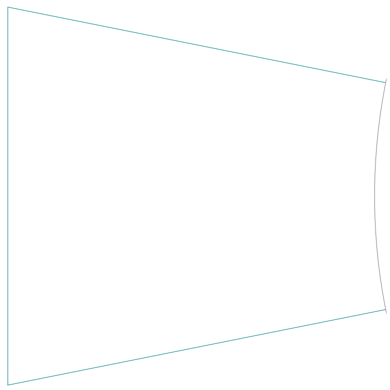
26.6. Options from TikZ: trim left or right

See the `pgfmanual`

26.7. TikZ Controls \pgfinterruptboundingbox and \endpgfinterruptboundingbox

This command temporarily interrupts the calculation of the box and configures a new box. See the `pgfmanual`

26.7.1. 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}
```

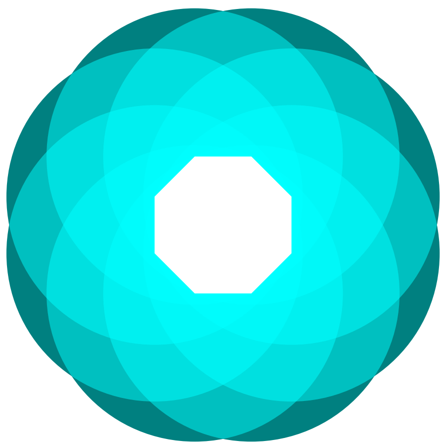
26.8. Reverse clip: tkzreverseclip

Pour utiliser cette option, une boîte de délimitation doit être définie.

```
\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} }}
```

26.8.1. Exemple avec \tkzClipPolygon[out]

`\tkzClipPolygon[out]`, `\tkzClipCircle[out]` use this option.



```
\begin{tikzpicture}[scale=.75]
\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}
```

Cinquième partie

Marquage

26.9. Marquer un segment \tkzMarkSegment

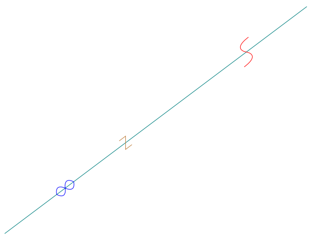
`\tkzMarkSegment[⟨options locales⟩](⟨pt1,pt2⟩)`

La macro permet de placer une marque sur un segment.

options	défaut	définition
pos	.5	position de la marque
color	black	couleur de la marque
mark	none	choix de la marque
size	4pt	taille de la marque

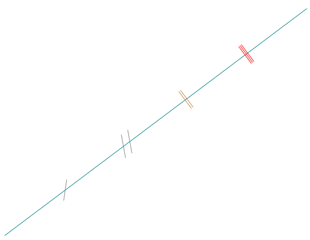
Les marques possibles sont celles fournies par Ti*k*Z, mais d'autres marques ont été créées sur la base d'une idée de Yves Combe.

26.9.1. Plusieurs marques



```
\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}
```

26.9.2. Utilisation d'une marque



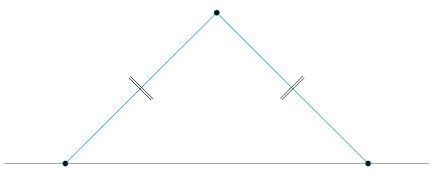
```
\begin{tikzpicture}
  \tkzDefPoint(2,1){A}
  \tkzDefPoint(6,4){B}
  \tkzDrawSegment(A,B)
  \tkzMarkSegment[color=gray,pos=0.2,mark=s|](A,B)
  \tkzMarkSegment[color=gray,pos=0.4,mark=s||](A,B)
  \tkzMarkSegment[color=brown,pos=0.6,mark=||](A,B)
  \tkzMarkSegment[color=red,pos=0.8,mark=|||](A,B)
\end{tikzpicture}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

26.10. Marquer des \tkzMarkSegments

`\tkzMarkSegments[⟨options locales⟩](⟨pt1,pt2 pt3,pt4 ...⟩)`

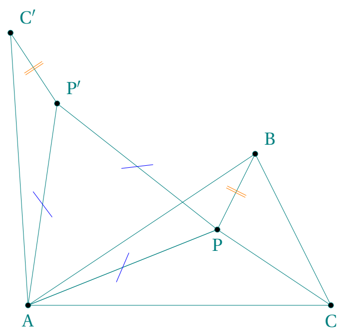
Les arguments sont une liste de paires de points séparées par des espaces. Les styles de Ti*k*Z sont disponibles pour les tracés.

26.10.1. Les marques pour un triangle isocèle



```
\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}
```

26.11. Une autre marque




```
\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}
```

26.12. Marquer un arc \tkzMarkArc

\tkzMarkArc[⟨options locales⟩](⟨pt1,pt2,pt3⟩)

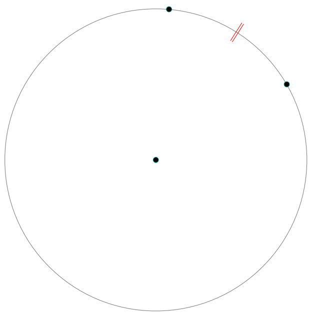
La macro vous permet de placer une marque sur un arc. pt1 est le centre, pt2 et pt3 sont les extrémités de l'arc.

options	défaut	définition
pos	.5	position de la marque
color	black	couleur de la marque
mark	none	choix de la marque
size	4pt	taille de la marque

Les marques possibles sont celles fournies par Ti*k*Z, mais d'autres marques ont été créées sur la base d'une idée de Yves Combe.

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

26.12.1. Plusieurs marques



```
\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}
```

26.13. Marquer un angle : \tkzMarkAngle

Une opération plus délicate car il y a de nombreuses options. Les symboles utilisés pour le marquage en plus de ceux de Ti*k*Z sont définis dans le fichier `tkz-lib-marks.tex` et désignés par les caractères suivants :

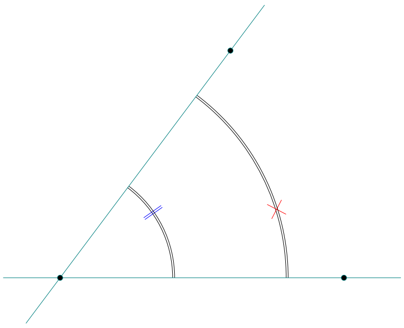
`|, ||,|||, z, s, x, o, oo`

`\tkzMarkAngle[⟨options locales⟩](⟨A,O,B⟩)`

O est le sommet. Attention, les arguments varient en fonction des options. Plusieurs marquages sont possibles. Vous pouvez simplement dessiner un arc ou ajouter une marque sur cet arc. Le style de l'arc est choisi avec l'option **`arc`**, le rayon de l'arc est donné par **`mksize`**, l'arc peut, bien sûr, être coloré.

options	défaut	définition
arc	l	choice of l, ll and lll (simple, double or triple).
size	1 (cm)	rayon de l'arc.
mark	none	choix de la marque.
mksize	4pt	symbol size (mark).
mkcolor	black	symbol color (mark).
mkpos	0.5	position of the symbol on the arc.

26.13.1. Exemple avec mark = x et avec mark =||



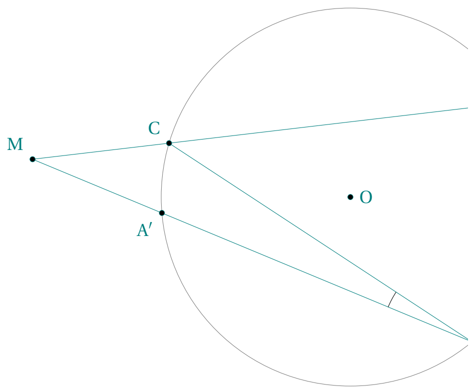
```
\begin{tikzpicture}[scale=.75]
    \tkzDefPoints{0/0/O,5/0/A,3/4/B}
    \tkzMarkAngle[size = 4,mark = x,
                  arc=ll,mkcolor = red,mkpos=.33](A,O,B)
    \tkzMarkAngle[size = 2,mark = ||,
                arc=ll,mkcolor = blue,mkpos=.66](A,O,B)
    \tkzDrawLines(O,A O,B)
    \tkzDrawPoints(O,A,B)
\end{tikzpicture}
```

`\tkzMarkAngles[⟨options locales⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.`

Avec les options courantes, il existe une macro pour les angles multiples.

26.14. Problème pour marquer un petit angle : Option veclen

Le problème vient de l'action "decorate" et de la valeur utilisée pour la taille dans \tkzMarkAngle. La solution est d'encapsuler la macro \tkzMarkAngle. Dans l'exemple suivant sans le "scope", le résultat est : Erreur LaTeX : Dimension trop grande.




```
\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}
```

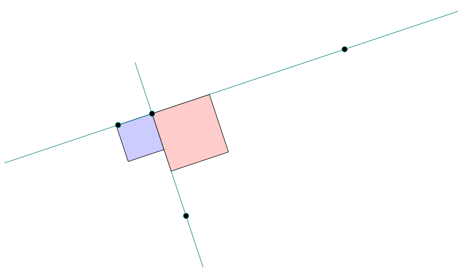
26.15. Marquer un angle droit \tkzMarkRightAngle

\tkzMarkRightAngle[⟨options locales⟩](⟨A,O,B⟩)

L'option **german** vous permet de changer le style du dessin. L'option **size** permet de changer la taille du dessin.

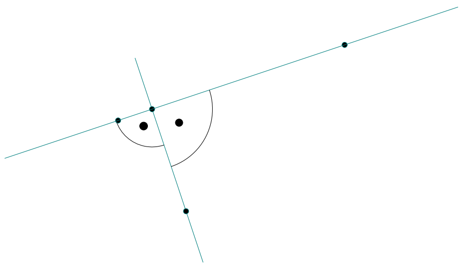
options	défaut	définition
german	normal	arc allemand avec point intérieur.
size	0.2	taille du côté.

26.15.1. Exemple de marquage d'un angle droit



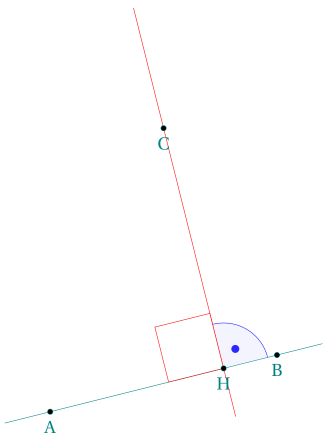
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
  \tkzDefPointBy[projection = onto B--A](P)  \tkzGetPoint{H}
  \tkzDrawLines[add=.5 and .5](P,H)
  \tkzMarkRightAngle[fill=blue!20,size=.5,draw](A,H,P)
  \tkzDrawLines[add=.5 and .5](A,B)
  \tkzMarkRightAngle[fill=red!20,size=.8](B,H,P)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

26.15.2. Exemple de marquage d'un angle droit, à l'allemande



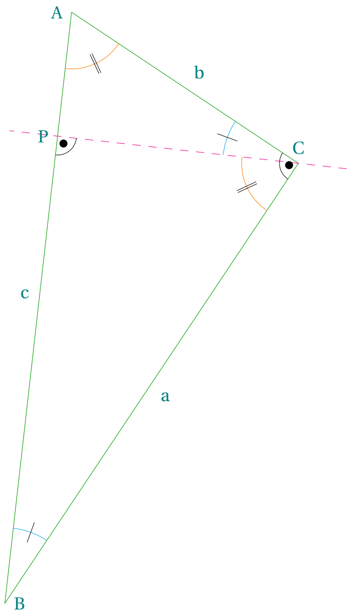
```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,3/1/B,0.9/-1.2/P}
  \tkzDefPointBy[projection = onto B--A](P)  \tkzGetPoint{H}
  \tkzDrawLines[add=.5 and .5](P,H)
  \tkzMarkRightAngle[german,size=.5,draw](A,H,P)
  \tkzDrawPoints[](A,B,P,H)
  \tkzDrawLines[add=.5 and .5](A,B)
  \tkzMarkRightAngle[german,size=.8](P,H,B)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

26.15.3. Mélange de styles




```
\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}
```

26.15.4. Exemple complet




```
\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}
```

26.16. \tkzMarkRightAngles

`\tkzMarkRightAngles[⟨options locales⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.`

Avec des options communes, il existe une macro pour des angles multiples.

26.17. Angles Library

Si vous préférez utiliser la bibliothèque Ti*k*Z`angles`, vous pouvez marquer les angles avec la macro `\tkzPicAngle` et `\tkzPicRightAngle`.

`\tkzPicAngle[⟨tikz options⟩](⟨A,O,B⟩)`

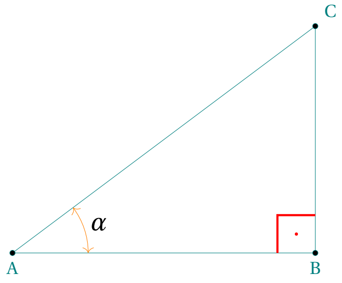
options	exemple	définition
tikz option	see below	drawing of the angle $\widehat{AOB}$.

`\tkzPicRightAngle[⟨tikz options⟩](⟨A,O,B⟩)`

options	exemple	définition
tikz option	see below	tracé de l'angle droit $\widehat{AOB}$.

Vous devez connaître les options possibles de la **`angles`** *“ library”*

26.17.1. Angle avec 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}
```

Sixième partie

Étiquetage

27. Étiquetage

27.1. Etiquette pour un point

Il est possible d'ajouter plusieurs étiquettes au même endroit en utilisant cette macro plusieurs fois.

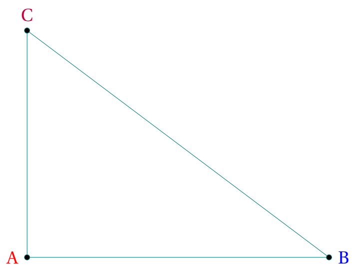
`\tkzLabelPoint[⟨options locales⟩](⟨point⟩){⟨label⟩}`

arguments	example
point	`\tkzLabelPoint(A){$A_1$}`

options	défaut	définition
TikZ options		couleur, position etc.

En option, nous pouvons utiliser n'importe quel style de Ti*k*Z, en particulier le placement avec au-dessus, à droite, les points...

27.1.1. Exemple avec \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}
```

27.1.2. Label et référence

La référence d'un point est l'objet qui permet d'utiliser le point, l'étiquette est le nom du point qui sera affiché.


```
\begin{tikzpicture}
   \tkzDefPoint(2,0){A}
   \tkzDrawPoint(A)
   \tkzLabelPoint[above](A){$A_1$}
 \end{tikzpicture}
```

27.2. Ajouter des étiquettes aux points \tkzLabelPoints

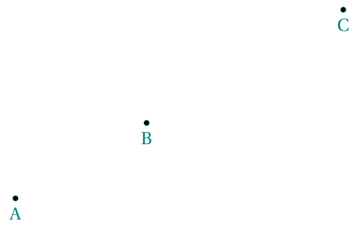
Il est possible de placer plusieurs étiquettes rapidement lorsque les références des points sont identiques à celles des étiquettes et que les étiquettes sont placées de la même manière par rapport aux points. Par défaut, `en bas à droite` est choisi.

`\tkzLabelPoints[⟨options locales⟩](⟨A₁,A₂,...⟩)`

arguments	example	result
list of points	`\tkzLabelPoints(A,B,C)`	Affichage de A, B and C

Cette macro réduit le nombre de lignes de code, mais il n'est pas évident que tous les points nécessitent le même positionnement de l'étiquette.

27.2.1. Exemple avec \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}
```

28. Étiquette d'un segment

\tkzLabelSegment[⟨options locales⟩](⟨pt1,pt2⟩){⟨label⟩}

Cette macro permet de placer une étiquette le long d'un segment ou d'une ligne. Les options sont celles de Ti*k*Z par exemple `pos`.

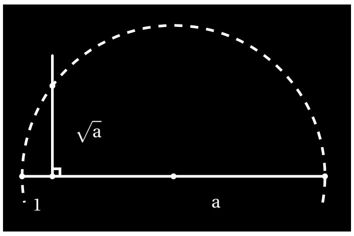
argument	example	definition
label	`\tkzLabelSegment(A,B){5}`	texte de l'étiquette
(pt1,pt2)	(A,B)	étiquette le long de [AB]

options	défaut	définition
pos	.5	position de l'étiquette

28.0.1. Premier exemple

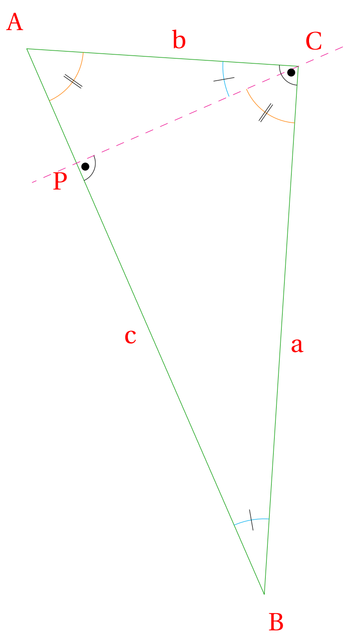

```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefPoint(6,0){B}
\tkzDrawSegment(A,B)
\tkzLabelSegment[above,pos=.8](A,B){$a$}
\tkzLabelSegment[below,pos=.2](A,B){$4$}
\end{tikzpicture}
```

28.0.2. Exemple : tableau noir




```
\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}
```

28.0.3. Étiquettes et options : swap

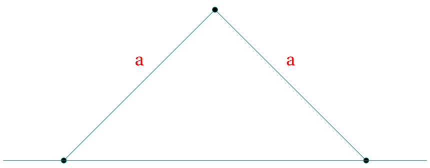



```
\begin{tikzpicture}[rotate=-60]
\tkzSetUpStyle[red,auto]{label style}
\tkzDefPoint(0,1){A}
\tkzDefPoint(2,4){C}
\tkzDefPointWith[orthogonal normed,K=7](C,A)
\tkzGetPoint{B}
\tkzDefSpcTriangle[orthic](A,B,C){N,O,P}
\tkzDefTriangleCenter[circum](A,B,C)
\tkzGetPoint{O}
\tkzDrawPolygon[green!60!black](A,B,C)
\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[⟨options locales⟩](⟨pt1,pt2 pt3,pt4 ...⟩)`

Les arguments sont une liste de couples à deux points. Les styles de TikZ sont disponibles pour le tracé.

28.0.4. Étiquettes pour un triangle isocèle




```
\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}
```

29. Ajouter des étiquettes sur une ligne droite \tkzLabelLine

`\tkzLabelLine[⟨options locales⟩](⟨pt1,pt2⟩){⟨label⟩}`

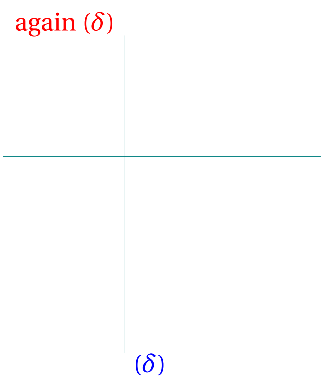
arguments	défault	définition
label		`\tkzLabelLine(A,B){$\Delta$}`

options	défaut	définition
pos	.5	**pos** est une option pour Ti*k*Z, mais elle est essentielle dans ce cas.…

En option, et en plus de **pos**, vous pouvez utiliser tous les styles de Ti*k*Z, en particulier le placement avec **above**, **right**, …

29.0.1. Exemple avec \tkzLabelLine

Une option importante est **pos**, c'est celle qui permet de placer l'étiquette le long de la droite. La valeur de **pos** peut être supérieure à 1 ou négative.




```
\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}
```

29.1. Etiquette d'un angle : \tkzLabelAngle

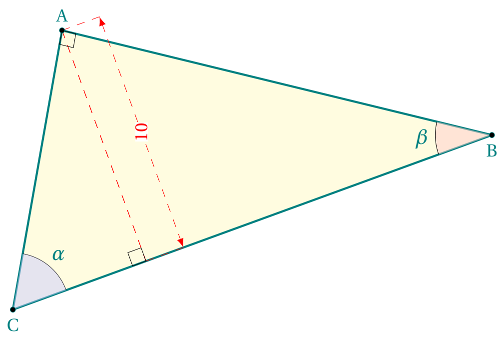
`\tkzLabelAngle[⟨options locales⟩](⟨A,O,B⟩)`

Il n'y a qu'une seule option, dist (avec ou sans unité), qui peut être remplacée par l'option pos de TikZ (sans unité pour cette dernière). Par défaut, la valeur est en centimètres.

options	défaut	définition
pos	1	ou dist, contrôle la distance entre le haut et l'étiquette.

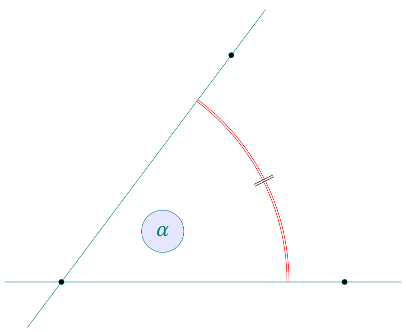
Il est possible de déplacer l'étiquette avec toutes les options de TikZ : rotation, décalage, dessous, etc.

29.1.1. Exemple d'auteur js bibra stackexchange



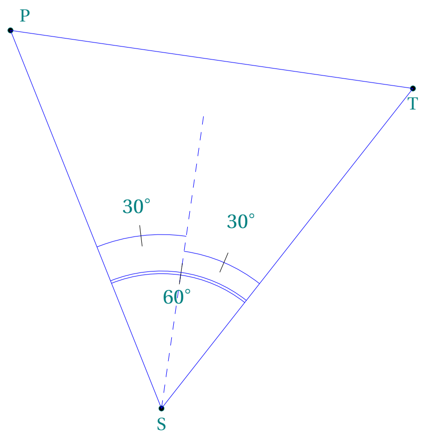

```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoint(0,0){C}
  \tkzDefPoint(20:9){B}
  \tkzDefPoint(80:5){A}
  \tkzDefPointsBy[projection=onto B--C](A){a}
  \tkzDrawPolygon[thick,fill=yellow!15](A,B,C)
  \tkzDrawSegment[dashed, red](A,a)
  \tkzDrawSegment[style=red, dashed,
  dim={$10$,15pt,midway,font=\scriptsize,
   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}
```

29.1.2. Avec 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}
```

29.1.3. pos et \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[⟨options locales⟩](⟨A,O,B⟩)(⟨A',O',B'⟩)etc.

Avec des options communes, il existe une macro pour des angles multiples.

Il reste enfin à pouvoir donner une étiquette pour désigner un cercle et si plusieurs possibilités s'offrent, nous verrons ici \tkzLabelCircle.

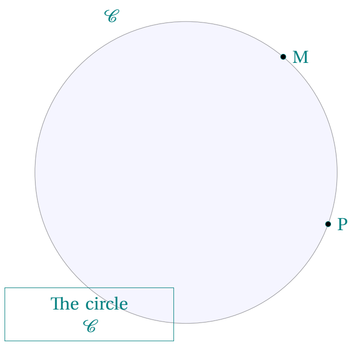
29.2. Donner une étiquette à un cercle

\tkzLabelCircle[⟨tikz options⟩](⟨O,A⟩)(⟨angle⟩){⟨label⟩}

options	défaut	définition
tikz options		cercle O centre passant par A

Nous pouvons utiliser les styles de TikZ. L'étiquette est créée et donc "passée" entre accolades.

29.2.1. Exemple




```
\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}
```

30. Etiquette d'un arc

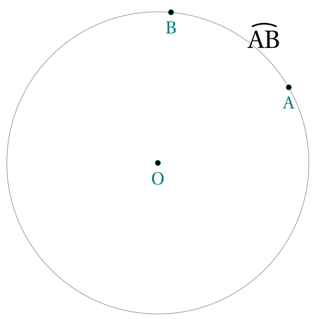
\tkzLabelArc[⟨options locales⟩](⟨pt1,pt2,pt3⟩){⟨label⟩}

Cette macro permet de placer une étiquette le long d'un arc. Les options sont celles de Ti*k*Z par exemple `pos`.

argument	exemple	définition
label	**\tkzLabelArc(A,B){5}**	label text
(pt1,pt2,pt3)	(O,A,B)	étiquette le long de l'arc $\widearc{AB}$

options	défaut	définition
pos	.5	label's position

30.0.1. Étiquette sur l'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}
```

Septième partie

Compléments

31. Utilisation du compas

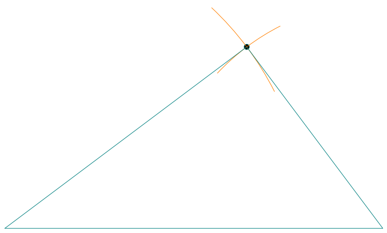
31.1. Macro principale \tkzCompass

`\tkzCompass[⟨options locales⟩](⟨A,B⟩)`

Cette macro permet de laisser une trace de compas, c'est-à-dire un arc de cercle à un point désigné. Le centre doit être indiqué. Plusieurs options spécifiques vont modifier l'aspect de l'arc ainsi que les options de TikZ telles que le style, la couleur, l'épaisseur du trait, etc.
Vous pouvez définir la longueur de l'arc avec l'option `length` ou l'option `delta`.

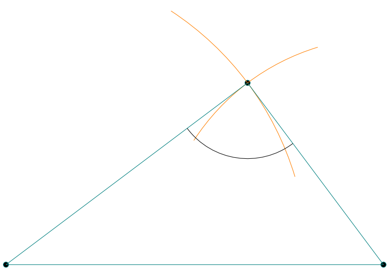
options	défaut	définition
delta	0 (deg)	Augmente l'angle de l'arc de manière symétrique
length	1 (cm)	Modifie la longueur (en cm)

31.1.1. 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}
```

31.1.2. 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}
```

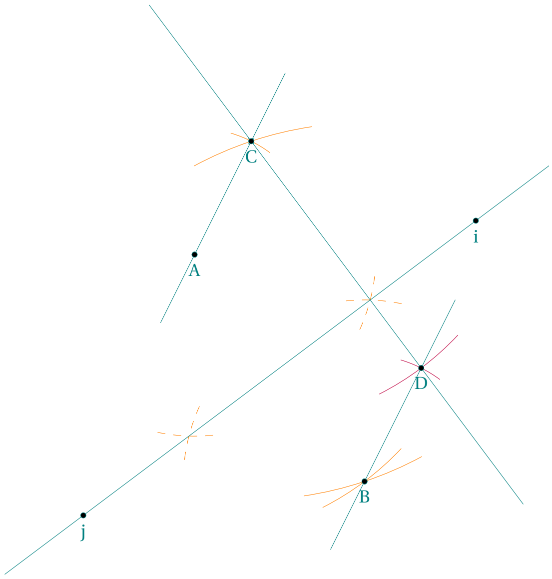
31.2. Constructions multiples \tkzCompasss

`\tkzCompasss[⟨options locales⟩](⟨pt1,pt2 pt3,pt4,…⟩)`

Attention les arguments sont des listes de deux points. Cela permet d'économiser quelques lignes de code.

options	défaut	définition
delta	0	Modifie l'angle de l'arc en l'augmentant symétriquement
length	1	Modifie la longueur

31.2.1. Utilisation de \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}
```

32. Le Show

32.1. Montrer les constructions de certaines lignes \tkzShowLine

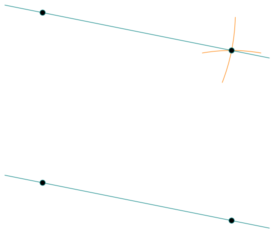
`\tkzShowLine[⟨options locales⟩](⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)`

Ces constructions concernent les médiatrices, les droites perpendiculaires ou parallèles passant par un point donné et les bissectrices. Les arguments sont donc des listes de deux ou trois points. Plusieurs options permettent d'ajuster les constructions. L'idée de cette macro vient de `Yves Combe`.

options	default	definition
mediator	mediator	affiche les constructions d'un médiateur
perpendicular	mediator	constructions pour une perpendiculaire
orthogonal	mediator	idem
bisector	mediator	constructions pour une bissectrice
K	1	cercle dans un triangle
length	1	in cm, longueur d'un arc
ratio	.5	rapport de longueur d'arc
gap	2	placer le point de construction
size	1	rayon d'un arc (voir bissectrice)

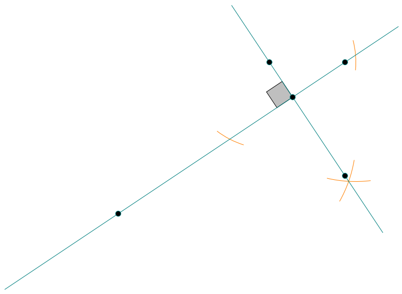
Il faut ajouter, bien sûr, tous les styles de Ti*k*Z pour les tracés…

32.1.1. Exemple of \tkzShowLine et parallel



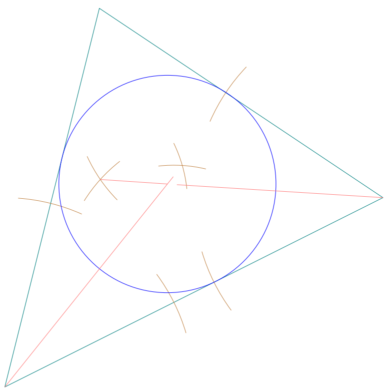
```
\begin{tikzpicture}
 \tkzDefPoints{-1.5/-0.25/A,1/-0.75/B,-1.5/2/C}
 \tkzDrawLine(A,B)
 \tkzDefLine[parallel=through C](A,B) \tkzGetPoint{c}
 \tkzShowLine[parallel=through C](A,B)
 \tkzDrawLine(C,c) \tkzDrawPoints(A,B,C,c)
\end{tikzpicture}
```

32.1.2. Exemple de \tkzShowLine et perpendicular



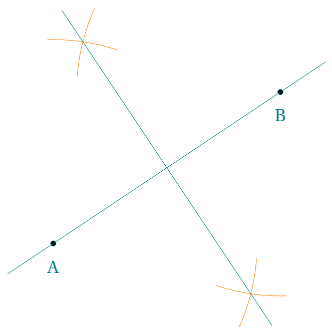
```
\begin{tikzpicture}
\tkzDefPoints{0/0/A, 3/2/B, 2/2/C}
\tkzDefLine[perpendicular=through C,K=-.5](A,B) \tkzGetPoint{c}
\tkzShowLine[perpendicular=through C,K=-.5,gap=3](A,B)
\tkzDefPointBy[projection=onto A--B](c)\tkzGetPoint{h}
\tkzMarkRightAngle[fill=lightgray](A,h,C)
\tkzDrawLines[add=.5 and .5](A,B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

32.1.3. Exemple de \tkzShowLine et 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}
```

32.1.4. Exemple de \tkzShowLine et 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}
```

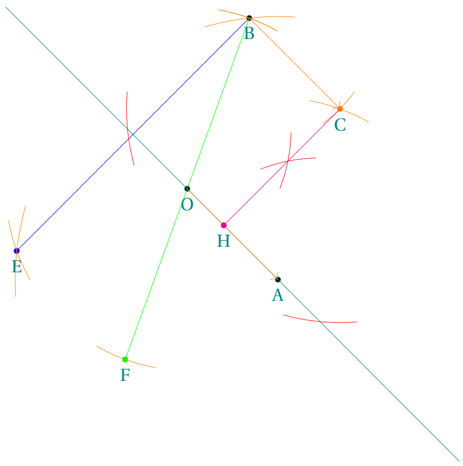
32.2. Constructions de certaines transformations \tkzShowTransformation

\tkzShowTransformation[⟨options locales⟩](⟨pt1,pt2⟩) or (⟨pt1,pt2,pt3⟩)

Ces constructions concernent les symétries orthogonales, les symétries centrales, les projections orthogonales et les translations. Plusieurs options permettent d'ajuster les constructions. L'idée de cette macro vient de Yves Combe.

options	défaut	définition
reflection= over pt1--pt2	reflection	constructions de symétrie orthogonale
symmetry=center pt	reflection	constructions de symétrie centrale
projection=onto pt1--pt2	reflection	constructions d'une projection
translation=from pt1 to pt2	reflection	constructions d'une traduction
K	1	cercle dans un triangle
length	1	longueur de l'arc
ratio	.5	rapport de longueur d'arc
gap	2	placer le point de construction
size	1	rayon d'un arc (voir bissectrice)

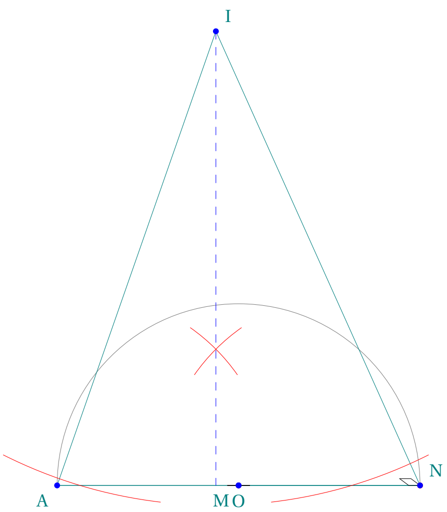
32.2.1. Exemple d'utilisation de \tkzShowTransformation




```
\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}
```

32.2.2. Un autre exemple de l'utilisation de \tkzShowTransformation

Vous retrouverez cette figure, mais sans les éléments de construction.




```
\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}
```

33. Rapporteur

Basée sur une idée d'Yves Combe, la macro suivante permet de dessiner un rapporteur. Le principe de fonctionnement est encore plus simple. Il suffit de nommer une demi-droite (un rayon). Le rapporteur sera placé sur l'origine O, la direction de la demi-droite est donnée par A. L'angle est mesuré dans la direction directe du cercle trigonométrique.

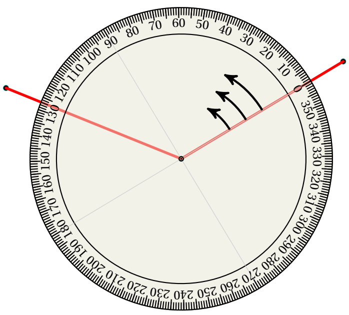
33.1. La macro \tkzProtractor

\tkzProtractor[⟨options locales⟩](⟨O,A⟩)

options	défaut	définition
lw	0.4 pt	épaisseur du trait
scale	1	ratio : ajuste la taille du rapporteur
return	false	cercle trigonométrique indirect

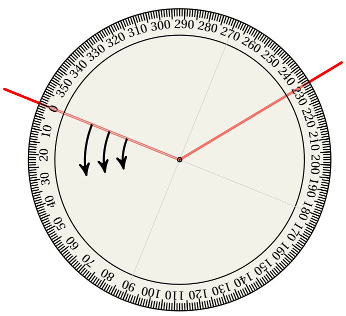
33.1.1. Le rapporteur circulaire

Mesure dans le sens de la marche



```
\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}
```

33.1.2. Le rapporteur circulaire, transparent et retourné



```
\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}
```

34. Outils divers et outils mathématiques

34.1. Dupliquer un segment

Il s'agit de construire sur une demi-ligne donnée un segment de même longueur qu'un segment donné.

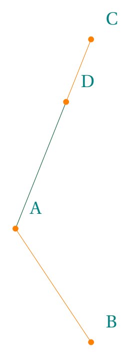
\tkzDuplicateSegment(⟨pt1,pt2⟩)(⟨pt3,pt4⟩){⟨pt5⟩}

Il s'agit de créer sur une demi-droite donnée un segment de même longueur qu'un segment donné. C'est en fait la définition d'un point. `\tkzDuplicateSegment` est le nouveau nom de `\tkzDuplicateLen`.

arguments, exemples et explications		
(pt1,pt2)(pt3,pt4){pt5}	`\tkzDuplicateSegment(A,B)(E,F){C}`	AC=EF and C ∈ [AB)

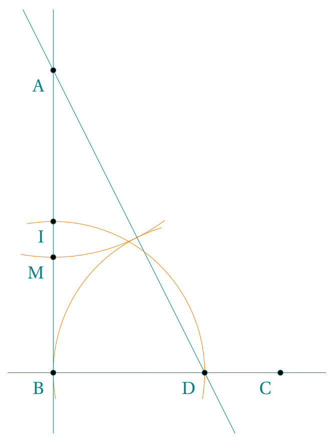
The macro `\tkzDuplicateLength` *est identique à celui-ci.*

34.1.1. Utilisation de\tkzDuplicateSegment



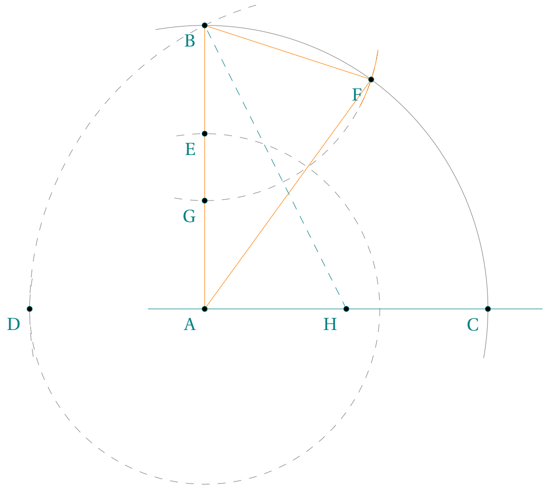

```
\begin{tikzpicture}[scale=.5]
 \tkzDefPoints{0/0/A,2/-3/B,2/5/C}
 \tkzDuplicateSegment(A,B)(A,C)
 \tkzGetPoint{D}
 \tkzDrawSegments[new](A,B A,C)
 \tkzDrawSegment[teal](A,D)
 \tkzDrawPoints[new](A,B,C,D)
 \tkzLabelPoints[above right=3pt](A,B,C,D)
\end{tikzpicture}
```

34.1.2. Proportion d'or avec \tkzDuplicateSegment




```
\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}
```

34.1.3. Triangle d'or ou triangle sublime




```
\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}
```

34.2. Longueur du segment \tkzCalcLength

Il existe une option dans Ti*k*Z nommée **veclen**. Cette option est utilisée pour calculer AB si A et B sont deux points.
Le seul problème pour moi est que la version de Ti*k*Z n'est pas assez précise dans certains cas. Ma version utilise le paquet `xfp` et est plus lente, mais plus précise.

\tkzCalcLength[⟨options locales⟩](⟨pt1,pt2⟩)

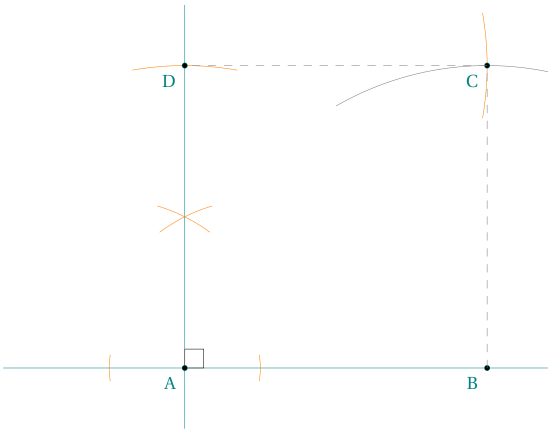
Vous pouvez enregistrer le résultat avec la macro **\tkzGetLength**, par exemple **\tkzGetLength{dAB}** définit la macro **\dAB**.

arguments, exemples et explications		
(pt1,pt2){nom de la macro}	\tkzCalcLength(A,B)	\dAB donne AB en cm

Une seule option

options	défaut	exemple
cm	true	\tkzCalcLength(A,B) Après \tkzGetLength{dAB} \dAB donne AB en cm

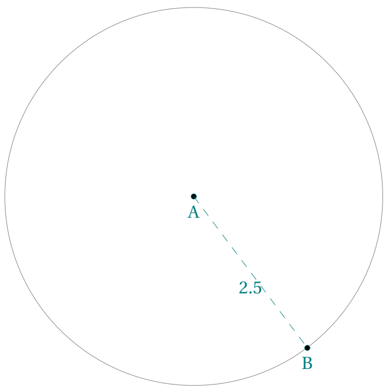
34.2.1. Construction d'un carré au compas




```
\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}
```

34.2.2. Exemple

La macro `\tkzDefCircle[radius](A,B)` définit le rayon que nous récupérons avec `\tkzGetLength`, ce résultat est dans `cm`.




```
\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}
```

34.3. Transformation de pt en cm ou de cm en pt

Je ne suis pas sûr que cela soit nécessaire et il s'agit seulement d'une division par 28,45274 et d'une multiplication par le même nombre. Les macros sont :

`\tkzpttocm(⟨number⟩){⟨nom de la macro⟩}`

Le résultat est stocké dans une macro.

arguments, exemples et explications		
`(number){nom de la macro}`	`\tkzpttocm(120){len}`	`\len` donne un nombre de `cm`

Vous devrez utiliser `\len` en même temps que `cm`.

34.4. Change of unit

\tkzcmtopt(⟨number⟩){⟨name of macro⟩}

Le résultat est stocké dans une macro.

arguments, exemples et explications		
`(number){nom de la macro}`	`\tkzcmtopt(5){len}`	`\len` longueur en **pts**

Le résultat peut être utilisé avec `\len` *pt*

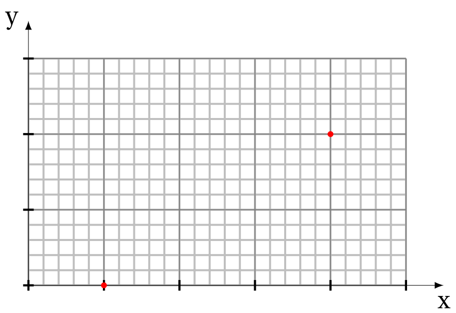
34.5. Obtenir les coordonnées d'un point

\tkzGetPointCoord(⟨A⟩){⟨name of macro⟩}

arguments, exemples et explications		
`(point){nom de la macro}`	`\tkzGetPointCoord(A){A}`	`\Ax` and `\Ay` coordonnées de A

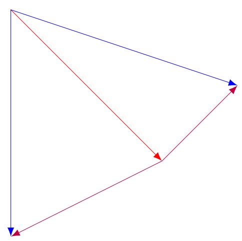
Enregistre dans deux macros les coordonnées d'un point. Si le nom de la macro est **p**, *alors* \x *et* \y *donnent les coordonnées du point choisi avec le cm comme unité.*

34.5.1. Coordonner le transfert avec \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}
```

34.5.2. Somme de vecteurs avec \tkzGetPointCoord



```
\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}
```

34.6. Échanger les étiquettes des points

\tkzSwapPoints(⟨pt1,pt2⟩)

arguments, exemples et explications

(pt1,pt2)	**\tkzSwapPoints**(A,B)	now A a pour coordonnées B

Les points ont échangé leurs coordonnées.

34.6.1. Utilisation de \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}
```

34.7. Produit en points

En géométrie euclidienne, le produit de points des coordonnées cartésiennes de deux vecteurs est largement utilisé.

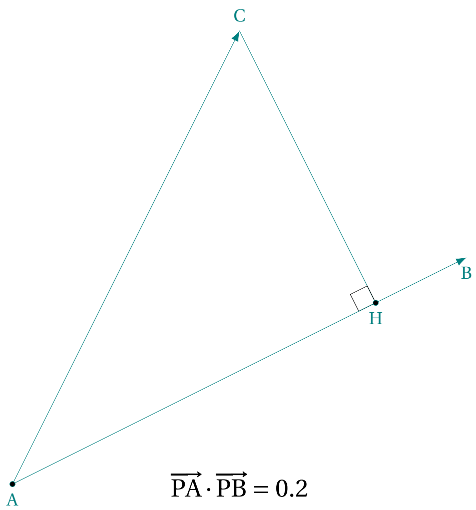
\tkzDotProduct(⟨pt1,pt2,pt3⟩)

Le produit en points de deux vecteurs $\vec{u} = [a, b]$ et $\vec{v} = [a', b']$ est définie comme suit : $\vec{u} \cdot \vec{v} = aa' + bb'$
$\vec{u} = \overrightarrow{pt1pt2}$ $\vec{v} = \overrightarrow{pt1pt3}$
arguments, exemples et explications

(pt1,pt2,pt3)	**\tkzDotProduct**(A,B,C)	le résultat est $\overrightarrow{AB} \cdot \overrightarrow{AC}$

Le résultat est un nombre qui peut être récupéré à l'aide de la fonction **\tkzGetResult**.

34.7.1. Exemple simple

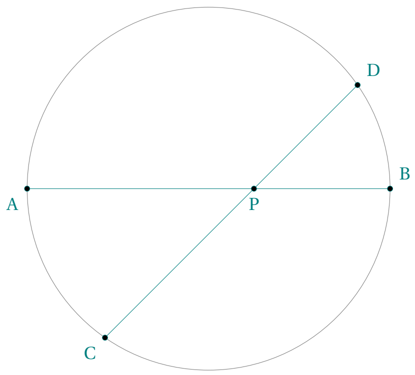


$\overrightarrow{PA} \cdot \overrightarrow{PB} = 0.2$

$PA \times PH = 0.2$

```
\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) {$\overrightarrow{PA}\cdot%
   \overrightarrow{PB}=\pabc$};
  \tkzDotProduct(A,H,B) \tkzGetResult{phab}
 % \pgfmathparse{round(10*\phab)/10}
  \let\phab\pgfmathresult
  \node at (1,-4) {$PA \times PH = \phab $};
\end{tikzpicture}
```

34.7.2. Points cocycliques



$\overrightarrow{PA} \cdot \overrightarrow{PB} = \overrightarrow{PC} \cdot \overrightarrow{PD}$

$\overrightarrow{PA} \cdot \overrightarrow{PB} = -15.0$

$\overrightarrow{PC} \cdot \overrightarrow{PD} = -15.0$

```
\begin{tikzpicture}[scale=.6]
  \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}
```

34.8. Puissance d'un point par rapport à un cercle

\tkzPowerCircle(⟨pt1⟩)(⟨pt2,pt3⟩)

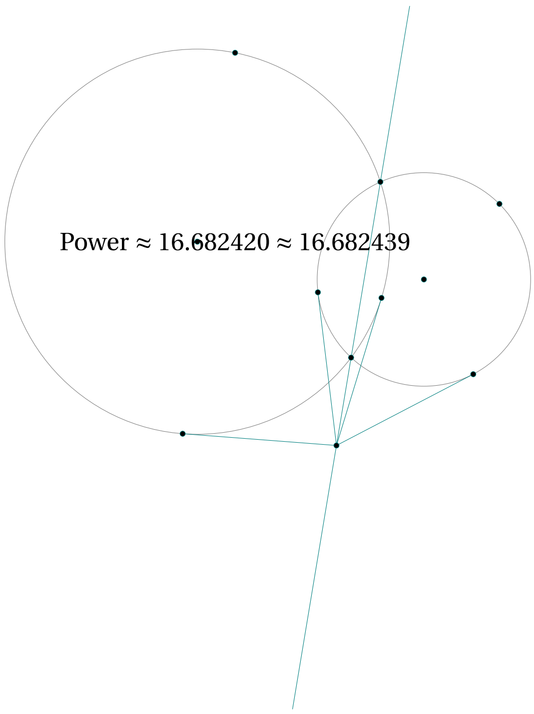
arguments, exemples et explications

(pt1)(pt2,pt3)	\tkzPowerCircle(A)(O,M)	puissance de A

Le résultat est un nombre qui représente la puissance d'un point par rapport à un cercle.

34.8.1. Le pouvoir de l'axe radical

Dans cet exemple, l'axe radical (EF) a été tracé. Un point H a été choisi sur (EF) et la puissance du point H par rapport au cercle de centre A a été calculée ainsi que PS^2. Vous pouvez vérifier que la puissance de H par rapport au cercle de centre C ainsi que HS'^2, HT^2, HT'^2 donnent le même résultat.




```
\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}
```

34.9. Axe radical

En géométrie, l'axe radical de deux cercles non concentriques est l'ensemble des points dont les puissances par rapport aux cercles sont égales. Ici, `\tkzDefRadicalAxis(A,B)(C,D)` donne l'axe radical des deux cercles $\mathscr{C}(A,B)$ et $\mathscr{C}(C,D)$.

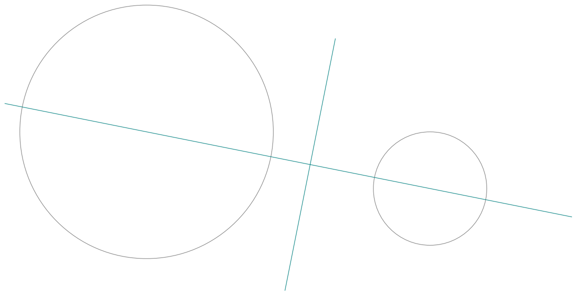
\tkzDefRadicalAxis(⟨pt1,pt2⟩)(⟨pt3,pt4⟩)

arguments, exemples et explications

(pt1,pt2)(pt3,pt4)	\tkzDefRadicalAxis(A,B)(C,D)	Centres A et C.

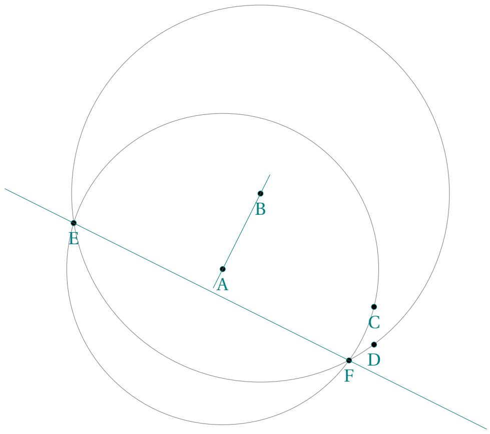
Le résultat est deux points de l'axe radical.

34.9.1. Deux cercles disjoints



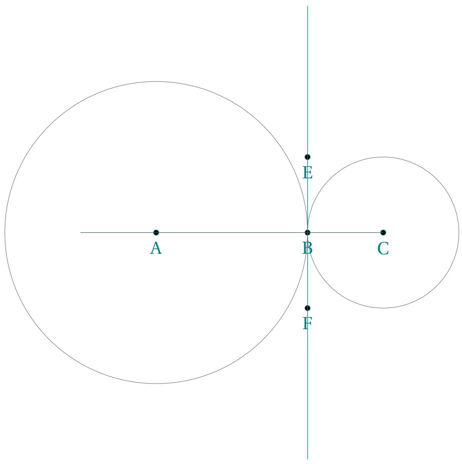
```
\begin{tikzpicture}[scale=.75]
    \tkzDefPoints{-1/0/A,0/2/B,4/-1/C,4/0/D}
    \tkzDrawCircles(A,B C,D)
    \tkzDefRadicalAxis(A,B)(C,D)
     \tkzGetPoints{E}{F}
    \tkzDrawLine[add=1 and 2](E,F)
    \tkzDrawLine[add=.5 and .5](A,C)
\end{tikzpicture}
```

34.10. Deux cercles qui se croisent



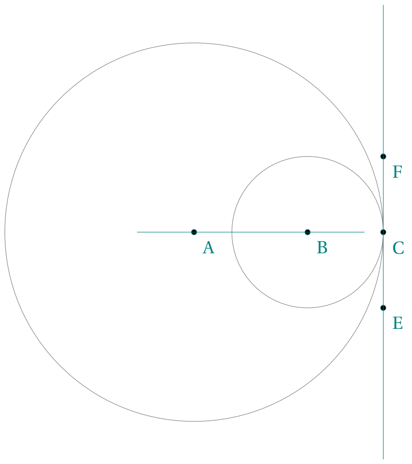

```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{-1/0/A,0/2/B,3/-1/C,3/-2/D}
  \tkzDrawCircles(A,C B,D)
  \tkzDefRadicalAxis(A,C)(B,D)
  \tkzGetPoints{E}{F}
  \tkzDrawPoints(A,B,C,D,E,F)
    \tkzLabelPoints(A,B,C,D,E,F)
  \tkzDrawLine[add=.25 and .5](E,F)
  \tkzDrawLine[add=.25 and .25](A,B)
\end{tikzpicture}
```

34.11. Deux cercles tangents extérieurement



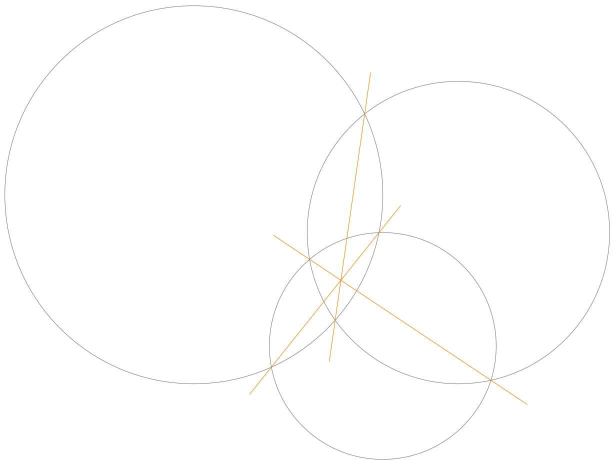

```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoints{0/0/A,4/0/B,6/0/C}
  \tkzDrawCircles(A,B C,B)
  \tkzDefRadicalAxis(A,B)(C,B)
  \tkzGetPoints{E}{F}
  \tkzDrawPoints(A,B,C,E,F)
    \tkzLabelPoints(A,B,C,E,F)
  \tkzDrawLine[add=1 and 1](E,F)
  \tkzDrawLine[add=.5 and .5](A,B)
\end{tikzpicture}
```

34.12. Deux cercles tangents intérieurement




```
\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}
```

34.12.1. Trois cercles



```
\begin{tikzpicture}[scale=.5]
  \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}
```

34.13. \tkzIsLinear, \tkzIsOrtho

\tkzIsLinear(⟨pt1,pt2,pt3⟩)

arguments	example	explanation
(pt1,pt2,pt3)	**\tkzIsLinear**(A,B,C)	A,B,C aligned ?

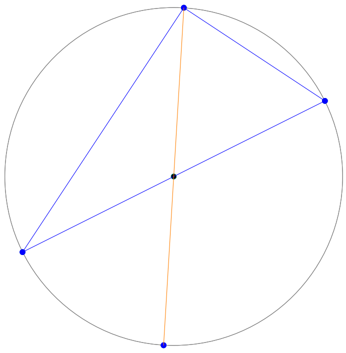
\tkzIsLinear *permet de tester l'alignement des trois points* pt1,pt2,pt3.

\tkzIsOrtho(⟨pt1,pt2,pt3⟩)

arguments, exemples et explications		
(pt1,pt2,pt3)	**\tkzIsOrtho**(A,B,C)	(AB) ⊥ (AC) ?

\tkzIsOrtho *permet de tester l'orthogonalité des droites* (pt1pt2) *et* (pt1pt3).

34.13.1. Utilisation de \tkzIsOrtho et \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}
```

Huitième partie

Travailler avec le style

35. Styles prédéfinis

La manière de procéder dépendra de l'utilisation que vous ferez du paquet. Une méthode qui me semble correcte consiste à utiliser autant que possible des styles prédéfinis afin de séparer le contenu de la forme. Cette méthode sera la bonne si vous envisagez de créer un document (comme cette documentation) avec de nombreuses figures. Nous verrons comment définir un style global pour un document. Nous verrons comment utiliser un style localement.
Le fichier **tkz-euclide.cfg** contient les styles prédéfinis des principaux objets. Parmi ceux-ci, les plus importants sont les points, les lignes, les segments, les cercles, les arcs et les tracés au compas. Si vous utilisez toujours les mêmes styles et si vous créez de nombreuses figures, il est intéressant de créer vos propres styles. Pour ce faire, vous devez savoir quelles sont les caractéristiques que vous pouvez modifier. Il sera nécessaire de connaître quelques notions de Ti*k*Z.
Les styles prédéfinis sont des styles globaux. Ils existent avant la création des figures. Il vaut mieux éviter de les changer entre deux figures. Par contre ces styles peuvent être modifiés dans une figure temporairement. Là, les styles sont définis localement et n'influencent pas les autres figures.
Pour le document que vous lisez, voici comment j'ai défini les différents styles.

```
\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}
```

La macro **\tkzSetUpColors** vous permet de définir la couleur d'arrière-plan ainsi que la couleur du texte. Si vous ne l'utilisez pas, les couleurs de votre document seront utilisées ainsi que les polices. Voyons comment définir les styles des principaux objets.

36. Style des points

C'est ainsi que les points sont définis :

```
\tikzset{point style/.style = {%
       draw         = \tkz@euc@pointcolor,
       inner sep    = 0pt,
       shape        = \tkz@euc@pointshape,
       minimum size = \tkz@euc@pointsize,
       fill         = \tkz@euc@pointcolor}}
```

Il est bien sûr possible d'utiliser **\tikzset** mais vous pouvez utiliser une macro fournie par le paquet. Vous pouvez utiliser la macro **\tkzSetUpPoint** globalement ou localement, examinons cette possibilité.

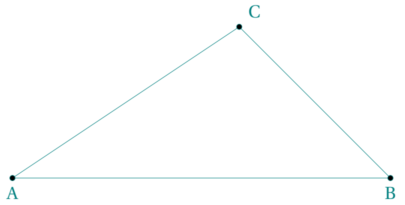
36.1. Utilisation de \tkzSetUpPoint

\tkzSetUpPoint[⟨options locales⟩]

options	défaut	définition
color	black	couleur du point
size	3	taille du point
fill	black!50	couleur du point intérieur
shape	circle	point shape circle, cross or cross out

36.1.1. Style global ou style local

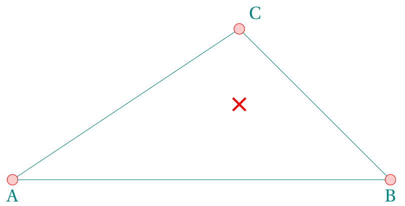
Tout d'abord voici une figure créée avec les styles de ma documentation, puis le style des points est modifié dans l'environnement **tikzspicture**.
Vous pouvez utiliser la macro `\tkzSetUpPoint` globalement ou localement. Si vous placez cette macro dans votre préambule ou avant votre première figure, le style de point sera valable pour toutes les figures de votre document. Il sera possible d'utiliser un autre style localement en utilisant cette commande dans un environnement **tikzpicture**. Examinons cette possibilité.




```
\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}
```

36.1.2. Style local

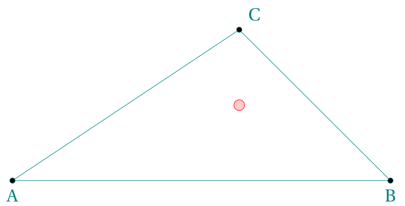
Le style des points est modifié localement dans la deuxième figure




```
\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}
```

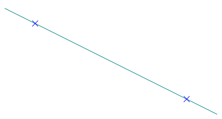
36.1.3. Style et scope

Les points récupèrent le style initial. Le point D a un nouveau style limité par l'environnement `scope`. Il est également possible d'utiliser `{...}` ouLes points retrouvent le style initial. Le point D a un nouveau style limité par l'environnement `scope`. Il est également possible d'utiliser `{...}` ou `{begingoup ... \groupe de fin`.



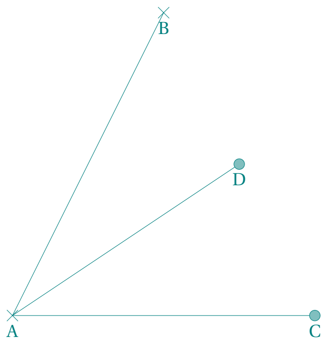

```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,5/0/B,3/2/C,3/1/D}
  \tkzDrawPolygon(A,B,C)
  \tkzDrawPoints(A,B,C)
  \begin{scope}
    \tkzSetUpPoint[size=4,color=red,fill=red!20]
    \tkzDrawPoint(D)
  \end{scope}
  \tkzLabelPoints(A,B)
  \tkzLabelPoints[above right](C)
\end{tikzpicture}
```

36.1.4. Exemple simple avec \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}
```

36.1.5. Utilisation de \tkzSetUpPoint dans un groupe




```
\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}
```

37. Style des lignes

Vous avez plusieurs possibilités pour modifier le style d'une ligne. Vous pouvez modifier le style d'une ligne avec `\tkzSetUpLine` ou modifier directement le style des lignes avec `\tikzset{line style/.style = ... }`
Rappel concernant **`largeur de ligne`** : Il existe un certain nombre de styles prédéfinis qui fournissent des moyens plus "naturels" de définir la largeur de ligne. Vous pouvez également redéfinir ces styles.

style et valeur prédéfinis de **`largeur de ligne`**

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

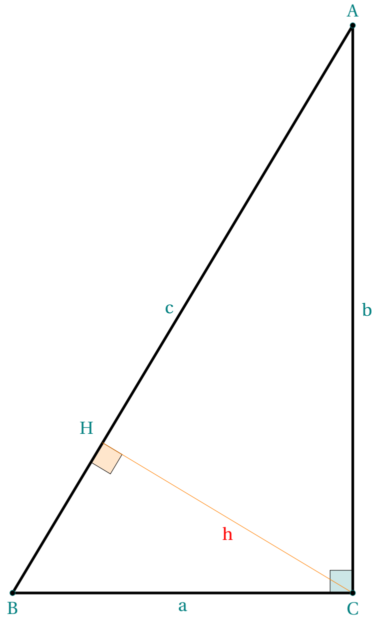
37.1. Utilisation de \tkzSetUpLine

Il s'agit d'une macro qui permet de définir le style de toutes les lignes.

`\tkzSetUpLine[⟨options locales⟩]`

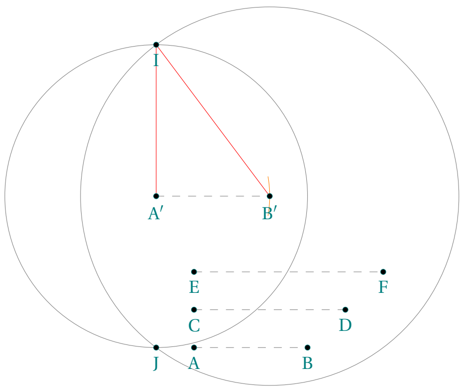
options	défaut	définition
`color`	`black`	`couleur des lignes de construction`
`line width`	`0.4pt`	`l'épaisseur des lignes de construction`
`style`	`solid`	`le style des lignes de construction`
`add`	`.2 and .2`	`modifier la longueur d'un segment de ligne`

37.1.1. Modifier la largeur de la ligne



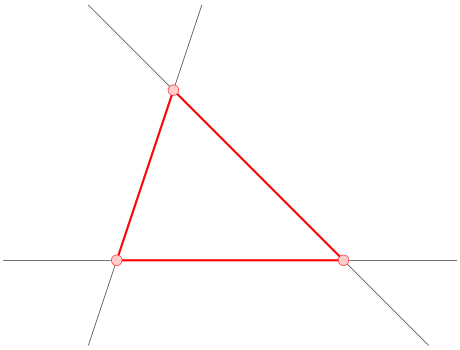

```
\begin{tikzpicture}[scale=.75]
\tkzSetUpLine[line width=1pt]
\begin{scope}[rotate=-90]
    \tkzDefPoints{0/6/A,10/0/B,10/6/C}
    \tkzDefPointBy[projection = onto B--A](C)
    \tkzGetPoint{H}
    \tkzMarkRightAngle[size=.4,
                         fill=teal!20](B,C,A)
    \tkzMarkRightAngle[size=.4,
                         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}
```

37.1.2. Modifier le style de la lignee




```
\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}
```

37.1.3. Exemple 3 : prolonger les lignes



```
\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}
```

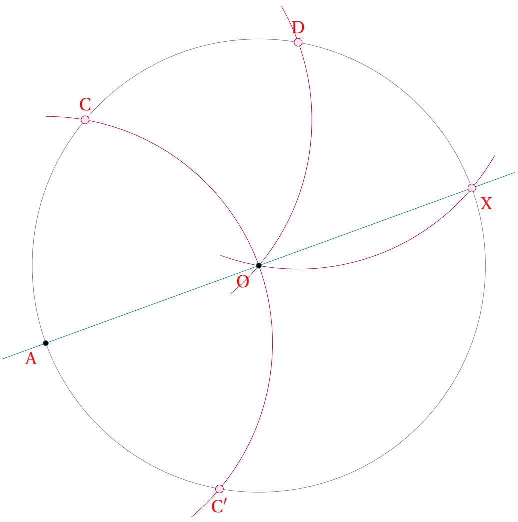
38. Style de l'arc

38.1. The macro \tkzSetUpArc

\tkzSetUpArc[⟨options locales⟩]

options	default	definition
color	black	colour of the lines
line width	0.4pt	thickness of the lines
style	solid	style of construction lines

38.1.1. Utilisation de \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}
```

39. style des traits de compass, macro de configuration \tkzSetUpCompass

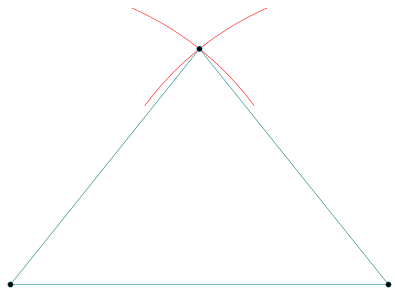
La macro suivante permet de comprendre la construction d'une figure en montrant les tracés au compas nécessaires pour obtenir certains points.

39.1. La macro \tkzSetUpCompass

\tkzSetUpCompass[⟨options locales⟩]

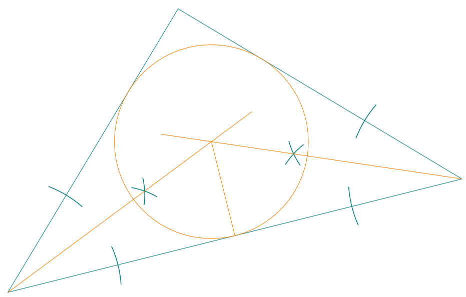
options	défaut	définition
color	black	couleur des lignes de construction
line width	0.4pt	l'épaisseur des lignes de construction
style	solid	style de lignes : solid, dashed,dotted,...
delta	0	modifie la longueur de l'arc

39.1.1. Utilisation de \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}
```

39.1.2. Utilisation de \tkzSetUpCompass avec \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}
```

40. Style de l'étiquette

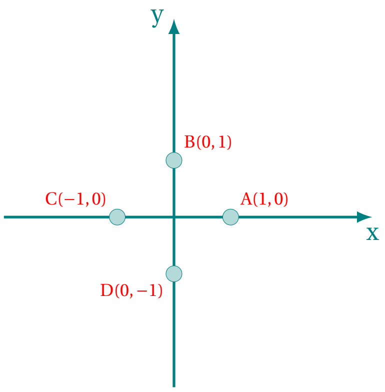
40.1. La macroo \tkzSetUpLabel

La macro `\tkzSetUpLabel` est utilisée pour définir le style des étiquettes de points.

`\tkzSetUpStyle[⟨options locales⟩]`

Les options sont les mêmes que celles de Ti*k*Z

40.1.1. Utilisation de \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}
```

41. Style propre

Vous pouvez définir votre propre style avec `\tkzSetUpStyle`

41.1. La macro \tkzSetUpStyle

`\tkzSetUpStyle[⟨options locales⟩]`

Les options sont les mêmes que celles de Ti*k*Z

41.1.1. Utilisation de \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}
```

42. Comment utiliser arrows

Dans certains pays, les flèches sont utilisées pour indiquer le parallélisme des lignes, pour représenter des demi-droites ou les côtés d'un angle (rayons).
Voici quelques exemples de placement de ces flèches. **tkz-euclide** charge une bibliothèque appelée **arrows.meta**.
`\usetikzlibrary{arrows.meta}`
Cette bibliothèque est utilisée pour produire différents styles de pointes de flèches. Les exemples suivants en utilisent quelques-uns.

42.1. Flèches aux extrémités d'un segment, d'un rayon ou d'une ligne

Stealth, **Triangle**, **To**, **Latex** et …qui peut être combiné avec **reversed**. Il est facile de placer une flèche à une ou deux extrémités.

1. **-Triangle** et **Segment**


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[-Triangle](A,B)
\end{tikzpicture}
```

2. **Stealth-Stealth** et **Segment**


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[Stealth-Stealth](A,B)
\end{tikzpicture}
```

3. **Latex-Latex** et **Line**


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawLine[red,Latex-Latex](A,B)
  \tkzDrawPoints(A,B)
\end{tikzpicture}
```

4. **To-To** et **Segment**


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[To-To](A,B)
\end{tikzpicture}
```

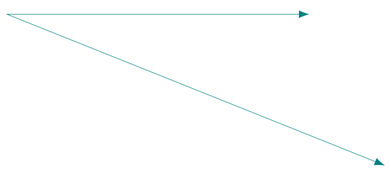
5. **Latex-Latex** et **Segment**


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[Latex-Latex](A,B)
\end{tikzpicture}
```

6. **Latex-** et **Segment**


```
\begin{tikzpicture}
  \tkzDefPoints{0/0/A,4/0/B}
  \tkzDrawSegment[Latex-](A,B)
\end{tikzpicture}
```

7. **-Latex** et **Segments**



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B,5/-2/C}
 \tkzDrawSegments[-Latex](A,B A,C)
\end{tikzpicture}
```

42.1.1. Mise à l'échelle d'une tête de flèche

```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B}
 \tkzDrawSegment[{Latex[scale=2]}-{Latex[scale=2]}](A,B)
\end{tikzpicture}
```

42.1.2. Utilisation d'un style vectoriel

\tikzset{vector style/.style={>=Latex,->}}
Vous pouvez redéfinir ce style.


```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,4/0/B}
\tkzDrawSegment[vector style](A,B)
\end{tikzpicture}
```

42.2. Flèches sur le point central d'un segment de droite

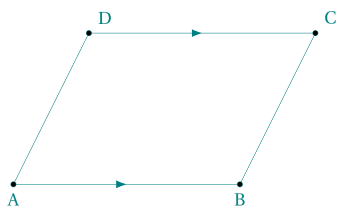
Les flèches sur les droites sont utilisées pour indiquer que ces droites sont parallèles. Cela dépend des pays, en France on préfère indiquer à l'extérieur de la figure que (A, B) ∥ (D, C). Le code est une adaptation d'une réponse de **Muzimuzhi Z** sur le site tex.stackexchange.com.

Syntax :
- tkz arrow (**Latex** par défaut)
- 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**

Exemples d'utilisation :

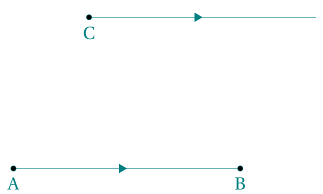
```
\tkzDrawSegment[tkz arrow=Stealth] (A,B)
\tkzDrawSegment[tkz arrow={To[scale=3] at .4}](A,B)
\tkzDrawSegment[tkz arrow={Latex[scale=5,blue] at .6}](A,B)
```

42.2.1. Dans un parallélogramme




```
\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}
```

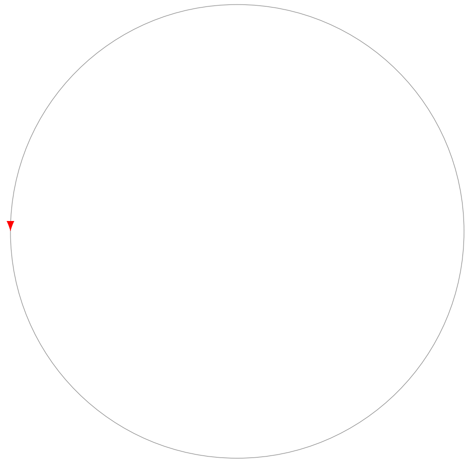
42.2.2. Une ligne parallèle à une autre




```
\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}
```

42.2.3. Flèche sur un cercle

Il est possible de placer une flèche sur le premier quart d'un cercle. Une rotation permet de déplacer la flèche.



```
\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}
```

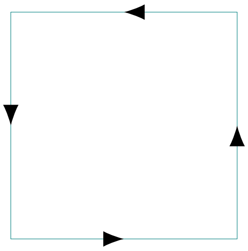
42.3. Flèches sur tous les segments d'un polygone

Certains utilisateurs de mon logiciel m'ont demandé de pouvoir placer une flèche de chaque côté d'un polygone. J'ai utilisé un style proposé par Paul Gaborit sur le site tex.stackexchange.com.

```
\tikzset{tkz arrows/.style=
{postaction={on each path={tkz arrow={Latex[scale=2,color=black]}}}}}
```

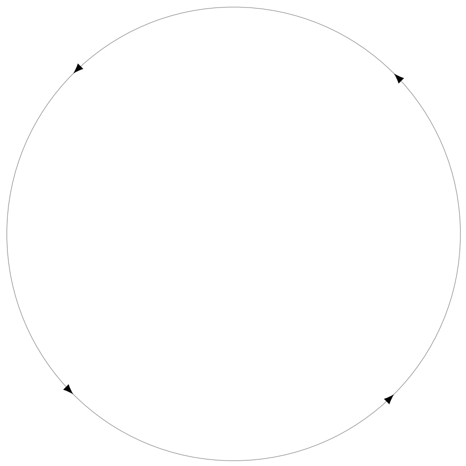
Vous pouvez modifier ce style. Avec `tkz arrows` vous pouvez mettre une flèche sur chaque segment d'un polygone

42.3.1. Flèche sur chaque segment avec tkz arrows



```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,3/0/B}
 \tkzDefSquare(A,B) \tkzGetPoints{C}{D}
 \tkzDrawPolygon[tkz arrows](A,...,D)
\end{tikzpicture}
```

42.3.2. Utiliser **tkz flèches** avec un cercle



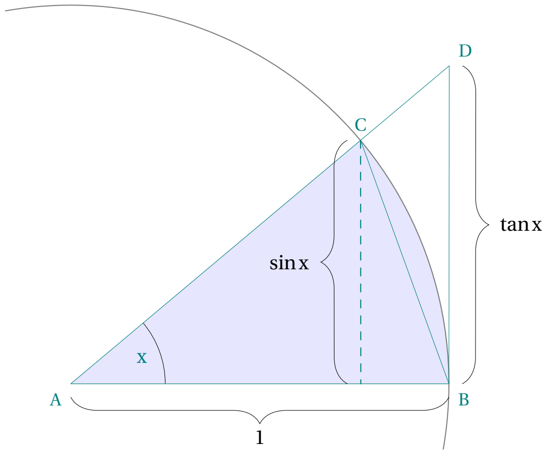
```
\begin{tikzpicture}
 \tkzDefPoints{0/0/A,3/0/B}
 \tkzDrawCircle[tkz arrows](A,B)
\end{tikzpicture}
```

Neuvième partie

Exemples

43. Différents auteurs

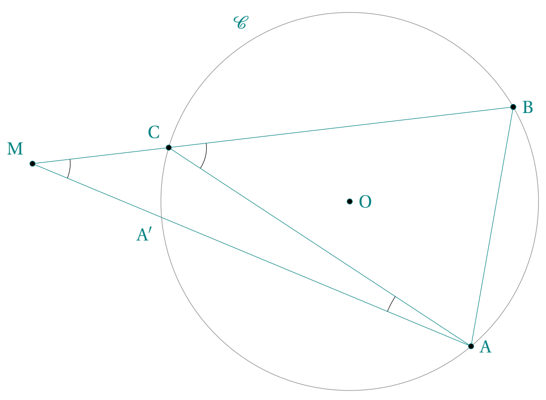
43.1. Code d'Andrew Swan




```
\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}
```

43.2. Exemple: Dimitris Kapeta

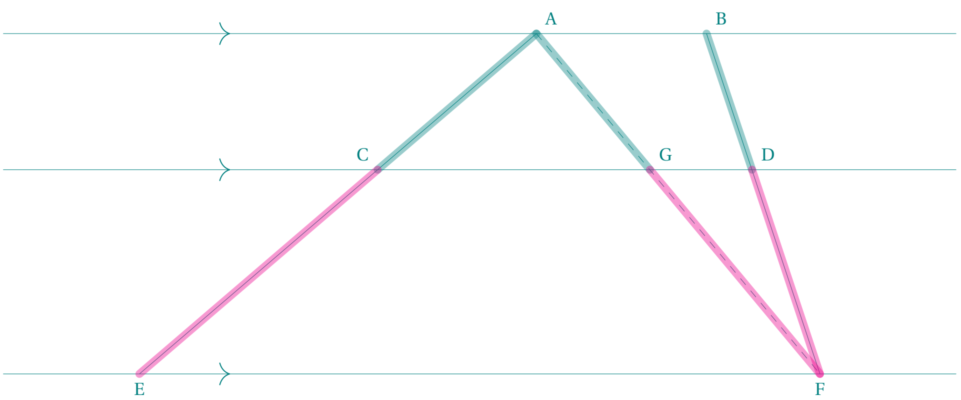
Dans cet exemple, vous devez utiliser `mkpos=.2` avec `\tkzMarkAngle` car la mesure de $\widehat{CAM}$ est trop petite. Une autre possibilité consiste à utiliser `\tkzFillAngle`.




```
\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}
```

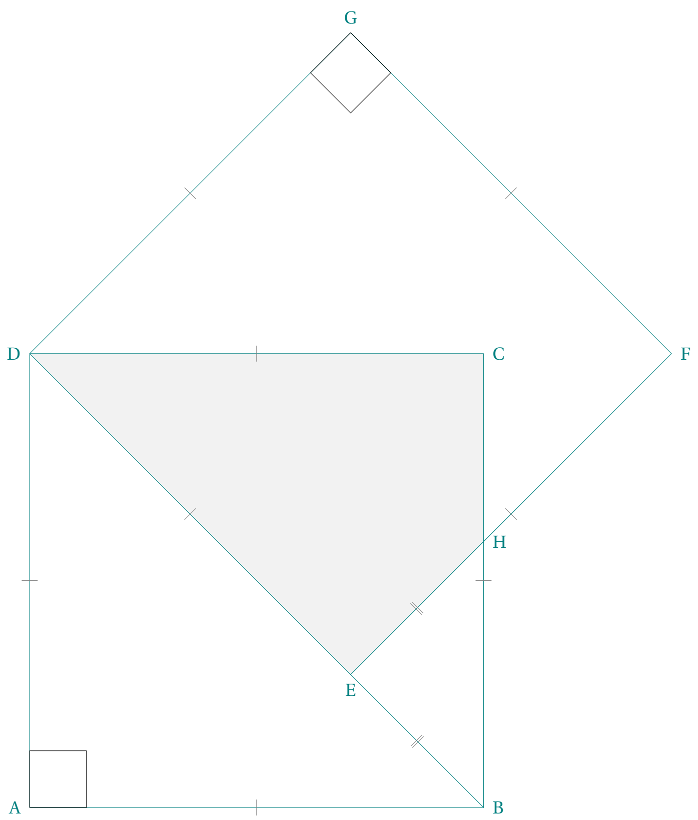
43.3. Exemple : John Kitzmiller

Prouver que $\dfrac{AC}{CE} = \dfrac{BD}{DF}$.
Un autre exemple intéressant de John, vous pouvez voir comment utiliser des options supplémentaires telles que **`decoration`** et **`postaction`** de Ti*k*Z avec **`tkz-euclide`**.



```
\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}
```

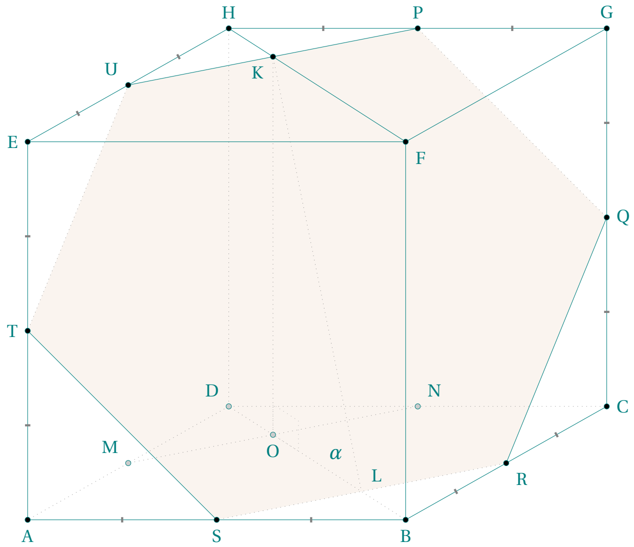
43.4. Exemple 1: Indonesia



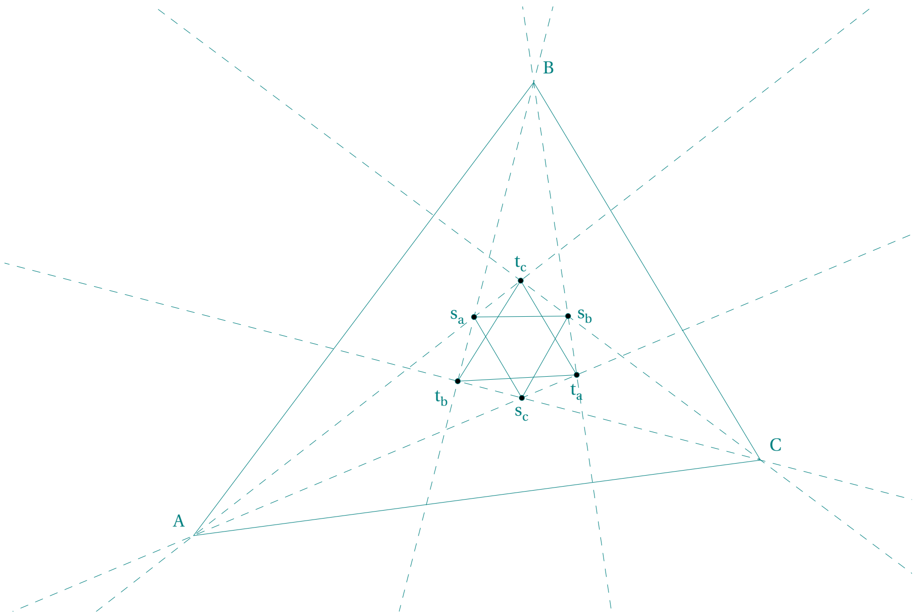
```
\begin{tikzpicture}[scale=3]
   \tkzDefPoints{0/0/A,2/0/B}
   \tkzDefSquare(A,B) \tkzGetPoints{C}{D}
   \tkzDefPointBy[rotation=center D angle 45](C)\tkzGetPoint{G}
   \tkzDefSquare(G,D)\tkzGetPoints{E}{F}
   \tkzInterLL(B,C)(E,F)\tkzGetPoint{H}
   \tkzFillPolygon[gray!10](D,E,H,C,D)
   \tkzDrawPolygon(A,...,D)\tkzDrawPolygon(D,...,G)
   \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}
```

43.5. Exemple 2: 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

43.6. Illustration du théorème de Morley par Nicolas François

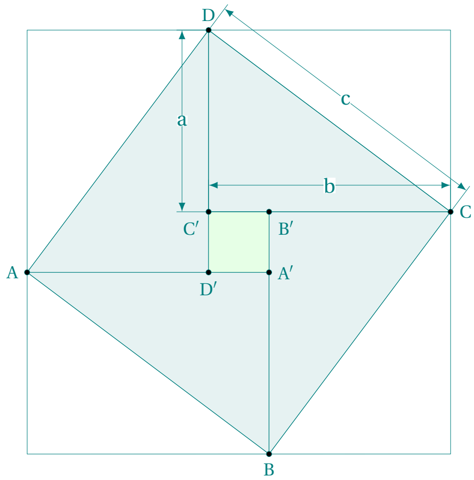



```
\begin{tikzpicture}
  \tkzInit[ymin=-3,ymax=5,xmin=-5,xmax=7]
  \tkzClip
  \tkzDefPoints{-2.5/-2/A,2/4/B,5/-1/C}
  \tkzFindAngle(C,A,B) \tkzGetAngle{anglea}
  \tkzDefPointBy[rotation=center A angle 1*\anglea/3](C) \tkzGetPoint{TA1}
  \tkzDefPointBy[rotation=center A angle 2*\anglea/3](C) \tkzGetPoint{TA2}
  \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}
```

43.7. Gou gu theorem / Théorème de Pythagore par Zhao Shuang

Gou gu theorem / Théorème de Pythagore par Zhao Shuang

Pythagore n'est pas la première personne à avoir découvert ce théorème dans le monde. La Chine ancienne a découvert ce théorème bien avant lui. Le théorème de Pythagore porte donc un autre nom en Chine, le théorème de Gou-Gu. Zhao Shuang était un ancien mathématicien chinois. Il a redécouvert le "Gou gu therorem", qui est en fait la version chinoise du "théorème de Pythagore". Zhao Shuang a utilisé une méthode appelée "principe de coupe et de compensation", il a créé une image of "Pythagorean Round Square" Ci-dessous, la figure utilisée pour illustrer la preuve de l'existence du "Gou gu theorem." (code de Nan Geng)

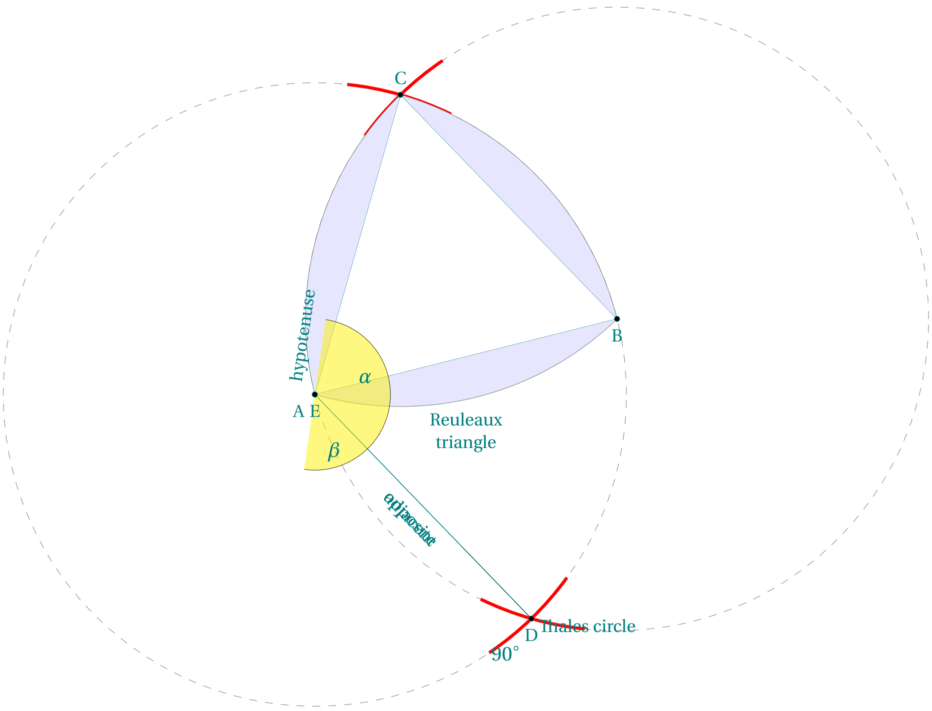



```
\begin{tikzpicture}[scale=.8]
  \tkzDefPoint(0,0){A} \tkzDefPoint(4,0){A'}
  \tkzInterCC[R](A, 5)(A', 3)
  \tkzGetSecondPoint{B}
  \tkzDefSquare(A,B)   \tkzGetPoints{C}{D}
  \tkzCalcLength(A,A') \tkzGetLength{lA}
  \tkzCalcLength(A',B) \tkzGetLength{lB}
  \pgfmathparse{\lA-\lB}
  \tkzInterLC[R](A,A')(A',\pgfmathresult)
  \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}
```

43.8. Reuleaux-Triangle

Reuleaux-triangle par Stefan Kottwitz

La géométrie est un domaine classique bien connu des mathématiques. Vous connaissez peut-être la géométrie euclidienne de l'école, avec ses constructions au compas et à la règle. au compas et à la règle. Les professeurs de mathématiques peuvent être très intéressés par dessiner des constructions géométriques et des explications. Les constructions sous-jacentes peuvent nous aider à réaliser des dessins généraux où nous avons besoin d'intersections et de tangentes de lignes et de cercles, même si cela ne ressemble pas à de la géométrie. Ici, nous nous souviendrons des dessins géométriques de l'école. Nous utiliserons le paquet tkz-euclide, qui fonctionne au-dessus de TikZ. Nous construirons un triangle équilatéral. Puis nous l'étendrons pour obtenir un triangle de Reuleaux, et ajouterons des annotations. Le code est expliqué en détail dans le LaTeX Cookbook., Chapter 10, Advanced Mathematics, Drawing geometry pictures. Stefan Kottwitz



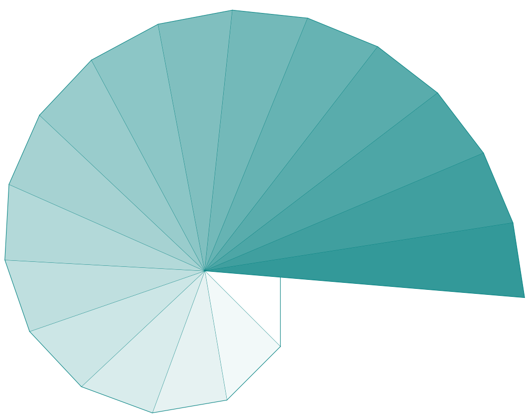
```
\begin{tikzpicture}
  \tkzDefPoint(0,0){A} \tkzDefPoint(4,1){B}
  \tkzInterCC(A,B)(B,A) \tkzGetPoints{C}{D}
  \tkzInterLC(A,B)(B,A) \tkzGetPoints{F}{E}
  \tkzDrawCircles[dashed](A,B B,A)
  \tkzDrawPolygons(A,B,C A,E,D)
  \tkzCompasss[color=red, very thick](A,C B,C A,D B,D)
  \begin{scope}
    \tkzSetUpArc[thick,delta=0]
    \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}
```

44. Quelques exemples intéressants

44.1. Racine carrée des entiers

Racine carrée des entiers

Comment obtenir 1, $\sqrt{2}$, $\sqrt{3}$ *avec une règle et un compas.*

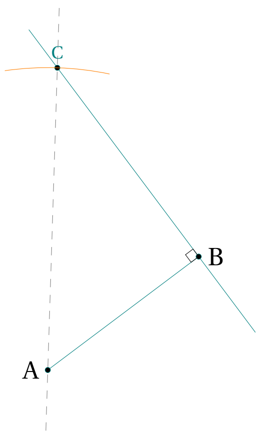


```
\begin{tikzpicture}
  \tkzDefPoint(0,0){O}
  \tkzDefPoint(1,0){a0}
   \tkzDrawSegment(O,a0)
  \foreach \i [count=\j] in {0,...,16}{%
    \tkzDefPointWith[orthogonal normed](a\i,O)
    \tkzGetPoint{a\j}
       \pgfmathsetmacro{\c}{5*\i}
    \tkzDrawPolySeg[fill=teal!\c](a\i,a\j,O)}
 \end{tikzpicture}
```

44.2. A propos du triangle rectangle

A propos du triangle rectangle

On a un segment [AB] *et on veut déterminer un point* C *tel que* AC = 8 *cm et que* ABC *est un triangle rectangle dans* B.

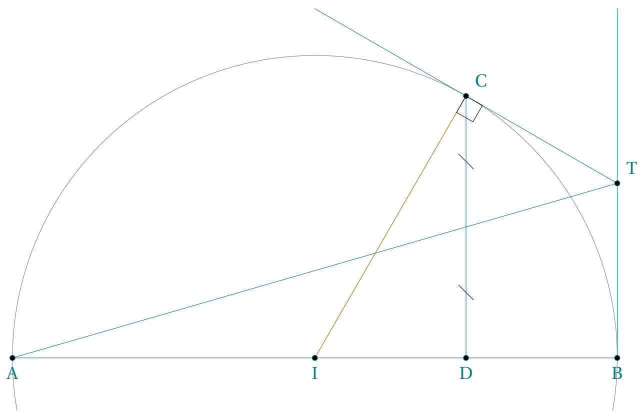



```
\begin{tikzpicture}[scale=.5]
  \tkzDefPoint["$A$" left](2,1){A}
  \tkzDefPoint["$B$" right](6,4){B}
  \tkzDefPointWith[orthogonal,K=-1](B,A)
  \tkzDrawLine[add = .5 and .5](B,tkzPointResult)
  \tkzInterLC[R](B,tkzPointResult)(A,8)
  \tkzGetPoints{J}{C}
  \tkzDrawSegment(A,B)
  \tkzDrawPoints(A,B,C)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

44.3. Archimède

Archimède

Il s'agit d'un problème ancien prouvé par le grand mathématicien grec Archimède. La figure ci-dessous représente un demi-cercle de diamètre AB. *Une ligne tangente est tracée et touche le demi-cercle en* B. *Une autre ligne tangente est tracée en un point* C *du demi-cercle. On projette le point* C *du segment de droite* [AB] *sur un point* D. *Les deux droites tangentes se coupent au point* T. *Prouver que la droite* (AT) *est bissectrice de* (CD)

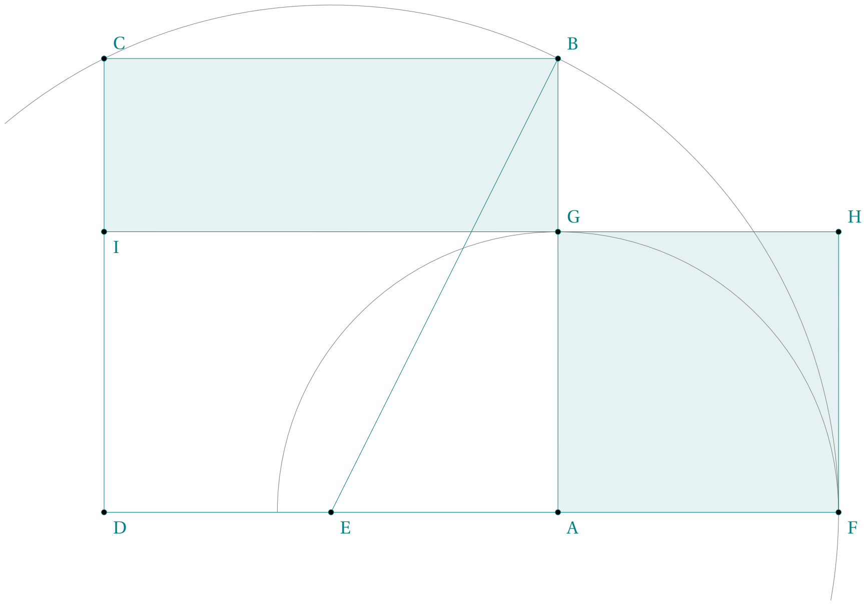


```
\begin{tikzpicture}[scale=1]
  \tkzDefPoint(0,0){A}\tkzDefPoint(6,0){D}
  \tkzDefPoint(8,0){B}\tkzDefPoint(4,0){I}
  \tkzDefLine[orthogonal=through D](A,D)
  \tkzInterLC[R](D,tkzPointResult)(I,4) \tkzGetSecondPoint{C}
  \tkzDefLine[orthogonal=through C](I,C)    \tkzGetPoint{c}
  \tkzDefLine[orthogonal=through 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}
```

44.3.1. Carré et rectangle de même aire ; nombre d'or

Livre II, proposition XI _Les éléments d'Euclide_

Construire un carré et un rectangle de même surface.

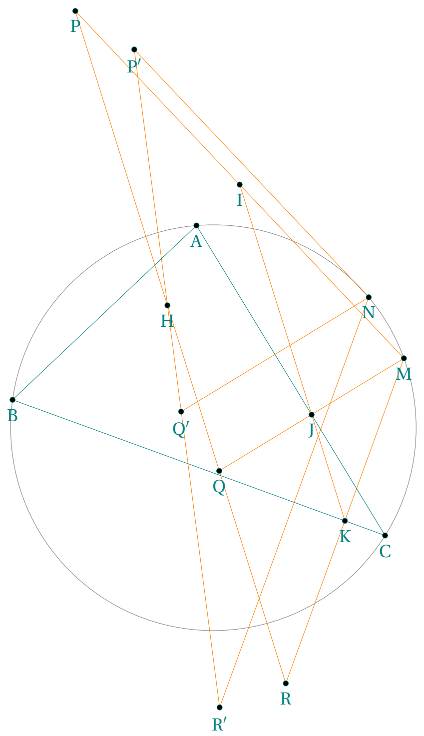



```
\begin{tikzpicture}[scale=.75]
 \tkzDefPoint(0,0){D} \tkzDefPoint(8,0){A}
 \tkzDefSquare(D,A) \tkzGetPoints{B}{C}
 \tkzDefMidPoint(D,A) \tkzGetPoint{E}
 \tkzInterLC(D,A)(E,B)\tkzGetSecondPoint{F}
 \tkzInterLC[near](B,A)(A,F)\tkzGetFirstPoint{G}
 \tkzDefSquare(A,F)\tkzGetFirstPoint{H}
 \tkzInterLL(C,D)(H,G)\tkzGetPoint{I}
 \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}
```

44.3.2. Droite de Steiner et droite Simson

Droite de Steiner et droite Simson

Considérons le triangle ABC et un point M sur son cercle. Les projections de M sur les côtés du triangle sont sur une ligne (ligne de Steiner). Les trois points les plus proches de M sur les lignes AB, AC et BC sont colinéaires. C'est la ligne de Simson.

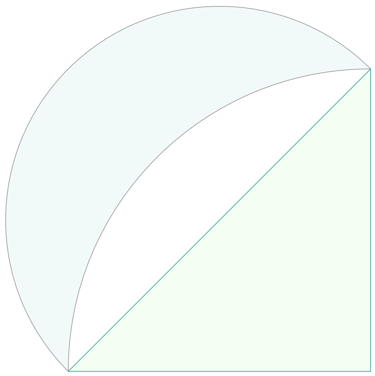



```
\begin{tikzpicture}[scale=.75,rotate=-20]
  \tkzDefPoint(0,0){B}
  \tkzDefPoint(2,4){A} \tkzDefPoint(7,0){C}
  \tkzDefCircle[circum](A,B,C)
  \tkzGetPoint{O}
  \tkzDrawCircle(O,A)
  \tkzCalcLength(O,A)
  \tkzGetLength{rOA}
  \tkzDefShiftPoint[O](40:\rOA){M}
  \tkzDefShiftPoint[O](60:\rOA){N}
  \tkzDefTriangleCenter[orthic](A,B,C)
  \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}
```

44.4. Lune d'Hippocrate

Lune d'Hippocrate

D'après wikipedia : En géométrie, la lune d'Hippocrate, nommée d'après Hippocrate de Chios, est une lune délimitée par les arcs de deux cercles dont le plus petit a pour diamètre une corde formant un angle droit sur le plus grand cercle.Dans la première figure, l'aire de la lune est égale à l'aire du triangle ABC. Hippocrate de Chios (mathématicien de la Grèce antique)

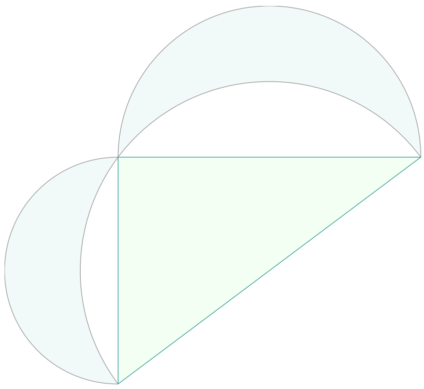


```
\begin{tikzpicture}
 \tkzInit[xmin=-2,xmax=5,ymin=-1,ymax=6]
 \tkzClip % allows you to define a bounding box
   % large enough
  \tkzDefPoint(0,0){A}\tkzDefPoint(4,0){B}
  \tkzDefSquare(A,B)
  \tkzGetFirstPoint{C}
  \tkzDrawPolygon[fill=green!5](A,B,C)
   \begin{scope}
     \tkzClipCircle[out](B,A)
     \tkzDefMidPoint(C,A) \tkzGetPoint{M}
     \tkzDrawSemiCircle[fill=teal!5](M,C)
   \end{scope}
   \tkzDrawArc[delta=0](B,C)(A)
\end{tikzpicture}
```

44.5. Lunes de Hasan Ibn al-Haytham

Lunes de Hasan Ibn al-Haytham

D'après wikipedia : le mathématicien arabe Hasan Ibn al-Haytham (nom latinisé Alhazen) a montré que deux lunes, formées sur les deux côtés d'un triangle rectangle, dont les limites extérieures sont des demi-cercles et dont les limites intérieures sont formées par le cercle du triangle, alors les aires de ces deux lunes additionnées sont égales à l'aire du triangle. Les lunes ainsi formées à partir d'un triangle rectangle sont appelées lunes d'Alhazen.

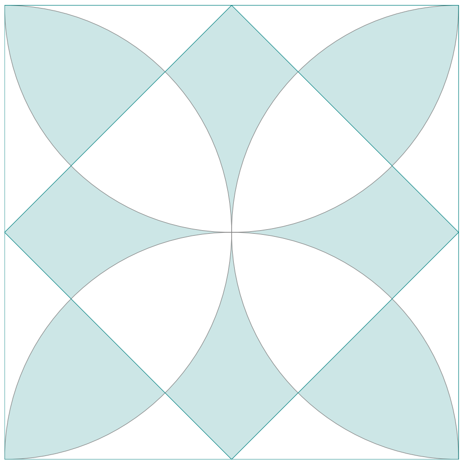


```
\begin{tikzpicture}[scale=.5,rotate=180]
  \tkzInit[xmin=-1,xmax=11,ymin=-4,ymax=7]
  \tkzClip
  \tkzDefPoints{0/0/A,8/0/B}
  \tkzDefTriangle[pythagore,swap](A,B)
  \tkzGetPoint{C}
  \tkzDrawPolygon[fill=green!5](A,B,C)
  \tkzDefMidPoint(C,A) \tkzGetPoint{I}
  \begin{scope}
    \tkzClipCircle[out](I,A)
    \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}
```

44.6. À propos des cercles de découpe

cercles de découpe

Le problème est la gestion de la boîte de délimitation. Il faut d'abord définir un rectangle dans lequel la figure sera insérée. Cela se fait avec les deux premières lignes.



```
\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}
```

44.7. Triangles isocèles semblables

Triangles isocèles semblables

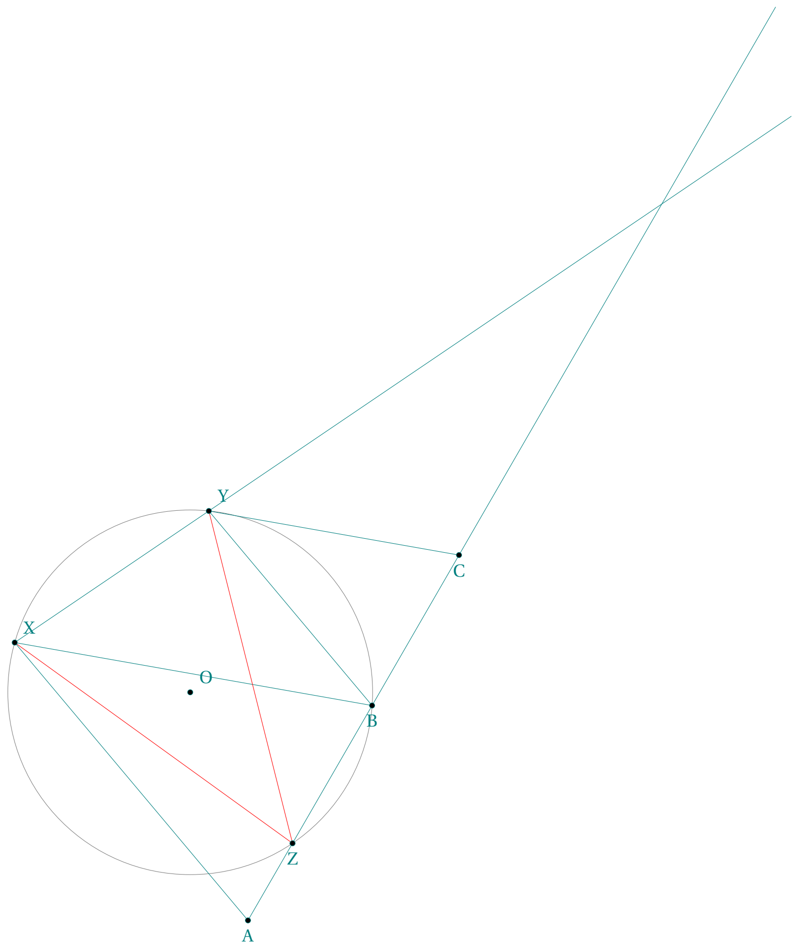
*Ce qui suit est tiré de l'excellent site **Descartes et les Mathématiques**. Je n'ai pas modifié le texte et je ne suis que l'auteur de la programmation des figures.* http://debart.pagesperso-orange.fr/seconde/triangle.html

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
Bibliographie :

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

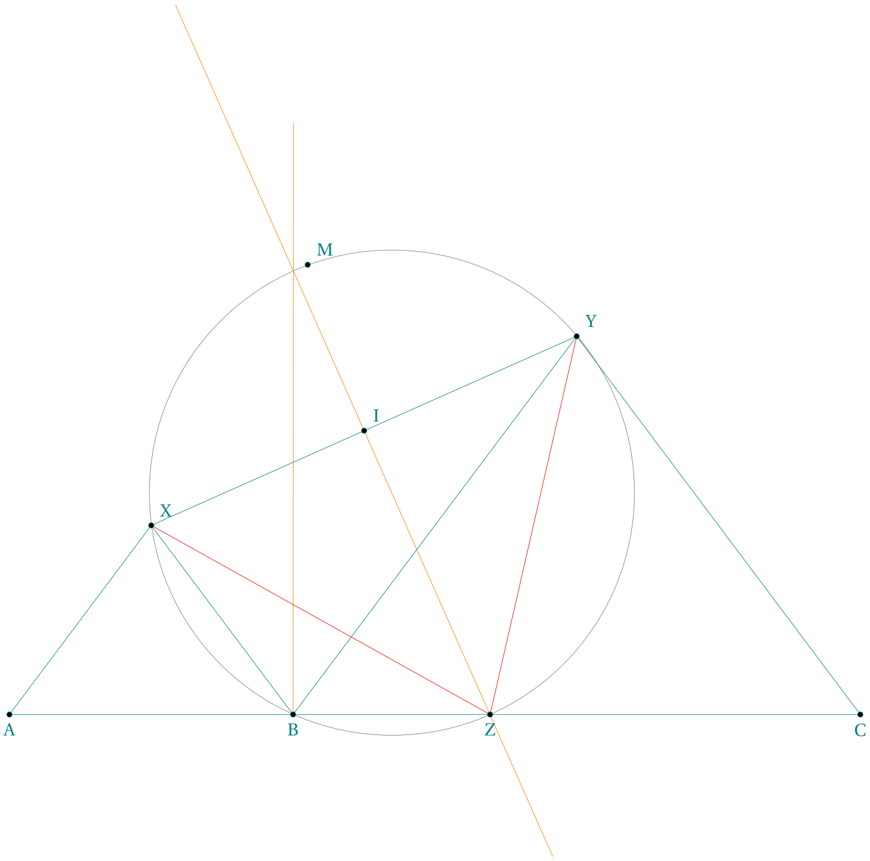
Deux énoncés ont été proposés, l'un par le magazine *Tangente* et l'autre par *Le Monde*.
Rédacteur du magazine "Tangente" : Deux triangles isocèles similaires AXB et BYC sont construits avec des sommets principaux X et Y, tels que A, B et C sont alignés et que ces triangles sont "indirects". Soit α l'angle au sommet $\widehat{AXB} = \widehat{BYC}$. Nous construisons ensuite un troisième triangle isocèle XZY similaire aux deux premiers, avec un sommet principal Z et "indirect". Nous demandons de démontrer que le point Z appartient à la droite (AC).
Rédacteur du "Le Monde" : Nous construisons deux triangles isocèles similaires AXB et BYC avec des sommets principaux X et Y, tels que A, B et C sont alignés et que ces triangles sont "indirects". Soit α l'angle au sommet $\widehat{AXB} = \widehat{BYC}$. Le point Z du segment de droite [AC] est à égale distance des deux sommets X et Y.
À quel angle voit-il ces deux sommets ?
Les constructions et leurs codes associés se trouvent sur les deux pages suivantes, mais vous pouvez chercher avant de regarder. La programmation respecte (il me semble ...) mon raisonnement dans les deux cas.

44.8. Version révisée de "Tangente"




```
\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}
```

44.9. "Le Monde" version

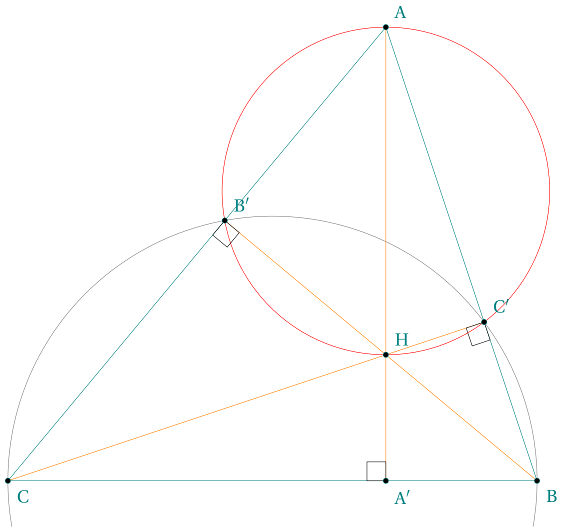



```
\begin{tikzpicture}[scale=1.25]
  \tkzDefPoint(0,0){A}
  \tkzDefPoint(3,0){B}
  \tkzDefPoint(9,0){C}
  \tkzDefPoint(1.5,2){X}
  \tkzDefPoint(6,4){Y}
  \tkzDefCircle[circum](X,Y,B) \tkzGetPoint{O}
  \tkzDefMidPoint(X,Y)               \tkzGetPoint{I}
  \tkzDefPointWith[orthogonal](I,Y)  \tkzGetPoint{i}
  \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}
```

44.10. Hauteurs du triangle

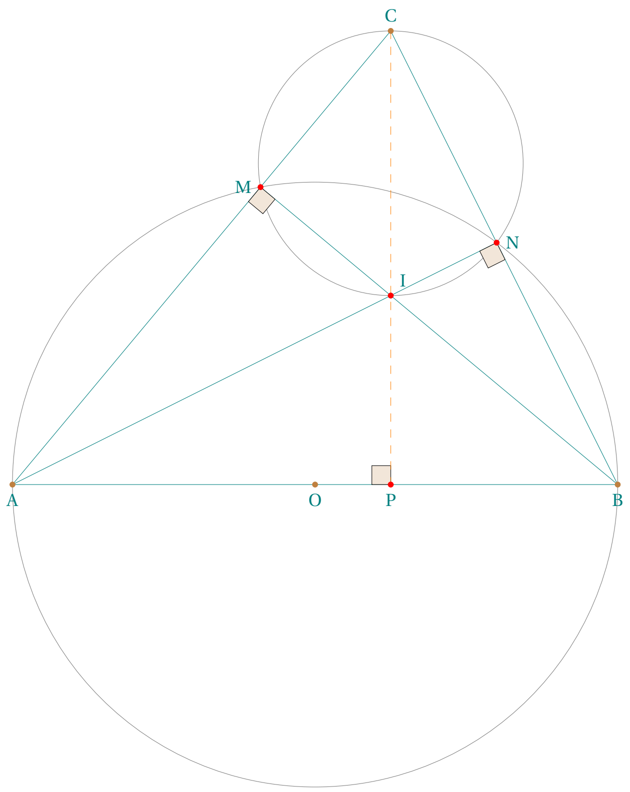
Hauteurs du triangle

De Wikipedia : Ce qui suit provient à nouveau de l'excellent site ***Descartes et les Mathématiques****.* `http://debart.pagesperso-orange.fr/geoplan/geometrie_triangle.html`*. Les trois hauteurs d'un triangle se croisent au même point 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}
```

44.11. Hauteurs - autres constructions

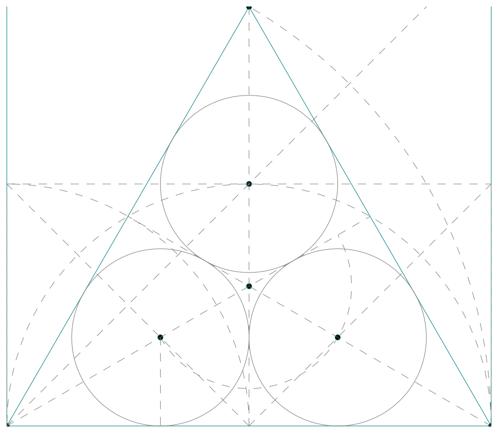



```
\begin{tikzpicture}
\tkzDefPoint(0,0){A} \tkzDefPoint(8,0){B}
\tkzDefPoint(5,6){C}
\tkzDefMidPoint(A,B)\tkzGetPoint{O}
\tkzDefPointBy[projection=onto A--B](C) \tkzGetPoint{P}
\tkzInterLC[common=A](C,A)(O,A)
\tkzGetFirstPoint{M}
\tkzInterLC(C,B)(O,A)
\tkzGetSecondPoint{N}
\tkzInterLL(B,M)(A,N)\tkzGetPoint{I}
\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}
```

44.12. Trois cercles dans un triangle équilatéral

Trois cercles dans un triangle équilatéral

D'après Wikipédia : En géométrie, les cercles de Malfatti sont trois cercles à l'intérieur d'un triangle donné tels que chaque cercle est tangent aux deux autres et à deux côtés du triangle. Ils doivent leur nom à Gian Francesco Malfatti, qui a étudié le problème de la construction de ces cercles en croyant à tort qu'ils auraient la plus grande aire totale possible parmi trois cercles disjoints à l'intérieur du triangle. Vous trouverez ci-dessous une étude d'un cas particulier avec un triangle équilatéral et trois cercles identiques.

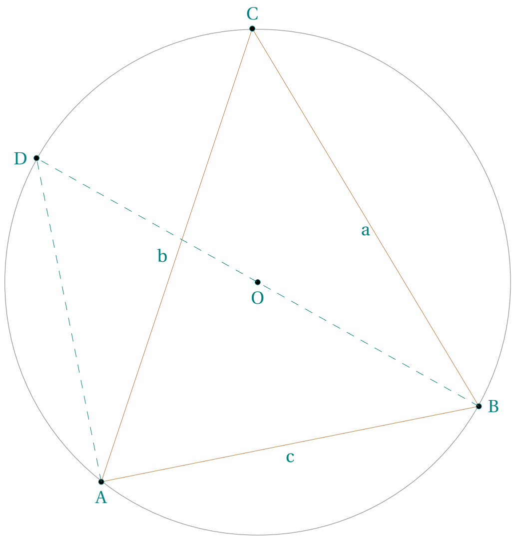


```
\begin{tikzpicture}[scale=.8]
  \tkzDefPoints{0/0/A,8/0/B,0/4/a,8/4/b,8/8/c}
  \tkzDefTriangle[equilateral](A,B) \tkzGetPoint{C}
  \tkzDefMidPoint(A,B) \tkzGetPoint{M}
  \tkzDefMidPoint(B,C) \tkzGetPoint{N}
  \tkzDefMidPoint(A,C) \tkzGetPoint{P}
  \tkzInterLL(A,N)(M,a) \tkzGetPoint{Ia}
  \tkzDefPointBy[projection = 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}
```

44.13. Loi des sinus

Loi des sinus

D'après wikipedia : *En trigonométrie, la loi des sinus, la loi des sinus, la formule des sinus ou la règle des sinus est une équation reliant les longueurs des côtés d'un triangle (n'importe quelle forme) aux sinus de ses angles.*




```
\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}
```

Dans le triangle ABC

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} \tag{1}$$

$$\widehat{C} = \widehat{D}$$

$$\frac{c}{2R} = \sin D = \sin C \tag{2}$$

Dans ce cas

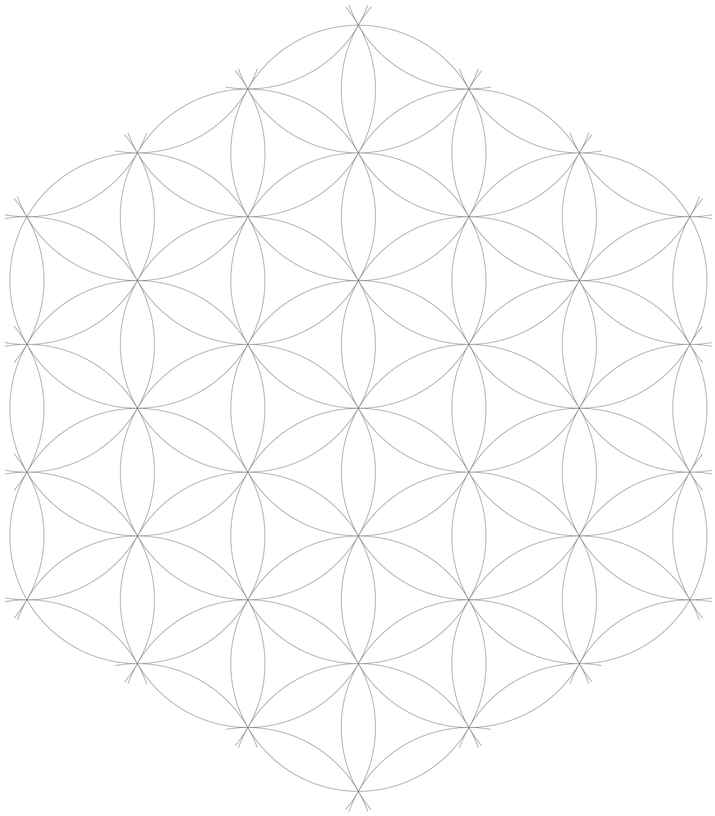
$$\frac{c}{\sin C} = 2R$$

44.14. Fleur de vie

Livre IV, proposition XI _Les éléments d'Euclide_

La géométrie sacrée peut être décrite comme un système de croyances attribuant une valeur religieuse ou culturelle à de nombreuses formes fondamentales de l'espace et du temps. Selon ce système de croyances, les modèles de base de l'existence sont perçus comme sacrés parce qu'en les contemplant, on contemple l'origine de toutes choses. En étudiant la nature de ces formes et leurs relations mutuelles, on peut chercher à comprendre les lois scientifiques, philosophiques, psychologiques, esthétiques et mystiques de l'univers. La Fleur de vie est considérée comme un symbole de géométrie sacrée, dont on dit qu'elle contient des valeurs religieuses anciennes décrivant les formes fondamentales de l'espace et du temps. En ce sens, elle est l'expression visuelle des liens que la vie tisse à travers toute l'humanité, et certains pensent qu'elle contient une sorte d'enregistrement akashique des informations de base de tous les êtres vivants.

L'un des magnifiques arrangements de cercles découverts dans le temple d'Osiris à Abydos, en Égypte (Rawles 1997).
Weisstein, Eric W. "Flower of Life." D'Après MathWorld–A Wolfram Web Resource.
http://mathworld.wolfram.com/FlowerofLife.html

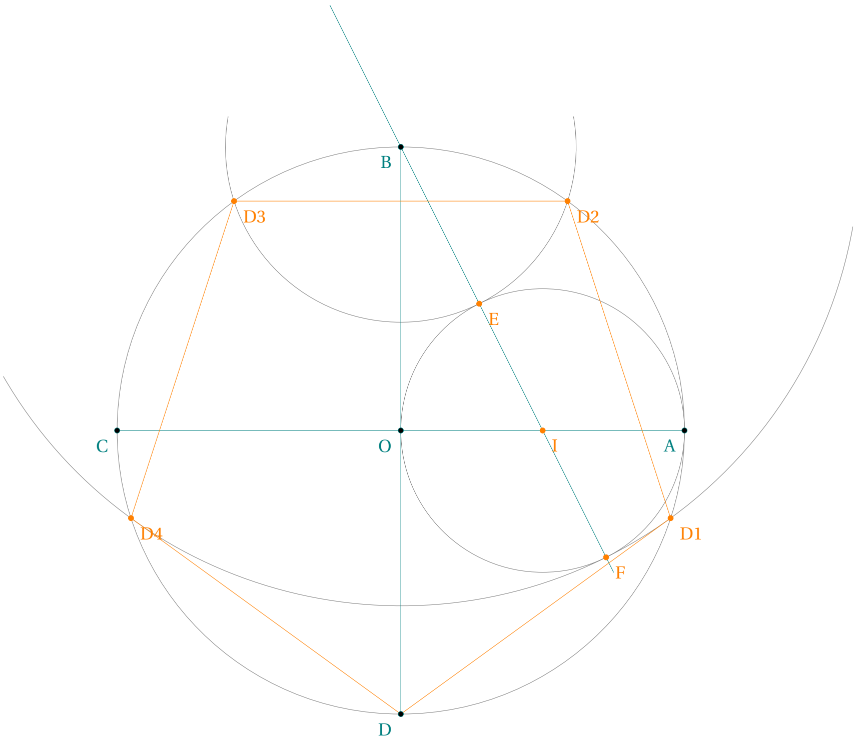


```
\begin{tikzpicture}[scale=.75]
  \tkzSetUpLine[line width=2pt,color=teal!80!black]
  \tkzSetUpCompass[line width=2pt,color=teal!80!black]
   \tkzDefPoint(0,0){O}  \tkzDefPoint(2.25,0){A}
   \tkzDrawCircle(O,A)
\foreach \i in {0,...,5}{
   \tkzDefPointBy[rotation= center O angle 30+60*\i](A)\tkzGetPoint{a\i}
   \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}
```

44.15. Pentagone en cercle

Livre IV, proposition XI _Les éléments d'Euclide_

Inscrire un pentagone équilatéral et équiangulaire dans un cercle donné.

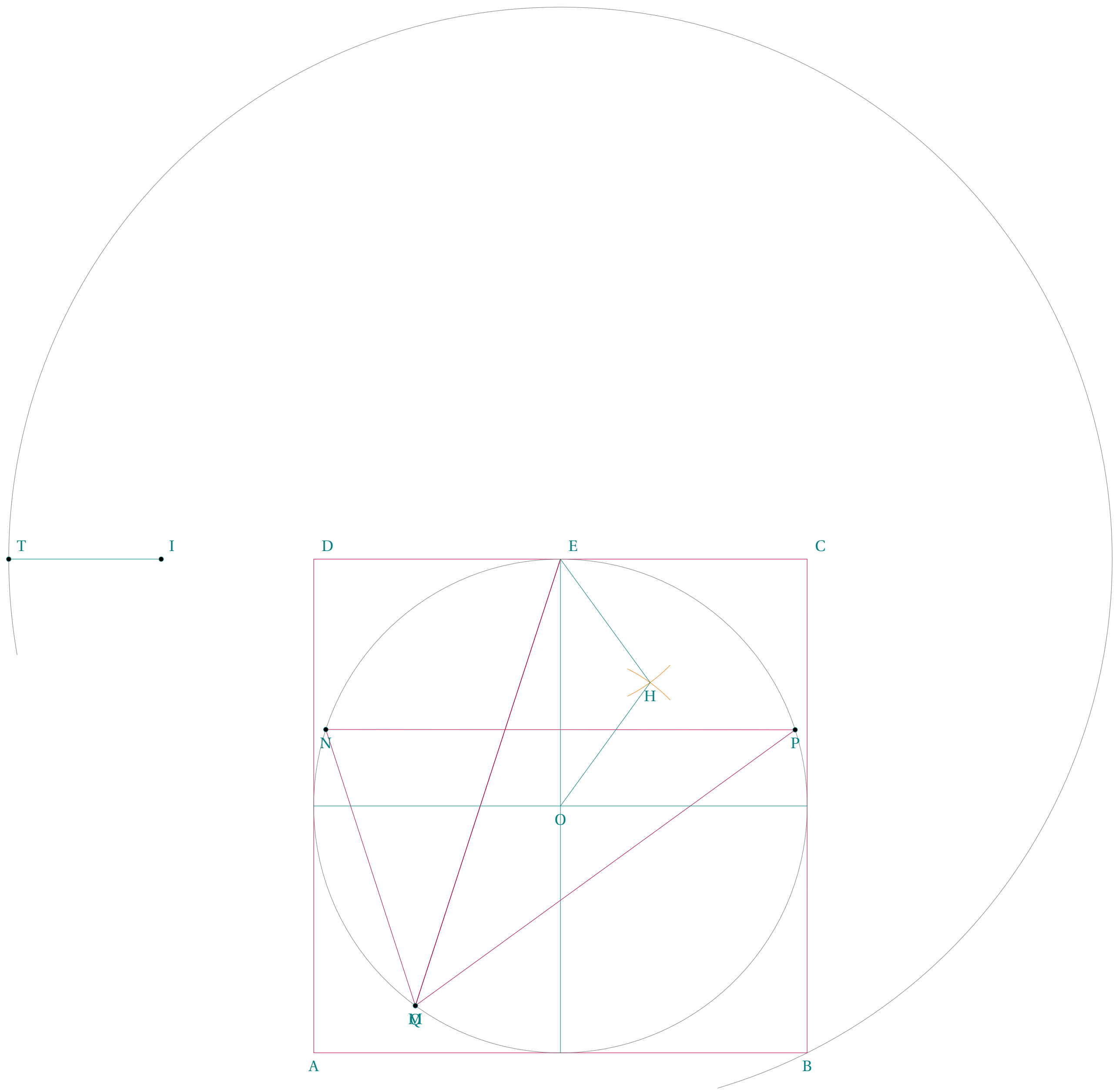



```
\begin{tikzpicture}[scale=.75]
   \tkzDefPoint(0,0){O}
   \tkzDefPoint(5,0){A}
   \tkzDefPoint(0,5){B}
   \tkzDefPoint(-5,0){C}
   \tkzDefPoint(0,-5){D}
   \tkzDefMidPoint(A,O)             \tkzGetPoint{I}
   \tkzInterLC(I,B)(I,A)            \tkzGetPoints{F}{E}
   \tkzInterCC(O,C)(B,E)            \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}
```

44.16. Pentagone dans un carré

Pentagone dans un carré

: *Inscrire un pentagone équilatéral et équiangulaire dans un carré donné.*



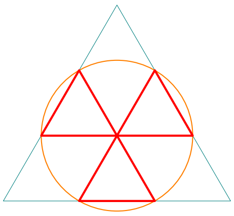
```
\begin{tikzpicture}[scale=.75]
  \tkzDefPoints{0/0/O,-5/-5/A,5/-5/B}
  \tkzDefSquare(A,B)   \tkzGetPoints{C}{D}
  \tkzDefMidPoint(A,B) \tkzGetPoint{F}
  \tkzDefMidPoint(C,D) \tkzGetPoint{E}
  \tkzDefMidPoint(B,C) \tkzGetPoint{G}
  \tkzDefMidPoint(A,D) \tkzGetPoint{K}
  \tkzInterLC(D,C)(E,B)                  \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}
```

44.17. Hexagone Inscrit

Hexagone Inscrit

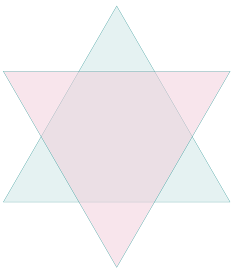
Inscrire un hexagone régulier dans un triangle équilatéral donné, parfaitement à l'intérieur de celui-ci (sans bordures).

44.17.1. Hexagone Inscrit version 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}
```

44.17.2. Hexagone Inscrit version 2

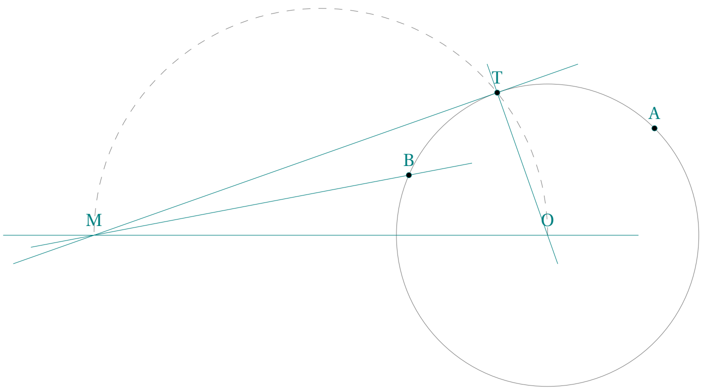


```
\begin{tikzpicture}[scale=.5]
 \pgfmathsetmacro{\c}{6}
 \tkzDefPoints{0/0/A,\c/0/B}
 \tkzDefTriangle[equilateral](A,B)\tkzGetPoint{C}
 \tkzDefTriangleCenter[centroid](A,B,C)
 \tkzGetPoint{I}
 \tkzDefPointsBy[rotation= center I%
                angle 60](A,B,C){a,b,c}
 \tkzDrawPolygon[fill=teal!20,opacity=.5](A,B,C)
 \tkzDrawPolygon[fill=purple!20,opacity=.5](a,b,c)
\end{tikzpicture}
```

44.18. PPuissance d'un point par rapport à un cercle

Puissance d'un point par rapport à un cercle

$\overline{MA} \times \overline{MB} = MT^2 = MO^2 - OT^2$

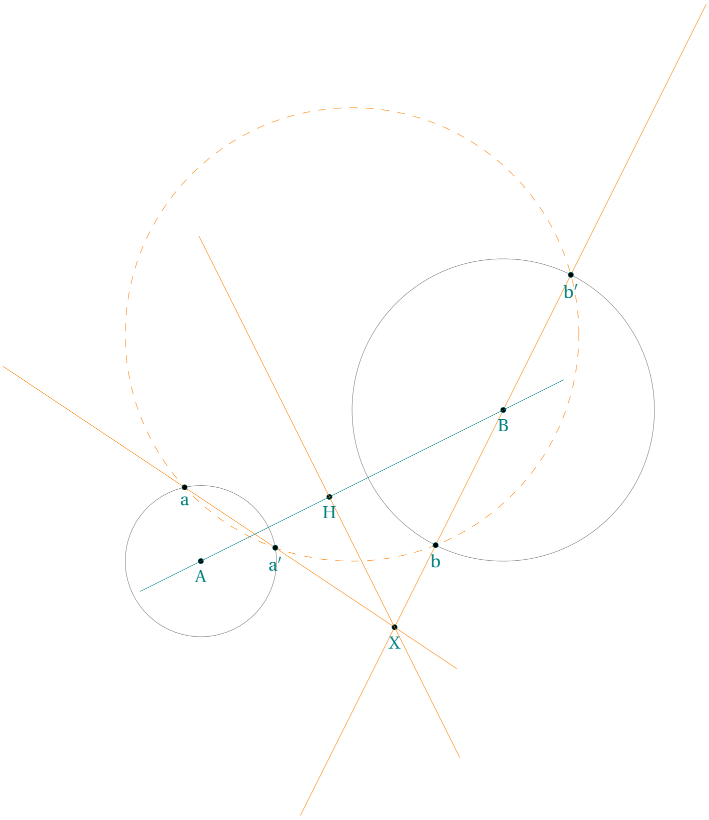



```
\begin{tikzpicture}
 \pgfmathsetmacro{\r}{2}%
 \pgfmathsetmacro{\xO}{6}%
 \pgfmathsetmacro{\xE}{\xO-\r}%
 \tkzDefPoints{0/0/M,\xO/0/O,\xE/0/E}
 \tkzDefCircle[diameter](M,O)
 \tkzGetPoint{I}
 \tkzInterCC(I,O)(O,E) \tkzGetPoints{T}{T'}
 \tkzDefShiftPoint[O](45:2){B}
 \tkzInterLC(M,B)(O,E) \tkzGetPoints{A}{B}
 \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}
```

44.19. Axe radical de deux cercles non concentriques

Axe radical de deux cercles non concentriques

D'Après Wikipédia : *En géométrie, l'axe radical de deux cercles non concentriques est l'ensemble des points dont les puissances par rapport aux cercles sont égales. Pour cette raison, l'axe radical est également appelé ligne de puissance ou bissectrice de puissance des deux cercles. La notation axe radical a été utilisée par le mathématicien français M. Chasles comme axe radical.*

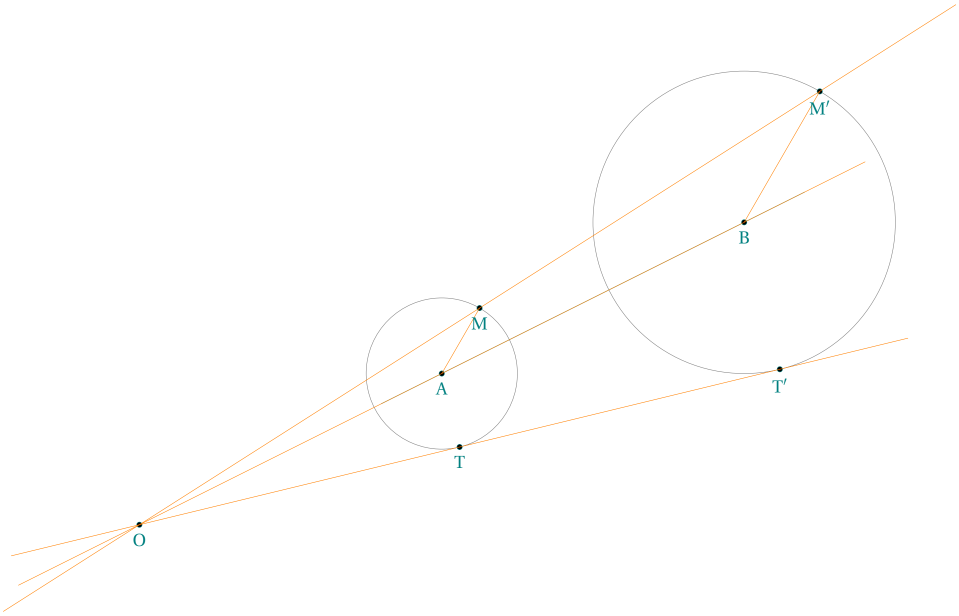



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B,2/3/K}
\tkzDefCircle[R](A,1)\tkzGetPoint{a}
\tkzDefCircle[R](B,2)\tkzGetPoint{b}
\tkzDefCircle[R](K,3)\tkzGetPoint{k}
\tkzDrawCircles(A,a B,b)
\tkzDrawCircle[dashed,new](K,k)
\tkzInterCC(A,a)(K,k) \tkzGetPoints{a}{a'}
\tkzInterCC(B,b)(K,k) \tkzGetPoints{b}{b'}
\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}
```

44.20. Centre homothétique externe

Centre homothétique externe

D'Après Wikipedia : *Étant donné deux cercles non concentriques, tracez des rayons parallèles et dans la même direction. La ligne joignant les extrémités des rayons passe alors par un point fixe de la ligne des centres qui divise cette ligne extérieurement dans le rapport des rayons. Ce point est appelé centre homothétique externe, ou centre externe de similitude (Johnson 1929, pp. 19-20 et 41).*

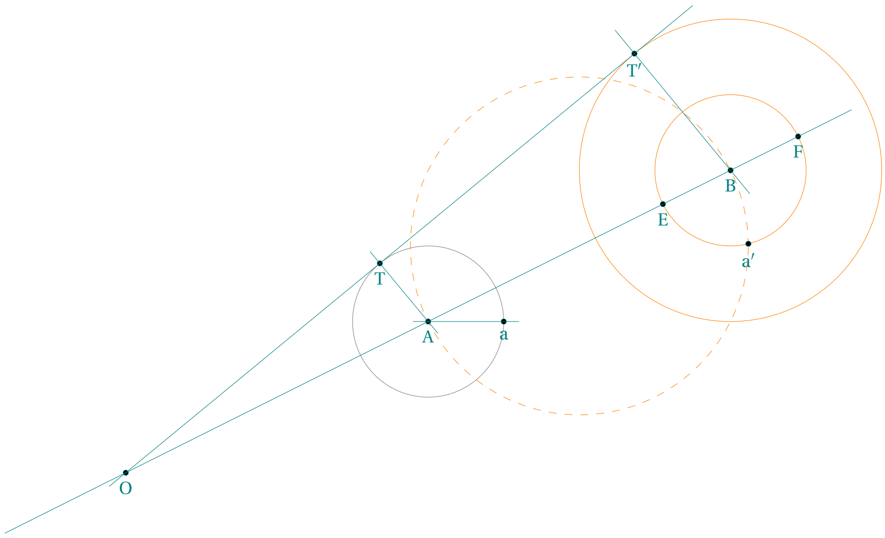



```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B,2/3/K}
\tkzDefCircle[R](A,1)\tkzGetPoint{a}
\tkzDefCircle[R](B,2)\tkzGetPoint{b}
\tkzDrawCircles(A,a B,b)
\tkzDrawLine(A,B)
\tkzDefShiftPoint[A](60:1){M}
\tkzDefShiftPoint[B](60:2){M'}
\tkzInterLL(A,B)(M,M') \tkzGetPoint{O}
\tkzDefLine[tangent from = 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}
```

44.21. Tangentes à deux cercles

Tangentes à deux cercles

FPour deux cercles, il existe généralement quatre lignes distinctes qui sont tangentes aux deux si les deux cercles sont extérieurs l'un à l'autre. Pour deux d'entre elles, les lignes tangentes externes, les cercles tombent du même côté de la ligne; les lignes tangentes externes se croisent au centre homothétique externe

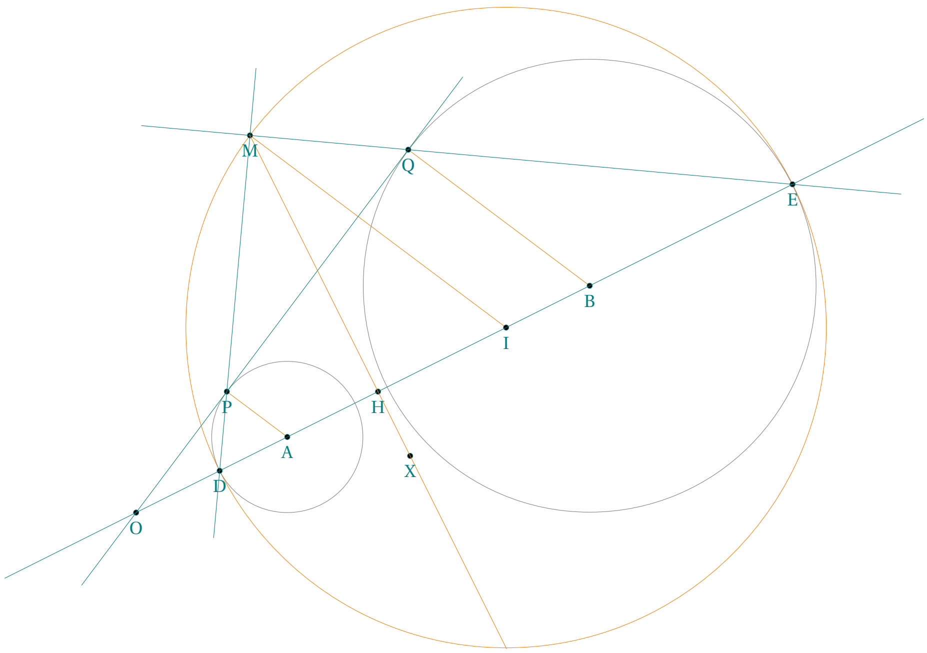



```
\begin{tikzpicture}
 \pgfmathsetmacro{\r}{1}%
 \pgfmathsetmacro{\R}{2}%
 \pgfmathsetmacro{\rt}{\R-\r}%
 \tkzDefPoints{0/0/A,4/2/B,2/3/K}
 \tkzDefMidPoint(A,B) \tkzGetPoint{I}
 \tkzInterLC[R](A,B)(B,\rt) \tkzGetPoints{E}{F}
 \tkzInterCC(I,B)(B,F) \tkzGetPoints{a}{a'}
 \tkzInterLC[R](B,a)(B,\R) \tkzGetPoints{X'}{T'}
 \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}
```

44.22. Tangentes à deux cercles à axe radical

Tangentes à deux cercles à axe radical

Dès que deux cercles ne sont pas concentriques, on peut construire leur axe radical, l'ensemble des points de même puissance par rapport aux deux cercles. On sait que l'axe radical est une droite orthogonale à la droite des centres. Notons que si l'on désigne P *et* Q *comme les points de contact d'une des tangentes extérieures communes aux deux cercles et* D *et* E *comme les points des cercles extérieurs à [AB], alors (DP) et (EQ) se coupent sur l'axe radical des deux cercles. Nous montrerons que cette propriété est toujours vraie et qu'elle permet de construire des tangentes communes, même lorsque les cercles ont le même rayon.*

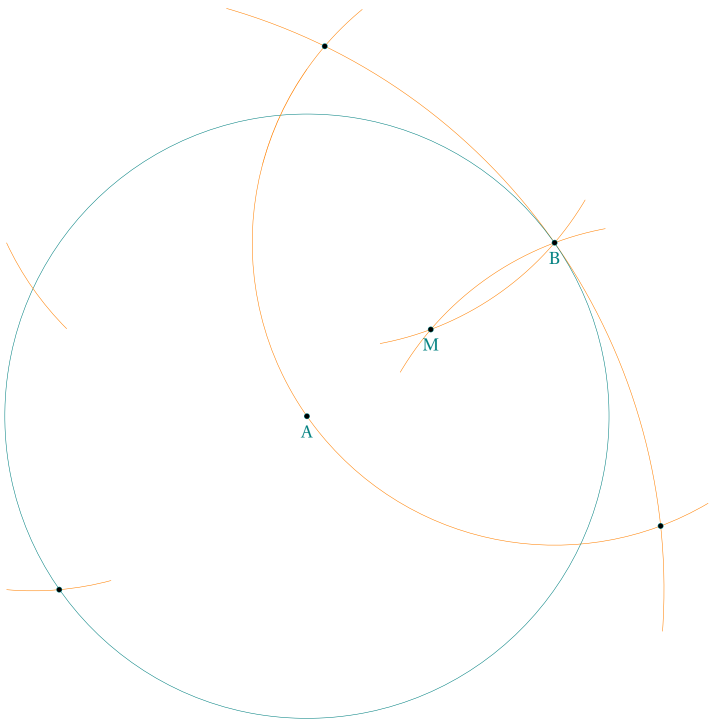


```
\begin{tikzpicture}
\tkzDefPoints{0/0/A,4/2/B,2/3/K}
\tkzDefCircle[R](A,1) \tkzGetPoint{a}
\tkzDefCircle[R](B,3) \tkzGetPoint{b}
\tkzInterCC[R](A,1)(K,3) \tkzGetPoints{a}{a'}
\tkzInterCC[R](B,3)(K,3) \tkzGetPoints{b}{b'}
\tkzInterLL(a,a')(b,b')  \tkzGetPoint{X}
\tkzDefPointBy[projection= onto A--B](X) \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}
```

44.23. Milieu d'un segment au compas

Milieu d'un segment au compas

Cet exemple consiste à déterminer le milieu d'un segment à l'aide d'un compas uniquement.



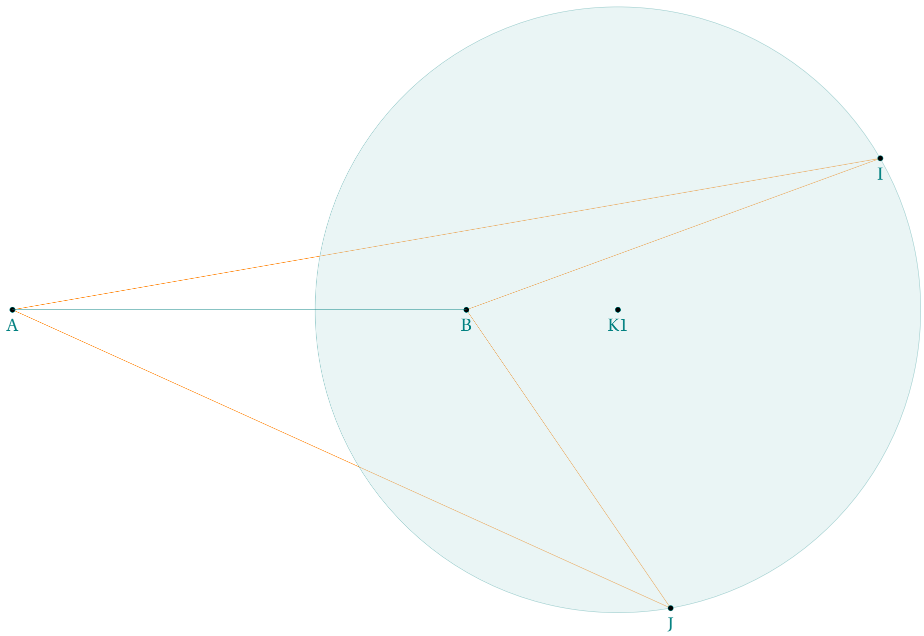

```
\begin{tikzpicture}
\tkzDefPoint(0,0){A}
\tkzDefRandPointOn[circle= center A radius 4]    \tkzGetPoint{B}
\tkzDefPointBy[rotation= center A angle 180](B)  \tkzGetPoint{C}
\tkzInterCC(A,B)(B,A)                            \tkzGetPoints{I}{I'}
\tkzInterCC(A,I)(I,A)                            \tkzGetPoints{J}{B}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

44.24. Définition d'un cercle d' _Apollonius_

Définition d'un cercle d' _Apollonius_

D'Après Wikipedia : *Apollonius a montré qu'un cercle peut être défini comme l'ensemble des points d'un plan qui ont un rapport spécifié de distances à deux points fixes, appelés foyers. Ce cercle apollinien est à la base du problème de la poursuite d'Apollonius. ... Les solutions de ce problème sont parfois appelées les cercles d'Apollonius.*

Explication
Un cercle est l'ensemble des points d'un plan qui sont équidistants d'un point O donné. La distance r du centre est appelée rayon, et le point O est appelé centre. C'est la définition la plus simple, mais ce n'est pas la seule. Apollonios de Perga donne une autre définition : L'ensemble des points dont les distances à deux points fixes sont dans un rapport constant est un cercle.
Avec `tkz-euclide` il est facile de vous montrer la dernière définition

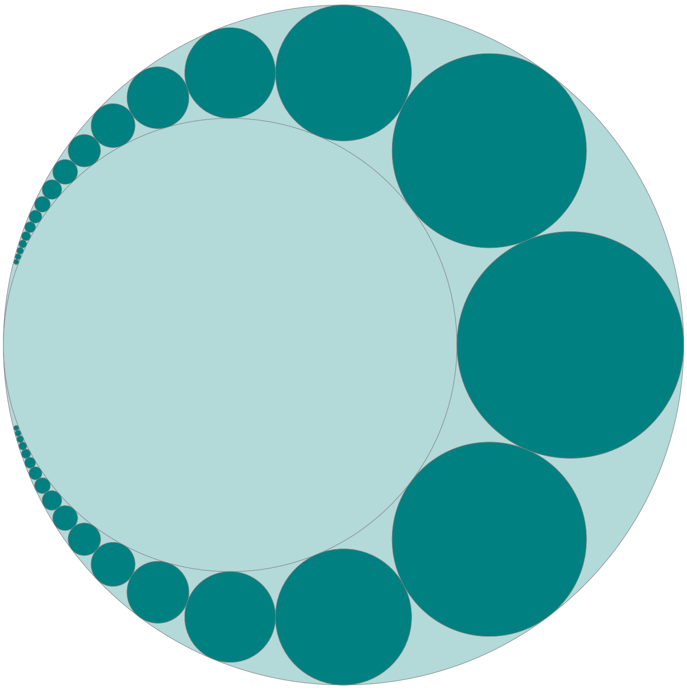



```
\begin{tikzpicture}[scale=1.5]
    % Firstly we defined two fixed point.
    % The figure depends of these points and the ratio K
\tkzDefPoint(0,0){A}
\tkzDefPoint(4,0){B}
    % tkz-euclide.sty knows about the apollonius's circle
    % with K=2 we search some points like  I such as IA=2 x IB
\tkzDefCircle[apollonius,K=2](A,B) \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}
```

44.25. Application de l'inversion : Chaîne de Pappus

Chaîne de Pappus

De Wikipedia *En géométrie, la chaîne de Pappus est un anneau de cercles entre deux cercles tangents étudié par Pappus d'Alexandrie au IIIe siècle après J.-C.*

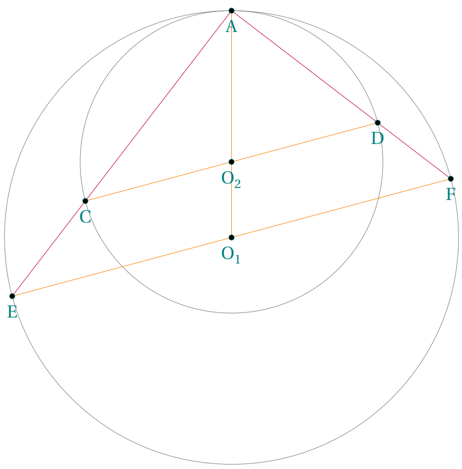


```
\begin{tikzpicture}[ultra thin]
  \pgfmathsetmacro{\xB}{6}%
  \pgfmathsetmacro{\xC}{9}%
  \pgfmathsetmacro{\xD}{(\xC*\xC)/\xB}%
  \pgfmathsetmacro{\xJ}{(\xC+\xD)/2}%
  \pgfmathsetmacro{\r}{\xD-\xJ}%
  \pgfmathsetmacro{\nc}{16}%
  \tkzDefPoints{0/0/A,\xB/0/B,\xC/0/C,\xD/0/D}
  \tkzDefCircle[diameter](A,C) \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}
```

44.26. Livre des lemmes proposition 1 Archimède

Livre des lemmes proposition 1 Archimède

Si deux cercles se touchent en A*, et si* [CD], [EF] *sont des diamètres parallèles,* A, C *et* E *sont alignés.*




```
\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}
```

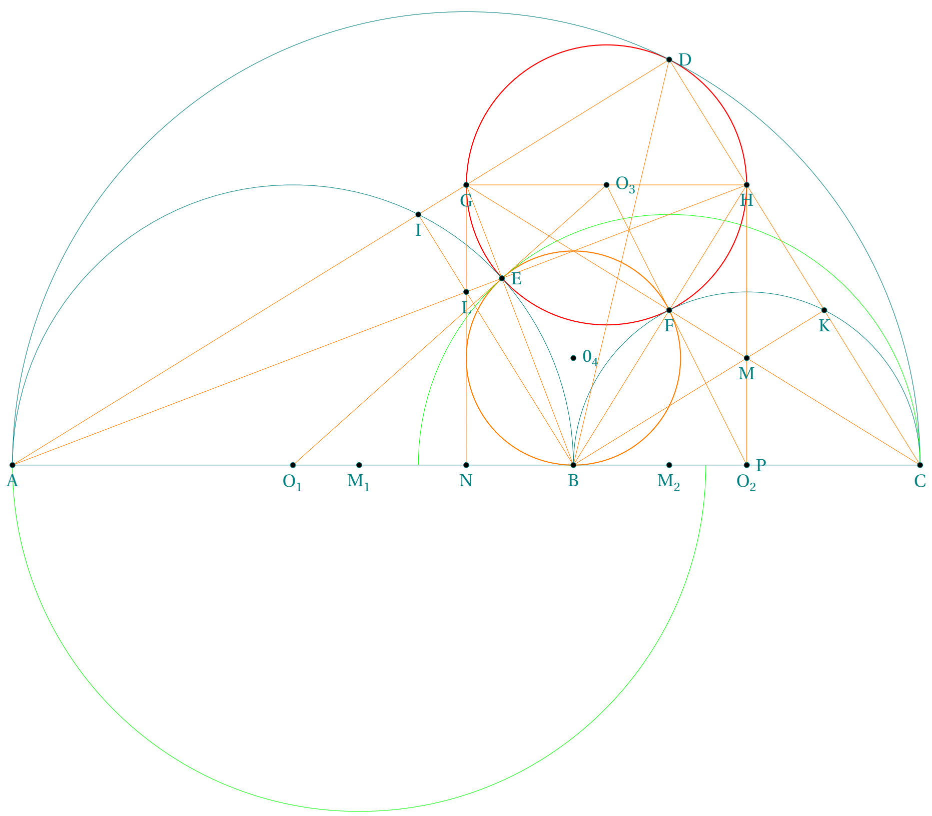
(CD) $\parallel$ (EF) (AO_1) is secant to these two lines so $\widehat{A0_2C} = \widehat{A0_1E}$.
Since the triangles AO_2C and AO_1E are isosceles the angles at the base are equal widehat$AC0_2$ = $\widehat{AE0_1} = \widehat{CA0_2} = \widehat{EA0_1}$. Thus A,C and E are aligned

44.27. Livre des lemmes proposition 6 Archimède

Livre des lemmes proposition 6 Archimède

Soit AC*, le diamètre d'un demi-cercle, divisé en* B *de sorte que* $AC/AB = \phi$ *ou dans n'importe quel rapport. Décrire les demi-cercles à l'intérieur du premier demi-cercle et sur* AB, BC *comme diamètres, et supposer un cercle tracé touchant les trois demi-cercles. Si* GH *est le diamètre de ce cercle, trouver la relation entre* GH *et* AC.

```
\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}
```



Soit GH le diamètre du cercle qui est parallèle à AC, et que le cercle touche les demi-cercles sur AC, AB, BC en D, E, F respectivement.
Alors, selon la Proposition 1, A,G et D sont alignés, ainsi que D, H et C. De même, A E et H sont alignés, C F et G sont alignés, tout comme B E et G, et B F et H.
Laissons (AD) rencontrer le demi-cercle sur [AC] en I, et laissons (BD) rencontrer le demi-cercle sur [BC] en K. Relions CI, CK en rencontrant AE, BF en L, M, et laissons GL, HM prolongées rencontrer AB en N, P respectivement.
Maintenant, dans le triangle AGB, les perpendiculaires de A, C sur les côtés opposés se rencontrent en L. Par conséquent, selon les propriétés des triangles, (GN) est perpendiculaire à (AC). De même, (HP) est perpendiculaire à (BC).
Encore une fois, puisque les angles en I, K, D sont droits, (CK) est parallèle à (AD), et (CI) à (BD).
Par conséquent,

$$\frac{AB}{BC}=\frac{AL}{LH}=\frac{AN}{NP} \quad \text{and} \quad \frac{BC}{AB}=\frac{CM}{MG}=\frac{PC}{NP}$$

donc

$$\frac{AN}{NP}=\frac{NP}{PC} \quad \text{so} \quad NP^2=AN\times PC$$

Supposons maintenant que B divise [AC] selon la divine proportion, c'est-à-dire :

$$\phi=\frac{AB}{BC}=\frac{AC}{AB} \quad \text{then} \quad AN=\phi NP \text{and} \quad NP=\phi PC$$

nous avons donc

$$AC=AN+NP+PC \quad \text{either} \quad AB+BC==AN+NP+PC \quad \text{or} \quad (\phi+1)BC=AN+NP+PC$$

nous obtenons donc

$$(\phi+1)\mathrm{BC} = \phi\mathrm{NP} + \mathrm{NP} + \mathrm{PC} = (\phi+1)\mathrm{NP} + \mathrm{PC} = \phi(\phi+1)\mathrm{PC} + \mathrm{PC} = \phi^2 + \phi + 1)\mathrm{PC}$$

puisque

$$\phi^2 = \phi + 1 \quad \text{then} \quad (\phi+1)\mathrm{BC} = 2(\phi+1)\mathrm{PC} \quad \text{i.e.} \quad \mathrm{BC} = 2\mathrm{PC}$$

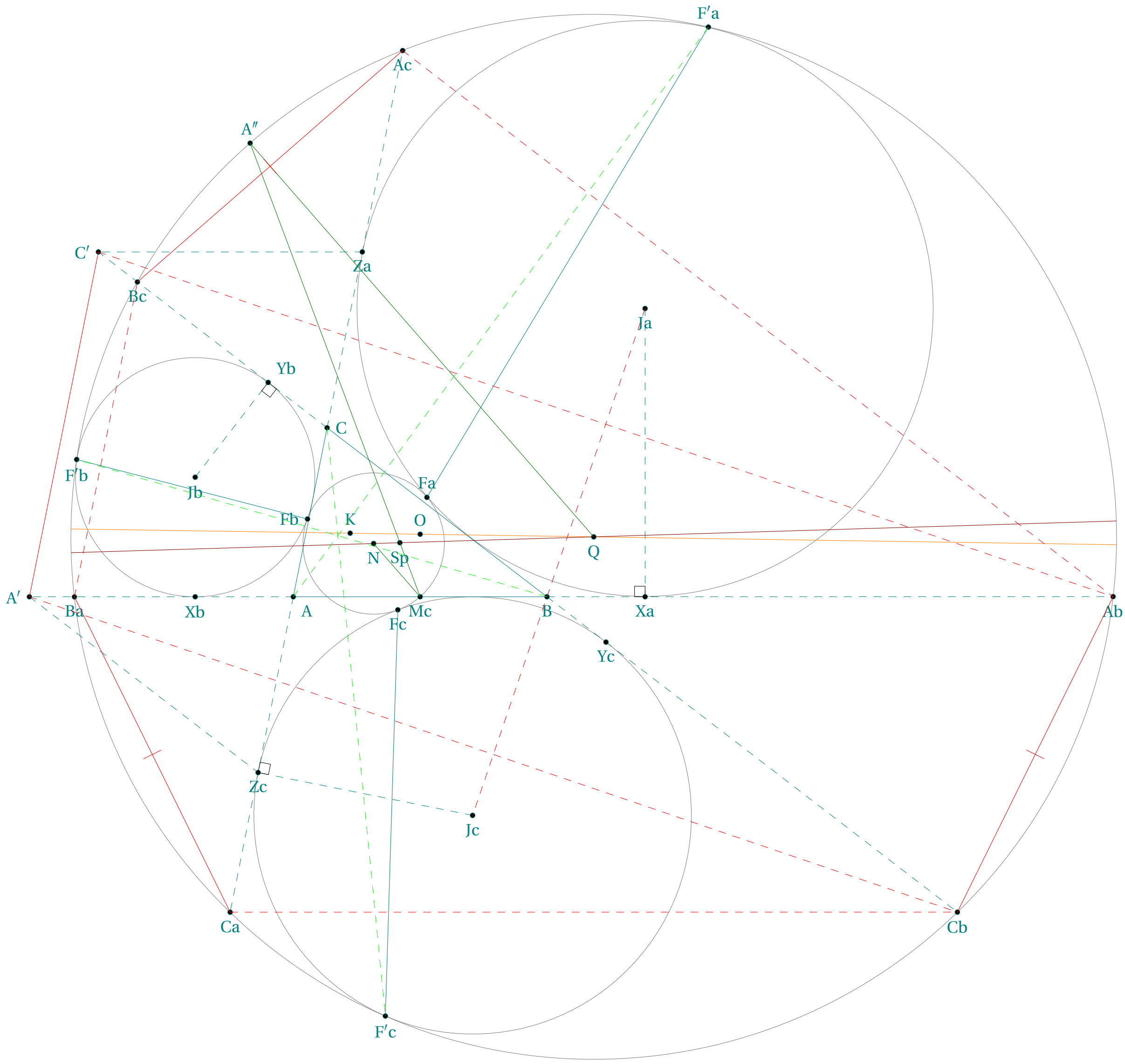
c'est à dire, p is the middle of the segment BC.
Une partie de la preuve de `https://www.cut-the-knot.org`

44.28. "Le" Le cercle d' APOLLONIUS

Le cercle d'Apollonius d'un triangle _Apollonius_

Le cercle qui touche les trois excircles d'un triangle et les englobe est souvent appelé "Le" cercle d'Apollonius (Kimberling 1998, p. 102)

Explication
Le but des premiers exemples était de montrer la simplicité avec laquelle nous pouvions recréer ces propositions. Avec TikZ il faut faire des calculs et utiliser la trigonométrie alors qu'avec `tkz-euclide` il suffit de construire des objets simples
Mais n'oubliez pas que derrière ou bien au-dessus de `tkz-euclide` il y a TikZ. Je ne fais que créer une interface entre TikZ et l'utilisateur de mon paquet.
Le dernier exemple est très complexe et c'est pour vous montrer tout ce que l'on peut faire avec `tkz-euclide`.


F′a
Ac
A″
C′
Za
Bc
Ja
Yb
C
F′b
Jb
Fa
Fb
K
O
N
Sp
Q
A′
Ba
Xb
A
Mc
Fc
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}
```

Dixième partie

FAQ

45. FAQ

45.1. Erreurs les plus courantes

Pour le moment, je me base sur ma propre expérience, car ayant changé de syntaxe plusieurs fois, j'ai commis un certain nombre d'erreurs. Cette section va être étoffée. Avec la version 4.05, de nouveaux problèmes peuvent apparaître.

— L'erreur que je commets le plus souvent est d'oublier de mettre un "s" dans la macro utilisée pour dessiner plus d'un objet : comme `\tkzDrawSegment(s)` ou `\tkzDrawCircle(s)`, ok comme dans cet exemple `\tkzDrawPoint(A,B)` quand vous avez besoin de `\tkzDrawPoints(A,B)` ;

— N'oubliez pas que depuis la version 4, l'unité est obligatoirement le "cm", il est donc nécessaire de retirer l'unité comme ici `\tkzDrawCircle[R](O,3cm)` qui devient `\tkzDrawCircle[R](O,3)`. Les options traditionnelles de `TikZ` conservent leurs unités, par exemple `below right = 12pt`, tandis que l'on écrira `size=1.2` pour positionner un arc dans `\tkzMarkAngle` ;

— L'erreur suivante m'arrive encore de temps en temps. Un point qui est créé a son nom entre crochets tandis qu'un point qui est utilisé soit comme option soit comme paramètre a son nom entre accolades. Exemple `\tkzGetPoint(A)` Lors de la définition d'un objet, utilisez des accolades et non des crochets, donc écrivez : `\tkzGetPoint{A}` ;

— Les changements dans l'obtention des points d'intersection entre les lignes et les cercles échangent parfois les solutions, ce qui conduit soit à une mauvaise figure soit à une erreur.

— `\tkzGetPoint{A}` à la place de `\tkzGetFirstPoint{A}`. Lorsqu'une macro donne deux points comme résultats, soit nous récupérons ces points en utilisant `\tkzGetPoints{A}{B}`, soit nous récupérons seulement l'un des deux points, en utilisant `\tkzGetFirstPoint{A}` ou `\tkzGetSecondPoint{A}`. Ces deux points peuvent être utilisés avec la référence `tkzFirstPointResult` ou `tkzSecondPointResult`. Il est possible qu'un troisième point soit donné comme `tkzPointResult` ;

— Mélanger les options et les arguments ; toutes les macros qui utilisent un cercle ont besoin de connaître le rayon du cercle. Si le rayon est donné par une mesure, alors l'option inclut un `R`.

— Les angles sont donnés en degrés, plus rarement en radians.

— Si une erreur se produit dans un calcul lors du passage de paramètres, il vaut mieux effectuer ces calculs avant d'appeler la macro.

— Ne mélangez pas la syntaxe de `pgfmath` et de `xfp`. J'ai souvent choisi `xfp` mais si vous préférez pgfmath, faites vos calculs avant de passer les paramètres.

— Erreur "dimension trop grande" : Dans certains cas, cette erreur se produit. Une façon de l'éviter est d'utiliser l'option "`veclen`". Lorsque cette option est utilisée dans une portée, la fonction "veclen" est remplacée par une fonction dépendante de "xfp". Ne pas utiliser de macros d'intersection dans cette portée. Par exemple, une erreur se produit si vous utilisez la macro `\tkzDrawArc` avec un angle trop petit. L'erreur est produite par la bibliothèque `decoration` lorsque vous voulez placer une marque sur un arc. Même si la marque est absente, l'erreur est toujours présente.

Index