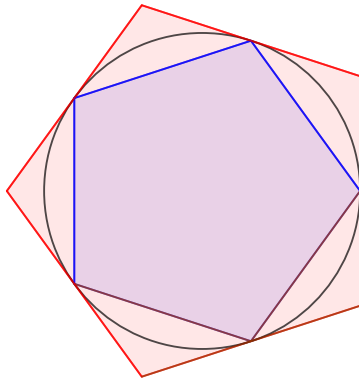




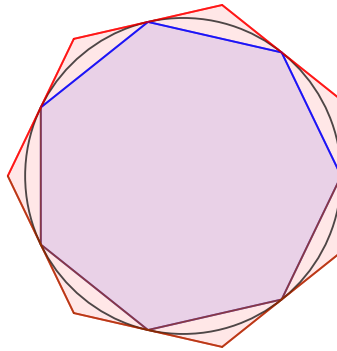
La méthode d'Archimède : correction

La **méthode d'Archimède** consiste à donner une approximation de π . Pour réaliser cela, on va construire des polygones réguliers à n côtés inscrits et circonscrits à un cercle de rayon $\frac{1}{2}$. On va ensuite calculer le périmètre de ces polygones.

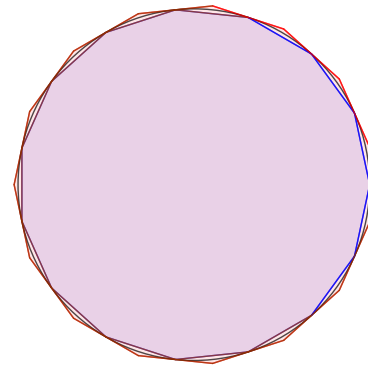
Le périmètre d'un tel cercle est π . Donc plus le nombre de côtés de ces polygones sera élevé, plus l'approximation de π sera bonne.



$n = 5$



$n = 7$



$n = 15$

On note a_n le périmètre d'un polygone inscrit à n côtés et on note b_n le périmètre d'un polygone circonscrit à n côtés. Le but est de déterminer une expression de a_n et de b_n en fonction de n et de calculer des valeurs de ces réels pour des n très grands.

1 Périmètre des polygones inscrits

On prend le cas de la figure ci-contre. On va déterminer la longueur d'un côté du polygone avant d'en déduire son périmètre. Pour cela, on prend deux sommets consécutifs A et B. On note H le pied de la hauteur issue de O.

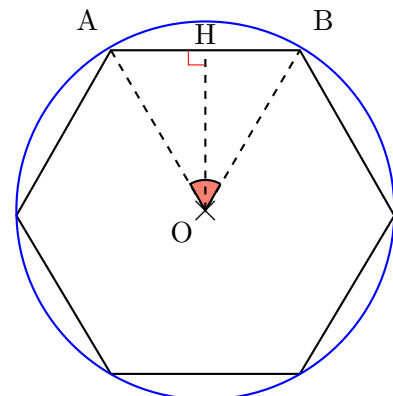
est la hauteur issue de O mais aussi sa médiatrice puisque ABO est isocèle. Donc $\widehat{AOH}$ mesure la moitié de $\frac{2\pi}{n}$ soit $\frac{\pi}{n}$.

1. Quelles sont les valeurs des longueurs de AO et OB ?

Le cercle est de rayon $\frac{1}{2}$. Or A et B sont sur le cercle donc $OA = OB = \frac{1}{2}$.

2. Donner, en radian, la mesure de l'angle $\widehat{AOH}$.

Un tour complet mesure 2π . Si on réalise un polygone à n côtés alors chacun des angles mesure $\frac{2\pi}{n}$. Or [OH]



3. En utilisant les formules de trigonométrie dans un triangle rectangle, exprimer AH en fonction de n .

AOH est rectangle en H alors $\sin(\widehat{AOH}) = \frac{AH}{AO}$. Ce qui donne $\sin\left(\frac{\pi}{n}\right) = \frac{AH}{\frac{1}{2}}$ soit $AH = \frac{\sin\left(\frac{\pi}{n}\right)}{2}$.

4. En déduire l'expression de a_n .

$AB = 2 \times AH = 2 \times \frac{\sin\left(\frac{\pi}{n}\right)}{2} = \sin\left(\frac{\pi}{n}\right)$. Un côté d'un polygone inscrit a donc pour longueur $\sin\left(\frac{\pi}{n}\right)$. Puisqu'il y a n côtés, on a :

$$a_n = n \sin\left(\frac{\pi}{n}\right)$$

2 Périmètre des polygones circonscrits

On prend cette fois-ci la figure ci-contre. En raisonnant de la même façon que pour la précédente partie, déterminer la longueur de AH, puis de AB et enfin de b_n .

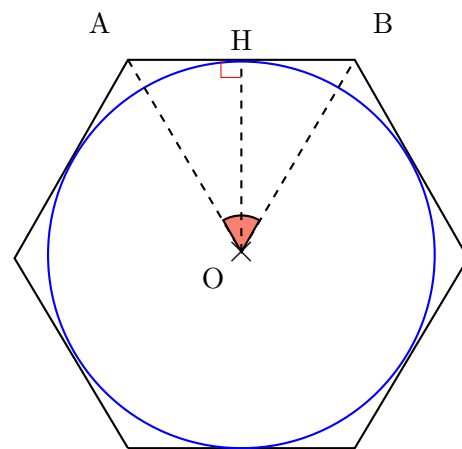
On se place dans le triangle AOH rectangle en H.

$\tan(\widehat{AOH}) = \frac{AH}{AO}$ ce qui donne $\tan\left(\frac{\pi}{n}\right) = \frac{AH}{\frac{1}{2}}$ et donc

$$AH = \frac{\tan\left(\frac{\pi}{n}\right)}{2}.$$

$$AB = 2 \times AH = 2 \times \frac{\tan\left(\frac{\pi}{n}\right)}{2} = \tan\left(\frac{\pi}{n}\right).$$

Puisque qu'il y a n côtés, on a $b_n = n \tan\left(\frac{\pi}{n}\right)$.



3 Direction Python !

Maintenant que l'on a déterminé, en fonction du nombre de côtés n , les périmètres a_n et b_n , il est temps de tester cette méthode d'approximation sur Python.

- Écrire une fonction Archimède qui prend en paramètre une variable entière n représentant le nombre de côtés des polygones qui renvoie le couple $(a_n ; b_n)$.

```
1 from math import sin, tan, pi
2 def Archimede(n):
3     return n*sin(pi/n), n*tan(pi/n)
```

2. Tester le programme pour $n = 5$. Quelle approximation est meilleure ?

Archimede(5) et le programme retourne (2.938926261462366, 3.6327126400268046). L'approximation avec a_5 est meilleure.

3. On cherche une valeur approchée de π à 10^{-p} près. On remarque, que : $\forall n \in \mathbb{N}, a_n \leq \pi \leq b_n$.
Il suffit donc d'avoir $b_n - a_n \leq 10^{-p}$ pour obtenir l'approximation recherchée.
A quelle condition sort-on de la boucle ?

```
1 def ArchimedePrecision(p):
2     n = 4
3     a,b = archimedeSimple(4)
4     while ... :
5         n = ...
6         a,b = archimedeSimple(n)
7     return a,b,n
```

On sort de la boucle quand la différence entre a_n et b_n est inférieure à 10^{-p} .

4. Que représentent les variables p et n ?

p représente la précision souhaitée et n représente le nombre le côtés des polygones réguliers.

5. Compléter les lignes 4 et 5 du programme.

```
1 def ArchimedePrecision(p):
2     n = 4
3     a,b = archimedeSimple(4)
4     while b-a>10**-p:
5         n = n+1
6         a,b = archimedeSimple(n)
7     return a,b,n
```

6. En testant le précédent programme, pour quelle valeur de n a-t-on une précision de 10^{-10} .

On tape ArchimedePrecision(10) et le programme nous retourne (3.1415926535564602, 3.1415926536564593, 393741).
Il faut donc un polygone à 393 741 côtés pour avoir une telle précision.