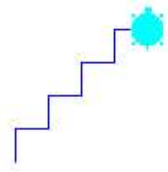


La boucle Pour



1- Voici 4 programmes qui permettent à la tortue joe de dessiner la figure ci-contre.

```
joe.forward(20)   for i in range (4):   for i in range(0,12,3):   for i in range (5,9):
joe.right(90)      joe.forward(20)      joe.forward(20)      joe.forward(20)
joe.forward(20)    joe.right(90)         joe.right(90)        joe.right(90)
joe.left(90)       joe.forward(20)       joe.forward(20)      joe.forward(20)
joe.forward(20)    joe.left(90)          joe.left(90)         joe.left(90)
joe.right(90)
joe.forward(20)
joe.left(90)
joe.forward(20)
joe.right(90)
joe.forward(20)
joe.left(90)
joe.forward(20)
joe.right(90)
joe.forward(20)
joe.left(90)
```

Quel(s) avantage(s) ont les trois derniers programmes?

Quelle(s) condition(s) doit-on avoir pour qu'on puisse simplifier de cette manière un programme?

On parle alors d'une boucle **Pour**.

Une boucle **Pour** s'écrit sous la forme suivante:

Langage naturel
Pour i=départ à fin par pas de p faire

instructions

Fin Pour

où i, **départ**, **fin** et **p** sont des entiers

Pour i=départ à fin par pas de 1 faire

instructions

Fin Pour

Langage python
for i in range (départ,N,p):

instruction1

instruction2

...

instructionK

avec **N=fin-départ+1**

for i in range (N):

instruction1

instruction2

...

instructionK

avec **N=fin-départ+1**

départ=0 et fin=N-1

Les instructions doivent être alignées.

i est le compteur de boucles,

départ est la valeur initiale du compteur,

fin est la valeur finale du compteur (ie La dernière itération a lieu quand i=fin).

p est l'incrément. Après chaque itération, le compteur est augmenté (ie incrémenté) de **p** unités.

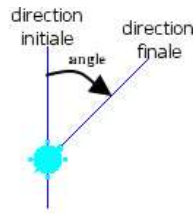
Conditions d'utilisation:

- **Une série d'instructions se répète.**
- **Le nombre de répétitions (d'itérations) est connu.**

La boucle Pour

2-a- En vous aidant du 1-, compléter les boucles Pour.

joe.forward(20)	for i in range():	for i in range(, ,5):	for i in range (,):
joe.right(120)	joe.	joe.	joe.
joe.forward(20)			
joe.right(120)			
joe.forward(20)			
joe.right(120)			



b- En vous aidant de

, que dessine la tortue joe?

c- Lancer TigerJython.

Ouvrir le fichier TriangleEquilateral.py du dossier BouclePour et compléter le programme avec l'une des deux boucles précédentes.

Exécuter le programme.

3- Réaliser les défis suivants:

♦ Avec une seule boucle **Pour**:

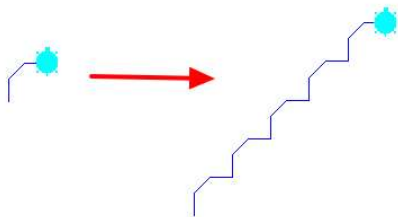
Défi A01: dessiner un carré dont les côtés mesurent 100 pixels. Appeler votre fichier DefiA01.

Défi A02: dessiner un pentagone de côté 100 pixels.

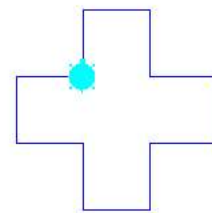
Défi A03: dessiner un hexagone de côté 100 pixels.

Défi A04: dessiner un octogone de côté 100 pixels.

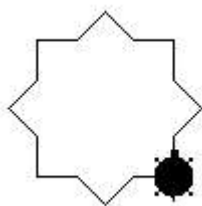
Défi A05: dessiner la figure suivante:
(20 pixels par côté).



Défi A06: dessiner une croix dont les côtés mesurent 100 pixels.

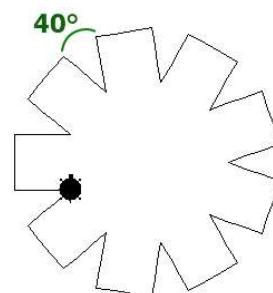


Défi A07:



côté=30 pixels

Défi A08:



côté=50 pixels angle=40° (360°/nb de branches)

La boucle Pour

- ♦ Avec deux boucles **Pour** imbriquées:

1- Analyse d'un exemple

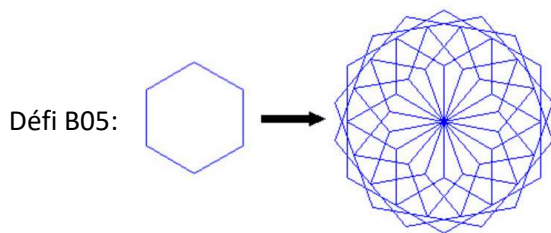
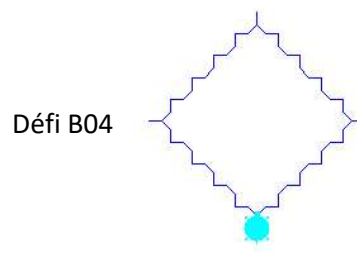
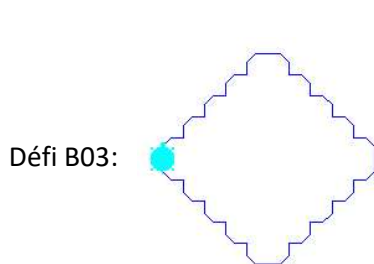
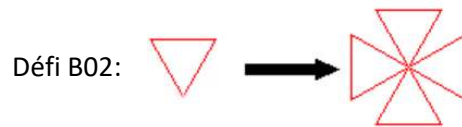
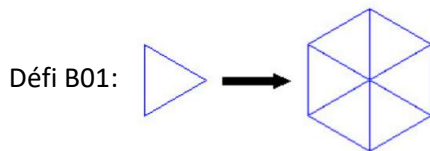
Ouvrir et exécuter le fichier LigneTriangles.py du dossier BouclePour.

La boucle `for j in range(3)` est imbriquée dans la boucle `for i in range(4)`

Quel est le rôle de la boucle `for j in range(3)` ?

Quel est le rôle de la boucle `for i in range(4)` ?

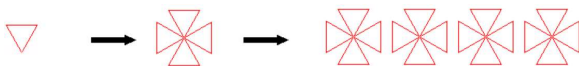
2- Les défis



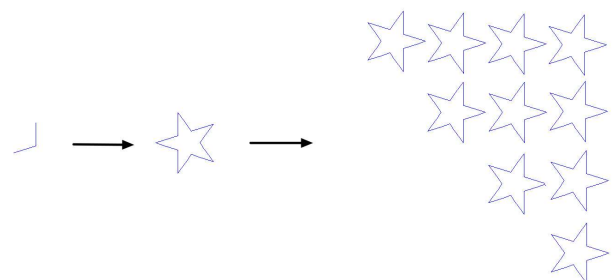
angle=20°

- ♦ Avec trois boucles **Pour** imbriquées:

Défi C01 :



Défi C02 :



Défi C03 :

