

# Quelques exemples de récursivité avec Géotortue et Python

Jean-François Kentzel(\*)

Les logiciels GéoTortue et Python sont téléchargeables gratuitement.

**Python** est un langage très évolué mais son utilisation élémentaire est assez simple et permet de montrer au lycée un langage plus concis que d'autres (comme Algobox par exemple).

L'utilisation de **Géotortue** (y compris au collège) est encore plus simple.

Il n'est pas question dans ce qui suit de l'excellent logiciel **AmiensPython** (version 2.0 sortie fin Septembre) pour des raisons de délais trop courts.

En bref : Amiens Python est une version de Python très conseillée pour le lycée !

Voir les références en fin d'article.

Une version électronique de ce texte, plus pratique pour les liens et surtout pour les couleurs dans les scripts, est disponible sur la page Disciplines/Math/TICE/Blog du lycée Pardailhan à Auch-32. On y trouve aussi la version AmiensPython du principal script (Von Koch) de ce texte ; les différences avec Python sont minimes.

## 1) Récursivité

Pour éviter les pointillés de  $n! = 1 \times 2 \times 3 \times \dots \times n$  (pointillés rédhibitoires si on adresse une commande à une machine !), on peut poser  $a = 1$  et  $k = 1$  et dire : tant que  $k < n + 1$ , on remplace  $a$  par  $a \times k$  et  $k$  par  $k + 1$ ; on effectue alors une boucle ordinaire : c'est le mode itératif.

On dit qu'un programme de calcul (un algorithme) est **récursif** s'il s'appelle lui-même.

Le programme Fact est défini par  $\text{Fact}(0) = 1$  et si  $n > 0$ ,  $\text{Fact}(n) = \text{Fact}(n - 1) \times n$ . Pour calculer  $n!$ , le programme Fact s'auto-appelle ( $n$  fois) : il est récursif.

De même le programme récursif Fib (qui calcule le  $n$ -ème terme de la suite de Fibonacci) est défini par :

$\text{Fib}(1) = \text{Fib}(2) = 1$  et si  $n > 2$ ,  $\text{Fib}(n) = \text{Fib}(n - 1) + \text{Fib}(n - 2)$ .

À chaque étape le programme Fib s'appelle deux fois.

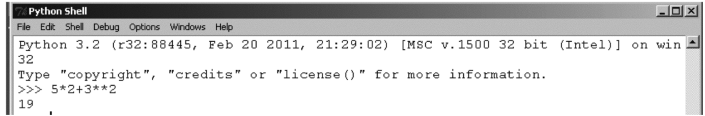
(une définition itérative du  $n$ -ème terme  $b$  de la suite de Fibonacci étant : on pose  $a = b = k = 1$ , puis, tant que  $k < n - 1$ , on remplace  $a$  par  $b$ ,  $b$  par  $a + b$  et  $k$  par  $k + 1$ )

Dans ces deux exemples la différence entre les modes de calcul itératif et récursif est uniquement formelle. Dans ce qui suit on ne s'aventure pas sur le terrain des avantages comparés de ces deux façons de décrire des calculs.

(\*) Jkentzel@ac-toulouse.fr (lycée Pardailhan à Auch-32).

## 2) Démarrage avec Python

Quand on ouvre Python, on est *a priori* en mode interactif :

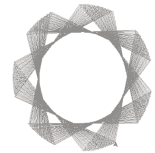


Il faut cliquer sur File puis sur New Window si on veut écrire un script de programme (la page interactive est alors accessible en cliquant sur Run et Python Shell)

Le langage Python est rapidement utilisable : ci-dessous, le programme de gauche donne, en tapant F5 ou Ctrl+F5, le dessin de droite (où les points sont cocycliques...) dans une fenêtre graphique.

```
from turtle import *
i=0
while i<200:
    forward(250)
    left(103)
    i=i+1
```

**Commentaires :**  
**turtle** signifie tortue. C'est un module, ou un mode, de Python qu'il faut appeler ici. Il y en a d'autres comme math, random, ...  
**forward(x)** : avancer de  $x$  pixels  
**left (y)** : tourner (à gauche !) de  $y$  degrés.



Avec AmiensPython tous les modules utiles au lycée sont automatiquement importés et on a directement sous les yeux les deux pages : Edition de script et Exécution.

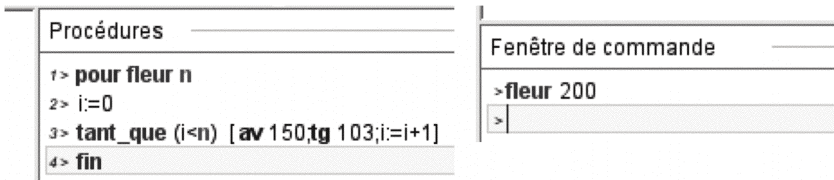
## 3) Démarrage avec Géotortue

Les programmes sont qualifiés de procédures. On peut voir en ouvrant une page GéoTortue qu'on a d'un seul coup les trois pages de Python évoquées ci-dessus : la page des programmes (appelés « procédures »), la page « Fenêtre de commandes » qui permet de faire exécuter les procédures et la page graphique qui est directement, par défaut, en mode tortue.

Les scripts figurant dans cet article sont compréhensibles avec les commandes élémentaires suivantes qui permettent de diriger « la tortue » (héritière de la célèbre tortue de Logo) :

Commande	Traduction pour Python	Traduction pour GéoTortue
Avancer de $x$ pixels	forward( $x$ )	av $x$
Tourner à gauche de $y$ degrés	left( $y$ )	tg $y$
Tourner à droite de $y$ degrés	right( $y$ )	td $y$

On obtient le même dessin que ci-dessus avec



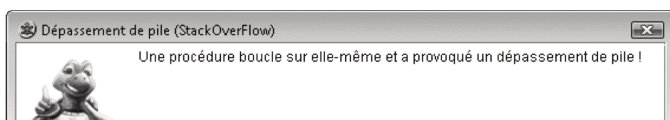
Le « pour » doit être écrit au début de chaque programme ou sous-programme de Géotortue. Il n'a rien à voir avec le « pour  $i$  de 1 à  $n$  » classique qui permet d'obtenir

des boucles. La traduction littérale pour Python est `def` (quand on définit un objet, comme au 4)).

#### 4) Remarque

On peut exécuter ce dessin avec un programme récursif, par exemple avec GéoTortue. On nomme Récursifleur ce programme.

Mais la commande `>Récursifleur 200` donne un dessin qui ne s'arrête pas et finit par ressembler à une couronne noire en déclenchant après quelques minutes le message d'erreur suivant :



Il faut un test d'arrêt :

```
1> pour Récursifleur n
2> si (n>1) alors [ av 150 ; tg 103 ; Récursifleur n-1 ]
3> fin
```

#### 5) Le flocon de Von Koch

Le script pour Python qui suit (à gauche) est copié-collé à la page 94/137 de [1], puis une autre copie suivie des modifications de syntaxe nécessaires donne celui de GéoTortue (à droite).

$l$  est la longueur de départ et  $n$  est le nombre d'étapes à partir de la première.

```
Von Koch.py - F:\PYTHON\Von Koch.py
File Edit Format Run Options Windows Help
# Flocon de Von Koch
from turtle import*

def segment(l,n): # c'est un sous-programme
    if n==0:
        forward(l)
    else:
        segment(l/3,n-1)
        left(60)
        segment(l/3,n-1)
        right(120)
        segment(l/3,n-1)
        left(60)
        segment(l/3,n-1)

def flocon(l,n):
    segment(l,n)
    right(120)
    segment(l,n)
    right(120)
    segment(l,n)

# il reste à appeler les données l et n
etape=int(input("donnez n=le nombre d'étapes"))
taille=float(input("donnez l=le côté initial"))

flocon(taille,etape) # le coeur du programme!

exitonclick() # détail pratique pour quitter
```

```
GéoTortue - Von Koch.trt *
Procédures
1> pour segment l n
2> si (n==0) alors [ av l ] sinon [segment l/3 n-1]
3> tg 60
4> segment l/3 n-1
5> td 120
6> segment l/3 n-1
7> tg 60
8> segment l/3 n-1
9> fin
10>
11> pour flocon l n
12> segment l n
13> td 120
14> segment l n
15> td 120
16> segment l n
17> fin
```

**Premiers commentaires**

Le script pourrait être raccourci car il contient des répétitions, mais il serait alors moins lisible.

Avec Python les commentaires précédés par # ne sont pas lus par l'ordinateur. GéoTortue semble intransigeant pour l'écriture de « si alors sinon » sur une seule ligne.

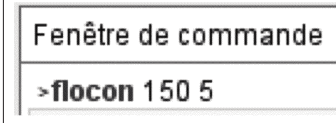
La seule différence entre les scripts pour les deux logiciels réside dans ce qu'il faut faire pour obtenir l'exécution du programme :

Avec Python

Les demandes des entrées  $l$  et  $n$  figurent (sous une forme *a priori* peu naturelle !) à la fin du programme.  
Il reste à taper f5 ou Ctrl+f5.  
On donne ensuite  $l$  et  $n$  lorsqu'elles sont demandées.

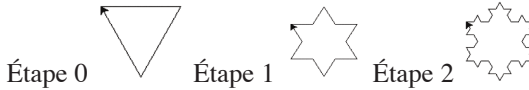
Avec GéoTortue

On tape directement les entrées  $l$  et  $n$  :



**5) Commentaires sur les scripts du flocon**

L'intérêt de ces scripts est que nous présentons tous le flocon sous la forme :

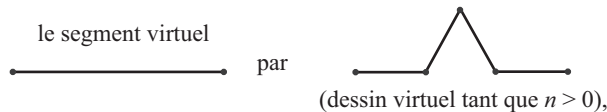


alors qu'ils déclenchent (pour  $n = 5$ ) un dessin qui commence ainsi :



On voit que sa « chronologie » est assez différente !

Le programme principal « flocon » réalise (virtuellement) un triangle équilatéral mais (sauf si on a entré  $n = 0$  au départ !) il ne le fait pas tracer. Il faut noter que la ligne forward( $l$ ) = av  $l$  est la seule qui fait effectivement tracer un segment. C'est le sous-programme « segment » qui est récursif, c'est-à-dire qu'il s'appelle, quatre fois, tant que  $n > 0$ , ce qui permet à chaque étape de remplacer



cependant qu'à chaque étape  $n$  est diminué de 1.

On trouve en annexe le déroulement complet du programme dans le cas simple où  $n = 1$ .

## Références

### Python

[1] Algorithmique au lycée - Colloque de Toulouse – 9 et 10 Juin 2011– de Yannick Le Bastard (LEGTA Frédéric Bazille-34).

Une progression avec Python 3 ambitieuse et semblant réaliste pour le lycée.  
Téléchargeable sur le site de l'IREM de Toulouse.

[2] Apprendre à programmer avec Python 3 de Gérard Swinnen (Liège).  
Introduction plus complète à d'autres fonctionnalités de Python.  
Éditions Eyrolles-téléchargeable.

[3] Le Site du Zéro. Excellents tutoriels (attention aux différences entre les versions 2 et 3 !).

Remarque essentielle : les scripts figurant dans ces textes peuvent souvent être copiés puis collés sur une page Python mais il faut alors refaire l'« indentation », c'est-à-dire l'ensemble des retraits au début des lignes (avec la touche de tabulation), car elle n'est pas seulement un confort visuel : elle fait partie du script. Python vous guide (bien) si vous oubliez de le faire.

### Géotortue

[4] On trouve facilement la page officielle sur Internet. Tout est simple et très bien expliqué ... en français ! Les pages sont pleines de liens passionnants : ne pas manquer, par exemple, l'étonnante page du collège Gabriel Péri de Bezons (95) ou celles relatives à l'ancêtre de Géotortue (la tortue de LOGO), aux extensions (géométries dans l'espace et non euclidiennes), aux activités pour les élèves, etc. GéoTortue est développé par Salvatore Tummarello avec la participation de E. Adam, J.-F. Jamart et S. Petitjean. Il est soutenu par l'IREM de Paris-Nord.

### AmiensPython

[5] Adresse du site officiel : <http://amienspython.tuxfamily.org/>  
Téléchargement gratuit (éventuellement sur une clé USB).

AmiensPython – Documentation pour l'enseignant : 83 pages limpides (et aérées).  
Voir aussi 80 programmes, classiques ou originaux, prêts à tourner. *À découvrir absolument !*

AmiensPython est un logiciel dérivé de Python 2, développé par Vincent Maille avec la collaboration de Julien Pollet, Agnès Baraquin et François Prédinas.







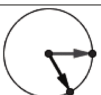

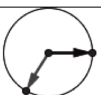
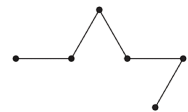

### L'itératif et le récursif

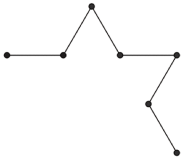

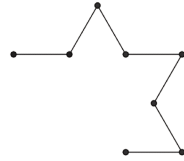

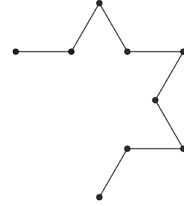

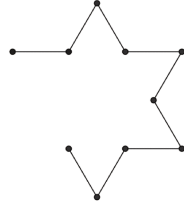

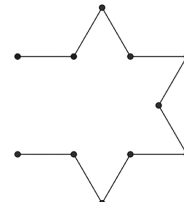
[6] Itération et récursivité, Jacques Arsac 1993, Université Pierre et Marie Curie.  
<http://hal.archives-ouvertes.fr/docs/00/35/91/90/PDF/d14p081.pdf>


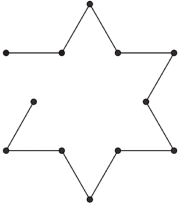

Il y a des tas d'autres pages sur le Net, par exemple celle-ci, anecdotique mais instructive :

Fibonacci calculé avec différents langages informatiques

**Annexe : déroulement du programme flocon de Von Koch pour  $n = 1$**   
(remerciements à Bernard Langer pour nos échanges et la réalisation de cette annexe)

Instruction	Direction de la tortue	Dessin obtenu
Flocon (3,1)		
Segment(3,1)		
Segment(1,0)		
Left(60)		
Segment(1,0)		
Right(120)		
Segment(1,0)		
Left(60)		
Segment(1,0)		
Right(120)		
Segment(3,1)		
Segment(1,0)		
Left(60)		

Segment(1,0)		
Right(120)		
Segment(1,0)		
Left(60)		
Segment(1,0)		
Right(120)		
Segment(3,1)		
Segment(1,0)		
Left(60)		
Segment(1,0)		

Right(120)		
Segment(1,0)		
Left(60)		
Segment(1,0)		