

# La récursivité et Fibonacci

## Le problème

Léonard de Pise (vers 1175 ; vers 1250), dit Fibonacci, fut un précurseur dans le domaine de la modélisation. Il reste célèbre pour avoir étudié une suite qui correspond à la progression d'une population de couples de lapins. Cette suite est construite ainsi :

On part de **1** et **1**, que l'on ajoute ; on obtient 2.

On prend 1 et 2, que l'on ajoute ; on obtient 3.

On poursuit ainsi le processus : le terme suivant de la suite est la somme des deux précédents.

$$\begin{aligned} 1 + 1 &= 2. \\ 1 + 1 + 2 &= 3. \\ 1 + 1 + 2 + 3 &= 5. \\ 1 + 1 + 2 + 3 + 5 &= 8. \\ &\text{etc.} \end{aligned}$$

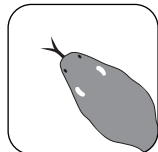
## Complément culturel

La suite de Fibonacci est en relation avec le fameux *nombre d'or*  $\phi = \frac{1+\sqrt{5}}{2}$ .

En effet, le rapport de deux consécutifs de ses éléments se rapproche de plus en plus de cette valeur.

€

## Les programmes



Le programme « Fib » renvoie les  $n$  premiers termes de la suite de Fibonacci. La variable  $I$  est un compteur de boucles, et tant que  $I$  est inférieur ou égal à  $n$ , la boucle est effectuée.

le programme

le résultat

```
>>> def Fib(n):
    A=1
    B=1
    print(A)
    print(B)
    K=2
    while K<=n:
        C=A+B
        print(C)
        A=B
        B=C
        K=K+1
```

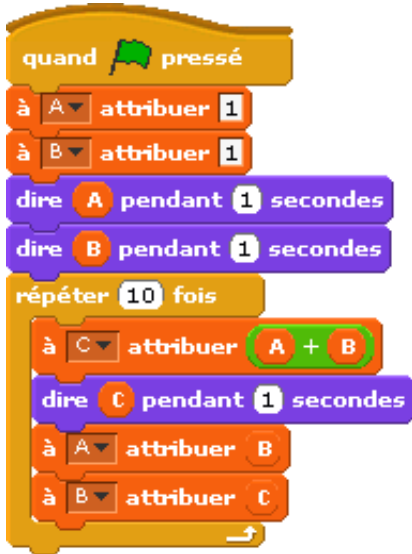
```
>>> Fib(10)
1
1
2
3
5
8
13
21
34
55
89
```

SCRATCH

## avec Scratch



On définit trois variables A, B et C. On donne à A et B la valeur 1 (au départ), puis on calcule leur somme que l'on met dans C. On remplace alors A par B, et B par C, puis on calcule leur somme que l'on met dans C, on poursuit ainsi le processus.



## Prolongements

La notion de récursivité est une notion centrale en informatique. Un programme est dit *récursif* lorsque, dans son déroulement, il s'appelle lui-même. On peut faire en sorte d'obtenir le terme de rang  $n$  de la suite de Fibonacci en utilisant, sous Python, le programme suivant :

le programme

```
>>> def Fibonacci (n) :
    if n==0:
        return (1)
    if n==1:
        return (1)
    if n>1:
        return (Fibonacci (n-1)+Fibonacci (n-2))
```

le résultat

```
>>> Fibonacci (10)
89
```

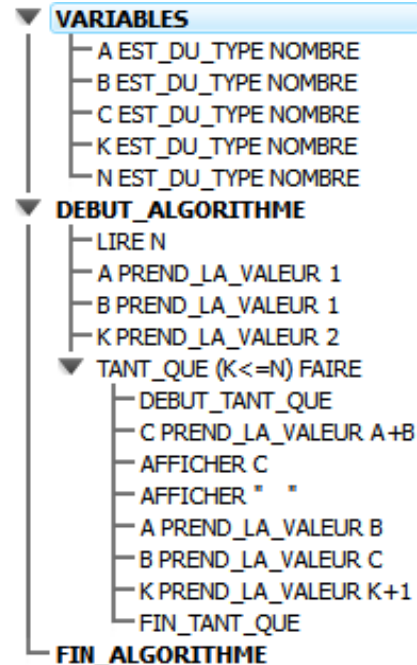


## avec AlgoBox

Nous devons déclarer au départ du programme toutes les variables utilisées :

- Les variables A, B et C permettent de calculer les termes successifs de la suite ;
- La variable K est un compteur de boucles ;
- La variable N permet d'arrêter le programme.

Lors de l'affichage des termes de la suite, on affiche une espace entre deux valeurs renvoyées.



# **Tangente** **ÉDUCATION**

Trimestriel - n°15 janvier 2011

## **Spécial programmation**

Publié par les Éditions POLE, SAS au capital de 42 000 euros  
80 bd Saint-Michel - 75006 Paris  
Commission paritaire : 0112 R 89189  
Dépôt légal à parution

Directeur de Publication et de la Rédaction  
Coordination du numéro  
Gilles COHEN

Secrétaire de rédaction : Édouard THOMAS

Rédaction du numéro : Jean-Alain RODDIER

Maquette : au journal

Marketing et Publicité  
Mail : [pub@poleditions.com](mailto:pub@poleditions.com)

Abonnements : Tél. : 01 47 07 99 06  
Fax : 01 47 07 88 13

Ce numéro de Tangente-Éducation  
a été imprimé par Imprimerie Belz  
85000 - La Roche-sur-Yon