



## *Suite de Fibonacci*

Auteur : Hélène Landart

Niveaux scolaires :  
Première S

Mots clés :  
Algorithme/Programmation

### ENONCE

Déterminer les termes de la suite de Fibonacci jusqu'au terme d'indice n.

La suite de Fibonacci est définie par :

$$\begin{cases} u_0 = 0 \\ u_1 = 1 \\ u_{n+2} = u_{n+1} + u_n \end{cases}$$

Pour obtenir la liste des termes de la suite jusqu'au terme de rang n, on réalise le programme suivant :

```
Fibo.py 001/012 ▶
def F(n):
    S=0
    T=1
    if n==0:
        print("U0=",S)
    elif n>=1:
        print("U0=",S,"\\nU1=",T)
    for i in range(2,n+1):
        U=S+T
        S=T
        T=U
        print("U",i,"=",T)
FILE RUN SYMBOL CHAR A↔a ▶
```

Dans ce programme :

- S contient la valeur du terme  $u_n$ ,
- T contient la valeur du terme  $u_{n+1}$ ,
- U contient la valeur du terme  $u_{n+2}$ .

# Graph 90+E



## Le principe du programme :

Pour chaque valeur de  $i$ , on met dans :

- U la somme des termes de S et T (U contient alors la nouvelle valeur de T),
- S le terme suivant de  $S \rightarrow T$ ,
- T le terme suivant de  $T \rightarrow U$ .

Pour tester le programme, il ne reste plus qu'à aller dans l'onglet {RUN}. Ne pas oublier d'enregistrer le programme.

Appeler ensuite la fonction "F" en précisant la valeur de  $n$ .

Prenons, par exemple,  $n = 4$ .

```
Fibo.py 012/012
elif n>=1:
    print("U0=",S,"\n")
    for i in range(2,n+1):
        U=S+T
        S=T
        T=U
    print("U", i, "=", S)
```

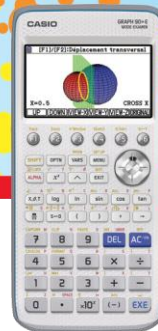
```
>>>F(4)
U0= 0
U1= 1
U 2 = 1
U 3 = 2
U 4 = 3
>>>
```

Remarque : il est aussi possible d'utiliser les listes ou la récursivité pour les termes de la suite de Fibonacci.

- Avec les listes : la liste L contiendra les termes successifs de la suite de Fibonacci jusqu'au terme de rang  $n$ .

```
Fibolist.py 001/010
def F(n):
    L=[0]
    if n==0:
        return L
    elif n==1:
        return L+[1]
    L=[0,1]
    for i in range(2,n+1):
        L=L+[L[i-1]+L[i-2]]
    return L
```

# Graph 90+E



Cela donne alors la présentation suivante pour les résultats :

```
MicroPython v1.9.4
|CASIO COMPUTER CO.,
>>>from Fibolist impo
>>>F(5)
[0, 1, 1, 2, 3, 5]
>>>
```

RUN A↔a CHAR

- Avec la récursivité : pour avoir plus rapidement le terme d'indice n.

```
Fiborecu.py 001/008 ▶
def F(n):
    L=[0]
    if n==0:
        return 0
    elif n==1:
        return 1
    else:
        return(F(n-1)+F(n-2))
```

FILE RUN SYMBOL CHAR A↔a ▶