Suite de Fibonacci : écriture de fonctions

Voici la structure que doit avoir un programme pour lequel le calcul de l'élément d'indice n de la suite de Fibonacci est encapsulé dans une fonction :

<sxh python; title : fibonacci05_fonction.py> #! /usr/bin/env python # -*- coding: utf-8 -*- “”“ Calculs des premiers éléments de la suite de Fibonacci. Référence : http://fr.wikipedia.org/wiki/Suite_de_Fibonacci ”“” def fibonacci_item(n):

```
    """
    Renvoie l'élément d'indice n de la suite de Fibonacci
    """
    ...
```

if name == 'main':

```
    # le programme "principal" ....
```

</sxh>

Le rôle de la structure conditionnelle **if __name__ == '__main__':** est de n'exécuter la suite du code **que** si le programme python concerné est le programme principal. Il se peut en effet que ce fichier soit appelé en tant que module par une directive d'importation écrite dans un autre programme. Dans ce dernier cas, le code qui suit la ligne **if __name__ == '__main__':** ne sera pas lancé, mais toutes les fonctions définies seront reconnues et utilisables par le programme appelant !

Voici une proposition complète : <sxh python; title : fibonacci05_fonction.py> #! /usr/bin/env python # -*- coding: utf-8 -*- “”“ Calculs des premiers éléments de la suite de Fibonacci. Référence : http://fr.wikipedia.org/wiki/Suite_de_Fibonacci ”“” def fibonacci_item(n):

```
    """
    Renvoie l'élément d'indice n de la suite de Fibonacci
    """
    a, b = 0, 1
    if n==0:
        return a
    elif n==1:
        return b
    for i in range(1,n):
        a, b = b, a + b
    return b
```

if name == 'main':

```
    i=input("Suite de Fibonacci. Donnez l'indice de l'élément souhaité ? ")
    print ("Élément de la suite : "),
    print fibonacci_item(i)
    print ('Premiers éléments de la suite : ')
    for j in range(10):
```

```
      print j,fibonacci_item(j)
```

</sxh>

On peut compléter les fonctionnalités par une fonction **fibonacci_list(n)** qui génère et renvoie la liste des éléments de la suite de Fibonacci jusqu'à l'élément n inclus. Finalement, on peut aussi proposer des alternatives (aussi efficace ?) sous forme de fonctions appelant d'autres fonctions, avec **fibonacci_list_from_items(n)** qui construirait la liste à partir de la fonction donnant un élément particulier, et **fibonacci_item_from_list(n)** qui renverrait l'élément d'indice n comme dernier élément de la liste !

Voici ce que cela donne : <sxh python; title : fibonacci06_fonctions.py> #! /usr/bin/env python # -*- coding: utf-8 -*- “”“ Calculs des premiers éléments de la suite de Fibonacci. Référence : http://fr.wikipedia.org/wiki/Suite_de_Fibonacci ”“” def fibonacci_item(n):

```
  """
  Renvoie l'élément d'indice n de la suite de Fibonacci
  """
  a, b = 0, 1
  if n==0:
      return a
  elif n==1:
      return b
  for i in range(1,n):
      a, b = b, a + b
  return b
```

def fibonacci_list(n):

```
  """
  Renvoie la liste des éléments de la suite de Fibonacci jusqu'à l'élément n
inclus.
  """
  a,b,ans = 0,1,[0,1]
  if n==0:
      return [0]
  for i in range(1,n):
      a, b = b, a + b
      ans.append(b)
  return ans
```

def fibonacci_list_from_items(n):

```
  # construit la liste à partir de la fonction donnant un élément
  ans = []
  for i in range(n+1):
      ans.append(fibonacci_item(i))
  return ans
```

def fibonacci_item_from_list(n):

```
  # renvoie l'élément d'indice n comme dernier élément de la liste
  return fibonacci_list(n)[n]
```

if name == 'main':

```
  i=input("Suite de Fibonacci. Donnez l'indice de l'élément souhaité ? ")
  print ("Élément de la suite : "),
  print fibonacci_item(i)
  print ('Premiers éléments de la suite : ')
  for j in range(10):
      print j,fibonacci_item(j)
```

```
  print ('Avec fibonacci_item_from_list : ')
  for j in range(10):
      print j,fibonacci_item_from_list(j)
```

```
  print ("Liste des éléments de la suite de Fibonnaci jusqu'à l'élément
souhaité, inclus : "),
  print fibonacci_list(i)
  print ('Premières listes : ')
  for j in range(10):
      print j,fibonacci_list(j)
```

```
  print "Avec fibonacci_list_from_items"
  for j in range(10):
      print j,fibonacci_list_from_items(j)
```

</sxh>

Des fonctions qui appellent d'autres fonctions ! Mais que voilà une idée intéressante, qui peut déboucher sur une écriture récursive d'une fonction donnant l'élément d'indice n de la suite de Fibonacci quir par définition est la somme de l'élément d'indice n-1 de la suite de Fibonacci, et de l'élément d'indice n-2 de la suite de Fibonacci !

<sxh python; title : fibonacci07_fonction_recursive.py> #! /usr/bin/env python # -*- coding: utf-8 -*- “”” Calculs des premiers éléments de la suite de Fibonacci. Référence : http://fr.wikipedia.org/wiki/Suite_de_Fibonacci Application de la définition par récursivité. ””” def fibonacci_item_recursive(n):

```
  """
  Renvoie l'élément d'indice n de la suite de Fibonacci
  """
  ... (?)
  return fibonacci_item_recursive(n-1)+fibonacci_item_recursive(n-2)
```

if name == 'main':

```
  ...
```

</sxh>

Pour la suite, cliquez ici !