

Énoncé

On place une somme d'argent initiale sur un livret d'épargne qui rapporte 2,5 % par an.

- On note (C_n) la suite, définie pour tout $n \in \mathbb{N}$, représentant le capital obtenu au bout de n années. On fixe $C_0 = 1500$ €. Quelle est l'expression et la nature de cette suite ?
- Au bout de 10 ans, quelle sera la valeur du capital si on ne retire ou n'ajoute aucune somme supplémentaire ? On souhaite réaliser un script en Python pour répondre à cette question qui consistera à définir la fonction **C** qui renverra le capital acquis au bout de n années où n est passé en paramètre.
- On souhaite savoir au bout de combien de temps le capital acquis dépassera le montant de 2500 €. Définir la fonction **Annee** qui renvoie le nombre d'années nécessaire pour dépasser le capital c passé en paramètre.

1. Définir la suite (C_n)

Le capital va être augmenté chaque année de 2,5%. Exprimons d'abord le coefficient multiplicateur correspondant à cette augmentation.

$$CM = 1 + \frac{2,5}{100} = 1,025$$

Ainsi à l'issue de la première année :

$$C_1 = C_0 \times CM = C_0 \times 1,025 = 1537,50 \text{ €}.$$

À l'issue de la deuxième année :

$$C_2 = C_1 \times CM = C_1 \times 1,025 = 1575,9375 \text{ €}.$$

Et ainsi de suite pour les années suivantes. Chacun des termes successifs de la suite est obtenu en multipliant le terme précédent par CM . Ainsi (C_n) est une suite géométrique de raison 1,025 et de premier terme $C_0 = 1500$.

```

NORMAL FLOTT AUTO RÉEL RAD MP
1500*1.025
1537.5
1537.5*1.025
1575.9375

```

2. Définir la suite (C_n) en Python

On crée le script **Capital** de type **Calculs Mathématiques**, à l'intérieur duquel on va définir la fonction **C** de paramètre **n** et qui renverra le terme C_n définie à la question précédente.

Pour cela on va utiliser, ce qu'on appelle en algorithmie, un accumulateur multiplicatif jouer par la variable **u**.

À la fin de l'exécution de la boucle, la fonction **C** renvoie la valeur accumulée dans **u**.

L'emploi de la boucle **for** est adapté à notre besoin puisque dans cette question, on connaît le nombre d'itération que l'on souhaite réaliser. Ce nombre nous est donné par le paramètre **n**.

En console, on retrouve les résultats précédents et on obtient un capital d'environ 1920 € au bout de la 10^{ème} année.

```

ÉDITEUR : CAPITAL
LIGNE DU SCRIPT 0008
# Calculs Mathématiques
from math import *

def C(n):
    u = 1500
    for i in range(n):
        u = u*1.025
    return u

```

```

PYTHON SHELL
>>> # L'exécution de CAPITAL
>>> from CAPITAL import *
>>> C(0)
1500
>>> C(1)
1537.5
>>> C(2)
1575.9375
>>> C(10)
1920.126816294535
>>> |

```

2. Définir Annee en Python

On complète le script **Capital** avec la fonction **Annee** de paramètre **c** et qui renvoie le nombre d'année nécessaire pour que la suite dépasse **c**.

Cette fois-ci, on va utiliser une structure de boucle de type **while** puisqu'on ne connaît pas à l'avance le nombre d'itération. On connaît en revanche le seuil à dépasser puisqu'il nous est indiqué en paramètre.

Une fois la fonction saisie, on vérifie en console que le seuil de 1500€ est dépassé dès $n = 0$ puisqu'il s'agit de la valeur initiale de notre suite.

Il faut une année pour dépasser le seuil de 1530€.

Il faut dix années pour dépasser le seuil de 1920€.

Enfin, le seuil de 2500€ sera dépassé au bout de 21 années.

Un prolongement peut être proposé. En réalité, la fonction **c** peut être assez facilement généralisée pour le calcul de n'importe quel terme d'une suite géométrique.

Pour cela, il suffit de passer en paramètre, en plus du rang souhaité, le premier terme **u0** de la suite géométrique ainsi que sa raison **q**.

Appelons **G** notre nouvelle fonction de paramètre **u0**, **q** et **n**.

De la même façon, il suffit de modifier **Annee** en ajoutant les paramètres **u0** et **q** pour la transformer en une fonction de seuil « universelle » pour suite géométrique.

Appelons **Seuil** notre nouvelle fonction de paramètre **u0**, **q** et **c**.

Notre script est désormais enrichi des fonctions ci-contre. Testons les en console.

Ces nouvelles fonctions permettent de tester immédiatement de nouveaux paramètres au problème.

Ainsi **G(2000,1.03,10)** permet de voir qu'au bout de 10 ans, 2000€ placés à 3% l'an garantissent un capital acquis d'environ 2688€.

On retrouve bien qu'il faudra 21 années pour dépasser le seuil de 2500€ si l'on place 1500€ à 2,5% d'intérêts à l'aide de la commande **Seuil(1500,1.025,2500)**.

Tandis que la commande **Seuil(500,1.025,2500)** nous apprend qu'il faudra 66 années pour dépasser ce seuil de 2500€.

De même si on place 2000€ à 1% l'an, le capital sera de 2209€ au bout de 10 ans et si l'on place 1000€ à 2,5% pendant 10 ans également, le capital sera de 1280€.

```
ÉDITEUR : CAPITAL
LIGNE DU SCRIPT 0014

def C(n):
    u = 1500
    for i in range(n):
        u = u*1.025
    return u

def Annee(c):
    n = 0
    while C(n)<c:
        n += 1
    return n
```

```
PYTHON SHELL

>>> # L'exécution de CAPITAL
>>> from CAPITAL import *
>>> Annee(1500)
0
>>> Annee(1530)
1
>>> Annee(1920)
10
>>> Annee(2500)
21
>>> |
```

```
ÉDITEUR : CAPITAL
LIGNE DU SCRIPT 0026

def G(u0,q,n):
    u = u0
    for i in range(n):
        u = u*q
    return u

def Seuil(u0,q,c):
    n = 0
    while G(u0,q,n)<c:
        n += 1
    return n
```

```
PYTHON SHELL

>>> G(2000,1.03,10)
2687.832758688244
>>> Seuil(1500,1.025,2500)
21
>>> Seuil(500,1.025,2500)
66
>>> G(2000,1.01,10)
2209.244250822409
>>> G(1000,1.025,10)
1280.084544196357
>>> |
```