

Calcul d'une mensualité d'un prêt



Dans un script EMPRUNT

1°) On emprunte une somme c à un taux d'intérêt mensuel de $t\%$ avec une mensualité de m .

Ecrire une fonction `u` qui prend comme arguments c , t , m et n et qui renvoie le capital restant dû après n mensualités.

Application : Calculer le capital restant dû si on emprunte 30000€ au taux annuel de 5% avec une mensualité de 1100€ après 24 mois de remboursement.

```
PYTHON SHELL
>>> # L'exécution de EMPRUNT
>>>
>>> # Shell Reinitialized
>>>
>>>
>>> from EMPRUNT import *
>>> u(30000,5/12,1100,24)
```

2°) Ecrire une fonction `calculM` qui prend comme arguments c , t et n et qui renvoie la mensualité (à l'euro près) correspondant à un emprunt d'une somme c à un taux d'intérêt mensuel de $t\%$ avec n mensualités.

Indication : On pourra initialiser m à c/n et incrémenter m de 1€ tant que le capital restant dû (après n mensualités) est positif.

Application : Calculer la mensualité d'un emprunt de 30.000€ sur 24 mois au taux annuel de 5%.

```
PYTHON SHELL
>>> calculM(30000,5/12,24)
>>> u(30000,5/12,1100,24)
-21.61727652626814
>>> u(30000,5/12,1100,24)
3.568644007726334
```

3°) Ecrire une fonction `cout` qui prend comme arguments c , t et n et qui renvoie le coût du crédit.

Application : Calculer le coût du crédit pour un capital de 30.000€ sur 24 mois au taux annuel de 5%.

4°) Ecrire une fonction `pourcent` qui prend comme arguments c , t et n et qui renvoie le pourcentage du coût du crédit par rapport à la somme empruntée.

Application : Quel est le pourcentage du coût du crédit par rapport au capital emprunté pour le crédit suivant :

30.000€ à 5% annuel sur 24 mois.

Qu'en est-il si on emprunte sur 60 mois ?

5°) Question subsidiaire :

Les banques utilisent la règle (fausse) suivante :

Le taux mensuel = taux annuel / 12

Pour un emprunt de 250.000€ à 3% annuel sur 240 mois, calculer le coût du crédit avec la méthode de calcul de la banque puis avec la méthode mathématiquement correcte (qui consiste à calculer le taux mensuel équivalent).

Quelle est la somme dont l'emprunteur s'est fait spolier ?



Calcul d'une mensualité d'un prêt

Fonction `u`

On initialise la variable `d` à `c` le montant dû initialement.

Pour chacun des `n` mois, le capital dû augmente de `t`% moins la mensualité remboursée.

Il suffit de faire ce calcul `n` fois à l'aide d'une boucle `for`.

Application : Calculer le capital restant dû si on emprunte 30.000€ au taux annuel de 5% avec une mensualité de 1100€ après 24 mois de remboursement.

On trouve un capital restant dû de 5443,73€.

```
ÉDITEUR : EMPRUNT
LIGNE DU SCRIPT 0001
def u(c,t,m,n):
    d=c
    for i in range(n):
        d=d*(1+t/100)-m
    return d
```

```
>>> from EMPRUNT import *
>>> u(30000,5/12,1100,24)
5443.727479351415
```

Fonction `calculM`

2°) On initialise la variable `m` qui représente la mensualité recherchée à `c/n` (qui correspond à la mensualité d'un prêt à taux 0%). On pouvait aussi initialiser `m` à 0 mais cela va prendre plus de temps à la fonction pour trouver `m`.

Tant que le capital restant dû à la fin des `n` mois est positif on incrémente `m` de 1.

À la fin de cette boucle on renvoie `m` qui va correspondre à la plus petite mensualité pour laquelle le capital restant dû est négatif. On pourrait aussi renvoyer `m-1` la plus grande mensualité pour laquelle le capital restant dû est positif.

Application : Calculer la mensualité d'un emprunt de 30.000€ sur 24 mois au taux annuel de 5%.

On trouve 1.317€ qu'on peut vérifier en calculant le capital restant dû pour une mensualité de 1.316€ et 1.317€.

Remarque 1 : On pourrait améliorer le script de cette fonction pour obtenir une valeur approchée de la mensualité au centime près en procédant par approximations successives : à l'euro près comme nous venons de le faire, puis au dixième d'euro près (en initialisant `d` à la mensualité obtenue à l'euro près précédente -1) et en incrémentant `m` de 0,1 à chaque tour de boucle et recommencer avec un incrément pour `d` de 0,01€.

Remarque 2 : Si on dispose de la somme des termes d'une suite géométrique ou du raisonnement par récurrence on démontre que

$$m = c \times \frac{\frac{t}{100}}{1 - \left(1 + \frac{t}{100}\right)^{-n}}$$

```
ÉDITEUR : EMPRUNT
LIGNE DU SCRIPT 0016
def calculM(c,t,n):
    m=c/n
    while u(c,t,m,n)>0:
        m=m+1
    return m
```

```
PYTHON SHELL
>>> calculM(30000,5/12,24)
1317.0
>>> u(30000,5/12,1317,24)
-21.61727652626814
>>> u(30000,5/12,1316,24)
3.568644007726334
```

Calcul d'une mensualité d'un prêt

Fonction `cout`

Grâce à la fonction `calculM` qui permet de calculer la mensualité, le coût est rapidement obtenu en faisant `n*m` (qui correspond à ce que l'emprunteur a remboursé à la fin des `n` mois) moins `c` le capital emprunté.

```
def cout(c,t,n):
    m=calculM(c,t,n)
    return m*n-c
```

```
>>> cout(30000,5/12,24)
1608.0
```

Fonction `pourcent`

Le pourcentage recherché étant `cout/c*100` la fonction s'écrit facilement :

```
def pourcent(c,t,n):
    a=cout(c,t,n)
    return a/c*100
```

Application : Quel est le pourcentage du coût du crédit par rapport au capital emprunté pour le crédit suivant :

30.000€ à 5% annuel sur 24 mois.

On trouve 5,36%.

Qu'en est-il si on emprunte sur 60 mois ?

On trouve 13,4%.

```
>>> pourcent(30000,5/12,24)
5.36
>>> pourcent(30000,5/12,60)
13.4
```

Question subsidiaire

On calcule tout d'abord le coût avec un taux mensuel de $t = \frac{3}{12}\%$ et on obtient 82.800€.

Le taux mensuel équivalent à un taux annuel de 5% est

$$t = \left(\left(1 + \frac{5}{100} \right)^{\frac{1}{12}} - 1 \right) \times 100$$

Ce qui donne $t \approx 0,4074\%$ (contre $\frac{5}{12} = 0,4167\%$)

Ainsi le coût de l'emprunt pour ce taux est 81.600€.

La banque a donc gagné 1.200€ avec ce « petit détail technique ».

```
PYTHON SHELL
>>>
>>> # L'exécution de EMPRUNT
>>>
>>> # Shell Reinitialized
>>> from EMPRUNT import *
>>> cout(250000,3/12,240)
82800.000000000001
>>> t=((1+3/100)**(1/12)-1)*100
>>> cout(250000,t,240)
81600.0
>>> |
Fns... a A # Outils Éditer Script
```