

<https://www.mesmaths.com/spip.php?article435>



TP Python n°2

- 2nde : Maths - Activités -

Date de mise en ligne : samedi 31 août 2019

Copyright © www.mesmaths.com - Tous droits réservés

situation

Voici un programme de calcul :

1. prendre un nombre
2. le multiplier par 0,75
3. lui soustraire 2

Question 1 : faire plusieurs essais "à la main" en prenant comme nombre au départ : 20 ; 13 ; 7,75

réponse

$$20 * 0,75 - 2 = 13$$

$$13 * 0,75 - 2 = 7,75$$

$$7,75 * 0,75 - 2 = 3,8125$$

Question 2 : faites ces mêmes calculs dans une console python (pyzo ou calculatrice) .

réponse

```
>>> 20*0.75-2
13.0
>>> 13*0.75-2
7.75
>>> 7.75*0.75-2
3.8125
```

amélioration

Vous avez sans doute remarqué que les calculs étaient répétitifs ! et on a bien envie de trouver une façon de faire

plus efficace.

Pour cela, on va créer une **fonction** au sens informatique du terme dans la zone qui se nomme **script** ; sur la calculatrice, on va **éditer** le script (en nommant un programme).

En voici une syntaxe possible :

```
def prog(a):  
    a = a * 0.75  
    a = a - 2  
    return a
```

remarque : il est possible d'avoir des syntaxes plus efficaces ; vous pouvez essayer.

Question 1 : saisir ce code et l'exécuter ; que se passe-t-il ?

réponse

Rien !

Il faut bien comprendre l'idée : vous avez construit une fonction ; le programme une fois compilé et exécuté, la fonction est prête à l'emploi ... et n'attend que ça. Si on ne fait rien, il ne se passe rien !

Question 2 : une fois le programme compilé, saisir prog(20) dans la console ; que se passe-t-il ?

Aide :

- **Pyzo** : pour exécuter -> **Ctrl+shift+E**
- **calculatrice** : taper **Exéc** ; on retrouve la fonction prog() en tapant sur la touche **var**

réponse

```
>>> prog(20)  
13.0
```

répétitions

Question 1 : trouver une manière d'appliquer successivement trois fois de suite le programme de calcul précédent en partant du nombre 20

réponse

```
>>> a = 20
>>> a = prog(a)
>>> a = prog(a)
>>> a = prog(a)
>>> a
3.8125
```

ou alors :

```
>>> prog(prog(prog(20)))
3.8125
```

Vous serez d'accord pour constater que les deux méthodes proposées ne sont pas très efficaces ...

On va à présent utiliser une boucle qui va permettre de répéter 3 fois ce programme de calcul ; voici la syntaxe qui signifie "répéter 3 fois" :

```
for i in range(3) :
```

Question 2 : en initialisant une variable à la valeur 20, faire exécuter dans la console trois fois de suite la fonction prog() à la variable initialement égale à 20.

réponse

```
>>> a=20
>>> for i in range(3) :
...     a = prog(a)
...
...
>>> a
3.8125
```

amélioration2

On va améliorer cette syntaxe en créant une nouvelle fonction (dans la zone script) que l'on va nommer 'rep' (pour

répéter) ; on voudra qu'elle répète la fonction prog() autant de fois que l'on veut, sur un nombre choisi par l'utilisateur.

Question 1 : combien cette fonction va-t-elle avoir de paramètres ?

réponse

2 :

- un premier paramètre qui sera la valeur initiale de la variable (20 dans notre exemple)
- un second paramètre qui sera le nombre de répétitions (3 dans notre exemple)

On pourra les nommer a (comme précédemment) et n (qui sera le nombre de répétitions).

Attention, il faudra bien respecter l'ordre de ces paramètres dans les saisies ultérieures

Question 2 : compléter le script précédent par une fonction rep(a,n) qui permettra d'appliquer à une valeur a le programme de calcul précédent n fois.

Vérifier en particulier que rep(20,3) donne bien la valeur 3.8125

réponse

```
def rep(a,n):  
    for i in range(n):  
        a = prog(a)  
    return a
```

bilan

Créer vous une fiche en donnant des 'définitions' pour les mots suivants :

remarque : l'ordre des mots et volontairement peu logique ... à vous d'en regrouper certains !



- fonction (au sens informatique)
- console
- range()
- paramètre(s)
- indentation
- return
- boucle for
- script
- def()
- :
- autres ??