

JORRO Fabienne et Rémi

Professeurs de mathématiques

Lycée Albert Camus– Fréjus 83600 – Var

Outils : Support numérique : soit en salle informatique, soit en salle de cours avec utilisation possible des calculatrices pour les scripts python ou d'une tablette ou encore d'un smartphone avec une application dédiée Python ou en ligne.

Nature : Bilans de connaissances et de compétences en lien avec les attendus de fin d'année (COVID)

Objectifs pédagogiques : Consolider la notion de fonction en informatique.

Voie : générale - technologique

Niveau de classe : Seconde

Thématique(s) du programme : Algorithmique et programmation ; notion de fonction. Écrire des fonctions simples ; lire, comprendre, modifier, compléter des fonctions plus complexes. Appeler une fonction.

Lire et comprendre une fonction renvoyant une moyenne.

Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire.

Résumé de l'article :

Bilan sur la partie « Notion de fonction » du programme de seconde avec un petit memento reprenant les syntaxes à connaître sur les fonctions, les tests ainsi que les boucles.

Un support aux exercices (via Qrcode et le site repl.it) permet aux élèves d'avoir les scripts pré-complétés ainsi que la possibilité de compiler leurs programmes.

L'ensemble des programmes complétés est disponible ici : <https://repl.it/@JORRO/les-programmes>

Pour les utilisateurs de JupyterLab (notamment sur tablette), un fichier source est disponible reprenant l'ensemble version élève et version professeur (double clic pour ouvrir les fichiers).



Petit memento sur les fonctions, tests et boucles

- Les fonctions :

```
def Ma_Fonction(x,y,...): # Entrées
    instructions
    .....
    return (...) # Le retour de la fonction
```

```
def somme(a,b): # les paramètres sont a et b
    y = a + b # calcul de la somme
    return y # la fonction retourne la somme
```

- Les tests :

- $x == y$ est vrai quand x est égal à y ,
- $x != y$ est vrai quand x est différent de y ,
- $x > y$ est vrai quand x est strictement supérieur à y ,
- $x < y$ est vrai quand x est strictement inférieur à y ,
- $x \geq y$ est vrai quand x est supérieur ou égal à y ,
- $x \leq y$ est vrai quand x est inférieur ou égal à y .
- On peut « tester » sur un intervalle :
 $2 < x \leq 3$ est vrai quand $x \in]2; 3]$

```
if (condition):
    instructions
else:
    instructions
```

```
if (condition):
    instructions
elif (condition): #on peut avoir plusieurs elif
    instructions
else:
    instructions
```

- Boucle bornée :

```
for i in range(n): # 0 <= i < n
    instructions # instruction faites n fois
```

- Boucle non bornée :

```
while (condition): #tant que la condition est vraie on fait les instructions
    instructions
```

Exercice 1

Voici le script d'une fonction :

```
def fonction1(a,b):
    resultat = (a + b)/2
    return (resultat)
```

1°) Combien d'argument(s) la fonction possède(nt)-t-elle ?

2°) Que retourne cette fonction quand on saisit :

a) `>>> fonction1(5,8)`

b) `>>> fonction1(-4,9)`

3°) Que retourne cette fonction ?

4°) Ecrire une fonction *Milieu* qui prend comme arguments les coordonnées de deux points *A* et *B* et qui renvoie les coordonnées du milieu du segment $[AB]$.



Exercice 2

Voici le script d'une fonction :

```
from random import *
def fonction2():
    return randint(0,1)
```

1°) Quel est le rôle de l'instruction sur la première ligne ?

2°) Utiliser dix fois cette fonction en tapant `>>> fonction2()`. Que constate-t-on ?

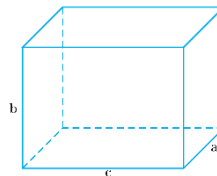
3°) Dans quelle situation cette fonction peut-elle avoir un intérêt ?

4°) Modifier le script de la fonction pour qu'elle simule le lancer d'un dé cubique équilibré.



Exercice 3

Une boîte en forme de pavé droit a pour dimensions a , b et c .



On a le script suivant :

```
def fonction3(a,b,c):  
    return a*b*c
```

1°) Que retourne *fonction3(4,3,10)* ?

2°) Que fait ce script ?

3°) Modifier le script afin qu'il retourne l'aire latérale de cette boîte.



Exercice 4

Un magasin loue des trottinettes électriques avec les tarifs suivants :

- Moins de 6 heures : 2 € l'heure et 1 € de frais de dossier,
- A partir de 6 heures : 1,5 € l'heure et 4 € de frais de dossier.

1°) Calculer le prix à payer pour 4 heures de location puis pour 8h de location.

2°) On note P la fonction qui, au nombre x d'heures de location, associe le prix à payer en euros.

Déterminer l'expression de $P(x)$ selon les valeurs de x suivantes :

- Si $0 < x < 6$, alors $P(x) = \dots$
- Si $6 \leq x$, alors $P(x) = \dots$

3°) Compléter le script ci-contre afin que la fonction *prix* retourne le prix à payer pour x heures de location.

```
def prix(x):  
    if 0 < x < 6:  
        p = .....  
    else:  
        p = .....  
    return (p)
```



4°) A l'aide de la fonction *prix* compléter le tableau suivant :

Nombre d'heures de location	4	6	7	8	14	15	16	17
Prix en euros								

5°) Julia dispose de 9 € pour louer une trottinette. Quel nombre maximal d'heures de location peut-elle envisager ? Même question pour Yannis qui a un budget de 26 € ?

6°) Compléter la fonction ci-contre qui prend comme argument un budget en € et qui retourne le nombre maximal possible d'heures de location.

```
def nombre_heures(budget):  
    h = 0  
    while .....  
        h = h + 1  
    return .....
```

Exercice 5

Le 25 mai 2020, Jules installe son nouvel aquarium d'une contenance de 400 litres. Par évaporation, le volume d'eau dans l'aquarium diminue de 2% par semaine. Compte tenu du nombre de poissons, cet aquarium doit maintenir en permanence au moins 360 litres d'eau.

Jules ajoute chaque lundi 7 litres d'eau pour compenser l'évaporation de 2%.

1°) On considère la fonction ci-contre qui renvoie le nombre de litres d'eau restant au bout de n semaines. Compléter le script.

```
def aquarium(n):  
    v = 400  
    for k in range(...):  
        v = .....  
    return v
```

2°) Peut-on dire que la préconisation concernant le volume d'eau est respecté au bout d'un an ? au bout de 2 ans ?

3°) Compléter la fonction suivante afin de déterminer au bout de combien de semaines il y aura moins de 360 litres d'eau dans l'aquarium.

```
def seuil_aqua():  
    semaine = 0  
    while ..... :  
        semaine = semaine + 1  
    return semaine
```

