

Fonction à un ou plusieurs arguments

Exercice 1
On considère la fonction suivante écrite dans le langage Python :

```
def f(x):  
    return 3*x - 1
```

- 1. Que renvoie $f(1)$? $f(2)$?
- 2. Que permet de faire cette fonction ?

Exercice 2
En prévision des soldes, un commerçant s'apprête à modifier ses étiquettes. Pour cela, il compte utiliser le programme suivant :

```
def solde(p,t):  
    return (1-t/100) * p
```

- 1. Quelle est la valeur renvoyée si on exécute `solde(40,30)` ?
- 2. Quel est le rôle de cet algorithme ?
- 3. Que doit saisir le commerçant pour connaître le nouveau prix d'un article valant 55€ et soldé à -60% ?

Exercice 3
La distance d'arrêt d d'un véhicule roulant à une vitesse v (exprimée en km/h) sur route mouillée est donnée par la formule $d = 0,01v^2 + 0,28v$. Écrire une fonction en langage Python qui, à partir de la donnée de la vitesse d'un véhicule, renvoie sa distance d'arrêt.

Exercice 4
1. Ecrire en langage Python une fonction nommée `dper` dont les paramètres sont trois nombres a , b et c représentant les côtés d'un triangle et qui renvoie le demi-périmètre de ce triangle.
2. Le mathématicien grec Héron d'Alexandrie a établi la formule suivante qui donne l'aire s d'un triangle de côtés a , b et c et de demi-périmètre p : $s = \sqrt{p(p-a)(p-b)(p-c)}$. Ecrire une fonction en Python permettant le calcul de l'aire d'un triangle en faisant appel à la fonction `dper` de la question 1.

Exercice 5
On rappelle qu'en langage Python le connecteur logique « et » s'écrit `and` et que le connecteur logique « ou » s'écrit `or`.

- 1. Compléter le programme suivant afin que la fonction `test` renvoie un booléen de valeur `True` lorsque le nombre x est dans l'intervalle $]2; 7[$.

```
def test(x):  
    return x > ..... and x < .....
```

- 2. En utilisant les connecteurs logiques `and` et `or`, écrire le programme d'une fonction `test2` d'argument un nombre x qui renvoie un booléen de valeur `True` si le nombre x appartient à la réunion des intervalles $[1 : 9]$ et $[14; 20[$.

Exercice 6
Un magasin de location de scooters affiche les tarifs ci-dessous :

Moins de 8 heures :	2€ l'heure et 1€ de frais de dossier
A partir de 8 heures	1,5€ l'heure et 5€ de frais de dossier

- 1. Calculer le prix à payer pour
(a) 5 heures de location (b) 20 heures de location
- 2. On note P la fonction qui, au nombre x d'heures de location associe prix à payer en euros. Compléter les phrases suivantes :
Si $0 < x < 8$ alors $P(x) = \dots$
Si $x \geq 8$ alors $P(x) = \dots$
- 3. Dans le programme ci-dessous, écrit en langage Python, la fonction `prix` doit renvoyer le prix à payer pour x heures de location. Compléter ce programme.

```
def prix(x):  
    if x < 8:  
        p = .....  
    else:  
        p = .....  
    return .....
```

Dans cette question, on suppose qu'il n'est possible de louer un scooter qu'un nombre entier d'heures.

Nombre d'heures de location	6	7	8	9	10	15	16	17	18
Prix (en euros)									

- 4. (a) Recopier et compléter le tableau ci-dessus.
(b) Etienne souhaite louer un scooter le plus longtemps possible mais il ne dispose que de b euros. Pour chaque valeur de b ci-dessous, indiquer le nombre d'heures maximal de location possible :
 $b = 15$
 $b = 28$
(c) En utilisant la fonction `prix` de la question 3., compléter le programme ci-contre afin que la fonction `scooter` renvoie le nombre maximal d'heures de location pour un budget de b euros.

```
def scooter(b):  
    x = 0  
    while ..... :  
        x = x + 1  
    return .....
```

Exercice 7
Écrire une fonction en Python permettant de dire si quatre points forment un parallélogramme. Cette fonction aura pour arguments x_A , y_A , x_B , y_B , x_C , y_C , x_D et y_D , et elle renverra la valeur `True` si $ABCD$ est un parallélogramme et la valeur `False` sinon.

Utilisation des fonctions en probabilités

Exercice 8
Une boîte contient 5 billes rouges, 8 billes noires et 10 billes jaunes. On tire au hasard une bille dans la boîte et on note sa couleur. Compléter le programme suivant afin de simuler cet expérience.

```
import random  
  
def billes():  
    r = random.randint(1,....)  
    if r <= 5:  
        return "La bille tirée est rouge"  
    if ..... :  
        return .....  
    else:  
        return .....
```

Exercice 9
On considère l'expérience aléatoire qui consiste à lancer un dé tétraédrique équilibré dont les faces sont numérotées de 1 à 4 et à noter le numéro obtenu.

- 1. Ecrire en langage Python une fonction `lancer_de` ayant pour paramètre un entier naturel n non nul, qui simule n fois l'expérience du lancer de ce dé et qui retourne la fréquence d'apparition du « 1 ».
- 2. Appeler la fonction précédente en prenant successivement $n = 100$, $n = 1000$, $n = 10000$ et $n = 1000000$. Que constate-t-on ?
- 3. Comment peut-on expliquer ces observations ?

Exercice 10
On considère une expérience aléatoire dont on ne connaît que la loi de probabilité :

Issue	Rouge	Jaune	Vert
Probabilité	0,42	0,27	0,31

- 1. Compléter l'algorithme suivant pour qu'il permette de simuler cette expérience aléatoire :

```
import random

def tirage():
    a = random.random()
    if a <= 0.42:
        return "Rouge"
    if ..... and ..... :
        return .....
    if ..... :
        return .....
```

2. Écrire une fonction `simul` en Python qui permet de faire 100 simulations de cette expérience aléatoire et qui renvoie la fréquence d'apparition de « Rouge ». Cette fonction fera appel à la fonction `tirage` précédente.

Utilisation des fonctions en statistiques

Exercice 11

On donne le programme suivant :

```
def moy(N):
    S = 0
    for k in range(N):
        x = float(input("Valeur ?"))
        n = int(input("Effectif ?"))
        S = S + n * x
    M = S/N
    return M
```

1. Que va renvoyer `moy(4)` si on saisit successivement : 7.5, 2, 8.5, 6, 9, 4, 12.5, 3?
2. Quel est le rôle de la fonction `moy`?
3. Que représente l'argument `N`?

Exercice 12

En Python, une liste est une suite d'objets séparés par une virgule. Par exemple, la variable `L` suivante est une liste contenant tous les entiers de 1 à 5 : `L = [1,2,3,4,5]`.

Compléter la fonction `plage2sigma` suivante afin qu'elle renvoie la proportion des valeurs de la liste `L` appartenant à l'intervalle $[m - 2\sigma; m + 2\sigma]$ où m et σ sont la moyenne et l'écart-type des valeurs de la liste `L` :

```
import statistics

def plage2sigma(L):
    # moyenne de la liste
    m = statistics.mean(L)
    # écart-type de la liste
    sigma = statistics.stdev(L)
    n = len(L) # longueur de la liste
    p = 0
    for valeur in L:
        if valeur >= ..... and valeur <=
            ..... :
            p = .....
    return .....
```