

SERIE : EXERCICES DE FONCTIONS SIMPLES

Application 1

Créez une fonction qui prend deux nombres comme arguments et retourne leur somme.

Exemple:

somme(1, 2) → 3

somme(4, 4) → 8

somme(-1, -1) → -2

```
def somme(a, b):  
    return a + b
```

Application 2

Créez une fonction qui prend deux nombres comme arguments et faire la soustraction.

Exemple:

soustraction(2, 1) → 1

soustraction(4, 2) → 2

soustraction(-3, -3) → 0

```
def soustraction(a, b):  
    return a - b
```

Application 3

Écrivez une fonction qui prend un nombre entier de minutes et le convertit en secondes.

Exemple:

convert(1) → 60

convert(2) → 120

convert(6) → 360

```
def convert(minutes):  
    return minutes * 60
```

Application 4 :

Écrivez une fonction qui convertit les heures en secondes.

Exemple:

convert(1) → 3600

convert(2) → 7200

convert(6) → 21600

```
def convert(heures):  
    return heures * 3600
```

Application 5 :

Créez une fonction qui prend la hauteur et la largeur et trouve le périmètre d'un rectangle.

Exemple:

getPerimeter(2, 4) → 12

getPerimeter(6, 10) → 32

getPerimeter(3, 6) → 18

```
def getPerimeter(hauteur, largeur):  
    return (hauteur + largeur) * 2
```

Application 6 :

Dans ce défi, un fermier vous demande de lui dire combien de pattes peuvent être comptées parmi tous ses animaux. Il y a trois espèces:

poulets = 2 pattes

vaches = 4 pattes

chevaux = 4 pattes

Le fermier a compté ses animaux et il vous donne un sous-total pour chaque espèce. Vous devez implémenter une fonction qui renvoie le nombre total de pattes de tous les animaux.

L'ordre des animaux transmis à la fonction est **nbrsPattes(poulets, vaches, chevaux)**.

Exemple:

nbrsPattes(1, 4, 2) → 26

nbrsPattes(2, 2, 2) → 20

nbrsPattes(2, 0, 3) → 16

N'oubliez pas que le fermier veut connaître le nombre total de pattes et non pas le nombre total d'animaux.

```
def nbrsPattes(poulets, vaches, chevaux):  
    return poulets * 2 + (vaches + chevaux) * 4;
```

Application 7 :

Créez une fonction qui prend une liste de nombres, et renvoie le plus grand nombre de la liste.

Exemple:

max([6, 9, 1, 2]) → 9

max([10, 66, 12, 98]) → 98

max([1, 1, 1, 1, 1]) → 1

```
def getMax(n):  
    return max(n)
```

Application 8:

Créez une fonction qui prend une liste et renvoie la somme de tous les nombres de la liste.

Exemple:

sommeL([1, 2, 3, 4]) → 10

sommeL([0, 0, 0, 1]) → 1

sommeL([-1, -2, 3]) → 0

```
def sommeL(l):  
    return sum(l)
```

Application 9 :

Créez une fonction qui prend une liste de nombres et renvoie le plus petit nombre de la liste.

Exemple:

getMin([9, 6, 1, 2]) → 1

getMin([8, 8, 8, 8]) → 8

getMin([-2, -8, -1]) → -8

```
def getMin(l):  
    return min(l)
```