Math93.com

TD - Les Fonctions
Débuter en Python - Partie 2

Table des matières

Deuxième partie

Les fonctions sous Python

II.1. Présentation des fonctions sous Python

def nom_ fonction(paramètres) :

def nom_ fonction(paramètres) :
instruction 1
instruction 2
...
return valeur

Cela définit une nouvelle fonction, les deux points entraînent une indentation délimitant la déclaration de la fonction.
Le bloc sert à effectuer une série d'actions. Le plus souvent il se termine par *return* pour renvoyer une ou plusieurs valeurs.

Exercice 1

On veut coder une fonction :

$$f(x) = x^2 - x + 41$$

f est le nom de la fonction et x le paramètre (ou argument) de cette fonction
x**2=x*x (notation puissance)

1. Dans l'éditeur PYTHON tapez

```
def f(x):
    valeur=x**2-x+41
    return valeur
```

2. Pour calculer l'image de 5 par f, tapez dans la console f(5). Même chose pour l'image de 8 tapez f(8). Vous devez obtenir l'affichage suivant dans la console python.

```
>>> f(5)
61
>>> f(8)
97
```

   Tester le programme avec des entiers naturels en écrivant dans la console (*à droite sur ripl.it ou votre logiciel IDLE*) directement $f(5)$ ou $f(8)$ par exemple.

3. Dans la fonction ci-dessus, la variable x se nomme paramètre (ou argument) de la fonction. On peut changer ce paramètre à notre guise, vérifiez-le :

```
# Dans l'éditeur PYTHON
def f(a):
    valeur=a**2-a+41
    return valeur
```

4. On peut même renvoyer directement l'image en gagnant une ligne (et une variable) :

```
# Dans l'éditeur PYTHON
def f2(x):
    return x**2-x+41
```

5. Il est possible d'effectuer des calculs directement avec les valeurs renvoyées par la fonction. Essayez :

```
# Dans la console PYTHON
>>> f(5)
61
>>> 2*f(5)+1 # doit renvoyer 2x61 + 1 = 123
123
```

6. Renvoyer un print(), c'est le mal!

ATTENTION

DANS LA MESURE DU POSSIBLE ON PROSCRIRA LE RENVOI D'UNE VALEUR AVEC print

```
# Dans l'éditeur PYTHON
def f3(x):
    valeur=x**2-x+41
    print( valeur) #C'EST TRES MAUVAIS
```

Essayer d'effectuer le calcul suivant comme dans la question 4°) :

```
# Dans la console PYTHON
>>> f3(5)
61
>>> 2*f3(5)+1
61
TypeError: unsupported operand type(s) for *: 'int' and 'NoneType'
```

II.2. Calculer le résultat d'une somme, produit, différence ..

II.2.1 Calculer le résultat d'une somme

Exercice 2

1. Écrire en python la fonction **somme** de paramètres *a* et *b*, qui renvoie la somme de deux nombres *a* et *b*.
2. TESTEZ votre fonction.

```
# Dans l'éditeur PYTHON
def somme(a,b): # cette fonction a 2 paramètres, a et b
        ...
        return ...
```

```
# Les tests dans la console PYTHON
>>> somme(5,3)
8
>>> somme(8,24)
32
```

On peut aussi tester en lançant le programme suivant :

```
# Dans l'éditeur PYTHON
def somme(a,b):
        ...
        return ...
print(somme(5,3)) #L'instruction est en dehors de la fonction.
print( 'somme(8,24)=', somme(8,24) )
```

II.2.2 Calculer le résultat d'une différence

Exercice 3

1. Écrire en python la fonction **difference** de paramètres *a* et *b*, qui renvoie la différence de deux nombres a et b. Remarquez que j'évite les accents dans la définition de ma fonction, c'est important pour la suite.
2. TESTEZ votre fonction.

```
# Dans l'éditeur PYTHON
def difference(a,b):
        ...
        return ...
```

```
# les tests dans la console PYTHON
>>> difference(5,3)
2
>>> difference(8,24)
-16
```

II.2.3 Calculer le résultat d'un produit

Exercice 4

1. Écrire en python la fonction **produit** de paramètres *a* et *b*, qui renvoie le produit de deux nombres a et b.
2. TESTEZ votre fonction.

```
# Dans l'éditeur PYTHON
def produit(a,b):
        ...
        return ...
```

```
# Les tests dans la console PYTHON
>>> produit(5,3)
15
>>> produit(8,24)
192
```

II.3. La division euclidienne

II.3.1 Calculer le quotient de la division euclidienne

Exercice 5

1. Écrire en python la fonction **EuclideQuotient** qui renvoie le quotient de la division euclidienne de deux entiers a et b. Si vous ne connaissez pas la syntaxe de l'opération, consultez le tableau page **??** ou allez chercher dans L'essentiel de python
2. TESTEZ votre fonction.

```
# Dans l'éditeur PYTHON
def EuclideQuotient(a,b):
        ...
        return ...
```

```
# les test dans la console PYTHON
>>> EuclideQuotient(5,3)
1
>>> EuclideQuotient(30,7)
4
```

II.3.2 Calculer le reste de la division euclidienne

Exercice 6

1. Écrire en python la fonction **EuclideReste** qui renvoie le reste de la division euclidienne de deux entiers a et b. Si vous ne connaissez pas la syntaxe de l'opération, consultez le tableau page **??** ou allez chercher dans L'essentiel de python
2. TESTEZ votre fonction.

```
# Dans l'éditeur PYTHON
def EuclideReste(a,b):
        ...
        return ...
```

```
# Les tests dans la console PYTHON
>>> EuclideReste(5,3)
2
>>> EuclideReste(55,7)
6
```

II.4. Des fonctions

II.4.1 Fonction de test

Exercice 7

Voici une fonction **mystere(a,b)** écrite en python.

```
# Dans l'éditeur PYTHON
def mystere(a,b):
        valeur = a>b
        return valeur
```

1. Testez cette fonction avec les valeurs suivantes :

```
# Dans la console PYTHON
>>> mystere(5,3)

>>> mystere(5,5)

>>> mystere(3,5)

>>> mystere(5*5,25)

>>> mystere(3.4-3.3,1)
```

   Que fait cette fonction si on utilise des entiers (int) ou des flottants (float) comme paramètres? Quelle type de variable renvoie-t-elle?

2. Essayez avec les valeurs suivantes :

```
# Dans la console PYTHON
>>> mystere("a","b")

>>> mystere("b","a")

>>> mystere("ab","ac")

>>> mystere("ac","ab")

>>> mystere("manger","mangez")
```

   Que fait cette fonction quand les paramètres sont des chaînes de caractères (str) minuscules sans accent?

II.4.2 Fonction Échange

Exercice 8

1. On donne le programme suivant.

```
# Dans l'éditeur PYTHON
x=1
y=5
print(x,y)
tmp=x
x=y
y=tmp
print (x,y)
```

   Avant d'essayer le programme, estimer les variables x et y après l'exécution de ce programme . Puis VÉRIFIEZ.

2. On donne le programme suivant.

```
# Dans l'éditeur PYTHON
def echange(x,y):
        tmp=x
        x=y
        y=tmp
a=1
b=5
echange(a,b)
print (a,b)
```

   Avant d'essayer le programme, que vaut la variable a après l'exécution de ce programme? Puis VÉRIFIEZ, vous risquez d'avoir une surprise.

3. Que va affichez le programme précédent si on rajoute à la fin l'instruction

   `print( echange(a,b) )`

   Vérifiez. Pourquoi cette affichage?

4. Dans la fonction, rajoutez l'instruction

   `return x,y`

```
# Dans l'éditeur PYTHON
def echange(x,y):
        tmp=x
        x=y
        y=tmp
        return x,y
```

   et testez le programme. Est ce que les valeurs de a et b ont été échangées? Vérifiez en rajoutant

   `print( a,b )`

   à la fin du programme.

5. Que peut on en déduire sur les variables utilisées dans une fonction?

II.4.3 Faire d'autres fonctions

Exercice 9

Coder en python les fonctions suivantes :

1. Le carré d'un nombre
2. La fonction inverse
3. La fonction qui concatène trois mots.

Par exemple

```
>>>tripleConcatenation("do","mi","nation")
domination
```

II.1. Utilisation du module math et docstrings

Remarque

Les fonction mathématiques de base sont présentes dans le *module math* qu'il faut appeler au début du programme sous la forme **import math**.
Pour le nombre π écrire tout simplement : **math.pi**

Astuce : En écrivant **from math import *** au début on importe directement toutes les fonctions et **from math import pi** on importe le nombre pi (enfin une valeur approchée!) ce qui permet d'écrire directement pi (au lieu de math.pi).
Les puristes ont tendances à proscrire l'utilisation de **from math import *** ...

Docstring

Les **docstrings** sont des chaines de caractères qui doivent être placées juste en dessous des définitions de fonction et entre 3 apostrophes ' ' '. On explique ainsi ce que fait la fonction. On indique aussi le typage des variables d'entrée en séparant les noms de variable de leur type par deux points. On indique le type de retour avec "->". Les docstrings sont récupérables dynamiquement avec la fonction **help()**.

```
def fonc(a: int,b:int)->bool:
    '''
    Renvoi True si b > a sinon False
    '''
    return b>a
```

Exercice 10

Écrire une fonction de paramètre r qui renvoie le périmètre d'un cercle de rayon r puis une autre (aussi de paramètre r) qui renvoie l'aire du disque de rayon r.

```
from math import pi

def perimetre(r:float)->float:
        ''' une valeur approchée du périmètre du cercle de rayon r '''
        return ...

def aire(r:float)->float:
        '''une valeur approchée de l'aire du disque de rayon r'''
        return ...
```

II.2. (Optionnel) Input() ou float(input()

nom=input("question") et a=float(input("question"))

- *var=input("question")* :
  La fonction *input*, dont la syntaxe est : *nom=input(question)* affiche une fenêtre où figure le texte *question* et un cadre blanc dans lequel on entrera ce qui est demandé.
  La réponse est alors affectée à la variable *var* qui est alors considérée comme une chaine de caractère (un mot).
- *a=float(input("question"))* : si la valeur de a demandée est considérée comme un nombre flottant,
- *a=int(input("question"))* : pour un entier.

Par exemple :

```
# Dans l'éditeur PYTHON
# On peut définir une fonction sans paramètre,
# juste pour éviter de l'appeler à chaque RUN
def questions()->None:
    nom=input("Donnez votre nom svp ")
    age=int (input("Donnez votre age svp ")    )
    quizz=float (input("Diviser 1 par 4 svp ")    )
    print("Bonjour",nom, "vous avez" ,age, "ans et avez dit que 1/4 donnait" ,quizz)
```

```
# Dans la console PYTHON on obtient :
>>> questions()
Donnez votre nom svp Evariste
Donnez votre age svp 31
Diviser 1 par 4 svp 0.25
Bonjour Evariste, vous avez 31 ans et avez dit que 1/4 donnait 0.25
```

Exercice 11

Faire un quizz. Construire la fonction pythonQuizz() qui pose les questions suivantes :

1. De quelle type est la variable a=3?
2. De quelle type est la variable b=25.6?
3. De quelle type est la variable c=25>2?
4. De quelle type est la variable d="Bonjour"?
5. Quelle est l'instruction qui permet d'afficher un message?

Et qui affiche les réponses de l'utilisateur. Comme par exemple :

```
Vous pensez que en python
a est du type nombre
b est du type décimal
c est du type inégalité
d est du type phrase
un message s'affiche avec l'instruction return
```

Toutes ces réponses sont fausses, appelez le professeur quand vous pensez avoir réussi votre programme et savoir répondre à ces questions.