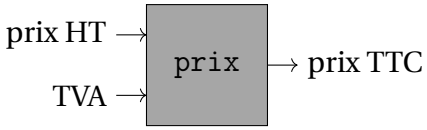
Utilisations des fonctions en Python

Les fonctions, comme les procédures, font partie des routines. Ce sont des portions de programme auxquelles nous avons donné un nom, afin de pouvoir les utiliser ensuite en rappelant juste ce nom.
Une fonction est utilisée comme une boîte noire : nous fournissons des valeurs (les paramètres effectifs) et nous récupérons « quelque chose ».

Exemple : La fonction ci-contre sert à calculer un montant TIC connaissant un prix hors taxes et un taux de TVA.



1 Définition une fonction

Nous définissons une fonction avec un mot clé et son nom en indiquant entre parenthèses ses paramètres formels avec éventuellement leur type, puis en donnant le code de la fonction (son contenu). Une procédure(ou méthode) revoit un affichage et ne peut donc pas être utilisée par une autre fonction.
En reprenant la fonction prix de l'exemple, compléter le script suivant :

```
def prix( ..................):
    """ ........................
    ..............................  """
    ..............................
    return ...................
```

Vous programmerez ce programme sous spyder.
Le texte entre """ et """ s'appelle une docstring
Le mot clé **Renvoyer** (`return`) est est suivi de la valeur que la fonction doit renvoyer. il met fin à la fonction.
Remarque : s'il n'y a pas de renvoie, nous parlons de procédure

Remarque :

Nous éviterons l'usage du `print` car c'est une fonction d'affichage, concrètement vous ne pourrez pas utiliser votre programme dans un autre programme.

Appel d'une fonction

Une fois la fonction définie, on peut l'appeler (pour qu'elle soit exécutée) en donnant son nom et en fournissant des valeurs aux paramètres (que nous appelons alors paramètres réels, paramètres effectifs ou arguments).

Par exemple dans la console, nous rentrons :

```
prix( 10,20)
```

Exercice 1 :

Considérons l'algorithme suivante :

Fonction :Mystère(nom : chaîne de caractère)
..... :
Début
    message ← 'Bonjour ' + nom
    Afficher message
Fin

Que donne l'appel Mystère("Jean")? .....
Compléter le script

Exercice 2 :

Considérons la fonction ci-contre :

1. Que renvoie `Mystere2(2,5)`?
2. Que renvoie `Mystere2(6,13)`?
3. Implémenter cette fonction en Python
   (a) Refaire les deux appels précédents
   (b) Justifier que `c=b`.

Fonction :Mystere2(a,b : nombres)
Rôle du calcul
Début
    $c \leftarrow a^2+b$
    $b \leftarrow a+b+c$
    $a \leftarrow a \times (a+1)$
    $c \leftarrow (b-a)/2$
    Renvoyer c
Fin

Remarque :

Rappelons que pour calculer une puissance en Python nous utilisons « ** ».
Tester

```
9**2
9**0.5
25**0.5
```

Que représente $a^{0.5}$ pour a un réel positif?

.....

Exercice 3 :

Soit ax^2+bx+c un polynôme. Écrire une fonction python qui prend pour argument a, b et c et renvoie de discriminant du polynôme

$\Delta =$ .....

```
def discriminant(a,b,c):
    delta =......
    return (..............)
```

Exercice 4 :

Considérons maintenant la fonction `racP(` suivante

```
import numpy as np
def racP ( a , b , c ) :
    delta= b**2-4*a*c
    xPlus=(-b+delta**0.5 ) / ( 2* a )
    xMoins=(-b-delta**0.5 ) / ( 2 * a )
    return xPlus , xMoins
```

1. Recopier soigneusement ce script et tester le avec $a=1$, $b=-5$ et $c=6$.
2. Que sont les variables de ce programme?
   .....
3. Que renvoie ce programme?
   .....
4. Que se passe-t-il si $a=1$, $b=1$ $c=1$? Nous verrons plus tard comment arranger ce programme.
   .....
5. Modifier ce programme pour qu'il utilise `discriminant`.
6. Faire un programme `racSomProd` qui calcule la somme et le produit de deux racines d'un polynôme de degré 2 à partir de ses cœfficients..
7. Tester avec $a=1$, $b=-5$ et $c=6$. Que constatez-vous?