

## TP Informatique 5 : Fonctions 1

### Definition : Les fonctions en python

Lorsque l'on doit effectuer une tâche de manière récurrente on a vu qu'on utilise un script python. Lorsque dans un script python on doit utiliser une sous-tâche de manière récurrente on peut alors utiliser une **fonction**. Les fonctions en informatique ont un fonctionnement similaire aux fonctions mathématiques. Une fonction  $f$  demande un argument  $x$  pour calculer une image  $y$  telle que  $y = f(x)$ . De la même manière une fonction en python qui s'écrit :

```
def fonction1(X):
    instruction 1
    instruction 2
    return Y
```

appelle un argument  $X$ , et va réaliser une série d'instructions et renvoyer la valeur  $Y$ . Dans la suite du programme on pourra à chaque fois que l'on souhaite réaliser les instructions concernées à partir d'une valeur  $X$  pour obtenir  $Y$  écrire la commande :

```
Y=fonction1(X)
```

Les fonctions en python peuvent nécessiter plusieurs arguments et renvoyer plusieurs valeurs :

```
def fonction2(X,Y):
    instruction 1
    instruction 2
    instruction 3
    return A,B
```

On pourra alors réaliser les instructions concernées en écrivant la commande :

```
A,B=fonction2(X,Y)
```

### Exemple :

```
def carré(X):
    Y=X**2
    return Y

def Cercle(X,Y):
    if X**2+Y**2==1:
        return True
    else :
        return False

def échange(X,Y):
    return Y,X
```

### Exercice 1 :

- Q.1** Écrire une fonction **AireD(R)** qui prend comme paramètre un réel  $R$  et qui retourne l'aire d'un disque de rayon  $R$ .
- Q.2** Écrire une fonction **VolC(R,h)** qui prend comme paramètres deux réels,  $R$  et  $h$ , et qui retourne le volume du cylindre de rayon  $R$  et de hauteur  $h$  ; on pourra utiliser la fonction précédente.

### Exercice 2 :

- Q.1** Écrire une fonction **Moyenne(L)** qui prend comme paramètre une liste  $L$  de réels, et qui retourne la moyenne des valeurs de  $L$ .

### Exercice 3 : Propagation d'incertitudes

En utilisant la bibliothèque **numpy** et les fonctions **np.random.uniform**, **np.mean** et **np.std** :

```
import numpy as np
X=np.random.uniform(X1,X2,N) # Renvoie un tableau de N valeurs distribué aléatoirement entre X1 et X2
X_moy=np.mean(X) # Renvoie la valeur moyenne du tableau de N valeurs X
```

```
uX=np.std(X,ddof=1) # Renvoie l'écart-type de la distribution des N valeurs du tableau X
```

- Q.1** Écrire une fonction **Tirage(X,DeltaX)** qui renvoie un tableau de 10 000 valeurs distribuées aléatoirement entre  $X - \text{DeltaX}$  et  $X + \text{DeltaX}$ .
- Q.2** Écrire une fonction **Somme(Xmoy,DeltaX,Ymoy,DeltaY)** qui renvoie la valeur moyenne et l'écart-type de la somme de deux tableau de 10 000 valeurs pour  $X$  et  $Y$  avec un demi interval  $\text{DeltaX}$  et  $\text{DeltaY}$ .

## Exercice 4 : Recherche de maximum

- Q.1** Écrire une fonction **max(L)** qui reçoit comme paramètre une liste d'entiers  $L$ , et qui retourne la plus grande valeur de cette liste.
- Q.2** Écrire une fonction **indice\_max(L)** qui reçoit comme paramètre une liste d'entiers  $L$ , et qui retourne la plus grande valeur de cette liste et l'indice du premier élément égal au maximum.

## Exercice 5 :

- Q.1** Écrire une fonction **Facto(n)** qui reçoit comme paramètre un entier naturel  $n$ , et qui retourne valeur de  $n!$ .
- Q.2** Écrire une fonction **Somme(N)** qui reçoit comme paramètre un entier naturel  $N$ , et qui retourne la valeur de

$$\sum_{n=0}^N \frac{1}{n!}.$$

## Exercice 6 : Repérage dans le plan

- Q.1** Écrire une fonction **Carte(r,theta)** qui reçoit comme paramètres les coordonnées polaires  $r$  et  $\theta$  d'un point  $M$ , et qui retourne les coordonnées cartésiennes  $x$  et  $y$  de ce point.
- Q.2** Écrire une fonction **polaire(x,y)** qui reçoit comme paramètres les coordonnées cartésiennes  $x$  et  $y$  d'un point  $M$ , et qui retourne les coordonnées polaires  $r$  et  $\theta$  de ce point.

## Exercice 7 : Primalité

- Q.1** Écrire une fonction nommée **TestP(n)** qui détermine si un entier  $n$  donné est premier ou non.
- Q.2** Écrire une fonction nommée **PGDP(n)** qui retourne le plus grand diviseur premier d'un entier naturel  $n$ .

## Exercice 8 : Écriture en chiffres romains

L'objectif est d'écrire une fonction qui convertit tout nombre entre 1 et 3999 en chiffres romains.

- Q.1** Écrire les nombres suivants en chiffres romains : 23, 144, 95, 3999
- Q.2** Pourquoi s'arrête-t-on à 3999?
- Q.3** Créer quatre fonctions **unite**, **dizaine**, **centaine** et **millier** qui génèrent respectivement l'écriture en chiffres romains des unités, dizaines, centaines et milliers d'un entier que l'on se donnera ultérieurement.
- Q.4** Créer une fonction **romains** qui donne l'écriture en chiffres romains d'un entier compris entre 1 et 3999 donné par l'utilisateur.