

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

Cours N°5

Fonctions Définies par l'Utilisateur

1

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

1. Introduction

- Dans tout langage de programmation évolué, il est possible de définir un bloc d'instructions qui pourra être appelé dans n'importe quelle partie du programme principal en faisant référence uniquement par son identifiant.
- Cette technique de programmation simplifie grandement les algorithmes (programmes) et fait appel à des **sous-programmes** effectuant chacun des tâches précises.
- Pour pouvoir utiliser un sous-programme il faut d'abord le définir.
- Cette définition doit être faite avant l'appel du sous-programme.

2

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

2. Notion de Fonction

Définition :
 Une **fonction** est un groupe d'instructions doté d'un nom générique, qui exécute une tâche déterminée ou un algorithme. La fonction est identifiée à un type et restitue une valeur en fin d'exécution.

Exemple :
 La fonction (prédéfinie) exponentielle en langage Python est **exp(x)** .
 La syntaxe fait référence à un nom générique (identificateur) de fonction **exp** et un argument **x**.

Remarque : Certaines fonctions peuvent avoir plusieurs arguments ou aucun argument.

3

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

3. Notion de Fonction Définie par l'Utilisateur

Définir une fonction :
 La définition d'une fonction se fait, en général, tout au début du script avant la partie du programme principal.
 La définition algorithmique d'une fonction est sous la forme:

Définir <Nom de la fonction> (argument1, argument2,..., argumentN):
 <Bloc d'instructions>
 retourner <Valeur de la fonction>

Traduction en langage Python

```
def <Nom_fonction> (argument1, argument2,..., argumentN):
    <Bloc_instructions>
    return <Valeur_fonction>
```

4

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

3. Notion de Fonction Définie par l'Utilisateur

Définir une fonction (suite) :

- Le mot-clé **def** annonce la définition d'une fonction.
- Le nom de la fonction doit respecter les règles de nomination des identificateurs. N'utilisez pas un nom de variable déjà instanciée pour nommer une fonction.
- Les parenthèses sont obligatoires, même si la fonction n'attend aucun argument.

ex: `def ma_fonction():`

- Attention à l'indentation: le bloc d'instructions et la ligne (ou les lignes) comportant **return** doivent être délimités par l'indentation.

5

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

3. Notion de Fonction Définie par l'Utilisateur

Exemple 1 : Programme utilisant une fonction qui renvoie le maximum de deux entiers.

Définir maximum(a1,b1) :

si $a1 \geq b1$ alors $max \leftarrow a1$
Sinon $max \leftarrow b1$ FSI
retourner max

Début { Programme Principal}

- 1) Lire(A,B)
- 2) $x \leftarrow 2 \times \text{maximum}(A,B)$
- 3) Écrire(x)

Fin {Programme Principal}

6

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM



3. Notion de Fonction Définie par l'Utilisateur

Programme Python

```
def maximum(a1,b1) :
    if a1 >= b1 :
        max = a1
    else :
        max = b1
    return max

# Début du Programme Principal
A=int(input('Entrer le premier nombre entier A:'))
B=int(input('Entrer le deuxième nombre entier B:'))

x=2*maximum(A,B)
print('La valeur de x:',x)
```

7



3. Notion de Fonction Définie par l'Utilisateur

- Dans cet exemple, les arguments "a1" et "b1" sont utilisés pour calculer le maximum. On les appelle **paramètres formels** de la fonction. Les **paramètres formels** sont utilisés uniquement dans le traitement à l'intérieur du sous-programme.
- Seul un appel de la fonction avec des **paramètres (arguments) effectifs** ("A" et "B" dans notre exemple) en place des paramètres formels déclenche l'exécution de la fonction "**maximum**".
- L'instruction **return** suivie de la variable "max" provoque l'arrêt de la fonction "**maximum**" et le retour de l'expression à l'endroit même où la fonction a été appelée.
- On dit de la variable "max" qu'elle est **locale** car elle n'est pas visible à l'extérieur de la fonction. **Une variable locale** est définie localement dans la fonction et sa portée est limitée à cette dernière. Elle n'est pas reconnue dans le programme principal ni dans les autres sous-programmes (s'ils existent).

8

3. Notion de Fonction Définie par l'Utilisateur

- Une fonction peut aussi faire appel à des **variables globales**, bien que pour des raisons de sécurité du code, cela devrait être au maximum évité.
- Ces variables sont visibles en dehors de la fonction elle-même (dans le programme principal par exemple) et doivent être explicitement déclarées comme telle dans l'en-tête de la fonction :

```
def <Nom_fonction> (argument1, argument2,..., argumentN):
    global var1,var2,...,varN
    <Bloc_instructions>
    return <Valeur_fonction>
```

9

3. Notion de Fonction Définie par l'Utilisateur

Exemple 2 : Programme utilisant une fonction qui calcule le factoriel d'un entier positif.

```
Définir fact(n) :
    variable globale L
    p ← 1
    pour i Dans (0,n+1) faire
        si i=0 alors p ← 1
        sinon p ← p×i FSI
    L ← p+1
    retourner p
```

Début { Programme Principal}

- 1) Lire(M)
- 2) S ← fact(M)² + L
- 3) Écrire(S)

Fin {Programme Principal}

10

Cours 5: Fonctions Définies par l'Utilisateur

3. Notion de Fonction Définie par l'Utilisateur

Programme Python

```
def fact(n):
    global L
    p=1
    for i in range(0,n+1):
        if i==0 :
            p=1
        else:
            p=p*i

    L=p+1
    return p

# Début du Programme Principal
M=int(input("M= "))
S=fact(M)**2+L
print("S=",S)
```

11

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

4. Notion de Procédure

Définition :

Une procédure est un groupement d'instructions, identifié par un nom, qui exécutent une tâche ou un algorithme déterminé. Une procédure modifie quelque chose dans l'environnement (affichage à l'écran, écriture dans un fichier ...) et ne retourne aucune valeur. Elle est une fonction qui ne renvoie aucune valeur.

Exemple :

exp(x) : une fonction qui renvoie une valeur.

print(end="\n") : c'est une tâche à exécuter mais ne renvoie aucune valeur.

12

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

4. Notion de Procédure

Définir une procédure :

La définition d'une procédure se fait, en général, tout au début du script avant la partie du programme principal.

La définition algorithmique d'une procédure est sous la forme:

Définir <Nom de la procédure> (argument1, argument2,..., argumentN):
<Bloc d'instructions>

Traduction en langage Python

```
def <Nom_procédure> (argument1, argument2,..., argumentN):
    <Bloc_instructions>
```

13

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

4. Notion de Procédure

Exemple : Programme utilisant une procédure qui permet d'avoir des sauts de lignes.

Définir saut(a) :

pour i Dans (0,a+1) faire
 Écrire(end="\n")

Début { Programme Principal}

- 1) Lire(N)
- 2) saut(N)
- 3) Écrire("fin du programme")

Fin {Programme Principal}

14

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

4. Notion de Procédure

Programme Python

```
def saut(a):  
    for i in range(1,a+1):  
        print(end="\n")  
  
# Début du Programme Principal  
N=int(input('N= '))  
saut(N)  
print("fin du programme")
```

15

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

5. Fonctions récursives

Définition :
Une fonction (procédure) est dite récursive si elle s'appelle d'elle-même.

Exemple: Programme permettant de faire le calcul du factoriel sous forme récursive.

16

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

5. Fonctions récursives

Algorithme Programme permettant de faire le calcul du factoriel sous forme récursive.

```

Définir factoriel(a) :
    si a <= 1 alors f ← 1
    Sinon f ← a × factoriel(a-1) FSI
    retourner f

Début { Programme Principal}
1) Lire(n)
2) resultat ← factoriel(n)
3) Écrire(resultat)
Fin {Programme Principal}
  
```

17

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM

Cours 5: Fonctions Définies par l'Utilisateur

Université Abou Bekr Belkaid - Tlemcen (Algérie)

5. Fonctions récursives

Programme permettant de faire le calcul du factoriel sous forme récursive.

```

def factoriel(a):
    if a<=1 :
        f=1
    else:
        f=a*factoriel(a-1)
    return f

# Début du Programme Principal
n=int(input("n= "))
resultat=factoriel(n)
print("Le factoriel de ",n," est: ",resultat)
  
```

Attention: Il faut toujours vérifier qu'en aucun cas on ne puisse avoir une boucle infinie qui bloquerait la machine.

18

Informatique2 2014-2015©Faculté des Sciences Semestre 2 LMD SM