Master 1 Humanités numériques – Algorithmique et programmation

TD 4 : Sous-programmes

J. Darmont – https://eric.univ-lyon2.fr/jdarmont/

Les exercices 1 et 2 sont à formuler en langage algorithmique, puis en Python. Tester l'exécution dans ce dernier cas.

Exercice 1 : Procédures et paramètres d'entrée

1. Transformer les algorithmes des exercices 2 et 5 du TD 3 (triangle et table de division) en procédures dans lesquelles la constante TAILLE devient un paramètre (formel) d'entrée *taille*.
2. Dans un programme principal, appeler chacune des procédures avec différentes valeurs du paramètre *taille*. Un paramètre réel peut être une variable ou une valeur.

Exercice 2 : Fonctions

1. Ajouter à la procédure du triangle le calcul du nombre de caractères *. Pour cela, initialiser une variable entière à 0, puis dans la boucle « Pour *i* de 1 à *taille* faire », lui ajouter *i* à chaque itération (*i* représente le nombre d'étoiles de chaque ligne).
2. Transformer la procédure en fonction retournant le nombre d'étoiles.
3. Ce nombre d'étoiles doit être récupéré dans le programme principal et affiché à l'écran.

Note : Python ne permet pas de définir des paramètres de sortie à une procédure. C'est pourquoi il a fallu transformer la procédure en fonction. Python permet plusieurs valeurs de retour dans les fonctions, qui font ainsi parfaitement office de paramètres de sortie.

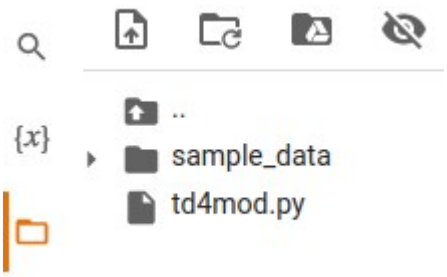
Exercice 3 : Modules Python standards

Python dispose d'une bibliothèque de modules standards[1] et bien d'autres modules ont été développés par des tiers. À titre d'exemple, dans un nouveau programme :

1. Importer toutes les fonctions du module math[2].
2. Afficher le nombre π (pi). Tester.
3. Afficher l'arrondi à deux décimales de π. Tester.
4. Importer toutes les fonctions du module random[3] (nombres aléatoires).
5. Afficher un nombre réel aléatoire quelconque. Tester.
6. Afficher une série de 10 nombres entiers entre 1 et 6 (simulation de lancers de dés). Tester.

Exercice 4 : Modules Python personnalisés

1. Copier/coller les deux fonctions des exercices 1 et 2 dans un fichier nommé td4mod.py (par exemple) de votre ordinateur (utiliser un *éditeur* de texte simple comme le *notepad* de Windows, pas un *traitement* de texte). Sans programme principal, ce fichier est un **module**.
2. Dans le *notebook*, cliquer sur l'icône  en haut à gauche. C'est l'espace des fichiers du *notebook*.
3. Cliquer ensuite sur l'icône  pour télécharger le fichier td4mod.py dans le *notebook*. Résultat attendu ci-dessous.



4. Dans une nouvelle cellule de code, importer toutes les fonctions du module td4mod et appeler les fonctions comme dans les exercices 1 et 2.
5. Tester !

1 https://docs.python.org/3/library/
2 https://docs.python.org/fr/3/library/math.html
3 https://docs.python.org/fr/3/library/random.html

Correction Exercice 1

```
Procédure triangleEtoiles(taille : Entier)
    Var i : Entier
    Var ligne : Chaîne
Début
    ligne ← ""
    Pour i de 1 à taille faire
        ligne ← ligne | "*"
        Écrire("*")
    Fin pour
Fin

Procédure tableDiv(taille : Entier)
    Var i, j : Entier
    Var quotient : Réel
Début
    Pour i de 1 à taille faire
        Pour j de 1 à taille faire
            quotient ← i ÷ j
            Écrire(i, " ÷ ", j, " = ", quotient)
        Fin pour
    Fin pour
Fin

Algorithme principal
    Var t : Entier
Début
    triangleEtoiles(3)
    triangleEtoiles(5)
    tableDiv(2)
    t ← 4
    tableDiv(t)
Fin
```

```
def triangleEtoiles(taille):
    ligne = ""
    for i in range(1, taille + 1):
        ligne += "*"
        print(ligne)

def tableDiv(taille):
    for i in range(1, taille + 1):
        for j in range(1, taille + 1):
            quotient = i / j
            print(i, "/", j, "=", quotient)
```

```
# Programme principal
triangleEtoiles(3)
triangleEtoiles(5)
tableDiv(2)
t = 4
tableDiv(t)
```

Correction Exercice 2

```
Fonction triangleEtoiles(taille : Entier) : Entier
    Var i, nbEtoiles : Entier
    Var ligne : Chaîne
Début
    ligne ← ""
    nbEtoiles ← 0
    Pour i de 1 à taille faire
        ligne ← ligne | "*"
        nbEtoiles ← nbEtoiles + i
        Écrire(ligne)
    Fin pour
    Retourner nbEtoiles
Fin

Algorithme principal
    Var n : Entier
Début
    n ← triangleEtoiles(3)
    Écrire(n)
    Écrire(triangleEtoiles(5))
Fin

def triangleEtoiles(taille):
    ligne = ""
    nbEtoiles = 0
    for i in range(1, taille + 1):
        ligne += "*"
        nbEtoiles += i
        print(ligne)
    return nbEtoiles

# Programme principal
n = triangleEtoiles(3)
print(n)
print(triangleEtoiles(5))
```

Correction Exercice 3

```
# Module math
from math import *
print(pi)
print(round(pi, 2))

# Module random
from random import *
print(random())
for i in range(11):
	print(randrange(6) + 1)
```

Correction Exercice 4

```
# Fichier td4mod.py (module td4mod)

def tableDiv(taille):
	for i in range(1, taille + 1):
		for j in range(1, taille + 1):
			quotient = i / j
			print(i, "/", j, "=", quotient)

def triangleEtoiles(taille):
	ligne = ""
	nbEtoiles = 0
	for i in range(1, taille + 1):
		ligne += "*"
		nbEtoiles += i
		print(ligne)
	return nbEtoiles


# Programme principal dans une cellule du notebook

# Importation du module
from td4mod import *

# Appels de la fonction triangeEtoiles
n = triangleEtoiles(3)
print(n)
print(triangleEtoiles(5))

# Appels de la fonction tableDiv
tableDiv(2)
t = 4
tableDiv(t)
```