

Master 1 Humanités numériques – Algorithmique et programmation

TD 4 : Sous-programmes

J. Darmont – <http://eric.univ-lyon2.fr/~jdarmont/>

Les exercices 1 et 2 sont à formuler en langage algorithmique textuel, en Blockly, puis en Python. Tester l'exécution dans ces deux derniers 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 *.
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 : Blockly et Python ne permettent pas de définir des paramètres de sortie à une procédure. C'est pourquoi il a fallu transformer la procédure en fonction. Blockly ne permet qu'une seule valeur de retour dans les fonctions, ce qui est tout-à-fait classique, mais limitant. En revanche, Python permet plusieurs valeurs de retour dans les fonctions, qui font ainsi parfaitement office de paramètres de sortie.

Exercice 3 : Modules Python

Python dispose d'une bibliothèque de modules standards¹ 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 turtle², qui permet de dessiner des formes géométriques.
2. Choisir une couleur (red, green, yellow...) à l'aide de la fonction pencolor(c), où c est une chaîne de caractères.
3. Répéter 4 fois les instructions suivantes :
 - a. Avancer de 300 pixels à l'aide de la fonction forward(d), où d est une distance entière ;
 - b. Tourner à droite de 90 degrés à l'aide de la fonction right(a), où a est un angle.

¹ <https://docs.python.org/3/library/>

² <https://docs.python.org/3/library/turtle.html>

4. Tester ! Le résultat apparaît dans une nouvelle fenêtre.

Exercice 4 : Modules personnalisés

1. Copier/coller vos deux fonctions dans un nouveau fichier Python, par exemple td4mod.py. Sans programme principal, c'est un module.
2. Dans un autre fichier Python, importer toutes les fonctions du module td4mod et appeler les fonctions comme dans les exercices 1 et 2.
3. Tester !

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 ← $i \div j$
 Écrire(i, " ÷ ", j, " = ", quotient)
 Fin pour
Fin pour

Fin

Algorithme principal

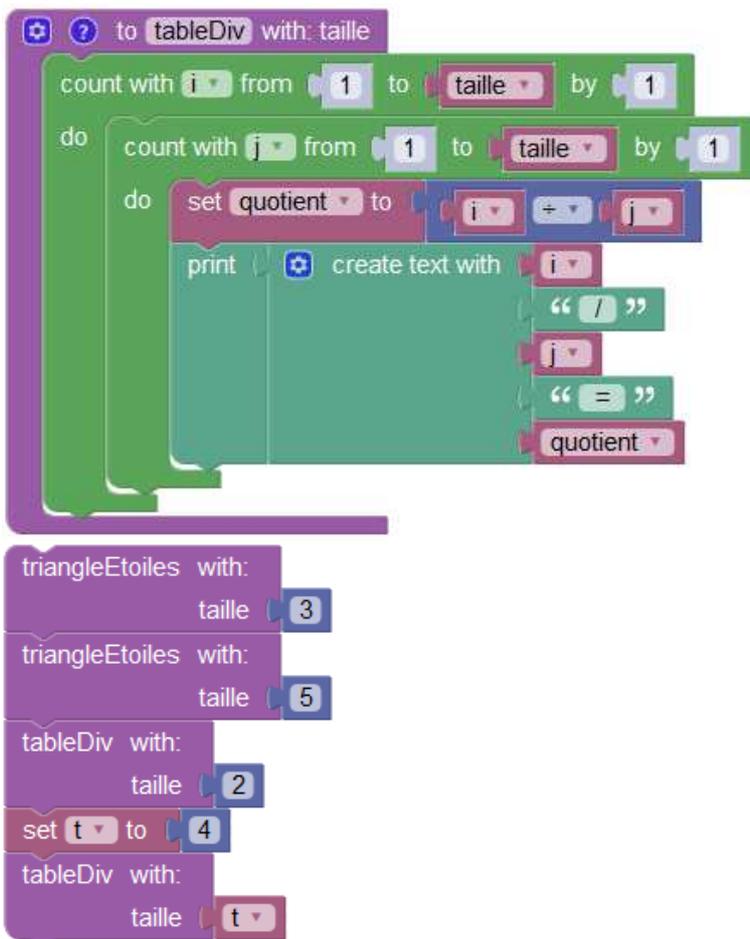
Var t : Entier

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("*")

Fin pour

Retourner nbEtoiles

Fin

Algorithme principal

Var n : Entier

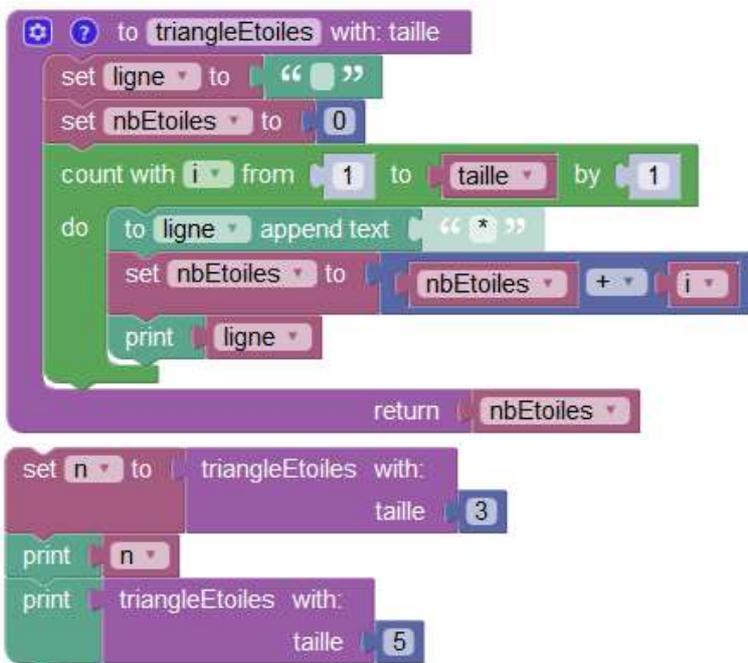
Début

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

```
from turtle import *
pencolor("green");
for i in range(1, 5):
    forward(300)
    right(90)
```

Correction Exercice 4

Fichier td4mod.py (module)

```
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 fichier td44.py
```

```
# Importation du module
```

```
from td4mod import *
```

```
# Appel des fonctions
```

```
n = triangleEtoiles(3)
```

```
print(n)
```

```
print(triangleEtoiles(5))
```

```
tableDiv(2)
```

```
t = 4
```

```
tableDiv(t)
```