

# TP1 Informatique

## Premier pas avec Python

### J1 MI 1003, groupe B3, Université Bordeaux

Afin d'effectuer les différentes tâches demandées, on pourra s'aider de l'aide-mémoire distribué avec ce polycopié. Vous pouvez également retrouver cet aide-mémoire sur le site :

<http://dept-info.labri.fr/ENSEIGNEMENT/INITINFO/initinfo/support.html>.

## Premières manipulations

**Exercice 1** Lancez un terminal, puis lancez *idle3* (ne pas oublier le &).

**Exercice 2** Testez les instructions suivantes provenant de l'exercice 1.1.9 du fascicule (et même un peu plus) tout en prenant le temps d'expliquer précisément tous les résultats obtenus.

```
11 * 34
13.4 - 6 # ceci est un commentaire
13 / 4   # division avec virgule
13 // 4  # division entiere
13 % 2
14 % 3
i = 5
i = i + 4
i
j = 0
k
i < j
i != 9
i == 9
i, j, i*i
"Coucou!"
print("Coucou!")
true
True
True and False
True or False
```

Il faut que tout soit clair. Si vous ne comprenez pas quoi que soit, n'hésitez pas à m'appeler.

## Premier programme

**Exercice 3** Lancez une *fenêtre d'édition* en cliquant sur **File, New Window**. Tapez la définition suivante issue du fascicule :

```
def f(x):
    return x*x + x + 1
```

Sauvegardez le programme sous le nom `premierprogramme.py` (par exemple), et testez-le en utilisant la touche *F5* (ou en cliquant sur **Run, Run Module**).

Tapez alors les expressions suivantes qui appellent cette fonction en lui passant différents arguments et prenez le temps d'expliquer en détail les résultats obtenus.

```

f(2)
f(x)
t=4
f(t)
x = (3*t + 1) / 2
x, f(x)
x = (3*t + 1) // 2
x, f(x)
f(x - t)
t = f(t)

```

**Exercice 4** Écrire une fonction `moyenne(a,b)` qui retourne la moyenne de deux nombres  $a$  et  $b$ . Testez-la.

**Exercice 5** Écrire une fonction `unité(n,c)` qui teste si le chiffre des unités de l'entier  $n$  est  $c$ ; la valeur calculée doit être *booléenne*, c'est-à-dire égale à `True` ou `False`. Testez-la.

## Les commandes `print` et `return`

**Exercice 6** Différence entre `print` et `return` : Écrire, enregistrer et exécuter le programme suivant :

```

def avecprint(u) :
    print(u+1)
def avecreturn(u) :
    return(u+1)

```

Puis tester dans l'interprète les deux fonctions précédemment écrites en tapant par exemple (Vous pouvez tenter d'autres choses si ça vous chante!) :

```

x = 1
avecprint(x)
avecreturn(x)
resultat1 = avecprint(x)
resultat2 = avecreturn(x)
resultat1
resultat2

```

Expliquez ce que fait la commande `print` et quelle est la différence avec `return`.

**Exercice 7** Prédire ce que vont faire les programmes suivants :

(a)	<pre> def fct(u) :     return u+3 u=3 fct(u) print(u) </pre>	(b)	<pre> def fct(u) :     return u+3 u=3 u=fct(u) print(u) </pre>	(c)	<pre> def fct2(u) :     u=u+3 u=3 fct2(u) print(u) </pre>
(d)	<pre> def fct3(u) :     return u+3     return u-2 u=3 u=fct3(u) print(u) </pre>	(e)	<pre> def fct4(u) :     print (u)     return u+3 u=3 u=fct4(u) print(u) </pre>	(f)	<pre> def fct5(u) :     return u+3     print (u) u=3 u=fct5(u) print(u) </pre>

Les recopier sur l'ordinateur, les tester et comparer avec vos prédictions. Il faut que tout soit clair!

## Modules

Lorsque l'on écrit des programmes, on souhaite pouvoir réutiliser du code déjà écrit. En python, cela est possible grâce aux modules. Un module est une bibliothèque contenant du code. Python contient de nombreux modules. Par exemple, il contient le module *math* qui permet de calculer des racines carrés, des cosinus et d'autres fonctions mathématiques.

Pour réutiliser le code du module *math*, il vous suffit d'importer le module en tapant la commande :

```
import math
```

Ensuite, pour calculer la racine carré, il vous suffit de taper :

```
math.sqrt(3)
```

Si vous souhaitez obtenir de l'aide concernant ce module, vous devez taper :

```
help(math)
```

Lorsque vous écrivez un programme dans un fichier, vous écrivez un module. Le nom du module est le nom du fichier sans son extension. Vous pouvez donc créer vous même vos propres modules et les importer de la même manière que vous avez importé la bibliothèque *math*. Cependant, le module à importer doit être situé dans le même répertoire que le programme qui importe le module.

**Exercice 8** Créez un nouveau programme (ou donc module) *somme* en ajoutant un fichier *somme.py* contenant le code suivant :

```
def g(n):  
    return n*(n+1)//2
```

Puis créez, exécutez et testez le programme *principal.py* contenant le code

```
import somme  
  
def imprimer_somme(n):  
    print("La somme des entiers allant de 0 à " + str(n) + " vaut:" )  
    print( somme.g(n) )
```

**Exercice 9** Écrire une fonction *bizarre(x)* qui calcule la moyenne entre  $x$  et  $f(x)$ . On utilisera les modules précédemment définis.

Certaines fonctions de certains modules sont très fréquemment utilisées. Dans ce cas, il est plus sympathique de pouvoir utiliser directement le nom de la fonction sans avoir à rappeler le nom du module. C'est possible sous python, pour cela, il suffit d'importer la fonction en écrivant la ligne :

```
from MODULE import FUNCTION
```

Par exemple, si l'on reprend l'exemple de la racine carré, on obtient :

```
from math import sqrt
sqrt(3)
```

Si vous souhaitez importer toutes les fonctions d'un module donné, il vous suffit d'écrire :

```
from MODULE import *
```

**Exercice 10** Reprendre l'exercice 9 en important dans le programme les fonctions à la place des modules.

## Pour ceux qui s'ennuient

**Exercice 11** Testez les instructions suivantes et tentez d'expliquer ce que vous observez.

```
x=2
(x < 3) and inconnu
(x < 3) or inconnu
x=4
(x < 3) and inconnu
(x < 3) or inconnu
```

**Exercice 12** Écrire une fonction `centaine(n,c)` qui teste si le chiffre des centaines de  $n$  est  $c$ .

**Exercice 13** Écrire une fonction `chiffredec(n,k)` qui renvoie le  $k$ -ième chiffre de  $n$  (en base décimale).

**Exercice 14 Exercice-défi !**

Écrire une fonction `maximum(a,b)` qui renvoie le maximum des deux entiers  $a$  et  $b$  sans utiliser des conditionnelles (`if` ou `else`) ou des boucles (`while` ou `for`).

(Attention, la solution de cet exercice n'est surtout pas à restituer lors des DS communs...)