

**DEV1 – DEV1L – Laboratoires Python****TD 03 – Programmer en Python**

Dans ce TD, nous allons en dire plus sur la notion de fonction en Python. Nous appliquerons ensuite nos connaissances de façon à écrire des petits scripts résolvant divers problèmes.

**Table des matières**

<b>1</b>	<b>Rappel : Python en mode script</b>	<b>2</b>
<b>2</b>	<b>Fonctions et modules Python</b>	<b>2</b>
<b>3</b>	<b>Exercices récapitulatifs</b>	<b>3</b>
<b>4</b>	<b>En résumé ...</b>	<b>4</b>

## Rappel

Vous devez absolument avoir fait et compris les TD01 et TD02 *avant* d'aborder celui-ci.

## 1 Rappel : Python en mode script

Lors du TD02 nous avons écrit nos premiers scripts en Python, voici l'exemple donné à cette occasion :

```
1 # Mon premier script Python !
2
3 # Un programme calculant l'aire du carré dont la
4 # longueur du côté est entré au clavier par l'utilisateur
5
6 print(" **** Bienvenue ! **** ")
7 x = input("Entrez la valeur du côté du carré : ")
8 x = int(x)
9 print("L'aire du carré vaut:", x**2)
```

aireCarre.py

Nous avons déjà rencontré plusieurs *fonctions* standard mise à votre disposition par Python : `print`, `input`, `int` (dans le script ci-dessus), `max`, `min`, `len` (voir TD01 et TD02), etc.

La liste complète des fonctions standards mises à votre disposition par Python est disponible dans la documentation officielle :

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

## 2 Fonctions et modules Python

En plus des fonctions standard évoquées ci-dessus, la *bibliothèque standard* de Python met à notre disposition des *modules* (sortes de classeurs contenant des fonctionnalités pouvant être utilisées par le programmeur<sup>1</sup>).

Il est nécessaire d'importer explicitement un tel module si on souhaite l'utiliser au moyen de la commande :

```
import nom_du_Module
```

### Exercice 1 Les modules en Python

Lancez Python en mode interactif et lancez le système d'aide au moyen de la fonction `help()`. Tapez ensuite `modules` pour avoir la liste des modules qui sont à votre disposition. Dans la liste qui s'affiche, essayez de retrouver le nom d'un module permettant probablement de :

1. Manipuler des dates et des heures ?

Le nom du module est :

2. Manipuler des données audio (son, musique, ...) ?

Le nom du module est :

### Exercice 2 Le module math de Python

1. Nous verrons plus tard que ces modules peuvent contenir d'autres choses que des fonctions, notamment des nouveaux types de données.

Lancez Python en mode interactif et importez le module `math` au moyen de la commande `import math`. Utilisez ensuite la fonction `help()` pour afficher l'aide de Python sur ce module. Utilisez l'aide qui s'affiche alors pour déterminer quelle fonction nous pourrions utiliser pour calculer :

On souhaite calculer	Quelle fonction du module utiliser ?
La racine carrée d'un nombre	
Le logarithme en base 2 d'un nombre	
La factorielle d'un nombre	

Il suffit de faire précéder le nom de la fonction par le nom du module lorsqu'on souhaite utiliser une fonction d'un module donné :

```

1  # Un programme qui calcule la racine carrée d'un nombre entré au clavier
2  # Attention: le programme plante si le nombre entré est négatif
3
4  import math
5
6  print(" **** Bienvenue ! **** ")
7  x = int(input(" Entrez un nombre positif : "))
8
9  print("La racine carrée de",x,"vaut", math.sqrt(x))

```

### 3 Exercices récapitulatifs

Les exercices qui suivent vous invitent à mettre en application tout ce que vous avez vu jusqu'à présent pour écrire vos propres scripts Python. N'hésitez pas à faire appel à votre professeur de labo si vous êtes bloqué sur l'un d'eux !

#### Exercice 3 Minimum et maximum

Écrivez un script qui demande trois nombres à l'utilisateur et qui affiche le minimum et le maximum de ces trois nombres à l'écran (Aide : utilisez les fonctions standard `min()` et `max()` de Python).

#### Exercice 4 Le plus grand commun diviseur (PGCD)

Écrivez un script qui demande deux nombres entiers à l'utilisateur et qui affiche le plus grand diviseur commun à ces deux nombres.

Par exemple : si l'utilisateur entre 6 et 9 le programme affiche 3, car 3 divise à la fois 6 et 9 et c'est le plus grand entier ayant cette propriété (Aide : le module `math` contient une fonction calculant le PGCD des nombres qu'elle reçoit en paramètre. Consultez l'aide de Python pour savoir laquelle.)

#### Exercice 5 Tirer un nombre au hasard

Écrivez un script qui demande deux nombres entiers  $n$  et  $m$  à l'utilisateur et qui affiche à l'écran un nombre au hasard tiré entre  $n$  et  $m$ .

(Aide : La fonction `randint()` du module `random` permet de générer un nombre au hasard situé entre les 2 bornes qui lui sont passées en paramètre. Consultez éventuellement l'aide de Python pour plus de détails.)

#### Exercice 6 Secondes en minutes

Écrivez un script qui demande un nombre de secondes à l'utilisateur et qui affiche le nombre de minutes que cela représente ainsi que le nombre de secondes résiduelles.

Par exemple, si l'utilisateur entre 217 secondes, le programme affiche **3 minutes et 37 secondes**. (Aide : relisez le TD01 pour vous remettre en tête les opérateurs mathématiques).

#### Exercice 7 Trier des nombres

Écrivez un script qui demande quatre nombres à l'utilisateur et qui les affiche à l'écran du plus petit au plus grand. Par exemple : si l'utilisateur entre 2, 3.1, -1 et 1.52, le programme affiche : -1, 1.52, 2, 3.1 à l'écran. (Aide : la fonction `sorted()` permet de trier une liste du plus petit au plus grand)

#### Exercice 8 Les intérêts sur une année

Écrivez un script qui demande un montant (un nombre décimal) à l'utilisateur et affiche les intérêts reçus après 1 an pour un montant placé en banque à 2% d'intérêts.

#### Exercice 9 Les intérêts sur plusieurs années

Écrivez un script qui demande un montant (un nombre décimal) à l'utilisateur ainsi qu'un nombre d'année et affiche les intérêts reçus après ce nombre d'années pour un montant placé en banque à 2% d'intérêts<sup>2</sup>.

## 4 En résumé ...

### Principaux points de matière du TD

Voici les principaux points abordés lors de ce TD. Vous devez absolument être à l'aise avec ceux-ci avant d'aborder la prochaine séance d'exercice.

1. Obtenir de l'aide sur les fonctions de la bibliothèque standard de Python.
2. Importer un module dans un script Python.
3. Utiliser l'aide de Python pour obtenir des informations sur les fonctionnalités d'un module de la bibliothèque standard.
4. Faire appel aux fonctions d'un module (après l'avoir importé) dans un script.

---

<sup>2</sup>. Il va probablement que vous cherchiez un peu sur le web si vous ne savez pas comment faire. Cela fait partie du travail d'informaticien de chercher :)