

Plusieurs exercices ont été écrits par mon collègue Pierre Duclosson.

Introduction

Objectifs :

- ☞ Prendre en main l'environnement de travail (partages réseaux, programmation en Python).
- ☞ Apprivoiser la syntaxe de Python.
- ☞ Représenter la trace d'exécution d'un programme avec un tableau d'état.
- ☞ Contrôler le flot d'exécution avec des boucles (`for` ou `while`), des instructions de branchement (`if`, `elif`, `else`).
- ☞ Écrire des fonctions en respectant une spécification.

Mode d'emploi :

- ☞ On garde le cours du chapitre 1 à portée de main.
- ☞ On recopie ses codes dans les emplacements prévus sur l'énoncé, il faut apprendre à bien indenter le code manuscrit, une partie de l'épreuve terminale se passe sur papier!

Méthode

Dans un environnement de développement Python, créer un fichier `tp2.py` et l'enregistrer dans un dossier pertinent de son espace personnel sur le réseau.

On séparera les codes de chaque exercice dans des **cellules** qui peuvent exécuter de façon indépendante.

Dans **Spyder**, on introduit une cellule avec la séquence de caractères `#%%`.

1 Boucles bornées

Exercice 1 *Maîtriser le* range

1. Faire afficher les entiers de 10 à 0 de manière décroissante.

.....
.....

2. Faire afficher les entiers pairs compris entre 0 et 50 dans l'ordre croissant.

.....
.....

3. Faire afficher les entiers pairs compris entre 0 et 50 dans l'ordre décroissants.

.....
.....

 **Exercice 2 Boucles imbriquées**

1. Quel affichage obtient-on en exécutant le code ci-dessous ?

.....
.....
.....
.....

```
for i in range(5):  
    for j in range(i, 5):  
        print(j, end="")  
    print()
```

2. Pour les questions suivantes, il faut utiliser obligatoirement deux boucles imbriquées et au maximum deux fois la fonction `print`.

a. Écrire un script Python qui produit l'affichage 1 ci-dessous.

.....
.....
.....
.....

b. Écrire un script Python qui produit l'affichage 2 ci-dessous.

.....
.....
.....
.....

Affichage 1

0
01
012
0123
01234

Affichage 2

01234
12345
23456
34567
45678

Exercice 3 *Tableau d'état*

Compléter le tableau d'états permettant de tracer l'exécution du programme ci-dessous. Les valeurs des variables sont mesurées après l'exécution de la ligne 5.

```
1 f = 0
2 g = 1
3 for i in range(6):
4     print(f)
5     f = f + g
6     g = f - g
```

Valeur de i
Valeur de f
Valeur de g

2 Conditionnelles

Exercice 4 *Conditionnelle avec deux clauses*

Pour transporter des poteaux sur une île, on dispose d'une barge dont la masse utile est de 1800 kg. Écrire un programme qui demande à l'utilisateur la masse d'un poteau puis le nombre de poteaux et qui affiche le message « Risque de surcharge! » si la masse utile est dépassée.

.....

.....

.....

.....

.....

.....

.....

Exercice 5 *Conditionnelle avec plus de deux clauses*

On rappelle que les années bissextiles sont celles qui sont multiples de 4 sauf si elles sont multiples de 100 mais pas de 400. Compléter le code suivant pour qu'il affiche « Année bissextile » ou « Année non bissextile » selon la valeur du nombre rentré par l'utilisateur.

```
n = int(input("Entrez le nombre de l'année : "))
if .....:
    print('Année non bissextile')
elif .....:
    print('Année bissextile')
elif .....:
    print('Année non bissextile')
else:
    print('Année bissextile')
```

3 Boucles non bornées

Exercice 6

Écrire un programme qui permet de saisir un mot de passe sous la forme d'une chaîne de caractères et qui ensuite redemande ce mot de passe jusqu'à ce qu'il soit correct.

.....

.....

.....

.....

Exercice 7

On appelle logarithme entier d'un nombre réel $x \geq 1$, le nombre de fois qu'il faut le diviser par 2 pour obtenir un nombre inférieur à 1.

Écrire un programme qui permet de saisir un nombre x et qui affiche son logarithme entier.

.....

.....

.....

.....

4 Fonctions

Exercice 8 *Jour de la semaine*

Écrire une fonction `jour_semaine` dont on donne la spécification ci-dessous :

```
def jour_semaine(m, d, y):  
    """  
    Paramètres :  
        m un entier représentant le mois dans l'année 1<=m<=12  
        d un entier représentant le jour dans le mois 1<=d<=31  
        y un entier représentant l'année  
    Valeur renvoyée :  
        un entier d0 représentant le rang du jour de la semaine entre 1  
        et 7  
    """
```

On appliquera les formules ci-dessous, valables pour le calendrier grégorien :

```
y0 = y - (14 - m)/12  
x = y0 + y0//4 - y0//100 + y0//400  
m0 = m + 12*((14-m)//12) - 2  
d0 = (d + x + (31 * m0)//12)%7
```

.....

.....

.....

.....

.....

.....

.....

Exercice 9 *Année bissextile*

Reprendre l'exercice 5 en écrivant une fonction `bissextile(a)` déterminant si une année est bissextile. On donne ci-dessous la signature de cette fonction c'est-à-dire l'en-tête avec les types attendus pour les paramètres et la valeur de retour.

```
def bissextile(a):  
    """Signature bissextile(a:int)->bool"""
```

.....

.....

.....

.....

.....
.....

 **Exercice 10** *Pour aller plus loin*

| Travaux pratiques pages 40 et 41 du **manuel Hachette**.