


## TP3 : les boucles

**Objectif :** maîtriser l'utilisation des boucles avec Python.

 **N.B** les exercices de ce TP sont déjà abordés durant les séances TD, donc cela suppose que vous avez déjà la solution algorithmique !


### Exercice 1 : Choix de la structure itérative

Pour chacun des problèmes suivants, proposer un programme Python en choisissant la structure itérative la mieux appropriée (for ou while) :

- Calcul et affichage de la somme des N entiers naturels (de 0 à N inclus) avec N saisi par l'utilisateur.
- Boucle d'attente qui se termine si l'utilisateur entre la valeur 100
- Lecture de 10 valeurs réelles comprises entre 0 et 20 (si ce n'est pas le cas, il faut redemander la valeur à l'utilisateur) ; puis afficher :
  - Le nombre de valeurs inférieur strictement à 10
  - Le nombre de valeurs supérieur ou égale à 10 et inférieur strictement à 15
  - Le nombre de valeurs supérieur ou égale à 15
- Calcul et affichage du plus grand nombre N tel que  $\sum_{i=0}^N i \leq X$  avec X un nombre supérieur à 1 saisi par l'utilisateur.  
Exemple : si l'utilisateur donne la valeur 10 pour X, alors le nombre N recherché est 4 (0+1+2+3+4 ≤ 10 par contre 0+1+2+3+4 +5 =15 > 10)

### Exercice 2 : Compte à rebours

Ecrire un programme Python qui demande à l'utilisateur de saisir un nombre entier n positif, puis afficher tous les nombres par ordre décroissant entre n et 0. Proposer deux versions, une avec la boucle 'for' et l'autre avec la boucle 'while'.

 Vous pouvez importer le module `time` et utiliser la méthode `sleep(0.2)` de ce module pour avoir un effet compte à rebours !

### Exercice 3 : Jeu du nombre mystère

Ecrire un programme Python qui tire une valeur aléatoire x entre 0 et 100 et qui demande à l'utilisateur de deviner cette valeur. L'algorithme répond à chaque fois si la valeur est plus grande, plus petite ou égale à la valeur à deviner. Un compteur incrémenté à chaque passage donnera le nombre de tours qui ont été nécessaires pour trouver le nombre mystère.

Vous pouvez utiliser le module **random** vu en TP1 !

## Exercice 4 : Factorielle itérative

Ecrire un programme Python qui détermine la factorielle d'un entier n (donné par l'utilisateur) avec les deux types de boucles. Votre programme doit permettre à l'utilisateur de choisir la boucle à utiliser puis réalise le traitement nécessaire en conséquence.

Votre programme doit permettre aussi l'affichage de l'évolution de la valeur de la factorielle à chaque itération !

### *Point d'étape 1 à faire valider*

## Exercice 5 : Location de vélos

Le but de cet exercice est de permettre à un service de location de vélos (tournant 24 heures sur 24) de facturer ses clients. Le programme demandera à l'utilisateur d'entrer les heures de début et de fin de location sous la forme d'entiers (on ne se préoccupe pas des minutes pour simplifier).

Les tarifs de location sont définis comme suit :

- 1 euro par heure si le vélo est loué entre 0h et 7h ou entre 17h et 24h ;
- 2 euros par heure si le vélo est loué entre 7h et 17h.

Votre programme que vous allez nommer `Velo.py` demandera à l'utilisateur de quelle heure à quelle heure se fait la location et calculera le prix de la location en conséquence.

Vous adopterez les simplifications suivantes :

- Les heures de début et fin de location sont des entiers (pas de demi ni de quart, toute heure entamée est due) ;
- L'heure du début de la location est toujours inférieure à l'heure de la fin de la location ; cela implique que la location ne peut pas se faire sur plus de 24 heures ; elle doit se faire dans la même journée.

Si les données introduites sont correctes, votre programme affichera simplement le coût de la location en respectant strictement les formats donnés dans les exemples de déroulement ci-dessous.

En cas de donnée incorrecte, votre programme devra afficher un message d'erreur et inviter l'utilisateur à corriger ces erreurs. Utilisez strictement les messages suivants :

- « Les heures doivent être comprises entre 0 et 24 ! » suivi d'un saut de ligne, si une des heures introduites par l'utilisateur n'est pas comprise entre 0 et 24 (inclus) ;
- « Attention ! l'heure de fin est identique à l'heure de début. » suivi d'un saut de ligne, si les heures de début et fin de location sont identiques ;
- « Attention ! le début de la location est après la fin ... » suivi d'un saut de ligne si l'heure de début de la location est supérieure à l'heure de fin.

**Exemples de déroulement :**

Il est impératif que votre code respecte le format de réponse suivant :

1) Exemple où la durée de location implique les deux tarifs :

Donnez l'heure de début de la location (un entier) : 10

Donnez l'heure de fin de la location (un entier) : 19

Vous avez loué votre vélo pendant

2 heure(s) au tarif horaire de 1.0 euro(s)

7 heure(s) au tarif horaire de 2.0 euro(s)

Le montant total à payer est de 16.0 euro(s).

2) Exemple où la durée de location n'implique qu'un seul tarif :

Donnez l'heure de début de la location (un entier) : 18

Donnez l'heure de fin de la location (un entier) : 20

Vous avez loué votre vélo pendant

2 heure(s) au tarif horaire de 1.0 euro (s)

Le montant total à payer est de 2.0 euro (s).

**Point d'étape 2 à faire valider**

Référence :

TP de Jamila Sam, EPFL