

EXERCICE 1. ** Contrôle de saisie

Il est fréquent de devoir vérifier qu'une donnée fournie par l'utilisateur vérifie certaines conditions.

- Écrivez un programme Python qui demande à l'utilisateur de saisir un nombre entier tant que celui-ci ne sera pas compris entre 0 compris et 10 exclus.
- Écrivez un second programme qui vérifiera la condition contraire de la précédente.

Vous nommerez ce programme `ex1x_controle_saisie.py` en remplaçant **x** par **a** ou **b**.

EXERCICE 2. ** Ni oui, ni non

Écrivez un programme qui simule le jeu de "Ni oui, ni non", c'est-à-dire qu'il est demandé à l'utilisateur d'entrer une chaîne de caractères (qui sera assimilée à la réponse à une question) tant que cette chaîne de caractères n'est ni oui, ni non. En revanche si la chaîne entrée est oui ou non, le jeu s'arrête.

Vous nommerez ce programme `ex2_ni_oui_ni_non.py`.

EXERCICE 3. ** Somme des entiers dépassant le million

Écrivez un programme Python qui détermine à partir de quelle valeur de n la somme $1 + 2 + 3 + \dots + n$ dépasse 1 000 000.

Vous nommerez ce programme `ex3_somme_depasant_le_million.py`.

EXERCICE 4. * Puissances de 2**

- Écrivez un programme Python qui affiche les puissances de 2 les unes à la suite des autres (c'est-à-dire sans saut de ligne) tant qu'elles sont inférieures à un nombre maximum demandé à l'utilisateur.
- Écrivez une deuxième version du programme qui n'affichera que la dernière puissance de 2 inférieure ou égale au maximum.
- Écrivez une troisième version du programme qui affichera la plus grande valeur de n telle que $2^n \leq max$.

Vous nommerez ces programmes `ex4x_puissances_2.py` en remplaçant **x** par **a**, **b** ou **c**.

EXERCICE 5. * Jeu "Deviner un nombre"**

Écrivez un programme Python qui propose à l'utilisateur de deviner un nombre compris entre 1 et 100. À chaque proposition de l'utilisateur, vous lui indiquerez si le nombre proposé est supérieur ou inférieur au nombre à deviner. Pour tirer au sort un nombre entier compris entre deux bornes, utiliser la fonction `randint` de la bibliothèque `random` que vous importerez en début de programme.

```
import random as ra
nombre_a_deviner = ra.randint(1, 100)
```

- Ajoutez une fonctionnalité à votre programme qui consiste à préciser à l'utilisateur, en fin de jeu, le nombre de propositions qu'il a dû faire pour deviner.
- Quelle est la stratégie la plus efficace pour gagner ? Montrez qu'on peut deviner un nombre en sept coups maximum.

Vous nommerez ce programme `ex07_deviner_un_nombre.py`.