TRAVAUX PRATIQUES n° 3
Instructions itératives : boucles for

Pour bien démarrer ce TP, commencez par lire la fiche méthode suivante :

⋆ **I5** - Utiliser une boucle `for`

Exercice 1 (*Échauffement*) :

1) Exécuter un par un les scripts suivants :

```
## Script 1
for k in range(10):
    print(k)

## Script 2
for i in range(10):
    print(i)

## Script 3
for k in range(4, 12):
    print(k)
```

2) Étant donnés deux nombres entiers `a` et `b`, on considère les deux scripts suivants :

```
## Script 1
for x in range(a):
    print(x)

## Script 2
for x in range(a, b):
    print(x)
```

Pour chacun de ces scripts, répondre aux questions suivantes :

a) `x` est-elle une variable ?
b) Quelles sont les valeurs successives de `x` au cours de la boucle ?
c) Combien de fois l'instruction `print(x)` est-elle exécutée ?
d) Est-ce qu'il y a une différence entre l'instruction `range(a)` et l'instruction `range(0, a)` ?

3) Écrire un script qui affiche tous les nombres entiers entre `1` et `100`.
4) Écrire un script qui affiche tous les nombres entiers pairs entre `1` et `100`.

On pourra tester la parité avec `k \% 2 == 0`.

5) Pour chacun des scripts suivants, deviner la valeur de la variable `x` à la fin de ce script, puis vérifier en l'exécutant :

```
## Script 1
x = 0
for k in range(5):
    x = 2 * x + 1

## Script 2
x = 0
for k in range(10, 15):
    x = 2 * x + 1
```

```python
## Script 3
x = 0
for k in range(2, 5):
    x = 2 * x + 1

## Script 4
x = 0
for k in range(2, 5):
    x = 2 * x + k
```

Exercice 2 (*Chantons !*) :

1) Écrire un script qui demande à l'utilisateur d'entrer un nombre de kilomètres puis qui affiche les paroles de la chanson suivante jusqu'au nombre de kilomètres demandés :

```
1 kilomètre à pied, ça use, ça use, 1 kilomètre à pied, ça use les souliers.
2 kilomètre à pied, ça use, ça use, 2 kilomètre à pied, ça use les souliers.
3 kilomètre à pied, ça use, ça use, 3 kilomètre à pied, ça use les souliers.
...
```

2) Améliorer votre script pour que le mot `kilomètre` soit au pluriel à partir de `2 kilomètres`.

Accumulateur

Pour calculer une somme ou un produit avec une boucle `for`, on utilise une variable accumulateur :

- ⋆ on l'**initialise** avant la boucle (`S = 0` pour une somme, `P = 1` pour un produit) ;
- ⋆ on la **met à jour** à chaque tour de boucle (`S = S + ...` ou `P = P * ...`) ;
- ⋆ on **affiche** (ou on utilise) sa valeur après la boucle.

Un produit commence à `1` et non à `0` : sinon le résultat resterait nul quel que soit le nombre de facteurs.

```python
S = 0
for k in range(1, n + 1):
    S = S + k
print(S)
```

Exercice 3 (*Calcul de sommes*) :

1) Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur de la somme :

$$1 + 2 + 3 + \cdots + n \overset{\text{noté}}{=} \sum_{k=1}^{n} k$$

Vérification : le script doit afficher 0 *pour* $n = 0$ *et* 5050 *pour* $n = 100$.

2) Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur de la somme :

$$1^2 + 2^2 + 3^2 + \cdots + n^2 \overset{\text{noté}}{=} \sum_{k=1}^{n} k^2$$

Vérification : le script doit afficher 0 *pour* $n = 0$ *et* 338350 *pour* $n = 100$.

3) Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur de la somme :

$$5e^5 + 6e^6 + 7e^7 + ... + ne^n \quad \overset{\text{noté}}{=} \quad \sum_{k=5}^{n} ke^k$$

Vérification : le script doit afficher 0 *pour* $n < 5$ *et environ* 327879.15 *pour* $n = 10$.

Exercice 4 (*Calcul de produits*) :

1) Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur du produit :

$$1 \times 2 \times 3 \times \cdots \times n \quad \overset{\text{noté}}{=} \quad \prod_{k=1}^{n} k \quad \overset{\text{noté}}{=} \quad n!$$

Vérification : le script doit afficher 1 *pour* $n = 0$ *et* 3628800 *pour* $n = 10$.

2) Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur du produit :

$$P = 2\prod_{k=1}^{n} \frac{4k^2}{4k^2 - 1}$$

Vérification : le script doit afficher 2 *pour* $n = 0$ *et environ* 3.14 *(*π*) pour* $n = 1000$.

Exercice 5 (*Calcul des termes d'une suite numérique*) : *Schéma général : on initialise* `u` *à la valeur de* u_0*, puis on effectue* n *tours de boucle pour obtenir* u_n.

1) On considère la suite $(u_n)_{n\geqslant 0}$ définie par

$$u_0 = 3 \qquad \text{et} \qquad \forall n \in \mathbb{N}, \quad u_{n+1} = 2u_n - 1$$

Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche tous les termes de la suite $(u_n)_{n\geqslant 0}$ de u_0 jusqu'à u_n.

Vérification : $u_{10} = 2049$.

2) Modifier ce dernier script pour qu'il n'affiche que la valeur de u_n.

3) On considère la suite $(u_n)_{n\geqslant 0}$ définie par

$$u_0 = 3 \qquad \text{et} \qquad \forall n \in \mathbb{N}, \quad u_{n+1} = 2u_n + n$$

Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur de u_n.

Vérification : $u_{10} = 4085$.

4) On considère la suite $(u_n)_{n\geqslant 4}$ définie par

$$u_4 = -3 \qquad \text{et} \qquad \forall n \geqslant 4, \quad u_{n+1} = (n-1)u_n - 1$$

Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur de u_n.

Vérification : $u_{10} = -69281$.

5) On considère la suite $(u_n)_{n\geqslant 0}$ définie par

$$u_0 = 3, \qquad u_1 = -1 \qquad \text{et} \qquad \forall n \in \mathbb{N}, \quad u_{n+2} = u_{n+1} + 3u_n + 1$$

Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule et affiche la valeur de u_n.

Vérification : $u_{10} = 4307$.

Exercice 6 (*Suite de Fibonacci*) : Dans cet exercice, on va décrire l'évolution d'une population de lapins selon le modèle de Fibonacci. Pour tout $n \in \mathbb{N}$, on note F_n le nombre de lapins en vie à la n^{e} génération, avec pour condition initiale $F_0 = 1$.

Le modèle est le suivant :

- ⋆ Chaque lapin en âge de procréer à la génération n va produire un nouveau lapin à la génération $n+1$.
- ⋆ Un lapin est capable de procréer au bout d'une génération après sa naissance.
- ⋆ Le premier lapin de la génération 0 vient de naître.
- ⋆ Les lapins sont immortels.

1) Justifier brièvement que l'on a $F_1 = 1$.
2) En déduire brièvement que
$$\forall n \in \mathbb{N}, \qquad F_{n+2} = F_{n+1} + F_n$$
3) Écrire un script qui demande à l'utilisateur d'entrer un nombre entier naturel `n` et qui calcule le nombre de lapins à la n^{e} génération.

Remarque : De façon étonnante, on retrouve les nombres de Fibonacci un peu partout dans la nature. Par exemple, le nombre de pétales d'une marguerite ou d'un tournesol appartient à la suite de Fibonacci. Ou encore dans la conversion des miles en kilomètres : on a pour tout $n \in \mathbb{N}^*$, F_n miles $\approx F_{n+1}$ km.