

Pour ce TP, vous allez devoir créer un programme Python qui génère et affiche une table de multiplication sous forme de grille.

Objectifs de l'exercice :

- Utiliser la boucle `for` pour générer des lignes et des colonnes.
- Utiliser la boucle `while` pour une interaction continue avec l'utilisateur.
- Utiliser les mots-clés `break` et `continue` pour le contrôle du flux dans les boucles.

Énoncé de l'exercice :

Vous allez écrire un programme Python qui demande à l'utilisateur de saisir un nombre entier positif. Le programme doit ensuite afficher la table de multiplication de ce nombre jusqu'à 10.

Étapes à suivre :

1. Demander un nombre à l'utilisateur :

- Utiliser une boucle `while` pour continuer à demander un nombre tant que l'utilisateur n'entre pas un entier positif.
- Si l'utilisateur entre un nombre négatif ou zéro, afficher un message d'erreur et continuer à demander un nombre.

2. Générer et afficher la table de multiplication :

- Utiliser une boucle `for` pour générer les lignes de la table de multiplication.
- Pour chaque ligne, utiliser une autre boucle `for` pour générer les colonnes (de 1 à 10).
- Afficher chaque ligne de la table de multiplication.

3. Option de recommencer ou de quitter :

- Après avoir affiché la table de multiplication, demander à l'utilisateur s'il souhaite entrer un autre nombre ou quitter le programme.
- Utiliser un mot-clé `break` pour sortir de la boucle principale si l'utilisateur choisit de quitter.
- Utiliser un mot-clé `continue` pour recommencer la boucle si l'utilisateur veut entrer un autre nombre.

Exemple de sortie attendue :

```
Entrez un nombre entier positif : -5
Erreur : Veuillez entrer un nombre entier positif.
Entrez un nombre entier positif : 0
Erreur : Veuillez entrer un nombre entier positif.
Entrez un nombre entier positif : 5
```

Table de multiplication de 5 :

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

```
Voulez-vous entrer un autre nombre ? (oui/non) : oui
Entrez un nombre entier positif : 3
```

Table de multiplication de 3 :

```
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

```
Voulez-vous entrer un autre nombre ? (oui/non) : non
Au revoir !
```

Contraintes :

- Vous devez utiliser au moins une boucle `for` et une boucle `while` .
- Vous devez utiliser les mots-clés `break` et `continue` au moins une fois dans votre programme.

- Gérez les entrées incorrectes de l'utilisateur de manière appropriée en affichant des messages d'erreur clairs.

Bonne chance !