# Les structures de contrôle répétitives en Python

## Engage

**Durée**: 15 minutes

Pour commencer, posez aux étudiants la question suivante : "Avez-vous déjà remarqué que certaines tâches dans la vie de tous les jours sont répétitives ? Pouvez-vous donner des exemples ?" Laissez les étudiants partager leurs idées. Expliquez ensuite que les structures de contrôle répétitives en programmation permettent d'automatiser ce type de tâches répétitives.

Donnez un exemple simple de tâche répétitive, comme compter de 1 à 10. Demandez aux étudiants comment ils procéderaient pour réaliser cette tâche manuellement. Discutez ensuite de la façon dont on pourrait automatiser cette tâche à l'aide d'une structure de contrôle répétitive en Python.

## Explore

**Durée**: 30 minutes

Présentez les deux principales structures de contrôle répétitives en Python : la boucle for et la boucle while. Expliquez les différences entre ces deux structures et quand il est préférable d'utiliser l'une ou l'autre.

Ensuite, guidez les étudiants à travers quelques exemples d'utilisation de ces boucles, en mettant l'accent sur leur syntaxe et leur fonctionnement. Voici quelques exemples :

* Utiliser une boucle for pour parcourir une liste et afficher chaque élément.
* Utiliser une boucle while pour demander à l'utilisateur de saisir un nombre jusqu'à ce qu'il entre une valeur valide.
* Combiner une boucle for et une instruction if pour ne afficher que les nombres pairs d'une liste.

Encouragez les étudiants à essayer ces exemples par eux-mêmes et à poser des questions s'ils ont des difficultés.

## Explain

**Durée**: 20 minutes

Approfondissez la compréhension des étudiants en expliquant les concepts clés des structures de contrôle répétitives :

* Le rôle de la condition de boucle dans les boucles for et while.
* L'importance de l'indentation pour délimiter le bloc de code à exécuter dans la boucle.
* L'utilisation des instructions break et continue pour contrôler le déroulement de la boucle.
* Les différences entre les boucles for et while en termes de syntaxe et d'utilisation.

Illustrez ces concepts à l'aide d'exemples supplémentaires et encouragez les étudiants à poser des questions pour clarifier leur compréhension.

## Elaborate

**Durée**: 30 minutes

Demandez aux étudiants de résoudre des problèmes plus complexes impliquant l'utilisation de structures de contrôle répétitives. Voici quelques suggestions :

* Écrire un programme qui affiche la table de multiplication d'un nombre donné.
* Créer un programme qui calcule la somme de tous les nombres pairs entre 1 et 100.
* Développer un jeu simple où l'utilisateur doit deviner un nombre choisi aléatoirement, en utilisant une boucle while pour gérer les tentatives.

Encouragez les étudiants à travailler en équipe pour résoudre ces problèmes et à partager leurs solutions avec la classe. Fournissez-leur une rétroaction constructive et aidez-les à surmonter les défis qu'ils rencontrent.

## Evaluate

**Durée**: 15 minutes

Pour évaluer la compréhension des étudiants, proposez-leur un exercice d'évaluation individuel. Par exemple :

Écrivez un programme Python qui affiche les nombres de 1 à 20, en sautant les multiples de 3.

Donnez aux étudiants 10-15 minutes pour résoudre cet exercice, puis demandez-leur de partager leurs solutions. Fournissez une rétroaction et des explications supplémentaires si nécessaire.

Alignement des normes :

* **Norme 1**: Comprendre les structures de contrôle répétitives en Python et leur utilisation dans la résolution de problèmes.
* **Norme 2**: Être capable d'écrire et de déboguer du code Python utilisant des boucles for et while.