



Python sur Trinket – Leçon 7

Révision

- Opérateurs mathématiques
- Listes

Discussion

- Des boucles de conditions

- o Répète une partie du code un nombre de fois indéterminé, jusqu'à ce qu'une condition est remplie.

```
while condition:  
    statement(s)
```

- o Nous utilisons les boucles lorsque nous savons exactement le nombre de fois que nous voulons exécuter un bloc de code (par exemple, 10 fois afin de créer un flocon de neige avec 10 dendrites)
 - o Des boucles de conditions sont utilisées lorsque nous ne connaissons pas le nombre de fois que nous voulons exécuter le code. L'exécution du code dépend ainsi d'une condition qui doit être réalisée.
 - Exemple: Si nous demandons à l'utilisateur de deviner un chiffre se situant entre le 1 et 10, il est impossible de savoir le nombre de fois où l'utilisateur se trompera avant d'obtenir la bonne réponse. La condition est ainsi le chiffre à deviner.
 - Les boucles de conditions peuvent devenir infinies si la condition n'est jamais remplie, ce qui peut faire boguer le programme. Il faut ainsi s'assurer que la condition est éventuellement remplie.
 - o Utiliser une boucle de conditions afin de recevoir la bonne saisie d'utilisateur
 1. Demander la saisie à l'utilisateur
 2. La saisie est incorrecte
 3. Demander une nouvelle saisie
 4. Arrêter la boucle lorsque la saisie est valide
-

Leçon

Devine le chiffre	
<pre>number = 7 guess = input("devine un chiffre entre 1 et 10 ") while int(guess) != number: guess = input("essaie encore") print("félicitations!")</pre>	<pre>Devine un chiffre entre 1 et 10 5 Essaie encore 7 Félicitations!</pre>

Pratique

- Créer votre propre jeu de devinette
 - o Demander le nom à l'utilisateur et introduisez-lui au jeu
 - o Utilisez les boucles de conditions afin que celui-ci soit appelé à faire des saisies respectant des conditions que vous avez déterminées.
 - o Félicitez l'utilisateur lorsqu'il a gagné
- Prolonger votre jeu de devinette . Par exemple ..
 - o Demander le niveau de difficulté à l'utilisateur (facile- deviner un nombre entre 1 et 10; moyen- 1 et 20; difficile- 1 et 50)

```
difficulty = input("entrez un niveau de difficulté -
facile/moyen/difficile")

if difficulty == "facile":
    max = 10
elif difficulty == "moyen":
    max = 20
elif difficulty == "difficile":
    max = 50

number = random.randint(1, max)
```

- o Lorsque l'utilisateur ne donne pas le bon chiffre, donnez-lui un indice en lui indiquant si la réponse est plus haut ou plus bas que le chiffre choisi.