

Fin du cours d'introduction à Python

Pierre Jaumier

Objectifs pédagogiques

À l'issue de cette séance, les participants seront capables de : - Créer un mini-jeu interactif en Python - Utiliser les variables pour stocker et manipuler des données - Comprendre et utiliser la boucle `while` - Gérer les saisies utilisateur avec `input()` - Convertir des types (chaînes → entiers) - Utiliser les structures conditionnelles (`if...elif...else`) - Gérer les erreurs avec les exceptions (`try...except`)

Introduction au jeu “Devine mon nombre !”

- Le but du jeu : deviner un nombre tiré au sort par l'ordinateur entre 1 et 30.
- Le joueur dispose de 5 essais maximum.
- À chaque tentative, le programme indique si le nombre proposé est trop grand ou trop petit.

Code complet du jeu (début)

```
from random import randint

nbr_essais_max = 5
nbr_essais = 1
borne_sup = 30
mon_nombre = randint(1, borne_sup)
ton_nombre = 0

print("J'ai choisi un nombre entre 1 et", borne_sup)
print("A vous de le deviner en", nbr_essais_max, "tentatives au maximum!")
```

Code complet du jeu (fin)

```
while ton_nombre != mon_nombre and nbr_essais <= nbr_essais_max:
    print("Essai no", nbr_essais)
    ton_nombre = int(input("Votre proposition: "))
    if ton_nombre < mon_nombre:
        print("Trop petit")
    elif ton_nombre > mon_nombre:
        print("Trop grand")
    else:
        print("Bravo! Vous avez trouvé", mon_nombre, "en", nbr_essais, "essai(s)")
    nbr_essais += 1

if ton_nombre != mon_nombre:
    print("Désolé, vous avez utilisé vos", nbr_essais_max, "essais en vain.")
    print("J'avais choisi le nombre", mon_nombre, ".")
```

Variables

Définition

Une variable est un contenant qui stocke une valeur.

En Python, on n'a pas besoin de déclarer le type d'une variable à l'avance.

Exemples

```
nbr_essais_max = 5      # entier
borne_sup = 30         # entier
mon_nombre = randint(1, borne_sup) # nombre aléatoire
ton_nombre = 0         # initialisation
```

On peut voir une variable comme une boîte étiquetée dans laquelle on place une valeur.

Boucle while

Quand l'utiliser ?

Quand on ne connaît pas à l'avance le nombre d'itérations.

Syntaxe

```
while condition:
    # bloc de code répété tant que la condition est vraie
```

Dans notre jeu :

```
while ton_nombre != mon_nombre and nbr_essais <= nbr_essais_max:
    ...
```

Incrémentation

Pour compter les essais, on incrémente la variable `nbr_essais` :

```
nbr_essais += 1
```

Cela revient à écrire :

```
nbr_essais = nbr_essais + 1
```

C'est une opération très courante dans les boucles.

Saisie utilisateur : `input()`

La fonction `input()` permet de récupérer une chaîne de caractères saisie par l'utilisateur.

Exemple :

```
ton_nombre = int(input("Votre proposition: "))
```

Attention : `input()` retourne toujours une **chaîne de caractères** (`str`). Il faut la convertir en **entier** (`int`) si besoin.

Gestion des erreurs : Exceptions

Si l'utilisateur saisit autre chose qu'un nombre, le programme plante.

Solution : utiliser try...except

```
while True:
    try:
        ton_nombre = int(input("Votre proposition: "))
        break
    except ValueError:
        print("Réponse non valide. Réessayez!")
```

Cette structure protège le programme des erreurs imprévues.

Structures conditionnelles : if...elif...else

Permet de prendre des décisions selon les valeurs.

```
if ton_nombre < mon_nombre:
    print("Trop petit")
elif ton_nombre > mon_nombre:
    print("Trop grand")
else:
    print("Bravo!")
```

Très utile pour gérer les différents cas possibles dans un jeu.

Bonnes pratiques de programmation

Noms de variables

- En minuscules
- Avec des mots significatifs
- snake_case: Sans espaces ni caractères spéciaux (utiliser _)

Commentaires

- Ne pas commenter ce qui est évident
- Expliquer pourquoi plutôt que comment
- Garder le code clair et lisible

Exercices supplémentaires

Exercice 1 : Deviner avec aide personnalisée

Demander le prénom du joueur et l'appeler par son nom pendant le jeu.

Exercice 2 : Rejouer plusieurs fois

Proposer de rejouer après chaque partie. Calculer : - Le pourcentage de réussite - Le nombre moyen d'essais pour les parties gagnées

Exercice 3 : Mode multijoueur

Faire jouer deux joueurs à tour de rôle jusqu'à ce que l'un trouve le nombre.

Concepts abordés dans ce chapitre

Concept	Utilité
Variables	Stocker des données
Boucle <code>while</code>	Répéter des actions sans connaître le nombre d'itérations
Incrémentation	Compter les tours ou les essais
Conditions	Prendre des décisions
<code>input()</code> / <code>int()</code>	Récupérer et convertir des données utilisateur
Exceptions	Gérer les erreurs sans planter le programme
Commentaires	Améliorer la lisibilité du code

Pour aller plus loin

- Formatage des nombres avec `.format()`
- Mesure du temps avec `time.time()`
- Génération de statistiques
- Jeux avec choix multiples