



TP 1 : Premiers pas vers l'IA avec Python-Solution

Objectifs :

- Exécuter un programme *Python* simple.
- Manipuler les notions de variable, type et interaction utilisateur.
- Identifier le rôle de l'indentation dans la structure d'un programme Python.
- Comprendre et utiliser les structures conditionnelles (*if / elif / else*).
- Expliquer comment un programme peut prendre une décision.
- Identifier le principe d'une IA faible basée sur des règles.
- Utiliser un outil d'intelligence artificielle (comme *ChatGPT*) pour rechercher des explications, comprendre le code et répondre aux questions, tout en gardant un esprit critique.

Exercice 1 : Interaction simple avec l'utilisateur

Copiez et exécutez le programme suivant, puis répondez aux questions ci-dessous :

```
name = input("Enter your name: ")
age = int(input("Enter your age: "))
print("Hello", name)
print("You are", age, "years old.")
print("Next year, you will be", age + 1)
```

1) Quel est le rôle de la fonction *input()* ?

Permet de récupérer une donnée saisie par l'utilisateur.

2) Que contient la variable *name* ?

Contient le nom de l'étudiant sous forme de chaîne de caractères.

3) Pourquoi utilise-t-on *int()* pour l'âge ?

Convertit la chaîne en entier.

4) Que se passe-t-il si on supprime *int()* ?

Sans *int()*, l'âge serait considéré comme du texte

Erreur lors de *age + 1*, car on ne peut pas additionner string et entier

5) Quel est le rôle de la fonction *print()* ?

Affiche un message à l'écran.

6) Quelle est la différence entre *"age"* et *age* ?

"age" : texte (String)

age: variable contenant une valeur

7) Ce programme est-il interactif ? Pourquoi ?

Oui, car il attend une entrée utilisateur (*input*), puis réagit en fonction de cette entrée.

Exercice 2 : Utilisation de la structure conditionnelle *if* et l'indentation

Copiez et exécutez le programme suivant, puis répondez aux questions ci-dessous :

```
# Extrait 1. Ceci est un
# commentaire Python.
password = "python123"
if password == "python123":
    print("Mot de passe correct")
    print("Accès autorisé")
print("Fin du programme")
```

```
# Extrait 2. Ceci est un autre
# commentaire Python.
password = "python123"
if password == "python123":
    print("Mot de passe correct")
print("Accès autorisé")
print("Fin du programme")
```

1) Quelle est la différence entre == et = en *Python* ?

= : opérateur d'affectation. Il permet d'assigner une valeur à une variable.

== : opérateur de comparaison. Il permet de vérifier si deux valeurs sont égales et renvoie un booléen (True ou False)

2) Quel est le rôle de l'instruction *if* ?

Permet d'exécuter du code sous condition.

3) Les deux programmes affichent-ils exactement le même résultat ? Quelle est la différence ?

Oui, si le mot de passe est correct.

4) Exécutez les deux extraits précédents avec un mot de passe incorrect, puis refaire la question 3.

Non, les deux programmes n'affichent pas le même résultat.

Dans le premier programme, les deux instructions *print()* sont à l'intérieur du bloc *if*, donc elles ne s'exécutent que si la condition est vraie.

Dans le deuxième programme, l'instruction : *print("Accès autorisé")* n'est pas indentée, donc elle s'exécute toujours, même si le mot de passe est incorrect.

5) Certaines lignes sont décalées vers la droite. Comment appelle-t-on ce décalage en *Python* et quel est son rôle dans l'exécution du programme ? »

Le décalage vers la droite s'appelle l'indentation.

En *Python*, l'indentation sert à définir les blocs d'instructions.

Elle indique quelles instructions appartiennent à une structure comme *if*, *elif*, *else*, une boucle, etc.

Contrairement au langage *C* qui utilise des accolades { }, *Python* utilise l'indentation pour organiser et structurer le programme.

Exercice 3 : Prise de décision simple (conditions)

Copiez et exécutez le programme suivant :

```
sleep_hours = int(input("How many hours do you sleep per night? "))
if sleep_hours < 6:
    print("You should sleep more for better health.")
elif sleep_hours <= 8:
    print("Good! You are getting a healthy amount of sleep.")
else:
    print("You are sleeping a lot! Make sure it doesn't affect your daily activities.")
```

1) Pourquoi utilise-t-on *elif* ?

Permet tester une nouvelle condition si la précédente est fausse.

2) Quand le bloc *else* est-il exécuté ?

S'exécute si toutes les conditions précédentes sont fausses.

3) Quel message est affiché si l'étudiant saisit 7 ?

"Good! You are getting a healthy amount of sleep."

4) Combien de messages peuvent être affichés en même temps ?

Un seul, car blocs *if* / *elif* / *else* sont mutuellement exclusifs : une seule condition peut être exécutée.

5) Ce programme prend-il une décision ? Expliquez.

Oui, car il choisit un message selon une condition et l'entrée.

Exercice 4 : Mini assistant intelligent pour étudiant

Ecrivez le code *Python* suivant, exécutez-le, puis répondez aux questions ci-dessous.

```
field = input("Enter your field of study: ")
study_hours = int(input("Enter your study hours per day: "))

if study_hours < 2:
    print("Advice: You need to improve your study habits.")
else:
    if field == "Computer Science":
        print("Good profile for Computer Science.")
    else:
        print("Keep working in your field.")
```

1) Quelle condition doit être vérifiée pour afficher le message : "Good profile for Computer Science." ?
`study_hours >= 2 ET field == "Computer Science"`

2) Pourquoi utilise-t-on une structure conditionnelle imbriquée (un *if* à l'intérieur d'un autre *if*) ?
La structure conditionnelle imbriquée permet de tester une seconde condition uniquement si la première est vérifiée.

3) Que se passe-t-il si l'étudiant saisit : (1) "Computer Science" et 1 heure, (2) "Mathematics" et 5 heures ?

(1) "You need to improve your study habits."

(2) "Keep working in your field."

4) Ajoutez la ligne suivante à la fin du programme : `print("Fin du programme")`, Où devez-vous placer cette instruction (niveau d'indentation) pour qu'elle s'exécute dans tous les cas ?

Sans indentation (niveau 0).

5) Où sont définies les règles de décision du programme ?

Dans les conditions *if*.

6) Ce programme peut-il apprendre de nouvelles règles ? Justifiez.

Non. Il applique des règles fixes écrites dans le code ; il ne modifie pas son comportement à partir de nouvelles données.

7) Pourquoi peut-on considérer ce programme comme une forme d'Intelligence Artificielle faible ?

Il est considéré comme une IA faible car il accomplit une tâche spécifique en utilisant des règles prédéfinies et ne possède ni intelligence générale ni compréhension.

Exercice 5 : Amélioration simple

Ecrivez le code *Python* suivant, exécutez-le, puis répondez aux questions ci-dessous.

```
field = input("Enter your field of study: ")
study_hours = int(input("Enter your study hours per day: "))

if study_hours < 2:
    print("You should increase your study time.")
elif field == "Mathematics":
    print("Mathematics requires regular practice.")
elif field == "Computer Science":
    print("Good profile for Computer Science.")
else:
    print("Keep working and stay motivated.")
```

1) Ce programme est-il réellement intelligent ?

Non, il applique des règles prédéfinies.

2) Peut-il apprendre à partir de l'expérience ?

Non, il ne peut pas apprendre tout seul

3) Quelle est la principale limite de ce système ?

Sa principale limite est l'utilisation de règles fixes et limitées.

4) Quelle est la différence entre ce programme et l'apprentissage automatique (*Machine Learning*) ?

Contrairement à ce programme le machine learning apprend à partir des données.