

TP1 : Introduction à Python - Syntaxe, Variables, Types de Données, Structures Conditionnelles et Boucles

Objectifs

À la fin de ce TP, l'étudiant devra savoir :

1. Comprendre la syntaxe de base de Python.
2. Utiliser des variables et des types de données courants (entiers, chaînes de caractères, booléens).
3. Mettre en œuvre des structures conditionnelles (if-else).
4. Utiliser des boucles (for et while) pour itérer sur des éléments.
5. Manipuler des chaînes de caractères et des entrées utilisateur.

1. Syntaxe de Base et Variables

Exercice 1 : Affichage et Variables

1. Affichez un message de bienvenue en utilisant la fonction print.
2. Créez une variable nom et une variable age, puis affichez une phrase les combinant (exemple : "Bonjour [nom], vous avez [age] ans.").

Solution attendue

```
# Affichage d'un message
print("Bienvenue dans le TP Python !")

# Déclaration de variables
nom = "Alice"
age = 25

# Affichage combiné des variables
print(f"Bonjour {nom}, vous avez {age} ans.")
```

2. Types de Données et Opérations

Exercice 2 : Types de Données et Opérations Simples

1. Créez une variable x de type entier et une variable y de type flottant.
2. Calculez et affichez la somme, la différence, le produit et le quotient de x et y.

Solution attendue

```
# Déclaration des variables
x = 10 # entier
y = 5.5 # flottant

# Calculs et affichage
print("Somme:", x + y)
```

```
print("Différence:", x - y)
print("Produit:", x * y)
print("Quotient:", x / y)
```

3. Structures Conditionnelles (if-else)

Exercice 3 : Structures Conditionnelles

1. Demandez à l'utilisateur d'entrer un nombre et affichez s'il est pair ou impair.
2. Implémentez une condition pour vérifier si le nombre est positif, négatif ou nul.

Solution attendue

```
# Demande à l'utilisateur d'entrer un nombre
nombre = int(input("Entrez un nombre : "))

# Vérification de la parité
if nombre % 2 == 0:
    print(f"{nombre} est un nombre pair.")
else:
    print(f"{nombre} est un nombre impair.")

# Vérification si le nombre est positif, négatif ou nul
if nombre > 0:
    print(f"{nombre} est positif.")
elif nombre < 0:
    print(f"{nombre} est négatif.")
else:
    print("Le nombre est nul.")
```

4. Boucles for et while

Exercice 4 : Boucle for

1. Utilisez une boucle for pour afficher les nombres de 1 à 10.
2. Affichez les carrés de ces nombres (exemple : le carré de 2 est 4).

Solution attendue

```
# Boucle for pour afficher les nombres de 1 à 10
for i in range(1, 11):
    print(i)

# Boucle for pour afficher les carrés des nombres de 1 à 10
for i in range(1, 11):
    print(f"Le carré de {i} est {i**2}.")
```

Exercice 5 : Boucle while

1. Utilisez une boucle while pour afficher les nombres de 1 à 10.
2. Demandez à l'utilisateur de saisir un nombre et affichez la multiplication de ce nombre par 2 à chaque itération. Arrêtez lorsque le résultat dépasse 20.

Solution attendue

```
# Boucle while pour afficher les nombres de 1 à 10
```

```
i = 1
while i <= 10:
    print(i)
    i += 1
```

```
# Boucle while pour multiplier un nombre par 2 à chaque itération
```

```
nombre = int(input("Entrez un nombre pour multiplier par 2 : "))
resultat = nombre
```

```
while resultat <= 20:
    print(resultat)
    resultat *= 2
```

5. Manipulation des Chaînes de Caractères et Entrées Utilisateur

Exercice 6 : Manipulation de Chaînes

1. Demandez à l'utilisateur de saisir une phrase.
2. Affichez la longueur de la phrase.
3. Affichez la phrase en majuscules et en minuscules.

Solution attendue

```
# Demander à l'utilisateur de saisir une phrase
```

```
phrase = input("Entrez une phrase : ")
```

```
# Affichage de la longueur de la phrase
```

```
print(f"La longueur de votre phrase est : {len(phrase)} caractères.")
```

```
# Affichage de la phrase en majuscules et minuscules
```

```
print(f"En majuscules : {phrase.upper()}")
```

```
print(f"En minuscules : {phrase.lower()}")
```

Quiz (10 questions)

1. Quel est le résultat de l'expression $3 + 4.5$ en Python ?
 - a) 7.5
 - b) 7

- c) "34.5"
- d) "3 + 4.5"

2. **Quelle fonction permet d'afficher un message à l'écran ?**

- a) `print()`
- b) `input()`
- c) `display()`
- d) `output()`

3. **Quel est le type de la variable `x = 10` ?**

- a) `float`
- b) `int`
- c) `string`
- d) `boolean`

4. **Comment tester si une variable `x` est paire en Python ?**

- a) `if x == 0:`
- b) `if x % 2 == 0:`
- c) `if x = 2:`
- d) `if x != 2:`

5. **Que fait l'instruction `input()` ?**

- a) Affiche une chaîne de caractères.
- b) Permet à l'utilisateur de saisir une valeur.
- c) Déclare une variable.
- d) Effectue une addition.

6. **Quelle commande permet de boucler de 1 à 10 en Python ?**

- a) `for i in range(1, 10):`
- b) `for i in range(1, 11):`
- c) `for i in range(1, 9):`

- d) `while i <= 10:`
7. **Que fait la commande `x.upper()` ?**
- a) Convertit x en majuscules.
 - b) Convertit x en minuscules.
 - c) Renvoie la longueur de x.
 - d) Inverse la casse de x.
8. **Quel est le résultat de `10 % 3` ?**
- a) 3
 - b) 1
 - c) 10
 - d) 0
9. **Comment déclarer une variable nom avec la valeur "Alice" ?**
- a) `nom = 'Alice'`
 - b) `nom = Alice`
 - c) `nom : "Alice"`
 - d) `nom => Alice`
10. **Quel type de boucle est utilisé pour répéter un bloc de code un nombre spécifique de fois ?**
- a) `while`
 - b) `for`
 - c) `repeat`
 - d) `do-while`
-

Conclusion

Ce TP vous a permis de découvrir les bases du langage Python, incluant la syntaxe, les variables, les types de données, les structures conditionnelles et les boucles. Ces concepts sont essentiels pour construire des programmes simples mais puissants. Vous serez à

même d'appliquer ces connaissances pour des projets plus complexes à mesure que vous progresserez.