

TP2 : Programmation Python : Boucles, Instructions de contrôle et Fonctions

Objectifs pédagogiques

À la fin de ce TP, vous serez capable de :

- Utiliser les **structures de contrôle** (`if`, `elif`, `else`)
- Manipuler les **boucles** (`for`, `while`)
- Écrire et utiliser des **fonctions** personnalisées
- Appliquer ces notions dans de **petits algorithmes pratiques**

Pré-requis

- Python ≥ 3.8 installé
- Notions de base sur les types (`int`, `float`, `str`, `list`, etc.)

Plan du TP

1. Instructions conditionnelles
2. Boucles (`for`, `while`)
3. Fonctions simples et avancées
4. Exercices d'application complets

PARTIE 1 — INSTRUCTIONS CONDITIONNELLES

☛ Exercice 1 : Pair ou impair

Demandez à l'utilisateur de saisir un nombre et affichez s'il est pair ou impair.

```
n = int(input("Entrez un nombre : "))  
  
if n % 2 == 0:  
    print(f"{n} est pair.")  
else:  
    print(f"{n} est impair.")
```

☛ Exercice 2 : Calcul de moyenne et appréciation

Énoncé : Demandez trois notes et affichez la moyenne ainsi qu'une appréciation :

- ≥ 16 : Excellent
- 12–15 : Bien
- 10–11 : Passable
- < 10 : Insuffisant

```
note1 = float(input("Note 1 : "))  
note2 = float(input("Note 2 : "))
```

```
note3 = float(input("Note 3 : "))

moyenne = (note1 + note2 + note3) / 3
print("Moyenne :", moyenne)

if moyenne >= 16:
    print("Appréciation : Excellent")
elif moyenne >= 12:
    print("Appréciation : Bien")
elif moyenne >= 10:
    print("Appréciation : Passable")
else:
    print("Appréciation : Insuffisant")
```

PARTIE 2 — BOUCLES

☛ Exercice 3 : Table de multiplication

Affichez la table de multiplication d'un nombre donné par l'utilisateur.

```
n = int(input("Entrez un nombre : "))
for i in range(1, 11):
    print(f"{n} x {i} = {n * i}")
```

☛ Exercice 4 : Somme des nombres de 1 à N

Demandez un nombre N et calculez la somme des nombres de 1 à N.

```
N = int(input("Entrez un nombre : "))
somme = 0
for i in range(1, N + 1):
    somme += i
print("Somme =", somme)
```

☛ Exercice 5 : Boucle while et compteur

Demandez à l'utilisateur des nombres jusqu'à ce qu'il entre 0.
Calculez la somme et le nombre d'entrées.

```
somme = 0
compteur = 0
n = int(input("Entrez un nombre (0 pour arrêter) : "))

while n != 0:
    somme += n
    compteur += 1
    n = int(input("Entrez un nombre (0 pour arrêter) : "))
```

```
if compteur > 0:
    print("Somme =", somme, "| Moyenne =", somme / compteur)
else:
    print("Aucune entrée.")
```

PARTIE 3 — FONCTIONS

☛ Exercice 6 : Fonction de calcul de factorielle

Définissez une fonction `factorielle(n)` qui renvoie $n!$.

```
def factorielle(n):
    resultat = 1
    for i in range(1, n + 1):
        resultat *= i
    return resultat

print("5! =", factorielle(5))
```

☛ Exercice 7 : Fonction avec conditions

Écrire une fonction `est_pair(n)` qui retourne `True` si n est pair, sinon `False`.

```
def est_pair(n):
    return n % 2 == 0

print(est_pair(10)) # True
print(est_pair(7)) # False
```

☛ Exercice 8 : Fonction avec valeurs par défaut

Écrire une fonction `saluer(nom="inconnu")` qui affiche un message de bienvenue.

```
def saluer(nom="inconnu"):
    print(f"Bonjour {nom}, bienvenue dans le TP2 !")

saluer()
saluer("Alice")
```

☛ Exercice 9 : Fonction retournant plusieurs valeurs

Écrire une fonction `operations(a, b)` qui renvoie la somme, la différence et le produit de deux nombres.

```
def operations(a, b):
    return a + b, a - b, a * b
```

```
s, d, p = operations(10, 5)
print("Somme =", s, "| Différence =", d, "| Produit =", p)
```

PARTIE 4 — MINI-PROJET D'APPLICATION

🎯 Mini-projet : Calculateur statistique

Créez un petit programme qui demande à l'utilisateur une liste de notes et calcule :

- la moyenne,
- la note max,
- la note min,
- et la variance.

```
def statistiques(notas):
    moyenne = sum(notas) / len(notas)
    var = sum((x - moyenne) ** 2 for x in notas) /
len(notas)
    return moyenne, max(notas), min(notas), var

n = int(input("Combien de notes ? "))
liste_notas = [float(input(f"Note {i+1} : ")) for i in
range(n)]

m, maxv, minv, v = statistiques(liste_notas)
print(f"Moyenne={m:.2f}, Max={maxv}, Min={minv},
Variance={v:.2f}")
```

☐ Question de réflexion

1. Quelle est la différence entre une boucle `for` et une boucle `while` ?
2. Que se passe-t-il si on oublie la condition d'arrêt d'une boucle `while` ?
3. Quelle est la différence entre `return` et `print()` dans une fonction ?

☐ Travail à rendre

- Un fichier **TP2_programmation_base.py** contenant tous les exercices corrigés.
- Une capture d'écran du mini-projet fonctionnant dans ton terminal.