

*Groupe : Ti
Durée : 1h30min*

*Enseignant : CHEBBIM
Année universitaire : 2024/2025*

Plan de TP : Initiation à la programmation Python

1. Introduction et configuration de l'environnement

- **Objectif** : Installer Python et un éditeur de texte adapté.
 - **Installation de Python** : Télécharger et installer Python à partir de python.org.
 - **Installation de Visual Studio Code (VS Code)** ou un autre éditeur de code comme PyCharm.
 - **Vérification de l'installation** : Ouvrir le terminal ou l'invite de commande et vérifier la version de Python avec la commande :


```
python --version
```
 - **Introduction à l'environnement Python** : Utilisation de l'interpréteur Python dans le terminal pour tester des commandes simples.

2. Syntaxe de base et premiers programmes

Objectif : Comprendre la syntaxe de base de Python et écrire des programmes simples.

2.1. Affichage de texte

- **Exercice** : Afficher "Bonjour, monde !" avec la fonction `print()`.

```
print("Bonjour, monde !")
```

2.2. Les commentaires

- **Exercice** : Ajouter des commentaires dans le code pour expliquer son fonctionnement.

```
# Ceci est un commentaire  
print("Cela affiche du texte.") # Commentaire à la fin de la ligne
```

2.3. Variables et types de données

- **Exercice** : Créer des variables de différents types (entiers, flottants, chaînes de caractères) et les afficher.

```
x = 10 # Variable entière  
y = 3.14 # Variable flottante  
nom = "Alice" # Variable chaîne de caractères  
print(x, y, nom)
```

3. Les opérateurs

- **Objectif** : Apprendre à utiliser les opérateurs arithmétiques et de comparaison.

3.1. Opérateurs arithmétiques

- **Exercice** : Utiliser les opérateurs arithmétiques pour effectuer des calculs simples.

```
a = 15  
b = 5  
print(a + b) # Addition  
print(a - b) # Soustraction  
print(a * b) # Multiplication  
print(a / b) # Division
```

3.2. Opérateurs de comparaison

- **Exercice** : Comparer deux valeurs et afficher le résultat.

```
a = 10  
b = 5  
print(a > b) # Est-ce que a est supérieur à b ?  
print(a == b) # Est-ce que a est égal à b ?
```

4. Les structures conditionnelles

- **Objectif** : Apprendre à prendre des décisions dans un programme à l'aide de `if`, `elif`, et `else`.

4.1. Structure `if`

- **Exercice** : Créer un programme qui affiche un message en fonction de la valeur d'une variable.

```
age = 18
if age >= 18:
    print("Vous êtes majeur.")
else:
    print("Vous êtes mineur.")
```

4.2. Structure `if-elif-else`

- **Exercice** : Ajouter plusieurs conditions dans un programme pour vérifier différentes tranches d'âge.

```
age = 25
if age < 18:
    print("Vous êtes mineur.")
elif age < 65:
    print("Vous êtes adulte.")
else:
    print("Vous êtes senior.")
```

5. Les boucles

- **Objectif** : Apprendre à répéter des actions avec les boucles `for` et `while`.

5.1. La boucle `for`

- **Exercice** : Afficher les nombres de 1 à 5 avec une boucle `for`.

```
for i in range(1, 6):
    print(i)
```

5.2. La boucle `while`

- **Exercice** : Afficher les nombres de 1 à 5 avec une boucle `while`.

```
i = 1
while i <= 5:
    print(i)
```

```
i += 1
```

6. Les fonctions

- **Objectif** : Comprendre comment définir et appeler des fonctions en Python.

6.1. Définition d'une fonction

- **Exercice** : Créer une fonction simple qui affiche un message.

```
def dire_bonjour():  
    print("Bonjour, tout le monde !")  
  
dire_bonjour() # Appel de la fonction
```

6.2. Fonction avec paramètres

- **Exercice** : Créer une fonction qui prend un paramètre et affiche un message personnalisé.

```
def saluer(nom):  
    print(f"Bonjour, {nom} !")  
  
print(saluer("Alice"))  
print(saluer("Bob"))
```

6.3. Fonction avec retour

- **Exercice** : Créer une fonction qui additionne deux nombres et retourne le résultat.

```
def additionner(a, b):  
    return a + b  
  
resultat = additionner(3, 4)  
print("Le résultat de l'addition est:", resultat)
```

7. Les listes

- **Objectif** : Apprendre à travailler avec des listes en Python.

7.1. Création et manipulation d'une liste

- **Exercice** : Créer une liste de nombres et afficher chaque élément de la liste.

```
nombres = [1, 2, 3, 4, 5]
for nombre in nombres:
    print(nombre)
```

7.2. Ajouter, supprimer des éléments dans une liste

- **Exercice** : Ajouter un élément à la fin de la liste et supprimer un élément.

```
nombres.append(6) # Ajouter un élément à la fin
print(nombres)

nombres.remove(3) # Supprimer un élément de la liste
print(nombres)
```

8. Les dictionnaires

- **Objectif** : Découvrir les dictionnaires, une structure de données clé-valeur.

8.1. Création et utilisation d'un dictionnaire

- **Exercice** : Créer un dictionnaire pour stocker des informations sur un étudiant (nom, âge, note).

```
etudiant = {
    "nom": "Alice",
    "âge": 20,
    "note": 15
}
print(etudiant["nom"])
print(etudiant["âge"])
```

8.2. Ajouter et modifier des éléments dans un dictionnaire

- **Exercice** : Ajouter une nouvelle clé et modifier une valeur existante.

```
etudiant["ville"] = "Paris" # Ajouter une nouvelle clé
etudiant["note"] = 18 # Modifier une valeur
print(etudiant)
```