Chapitre 2 : Bases de Python

2.1 Variables et Types de Données

Les variables en Python sont utilisées pour stocker des données. Elles ne nécessitent pas de déclaration explicite et leur type est déduit en fonction de la valeur assignée. Python prend en charge plusieurs types de données :

- **Chaîne de caractères (String) : Une suite de caractères, ex. "Bonjour, Monde!"**
- **Entier (Integer) : Nombres entiers, ex. 123**
- **Flottant (Float) : Nombres à virgule flottante, ex. 3.14**
- **Booléen (Boolean) : Représente True ou False**
- **Liste (List) : Une collection ordonnée et modifiable, ex. [1, 2, 3, 4, 5]**
- **Tuple (Tuple) : Une collection ordonnée et immuable, ex. (1, 2, 3)**
- **Dictionnaire (Dictionary) : Une collection de paires clé-valeur, ex. {"nom": "Alice", "âge": 25}**

Exemple d'utilisation des variables :

```
chaine = "Bonjour, Monde!"

entier = 123

flottant = 3.14

booleen = True

liste = [1, 2, 3, 4, 5]

tuple_items = (1, 2, 3)

dictionnaire = {"nom": "Alice", "âge": 25}


print(chaine)       # Affiche : Bonjour, Monde!

print(entier)       # Affiche : 123

print(flottant)     # Affiche : 3.14

print(booleen)      # Affiche : True

print(liste)        # Affiche : [1, 2, 3, 4, 5]
```

print(tuple_items) # Affiche : (1, 2, 3)

print(dictionnaire) # Affiche : {'nom': 'Alice', 'âge': 25}

2.2 Opérateurs et Expressions

Les opérateurs sont utilisés pour effectuer des opérations sur des variables et des valeurs.

- **Opérateurs arithmétiques : +, -, *, /, %**
- **Opérateurs d'affectation : =, +=, -=, *=, /=**
- **Opérateurs de comparaison : ==, !=, >, <, >=, <=**
- **Opérateurs logiques : and, or, not**

Exemple :

python

CopyEdit

```
a = 10
b = 5


# Arithmétique
print(a + b)  # Affiche : 15


# Comparaison
if a > b:
    print("a est plus grand que b")


# Logique
if a > 5 and b < 10:
    print("Les deux conditions sont vraies")
```

2.3 Instructions Conditionnelles

Les instructions conditionnelles permettent d'exécuter des actions différentes en fonction des conditions.

Exemple :

```
num = 10


if num > 0:
    print("Le nombre est positif.")
elif num < 0:
    print("Le nombre est négatif.")
else:
    print("Le nombre est zéro.")
```

2.4 Boucles en Python

Les boucles permettent d'exécuter un bloc de code à plusieurs reprises tant qu'une condition est remplie.

Exemple d'une boucle for :

```
python

CopyEdit

for i in range(5):
    print(f"Valeur de i : {i}")
```

Exemple d'une boucle while :

```
compteur = 0
while compteur < 5:
    print(f"Compteur : {compteur}")
    compteur += 1
```

2.5 Fonctions

Les fonctions sont des blocs de code réutilisables définis avec le mot-clé def.

Exemple :

```
def saluer(nom):
    return f"Bonjour, {nom}!"


print(saluer("Alice"))  # Affiche : Bonjour, Alice!
```