Les **variables** et l'**assignation** sont des concepts fondamentaux en programmation, et en Python, elles jouent un rôle crucial dans la gestion des données. Comprendre comment elles fonctionnent est essentiel pour écrire des programmes efficaces.

**1. Les Variables en Python**

Une **variable** est un espace de stockage où tu peux enregistrer une valeur, qu'il s'agisse d'un nombre, d'un texte, ou même d'une structure de données complexe. Une variable **porte un nom** et tu peux l'utiliser pour accéder à la valeur qui lui est assignée.

En Python, contrairement à d’autres langages comme C ou Java, tu n'as pas besoin de spécifier le type de données de la variable avant de l'utiliser. Python déduit automatiquement le type de données basé sur la valeur qui lui est assignée.

**Exemple simple :**

python

CopyEdit

nom = "Alice" # La variable nom contient une chaîne de caractères

age = 25 # La variable age contient un entier

* nom est une variable qui contient une chaîne de caractères ("Alice").
* age est une variable qui contient un entier (25).

**2. L'Assignation de Valeurs aux Variables**

L'**assignation** consiste à donner une valeur à une variable en utilisant l'opérateur =. Cela signifie que tu attribues une valeur à une variable, et cette variable « pointe » vers cette valeur.

L'assignation en Python est effectuée de manière simple avec le symbole =, qui est l'opérateur d'assignation.

**Syntaxe de l'assignation :**

python

CopyEdit

variable = valeur

Par exemple :

python

CopyEdit

x = 10 # La variable x contient la valeur 10

y = 5 # La variable y contient la valeur 5

**3. La Déclaration de Variables**

En Python, tu n'as pas besoin de déclarer explicitement une variable avant de l'utiliser. Tu peux créer et assigner une valeur à une variable en une seule étape, comme montré ci-dessus.

python

CopyEdit

x = 15 # Ici, x est créé et a la valeur 15

Contrairement à certains langages comme C, tu n’as pas besoin de dire à Python quel type de données x est au moment de l'assignation. Python détecte automatiquement qu’il s’agit d’un entier (int) dans ce cas.

**4. Réassignation de Variables**

Une variable peut être **réassignée** plusieurs fois au cours du programme, ce qui signifie que tu peux changer la valeur qu'elle contient.

**Exemple de réassignation :**

python

CopyEdit

x = 10 # x vaut 10

x = 20 # x est réassigné à 20

x = 15 # x est à nouveau réassigné à 15

La variable x a successivement pris les valeurs 10, 20, et 15 à chaque étape.

**5. Types de Variables**

Lorsque tu assignes une valeur à une variable, Python déduit son type. Voici quelques exemples courants de types :

* **Entier (int)** : Un nombre entier, sans décimales.

python

CopyEdit

age = 30

* **Flottant (float)** : Un nombre réel, avec une décimale.

python

CopyEdit

prix = 99.99

* **Chaîne de caractères (str)** : Une séquence de caractères.

python

CopyEdit

nom = "Alice"

* **Booléen (bool)** : Représente True (vrai) ou False (faux).

python

CopyEdit

est\_marie = True

* **Liste (list)** : Une collection ordonnée et modifiable de valeurs.

python

CopyEdit

fruits = ["pomme", "banane", "cerise"]

* **Dictionnaire (dict)** : Une collection de paires clé-valeur.

python

CopyEdit

personne = {"nom": "Alice", "age": 30}

* **Ensemble (set)** : Une collection non ordonnée d’éléments uniques.

python

CopyEdit

nombres = {1, 2, 3, 4, 5}

**6. Les Variables et l’Utilisation de l’Opérateur =**

Il est important de noter que l'opérateur = en Python **n'est pas** une comparaison. En Python, = est utilisé pour **l'assignation**, tandis que pour la comparaison, on utilise ==.

**Exemple :**

python

CopyEdit

x = 5 # Assignation de la valeur 5 à x

y = 10 # Assignation de la valeur 10 à y

if x == y: # Comparaison pour voir si x est égal à y

print("x et y sont égaux.")

else:

print("x et y ne sont pas égaux.")

Dans cet exemple, == compare les valeurs de x et y, tandis que = est utilisé pour assigner une valeur à une variable.

**7. Les Variables Locales et Globales**

En Python, il existe deux types de variables selon leur **portée** :

* **Les variables locales** sont définies à l'intérieur d'une fonction ou d'un bloc de code, et ne peuvent être utilisées qu'à l'intérieur de ce bloc.
* **Les variables globales** sont définies à l'extérieur de toutes les fonctions et sont accessibles dans tout le programme.

**Exemple de variables locales :**

python

CopyEdit

def ma\_fonction():

x = 5 # Variable locale

print(x)

ma\_fonction()

**Exemple de variables globales :**

python

CopyEdit

x = 10 # Variable globale

def ma\_fonction():

print(x) # La variable globale peut être utilisée dans la fonction

ma\_fonction()

**8. Les Variables Immuables et Mutables**

Les types de données en Python se divisent aussi en deux grandes catégories : **mutables** et **immuables**.

* **Immuables** : Les variables de type int, float, str, tuple, etc. sont immuables, ce qui signifie que tu ne peux pas changer directement leur valeur une fois assignée. Par exemple, une chaîne de caractères ne peut pas être modifiée en place.

python

CopyEdit

x = "Hello"

# x[0] = "h" # Cela produira une erreur car les chaînes sont immuables

* **Mutables** : Les variables de type list, dict, set sont mutables, ce qui signifie que tu peux changer leurs éléments après leur création.

python

CopyEdit

liste = [1, 2, 3]

liste[0] = 10 # On modifie la liste en place

print(liste) # Affiche : [10, 2, 3]

**9. Renommer ou Réassigner des Variables**

Une variable peut être **renommée** ou **réassignée** à une nouvelle valeur à tout moment dans le programme. Cependant, il est important de ne pas confondre cela avec des opérateurs comme == (comparaison) ou d'autres expressions complexes.