

Les Variables et Opérateurs en Python

🔦 Définition et Création

Une **variable** est un espace de stockage nommé qui contient une valeur. En Python, vous n'avez pas besoin de déclarer le type de la variable ; l'interpréteur le détermine automatiquement en fonction de la valeur que vous lui attribuez.

🏠 Types de Données de Base

Python dispose de plusieurs types de données fondamentaux :

- **Nombres entiers (int)** : Pour les nombres sans décimales. Exemple : `age = 25`.
- **Nombres flottants (float)** : Pour les nombres à virgule. Exemple : `taille = 1.75`.
- **Chaînes de caractères (str)** : Pour le texte. Elles sont délimitées par des guillemets simples ou doubles. Exemple : `nom = "Alice"`.
- **Booléens (bool)** : Représentent une valeur de vérité, qui peut être soit **True** (vrai) soit **False** (faux). Exemple : `est_majeur = True`.

📝 Règles pour nommer les variables

Les noms de variables doivent suivre des règles précises pour être valides en Python. S'ils ne les respectent pas, cela entraînera une erreur de syntaxe.

- Un nom de variable doit commencer par une lettre (a-z, A-Z) ou un trait de soulignement (`_`). Il ne peut pas commencer par un chiffre.
- Un nom de variable ne peut contenir que des caractères alphanumériques (a-z, A-Z, 0-9) et des traits de soulignement (`_`).
- Les noms de variables sont sensibles à la casse (`age`, `Age` et `AGE` sont des variables différentes).
- Un nom de variable ne peut pas être un mot-clé Python réservé (comme `if`, `for`, `class`, etc.).

```
root@python:~$
```

```
# Exemples de noms de variables valides
mon_age = 30
_nom_utilisateur = "admin"
vitesse_1 = 120
```

🔧 Les Opérateurs Fondamentaux

2. Les Opérateurs Arithmétiques ➕

Ils servent à effectuer des opérations mathématiques.

Opérateur	Description	Exemple
+	Addition	5 + 3 donne 8
-	Soustraction	10 - 4 donne 6
*	Multiplication	2 * 5 donne 10
/	Division	10 / 3 donne 3.333 ...
//	Division entière	10 // 3 donne 3
%	Modulo (reste de la division)	10 % 3 donne 1
**	Puissance	2 ** 3 donne 8

1. L'Opérateur d'Affectation (=)

Cet opérateur est utilisé pour attribuer une valeur à une variable. Les opérateurs combinés permettent de simplifier les opérations d'affectation en les fusionnant avec un opérateur arithmétique.

```
root@python:~$
```

```
# Affectation d'une valeur à une variable
a = 10
# Opérateur d'affectation combiné
a += 5 # équivaut à a = a + 5. 'a' vaut maintenant 15.
a -= 2 # équivaut à a = a - 2. 'a' vaut maintenant 13.
a *= 3 # équivaut à a = a * 3. 'a' vaut maintenant 39.
a /= 2 # équivaut à a = a / 2. 'a' vaut maintenant 19.5.
```

4. Les Opérateurs Logiques 🧠

Ils permettent de combiner des conditions booléennes.

- **and** : L'expression est **True** si **toutes** les conditions sont **True**.
- **or** : L'expression est **True** si au moins **une** des conditions est **True**.
- **not** : Inverse le résultat de la condition.

```
root@python:~$
```

```
x = 5
y = 10
print(x > 0 and y > 0) # Affiche True
print(x > 10 or y > 5) # Affiche True
print(not (x == 5))    # Affiche False
```

3. Les Opérateurs de Comparaison ⚖️

Ils comparent deux valeurs et renvoient toujours un résultat booléen (**True** ou **False**).

Opérateur	Description	Exemple
=	Égal à	5 == 5 est True
≠	Différent de	5 != 6 est True
>	Strictement supérieur à	5 > 3 est True
<	Strictement inférieur à	5 < 3 est False
≥	Supérieur ou égal à	5 ≥ 5 est True
≤	Inférieur ou égal à	5 ≤ 3 est False

THE
PYTHONIC
WAY

HACK
AGAINST
MACHISM



*FOLLOW
THE
PYTHONIC
WAY*

*HACK
AGAINST
MACHISM*