

# Variables et constantes

## Principes

Les variables et constantes sont des [éléments fondamentaux en programmation](#). Elles permettent de [stocker des données](#) et d'[effectuer des opérations](#) sur ces données. Comprendre leurs principes et caractéristiques est [essentiel](#) pour écrire un code efficace.

**Variable** : Une variable est un espace de stockage qui peut contenir différentes valeurs au cours de l'exécution d'un programme.

**Constante** : Une constante est un espace de stockage dont la valeur ne peut pas changer une fois définie.

## Nom

Les noms des variables et constantes doivent être [descriptifs](#) pour refléter leur contenu. Les conventions de nommage incluent :

- Utiliser des lettres, chiffres et underscores ( \_ ).
- Ne pas commencer par un chiffre.
- Éviter les mots réservés du langage.

## Type

Les types de variables définissent la nature des données qu'elles peuvent contenir. Les types courants incluent :

- **Entier** : Pour les nombres entiers (ex. : int).
- **Flottant** : Pour les nombres à virgule (ex. : float).
- **Chaîne** : Pour les textes (ex. : string).
- **Booléen** : Pour les valeurs vraies ou fausses (ex. : bool).
- **Date** : Pour les dates (ex. date).

## Taille

La taille d'une variable dépend de son type et de l'implémentation du langage de programmation.

Par exemple, un int peut occuper 4 octets, tandis qu'un float peut en occuper 8.

## Opérations possibles

Les opérations que l'on peut effectuer sur les variables dépendent de leur type :

- **Arithmétiques** : Addition, soustraction, multiplication, division pour les types numériques.
- **Logiques** : ET, OU, NON pour les booléens.
- **Concaténation** : Joindre des chaînes de caractères.

## Déclaration

La déclaration consiste à informer le programme de **l'existence d'une variable ou d'une constante**, souvent en spécifiant son type. Ce qui permet de choisir une forme de codification avec un "=".

Exemple en Python :

```
age = 30  
  
PI = 3.14
```

## Initialisation et Affectation

- **Initialisation** : C'est le processus d'attribution de sa première valeur à une variable lors de sa déclaration.
- **Affectation** : C'est le processus de changement de la valeur d'une variable après sa déclaration.

Exemple :

```
compteur = 0 # Initialisation  
  
compteur += 1 # Affectation
```

## Cas particuliers

Les cas particuliers incluent les **variables nulles** (qui ne contiennent pas de valeur) et les **constantes immuables** (qui ne peuvent pas être modifiées après leur création).

Exemple :

```
nom = null # Variable nulle
```

## Incrémentatation

L'incrémentatation est une opération qui **augmente la valeur d'une variable entière**, souvent utilisée

avec les compteurs.

Exemples :

```
* compteur += 1 # Incrémenter le compteur de 1
* compteur++;
* compteur = compteur + 1;
```

Ces exemples donnent le même résultat où le chiffre augmentera à chaque fois de 1.

## Concaténation

La concaténation est l'opération qui consiste à [joindre des chaînes de caractères](#) pour former une [nouvelle chaîne](#).

Exemple :

```
nom_complet = "Alice" + " " + "Dupont" # Concaténation de chaînes
= Alice Dupont
```

From:

<https://wiki.siochaptalqper.fr/> - Wiki SIO Chaptal

Permanent link:

<https://wiki.siochaptalqper.fr/doku.php?id=bloc1:prog:variables-constantes&rev=1732269196>

Last update: **2024/11/22 10:53**

