C - Les noms de variables

En C, chaque variable doit porter un nom clair et précis qui permet de comprendre facilement ce qu'elle représente. Le choix des noms est essentiel pour écrire un code lisible, cohérent et professionnel. Respecter les règles de syntaxe et adopter de bonnes conventions de nommage facilite la maintenance du programme et la collaboration entre développeurs.

5TTR 6TTR Découverte

Objectifs

À la fin de ce chapitre, tu sauras :

- Comprendre les **règles de nommage** des variables en langage C.
- Reconnaître les **erreurs à éviter** dans les noms de variables.
- Appliquer les **bonnes conventions d'écriture** pour rendre ton code clair et lisible.

Rappel : qu'est-ce qu'une variable ?

Une **variable** est un espace de mémoire qui contient une valeur d'un type donné. Elle a :

- un **nom** (identifiant),
- un **type** (ex : `int`, `float`, `char`),
- et une **valeur**.

Règles de base pour nommer une variable

Le langage C impose certaines règles strictes :

✅ **Un nom de variable doit :**

- Commencer par **une lettre** (majuscule ou minuscule) ou un **underscore** `_` ;

- Contenir uniquement des **lettres, chiffres ou** `_` ;
- Ne pas dépasser 31 caractères (pour rester lisible) ;
- Être **unique** dans un même bloc de code.

❌ **Un nom de variable ne peut pas :**

- Commencer par un chiffre ;
- Contenir des espaces ou des caractères spéciaux (`-`, `?`, `!`, etc.) ;
- Utiliser un **mot réservé du C** (ex : `int`, `return`, `if`, `while`, …).

Exemples :

```c
int age;          // ✅ correct
float moyenne_1;  // ✅ correct
int 2notes;       // ❌ incorrect (commence par un chiffre)
char prénom;      // ❌ incorrect (accent interdit)
int if;           // ❌ incorrect (mot réservé)
```

> Le C est ***case-sensitive*** (sensible à la casse): il fait la **différence entre majuscules et minuscules**. C'est le cas aussi pour les noms de variables. => `prix` et `Prix` sont deux noms de variables différents.

Sens des noms

Un bon nom de variable doit **décrire ce qu'elle contient**. C'est essentiel pour que ton code soit **compréhensible** pour toi et pour les autres.

Exemples :

```c
float temperature;    // clair
float t;              // trop court : on ne sait pas ce que ça signifie
```

Conventions d'écriture

Même si le compilateur ne les impose pas, certaines **conventions** rendent ton code plus lisible.

🟢 Pour les variables et fonctions :

➡️ **utilise le style "camelCase" ou "snake_case"**

- `camelCase` : la première lettre minuscule, les mots suivants en majuscule. → `noteMoyenne`, `nombreEtudiants`

© 2025 Mr. Lambrechts

- `snake_case` : les mots séparés par un underscore `_`. → `note_moyenne`, `nombre_etudiants`

Les deux styles sont corrects, mais il est important d'en choisir **un seul** et de s'y tenir dans tout ton projet => l'important c'est la **consistance/cohérence/uniformité/régularité**.

🟡 Pour les constantes :

➡️ écris-les en **majuscules** avec des underscores :

```c
#define TAUX_TVA 0.21
const int VITESSE_LUMIERE = 299792458;
```

🔵 Pour les noms de fonctions :

➡️ commence par un **verbe**, pour indiquer une action :

```c
calculerMoyenne();
afficherResultat();
```

Bonnes pratiques générales

- Choisis des noms **clairs et précis** : `note_finale` est mieux que `n` ou `x`.
- Reste **cohérent** : n'alterne pas entre `camelCase` et `snake_case`.
- N'utilise pas de majuscules au hasard : `Age` et `age` sont **deux variables différentes** en C.
- Évite les abréviations obscures : `nb_eleve` vaut mieux que `nbe` ou `ne`.
- Sois attentif à l'**orthographe** : `moyenne`, pas `moyene`.

Exemple de bonnes pratiques

```c
#include <stdio.h>

int main() {
    int nombre_etudiants = 25;
    float moyenne_generale = 13.8;
    char premiere_lettre = 'M';

    printf("Il y a %d étudiants.\n", nombre_etudiants);
    printf("La moyenne générale est de %.1f.\n", moyenne_generale);
    printf("La première lettre est %c.\n", premiere_lettre);
```

© 2025 Mr. Lambrechts

```
11
12      return 0;
13  }
```

Ce code est **facile à lire** et les noms sont **cohérents** et **significatifs**.

À retenir

- Un nom de variable doit **commencer par une lettre**, ne pas contenir de caractères spéciaux ni de mots réservés.
- Un nom **clair** facilite la lecture et la compréhension du code.
- Choisis une **convention d'écriture** (`camelCase` ou `snake_case`) et respecte-la partout.
- En C, la casse est **sensible** (`note` ≠ `Note`).

Souhaites-tu que je crée une **fiche d'exercices (10 activités)** où les élèves doivent corriger ou proposer de bons noms de variables selon ces règles ?

© 2025 Mr. Lambrechts