

# INTRODUCTION AU LANGAGE C

Le langage C est un langage de programmation procédural. Il a été initialement développé par Dennis Ritchie entre 1969 et 1973. Il a été principalement développé en tant que langage de programmation système pour écrire un système d'exploitation.

Beaucoup de langages ultérieurs ont emprunté la syntaxe / les fonctionnalités directement ou indirectement au langage C. Comme la syntaxe de Java, PHP, JavaScript et de nombreux autres langages sont principalement basés sur le langage C. C++ est presque un sur-ensemble du langage C (peu de programmes peuvent compiler en C, mais pas en C++).

## Structure d'un programme C

Par structure, on entend que tout programme ne peut être écrit que dans cette structure. L'écriture d'un programme C dans n'importe quelle autre structure conduira donc à une erreur de compilation.

La structure d'un programme C est la suivante:

## Inclusion de fichiers d'en-tête

Le premier et le plus important composant est l'inclusion des fichiers d'en-tête dans un programme C.

Un fichier d'en-tête est un fichier d'extension **.h** qui contient des déclarations de fonction C et des définitions de macro à partager entre plusieurs fichiers source. Quelques fichiers d'en-tête C:

- **stdio.h** : Définit les fonctions principales d'entrée et de sortie.
- **stdlib.h** : Définit les fonctions de conversion numérique, allocation de mémoire.
- **string.h** : Définit les fonctions de gestion de chaîne.
- **math.h** : Définit les fonctions mathématiques communes.
- **stdint.h** : Définit les types entiers de largeur exacte.
- **stddef.h** : Définit plusieurs types et macros utiles.

## Syntaxe pour inclure un fichier d'en-tête dans C

```
1 #include <(nom_fichier_entete).h>
```

## Déclaration du programme principal

La prochaine partie d'un programme C consiste à déclarer la fonction main ().

## La syntaxe pour déclarer la fonction principale

```
1 int main()  
2 {  
3     // Corps de la fonction  
}
```

## Déclaration de variables

La partie suivante de tout programme C est la déclaration de variable. Il fait référence aux variables à utiliser dans la fonction.

**Remarque :** dans le programme C, aucune variable ne peut être utilisée sans avoir été déclarée. Toujours dans un programme C, les variables doivent être déclarées avant toute opération dans la fonction.

**Exemple :**

```

1  int main()
2  {
3      int a;
4      // manipulations
5      ...
6  }
```

## Corps

le corps d'une fonction dans le programme C fait référence aux opérations effectuées dans les fonctions.

**Exemple :**

```

1  int main()
2  {
3      int a=5;
4      // corps
5      a=a+5;
6      print("%d", a);
7      ...
8  }
```

## Premier programme C

```

1  #include < stdio.h >
2  int main(void)
3  {
4      print("Hello world");
5      return 0;
6  }
```

**Ligne 1 [ #include < stdio.h> ] :**

Dans un programme C, toutes les lignes qui commencent par un # sont traitées par le préprocesseur qui est un programme invoqué par le compilateur. Dans un terme très basique, le préprocesseur prend un programme C et produit un autre

programme C. Le programme n'a pas de lignes commençant par #, toutes ces lignes sont traitées par le préprocesseur.

Dans l'exemple ci-dessus, le préprocesseur copie le code pré-traité de **stdio.h** notre fichier. les fichiers **.h** sont appelés fichiers d'en-tête dans C. Ces fichiers d'en-tête contiennent généralement une déclaration de fonctions.

On a besoin de **stdio.h** pour la fonction **printf ()** utilisée dans le programme.

**Ligne 2 [ `int main(void) :`**

Il doit y avoir un point de départ à partir duquel l'exécution du programme C compilé commence. En C, l'exécution commence typiquement par la première ligne de **main()**.

Le vide entre parenthèses indique que le **main** ne prend aucun paramètre. **main()** peut être écrit pour prendre des paramètres également.

L' **int** écrit avant **main** indique le type de retour de **main()**. La valeur retournée par **main** indique l'état de fin du programme.

**Ligne 3 et 6 [ { et } ] :**

En langage C, une paire de crochets définit une portée et est principalement utilisée dans les fonctions et les déclarations de contrôle comme **if** , **else**, les boucles.

Toutes les fonctions doivent commencer et se terminer avec des crochets.

**Ligne 4 [ `printf("Hello world"); :`**

**printf()** est une fonction de bibliothèque standard pour afficher quelque chose sur une sortie standard. Le point-virgule (;) à la fin de **printf** indique terminaison de ligne. En C, le point-virgule est toujours utilisé pour indiquer la fin de la déclaration.

**Ligne 5 [ `return 0; :`**

L'instruction **return** renvoie la valeur de **main ()**. La valeur renvoyée peut être utilisée par le système d'exploitation pour connaître l'état de fin de votre programme. La valeur 0 signifie généralement que la terminaison est réussie.