

Introduction au Langage C

0

Le langage C est un langage de programmation procédural développé dans les années 1970 par Dennis Ritchie aux Bell Labs. Il est largement utilisé pour développer des systèmes d'exploitation, des logiciels embarqués, et des applications nécessitant une gestion fine des ressources.

Structure de Base

Un programme en C est composé de fonctions, dont la principale est main(). Voici un exemple de structure de base :

```
#include <stdio.h>
int main() {
    printf("Hello, World! \n");
    return 0;
}
```

#include <stdio.h> : Inclut la bibliothèque standard d'entrée/sortie.

int main(): Point d'entrée du programme.

0

printf("Hello, World!\n"); : Affiche du texte à l'écran.

return 0; : Indique que le programme s'est terminé avec succès.

Types de Données

Les types de données de base en C incluent :

int: Entier.

0

float : Nombre flottant (réel).

double : Nombre flottant double précision.

char: Caractère.

void : Type vide, utilisé pour les fonctions qui ne

retournent aucune valeur.

Variables et Constantes

Variables:

Espaces mémoire nommés pour stocker des données.

```
int age = 25;
float pi = 3.14;
char lettre = 'A';
```

Constantes:

Valeurs fixes qui ne peuvent pas être modifiées.

```
const int JOURS_PAR_SEMAINE = 7;
```

Opérateurs

Arithmétiques: +, -, *, /, %.

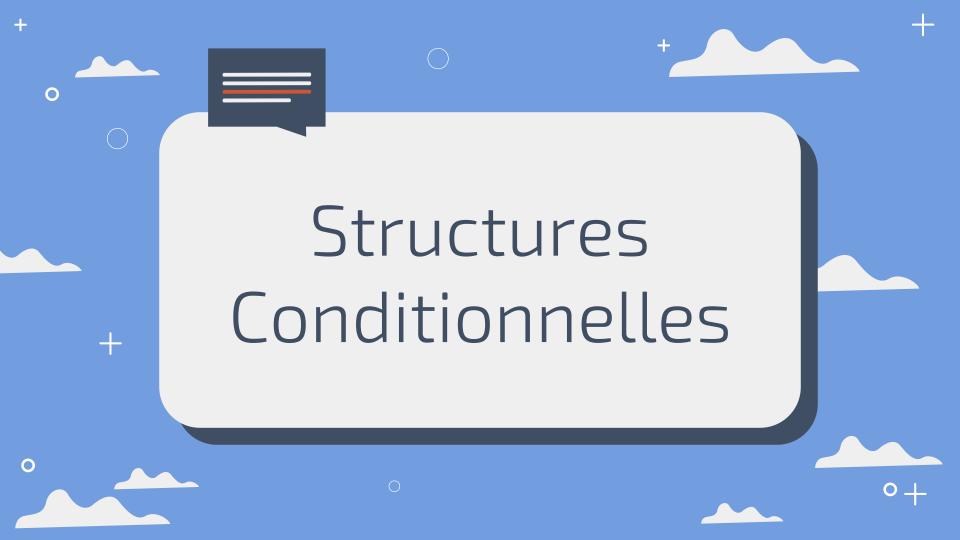
Comparaison: ==, !=, >, <, >=, <=.

Logiques: && (ET), || (OU), ! (NON).

Structures Conditionnelles

Les structures de contrôle en C permettent de gérer le flux d'exécution du programme en fonction de conditions ou de répéter des blocs de code.

Voici une explication détaillée des structures de contrôle en C :



° if

La structure if exécute un bloc de code si la condition est vraie.

```
Exemple:
```

```
int age = 18;
  if (age >= 18) {
    printf("Vous êtes majeur.\n");
  }
```

if-else

La structure if-else permet d'exécuter un bloc de code si la condition est vraie, et un autre bloc si elle est fausse.

```
int age = 16;
if (age >= 18) {
    printf("Vous êtes majeur.\n");
} else {
    printf("Vous êtes mineur.\n");
}
```

else-if

La structure else-if permet de tester plusieurs conditions successivement.

```
int note = 75;
if (note >= 90) {
    printf("Très bien !\n");
} else if (note >= 70) {
    printf("Bien.\n");
} else {
    printf("À améliorer.\n");
}
```

switch-case

La structure switch-case permet de tester la valeur d'une variable et d'exécuter un bloc de code correspondant à cette valeur.

```
int jour = 3;
switch (jour) {
   case 1: printf("Lundi\n"); break;
   case 2: printf("Mardi\n"); break;
   case 3: printf("Mercredi\n"); break;
   default: printf("Jour invalide.\n");
}
```



while

La boucle while répète un bloc de code tant qu'une condition est vraie.

```
int i = 0;
while (i < 5) {
    printf("i = %d\n", i);
    i++;
}</pre>
```

do-while

La boucle do-while est similaire à while, mais elle exécute le bloc de code au moins une fois, même si la condition est fausse.

```
int i = 0;
do {
    printf("i = %d\n", i);
    i++;
    } while (i < 5);</pre>
```

for

La boucle for est utilisée pour répéter un bloc de code un nombre spécifié de fois.

```
int i;
for (i = 0 ; i < 5 ; i++) {
    printf(" valeur de i = %d\n", i);
}</pre>
```



Mots-clés de Contrôle

break : Sort immédiatement d'une boucle ou d'un switch.

continue : Passe à l'itération suivante d'une boucle.

return: Termine l'exécution d'une fonction et retourne une valeur.

```
for (int i = 0; i < 10; i++) {
   if (i == 5) { break; // Sort de la boucle quand i == 5 }
   if (i % 2 == 0) { continue; // Passe à l'itération suivante si i est pair }
   printf("i = %d\n", i);
}</pre>
```