

Fundamentos de Algoritmos

Constantes, Variáveis e Tipos de Dados

DEFINIÇÃO E TIPOS DE VARIÁVEIS

Uma variável é um **local na memória** que **armazena um valor** que **pode ser modificado** durante a execução do programa. Existem diversos tipos de variáveis, como:

- **Inteiro**: armazena números inteiros (sem casas decimais).
- **Real**: armazena números reais (com casas decimais).
- **Caractere**: armazena um único caractere.
- **Ponteiro**: armazena o endereço de outro local na memória.

As variáveis são declaradas com o **tipo de dado** seguido do **nome da variável**, podendo ou não ser **inicializada** com um valor no momento da **declaração**.



```
1 int numero = 10;           // variável do tipo inteiro inicializada com o valor 10
2 float temperatura = 25.5;  // variável do tipo real inicializada com o valor 25.5
3 char letra = 'a';          // variável do tipo caractere inicializada com o valor 'a'
```

DEFINIÇÃO E TIPOS DE CONSTANTES EM C:

Uma **constante** é um **valor fixo** que **não pode ser alterado** durante a execução do programa. Existem dois tipos principais de constantes:

- **Constante numérica:** armazenam valores numéricos, como inteiros (1, 2, 3) ou reais (3.14, 2.718).
- **Constante de caractere:** armazenam um único caractere, como 'a', 'B', '1'.

DECLARAÇÃO E USO DE CONSTANTES

Constantes são declaradas com a palavra-chave **const**
É recomendável usar **nomes em letras maiúsculas** para constantes.

Exemplo de declaração de constante:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      const int NUM_MAX = 100; // constante numérica
6      const char LETRA = 'A';  // constante de caractere
7
8      return 0;
9  }
```

ESCOPO DE VARIÁVEIS



Existem dois tipos de escopo:

- **Local**: a variável só é acessível **dentro do bloco** onde foi declarada.
- **Global**: a variável é acessível **em todo o programa**.

O escopo de uma variável define a região do programa onde ela é visível e acessível.



EXEMPLO DE ESCOPO

```
1  #include <stdio.h>
2
3  // Variável global
4  int global = 10;
5
6  int main() {
7      // Variável local ao bloco main
8      int local_main = 20;
9
10     printf("Variável global dentro de main: %d\n", global); // Acesso à variável global
11     printf("Variável local dentro de main: %d\n", local_main); // Acesso à variável local
12
13     {
14         // Novo bloco com outra variável local
15         int local_inner = 30;
16
17         printf("Variável global dentro do bloco interno: %d\n", global); // Acesso à variável global
18         printf("Variável local dentro do bloco interno: %d\n", local_inner); // Acesso à variável local
19         printf("Variável local de main dentro do bloco interno: %d\n", local_main); // Acesso à variável local de main
20     }
21
22     // printf("Variável local dentro do bloco interno: %d\n", local_inner); // Esta linha causaria um erro, pois local_inner não está mais no escopo
23
24     printf("Variável global fora de main: %d\n", global); // Acesso à variável global
25     // printf("Variável local de main fora de main: %d\n", local_main); // Esta linha causaria um erro, pois local_main não está mais no escopo
26     return 0;
27 }
```

TIPOS BÁSICOS DE DADOS

Cada tipo de dado possui um tamanho e um intervalo de valores válidos específicos

int

inteiro (1, 2, 3)

float

real (3.14, 2.718)

char

caractere ('a', 'B', '1')

void

indica a ausência de valor

OPERADORES DE CONVERSÃO DE TIPOS DE DADOS

Os operadores de conversão de tipos de dados permitem converter um valor de um tipo para outro.

(int)x : converte o valor de x para inteiro.
(float)y : converte o valor de y para real.
(char)z : converte o valor de z para caractere.

EXEMPLO PRÁTICO:

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float base, altura, area;
6
7      printf("Digite a base do retângulo: ");
8      scanf("%f", &base);
9      printf("Digite a altura do retângulo: ");
10     scanf("%f", &altura);
11
12     area = base * altura;
13
14     printf("A área do retângulo é: %.2f\n", area);
15
16     return 0;
17 }
```

DESAFIO

1. Modifique o programa para calcular e imprimir o perímetro do retângulo (**$\text{perímetro} = 2 * (\text{base} + \text{altura})$**)
2. Escreva um programa que leia a temperatura em Celsius e converta para Fahrenheit (**$F = (C * 9/5) + 32$**)

