

LINGUAGEM C: FUNDAMENTOS

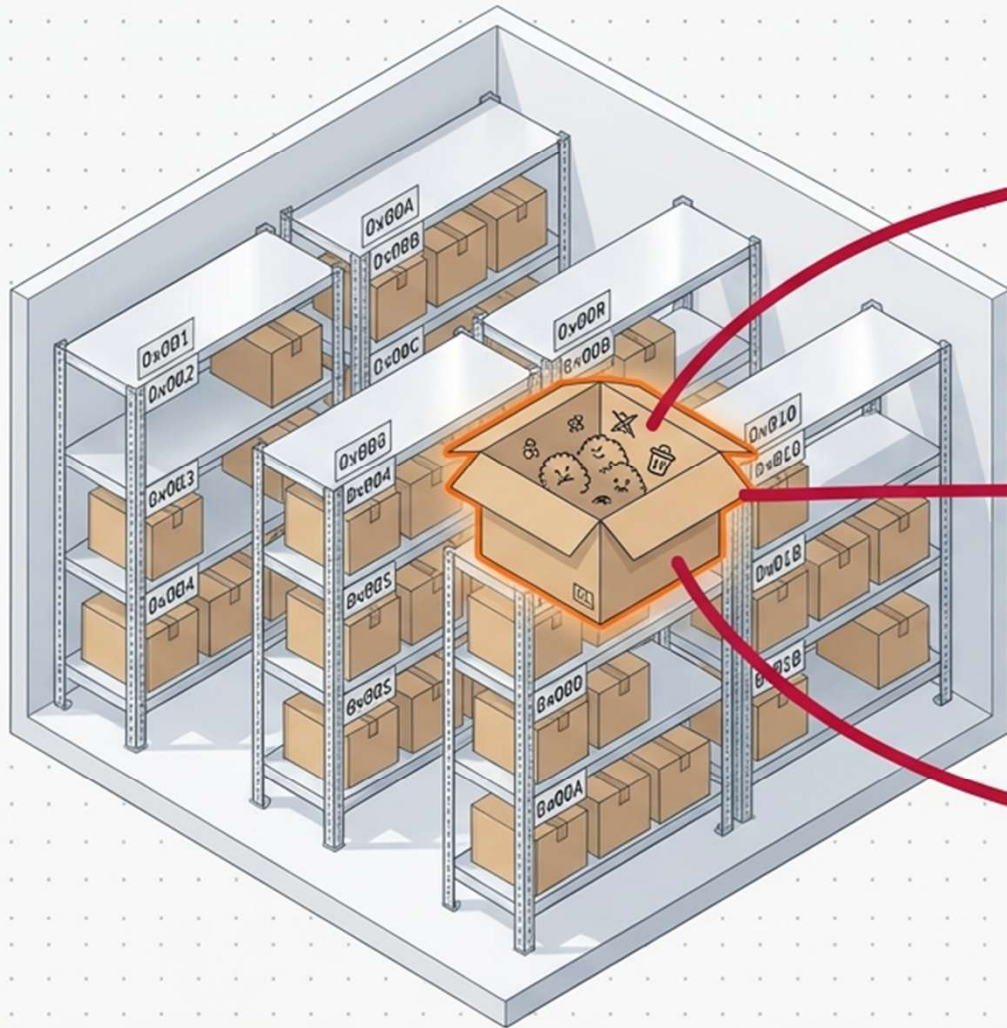
Variáveis, Entradas
Explicados de Forma Simples

Prof. Rafael Muniz |



FOCO INICIAL:
Conceitos-chave!

Variáveis e a Memória Física



A Caixa (O Espaço): Declarar uma variável não associa um valor automaticamente. Apenas reserva uma área física na memória.

`int idade;` ← Declaração apenas reserva espaço

O Rótulo (O Nome): O compilador não usa o nome digitado; ele mapeia esse nome para o endereço físico da memória.

`idade` → `0x003` ← Mapeamento Interno

A Sujeira (Lixo de Memória): Se ler uma variável recém-declarada, verá resíduos de dados antigos que ocupavam aquele espaço antes. É vital inicializar a variável!

`printf("%d", idade);` // Saída imprevisível!

Resultado: Lixo
(ex: -2849102)

Os Três Tipos Fundamentais



1,2,3

int

O que armazena: Valores numéricos inteiros (ex: 2, 10).

No printf/scanf:
Usa-se **%i** ou **%d**



3.14

float

O que armazena: Valores reais / com casas decimais.

No printf/scanf: Usa-se %f



'A'

char

O que armazena: Apenas um único caractere (letra ou símbolo).

Sintaxe: Deve estar entre aspas simples (ex: **'A'**, **'2'**).

Aviso Crítico: Se armazenar o símbolo '2', não poderá usá-lo para realizar somas! É apenas um desenho para o computador.

No printf/scanf: Usa-se %c

Constantes: Dados Protegidos

Uma constante é um valor que, após ser atribuído, não pode ser mais alterado durante a execução.

Método 1: A Diretiva



Não usa sinal de igual (=)
nem ponto e vírgula (;).

```
#define PI 3.1415  
#define LETRA 'a'
```

Método 2: O Modificador



Usa a sintaxe normal de variável,
mas trancada com a palavra const.

```
const float PI = 3.1415;  
const char LETRA = 'a';
```

0 Kit de Ferramentas: Operadores Básicos

Aritméticos (Cálculos)

+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da Divisão (Módulo)

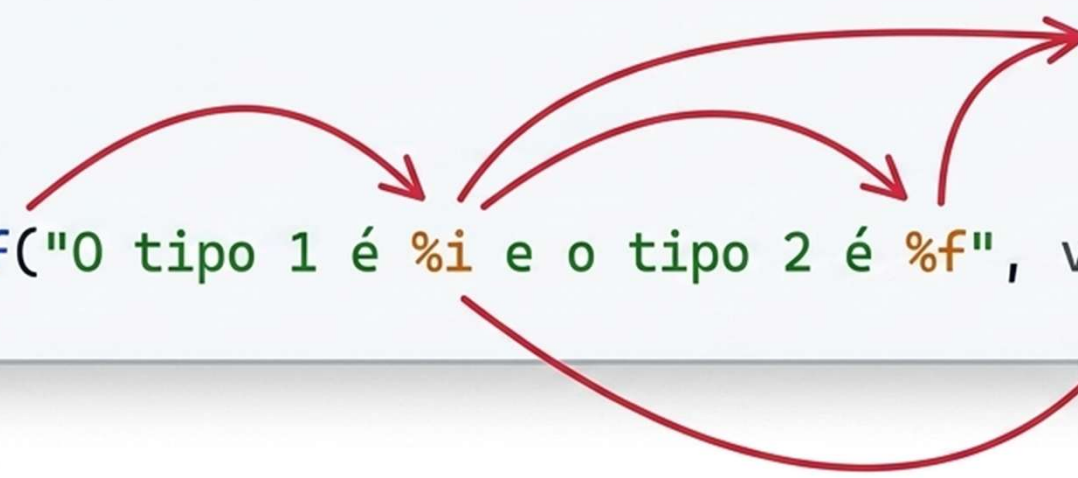
Relacionais (Comparações)

> / <	Maior que / Menor que
>= / <=	Maior ou igual / Menor ou igual
<u>==</u>	Igual a
!=	Diferente de

Atenção: são dois sinais!

Saída de Dados: O Comando printf

```
printf("O tipo 1 é %i e o tipo 2 é %f", varInt, varFloat);
```



Marcadores de Posição. A primeira letra ajuda a decorar (i = int, f = float, c = char).

As Variáveis. Ficam listadas após o fim do texto, separadas por vírgula.



Regra de Ouro

A quantidade de marcadores % dentro do texto precisa ser exatamente igual à quantidade de variáveis listadas no final.

Entrada de Dados: A Anatomia do scanf

Como ler dados digitados no teclado e guardá-los na memória.

O formato do valor que o teclado vai ler (ex: %i, %f, %c).

O 'E' Comercial. É o coração do scanf. Ele indica o endereço físico da memória da variável. O scanf precisa saber o 'CEP' físico de onde guardar o dado.

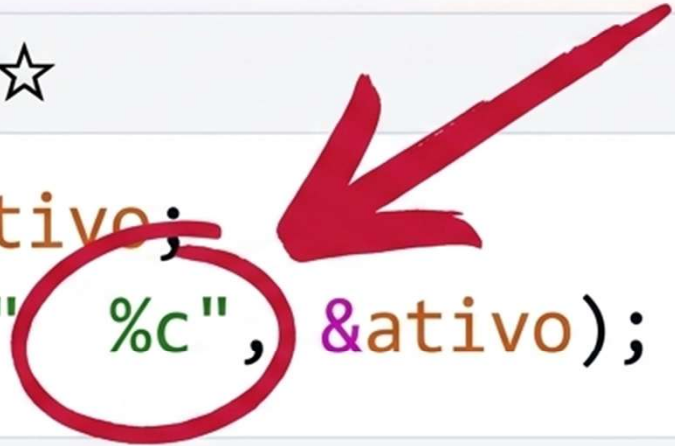
`scanf("%tipo", &variável);`

Separa os dois parâmetros obrigatórios da função.

Cuidado com o Lixo do Teclado!

Dica Pró ☆

```
char ativo;  
scanf(" %c", &ativo);
```



Observe o espaço antes do %c!

Por que esse espaço existe? Para evitar que o compilador ignore a leitura do char. Esse espaço 'engole' a tecla Enter ou espaços em branco deixados no teclado por leituras anteriores, garantindo que o programa leia o símbolo correto.

Bibliografia de Referência

- ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**. 3. ed. Pearson, 2012.
 - BUFFONI, S. **Apostila de Algoritmo Estruturado**. FIAA. 4ed, 2003.
 - FORBELLONE,
FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3ed. Pearson, 2005.
 - KERNIGHAN, B. W.; RITCHIE, D. M. **Linguagem de Programação C**. Rio de Janeiro: Campus, 1989.
-