

Fatec

Jales

Prof. José Camargo

CPS
Centro
Paula Souza



GOVERNO DO ESTADO
SÃO PAULO

Faculdade de Tecnologia "Professor José Camargo" - Fatec Jales

Curso: Análise e Desenvolvimento de Sistemas

Período: 2º Semestre

Disciplina: Linguagem de Programação

Professor: Profa. Dra. Ligia Rodrigues Prete

E-mail: ligia.prete@fatec.sp.gov.br

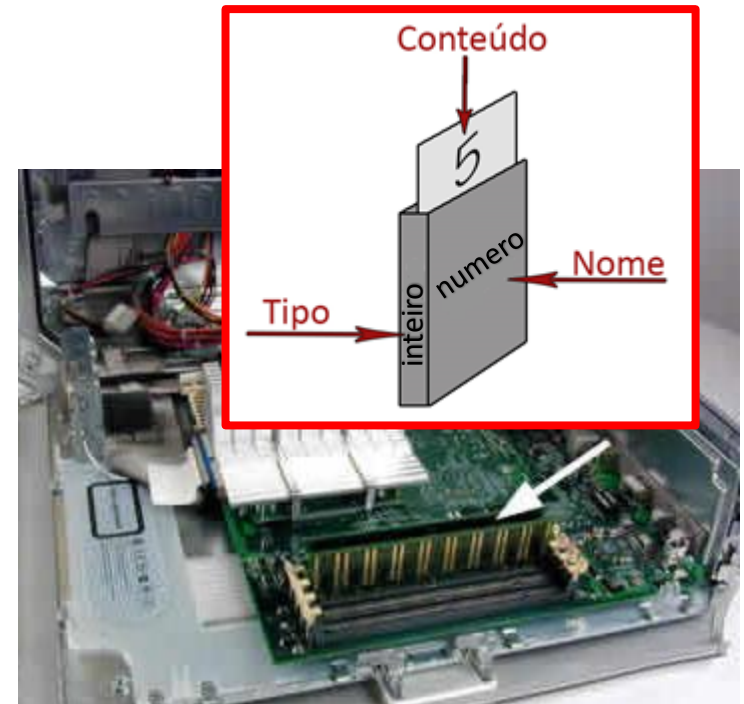
2 – Variáveis, constantes, operadores e expressões

Variável

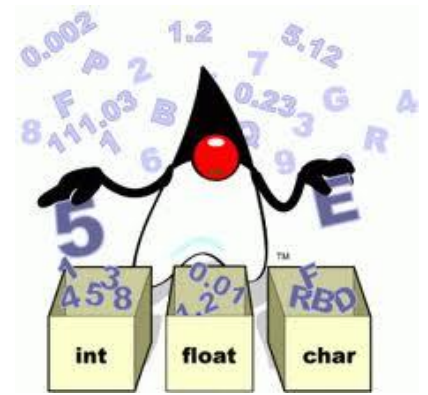
A variável representa um valor no algoritmo. Ela possui:

- Nome;
- Conteúdo;
- Tipo de dados;
- Só existe em tempo de execução (memória RAM).

Exemplo:
variável numero: inteiro



Tipos de Dados



- **Inteiro (int):** armazena somente números inteiros;
Exemplo: -4, 0, 45
- **Real (float e double):** números que possuem a parte fracionária;
Exemplo: -4.50, 78.344, 7.0
- **Caractere (char):** é todo texto colocado entre aspas;
Exemplo: “Olá mundo”, “Algoritmo”, “123”
- **Lógico (bool):** armazena dois valores
Exemplo: verdadeiro ou falso

Formatos de Leitura e Escrita

- A leitura e escrita de variáveis podem ser realizadas através das funções `scanf` (leitura) e `printf` (escrita), utilizando os **formatos** adequados para serem lidos ou escritos.

Algoritmo	Linguagem C	Formato
caractere	char	%c
cadeia	char[]	%s
inteiro	int	%d
real	float	%f
real	double	%f
logico	bool	%d

Exemplo de Variáveis e Tipos de Dados

```
1  #include <stdio.h>
2  #include <locale.h>
3  #include <stdbool.h>
4  void main()
5  {
6      setlocale(LC_ALL, "");
7      int a, b, c;
8      float x, y, z;
9      char sexo;
10     char nome[20];
11     bool situacao;
12     printf("Informe um valor inteiro para a variável a: ");
13     scanf("%d", &a);
14     printf("Informe um valor inteiro para a variável b: ");
15     scanf("%d", &b);
16     c = a + b;
17     printf("A soma de %d + %d = %d \n\n", a, b, c);
18     printf("Informe um valor real para a variável x: ");
19     scanf("%f", &x);
20     printf("Informe um valor real para a variável y: ");
21     scanf("%f", &y);
22     z = x + y;
23     printf("A soma de %0.2f + %0.2f = %0.2f \n\n", x, y, z);
24     printf("Informe o seu nome: ");
25     setbuf(stdin, NULL);
26     gets(nome);
27     printf("Informe o seu sexo: ");
28     scanf(" %c", &sexo);
29     printf("O nome informado é %s e o sexo é %c \n\n", nome, sexo);
30     situacao = true && true;
31     printf("A situação = %d \n", situacao);
32 }
```

Informe um valor inteiro para a variável a: 10
Informe um valor inteiro para a variável b: 20
A soma de 10 + 20 = 30

Informe um valor real para a variável x: 5,25
Informe um valor real para a variável y: 6,25
A soma de 5,25 + 6,25 = 11,50

Informe o seu nome: Maria
Informe o seu sexo: F
O nome informado é Maria e o sexo é F
A situação = 1

Definindo Constante (#define)

- Constantes representam dados cujos valores permanecem inalterados durante a execução do algoritmo.

- O formato geral é:



Nome da constante com letras
em maiúsculo

```
#define IDENTIFICADOR valor
```

- Se colocarmos estas linhas no header, o que vai acontecer é o seguinte: o pré-processador irá verificar o código fonte e sempre que encontrar a diretiva #define irá substituir cada ocorrência do identificador no código fonte pelo valor definido.

Exemplo de Constante (#define)

```
1  #include <stdio.h>
2  #define PI 3.14159
3
4  void main()
5  {
6      double c, r = 5.0;
7
8      c = 2 * PI * r; // utiliza-se o valor de Pi.
9      printf("O comprimento da circunferencia = %f \n", c);
10 }
11
```

```
O comprimento da circunferencia = 31.415900
```

Declarando Constante (const)

- Nós podemos transformar uma variável numa constante do gênero:

```
const double pi = 3.14159;
```

- Com o prefixo "**const**", dizemos que a variável não poderá alterar o seu valor.

Exemplo de Constante (const)

```
1  #include <stdio.h>
2
3  void main()
4  {
5      const double pi = 3.14159;
6      double c, r = 5.0;
7
8      c = 2 * pi * r; // utiliza-se o valor de Pi.
9      printf("O comprimento da circunferencia = %f \n", c);
10 }
11
```

```
O comprimento da circunferencia = 31.415900
```

Operadores Aritméticos

Um operador é um símbolo usado para criar expressões, a partir de constantes e variáveis. Por exemplo, usando o operador aritmético +, a variável x e a constante 2, podemos criar a expressão $x+2$

Operação	Algoritmo	Linguagem C (int)	Linguagem C (float ou double)
Soma	+	+	+
Subtração	-	-	-
Multiplicação	*	*	*
Divisão Inteira	DIV ou \	/	não se aplica
Divisão Real	/	não se aplica	/
Resto da Divisão Inteira	MOD ou %	%	não se aplica
Exponenciação	EXP ou ^	não se aplica	POW()
Raiz quadrada	RAIZQ	não se aplica	SQRT()

Operadores Relacionais

Descrição	Algoritmo	Linguagem C
igual	=	==
diferente de	<>	!=
menor	<	<
menor ou igual	<=	<=
maior	>	>
maior ou igual	>=	>=

Operadores Lógicos

Operador	Algoritmo	Linguagem C	Descrição
e	e	&&	Operador que resulta VERDADEIRO somente se seus dois operandos lógicos forem verdadeiros.
ou	ou		Operador que resulta VERDADEIRO quando um dos seus operandos lógicos for verdadeiro
não	nao	!	Operador de negação

Tabela Verdade

- Sejam A e B duas variáveis lógicas, que assumem valores verdadeiro (V) - falso (F).

A	B	A e B
V	V	V
V	F	F
F	V	F
F	F	F

A	não A
V	F
F	V

A	B	A ou B
V	V	V
V	F	V
F	V	V
F	F	F

Exemplo de Operadores

```
1  #include <stdio.h>
2  #include <stdbool.h>
3
4  void main()
5  {
6      int n1, n2;
7
8      printf("Digite um numero: ");
9      scanf("%d", &n1);
10     printf("Digite outro numero: ");
11     scanf("%d", &n2);
12     printf("%d - %d = %d \n", n1, n2, n1 - n2);
13     printf("%d * %d = %d \n", n1, n2, n1 * n2);
14     printf("%d >= %d = %d \n", n1, n2, n1 >= n2);
15     printf("%d == %d = %d \n", n1, n2, n1 == n2);
16     printf("%d != %d = %d \n", n1, n2, n1 != n2);
17     printf("F e V = %d \n", false && true);
18     printf("F ou V = %d \n", false || true);
19     printf("nao V = %d \n", !true);
20     printf("nao F = %d \n", !false);
21 }
22
```

```
Digite um numero: 5
Digite outro numero: 2
5 - 2 = 3
5 * 2 = 10
5 >= 2 = 1
5 == 2 = 0
5 != 2 = 1
F e V = 0
F ou V = 1
nao V = 0
nao F = 1
```