

Aula 2: Tipos Básicos e Variáveis

Professor(a): João Eduardo Montandon (103)

Virgínia Fernandes Mota (106)

[jemaf.github.io](https://github.com/jemaf)

<http://www.dcc.ufmg.br/~virginiaferm>

INTRODUÇÃO A PROGRAMAÇÃO - SETOR DE INFORMÁTICA



- Para cada comando serão apresentadas a sua:
 - **sintaxe**: formato geral do comando que deve ser aceita e respeitada como padrão.
 - **semântica**: o significado da ação realizada pelo comando, em tempo de execução.
- No texto (estático) de um programa (ou algoritmo), um valor pode ser representado na forma de constante ou de variável.
- A **constante** é representada em um programa diretamente pelo seu valor (que não se altera durante a execução do programa).

- A **variável** é representada no texto de um programa por um nome que corresponde a uma posição da memória que contém o seu valor.
- Em tempo de execução, o nome da variável permanece sempre o mesmo e seu valor pode ser modificado.
- **Nome de variável (identificador)**: é criado pelo programador e deve ser iniciado por uma letra que pode ser seguida por tantas letras, algarismos ou sublinha quanto se desejar e é aconselhável que seja significativo do valor que ela representa.

- Caracteres maiúsculos são diferentes de caracteres minúsculos (a variável **my_var** é diferente da variável **MY_VAR**)
- Uma variável não pode ter como nome uma **palavra reservada** da linguagem C (por exemplo **main**)
- Exemplos de nomes de variáveis:
 - Certo : nome, telefone, salario_func, x1
 - Errado: 1ano, sal/hora, #salario

- Todo valor (constante ou variável) de um programa tem um **tipo de dados** associado.
 - Um **tipo de dados** é constituído de dois conjuntos: um conjunto de objetos (domínio de dados) e de um conjunto de operações aplicáveis aos objetos do domínio.
 - Toda linguagem de programação tem embutido um conjunto de tipos de dados, também chamados de implícitos, primitivos ou básicos.
 - Tipos de dados básicos: inteiro, real, caracter e lógico.
 - Em C, os tipos acima são, respectivamente, **int**, **float/double** e **char**. Não há o tipo lógico em C.

Podemos criar um novo tipo de dados!(veremos no final do curso)

- **Inteiro:**

- **Domínio:** conjunto dos inteiros.
- **Operações:** usam dois argumentos inteiros e, de acordo com o resultado, são:
 - $+$, $-$, $*$, mod : resultado inteiro
 - $/$: resultado real
 - $<$, \leq , $>$, \geq , $=$, \neq : resultado lógico

- **Real:**

- **Domínio:** conjunto dos reais.
- **Operações:** usam dois argumentos reais e, de acordo com o resultado, são:
 - $+$, $-$, $*$, $/$: resultado real
 - $<$, \leq , $>$, \geq , $=$, \neq : resultado lógico

- **Caractere:**

- **Domínio:** conjunto dos caracteres alfanuméricos.
- **Operações:** usam dois argumentos do domínio e fornecem resultado lógico: $<$, \leq , $>$, \geq , $=$, \neq

- **Lógico:**

- **Domínio:** { verdadeiro, falso }.
- **Operações:** usam dois argumentos do domínio e fornecem resultado lógico:
 - Conectivos lógicos: conjunção (e, \wedge), disjunção (ou, \vee), disjunção exclusiva (xor, \oplus), negação (não, \neg). A negação trabalha somente com um argumento.
 - Conectivos relacionais: $=$, \neq .

Pseudolinguagem	C
$+$, $-$, $*$, $/$, $<$, $>$	$+$, $-$, $*$, $/$, $<$, $>$
Mod	$\%$
\leq	\leq
\geq	\geq
$=$	$==$
\neq	$!=$
\vee	$ $ ou $ $
\wedge	$\&\&$ ou $\&$
\neg	$!$

O símbolo " $=$ " em C é usado para atribuição.

Tipos de constantes e de variáveis

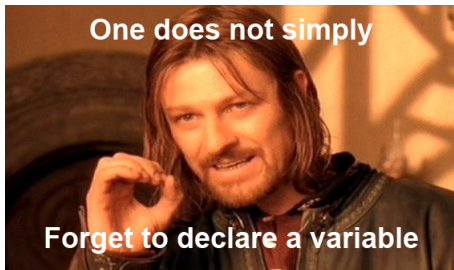
- O tipo básico associado a uma constante fica determinado pela própria apresentação da constante.
- Exemplos:
 - 7: inteiro
 - 7. ou 7.0: real
 - '7' : caractere
- Para as variáveis, devem ser feitas no **início** do programa (ou de um bloco) as **declarações** (de tipo) de variáveis.

Declaração de variáveis

- Uma variável precisa ser explicitamente declarada, antes de ser usada. A declaração tem uma série de objetivos:
 - Direciona o compilador a como o valor deve ser armazenado (qual o tipo de dado será armazenado na variável).
 - Uma quantidade suficiente de memória é alocada para a variável.
 - Declarada uma variável, toda vez que ela for referenciada em qualquer comando do programa, o computador vai trabalhar com o conteúdo de seu endereço, que é o valor da variável.
 - Pode inicializar o valor da variável. A posição de memória nunca está vazia, e por isso, uma variável, caso não seja inicializada, poderá armazenar um valor sem sentido algum para o programa. A inicialização é facultativa.

Sintaxe: tipo nome [= valor inicial];
 tipo nome1, nome2;

Declaração de variáveis



- ❶ Indique quais das constantes abaixo são do tipo inteiro:
☐ 1000 ☐ "0" ☐ -900" ☐ -456 ☐ 34 ☐ -1.56
- ❷ Indique quais das constantes abaixo são do tipo real:
☐ -678.0 ☐ "0.87" ☐ -9.12" ☐ -456.0
☐ "Cinco" ☐ -1.56
- ❸ Indique os nomes válidos para uma variável:
☐ endereco ☐ 21brasil ☐ nome_usuario ☐ nome*usuario
☐ cidade3 ☐ #cabeca

- 1 Indique quais das constantes abaixo são do tipo inteiro:
☒ 1000 ☐ "0" ☐ -900" ☒ -456 ☒ 34 ☐ -1.56
- 2 Indique quais das constantes abaixo são do tipo real:
☒ -678.0 ☐ "0.87" ☐ -9.12" ☒ -456.0
☐ "Cinco" ☒ -1.56
- 3 Indique os nomes válidos para uma variável:
☒ endereco ☐ 21brasil ☒ nome_usuario ☐ nome*usuario
☒ cidade3 ☐ #cabeca

- Comando de atribuição
 - **Sintaxe:** variável = expressão;
 - Semântica: atribuir à variável o resultado da expressão.
 - Uma variável só pode armazenar um valor por vez
 - A cada atribuição, realizamos uma operação de escrita na memória (essa operação sobrescreve o valor armazenado na variável anteriormente).

Comandos Básicos - Comando de atribuição

- Expressões aritméticas: fornecem resultado numérico (inteiro ou real).
 - Operações básicas: $+$, $-$, $*$, $/$
 - Exponenciação: $\text{potencia}(A, N)$
 - Funções matemáticas comuns: $\text{sen}(X)$, $\text{cos}(X)$, $\text{abs}(X)$, $\text{raiz}(X)$, $\text{arctan}(X)$, $\text{exp}(X)$, $\text{log}(X)$, $\text{ln}(X)$ etc.
 - Operador para inteiros: Mod
 - $M \text{ mod } I$: resto da divisão inteira de M por I
- Expressões lógicas: fornecem resultado lógico.
 - Conectivos lógicos: conjunção (e , \wedge), disjunção (ou, \vee), disjunção exclusiva (xor , \oplus), negação (não, \neg). A negação trabalha somente com um argumento.
 - Conectivos relacionais: $=$, \neq , $<$, \leq , \geq , $>$

Atenção: Em C não existe o tipo lógico, F ou V são representados como inteiro: 0 representa falso, 1 (ou simplesmente $\neq 0$) representa verdadeiro.

Comandos Básicos - Comando de atribuição

Tabela Verdade: Sejam A e B duas expressões lógicas.

A	B	A e B	A ou B	A xou B	não A
V	V	V	V	F	F
V	F	F	V	V	F
F	V	F	V	V	V
F	F	F	F	F	V

- Prioridade de execução das operações em uma expressão (precedência):
 - 1 Parênteses (dos mais internos para os mais externos)
 - 2 Expressões aritméticas, seguindo a ordem: funções, $*$ e $/$, $+$ e $-$
 - 3 Comparações: $=$, \neq , $<$, \leq , \geq , $>$
 - 4 não
 - 5 e
 - 6 ou e xou
 - 7 Da esquerda para a direita quando houver indeterminações.
- A atribuição só é realizada depois que a expressão for resolvida.

- Comentários
 - `int maior; //maior valor lido`
 - `/*Com isso eu consigo escrever mais de uma linha de comentário */`
- Comandos de entrada e saída
 - `leia (a, x); //serão lidos os valores das variáveis a, x nesta ordem`
 - `imprima("Valor de N =", n, "Fatorial de N=", fat);`

- Bloco

- Conjunto de comandos entre { }
- Pode-se declarar variáveis em seu interior.
- Delimitar o escopo das variáveis.

```
{  
<declaração de variáveis>;  
<comandos>;  
}
```

- Exemplo de Sequência Simples

Exemplo em pseudolinguagem	Exemplo em C
inteiro X,Y; leia(X,Y); X = X + Y; imprima(X);	int X, Y; scanf("%d %d", &X, &Y) X = X + Y; printf("%d", X);

1. Marque as declarações válidas:

- ☐ `int a,`
- ☐ `char c;`
- ☐ `int a,b,a;`
- ☐ `float f1,f2,f3,4f;`
- ☐ `int meu_nro;`
- ☐ `float leitura_sensor;`

2. Sendo $A=3$, $B=7$ e $C=4$, informe se as expressões abaixo são verdadeiras ou falsas:

- a) $(A + C) > B$
- b) $B \geq (A + 2)$
- c) $C = (B - A)$
- d) $(B + A) \leq C$
- e) $(C + A) > B$

1. Marque as declarações válidas:

☐ `int a,`

☒ `char c;`

☐ `int a,b,a;`

☐ `float f1,f2,f3,4f;`

☒ `int meu_nro;`

☒ `float leitura_sensor;`

2. Sendo $A=3$, $B=7$ e $C=4$, informe se as expressões abaixo são verdadeiras ou falsas:

a) $(A + C) > B$ **F**

b) $B \geq (A + 2)$ **V**

c) $C = (B - A)$ **V**

d) $(B + A) \leq C$ **F**

e) $(C + A) > B$ **F**

3. Sendo $A=7$, $B=3$ e $C=2$ e $D=10$, informe se as expressões abaixo são verdadeiras ou falsas.

a) $(A > C)E(C \geq D)$

b) $(A + B) > 10OU(A + B) \leq (C + D)$

c) $(A \geq C)E(D \geq C)$

d) NAO $(A > 3)$

3. Sendo $A=7$, $B=3$ e $C=2$ e $D=10$, informe se as expressões abaixo são verdadeiras ou falsas.

- a) $(A > C)E(C \geq D)$ **F**
- b) $(A + B) > 10OU(A + B) \leq (C + D)$ **V**
- c) $(A \geq C)E(D \geq C)$ **V**
- d) $NAO (A > 3)$ **F**

4. Desenvolver os algoritmos a seguir utilizando pseudolinguagem:
- a) Ler uma temperatura em graus Celsius e apresentá-la convertida em graus Fahrenheit. A fórmula de conversão: $F = (9 \cdot C + 160) / 5$
 - b) Calcular e apresentar o volume de uma lata de óleo cilíndrica, a partir da leitura do raio da base e da altura.
 - c) Ler os valores do comprimento, da largura e da altura de uma caixa, calcular e imprimir o seu volume.

Alguns pontos importantes

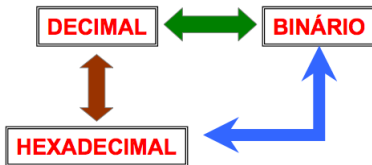
- Lembrando: Quando você declara um identificador dá a ele um tipo. Um tipo de objeto de dados determina como valores de dados são representados, que valores pode expressar, e que tipo de operações você pode executar com estes valores.

Tipo	Espaço que ocupa na memória	Faixa
char	1 byte	-128 a 127 (incluindo letras e símbolos)
int	4 bytes	-2147483648 a 2147483647
float	4 bytes	3.4E-38 a 3.4E+38 (6 casas de precisão)
double	8 bytes	1.7E-308 a 1.7E+308 (15 casas de precisão)

1 byte = 8 bits

1 bit pode ser 0 ou 1

Como fazer a conversão entre o sistema decimal e o sistema utilizado num computador?



- **Base** ou **raiz** de um sistema de numeração: é o número de algarismos distintos usados nesse sistema de numeração.
 - Exemplo: o sistema decimal possui base 10, isto é, usa 10 algarismos distintos.
- **Notação posicional**: notação usada por alguns sistemas numéricos, onde cada algarismo tem, além do seu valor absoluto, um valor de posição dentro de cada número desse sistema em que ele aparece.
 - Exemplo: O valor absoluto **2** no número **2000** representa uma grandeza diferente do que **2** em **20**.

- Base 10.
- Possui 10 algarismos distintos (algarismos arábicos = 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9) e usa notação posicional.
 - $7 = 7 \times 10^0$
 - $35 = 30 + 5 = 3 \times 10^1 + 5 \times 10^0$
 - $81,508 = 8 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1} + 0 \times 10^{-2} + 8 \times 10^{-3}$
- Na notação posicional (qualquer que seja a base) o primeiro algarismo a esquerda da vírgula, representa uma potência da base com expoente igual a 0 (zero) e esse expoente é inteiro e crescente para a esquerda.

Sistema Binário

- Base 2.
- Usa notação posicional e possui dois algarismos distintos: 0 e 1.

Decimal	Binário
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

E como eu faço essa conversão??

Sistema Hexadecimal

- Base 16.
- Usa notação posicional e possui 16 algarismos distintos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F.

Decimal	Hexadecimal	Decimal	Hexadecimal
0	0	8	8
1	1	9	9
2	2	10	A
3	3	11	B
4	4	12	C
5	5	13	D
6	6	14	E
7	7	15	F

E como eu faço essa conversão??

- **Conversão da base b (qualquer) para decimal:** Para converter um número na **base b** em decimal, basta somar os produtos dos algarismos pelas potências da **base b** que eles representam.

$$(10)_{16} = 1 \times 16^1 + 0 \times 16^0 = (16)_{10}$$

$$(F30A)_{16} = 15 \times 16^3 + 3 \times 16^2 + 0 \times 16^1 + 10 \times 16^0 = (62218)_{10}$$

$$(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (13)_{10}$$

$$(10001111)_2 = 2^7 + 2^3 + 2^2 + 2^1 + 2^0 = (143)_{10}$$

Conversão de base

- **Conversão de decimal para a base b (qualquer):** Para converter um número decimal para a base b, devem ser feitas divisões inteiras sucessivas por b até que se encontre quociente 0 (zero). O número correspondente na base b será formado pelos restos das divisões, da última até a primeira divisão, nessa ordem.

Ex.: Converter o número decimal abaixo para hexadecimal

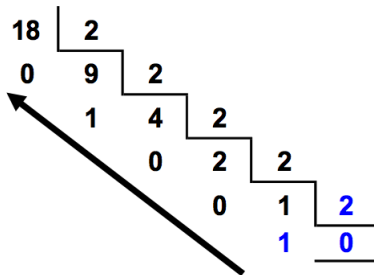
45286 \div 16 = 2830 \div 16 = 176 \div 16 = 11 \div 16 = 0

6 14 (E) 0 11 (B)

$(45286)_{10} \equiv (B0E6)_{16}$

Conversão de base

- Converter o número decimal abaixo para binário.



$$18_{(10)} = 10010_{(2)}$$

- Converter binário para hexadecimal: Como 16 é potência de 2 ($2^4 = 16$), nesta conversão, cada algarismo hexadecimal dá origem a quatro algarismos binários.

Hexadecimal para Binário

$$(9)_{16} = (1001)_2$$

$$(D)_{16} = (1101)_2$$

$$(13A)_{16} = (0001\ 0011\ 1010)_2$$

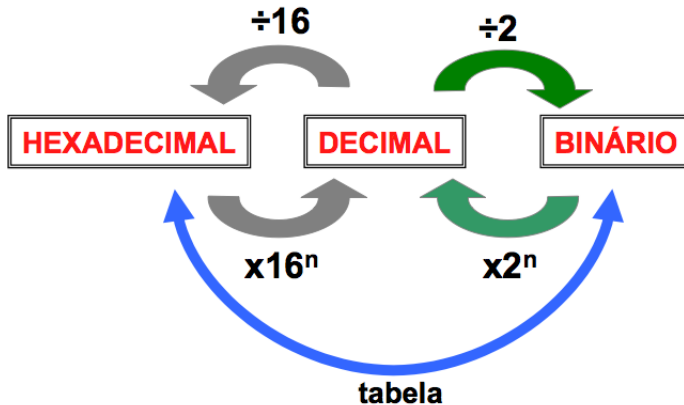
$$(FB09)_{16} = (1111\ 1011\ 0000\ 1001)_2$$

Binário para Hexadecimal

$$(1001)_2 = (9)_{16}$$

$$(0010\ 1101\ 1011\ 0101)_2 = (2DB5)_{16}$$

$$(1000\ 1001\ 1011\ 1111)_2 = (89BF)_{16}$$



5. Faça as conversões abaixo:
- a. $(10011)_2$ para decimal
 - b. 257 para binário
 - c. $(AB3)_{16}$ para binário e para decimal
 - d. $(000110100101)_2$ para hexadecimal

6. Construir um algoritmo para ler 5 valores inteiros, calcular e imprimir a soma desses valores.
7. Construir um algoritmo para ler 6 valores reais, calcular e imprimir a média aritmética desses valores.
8. Fazer um algoritmo para gerar e imprimir o resultado do número H , sendo $H = 1 + 1/2 + 1/3 + 1/4 + 1/5$.

9. Calcular o aumento que será dado a um funcionário, obtendo do usuário as seguintes informações : salário atual e a porcentagem de aumento. Apresentar o novo valor do salário e o valor do aumento.
10. A nota final de um aluno é dada pela média ponderada das notas das provas. Sabendo que o professor deu 3 provas, com pesos 4, 3 e 3, respectivamente, calcule a nota final do aluno.

Na próxima aula...

Construindo um programa em C