

# TIPOS DE DADOS

Aula 05 – Algoritmos

Prof. Allbert Velleniche de Aquino Almeida

E-mail: [professor@allbert.com.br](mailto:professor@allbert.com.br)

Site: <http://www.allbert.com.br>

# Tipos de dados

- Um programa recebe dados que precisam ser armazenados na memória principal do computador para serem utilizados no processamento.
  - **Tipo Inteiro**
  - **Tipo Real**
  - **Tipo Caracter**
  - **Tipo Lógico**

# Tipo Inteiro

- Dados do tipo inteiro são os dados numéricos positivos, negativos ou nulos que não possuem a parte decimal ou fracionária.
- Ex.: 32, -43, 0, 32768, -32767.

# Tipo Real

- Os dados do tipo real são dados numéricos positivos, negativos ou nulos cujos valores podem possuir componentes decimais ou fracionários. A parte inteira é separada da parte fracionária por um “ponto” decimal. Todo número inteiro é um caso particular de um número real.
- Ex.: 32, -43, 0, 32769, -32768, 32.01, -49.5

# Tipo Character

- Dados do tipo string são aqueles dados não numéricos cujo conteúdo pode conter uma sequência contendo letras, dígitos e símbolos especiais delimitados por aspas (“”).
- Ex.: “Rua São Bernardo, nº 44”, “987656589”, “a”.

# Tipo Lógico

- Os dados do tipo lógico são também chamados booleanos e podem assumir os valores verdadeiro ou falso. É comum encontrar outros pares de valores lógicos equivalentes, como sim/não, 1 /0, true/false, verdadeiro/falso.

# Variáveis



- Uma variável pode ser vista como uma caixa com um rótulo(nome) e tipo colado a ela, que num dado instante guarda um determinado valor.

# Declaração de variáveis

- Todas as variáveis devem ser declaradas no início do algoritmo. A declaração de uma variável deve seguir a seguinte sintaxe:

`<var1 >, <var2 >, ..., <varn > : <tipo >`

onde:

`var1, var2, ..., varn`: nomes das variáveis.

`<tipo >` identifica o tipo da variável e pode ter os seguintes valores: inteiro, real, caracter, lógico (booleano).

# Declaração de variáveis

## □ Válidos

- A

- a

- Nota

- X5

- A32

- NOTA1

- MATRICULA

- Nota\_1

- Dia

- IDADE

# Declaração de variáveis

## □ Inválidos

### □ 5b

- por começar por número

### □ e 12

### □ prova 2n

- Por conter espaço em branco

### □ x-y

### □ nota(2)

- Por conter caracter especial

### □ PARA

### □ algoritmo

- Por ser palavra reservada

# Declaração de variáveis

## PALAVRAS RESERVADAS

aleatorio	escreva	inicio	pausa
algoritmo	escreval	int	real
arquivo	faca	inteiro	procedimento
ate	falso	interrompa	repita
caractere	fimalgoritmo	leia	retorne
caso	fimenquanto	limpatela	se
cronometro	fimescolha	logico	senao
debug	fimfuncao	mod	timer
e	fimpara	nao	var
eco	fimprocedimento	ou	vetor
enquanto	fimrepita	outrocaso	verdadeiro
entao	fimse	para	Xou
escolha	funcao	passo	

# Operador de atribuição de variáveis

- Para atribuir um valor a uma variável dentro de um algoritmo, utilizamos o operador de atribuição representado por uma seta (<-) apontando para a esquerda.
- Exemplo:
  - ▣ `Peso <- 78.7 //` Este comando atribui à variável `Peso` o valor `78.7`.
  - ▣ `Nome <- "João da Silva" //` Este comando atribui à variável `Nome` o valor `"João da Silva"`.
  - ▣ `Achei <- FALSO //` Este comando atribui à variável `Achei` o valor `FALSO`.

# Comando inválido

- É importante lembrar que só se pode atribuir às variáveis valores do mesmo tipo da variável.
- Assim, o seguinte comando seria inválido:

Exemplo:

```
VAR
```

```
salario: REAL
```

```
INICIO
```

```
salario <- "Insuficiente"
```

# Comando inválido

- ❑ Deve estar claro, também, que sempre à esquerda do comando de atribuição deve haver somente um identificador de variável.
- ❑ Assim, são incorretos os seguintes comandos:

Exemplo:

```
2060 <- NumeroConta
```

```
NumeroAgencia+digitoControle <- 2345 + 0
```

```
NomeCliente+sobrenome <- "João" + "Silva"
```

# Expressões

Operador	Usado	Exemplos
Adição	+	$a + b$ ;
Subtração	-	$a - b$ ;
Multiplicação	*	$a * b$ ;
Divisão	/	$a/b$ ;
Divisão inteira	\	$a \backslash b$ ;
Resto	% ou MOD	$a \% b$ ;
Exponenciação	^ (base^expoente)	$a^b$ ;

# Precedência de operadores

Seq	Descrição	Símbolo	Observação
1°	Parênteses Funções	()	Resolvidos da esquerda para a direita
2°	Exponenciação Multiplicação Divisão Resto Divisão inteira	$\wedge$ $*$ $/$ $\%$ ou MOD $\backslash$	Resolvidos da esquerda para a direita
3°	Soma Subtração	$+$ $-$	
4°	Operadores relacionais	$>$ $<$ $>=$ $<=$ $<>$	
5°	Operadores lógicos	NÃO E OU	

# Linearização de Expressões

- Para a construção de algoritmos que realizam cálculo matemáticos, todas as expressões aritméticas devem ser linearizadas.
- Exemplo:

$$\left\{ \left[ \frac{2}{3} - (5 - 3) \right] + 1 \right\} . 5 \quad ( ( 2 / 3 - ( 5 - 3 ) ) + 1 ) * 5$$