

Algoritmos: Conceitos Fundamentais



Instituto de Ciências Matemáticas e de Computação

Slides elaborados pela

Prof(a). Simone do Rocio Senger de Souza

1o semestre/2012



DADOS, INSTRUÇÕES E EXPRESSÕES

Uso da Informação

- Computador manipula **informações** contidas em sua **memória**.
- Classificadas em dois tipos:
 - **Instruções**: comandam o funcionamento da máquina e determinam a maneira como os dados devem ser tratados.
 - **Dados**: informação que devem ser manipulada pelo computador.

1- DADOS

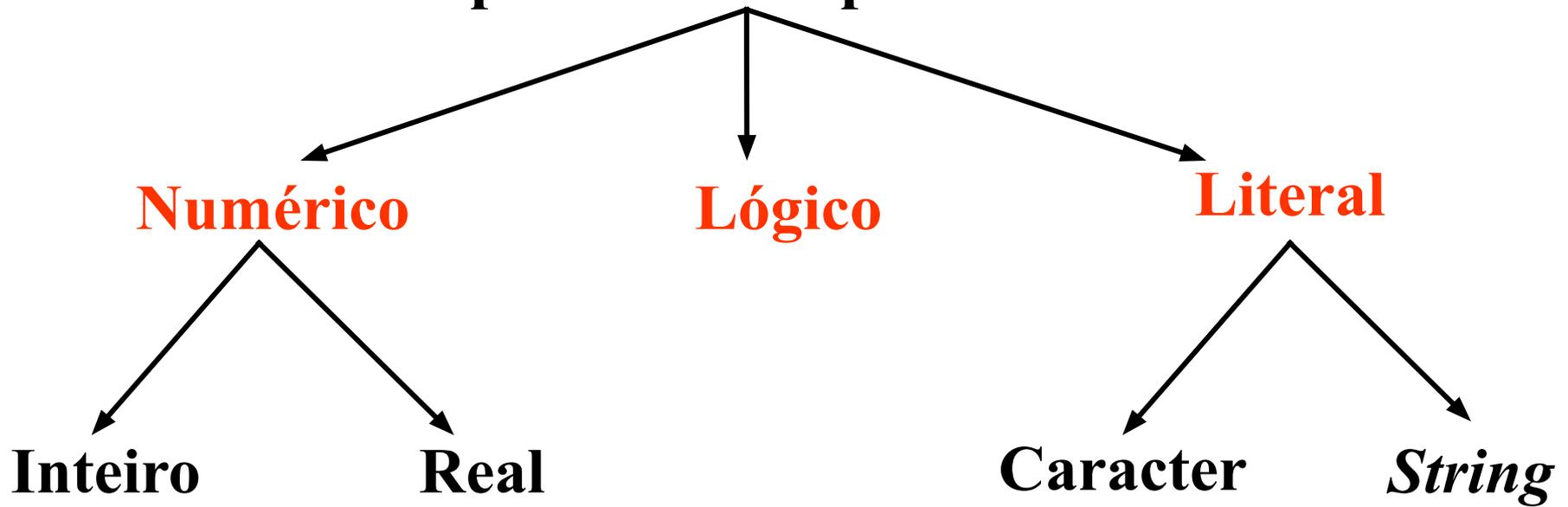
- Objetivo do computador: realizar tarefas envolvendo informações ou DADOS
- Exemplo de DADOS:
 - notas
 - nomes
 - medidas
- Existem vários tipos de dados
 - cada tipo é representado e processado de forma diferente

2- TIPO DE DADOS

- Definir um tipo de dados serve a dois propósitos:
 - Classificar os dados de acordo com as informações contidas neles.
 - indicar quanto espaço de memória deve ser alocado

2- TIPO DE DADOS

Tipos de dados primitivos



2- TIPO DE DADOS - INTEIRO

- **Inteiro**: Toda e qualquer informação numérica que pertença ao conjunto dos **números inteiros** relativos (negativa, nula ou positiva)
- Exemplos:
 - Ele tem **15** irmãos
 - A escada possui **8** degraus
 - Meu vizinho comprou **2** carros novos
- Linguagem Python: tipo escalar **int**

2- TIPO DE DADOS - REAL

- **Real**: Toda e qualquer informação numérica que pertença ao conjunto dos **números reais** (negativa, nula ou positiva)
- Exemplos:
 - Ela tem **1,73** metro de altura (usamos 1.73)
 - Meu saldo bancário é de **215,20** (215.20)
 - No momento estou pesando **82,5** Kg (82.5)

2- TIPO DE DADOS - REAL

- Números reais muito grandes ou muito pequenos são escritos em forma de ponto flutuante (ou notação científica)
- Exemplo:
+3.14E+07 (3.14 x 10⁷)
5.1E-18 (5.1 x 10⁻¹⁸)
- Linguagem Python: tipo escalar **float**

2- TIPO DE DADOS - Literal

- **Literal**: Toda e qualquer informação composta por um conjunto de caracteres alfanuméricos: **numéricos** (0...9), **alfabéticos** (A...Z, a... Z) e **especiais** (por exemplo, #, ?, !, @).
- Exemplos:
 - Constava na prova: *‘Use somente caneta!’*.
 - O parque municipal estava repleto de placas: *‘Não pise na grama’*.
 - O nome do vencedor é *‘Felisberto Laranjeira.’*
- **Linguagem Python**: tipo não escalar **string**

2- TIPO DE DADOS - LÓGICO

Lógico: Toda e qualquer informação que pode assumir apenas duas situações (**biestável**)

- Exemplos:
 - A porta pode estar *aberta* ou *fechada*.
 - A lâmpada pode estar *apagada* ou *acesa*.
- Linguagem Python: tipo escalar **bool** assumindo valores **True** ou **False**

2- TIPO DE DADOS - Python

- Python utiliza o conceito de **Objetos**.
- Programas manipulam *data objects*.
- Cada objeto pertence a um tipo.
 - Darth Vader é Sith.
 - Yoda é Jedi.
 - Chewbacca é Wookiee.

2- TIPO DE DADOS - Python

- Python divide os objetos em
 - escalar: Não podem ser subdivididos
 - numéricos: int e float.
 - lógico: bool (**True e False**)
 - **especial**: NoneType tem um único valor associado - **None**.

2- TIPO DE DADOS - Python

- Python divide os objetos em
 - Não escalar: Possuem estrutura interna
 - Strings e....**sem spoilers por enquanto**

2- TIPO DE DADOS - Manipulando

- comando `type()` retorna o tipo da variável:

```
>> type(34.9)
```

```
float
```

```
>> type(2)
```

```
int
```

2- TIPO DE DADOS - Manipulando

- **conversão de tipos**

```
>> int (34.9)
```

```
34
```

```
>> float(2)
```

```
2.0
```

2- TIPO DE DADOS - Operadores

- Operadores para int e float
 - Soma: $a+b$
 - Resulta int ou float
 - Produto: $a*b$
 - Resulta int ou float
 - Divisão: a/b
 - Resulta float

2- TIPO DE DADOS - Operadores

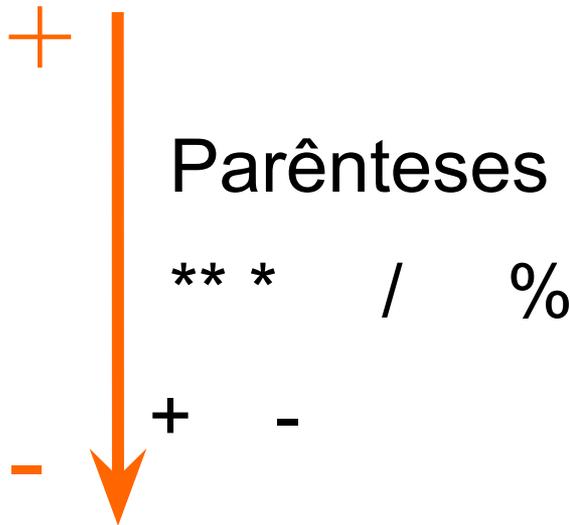
- Resto: $a\%b$ (módulo)
 - Resto inteiro da divisão de a e b inteiros
- Produto: $a**b$ (potência a^b)
 - Resulta int ou float
- Divisão inteira: $a//b$
 - Retorna o quociente inteiro de a/b

2- TIPO DE DADOS - Operadores

- Resto: $a\%b$ (módulo)
 - Resto inteiro da divisão de a e b inteiros
- Produto: $a**b$ (potência a^b)
 - Resulta int ou float
- Divisão inteira: $a//b$
 - Retorna o quociente inteiro de a/b

Expressão Aritmética

- Na resolução das expressões aritméticas, as operações guardam uma hierarquia entre si, sendo resolvidos da esquerda para direita se são de igual prioridade:



OPERADORES RELACIONAIS

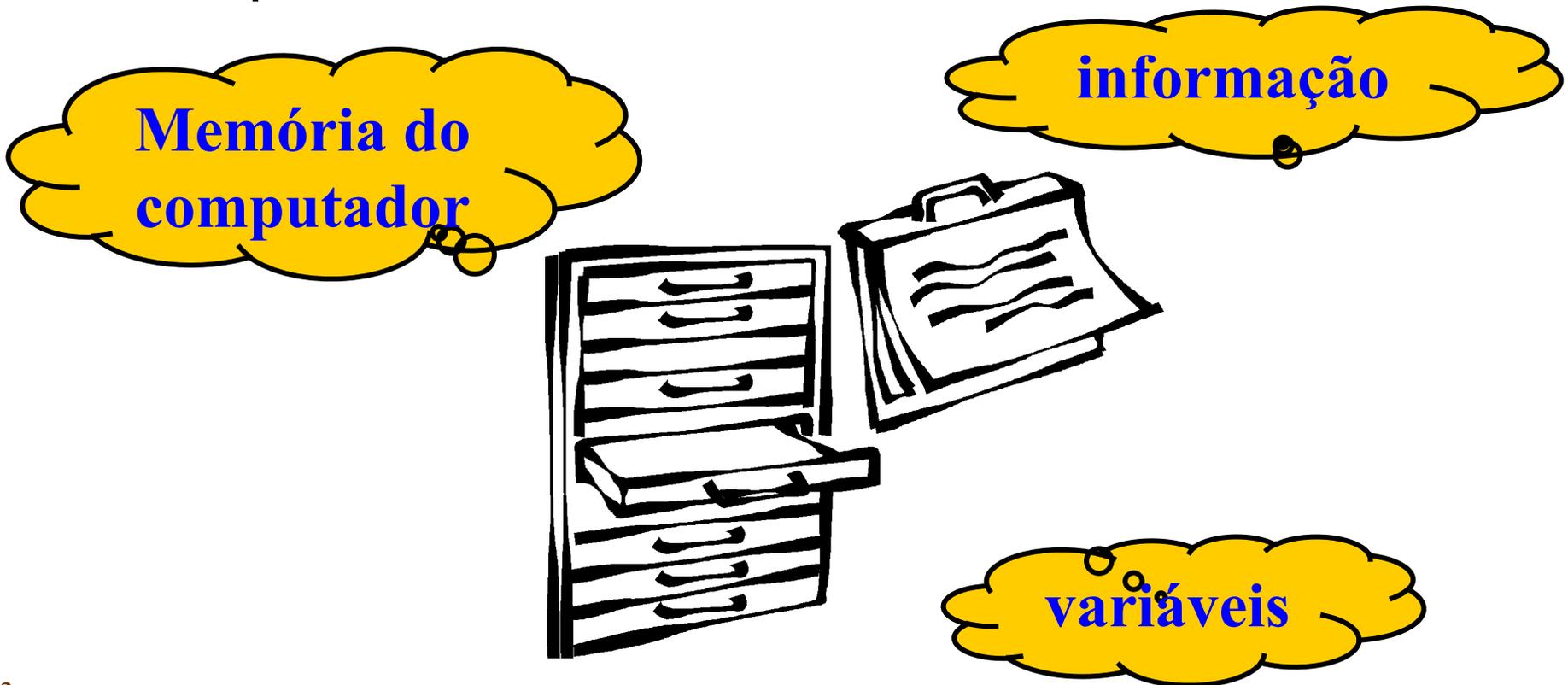
OPERADORES	==	igual
	!=	diferente
	>	maior
	<	menor
	>=	maior ou igual
	<=	menor ou igual

3- Variáveis

- Durante a execução do programa os **dados** estão sendo manipulados
- Para que o computador não esqueça das informações **contidas em um dado** é necessário guarda-las em sua **memória**.
- As **variáveis** guardam informações sobre os dados (o seu conteúdo) que estão sendo manipulados.

3- Variáveis

- Armazenamento das variáveis na memória do computador.



3- Variáveis

- Armazenamento das variáveis na memória do computador.



3- Variáveis

- O conteúdo da variável é **substituído** por outro que lhe será **atribuído**.
- O uso de uma variável em uma expressão representa o seu conteúdo **naquele momento**.
 - O **uso** não muda o seu conteúdo

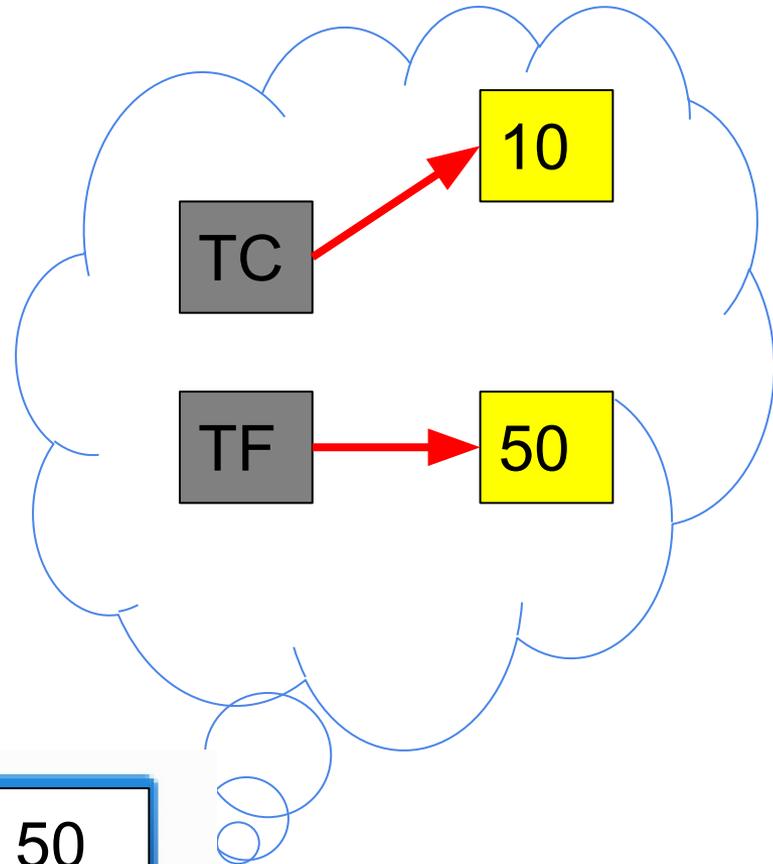
3- Variáveis

```
TC = 10  
TF = TC*9/5 + 32  
print(TF)
```

```
TC = 35
```

```
TF=TC*9/5+32
```

```
print(TF)
```



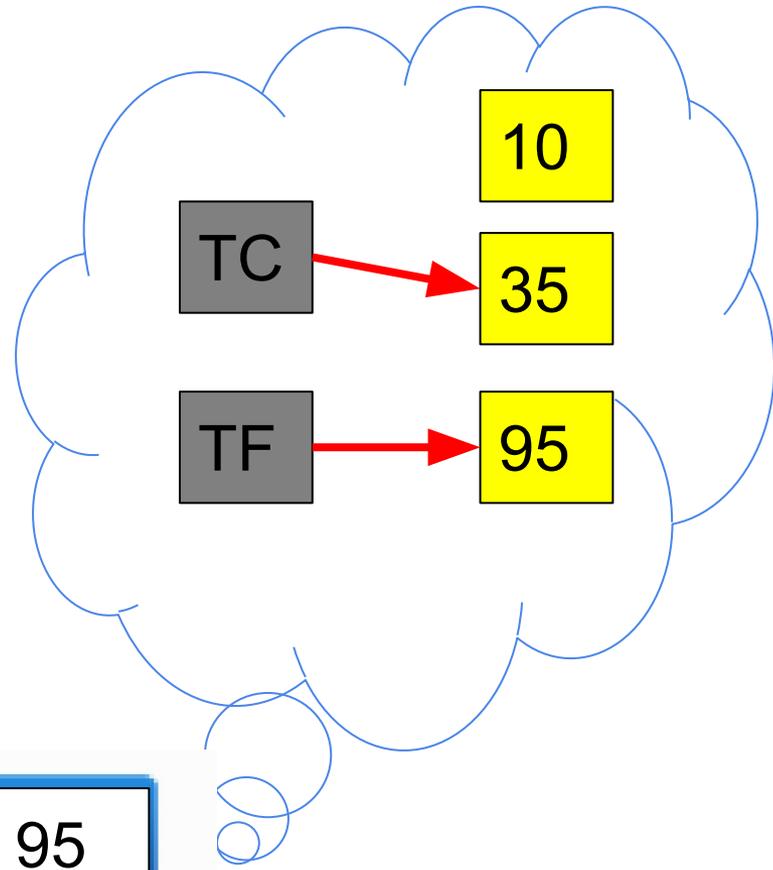
3- Variáveis

```
TC = 10  
TF = TC*9/5 + 32  
print(TF)
```

```
TC = 35
```

```
TF=TC*9/5+32
```

```
print(TF)
```



3- Variáveis

a,b = 10,20

print(a+b)  30

a = a+1

print(a)  11

a += 1

print(a)  12

print('a= ', a, 'b=', b)  a= 12 b= 20

Expressão Lógica

- Denomina-se **expressão lógica** aquela cujos operadores são lógicos ou relacionais e cujos operandos são relações ou variáveis ou constantes do tipo lógico.
- **Exemplo:**

$(A+B == 0) \text{ and } (C != 1)$

OPERADORES RELACIONAIS

- Utilizam-se os **operadores relacionais** para realizar comparações entre dois valores de mesmo tipo primitivo.
- Tais valores são representados por constantes, variáveis ou expressões aritméticas
- O resultado obtido de uma relação é sempre um **valor lógico**.

OPERADORES LÓGICOS

- Utilizam-se três **conectivos** básicos para a formação de novas proposições lógicas compostas a partir de outras proposições lógicas simples.

OPERADORES LÓGICOS

and

e

or

ou

not

não

TABELAS VERDADE

- **Tabelas Verdade** é o conjunto de todas as possibilidades combinatórias entre os valores de diversas variáveis lógicas, as quais se encontram em apenas duas situações (V ou F), e um conjunto de operadores lógicos

TABELAS VERDADE

Operação de Negação

A	not A
true	false
false	true

Expressão Lógica

TABELAS VERDADE

Operação de Conjunção (and)

A	B	A and B
true	true	true
true	false	false
false	true	false
false	false	false

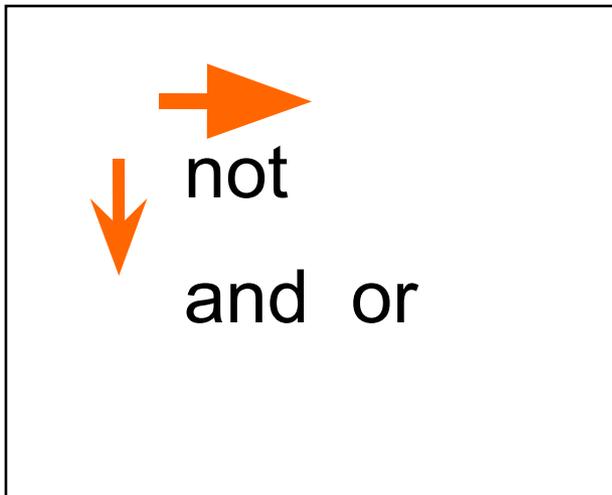
TABELAS VERDADE

Operação de Disjunção Não-Exclusiva (or)

A	B	A or B
true	true	true
true	false	true
false	true	true
false	false	false

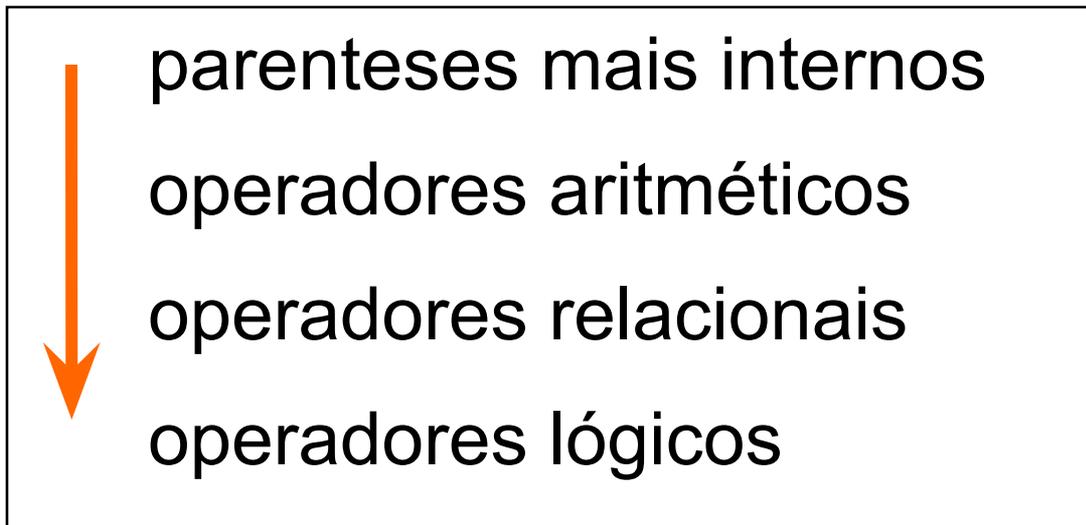
OPERADORES LÓGICOS

- Na resolução das expressões lógicas, os operadores lógicos guardam uma hierarquia entre si:



OPERADORES LÓGICOS

- Na resolução das expressões lógicas, os diversos operadores guardam uma hierarquia entre si:



Expressão Lógica

- Exemplos:

```
not ( 5 != 10 / 2 or True and 2 - 5 > 5 - 2 or True )
```

```
not ( 5 != 5 or True and -3 > 3 or True )
```

```
not ( False or True and False or True )
```

```
not ( False or False or True )
```

```
not ( False or True )
```

```
not ( True )
```

```
False
```

Strings

- Letras, espaços, caracteres especiais.
- não escalável.
- Exemplos:
 - frase="ola mundo"
 - palavra= " bonito"
 - novafrase=frase + palavra
 - concatenou gerando "ola mundo bonito"
 - oqueresulta=frase+palavra*5

Strings

- Ordem lexicográfica

“ana” < “joao”

True

“ana” < “ah”

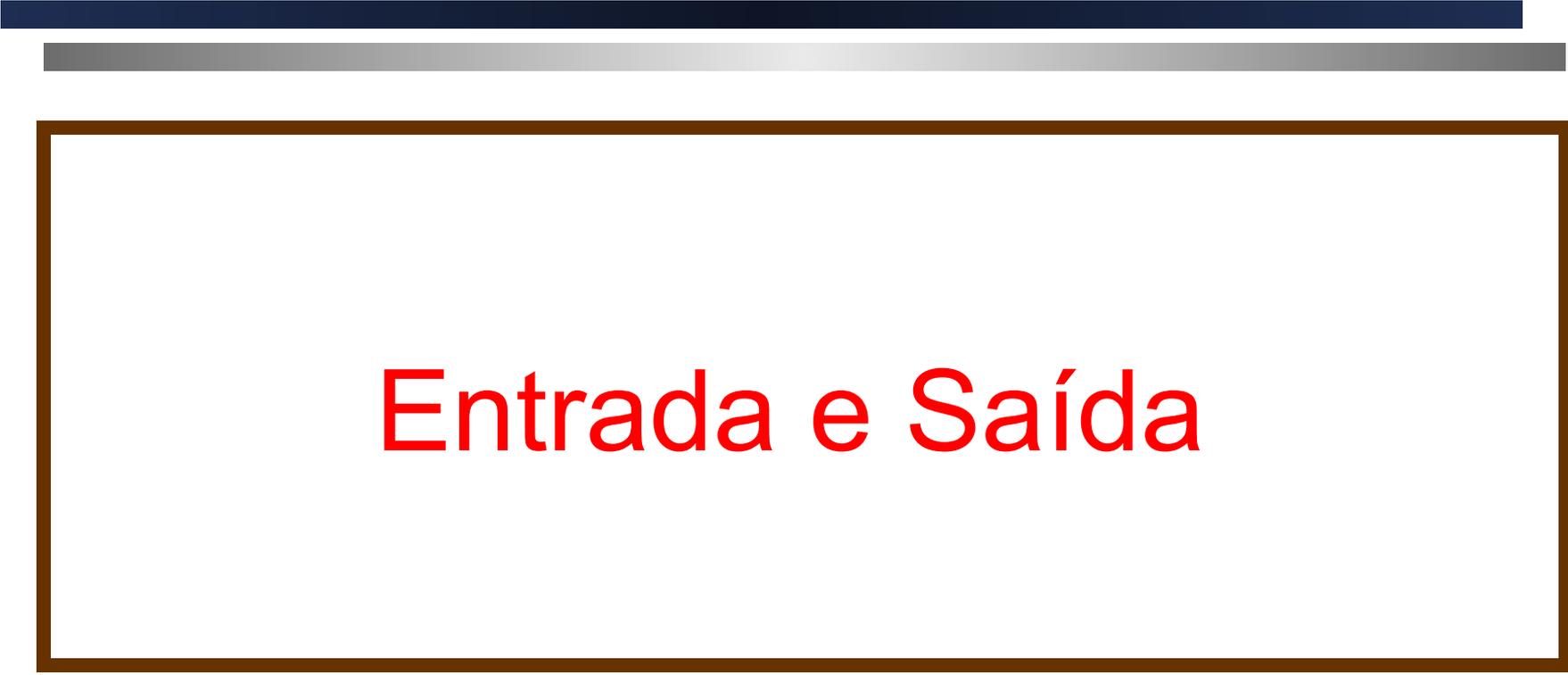
False

“ana” == “Ana”

False

“ana” > “Ana”

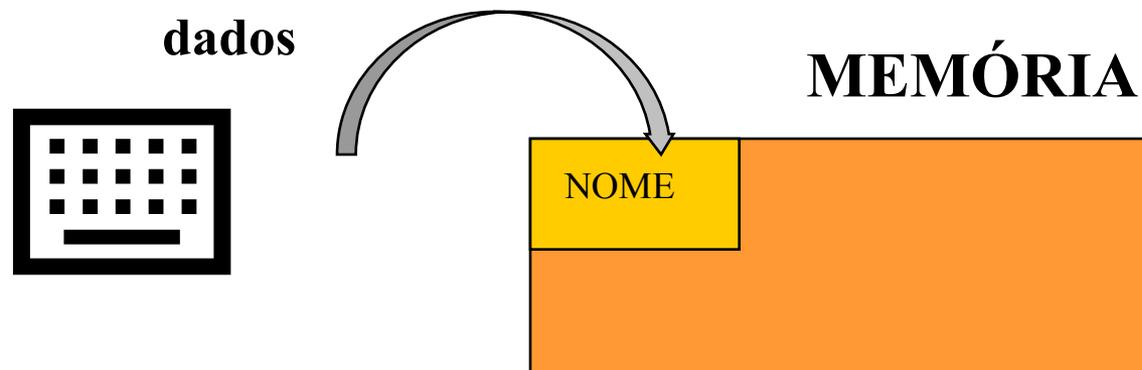
True



Entrada e Saída

Entrada

- O comando de entrada é utilizado para receber dados digitados pelo usuário (**DADOS DE ENTRADA**) e armazená-los em variáveis.
- Os dados de entrada são fornecidos ao sistema através de uma **unidade de entrada**, por exemplo o **teclado**.



Entrada - Input("")

- `input("")`: retorna uma string

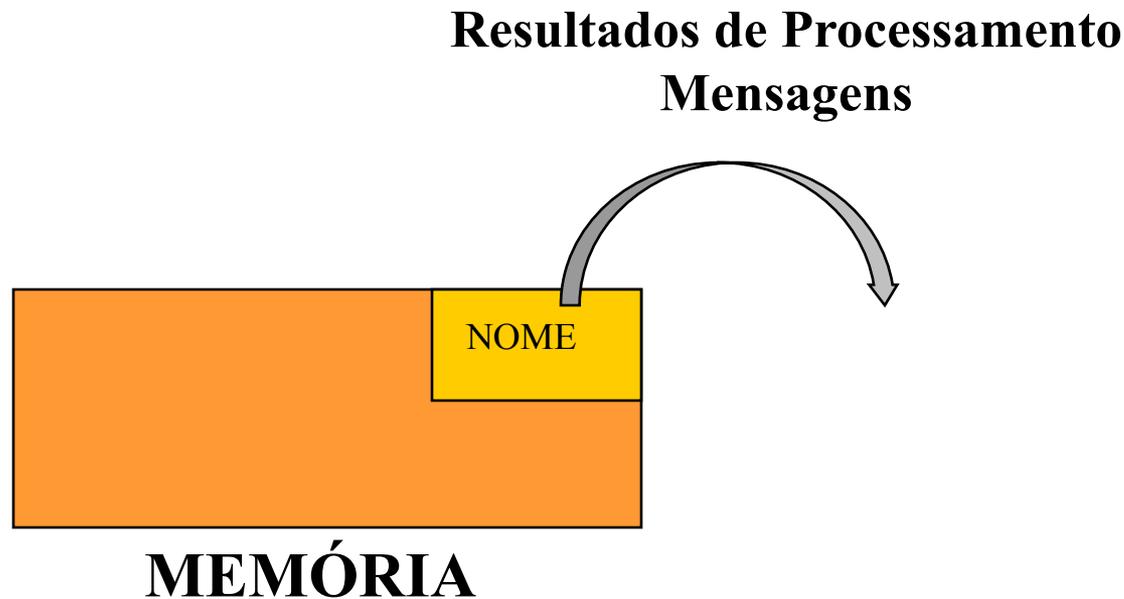
```
nome= input("digite um nome...")  
print(nome)
```

- Precisa converter para outro tipo.

```
idade = int(input("digite idade..."))  
print(idade+1)
```

Saída

- O comando de saída é utilizado para que o sistema forneça, numa **unidade de saída**, os **resultados** do processamento e **mensagens**.



Saída

- **Resultados de Processamento:** podem ser fornecidos através de conteúdos de **variáveis**, conteúdos de **constantes** e resultados de **expressões aritméticas e lógicas**.
- **Mensagens:** são utilizadas para que o programa dê **informações** ao usuário a respeito do **processamento** que está se realizando. Podem ser fornecidas através de conteúdo de **variáveis**, ou constantes do tipo **string** ou da **mensagem** propriamente dita.

Saída - print

```
idade = 10
```

```
str_idade=str(idade)
```

```
print("Joao tem ", idade, " anos")
```

```
print("Joao tem "+str_idade+ " anos")
```