



Aula 03:

Comandos de Decisão

Programação de Computadores I

Túlio Toffolo

www.toffolo.com.br

Departamento de Computação
Universidade Federal de Ouro Preto

Agenda

1. Bibliotecas
2. Biblioteca matemática
3. Operadores relacionais e de igualdade
4. Operadores lógicos
5. O comando if
6. O comando if-else
7. Exemplos e exercícios

Bibliotecas

Bibliotecas

A linguagem Python possui diferentes módulos que fornecem funções úteis para o programador.

- No entanto, para usar qualquer módulo em Python é necessário importá-lo antes.
- A importação é feita pelo comando `import`.
 - Exemplo: `import math` carrega a biblioteca matemática *.
 - Outro exemplo: `import numpy` carrega a biblioteca NumPy *.

(*) A importação é necessária **apenas uma vez** para cada programa.

Biblioteca matemática

Biblioteca matemática: **math**

A biblioteca `math` fornece várias **funções** e **constantes** para realizar cálculos matemáticos.

Nome	Descrição	Exemplo	Resultado	Módulo
<code>round</code>	Arredonda n casas	<code>round(5.35, 1)</code>	5.4	-
<code>abs</code>	Valor absoluto	<code>abs(-2)</code>	2	-
<code>sqrt</code>	Raiz quadrada	<code>sqrt(9)</code>	3	<code>math</code>
<code>cos</code>	Cosseno	<code>cos(30)</code>	0.1542514	<code>math</code>
<code>tan</code>	Tangente	<code>tan(7.3456)</code>	1.7945721	<code>math</code>
<code>sin</code>	Seno	<code>sin(pi)</code>	1.2246e-16	<code>math</code>
<code>trunc</code>	Parte inteira	<code>trunc(5.2)</code>	5	<code>math</code>
<code>ceil</code>	Próximo inteiro (teto)	<code>ceil(5.2)</code>	6	<code>math</code>
<code>floor</code>	Inteiro anterior (pisso)	<code>floor(5.2)</code>	5	<code>math</code>
<code>pow</code>	Potência	<code>pow(2,3)</code>	8	<code>math</code>

Constantes úteis da biblioteca **math**

Constante	Valor	Módulo
pi	3.141592...	math
e	2.718281...	math
tau	6.283185...	math
inf	representa um número infinito	math

Exemplos de uso

Crie um programa que solicita o valor x e, em seguida, calcula e imprime o valor de $\text{seno}(x)$.

```
1 import math
2
3 x = float(input("Digite o valor de x: "))
4 resultado = math.sin(x)
5 print(f"seno({x}) = {resultado}")
```

Exemplos de uso

Crie um programa para calcular e imprimir a raiz quadrada de 64.

```
1 import math
2
3 resultado = math.sqrt(64)
4 print(resultado)
```

Note que poderíamos ter usado o operador `**`:

```
1 resultado = 64 ** (1/2)
2 print(resultado)
```

Atenção: lembrem-se que `64 ** (1/2)` é diferente de `64 ** 1/2`.

Exemplos de uso

Criar um programa para calcular e imprimir o valor absoluto de um número digitado pelo usuário.

```
1 x = float(input("Digite um número: "))
2 resultado = abs(x)
3 print(f"O valor absoluto de {x} é {resultado}")
```

Note que a função `abs()` não precisa ser importada.

Exemplos de uso

Crie um programa para calcular e imprimir o cosseno de 30 radianos.

```
1 import math
2
3 resultado = math.cos(30)
4 print(resultado)
```

O programa imprimirá:

```
1 0.15425144988758405
```

Funções de logaritmo

`math.log(x)`

- Calcula o logaritmo natural de x .

`math.log10(x)`

- Calcula o logaritmo de x na base 10.

`math.log2(x)`

- Calcula o logaritmo de x na base 2.

`math.log(x, y)`

- Calcula o logaritmo de x na base y .

Exemplos de uso

Criar um programa para calcular e imprimir o logaritmo natural de 10, ou seja, o $\ln 10$ ou $\log_e 10$.

```
1 import math  
2 print(math.log(10))
```

```
1 2.302585092994046
```

Criar um programa para calcular $\log_{10} 1000$.

```
1 import math  
2 print(math.log10(1000))
```

```
1 3
```

Operadores relacionais e de igualdade

Operadores relacionais e de igualdade

- Permite a um programa realizar uma ação alternativa, a partir de um resultado **verdadeiro** ou **falso** produzido por uma condição.
- As **condições** são formadas utilizando-se os operadores de igualdade e os operadores relacionais.

Operadores relacionais e de igualdade

Operador	Na matemática	Operação
>	$>$	Maior
>=	\geq	Maior ou igual
<	$<$	Menor
<=	\leq	Menor ou igual
==	$=$	Igual
!=	\neq	Diferente

- Os operadores se **associam** da esquerda para a direita.

Erro comum

- Cuidado para não confundir **atribuição** e **igualdade**
- Em Python, `==` não é a mesma coisa que `=`
 - `=` é o operador de **atribuição**
 - `==` é o operador de **igualdade**

Exemplos:

```
1 x = 30
2 y = 20
3 print(f"x == y resulta em: {x == y}")
4 print(f"x > y resulta em: {x > y}")
5 print(f"x >= y resulta em: {x >= y}")
6 print(f"x < y resulta em: {x < y}")
7 print(f"x <= y resulta em: {x <= y}")
```

```
1 x == y resulta em: False
2 x > y resulta em: True
3 x >= y resulta em: True
4 x < y resulta em: False
5 x <= y resulta em: False
```

- Note que estas expressões retornam **True** ou **False**
- O que `type(10 == 20)` retornará? `<class 'bool'>`

Operadores lógicos

Operadores lógicos

No século 18, George Boole, matemático e filósofo britânico, desenvolveu um sistema algébrico lógico, que passou a ser conhecido como **Álgebra de Boole**.

- Base para a lógica dos computadores digitais modernos.
 - Expressões lógicas (expressões booleanas) retornam valor **True** ou **False**.
-

Em python, assim como outras linguagens, pode-se dizer que:

- `0` resulta em **False**
- `1` (ou qualquer valor $\neq 0$) resulta em **True**

Operadores lógicos

Utilizaremos três operadores lógicos nesta disciplina:

- `and`: operador de conjunção (**E**)
- `or`: operador de disjunção (**OU**)
- `not`: operador de negação (**NÃO**)

Operador **and**:

`0 and 0` → `0`

`0 and 1` → `0`

`1 and 0` → `0`

`1 and 1` → `1`

Operador **or**:

`0 or 0` → `0`

`0 or 1` → `1`

`1 or 0` → `1`

`1 or 1` → `1`

Operador **not**:

`not 0` → `1`

`not 1` → `0`

Tabela verdade

A	B	A and B	A or B	not A	not B
False	False	False	False	True	True
False	True	False	True	True	False
True	False	False	True	False	True
True	True	True	True	False	False

Ou... se preferirem usar **0 e 1**:

A	B	A and B	A or B	not A	not B
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

Precedência de operadores

Operador(es)	Descrição
()	Parênteses
**	Potência
*, /, %	Multiplicação, divisão, resto
+, -	Adição, subtração
<, <=, >, >=, !=, ==	Comparações
not	NÃO
and	E
or	OU

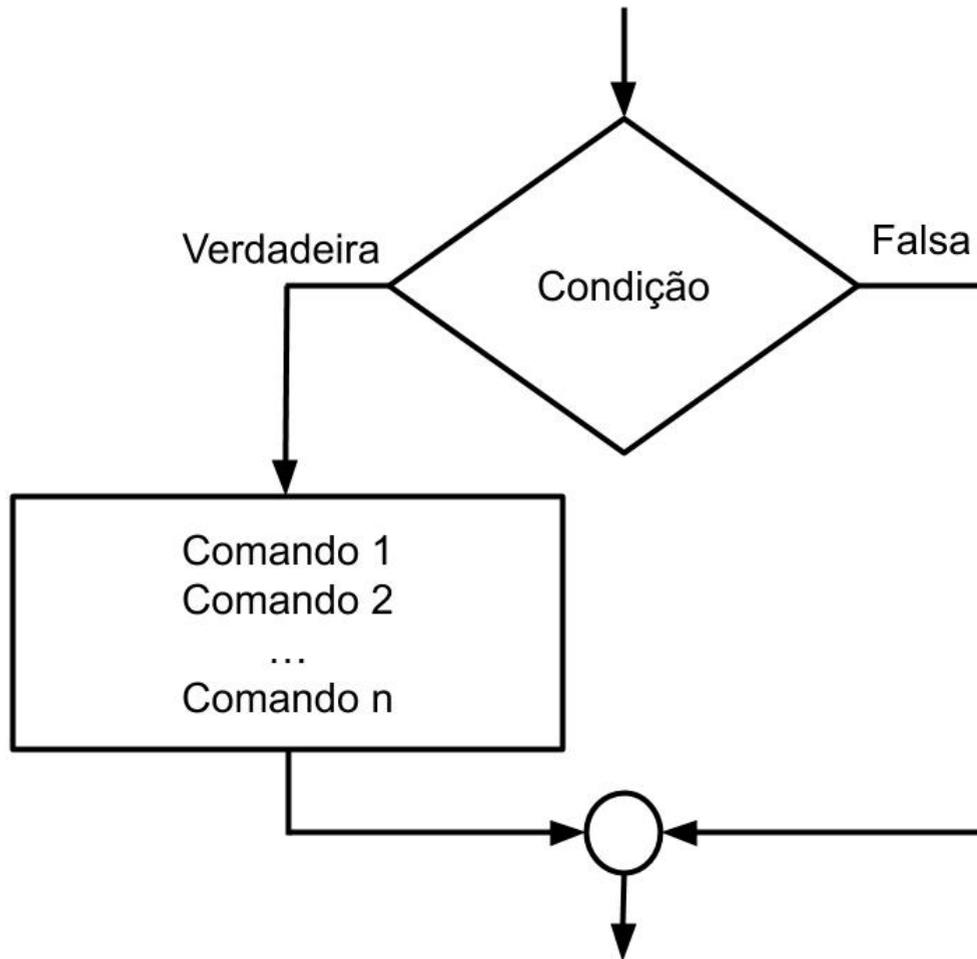
Exemplos:

True ou **False**? **1** ou **0**?

- `10 >= 10`
- `10 != 2 * 5`
- `10 <= 100 and 100 <= 1000 and 1000 <= 10 ** 10`
- `100 <= 20 * 3 + 18`
- `2000 % 2 == 0`
- `99 % 2 == 0`

0 comando if

O comando **if**



O comando `if`

Para utilizar o comando `if`, antes precisamos entender como funciona a **indentação** em Python.

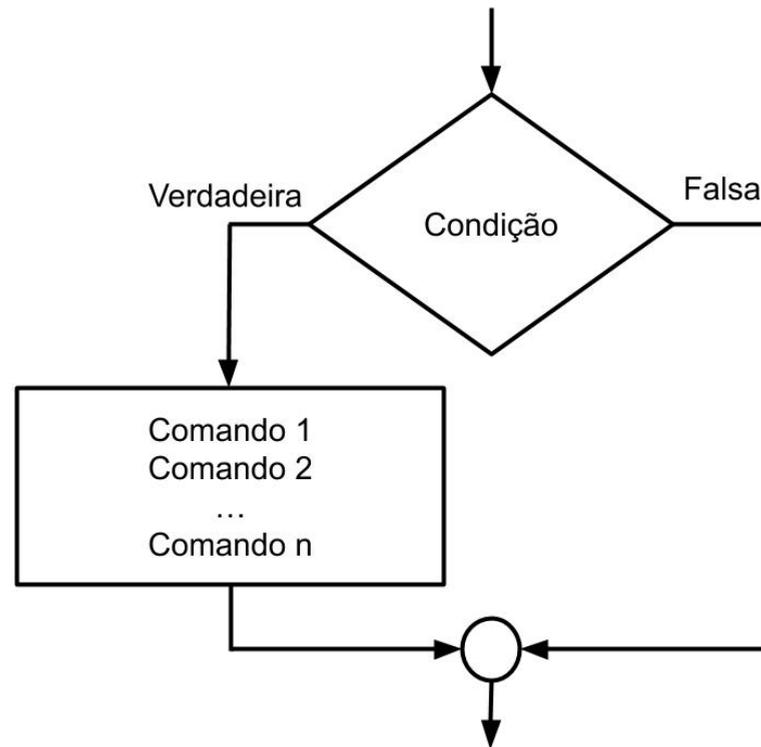
- A **indentação** é caracterizada pelos espaços em branco antes do início de uma linha de código.
- Python utiliza **indentação** para determinar se um comando está dentro ou fora de um **bloco**.
- Assim, tome muito cuidado com os espaços em branco!

O comando **if**

O código a seguir implementa o fluxograma ao lado:

```
1 ...  
2 if <Condição>:  
3     <Comando 1>  
4     <Comando 2>  
5     ...  
6     <Comando n>  
7 ...
```

Observe a indentação
(espaços no início da linha
de código)



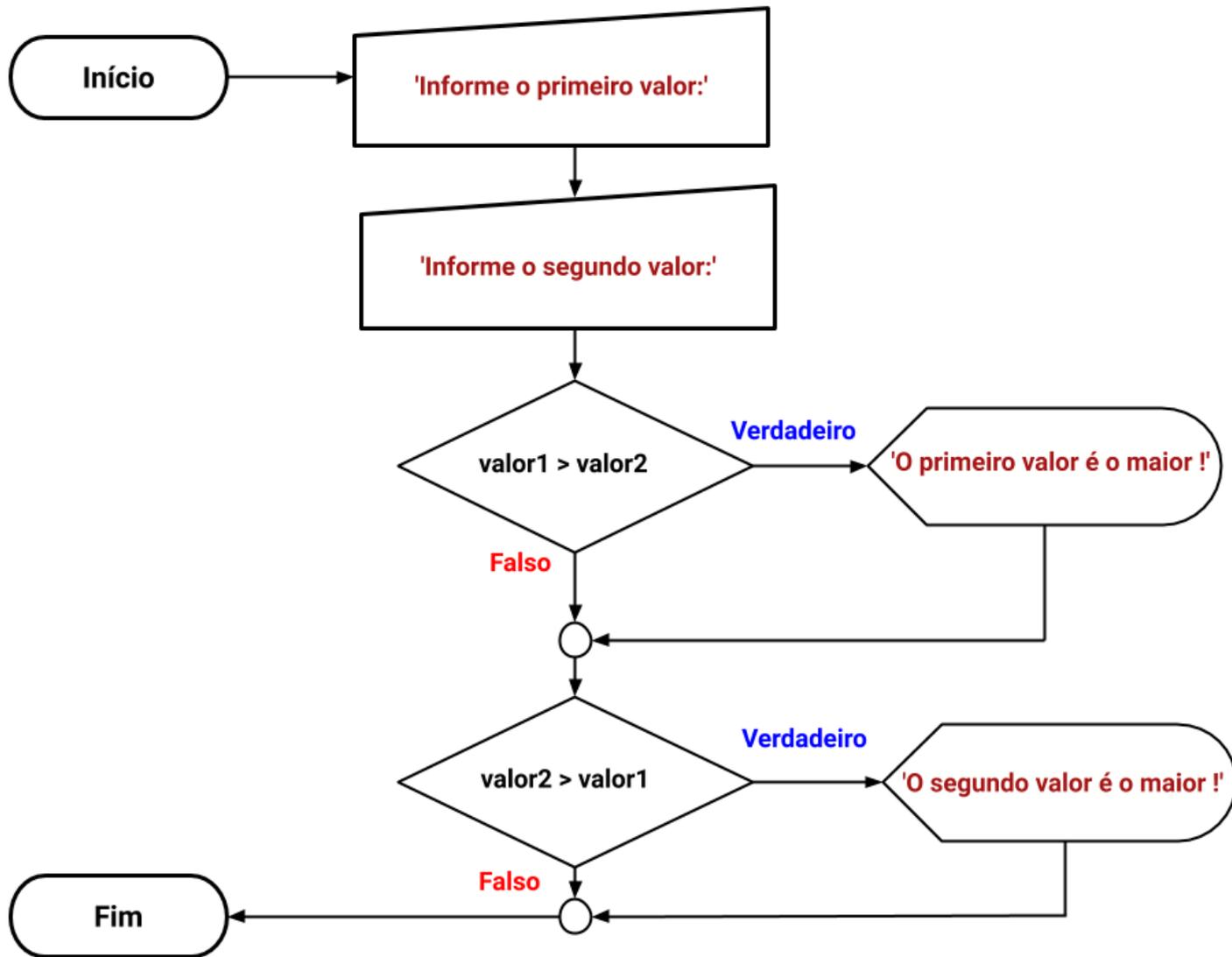
Exemplos de uso (1)

Crie um programa que lê dois números e indica qual é o maior.

```
1  ## Entrada
2  valor1 = int(input('Informe o primeiro valor: '))
3  valor2 = int(input('Informe o segundo valor: '))
4
5  ## Processamento e Saída
6  if valor1 > valor2:
7      print(f'O primeiro valor ({valor1}) é o maior!')
8  if valor2 > valor1:
9      print(f'O segundo valor ({valor2}) é o maior!')
```

Exemplos de saída: (texto digitado pelo usuário em destaque)

```
1  Informe o primeiro valor: 50
2  Informe o segundo valor: 10
3  O primeiro valor (50) é o maior!
```



Exemplos de uso (2)

Crie um programa que indica se um número lido é par ou ímpar.

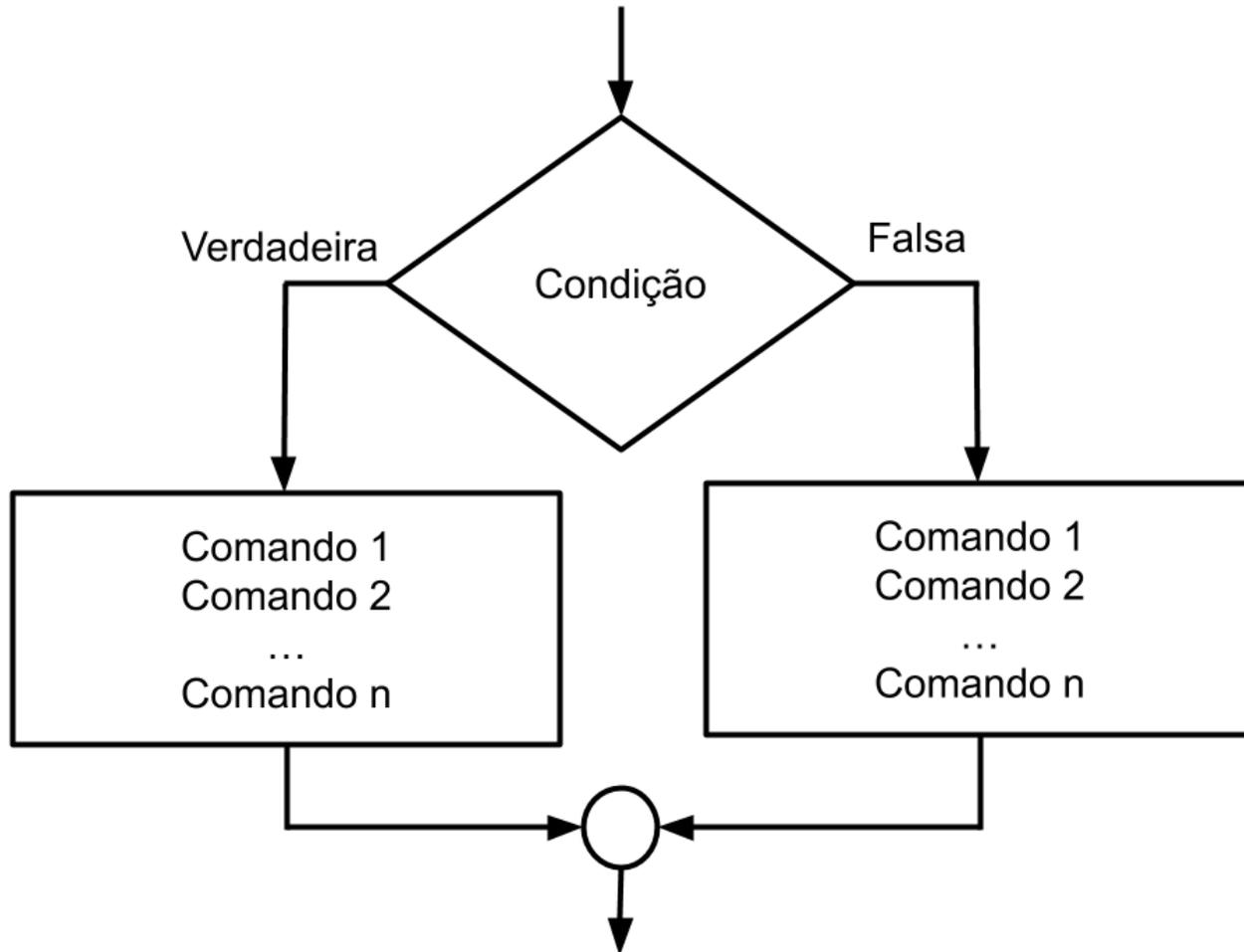
```
1  ## Entrada
2  nro = int(input("Digite um número inteiro: "))
3
4  ## Verificando se o número é par
5  if nro % 2 == 0:
6      print(f"{nro} é um número par!")
7
8  ## Verificando se o número é ímpar
9  if nro % 2 == 1:
10     print(f"{nro} é um número ímpar!")
```

Exemplo de saída: (texto digitado pelo usuário em destaque)

```
1  Digite um número inteiro: 55
2  55 é um número ímpar!
```

0 comando if-else

O comando **if-else**

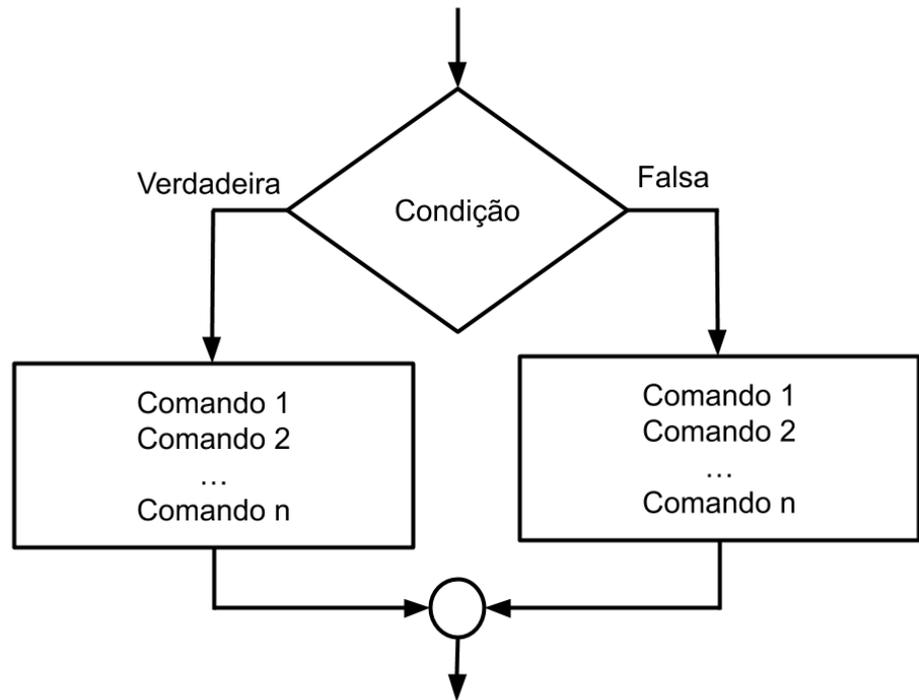


O comando **if-else**

O código a seguir implementa o fluxograma ao lado:

```
1  ...  
2  if <Condição>:  
3      <Comando 1>  
4      <Comando 2>  
5      ...  
6      <Comando n>  
7  else:  
8      <Comando 1>  
9      <Comando 2>  
10     ...  
11     <Comando n>  
12     ...
```

Note a indentação
mais uma vez!



Exemplos de uso (1)

Crie um programa que indica se um número lido é par ou ímpar.

```
1  ## Entrada
2  nro = int(input("Digite um número inteiro: "))
3
4  ## Verificando se o número é par
5  if nro % 2 == 0:
6      print(f"{nro} é um número par!")
7
8  ## Verificando se o número é ímpar
9  else:
10     print(f"{nro} é um número ímpar!")
```

Exemplo de saída: (texto digitado pelo usuário em destaque)

```
1  Digite um número inteiro: 55
2  55 é um número ímpar!
```

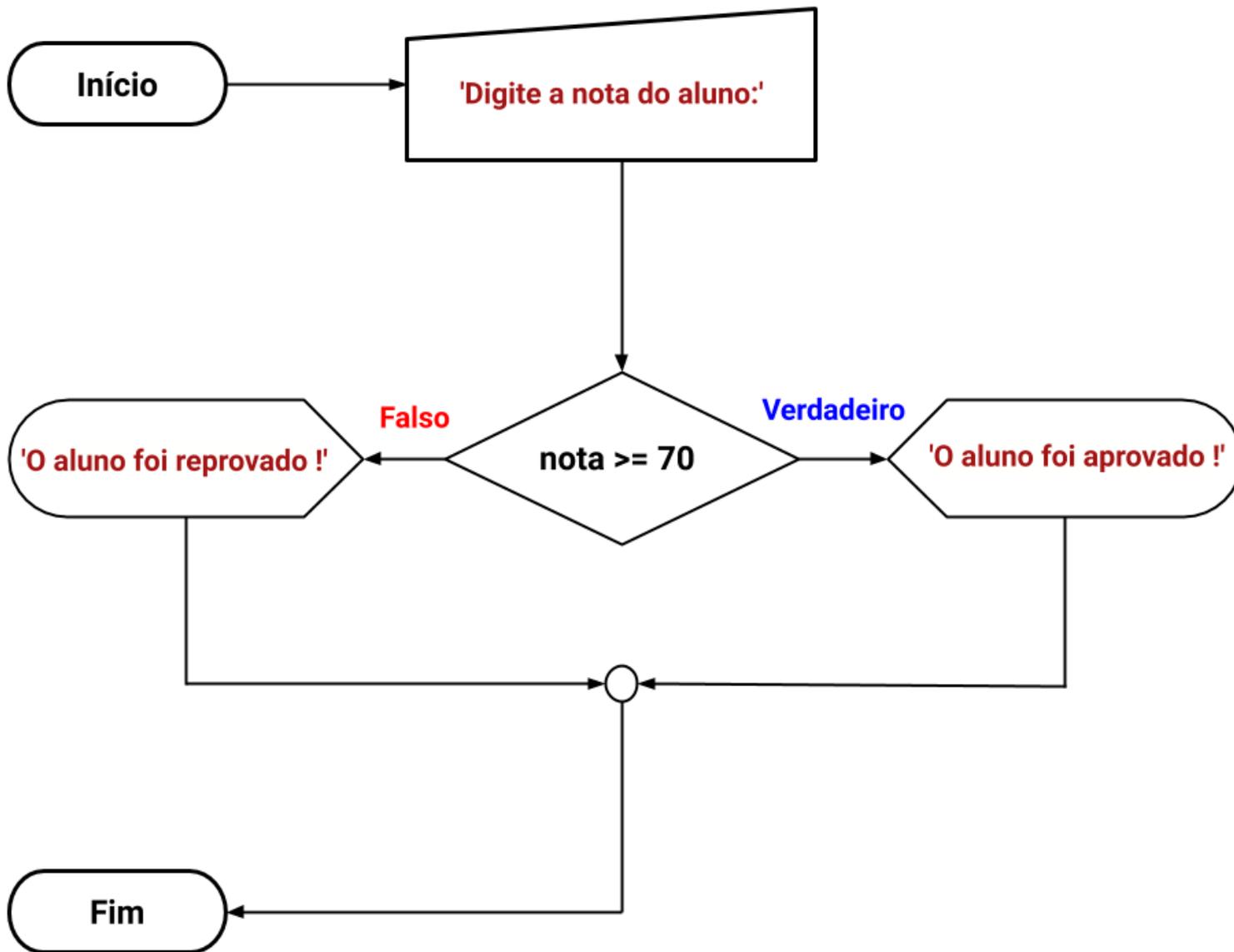
Exemplos de uso (2)

Crie um programa que lê a nota de um aluno e indica se o aluno foi aprovado ou reprovado.

- Considere que a nota mínima para aprovação é 70.

```
1 ## Entrada
2 nota = float(input('Digite a nota do aluno: '))
3
4 ## Processamento e Saída
5 if nota >= 70:
6     print('O aluno foi aprovado !')
7 else:
8     print('O aluno foi reprovado !')
```

```
1 Digite a nota do aluno: 65
2 O aluno foi reprovado !
```



Exemplos de uso (3)

Elabore um programa que calcule as raízes de uma equação do 2º grau qualquer através da fórmula de Bhaskara:

- Formato: $ax^2 + bx + c = 0$, sendo $a \neq 0$;
- Se $a = 0$, informar que não é uma equação do 2º grau;
- Caso contrário, calcular Δ e as raízes da equação:
 - $\Delta = b^2 - 4ac$,
 - $x_1 = \frac{-b + \sqrt{\Delta}}{2a}$ e $x_2 = \frac{-b - \sqrt{\Delta}}{2a}$.

Exemplos de uso (3)

```
1 import math
2
3 print('Cálculo das raízes de uma equação do 2o grau')
4 a = float(input('Digite o coeficiente A: '))
5 b = float(input('Digite o coeficiente B: '))
6 c = float(input('Digite o coeficiente C: '))
7
8 if a == 0:
9     print('Erro: Coeficiente a == 0 !')
10 else:
11     delta = b*b - 4*a*c
12     if delta < 0:
13         print('Erro: Delta < 0 !')
14     else:
15         x1 = (-b + math.sqrt(delta)) / (2*a)
16         x2 = (-b - math.sqrt(delta)) / (2*a)
17         print(f'x1 = {x1:.2f}')
18         print(f'x2 = {x2:.2f}')
```

Exemplos de uso (3)

Exemplos de saída: (texto digitado pelo usuário em destaque)

```
1 Cálculo das raízes de uma equação do 2o grau
2 Digite o coeficiente A: 10
3 Digite o coeficiente B: 20
4 Digite o coeficiente C: 5
5 x1 = -0.29
6 x2 = -1.71
```

```
1 Cálculo das raízes de uma equação do 2o grau
2 Digite o coeficiente A: 1
3 Digite o coeficiente B: 2
4 Digite o coeficiente C: 3
5 Erro: Delta < 0 !
```

Exemplos e exercícios

Exemplo 1

Faça um programa que lê um número e informa se o número é inteiro ou decimal.

```
1 import math
2
3 # Entrada
4 n = float(input('Digite um número: '))
5
6 # Processamento e saída
7 if n == math.trunc(n):
8     print('Número é inteiro!')
9 else:
10    print('Número é decimal!')
```

Exemplo 2

Desenvolva um programa que informa quantos pontos o time que jogou em casa ganhou considerando o resultado de um jogo de futebol. Para isso, o programa deve ler o número de gols marcados pelo time da casa (mandante) e o número de gols marcados pelo adversário (visitante).

Note que o time da casa ganha 3 pontos em caso de vitória, 1 ponto em caso de empate, e nenhum ponto em caso de derrota.

Exemplo 2

```
1 # Entrada
2 gols_pro = int(input('Informe o número de gols do mandante: '))
3 gols_contra = int(input('Informe o número de gols do visitante: '))
4
5 # Processamento e saída
6 if gols_pro > gols_contra:
7     print('O time da casa ganhou 3 pontos.')
8 else:
9     if gols_pro == gols_contra:
10        print('O time da casa ganhou 1 ponto.')
11    else:
12        print('O time da casa não ganhou nenhum ponto.')
```

Exemplo 3

Elabore um programa que leia três valores inteiros x , y e z e verifique se estes valores podem constituir os comprimentos de um triângulo e se caso positivo informe se este é um triângulo equilátero, isósceles ou escaleno.

Observações:

- O triângulo equilátero possui os três lados com medidas iguais;
- O triângulo isósceles possui dois lados com medidas iguais;
- O triângulo escaleno possui os três lados com medidas diferentes.
- Condição de desigualdade triangular:
 - $(x < y + z)$ and $(y < x + z)$ and $(z < x + y)$

Exemplo 3

```
1 # Entrada
2 x = int(input('Digite o valor de x: '))
3 y = int(input('Digite o valor de y: '))
4 z = int(input('Digite o valor de z: '))
5
6 # Processamento e Saída
7 if (x < y + z) and (y < x + z) and (z < x + y):
8     if (x == y) and (y == z):
9         print('Triângulo Equilátero')
10    else:
11        if (x == y) or (y == z) or (x == z):
12            print('Triângulo Isósceles')
13        else:
14            print('Triângulo Escaleno')
15 else:
16    print('Erro: os valores informados não constituem um triângulo.')
```

Perguntas?
