

Lógica de Programação

Estrutura condicional com Python

Prof.: Caio Malheiros

caio.duarte@sp.senai.br

Roteiro

- Operadores e tipos de operadores
 - Operadores aritméticos
 - Operadores relacionais
 - Operadores lógicos
- Estrutura condicional

Operadores e Tipos de Operadores

- Os operadores são meios pelo qual incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador. Temos três tipos de operadores:
- **Operadores Aritméticos**
- **Operadores Relacionais**
- **Operadores Lógicos**

Operadores e Tipos de Operadores

- **Operadores aritméticos** são os utilizados para obter resultados numéricos. Além da adição, subtração, multiplicação e divisão, podem utilizar também o operador para exponenciação.
- Os símbolos para os operadores aritméticos são:

Operador	Expressão, em que a = 4 e b = 8
Adição: +	a + b >>> 12
Subtração: -	a - b >>> -4
Multiplicação: *	a * b >>> 32
Divisão: /	b / a >>> 2
Expoente: **	a ** 2 >>> 16

Operadores e Tipos de Operadores

- **Operadores relacionais:**
- Os operadores relacionais são utilizados para **comparar valores (caracteres e números)**. Os valores a serem comparados podem ser caracteres ou variáveis.

Operador	Descrição	Exemplo	Resultado
`==`	Igualdade	`5 == 5`	`True`
`!=`	Diferente	`5 != 3`	`True`
`>`	Maior que	`5 > 3`	`True`
`<`	Menor que	`3 < 5`	`True`
`>=`	Maior ou igual a	`5 >= 5`	`True`
`<=`	Menor ou igual a	`3 <= 5`	`True`

Operadores e Tipos de Operadores

- Tendo duas variáveis **A = 5** e **B = 3**
- Os resultados das expressões seriam:

Expressão	Resultado
A = B	Falso
A <> B	Verdadeiro
A > B	Verdadeiro
A < B	Falso
A >= B	Verdadeiro
A <= B	Falso

Operadores e Tipos de Operadores

- **Operadores lógicos:** Os operadores lógicos servem para **combinar resultados de expressões**, retornando se o resultado final é verdadeiro ou falso.

Linguagem Python	Algoritmos
and	E
or	OU
not	NÃO

Operadores e Tipos de Operadores

- **Operadores lógicos:** Servem para combinar resultados de expressões, retornando se o resultado final é verdadeiro ou falso. Lembrando que em Linguagem Python são representados pelos valores **True** (verdadeiro) e **False** (falso).

Linguagem Python	Algoritmos
and	E
or	OU
not	NÃO

Operadores e Tipos de Operadores

- **and (e)** – O resultado será verdadeiro, somente se todas as condições forem verdadeiras.
- **or (ou)** - O resultado será falso, somente se todas as condições forem falsas.
- **not (não)** - O resultado será a negação do valor da expressão ou condição, ou seja, a negação de verdadeiro resulta em falso e vice-versa

Operadores e Tipos de Operadores

- Tabela verdade

1º Valor	Operador	2º Valor	Resultado
True	and	True	True
True	and	False	False
False	and	True	False
False	and	False	False
True	or	True	True
True	or	False	True
False	or	True	True
False	or	False	False
True	not	-----	False
False	not	-----	True

Operadores e Tipos de Operadores

- Exemplo

```
A = 15
B = 90
C = 29
print('A == B and B > C : ', A == B and B > C)
print('A != B or B < C : ', A != B or B < C)
print('not (A > B) : ', not (A > B))
print('A < B and B > C: ', A < B and B > C)
print('A >= B or B == C :', A >= B or B == C)
print('not (A <= B) :', not (A <= B))
```

```
C:\PythonCurso\venv\Scripts\python.exe
A == B and B > C : False
A != B or B < C : True
not (A > B) : True
A < B and B > C: True
A >= B or B == C : False
not (A <= B) : False

Process finished with exit code 0
```

Estrutura de decisão

- Esta estrutura de controle, podemos definir como sendo o primeiro divisor de águas no mundo da programação, pois literalmente no momento da programação;
- É utilizando esta estrutura que entra uma palavra mágica chamada “**se**”, que dependendo do resultado da operação que for realizada, executa algumas instruções, “**senão**” executa outras instruções.



Estrutura de decisão

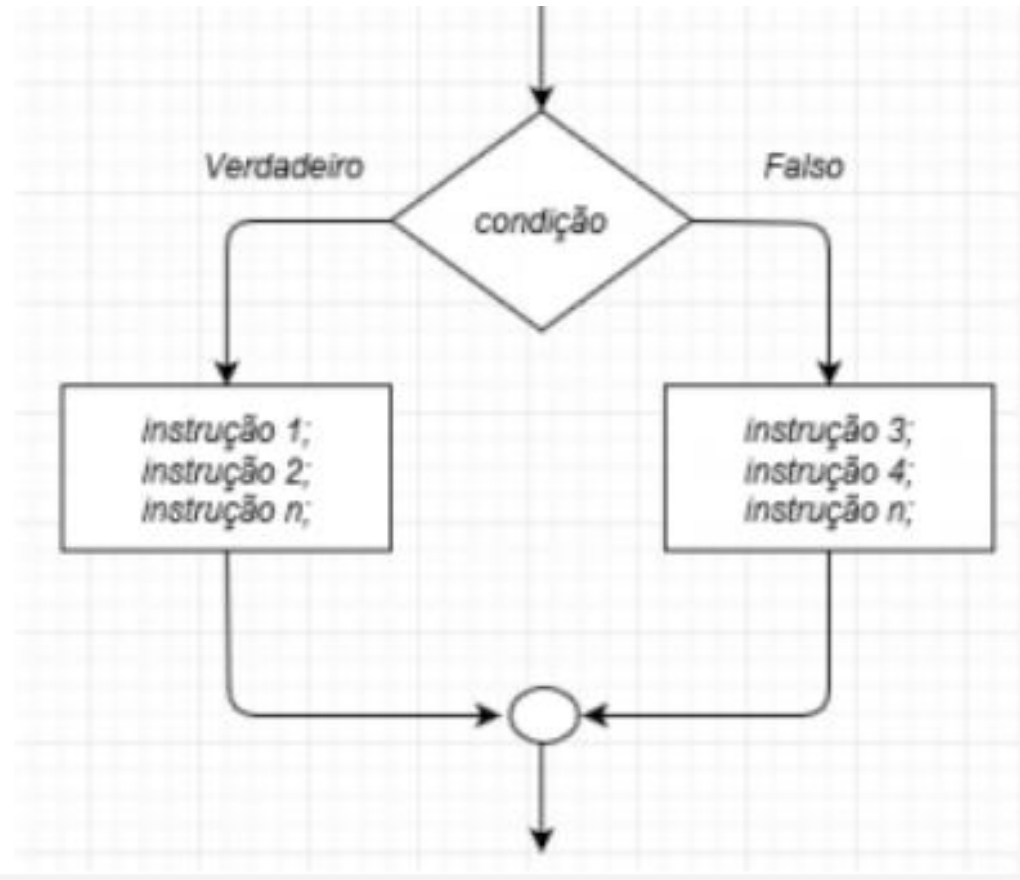
- As condições que são testadas em um Algoritmo resultam em valores do tipo lógico (booleano), portanto podem ter somente dois resultados:

Verdadeiro ou Falso

- Assim, a seleção de instruções pode seguir no máximo duas alternativas: uma se a condição for verdadeira e outra se a condição testada for falso.

Estrutura de decisão

```
se (condição) então  
    instrução 1;  
    instrução 2;  
    instrução n;  
senão  
    instrução 3;  
    instrução 4;  
    instrução n;  
fimse
```



Estrutura de decisão simples (SE)

- O **se** é uma estrutura de condição que permite avaliar uma expressão e, de acordo com seu resultado, executar uma determinada ação.
- No código a seguir temos um exemplo de uso do se, verificando se a variável idade é maior ou igual que 18. Em caso positivo, será impressa uma mensagem na tela e em caso negativo, neste caso não será realizada nenhuma instrução

Estrutura de decisão simples (SE)

- O **se** é uma estrutura de condição que permite avaliar uma expressão e, de acordo com seu resultado, executar uma determinada ação.
- No código a seguir temos um exemplo de uso do se, verificando **se a variável idade é maior ou igual que 18**. Em caso **positivo**, será **impressa** uma **mensagem** na tela e em caso **negativo**, neste caso não **será realizada nenhuma instrução**

Estrutura de decisão simples (SE)

- Exemplo
- Um exemplo desta situação seria aplicarmos um cálculo representando **divisão de 20 pedaços** de um **bolo entre algumas pessoas**, sendo que o número de pessoas, deverá ser informado pelo usuário. A condição para esta divisão é que o número de pessoas seja maior que 0 (zero).

Estrutura de decisão simples (SE)

```
qtd=int(input(' Digite a qtd de pessoas: '))
if qtd>0:
    qtd_bolos=20/qtd
    print("Cada pessoa vai receber ", qtd_bolos)
input("Pressione <enter> para continuar")
```

Dúvidas?
Ótimo dia para todos!