



# CK0211 - Fundamentos de Programação: Estrutura Condicional

Emanuele Santos

Bibliografia: Ascencio, Cap. 4

# Objetivos

- Apresentar as Estruturas Condicionais Simples e Composta em Algoritmos
- Apresentar as Estruturas Condicionais Simples e Composta em Python
- Explicar as expressões lógicas, operadores relacionais e lógicos e saber avaliar essas expressões



# ESTRUTURA CONDICIONAL EM ALGORITMOS

# Estrutura Condicional

- Simples
- Composta

# Estrutura Condicional Simples

**SE** condição  
**ENTÃO** comando

- O **comando** só será executado se a condição for **verdadeira**
- Uma condição é uma comparação ou uma expressão lógica que possui dois valores possíveis: **verdadeiro** ou **falso**

# Estrutura Condicional Simples

```
SE condição
ENTÃO INÍCIO
    comando1
    comando2
    comando3
FIM
```

- Os comandos 1, 2 e 3 só serão executados se a condição for **verdadeira**
- As palavras **INÍCIO** e **FIM** serão necessárias apenas quando 2 ou mais comandos forem executados

# Estrutura Condicional Composta

```
SE condição  
ENTÃO comando1  
SENÃO comando2
```

- Se a **condição** for verdadeira, será executado o comando1; caso contrário, será executado o comando 2

# Estrutura Condicional Composta

```
SE condição
ENTÃO INÍCIO
    comando1
    comando2
FIM
SENÃO INÍCIO
    comando3
    comando4
FIM
```

- Se a **condição** for verdadeira, serão executados os comandos 1 e 2; caso contrário, serão executados os comandos 3 e 4

# Exemplo

- Faça um algoritmo que receba duas notas e calcule e mostre a média aritmética e diga se o aluno foi aprovado (média superior ou igual a 7) ou reprovado

```
ALGORITMO  
DECLARE n1, n2, media NUMÉRICO  
LEIA n1  
LEIA n2  
media ← (n1+n2)/2  
ESCREVA "A média é ", media  
SE media >= 7  
ENTÃO ESCREVA "APROVADO"  
SENÃO ESCREVA "REPROVADO"  
FIM_ALGORITMO.
```



# ESTRUTURA CONDICIONAL EM PYTHON

# Estrutura Condicional Simples

não esquecer os dois pontos

```
if condição :  
    comando1
```

4 espaços delimitando um novo bloco

- Onde:
  - condição é uma expressão lógica que pode retornar verdadeiro ou falso
  - comando1 é um comando qualquer em Python

# Estrutura Condicional Simples

```
if condição :  
    comando1  
    comando2  
    comando3  
comando4
```

- Os comandos 1, 2 e 3 só serão executados se a condição for **verdadeira**
- O comando 4 será executado sempre, pois não está dentro do bloco do if

# Estrutura Condicional Composta

```
if condição :  
    comando1  
    comando2  
else:  
    comando3  
    comando4
```

- Se a **condição** for verdadeira, serão executados os comandos 1 e 2; caso contrário, serão executados os comandos 3 e 4

# Estrutura Condicional Composta

```
if condição1 :  
    comando1  
    comando2  
elif condição2 :  
    comando3  
    comando4  
else:  
    comando5
```

- Se a **condição1** for verdadeira, serão executados os comandos 1 e 2; caso contrário, será verificada a **condição2** e se ela for verdadeira, serão executados os comandos 3 e 4; caso contrário, será executado o comando 5

# Estrutura Condicional Composta

```
if condição1 :  
    comando1  
    comando2  
elif condição2 :  
    comando3  
    comando4  
else:  
    comando5
```

```
if condição1 :  
    comando1  
    comando2  
if condição2 :  
    comando3  
    comando4  
else:  
    comando5
```

- Os trechos de código acima são equivalentes?
- O que vai acontecer em cada caso se ambas as condições 1 e 2 forem verdadeiras?

# Expressões Aritméticas e Lógicas

- Expressões aritméticas são expressões formadas por operadores aritméticos e os operandos são variáveis numéricas ou outras expressões aritméticas
- Expressões lógicas são expressões cujos operadores são lógicos ou relacionais e os operandos são relações ou variáveis do tipo lógico

# Operadores aritméticos em Python

Operador	Exemplo	Comentário
=	<code>x = y</code>	O conteúdo da variável <code>y</code> é atribuído à variável <code>x</code>
+	<code>x + y</code>	Soma o conteúdo de <code>x</code> e de <code>y</code>
-	<code>x - y</code>	Subtrai o conteúdo de <code>y</code> do conteúdo de <code>x</code>
*	<code>x * y</code>	Multiplica o conteúdo de <code>x</code> por pelo conteúdo de <code>y</code>
/	<code>x / y</code>	Divide o conteúdo de <code>x</code> pelo conteúdo de <code>y</code> (Python v3)
%	<code>x % y</code>	Obtém o resto da divisão de <code>x</code> por <code>y</code> (inteiros)
//	<code>x // y</code>	Obtém o quociente inteiro da divisão de <code>x</code> por <code>y</code> (inteiros)
**	<code>x ** y</code>	Eleva o conteúdo de <code>x</code> à potência do conteúdo de <code>y</code>

# Prioridades

- Precedência entre operadores aritméticos

Prioridade	Operadores
1ª	parênteses mais internos
2ª	**
3ª	* / // %
4ª	+ -

**Em caso de empate (operadores de mesma prioridade), resolve-se da esquerda para a direita**

# Expressões aritméticas

- Exemplos

a)  $5 + 9 + 7 + 8/4$

$$5 + 9 + 7 + 2$$

$$23$$

b)  $1 - 4 * 3/6 - 3^{**}2$

$$1 - 4 * 3/6 - 9$$

$$1 - 12/6 - 9$$

$$1 - 2 - 9$$

$$-10$$

# Exercícios

- Supondo que A, B e C são variáveis de tipo int, com valores iguais a 5, 10 e -8, respectivamente, e uma variável float D, com valor 1.5, quais os resultados das expressões aritméticas a seguir:
  - a)  $2 * A \% 3 - C$
  - b)  $((20 // 3) // 3) + 8^{**}2/2$
  - c)  $(30 \% 4 * 3^{**}3)^{-1}$
  - d)  $-C^{**}2 + (D * 10)/A$

# Operadores Relacionais

- Utilizados em comparações entre dois **valores** de mesmo tipo primitivo
- Valores: variáveis ou expressões aritméticas

Operador	Função	Exemplo
==	igual a	$X == Y, Z == 3$
>	maior que	$X > Y, X > 18$
<	menor que	$X < Y, X < 10$
>=	maior ou igual a	$X >= Y, X <= 32$
<=	menor ou igual a	$X <= Y, X <= 32$
!=	diferente de	$X != Y, X != 56$

# Operadores Relacionais

- Resultado é sempre um valor lógico
- Exemplo
  - $A + B == C$
  - O resultado pode ser Verdadeiro (True) ou Falso (False), dependendo dos valores de A, B e C

```
>>> A = 3
>>> B = 5
>>> C = 8
>>> A + B == C
True
>>> A = 4
>>> A + B == C
False
```

# Operadores Relacionais

- Exemplos
  - $3 * 6 == 36/2$
  - $15 \% 4 < 19 \% 6$
  - $3 * 5 // 4 <= 3**2 / 0.5$

# Operadores Lógicos

- Três operadores lógicos

Operador	Função	Exemplo
<b>not</b>	não (negação)	<b>not</b> X
<b>or</b>	ou (disjunção)	X > Y <b>or</b> X > Z
<b>and</b>	e (conjunção)	X > Y <b>and</b> X < Z

# Tabelas-Verdade

- Conjunto de todas as possibilidades combinatórias entre os valores de diversas variáveis lógicas e um conjunto de operadores lógicos

Operação de negação	
A	not A
F	V
V	F

Operação de conjunção e disjunção			
A	B	A and B	A or B
F	F	F	F
F	V	F	V
V	F	F	V
V	V	V	V

# Tabelas-verdade

- Exemplo
  - Se chover **e** relampejar, eu fico em casa.  
Quando eu fico em casa?
  - Quando os termos chover e relampejar forem simultaneamente verdade
  - Se chover ou relampejar, eu fico em casa.  
Quando eu fico em casa?
  - Somente chovendo, somente relampejando ou chovendo e relampejando.

# Prioridade dos operadores lógicos

Prioridade	Operadores
1 <sup>a</sup>	<b>not</b>
2 <sup>a</sup>	<b>and</b>
3 <sup>a</sup>	<b>or</b>

# Prioridade de todos os operadores

Prioridade	Operadores
1 <sup>a</sup>	parênteses mais internos
2 <sup>a</sup>	operadores aritméticos
3 <sup>a</sup>	operadores relacionais
4 <sup>a</sup>	operadores lógicos

# Exemplos

- $2 < 5$  **and**  $15/3 == 5$
- $\text{False}$  **or**  $20 //(18/3) != (21/3) // 2$
- **not**  $\text{True}$  **or**  $3^{**}2/3 < 15 - 35 \% 7$
- **not**  $(5 != 10/2)$  **or**  $\text{True}$  **and**  $2 - 5 > 5 - 2$  **or**  $\text{True}$
- $2^{**}4 != 4 + 2$  **or**  $2 + 3 * 5/3 \% 5 < 0$

# Exercícios

- Determine os resultados das expressões lógicas a seguir, sabendo que  $A = 2$ ,  $B = 7$ ,  $C = 3.5$ ,  $L = \text{False}$ :
  - a)  $B == A * C$  **and**  $(L \text{ or } \text{True})$
  - b)  $B > A$  **or**  $B == A ** A$
  - c)  $L$  **and**  $B \% A >= C$  **or not**  $A <= C$
  - d)  $B / A == C$  **or**  $B / A != C$
  - e)  $L$  **or**  $B ** A <= C * 10 + A * B$

# Exemplo

- Faça um programa que receba duas notas e calcule e mostre a média aritmética e diga se o aluno foi aprovado (média superior ou igual a 7) ou reprovado

```
n1 = float(input("Entre com a nota 1: "))
n2 = float(input("Entre com a nota 2: "))
media = (n1+n2)/2
print("A média é: ", media)
if media >= 7:
    print("Aprovado")
else:
    print("Reprovado")
```

# Exercícios

- Escreva o algoritmo em pseudocódigo e o programa em Python para cada um dos exercícios a seguir

# Programa 1

- A nota final de um estudante é calculada a partir de três notas atribuídas e os respectivos pesos conforme a tabela abaixo:

NOTA	PESO
Trabalho de laboratório	2
Avaliação semestral	3
Exame final	5

- Faça um programa que receba as três notas, calcule e mostre a média ponderada e o conceito que segue a tabela:

MÉDIA PONDERADA	CONCEITO
8.0 —●— 10.0	A
7.0 —●— 8.0	B
6.0 —●— 7.0	C
5.0 —●— 6.0	D
0.0 —●— 5.0	E

# Programa 2

- Faça um programa para calcular a média aritmética  $M$  entre duas notas de um aluno e mostrar sua situação, que pode ser aprovado ( $M \geq 7$ ), reprovado ( $M < 4$ ) e AF ( $4 \leq M < 7$ ). Se o aluno ficar de AF, entre com a nota da AF e mostre a média e o resultado final

# Programas 3 e 4

- Faça um programa que receba dois números e mostre o maior deles
- Faça um programa que receba três números e mostre-os em ordem crescente. Suponha que o usuário digitará três números diferentes