

Programação de Computadores

- OPERADORES RELACIONAIS E LÓGICOS
- DECISÃO SIMPLES

Na aula anterior...

| Operador | Operação | Prioridade |
|----------|------------------------------|------------|
| + | Soma | 4º |
| - | Subtração | 4º |
| * | Multiplicação | 3º |
| / | Divisão | 3º |
| // | Parte Inteira | 3º |
| % | Resto de uma divisão inteira | 2º |
| + | Manutenção de sinal | 1º |
| - | Inversão de sinal | 1º |

Observações:

- Em uma expressão com operadores da mesma prioridade, as operações serão executadas de esquerda a direita.
- Nas linguagens com operadores para potência, eles teriam prioridade maior que + - / *

Na aula anterior...

Vamos criar um programa que solicite ao usuário um número inteiro com três dígitos e exiba esse número com os dígitos invertidos.

```
main.py
1 num = int(input("Digite um número com três dígitos: "))
2 d1 = num // 100
3 d2 = num % 100 // 10
4 d3 = num % 10
5 inverso = d3*100+d2*10+d1
6 print("O inverso do número digitado é", inverso)
```



Conceitos abordados nesta aula

- ⦿ A proposta desta aula é apresentar para vocês as Estruturas condicionais.
- ⦿ Veremos, portanto, os operadores lógicos e as estruturas condicionais if e if-else.



<https://vidaprogramador.com.br/>



Introdução às decisões

Tomadas de decisão são importantes na programação, na atividade profissional, na vida...



www.peytonbolin.com



<https://www.sbcoaching.com.br/>



<https://consultormarketing.digital/>



Fique ligado!

Na programação, as decisões devem estar bem definidas: não podem ser ambíguas.

Introdução às decisões

Na programação, as decisões devem estar bem definidas: **não podem ser ambíguas.**



Operadores Relacionais

| Operador | Significado | Exemplo | Resultado |
|----------|----------------|---------|------------|
| == | igual a | 5 == 5 | verdadeiro |
| | | 5 == 8 | falso |
| != | diferente de | 5 != 8 | verdadeiro |
| | | 5 != 5 | falso |
| > | maior que | 8 > 5 | verdadeiro |
| | | 5 > 8 | falso |
| < | menor que | 5 < 8 | verdadeiro |
| | | 8 < 5 | falso |
| >= | maior ou igual | 8 >= 5 | verdadeiro |
| | | 5 >= 8 | falso |
| <= | menor ou igual | 5 <= 8 | verdadeiro |
| | | 8 <= 5 | falso |

Comparações só podem ser feitas entre objetos de mesma natureza, isto é, variáveis do mesmo tipo de dado. O resultado de uma comparação será sempre um valor lógico.



Operadores Lógicos

| Operadores | Python |
|------------|--------|
| ou | or |
| e | and |
| não | not |

| A | B | A or B | A and B | not A |
|---|---|--------|---------|-------|
| F | F | F | F | V |
| F | V | V | F | V |
| V | F | V | F | F |
| V | V | V | V | F |

- ✓ or → basta que um dos seus valores seja V para que o resultado seja V.
- ✓ and → é necessário que todos os valores sejam V para que o resultado seja V.



Operadores Lógicos e Relacionais

Exemplo: Considere $A = 10$, $B = 5$, $C = 7$, $D = 3$, $F = 5$

a) $A > C \longrightarrow$ Verdadeiro

b) $A < B \longrightarrow$ Falso

c) $A == D \longrightarrow$ Falso

d) $A != C \longrightarrow$ Verdadeiro

e) $B <= F \longrightarrow$ Verdadeiro

f) $A >= D \longrightarrow$ Verdadeiro

g) $A > B$ e $C > D \longrightarrow$ Verdadeiro

h) $C > B$ ou $A == D \longrightarrow$ Verdadeiro

i) $C == B$ ou $A == D \longrightarrow$ Falso

j) $!(A == D) \longrightarrow$ Verdadeiro

k) $!(B == F) \longrightarrow$ Falso

l) $C > B$ e $A == D \longrightarrow$ Falso

Prioridade dos grupos de operadores

| Operadores | Prioridade |
|-------------|----------------|
| Lógicos | 4 ^o |
| Relacionais | 3 ^o |
| Aritméticos | 2 ^o |
| Parênteses | 1 ^o |

menor



maior

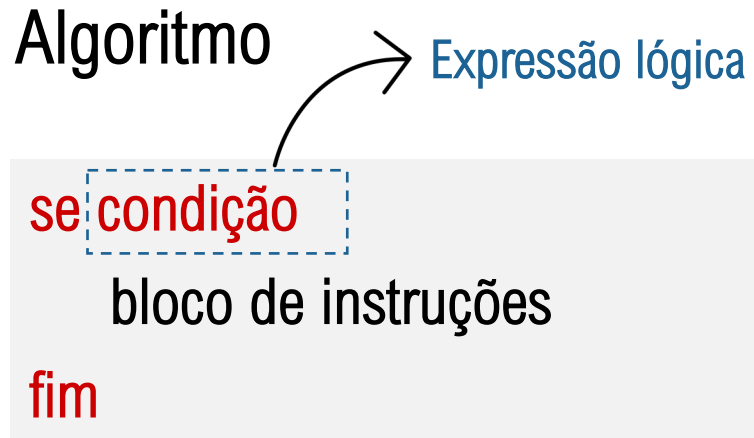
Estruturas Condicionais - if

Elas permitem avaliar condições para que um determinado bloco de código seja executado ou não.

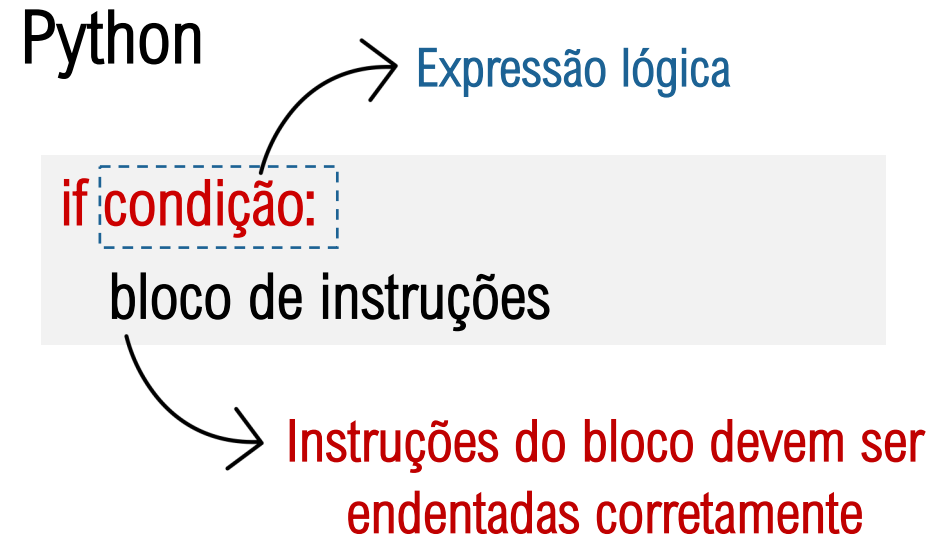
Estrutura Condicional Simples:

Nela é avaliada uma expressão lógica e se o resultado da avaliação for **verdadeiro** (*true*) o bloco de código inserido nela será executado.

Algoritmo



Python



Exemplos de aplicação

1- Crie um programa em Python que solicite ao usuário a sua idade e mostre se ele pode ter CHN .

```
algoritmo maiorDeIdade
    inicio
        inteiro idade
        escreva ("Digite a sua idade: ")
        leia (idade)
        se idade >= 18
            escreva ("Você pode ter CNH")
        fim
```



Exemplos de aplicação

1- Crie um programa em Python que solicite ao usuário a sua idade e mostre se ele pode ter CHN .

```
main.py
1  # Exemplo 01: programa que solicita a idade
2  # do usuário e mostra se ele pode ter CNH
3
4  idade = int(input("Digite a sua idade: "))
5
6  if idade >=18:
7      print("Parabéns! Você pode ter CNH")
```



Exemplos de aplicação

2- Escreva um programa em Python que solicite um número inteiro ao usuário e mostre-o caso o mesmo seja par.

```
algoritmo parOuImpar
  inicio
    inteiro num
    escreva ("Digite um número inteiro")
    leia (num)
    se (num%2==0) {
      escreva ("O número: " + num + " é par")
    }
  fim
```



Exemplos de aplicação

2- Escreva um programa em Python que solicite um número inteiro ao usuário e mostre-o caso o mesmo seja par.

main.py

```
1  # Exemplo2: programa que solicita um número
2  # inteiro ao usuário e mostre-o caso
3  # o mesmo seja par.
4
5  num = int(input("Entre com um número inteiro: "))
6  if num % 2 == 0:
7      print("O número: " , num , " é par.")
```



Exemplos de aplicação

3- Crie um programa em Python que solicite ao usuário três valores inteiros (A, B e C) e verifica se o valor armazenado em B é o menor.

```
algoritmo menorDeTrês
    inicio
        inteiro a, b, c
        escreva ("Digite o valor de a")
        leia (a)
        escreva ("Digite o valor de b")
        leia (b)
        escreva ("Digite o valor de c")
        leia (c)
        se a > b e b < c
            escreva ("O menor valor é o b")
        fim
```



Exemplos de aplicação

3- Crie um programa em Python que solicite ao usuário três valores inteiros (a, b e c) e verifica se o valor armazenado em b é o menor.

```
main.py
1  # Exemplo 03: Programa que verifica se o valor
2  # digitado para b é menor
3
4  a = int(input("Digite o valor de a: "))
5  b = int(input("Digite o valor de b: "))
6  c = int(input("Digite o valor de c: "))
7
8  if a > b and c > b:
9      print("O menor valor é o b")
```



Estruturas Condicionais – if-else

Elas permitem avaliar condições para que um determinado bloco de código seja executado ou não.

Estrutura Condicional Composta:

Nela é avaliada uma expressão lógica e se o resultado da avaliação for **verdadeiro (true)** será executado o primeiro bloco de código, caso contrário o bloco do senão.

Algoritmo

se condição

bloco de instruções V

senão

bloco de instruções F

fim

Expressão lógica



Python

if condição:

bloco de instruções V

else:

bloco de instruções F

Expressão lógica

Instruções do bloco
devem ser endentadas
corretamente



Estruturas Condicionais – if-else



Exemplos de aplicação

4- Crie um programa em Python que solicite ao usuário a sua idade e mostre se o mesmo pode ou não ter CHN.

```
algoritmo maiorDeIdade
    inicio
        inteiro idade
        escreva ("Digite a sua idade: ")
        leia (idade)
        se idade >= 18
            escreva ("Você pode ter CNH")
        senão
            escreva ("Você não pode ter CNH")
    fim
```



Exemplos de aplicação

4- Crie um programa em Python que solicite ao usuário a sua idade e mostre se o mesmo pode ou não ter CHN.

```
main.py
1  # Exemplo 04: programa que solicita a idade
2  # do usuário e mostra se ele pode ter CNH
3
4  idade = int(input("Digite a sua idade: "))
5
6  if idade >=18:
7      print("Parabéns! Você pode ter CNH")
8  else:
9      print("Você NÃO pode ter CNH")
```



Exemplos de aplicação

5- Crie um programa em Python que solicite ao usuário um número e mostre se o mesmo é par ou ímpar.

```
algoritmo parOuImpar
    inicio
        inteiro num
        escreva ("Digite um número inteiro")
        leia (num)
        se num % 2 == 0
            escreva ("O número digitado é par")
        senão
            escreva ("O número digitado é ímpar")
    fim
```



Exemplos de aplicação

5- Crie um programa em Python que solicite ao usuário um número e mostre se o mesmo é par ou ímpar.

```
main.py
1  # Exemplo 5: programa que solicita um número
2  # inteiro ao usuário e verifica se ele é par ou
3  # ímpar.
4
5  num = int(input("Entre com um número inteiro: "))
6  if num % 2 == 0:
7      print("O número: ", num, " é par.")
8  else:
9      print("O número: ", num, "é ímpar.")
```



Exemplos de aplicação

6- Crie um programa em Python que solicite duas notas de um aluno ao usuário, calcule a média e mostre se o mesmo está aprovado (média ≥ 6.0) ou reprovado caso contrário.

```
algoritmo media
    inicio
        real n1, n2, m
        escreva("Digite a nota 1")
        leia(n1)
        escreva("Digite a nota 2")
        leia(n2)
        m = (n1+n2)/2
        se (m >= 6) {
            escreva("Aprovado")
        }
        senão{
            escreva("Reprovado")
        }
    fim
```



Exemplos de aplicação

6- Crie um programa em Python que solicite duas notas de um aluno ao usuário, calcule a média e mostre se o mesmo está aprovado (média ≥ 6.0) ou reprovado caso contrário.

main.py

```
1  # Exemplo 06: Cálculo da média
2
3  n1 = float(input("Digite a nota 1: "))
4  n2 = float(input("Digite a nota 2: "))
5
6  media = (n1 + n2) / 2
7  if media >= 6.0:
8      print("Aprovado!!!")
9  else:
10     print("Reprovado!!!")
```



Exemplos de aplicação

7- Faça um programa que solicite ao usuário um número inteiro, calcule e mostre a raiz quadrada desse número. O programa deverá verificar antes se o número digitado é positivo, exibindo uma mensagem de alerta, caso seja negativo.

```
algoritmo raizQuadrada
  inicio
    real num, r
    escreva("Digite um número qualquer")
    leia(num)
    se (num > 0) {
      r = raiz(num)
      escreva("A raiz quadrada do número digitado é:", r)
    } else {
      escreva("Não é possível calcular raiz quadrada de número negativo")
    }
  fim
```



Exemplos de aplicação

7- Faça um programa que solicite ao usuário um número inteiro, calcule e mostre a raiz quadrada desse número. O programa deverá verificar antes se o número digitado é positivo, exibindo uma mensagem de alerta, caso seja negativo.

main.py

```
1 import math
2 num = float(input("Digite um número qualquer: "))
3 if num > 0:
4     r = math.sqrt(num)
5     print("A raiz quadrada de %.2f é %.2f" % (num, r))
6 else:
7     print("Em R, não há raiz quadrada de número negativo")
```



Alguma dúvida????



Exercícios de aplicação



Observações sobre exercícios

- ✓ Todos os exercícios devem ser resolvidos em Python.
- ✓ O código Python pode ser feito no IDLE ou no Repl.it e deve ser salvo um arquivo por exercício com a extensão .py
- ✓ Após finalizar todos os exercícios da aula, compacte os arquivos .py e envie no Blackboard.



Exercícios

- 1- Escreva um algoritmo que solicite um número ao usuário. Caso seja digitado um valor entre 0 e 9, mostre: “valor correto”, caso contrário mostre: “valor incorreto”.
- 2- Crie um algoritmo que solicite ao usuário o seu turno de trabalho e a quantidade de horas trabalhadas, calcule e mostre o valor do salário. Considere os valores de horas a seguir, de acordo com o turno de trabalho. Caso o turno seja igual a ‘N’ (utilize um caractere para representar) o valor da hora trabalhada é R\$ 45,00, caso contrário é R\$ 37,50

Exercícios

3- Faça um programa em Python que obtenha o valor de uma compra, calcular e mostrar o valor da compra considerando o desconto, conforme descrito abaixo:

- ✓ para compras acima de R\$ 200 a loja dá um desconto de 20%
- ✓ para as abaixo disso não tem desconto, mostre o valor da compra.

4- Escreva um programa em Python que solicite ao usuário os valores de três contas de consumo (p.ex. água, luz e telefone) e o valor de seu salário. Verifique se o salário é suficiente para pagar as três contas, caso não seja apresente a mensagem “Salário insuficiente!”. Caso seja, apresente o valor que restou do salário após pagar as contas.



That's all Folks!