



Programação de Computadores

- ✓ Operadores Relacionais e Lógicos
- ✓ Decisão simples

Na aula passada...

Operador	Operação	Prioridade
+	Soma	4º
-	Subtração	4º
*	Multiplicação	3º
/	Divisão	3º
//	Parte Inteira	3º
%	Resto de uma divisão inteira	2º
**	Exponenciação	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 passada...

math é o módulo do Python que reúne as funções matemáticas.

- ✓ É utilizado somente para números não complexos.
- ✓ Para utiliza-lo, devemos fazer a importação da biblioteca **math**:

```
import math
```



Funções	Descrição
math.fabs(x)	Retorna o valor absoluto, não negativo de x.
math.floor(x)	Retorna o maior número inteiro menor ou igual a x.
math.ceil(x)	Retorna o menor número inteiro maior ou igual a x
math.sqrt(x)	Retorna a raiz quadrada de x.
math.trunc(x)	Retorna a parte inteira de x.
math.factorial(x)	Retorna o produto de um inteiro x e todos os inteiros positivos menor que x.

Na aula passada...

Funções	Descrição
<code>math.radians(x)</code>	Retorna o valor da conversão de um ângulo de graus em radianos.
<code>math.sin(x)</code>	Retorna um valor representando o seno de um ângulo x .
<code>math.cos(x)</code>	Retorna um valor representando o cosseno de um ângulo x .
<code>math.tan(x)</code>	Retorna um valor representando a tangente de um ângulo x .
<code>math.asin(x)</code>	Retorna o arco-seno de um valor numérico.
<code>math.acos(x)</code>	Retorna o arco-cosseno de um valor numérico.
<code>math.atan(x)</code>	Retorna o arco-tangente de um valor numérico.
<code>math.hypot(x,y)</code>	Retorna a hipotenusa dos números (catetos) fornecidos.
<code>math.log(x,[base])</code>	Retorna o log de um dado número x na base em questão.
<code>math.pow(x,y)</code>	Retorna o valor de x elevado à potência y Se quisermos o resultado em inteiro, devemos usar a função embutida de Python, <code>pow()</code> , ou o operador <code>**</code> .
<code>math.pi</code>	Retorna o valor do número π

Mais em: <https://docs.python.org/3/library/math.html>



antes de continuar...

Dúvidas quanto aos exercícios da última aula?



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://vidadeprogramador.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/>



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

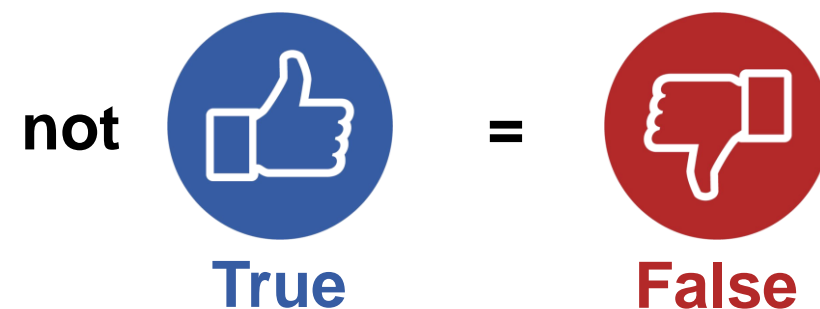
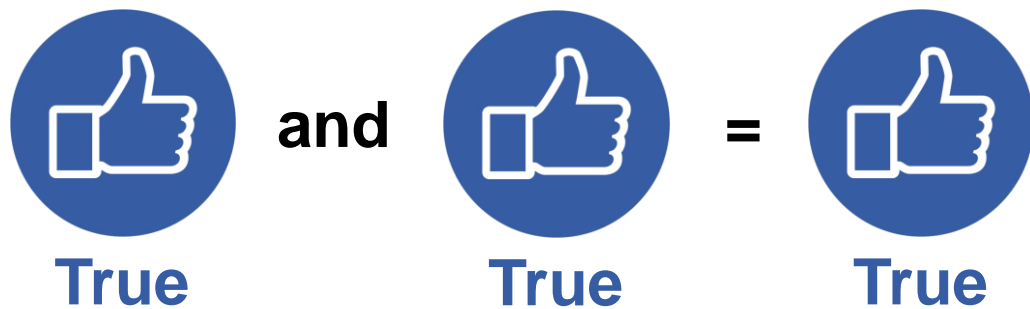
- ✓ **or** → basta que **um dos seus valores seja V** para que o resultado seja **V**.
- ✓ **and** → é necessário que **todos os valores sejam V** para que o resultado seja **V**.



Operadores Lógicos


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





Operadores Lógicos e Relacionais


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


a) $A > C$ 


b) $A < B$ 

c) $A == D$ 


d) $A != C$ 


e) $B <= F$ 


f) $A >= D$ 


g) $A > B$ e $C > D$ 

h) $C > B$ ou $A == D$ 

i) $C == B$ ou $A == D$ 

j) $!(A == D)$ 

k) $!(B == F)$ 

l) $C > B$ e $A == D$ 

Prioridade dos grupos de operadores

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

Menor



Maior



Estruturas Condicionais - if

As estruturas condicionais 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

se condição
bloco de instruções
fim

Expressão lógica



Python

if condição:
bloco de instruções

Expressão lógica

Instruções do bloco devem ser endentadas corretamente



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 .

```
exemplo1.py > ...  
1  #Exemplo 1 - Aula 04  
2  
3  idade = int(input('Digite a sua idade: '))  
4  if idade >= 18: #condição é uma expressão lógica  
5      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



Estruturas Condicionais – if-else

As estruturas condicionais 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



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(f'A raiz quadrada de {num:.2f} é {r:.2f}')
6 ▼ else:
7     print('Em R, não há raiz quadrada de número negativo')
```



Operadores ternários

- ✔ Operadores ternários são todos os operadores que podem receber três operandos.
- ✔ Como as expressões condicionais costumam ser os operadores ternários mais populares nas linguagens em que aparecem, acabamos por associar estes nomes e considerar que são a mesma coisa.
- ✔ A estrutura de uma expressão condicional é algo bem simples, veja só:

```
paz = True if vencer_o_thanos else False  
resposta = 'par' if x % 2 == 0 else 'impar'
```

Exemplos de aplicação

8- 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.

```
#Exemplo 8: Cálculo da média
```

```
n1 = float(input('Digite a 1ª nota: '))
```

```
n2 = float(input('Digite a 2ª nota: '))
```

```
media = (n1+n2)/2
```

```
resultado = 'Aprovado' if media >= 6 else 'Reprovado'
```

```
print(f'Você foi {resultado}, com média: {media}')
```



Bônus!!!

Normalmente, quando executamos algum código no VSCode, o terminal fica “poluído”.



se
Liga
Aí

Podemos utilizar a biblioteca **os** para limpar o terminal e exibir somente as informações que fazem parte da execução do programa.



```
limparterminal.py > ...  
1 import os  
2 os.system('cls' if os.name == 'nt' else 'clear')  
3  
4 nome = input('Digite o seu nome: ')  
5 print(f'Olá {nome}!!Este é um exemplo de terminal limpo!!!')  
6 input()
```

Alguma dúvida????



Exercícios de aplicação



Observações sobre exercícios

- ✓ Todos os exercícios devem ser resolvidos em Python.
- ✓ O código Python pode ser feito no IDLE, no Repl.it, ou na ferramenta que você ache mais adequada e deve ser salvo um arquivo por exercício com a extensão .py
- ✓ Após finalizar todos os exercícios da aula, coloque-os em uma pasta com o nome da aula, compacte a pasta e envie no Blackboard.



<https://youtu.be/BuxuUbgKwCg>



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.

Créditos

Esta aula foi elaborada com base no material produzido e cedido gentilmente pelos **Professores Alcides, Lédon, Amilton e Cristiane.**





That's all Folks!