

Computação I - Python

Aula 3 - Teórica: Tipos de dados, Strings, Estrutura Condicional

Exemplos com Expressões Booleanas

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



Expressões Booleanas

Faça uma função booleana que receba um número como argumento e retorne **True** caso esse número seja par, e retorne **False** caso contrário.

Expressões Booleanas

Faça uma função booleana que receba um número como argumento e retorne **True** caso esse número seja par, e retorne **False** caso contrário.

```
def par(numero):  
    """ Funcao Booleana que retorna True quando passamos um  
        numero par.  
        Parametro de Entrada: int  
        Valor de Retorno : bool """  
  
    return numero%2 == 0
```

Expressões Booleanas

Faça uma função booleana que receba um número como argumento e retorne **True** caso esse número seja par, e retorne **False** caso contrário.

```
def par(numero):  
  
    """Funcao Booleana que retorna True quando passamos um  
    numero par.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
  
    return numero%2 == 0
```

```
>>> par(6)  
True  
  
>>> par(1)  
False
```

Expressões Booleanas

Faça uma função booleana que receba um número como argumento e retorne **True** caso esse número seja **ímpar**, e **False** caso contrário. Use a função *par(numero)* definida anteriormente.

Expressões Booleanas

Faça uma função booleana que receba um número como argumento e retorne **True** caso esse número seja **ímpar**, e **False** caso contrário. Use a função *par(numero)* definida anteriormente.

```
def par(numero):  
    """Funcao Booleana que retorna True quando passamos um numero par.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
    return numero%2 == 0
```

Expressões Booleanas

Faça uma função booleana que receba um número como argumento e retorne **True** caso esse número seja **ímpar**, e **False** caso contrário. Use a função *par(numero)* definida anteriormente.

```
def par(numero):  
    """Funcao Booleana que retorna True quando passamos um numero par.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
  
    return numero%2 == 0
```

```
def impar(numero):  
    """Funcao Booleana que retorna True quando passamos um numero impar.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
  
    return not par(numero)
```

Expressões Booleanas

Defina uma função booleana que, dada uma idade, retorna *True* se a idade for maior ou igual a 18, e retorna *False* caso contrário.

Expressões Booleanas

Defina uma função booleana que, dada uma idade, retorna *True* se a idade for maior ou igual a 18, e retorna *False* caso contrário.

```
def maiorDeldade(idade):  
    """Funcao booleana que dada uma idade retorna True se a idade for maior ou igual a 18 e  
        False caso contrario.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
  
    return idade >=18
```

Expressões Booleanas

Defina uma função booleana que, dada uma idade, retorna *True* se a idade for maior ou igual a 18, e retorna *False* caso contrário.

```
def maiorDeldade(idade):  
    """Funcao booleana que dada uma idade retorna True se a idade for maior ou igual a 18 e  
        False caso contrario.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
  
    return idade >=18
```

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo equilátero, e retorne *False* caso contrário.

Expressões Booleanas

Defina uma função booleana que, dada uma idade, retorna *True* se a idade for maior ou igual a 18, e retorna *False* caso contrário.

```
def maiorDeldade(idade):  
    """Funcao booleana que dada uma idade retorna True se a idade for maior ou igual a 18 e  
        False caso contrario.  
    Parametro de Entrada: int  
    Valor de Retorno : bool"""  
  
    return idade >=18
```

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo equilátero, e retorne *False* caso contrário.

```
def equilatero(A,B,C):  
  
    """Funcao booleana que dados 3 numeros retorna True se eles formam um triangulo  
        equilatero e False caso contrario.  
    Parametros de Entrada: float ,float ,float  
    Valor de Retorno : bool"""  
  
    return A == B == C
```

Expressões Booleanas

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo isósceles, e retorne *False* caso contrário.

Expressões Booleanas

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo isósceles, e retorne *False* caso contrário.

```
def isosceles(A,B,C):  
    """Funcao booleana que dados 3 numeros retorna True se eles formam um triangulo  
        isoseles e False caso contrario.  
    Parametros de Entrada: float,float,float  
    Valor de Retorno : bool"""  
  
    return (A == B) or (A == C) or (C == B)
```

Expressões Booleanas

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo isósceles, e retorne *False* caso contrário.

```
def isosceles(A,B,C):  
    """Funcao booleana que dados 3 numeros retorna True se eles formam um triangulo  
        isoseles e False caso contrario.  
    Parametros de Entrada: float ,float ,float  
    Valor de Retorno : bool"""  
  
    return (A == B) or (A == C) or (C == B)
```

```
>>> isosceles(3.0,3.0,2.0)  
True  
>>> isosceles(3.0,3.0,3.0)  
True
```

Expressões Booleanas

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo isósceles, e retorne *False* caso contrário.

```
def isosceles(A,B,C):  
    """Funcao booleana que dados 3 numeros retorna True se eles formam um triangulo  
        isoseles e False caso contrario.  
    Parametros de Entrada: float ,float ,float  
    Valor de Retorno : bool"""  
  
    return (A == B) or (A == C) or (C == B)
```

```
>>> isosceles(3.0,3.0,2.0)  
True  
>>> isosceles(3.0,3.0,3.0)  
True
```

O que fizemos errado?

Expressões Booleanas

Defina uma função booleana que, dados 3 números, retorne *True* se esses números puderem representar o tamanho dos lados de um triângulo isósceles, e retorne *False* caso contrário.

```
def isosceles(A,B,C):  
    """Funcao booleana que dados 3 numeros retorna True se eles formam um triangulo  
        isoseles e False caso contrario.  
    Parametros de Entrada: float ,float ,float  
    Valor de Retorno : bool"""  
  
    return (A == B) or (A == C) or (C == B)
```

```
>>> isosceles(3.0,3.0,2.0)  
True  
>>> isosceles(3.0,3.0,3.0)  
True
```

O que fizemos errado?

```
def isosceles(A,B,C)  
    """Funcao booleana ..."""  
    return ((A == B) or (A == C) or (C == B)) and (not (A == B == C))
```


Expressões Booleanas

Podem existir diversas formas de se montar uma expressão booleana adequada:

```
def isosceles(A,B,C)
    """ Funcao booleana ... """
    return ((A == B) or (A == C) or (C == B)) and (not (A == B == C))
```

```
def isosceles(A,B,C)
    """ Funcao booleana ... """
    return ((A == B) or (A == C) or (C == B)) and (not equilatero(A,B,C))
```

```
def isosceles(A,B,C)
    """ Funcao booleana ... """
    return ((A == B) and (A != C)) or ((A == C) and (A != B)) or ((C == B) and (A != C))
```

Expressões Booleanas

Observe as expressões abaixo e determine o porquê de elas não poderem ser consideradas expressões booleanas, sabendo que:

- os parâmetros A e B são inteiros e valem 2 e 7
- o parâmetro C é do tipo float e vale 3.5
- o parâmetro L é booleano e vale *False*.

- 1 $(2 < 5)$ **and** $((15/3) = 5)$
- 2 $B = A * C$ **and** $(L$ **or** *True*)
- 3 *True* **not** L **or** *True* **and** $(A + B >= C)$

Expressões Booleanas

Observe as expressões abaixo e determine o porquê de elas não poderem ser consideradas expressões booleanas, sabendo que:

- os parâmetros A e B são inteiros e valem 2 e 7
- o parâmetro C é do tipo float e vale 3.5
- o parâmetro L é booleano e vale *False*.

- 1 $(2 < 5) \text{ and } ((15/3) \text{ == } 5)$: *SyntaxError: invalid syntax*
- 2 $B \text{ == } A * C \text{ and } (L \text{ or } \textit{True})$: *Atribui-se True a B, nenhum boolean é retornado*
- 3 $\textit{True not } L \text{ or } \textit{True and } (A + B >= C)$: *SyntaxError: invalid syntax*

Expressões Booleanas

Agora calcule o resultado das expressões booleanas corrigidas, sabendo que:

- os parâmetros A e B são inteiros e valem 2 e 7
- o parâmetro C é do tipo float e vale 3.5
- o parâmetro L é booleano e vale *False*.

- 1 $(2 < 5)$ **and** $((15/3) == 5)$
- 2 $B == A * C$ **and** $(L$ **or** *True*)
- 3 **not** L **or** *True* **and** $(A + B >= C)$

Expressões Booleanas

Agora calcule o resultado das expressões booleanas corrigidas, sabendo que:

- os parâmetros A e B são inteiros e valem 2 e 7
- o parâmetro C é do tipo float e vale 3.5
- o parâmetro L é booleano e vale *False*.

- 1 $(2 < 5)$ and $((15/3) == 5)$: **True**
- 2 $B == A * C$ and $(L$ or *True*) : **True**
- 3 *not* L or *True* and $(A + B >= C)$: **True**

Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

Computação I - Python

Aula 3 - Teórica: Tipos de dados, Strings, Estrutura Condicional

Exemplos com Expressões Booleanas

Apresentado por: Rafael Machado Andrade

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>

