

COMPUTAÇÃO I

Período 2020-2, turmas EN1 e EPT

Professor: Aloísio Pina

Expressões Lógicas

São expressões que têm como resultado um valor "lógico" (verdadeiro ou falso). Os valores lógicos em python são:

- verdadeiro: True
- falso: False

ATENÇÃO: Para que sejam valores válidos, True e False devem estar escritos com inicial maiúscula e demais letras em minúscula.

Exemplos:

In [1]: true

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-1-724ba28f4a9a> in <module>  
----> 1 true  
  
NameError: name 'true' is not defined
```

In [2]: false

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-2-b73d74fcede9> in <module>  
----> 1 false  
  
NameError: name 'false' is not defined
```

In [3]: TRUE

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-3-425fdcbd1209> in <module>  
----> 1 TRUE  
  
NameError: name 'TRUE' is not defined
```

In [4]: FALSE

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-4-d5612866378f> in <module>  
----> 1 FALSE  
  
NameError: name 'FALSE' is not defined
```

```
In [5]: True
```

```
Out[5]: True
```

```
In [6]: False
```

```
Out[6]: False
```

As expressões lógicas podem conter operadores relacionais e/ou lógicos.

Operadores relacionais:

Comparam operandos numéricos fornecendo um resultado lógico.

- igual: ==
- diferente: !=
- maior: >
- menor: <
- maior ou igual: >=
- menor ou igual: <=

Exemplos:

```
In [7]: 5 == 2
```

```
Out[7]: False
```

```
In [8]: 5 != 2
```

```
Out[8]: True
```

```
In [9]: 5 > 2
```

```
Out[9]: True
```

```
In [10]: 5 < 2
```

```
Out[10]: False
```

```
In [11]: 5 >= 2
```

```
Out[11]: True
```

```
In [12]: 5 <= 2
```

```
Out[12]: False
```

ATENÇÃO: não use apenas um sinal = para comparar números, pois em python isso serve para outra coisa.

Exemplo:

```
In [13]: 5 = 2
```

```
File "<ipython-input-13-5e437cf2364c>", line 1
  5 = 2
    ^
SyntaxError: cannot assign to literal
```

ATENÇÃO: não mude a ordem dos caracteres de um operador.

Exemplo:

```
In [14]: 5 != 2
```

```
File "<ipython-input-14-4680f64d26d2>", line 1
  5 =get_ipython().getoutput(' 2')
    ^
SyntaxError: cannot assign to literal
```

```
In [15]: 5 => 2
```

```
File "<ipython-input-15-190475a39a6f>", line 1
  5 => 2
    ^
SyntaxError: invalid syntax
```

```
In [16]: 5 =< 2
```

```
File "<ipython-input-16-4e8d0c46c756>", line 1
  5 =< 2
    ^
SyntaxError: invalid syntax
```

Operadores lógicos:

Comparam operandos lógicos fornecendo um resultado lógico.

- e: and (só é True se e somente se ambos os operandos forem True; é False caso contrário)
- ou: or (é True se pelo menos um dos operandos for True; é False caso contrário)
- não: not (operador unário, se o operando for True, o resultado é False; se o operando for False, o resultado é True)

Tabela verdade: mostra o resultado de uma operação lógica para todas as combinações possíveis de valores dos operandos.

Tabela verdade do and:

```
In [17]: True and True
```

```
Out[17]: True
```

```
In [18]: True and False
```

```
Out[18]: False
```

```
In [19]: False and True
```

```
Out[19]: False
```

```
In [20]: False and False
```

```
Out[20]: False
```

Tabela verdade do or:

```
In [21]: True or True
```

```
Out[21]: True
```

```
In [22]: True or False
```

```
Out[22]: True
```

```
In [23]: False or True
```

```
Out[23]: True
```

```
In [24]: False or False
```

```
Out[24]: False
```

Tabela verdade do not:

```
In [25]: not True
```

```
Out[25]: False
```

```
In [26]: not False
```

```
Out[26]: True
```

Precedência de operações:

- 1) Operações entre parênteses
- 2) Operações aritméticas
- 3) Operações relacionais
- 4) Operações lógicas (not, and, or, nesta ordem)

Exemplos:

```
In [27]: 5 * 2 ** 3 - 1 > 5 ** 2 * 3 + 1
```

```
Out[27]: False
```

```
In [28]: 5 * 2 ** 3 - (1 > 5) ** 2 * 3 + 1
```

```
Out[28]: 41
```

```
In [29]: True or True and False
```

```
Out[29]: True
```

```
In [30]: True or (True and False)
```

```
Out[30]: True
```

```
In [31]: (True or True) and False
```

```
Out[31]: False
```

```
In [32]: not True and False
```

```
Out[32]: False
```

```
In [33]: (not True) and False
```

```
Out[33]: False
```

```
In [34]: not (True and False)
```

```
Out[34]: True
```

Obs.: É possível criar uma expressão relacional usando mais de um operador, equivalente ao and de operações binárias.

Exemplo: $x < y < z$ equivale a: $(x < y) \text{ and } (y < z)$

Observações sobre o uso do and e do or:

- True and X depende só de X
- False and X é False
- True or X é True
- False or X depende só de X

Observações sobre o uso do not:

- not $(x == y)$ equivale a: $x != y$
- not $(x != y)$ equivale a: $x == y$
- not $(x > y)$ equivale a: $x <= y$
- not $(x < y)$ equivale a: $x >= y$
- not $(x >= y)$ equivale a: $x < y$
- not $(x <= y)$ equivale a: $x > y$
- not $(A \text{ and } B)$ equivale a: $(\text{not } A) \text{ or } (\text{not } B)$
- not $(A \text{ or } B)$ equivale a: $(\text{not } A) \text{ and } (\text{not } B)$