



PYTHON PARA JORNALISTAS DE DADOS

Operadores relacionais, lógicos e *if statements*

Rodolfo Viana
MBA em Jornalismo de Dados
31 de julho de 2021



Operadores relacionais | Comparação de valores

Vimos na aula anterior alguns tipos básicos de dados, como `str`, `int`, `float` e `bool`.

```
texto = "Terceira aula, guerreiros!" # str
inteiro = 8451 # int
ponto_flutuante = 16.97 # float
booleano = True # bool
```

Desses tipos, ainda não usamos `bool`. Ele aparece com mais frequência junto a **operadores relacionais** ou de **comparação** (em suma, operadores que checam se uma equivalência entre dados ou valores é *True* ou *False*).

Vamos supor que `x = 5`:

operador	significado	exemplo	resultado
<code>==</code>	igual a	<code>x == 1</code>	False
<code>!=</code>	não-igual a	<code>x != 2</code>	True
<code>></code>	maior que	<code>x > 5</code>	False
<code>>=</code>	maior ou igual a	<code>x >= 5</code>	True
<code><</code>	menor que	<code>x < 3</code>	False
<code><=</code>	menor ou igual a	<code>x <= 8</code>	True

Operadores relacionais | Comparação de valores

Os **operadores relacionais** servem para avaliar se **uma condição** entre dois valores, dois cálculos, duas variáveis etc. é verdadeira ou falsa.

Testem essas linhas no computador de vocês:



```
x = 12
y = 17
nome = "Rodolfo"

# Com variáveis
print(x <= y)

# Com cálculo de variáveis
print(x ** 2 - 20 == y ** 2 - 165)

# Com cálculo de valores
print(7 % 2 >= (12 / 6) / 2)

# Com funções
print(len(nome) < x)
```

Operadores lógicos | Comparação de valores

Além de avaliar se a relação entre dois valores satisfaz uma condição, pode-se ir além e avaliar **múltiplas condições**.

Para isso, temos os **operadores lógicos**.

Imagine que `x = 5`:

```
x = 12
y = 17
nome = "Rodolfo"

# Quero saber se 3.62 x 4 é:
# 1. maior que x, e
# 2. menor que y
```

operador	significado	exemplo	resultado
and	e (ambas as comparações)	<code>x > 1 and x < 4</code>	False
or	ou (uma das comparações)	<code>x > 1 or x < 4</code>	True
not	negação (inverte a resposta lógica)	<code>not x == 5</code>	False

Operadores lógicos | Comparação de valores

Sem rodar script algum, para cada uma dessas linhas, você consegue dizer:

1. a lógica?
2. o resultado?

```
x = 3.14159
linguagem = "Python"

# 1
print(x * 2 > 6 and x ** 2 > 9)

# 2
print(x * x // 6 < 1 or x % 2 > 1)

# 3
print(not x <= len(linguagem) or len(linguagem) >= x)
```

Controle de fluxo | If statements

`input()` é uma função que pede ao usuário um valor qualquer e salva numa variável como `str`.

`format()` é um método da classe `str` (ou seja, só funciona com `str`) que substitui a marcação `{}` pelo valor de uma variável.

Com essas informações, vocês conseguem me dizer o que este script faz?

```
nome = input("Digite seu nome: ")
idade = input("Digite sua idade: ")
frase = "{} tem {} anos, mas cara de {}".format(nome, idade, int(idade) - 1)
print(frase)
```

Controle de fluxo | If statements

Reparem que o script roda linha por linha, da primeira até a última, sem pular nenhuma. Mas e se quisermos controlar a execução, o fluxo do script?

Uma das formas é com **if statements**, que condiciona a execução de uma linha ao resultado de outra. Sua estrutura é assim:

```
if condicao_1:
    acao_1
elif condicao_2:
    acao_2
elif condicao_n:
    acao_n
else:
    acao_quando_nenhuma_condicao_foi_satisfeita
```

Controle de fluxo | If statements

Um exemplo desmembrado:

```
nome = input("Digite seu nome: ")
idade = input("Digite sua idade: ")
if int(idade) >= 18:
    frase = "{} tem {} anos. É maior de idade".format(nome, idade)
else:
    frase = "{} tem {} anos. É menor de idade".format(nome, idade)
print(frase)
```

Se for maior ou igual a 18...

...frase é isso.

Se não for...

...frase é isso.

Observações:

- A condição deste **if statement** é `int(idade) >= 18`.
- Internamente, o sistema avalia se a resposta é `True` ou `False`.
- A variável `frase` é definida a partir do resultado `True` ou `False`.
Ou seja, o sistema roda apenas uma das duas opções de `frase`.

Controle de fluxo | If statements

Um exemplo prático:

In [4]:

```
# Criação de uma coluna chamada 'punido' e preenchimento
# de cada linha da coluna com 'Sim', 'Não' e 'Incerto'
# (quando o resultado é a continuação do processo), de
# acordo com a penalidade
punido = []

for row in dataset['penalidade']:
    if row == 'EM ANDAMENTO':
        punido.append('Incerto')
    elif row == 'DECISÃO EM OUTRO PA':
        punido.append('Incerto')
    elif row == 'INQUÉRITO ADMINISTRATIVO':
        punido.append('Incerto')
    elif row == 'DADO PROVIMENTO':
        punido.append('Incerto')
```

Análise sobre
impunidade na
GCM [[íntegra](#)]

Controle de fluxo | If statements

Outro exemplo prático:

```
33     for x in lista:
34         time.sleep(0.1)
35         link = url.format(x[1], x[0])
36         try:
37             data = r.get(link).json()
38             for k, v in data.items():
39                 if k == "fotoUrl":
40                     image = r.get(v)
41                     if image.status_code == 200:
42                         filename = '{}_{}_{}.jpg'.format(e, x[1], x[0])
43                         with open('{}_{}'.format(data_dir, filename), 'wb') as f:
44                             f.write(image.content)
45         except:
46             pass
```

Webscraping de
fotos do TSE
[[íntegra](#)]

Controle de fluxo | If statements

Outro exemplo prático:

```
29 base = 'http://app-sisgvconsulta-prd.azurewebsites.net/ws/ws2.asmx'
30 deb_vereador = '/ObterDebitoVereadorJSON?ano=%s&mes=%s'
31 deb_lideranca = '/ObterDebitoLiderancaJSON?ano=%s&mes=%s'
32 gastos = pd.DataFrame(columns=COLUMNAS)
33 for ano in range(2015, ANO_ATUAL + 1):
34     for mes in range(1, 13):
35         print(f'Baixando registros de {mes}/{ano}')
36         json_data = r.urlopen(base + deb_vereador % (ano, mes)).read()
37         if json_data:
38             gastos = gastos.append(parse_expenses(json_data))
39         json_data = r.urlopen(base + deb_lideranca % (ano, mes)).read()
40         if json_data:
41             gastos = gastos.append(parse_expenses(json_data))
```

Webscraping de
gastos de
vereadores de São
Paulo [[íntegra](#)]

Controle de fluxo | Atividade 1

Um número é par quando, na divisão por 2, o resultado for um número inteiro. Ou seja, quando não há resto. Vejam:

$$\begin{array}{r} 14 \overline{) 2} \\ 0 \end{array}$$

$$\begin{array}{r} 15 \overline{) 2} \\ 1 \end{array}$$

Escreva um programa que:

1. peça ao usuário um número,
2. retorne ao usuário a informação sobre o número ser par ou ímpar.

Controle de fluxo | Atividade 2

Recentemente tem feito muito frio em algumas regiões do Brasil. Na minha categorização:

- Menos que 0° é congelante,
- De 0° a 10° é muito frio,
- De $10,1^{\circ}$ a 17° é friozinho,
- De $17,1^{\circ}$ a 24° é ameno,
- De $24,1^{\circ}$ a 30° é calor,
- Acima de 30° é muito calor.

Escreva um programa que:

1. peça ao usuário a temperatura no momento,
2. retorne ao usuário a informação sobre a temperatura de acordo com a tabela acima.

Entenda mais | Material complementar

Para assistir

→ *Estrutura de seleção if e else*, em Cursos Kane Chan [[link](#)]

Para ler

→ *Comandos de decisão*, em Escola Politécnica da PUC-RS [[link](#)]

→ *Operadores*, em IME-USP [[link](#)]

Documentação

→ Função embutida `len()` [[link](#)]

→ Função embutida `input()` [[link](#)]

→ Método da classe `str.format()` [[link](#)]

→ Comandos `if` [[link](#)]