Python para Ciência de Dados - Aula 02

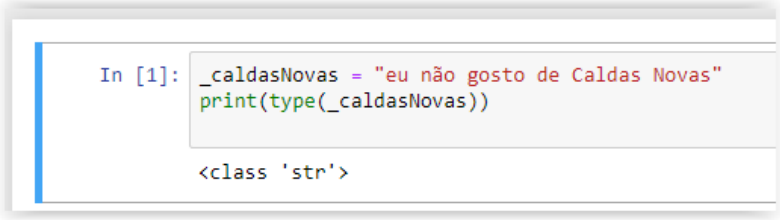
Sumário

1. Manipulação de Strings 2
2. Operadores 3
3. Condicionais 3
   3.1. Usando o elif 3
4. Loops 4
   4.1. Usando for 4
      4.1.1. Usando a função range() para gerar uma sequência 4
   4.2. Usando while 4

1. Manipulação de Strings

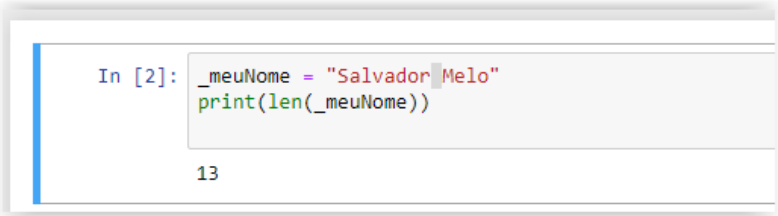
Sabemos que strings são cadeias de caracteres, e em Python utilizae um underline para identificar um string.

Exemplo:

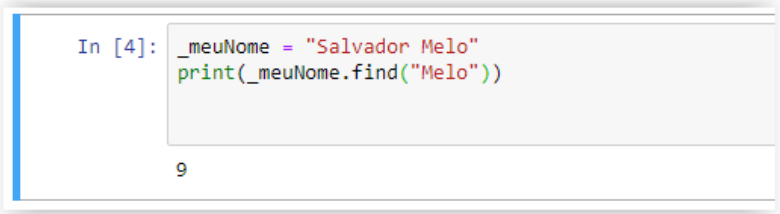


Temos várias funções em Python para trabalhar com strings, cujas as primeiras são:

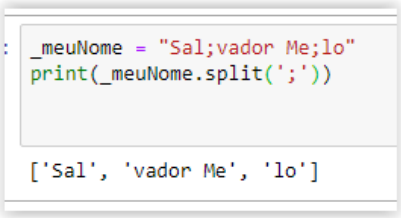
- len(): retorna a quantidade de caracteres armazenados em um string.

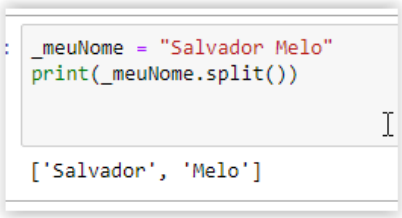


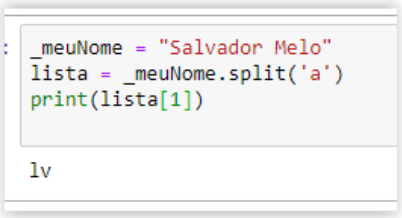
- string.find(): localiza uma determinada posição onde se encontra um substring de um string. A primeira posição é sempre 0. Se não encontrar, retorna -1.



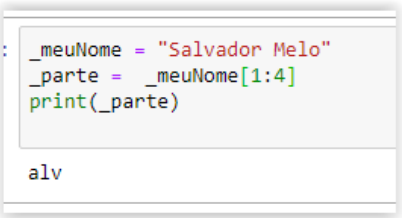
- string.split(): esssa função divide um string em várias partes. Se houver um marcador, separa sempre que encontrar o marcador, caso contrário separa sempre que encontrar em espaço em branco. O resultado é encaminhado para uma lista.







- Quebrando o string em uma parte específica do string.



- Concatenando dois strings.

```
_meuNome = "Salvador"
_meuSobrenome = " Melo"
print(_meuNome+_meuSobrenome)
```

```
Salvador Melo
```

2. Operadores

Os operadores aritméticos são basicamente os mesmos vistos em outros curso de programação, que possibilitam a realização de operações matemática básicas: somas, subtrações, divisões, dentre outras. E os os operadores lógicos são aqueles no qual é feita uma comparação, resultando Verdadeiro(true) ou Falso(false).

```
num1 = 10
num2 = 7
print(num1 + num2)
print(num1 - num2)
print(num1 * num2)
print(num1 / num2)
print(num1 % num2)
```

```
17
3
70
1.4285714285714286
3
```

```
num1 = 10
num2 = 7
print(num1 > num2)
print(num1 >= num2)
print(num1 < num2)
print(num1 <= num2)
print(num1 != num2)
print(num1 == num2)
```

```
True
True
False
False
True
False
```

3. Condicionais

Semelhante ao que já estudamos em outros curso de programação, o uso do if está presente em python. Um grande diferença é que você determina o que está dentro do bloco do if ou do else pela recuo da tabulação.

```
num1 = 7
num2 = 10
if num1 > num2:
    print("executado o if")
    print("{} maior que {}".format(num1,num2))
else:
    print("executado o else")
    print("{} maior que {}".format(num2,num1))
```

```
executado o else
10 maior que 7
```

```
num1 = 7
num2 = 10
if num1 > num2:
    print("executado o if")
print("{} maior que {}".format(num1,num2))
else:
    print("executado o else")
    print("{} maior que {}".format(num2,num1))
```

```
  File "<ipython-input-41-4a4765396380>", line 6
    else:
    ^
SyntaxError: invalid syntax
```

3.1. Usando o elif

Podemos também utilizar a estrutura do elif, semelhante ao elseif de outras linguagens, para adicionar novos testes dentro de um if.

```
num1 = 10
num2 = 10
if num1 > num2:
    print("executado o if")
    print("{} maior que {}".format(num1,num2))
elif num1 == num2:
    print("executado o elif")
    print("{} é igual ao {}".format(num2,num1))
else:
    print("executado o else")
    print("{} maior que {}".format(num2,num1))
```

```
executado o elif
10 é igual ao 10
```

4. Loops

Os laços de repetição em python são utilizados para repetir uma ou mais instruções. Entretanto, podemos acrescentar algumas funcionalidades diferentes, com iterações sobre os itens de qualquer tipo de sequência.

4.1. Usando for

```
numeros = [3,5,7,9]
for x in numeros:
    print(x)
print("terminou")
```

```
3
5
7
9
terminou
```

```
numeros = [3,5,7,9]
for x in numeros:
    print(x**2)
print(numeros)
```

```
9
25
49
81
[3, 5, 7, 9]
```

```
numeros = [3,5,7,9]
for x in numeros:
    print(x)
else:
    print("terminou")
```

```
3
5
7
9
terminou
```

4.1.1. Usando a função range() para gerar uma sequência

```
for x in range(5):
    print(x)
```

```
0
1
2
3
4
```

```
for x in range(3,5):
    print(x)
```

```
3
4
```

```
for x in range(1,10,2):
    print(x)
```

```
1
3
5
7
9
```

4.2. Usando while

```
cont = 1
while cont<10:
    print("cont vale ",cont)
    cont+=1
print("terminou")
```

```
cont vale  1
cont vale  2
cont vale  3
cont vale  4
cont vale  5
cont vale  6
cont vale  7
cont vale  8
cont vale  9
terminou
```

```
cont = 1
while cont<10:
    print("cont vale ",cont)
    cont+=1
else:
    print("terminou")
```

```
cont vale  1
cont vale  2
cont vale  3
cont vale  4
cont vale  5
cont vale  6
cont vale  7
cont vale  8
cont vale  9
terminou
```