

Aula 2 - Operações básicas, Variáveis, Strings, Listas e Indexadores

August 24, 2021

1 Operações básicas, Variáveis, Strings, Listas e Indexadores

operações matemáticas

soma - +

multiplicação *

divisão /

exponenciação **

resto da divisão %

```
[1]: 1+1
```

```
[1]: 2
```

```
[2]: 1*3
```

```
[2]: 3
```

```
[3]: 1/2
```

```
[3]: 0.5
```

```
[4]: 3**2
```

```
[4]: 9
```

```
[5]: 4%2
```

```
[5]: 0
```

Uso: Busca por números pares e impares por exemplo

```
[6]: 5%2
```

```
[6]: 1
```

```
[7]: 3+2-4
```

[7]: 1

Definindo ordens nas operações

```
[6]: (3+2)*(4+1)
```

[6]: 25

Tipos de dados variam para saber o tipo usamos o método type()

```
[7]: type(1)
```

[7]: int

```
[8]: type(2.0)
```

[8]: float

Definição de variável:

Local na memória onde armazeno um valor (posso redefinir ao longo do tempo)

Não posso iniciar uma variável com numeros e caracteres especiais

```
[12]: x=1
```

```
[13]: x=x+1
```

```
[13]: x
```

[13]: 2

```
[14]: &x=2
```

```
File "<ipython-input-14-97dfbcada485>", line 1
&x=2
^
SyntaxError: invalid syntax
```

```
[15]: x=1
      X=2
```

```
[16]: x
```

[16]: 1

```
[17]: X
```

```
[17]: 2
```

ou seja as variáveis são case sensitive

```
[ ]: nome_da_variavel=2
```

```
[ ]: nome_da_Variavel
```

Ele difere entre nome_da_variavel e nome_da_Variavel

Strings são tipos de dados que contem textos

```
[18]: 'nome'
```

```
[18]: 'nome'
```

```
[19]: string = 'nome'
```

```
[22]: print(string)
```

nome

```
[21]: type(string)
```

```
[21]: str
```

Método print apresenta coisas que você queira mostrar

String são instâncias da classe string o python e podemos acessar métodos da classe string É possível acessar um método por meio do .

```
[24]: var = 'Arnaldo'

      'nome: {}'.format(um=var)
```

```
↳ -----
```

```
↳ IndexError                                Traceback (most recent call↳
↳ last)
```

```
<ipython-input-24-f5cd75028a85> in <module>
    1 var = 'Arnaldo'
    2
----> 3 'nome: {}'.format(um=var)
```

`IndexError: tuple index out of range`

No dos parenteses a variável `var` é encaixada Mas se quiser passar duas variáveis ... o string tem o metodo

```
[25]: var = 'Arnaldo'
      idade = 50
      'nome: {um} e tenho {dois} anos'.format(um=var, dois = idade)
```

```
[25]: 'nome: Arnaldo e tenho 50 anos'
```

LISTAS

São os arrays do python, pois permitem o armazenamento de multiplas informações

Elas permitem armazenar as informações de python

Iniciam com abrir e fechar colchetes `[]`

Posso colocar qualquer tipo de valor dentro dela Posso fazer até listas de listas

```
[26]: lista = [1, 2, 3]
```

```
[27]: lista
```

```
[27]: [1, 2, 3]
```

```
[28]: type(lista)
```

```
[28]: list
```

Posso colocar quaisquer tipos de valor dentro de uma lista. Se colocar três variáveis diferentes ...

```
[ ]: lista = [1.0, 'Hello', 3]
```

```
[ ]: lista
```

Colocando listas dentro de listas

```
[31]: lista = [1.0, 'Hello', [1, 2, 3]]
```

```
[34]: lista[1][1]
```

```
[34]: 'e'
```

```
[35]: lista
```

```
[35]: [1.0, 'Hello', [1, 2, 3]]
```

Os elementos da lista possuem posições

Ou seja os elementos da lista são indexados

Posso obter qualquer elemento presente em uma lista

```
[36]: lista[2]
```

```
[36]: [1, 2, 3]
```

Os elementos da lista são indexados apartir do número zero.

Assim o primeiro elemento de uma lista é indexado pelo número zero E assim consecutivamente

```
[ ]: lista[1]
```

Posso acessar elementos dentro de outros elementos em uma lista

```
[37]: lista
```

```
[37]: [1.0, 'Hello', [1, 2, 3]]
```

```
[38]: lista[2][2]
```

```
[38]: 3
```

```
[39]: lista[1][4]
```

```
[39]: 'o'
```

SLICE

O python inicia a indexação no zero

Posso acessar multiplos valores ao mesmo tempo indicando:

posição inicial(inclusive):posiçãofinal(exclui valor final)

lista[anterior:final]

No caso abaixo começo do elemento 1 que é o 'Hello' e vou até o final

```
[41]: lista
```

```
[41]: [1.0, 'Hello', [1, 2, 3]]
```

```
[40]: lista[1:]
```

```
[40]: ['Hello', [1, 2, 3]]
```

No caso abaixo começo do elemento 1 da lista e vou até o elemento 2, note que nesse caso não é apresentado o elemento 2, pois o segundo elemento é excludente

```
[42]: lista[1:2]
```

```
[42]: ['Hello']
```

No caso abaixo como deixei o primeiro elemento em branco começo do elemento 0 e vou até o elemento 2 mas não apresento o elemento 2 apenas o elemento 1 'Hello'

```
[43]: lista[:2]
```

```
[43]: [1.0, 'Hello']
```

Os strings tbm tem a possibilidade de apresentar os resultados

```
[45]: string
```

```
[45]: 'nome'
```

```
[46]: string[2]
```

```
[46]: 'm'
```

```
[47]: string[:2]
```

```
[47]: 'no'
```

```
[23]: string[:4]
```

```
[23]: 'nome'
```