

Tutorial_Py_01

February 24, 2021

1 Tutorial Py # 01

O tutorial está dividido em três partes:

- 1) Operações matemáticas
- 2) Funções
- 3) Estrutura if-else

2 1) Operações matemáticas

Abaixo apresento algumas operações básicas.

- Soma: $4 + 2$

```
[2]: 4 + 2
```

```
[2]: 6
```

- Multiplicação: 4×2

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

```
[3]: 8
```

- Divisão: $4 \div 2$

```
[4]: 4 / 2
```

```
[4]: 2.0
```

- Potenciação: 4^2

```
[6]: 4 ** 2
```

```
[6]: 16
```

- Logaritmo natural: $\ln(2)$

```
[15]: import math  
      math.log(2)
```

[15]: 0.6931471805599453

- Raiz quadrada: $\sqrt{4}$

```
[17]: math.sqrt(4)
```

[17]: 2.0

- Número π

```
[18]: math.pi
```

[18]: 3.141592653589793

- Função `exp()`: e^2

```
[20]: math.exp(2)
```

[20]: 7.38905609893065

Além disso, posso atribuir valores a determinados caracteres (ou palavras), por exemplo, “a” e “test” podem ser iguais a “10” e “-5”.

Logo posso fazer cálculos com esses caracteres.

```
[21]: a = 10
      test = -5

      a * test
```

[21]: -50

3 2) Funções

Abaixo apresento a sintaxe para elaborar funções.

Posso atribuir um nome à função (para deixá-la salva nesse nome), por exemplo, “myfirstfunction”.

Em seguida, entre parênteses, declaro quais são variáveis da função, por exemplo (x,y).

Por fim, entre chaves, descrevo o comportamento da função com base nas variáveis declaradas anteriormente.

Vamos estudar a seguinte função $f(x, y) = x^2 + y^2 + \exp^x + \sqrt{y}$.

```
[27]: def myfirstfunction(x,y):
      return x**2 + y**2 + math.exp(x) + math.sqrt(y)
```

Para executar essa função, basta digitar o nome da função e atribuir valores às variáveis x e y:

```
[29]: myfirstfunction(x = 2, y = 3)
```

[29]: 22.12110690649953

```
[30]: myfirstfunction(x = 3, y = 2)
```

[30]: 34.49975048556076

```
[31]: myfirstfunction(x = 0.5, y = 4)
```

[31]: 19.89872127070013

4 3) Estrutura de seleção (if-else)

Em determinados problemas, precisamos construir estruturas condicionais TRUE ou FALSE.

Por exemplo: “se chover, estudarei em casa; caso contrário, estudarei na UFSC”.

Posso escrever essa frase em termos de código, de modo que o programa retorne o que devo fazer de acordo com a condição climática:

```
[33]: chuva = False # Vamos supor que não está chovendo, então defino chuva igual a
      ↪ false.

      if chuva:
          print("Estudar em casa")
      else:
          print("Estudar na UFSC")
```

Estudar na UFSC

Perceba que antes de executar o código, preciso definir se “chuva = True” ou “chuva = False”.

Após rodar o código, o interpretador ‘diz’ o que devo fazer.

Em outras palavras, como a primeira condição “if” não é “True” nesse exemplo acima, então o código retornou o resultado descrito em “else”, isto é, “Estudar na UFSC”.

Abaixo temos outro exemplo:

```
[35]: a = 5

      if a == 5: # Se "a" for igual a 5, a equação "a = a + 5" será executada.
          a = a + 5
      else:
          a = a + 0

      print(a)
```

10

Como defini “a = 5”, então a primeira condição “if” é verdadeira (True), e “a” é somado a “5”.

Portanto um novo valor de “a” é calculado. Para retornar o novo valor de “a”, é necessário digitar a letra “a” no console e apertar “enter”.

Até a próxima!