s04-conceitos-e-comandos-fundamentais-no-python

March 23, 2021

1 Introdução à Programação no Python

1.0.1 Semana 04

2 Ambiente

Na aula de hoje, vamos utilizar o Repl.it. Veremos como fazer a instalação local em um momento posterior do curso.

2.0.1 O primeiro programa

[1]:
```
print("Olá mundo")
```

```
Olá mundo
```

- `print()` => Função que imprime na tela (console)
- `""` => String, texto
- `Olá mundo` => mensagem a ser impressa

Comentários

[3]:
```
# O cerquilha (jogo da veja, hashtag etc) indica que esta linha é um comentário␣
↪e é ignorada pelo interpretador
print("Olá mundo")
```

```
Olá mundo
```

[5]:
```
"""
Três aspas seguidas fechadas por outras três indicam que todo o bloco entre as␣
↪aspas será
ignorado pelo interpretador. Eu poderia escrever um romance aqui dentro, e o␣
↪interpretador
nem perceberia, mas eu seria mal visto pelos meus pares. Use estes espaços com␣
↪consciência.
"""
print("Olá mundo")
```

```
Olá mundo
```

2.0.2 Cuidados ao digitar seus programas

[9]:
```
print("Olá mundo")
```

```
Olá mundo
```

- Caixa alta e caixa baixa.
- Aspas simples e duplas vivem em pares
- Preste atenção nas cores durante a digitação
- Parênteses são fundamentais em alguns casos, mas não em todos. Também vivem em pares
- Espaços são muito importantes

3 Operações matemáticas

3.0.1 Adição

[2]:
```
2+3
```

[2]: 5

3.0.2 Subtração

[3]:
```
5-3
```

[3]: 2

3.0.3 Multiplicação

[5]:
```
2*10
```

[5]: 20

3.0.4 Divisão

[6]:
```
20/4
```

[6]: 5.0

3.0.5 Exponenciação

[10]:
```
2**3
```

[10]: 8

3.0.6 Resto da divisão inteira

[13]:
```
10 % 3 # Lê-se 10 MOD 3
```

[13]: 1

[12]:
```
4 % 3
```

[12]: 1

[11]:
```
16 % 7
```

[11]: 2

[14]:
```
63 % 8
```

[14]: 7

3.0.7 Quociente de divisão inteira

[15]:
```
10 // 2
```

[15]: 5

[16]:
```
5 // 3
```

[16]: 1

3.0.8 Precedência

Parênteses podem ser utilizados para definir a precedência de operações matemáticas.

[21]:
```
1500 + (1500 * 5 / 100)
```

[21]: 1575.0

3.0.9 PEMDAS

Em geral a ordem de precedência é dada pelo acrônimo PEMDAS: Parênteses, Exponenciação, Multiplicação, Divisão, Adição e Subtração. A ordem da Multiplicação e Divisão não importa assim como com a Adição e Subtração.

4 Variáveis e Atribuição

O comando de atribuição = é utilizado para enviar valores para as variáveis, que representam locais na memória do computador.

[25]:
```
local = 5
local # colocar o nome de uma variável como última linha faz com que ela␣
↪apareça como saída
```

[25]:
```
5
```

[26]:
```
a = 2
b = 3
print(a+b)
```

```
5
```

[29]:
```
a = 2
k = 5
b = k + 2
print(a+b)
```

```
9
```

Cálculo de aumento de salário.

[30]:
```
salário = 1500
aumento = .05 # Outra forma de escrever um número decimal (ponto flutuante)
print(salário + salário * aumento)
```

```
1575.0
```

4.0.1 Exercícios

1) Converta as seguintes expressões matemáticas para que possam ser calculadas usando o interpretador Python.

$10 + 20 \times 30$

[31]:
```
10+20*30
```

[31]:
```
610
```

$4^2 \div 30$

[32]:
```
print(4**2/30)
```

```
0.5333333333333333
```

[33]:
```
(4**2/30)
```

[33]:
```
0.5333333333333333
```

$(9^4 + 2) \times 6 - 1$

[34]:
```
((9**4 + 2)*6-1)
```

[34]: 39377

2) Digite a seguinte expressão no interpretador. Coloque os parênteses de tal forma que NÃO altere a precedência das operações.

```
10 % 3 * 10 ** 2 + 1 - 10 * 4 / 2
```

[36]:
```
((10 % 3 * (10 ** 2)) + 1 - ((10 * 4) / 2))
```

[36]: 81.0

```
5 % 3 * 2
```

[38]:
```
5 * 3 % 2
```

[38]: 1

```
5 % 3 ** 2
```

[40]:
```
5 % (3 ** 2)
```

[40]: 5

4.0.2 Exemplos de programas com variáveis

Programa 1.

[1]:
```
# Um exemplo de programa
a = 2
b = 3
print(a + b)
```

```
5
```

Programa 2.

[2]:
```
# Um outro exemplo de programa
print(2 + 3)
```

```
5
```

Programa 3.

[3]:
```
# Um terceiro exemplo de programa
print(5)
```

```
5
```

4.0.3 Cálculo de aumento de salário

Programa 4.

[4]:
```
# IMPORTANTE: Variáveis devem ter nomes significativos, que facilitem o␣
↪entendimento do programa
salário = 1500
aumento = 5
print(salário + (salário * aumento / 100))
```

```
1575.0
```

4.0.4 Exercícios

3) Escreva um programa que exiba seu nome na tela.

[41]:
```
print("Jefferson")
```

```
Jefferson
```

4) Escreva um programa que exiba o resultado de $2a \times 3b$, onde a vale 3 e b vale 5.

[42]:
```
a = 3
b = 5
print(2 * a * 3 * b)
```

```
90
```

5) Modifique o Programa 1, de forma a calcular a soma de três variáveis.

[43]:
```
a = 2
b = 3
c = 5
print(a + b + c)
```

```
10
```

6) Modifique o Programa 4 de forma que ele calcule um aumento de 15% para um salário de R$ 750.

[44]:
```
salário = 750
aumento = 15
print(salário + (salário * aumento / 100))
```

```
862.5
```

4.1 Nomes de variáveis

Nome	Válido?	Comentários
a1	Sim	Nome inicia com letra
velocidade	Sim	Nome formado por letras

Nome	Válido?	Comentários
velocidade90	Sim	Nome formado por letras e números, mas iniciado por letra
salário_médio	Sim	O símbolo (_) é permitido e facilita a leitura de nomes grandes
salário médio	Não	Variáveis não podem conter espaços em branco.
_b	Sim	O underscore (_) é aceito em nomes de variáveis, mesmo no início
1a	Não	Nomes de variáveis não podem começar com números

4.2 Variáveis numéricas

Complete a segunda coluna indicando se a variável é do tipo inteiro ou ponto flutuante.

Número	Tipo numérico
5	int
5.0	float
4.3	float
-2	int
100	int
1.333	float

4.3 Variáveis lógicas

[46]:
```
resultado = True
aprovado = False
```

4.3.1 Operadores relacionais

Operador	Operação	Símbolo matemático
`==`	Igualdade	$=$
`>`	Maior que	$>$
`<`	Menor que	$<$
`!=`	Diferente	$\neq$
`>=`	Maior ou igual	$\geq$
`<=`	Menor ou igual	$\leq$

4.3.2 Operadores lógicos

Operador Python	Operação
`not`	não

Operador Python	Operação
and	e
or	ou

4.3.3 Operador not

[22]:
```
not True
```

[22]:
```
False
```

[47]:
```
not resultado
```

[47]:
```
False
```

[23]:
```
not False
```

[23]:
```
True
```

4.3.4 Operador and

V1	V2	V1 and V2
V	V	V
V	F	F
F	V	F
F	F	F

[26]:
```
print(True and True)
print(True and False)
print(False and True)
print(False and False)
```

```
True
False
False
False
```

[51]:
```
v1 = True
v2 = False

if (v1 and v2):
    print("Os dois são True")
else:
    print("Pelo menos um é False")
```

```
Pelo menos um é False
```

4.3.5 Operador or

V1	V2	V1 or V2
V	V	V
V	F	V
F	V	V
F	F	F

[27]:
```
print(True or True)
print(True or False)
print(False or True)
print(False or False)
```

```
True
True
True
False
```

[54]:
```
v1 = False
v2 = False

if (v1 or v2):
    print("Pelo menos um e True")
else:
    print("Os dois são False")
```

```
Os dois são False
```

4.3.6 Expressões lógicas

[56]:
```
salário = 1002
idade = 20
```

[57]:
```
#       TRUE     and     TRUE
salário > 1000 and idade > 18
```

[57]:
```
True
```

[59]:
```
salário = 200
idade = 30
```

[60]:
```
salário > 1000 and idade > 18
```

[60]:
```
False
```

4.4 Exercícios

7) Escreva uma expressão para determinar se uma pessoa deve ou não pagar imposto. Considere que pagam imposto pessoas cujo salário é maior que R$ 1.200,00.

[63]:
```
salário = 5000
base_imposto = 1200
salário > base_imposto
```

[63]:
```
True
```

8) Resolver esse exercício de criptoaritmética. Cada letra representa um algarismo. Letras iguais, algarismos iguais. Letras diferentes, algarismos diferentes. Que combinação de algarismos resolve a soma a seguir?

```
   S  E  N  D
+  M  O  R  E
---------------
 M  O  N  E  Y
```