Aula 02

Revisando

Instalando o Python no Windows : https://python.org.br/instalacao-windows/

Após instalado, abrimos o Prompt de Comando e digitamos :

- python –version : para ver se está instalado corretamente e a sua versão;
- python : para inicializar o Shell do Python;

Veremos mais sobre os arquivos adiante.
Para sairmos do Shell, usamos a função **exit()**.

Podemos usar comentários para documentar o código ou evitar que certar partes sejam executadas.

```
>>> # abaixo mostra a resposta da vida e do universo
>>> print('o valor é', 42)
o valor é 42
>>>
>>> x = 5
>>> # x = 10
>>> print("O valor de x é", x)
O valor de x é 5
```

Lembre de usar os comentários em locais estratégicos e que não sejam óbvios.

A função print() é usada para mostrar valores de variáveis e/ou valores puros de inteiros, floats, strings, etc.
A sintaxe da função é a seguinte :

print(argumento_1, argumento_2, ..., argumento_n)

```
>>> num = 42
>>> nome = 'augusto'
>>>
>>> print(nome)
augusto
>>> print('aqui vai mostrar o número', num)
aqui vai mostrar o número 42
>>> print('aqui vai mostrar apenas texto')
aqui vai mostrar apenas texto
>>> print(num, nome)
42 augusto
```

E podemos usar o argumento “sep” para definir qual será o separador das variáveis.

```
>>> # trocando o separador padrão
>>> print('sou um', "belo", 42, sep='.')
sou um.belo.42
```

Lembre de dar nomes que façam sentido para suas variáveis e “separar as palavras” por um sublinhado sem acentuação.

Tenha em mente que o Python é uma linguagem de programação de tipagem dinâmica e forte.

Os tipos numéricos vistos até agora foram os int e float, onde ambos aceitas os operadores aritméticos + - * / ** // %.
As strings são textos envoltos de aspas simples ou duplas.

Mais Strings

Caracteres de Escape

Caso precisemos usar aspas simples ou aspas duplas dentro da string, podemos usar de duas formas:
- usar a string oposta;
- usar `\` (contra-barra) como caracter de escape, caso necessário;

```
>>> print("sant'ana")
sant'ana
>>> print('"Nós nunca realmente crescemos, nós apenas
aprendemos a agir em público." - Bryan White')
"Nós nunca realmente crescemos, nós apenas aprendemos
a agir em público." - Bryan White
>>>
>>> nome = 'sant\'ana'
>>> print(nome)
sant'ana
```

Strings possuem vários caracteres de escape :

- \n : nova linha;
- \\ : contra barra;
- \' : aspas simples;
- \" : aspas duplas;
- \t : tabulação;

Exemplos :

```
>>> print('sou uma\nstring em\nvárias linhas')
sou uma
string em
várias linhas
>>>
>>> print('C:\\Windows\\system32')
C:\Windows\system32
>>>
>>> print('um menu\n\t1. item 1\n\t2. item 2')
um menu
        1. item 1
        2. item 2
```

Strings Literais

Strings podem se espalhar por múltiplas linhas. Uma forma é usar aspas triplas simples ou duplas.
A nova linha (\n) é automaticamente incluído na string, mas é possível previnir isso adicionando uma contrabrarra ao final da linha.

```
>>> print('''sou uma string
... dividida em três
... linhas''')
sou uma string
dividida em três
linhas
>>>
>>> print("""e eu sou uma
... string dividida em apenas \
... duas linhas""")
e eu sou uma
string dividida em apenas duas linhas
```

Se colocarmos duas ou mais strings literais lado a lado, elas serão concatenadas automaticamente.

```
>>> print('augusto' 'hertzog')
augustohertzog
```

Isso é particularmente útil quando você quer repartir uma string muito longa.

```
>>> print('o python é'
... 'versátil com strings')
o python éversátil com strings
```

Essa técnica só funciona com strings literais, não com variáveis ou expressões.

```
>>> nome = 'augusto'
>>> sobrenome = 'hertzog'
>>> print(nome sobrenome)
  File "<stdin>", line 1
    print(nome sobrenome)
          ^^^^^^^^^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
>>> print('augusto' sobrenome)
  File "<stdin>", line 1
    print('augusto' sobrenome)
          ^^^^^^^^^^^^^^^^^^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
>>>
>>> print('augusto' 'hertzog')
augustohertzog
```

Operadores + e * nas Strings

As strings são concatenadas usando o sinal de +.

```
>>> nome = 'Augusto'
>>> sobrenome = 'Hertzog'
>>> print(nome, sobrenome, sep='---')
Augusto---Hertzog
>>>
>>> nome_completo = nome + sobrenome
>>> print(nome_completo)
AugustoHertzog
```

E as strings podem ser repetidas com o operador *.

```
>>> print('augusto' * 10)
augustoaugustoaugustoaugustoaugustoaugustoaugustoaugustoaugustoaugustoaugusto
```

Método format()

A string possui um método chamado format, que é usado para exibir as strings formatadas de uma forma muito mais elegante.

```
>>> nome = 'augusto'
>>> idade = 15
>>>
>>> 'meu nome é {} e tenho {} anos'.format(nome, idade)
'meu nome é augusto e tenho 15 anos'
>>>
>>> texto = 'meu nome é {} e tenho {} anos'.format(nome, idade)
>>> texto
'meu nome é augusto e tenho 15 anos'
>>> print(texto)
meu nome é augusto e tenho 15 anos
```

Dentro das chaves, nós podemos usar de 3 formas :

- de acordo com a posição;
- um nome criado para a posição;
- o número indicando a posição (repare que ele começa em 0, veremos mais sobre isso adiante);

```
>>> nome = 'augusto'
>>> idade = 20
>>> 'meu nome é {} e tenho {} anos'.format(nome, idade)
'meu nome é augusto e tenho 20 anos'
>>> 'meu nome é {v1} e tenho {v2} anos'.format(v2=idade,v1=nome)
'meu nome é augusto e tenho 20 anos'
>>> 'meu nome é {0} e tenho {1} anos'.format(nome,idade)
'meu nome é augusto e tenho 20 anos'
```

Além disso, é possível especificar o número de casas decimais que devem ser exibidas para números de ponto flutuante, usando {:.nf}, onde “n” é o número de casas decimais.

```
>>> pi = 3.1415926535
>>> print('o valor de pi é {:.2f}'.format(pi))
o valor de pi é 3.14
>>> print('o valor de pi é {:.4f}'.format(pi))
o valor de pi é 3.1416
```

Também podemos usar o método “format()” para especificar o tamanho do campo e o alinhamento.

```
>>> # alinhando à esquerda
>>> print('-{:<10}-'.format('olá'))
-olá        -
>>> # alinhando à direita
>>> print('-{:>10}-'.format('olá'))
-       olá-
>>> # centralizando
>>> print('-{:^10}-'.format('olá'))
-   olá    -
```

Como visto acima, a string é preenchida com espaços, mas podemos especificar o que quisermos.

```
>>> # alinhando à esquerda, preenchendo com _
>>> print('-{:_<10}-'.format('olá'))
-olá_______-
>>> # alinhando à direita, preenchendo com +
>>> print('-{:+>10}-'.format('olá'))
-+++++++olá-
>>> # centralizando, preenchendo com *
>>> print('-{:*^10}-'.format('olá'))
-***olá****-
>>> # preenchendo com zeros a direita
>>> print('-{:0<10}-'.format('123'))
-1230000000-
>>> # preenchendo com zeros a esquerda
>>> print('-{:0>10}-'.format('123'))
-0000000123-
```

Função type()

A função type é usada quando queremos saber a que tipo pertence determinada variável ou valor. Ela vai retornar o tipo da variável ou valor.

```
>>> type('sou uma string')
<class 'str'>
>>>
>>> type(42)
<class 'int'>
>>>
>>> type(3.14)
<class 'float'>
>>>
>>> type(True)
<class 'bool'>
```

Repare que para todos os tipos mostrados acima retornaram `<class 'xxx'>`. Isso quer dizer que aquilo que foi passado para a função é na verdade uma classe.
Isso é algo característico do Python, onde todas as variáveis são classes, com seus atributos e métodos. Veremos mais sobre isso adiante.

Exercícios

1. Qual o caractere de escape utilizado para quebrar linha em uma string?
2. Escreva uma string que utilize o caractere de escape para imprimir uma citação entre aspas duplas e o nome do autor.
3. Qual é o resultado da expressão: 10 + 5 / 2 * 3 - 1
4. Como usar a função type para descobrir o tipo de dado de uma variável?
5. Qual o resultado da expressão: 10 / 5 + 3 * 2 - 1
6. Como utilizar o método format para inserir variáveis em uma string?
7. Qual o resultado da expressão: 5 ** 2
8. Qual o resultado da expressão: 10 % 3
9. Escreva uma string que utilize o caractere de escape para imprimir uma barra invertida.
10. Qual o resultado da expressão: 10 // 3
11. Escreva uma string que utilize o método format para imprimir o nome e a idade de uma pessoa.
12. Qual o resultado da expressão: 15 - 7 * (8 / 4 + 2)
13. Escreva uma string que utilize o método format para imprimir o valor de uma variável inteira.
14. Qual o resultado da expressão: 2 + 3 * 4
15. Escreva uma string que utilize o caractere de escape para imprimir uma tabulação.
16. Qual o resultado da expressão: 2 ** 3 + 4
17. Escreva uma string que utilize o método format para imprimir o valor de uma variável float com duas casas decimais.
18. Crie uma string que mostre seu ano de nascimento com base na idade atual.
19. Qual o resultado da expressão: (2 + 3) * 4
20. Escreva uma string que utilize o caractere de escape para imprimir um caractere de aspa simples.
21. Qual o resultado da expressão: 10 / (2 + 3) * 4
22. Crie uma variável que armazene um texto com seu nome completo, concatene com uma mensagem de saudação e a imprima.
23. Crie uma string que utilize o método format para inserir o seu nome em uma mensagem.
24. Crie uma string que utilize o método format para inserir a sua idade em uma mensagem.
25. Crie uma string que utilize o método format para inserir um número decimal em uma mensagem.
26. Crie uma string que utilize o método format para inserir um número inteiro em uma mensagem.
27. Crie uma string que utilize o método format para inserir duas strings em uma mensagem.
28. Crie uma string que utilize o método format para inserir três números inteiros em uma mensagem.
29. Crie uma string que utilize o método format para inserir duas strings e um número inteiro em uma mensagem.
30. Crie uma string que utilize o método format para inserir uma string e um número decimal em uma mensagem.
31. Crie uma string que utilize o método format para inserir dois números inteiros e um número decimal em uma mensagem.
32. Crie uma string que utilize o método format para inserir três números decimais em uma mensagem.
33. Crie uma string que utilize o método format para inserir uma string e dois números decimais em uma mensagem.
34. Crie uma string que utilize o método format para inserir três strings em uma mensagem.
35. Crie uma string que utilize o método format para inserir um número inteiro e duas strings em uma mensagem.

36. Crie uma string que utilize o método format para inserir duas strings e um número decimal em uma mensagem.
37. Crie uma string que utilize o método format para inserir três números inteiros em uma mensagem, cada um com um número de casas decimais diferente.
38. Crie uma string que utilize o método format para inserir uma string e um número inteiro em uma mensagem, onde o número inteiro é apresentado em formato de porcentagem.
39. Crie uma string que utilize o método format para inserir um número decimal em uma mensagem, onde o número é apresentado em notação científica.
40. Crie uma string que utilize o método format para inserir uma string e dois números inteiros em uma mensagem, onde os números inteiros são apresentados em formato de data.
41. Crie uma string que utilize o método format para inserir um número decimal em uma mensagem, onde o número é apresentado com um símbolo de moeda.
42. Crie uma string que utilize o método format para inserir uma string e um número decimal em uma mensagem, onde o número é apresentado com um número fixo de casas decimais.
43. Crie uma variável com um valor float e imprima uma string que informe o valor com duas casas decimais.
44. Crie uma variável com uma lista de três palavras e imprima uma string que concatene as palavras com um espaço entre elas.
45. Crie uma variável que armazene um texto com um número no meio e transforme esse número em seu dobro.
46. Crie uma variável com um texto contendo uma palavra repetida três vezes.
47. Crie uma variável com um número inteiro e imprima uma string que informe o seu valor em binário.
48. Crie 3 variáveis com as notas de três provas e mostre uma mensagem com a média do aluno.
49. Crie uma variável com um número inteiro e imprima uma string que informe se o número é par ou ímpar.
50. Crie uma string que contenha a palavra "olá" repetida 5 vezes utilizando o operador *.
51. Crie uma string que contenha a palavra "mundo" repetida 3 vezes utilizando o operador *.
52. Crie uma string que contenha a palavra "Python" repetida 2 vezes e a palavra "é legal" uma vez utilizando o operador +.
53. Crie uma string que contenha a palavra "programação" repetida 3 vezes utilizando o operador * e depois adicione " em Python" utilizando o operador +.
54. Crie uma string que contenha a palavra "amor" repetida 4 vezes utilizando o operador * e adicione " verdadeiro" no final utilizando o operador +.
55. Crie uma string que contenha a palavra "fácil" repetida 2 vezes, a palavra "Python" uma vez e a palavra "é" uma vez utilizando o operador +.
56. Crie uma string que contenha a palavra "casa" repetida 2 vezes utilizando o operador * e depois adicione " de praia" utilizando o operador +.
57. Crie uma string que contenha a palavra "caminhar" repetida 3 vezes utilizando o operador * e adicione " na natureza" no final utilizando o operador +.
58. Crie uma string que contenha a palavra "trabalho" repetida 4 vezes utilizando o operador * e depois adicione " duro" utilizando o operador +.
59. Crie uma string que contenha a palavra "festa" repetida 2 vezes utilizando o operador * e adicione " de aniversário" utilizando o operador +.
60. Crie uma string que contenha a palavra "piscina" repetida 3 vezes utilizando o operador * e depois adicione " em casa" utilizando o operador +.
61. Crie uma string que contenha a palavra "frio" repetida 4 vezes utilizando o operador * e adicione " na montanha" no final utilizando o operador +.
62. Crie uma string que contenha a palavra "comida" repetida 2 vezes, a palavra "gostosa" uma vez e a palavra "é" uma vez utilizando o operador +.
63. Crie uma string que contenha a palavra "férias" repetida 3 vezes utilizando o operador * e depois adicione " na praia" utilizando o operador +.

64. Crie uma string que contenha a palavra "estudar" repetida 4 vezes utilizando o operador * e adicione " é importante" utilizando o operador +.
65. Crie uma string que contenha a palavra "livro" repetida 2 vezes utilizando o operador * e depois adicione " de aventura" utilizando o operador +.
66. Crie uma string que contenha a palavra "esporte" repetida 3 vezes utilizando o operador * e adicione " faz bem à saúde" no final utilizando o operador +.
67. Crie uma string que contenha a palavra "passear" repetida 4 vezes utilizando o operador * e depois adicione " no parque" utilizando o operador +.
68. Crie uma string que contenha a palavra "bebida" repetida 2 vezes, a palavra "gelada" uma vez e a palavra "é" uma vez utilizando o operador +.
69. Crie uma string que contenha a palavra "música" repetida 3 vezes utilizando o operador * e adicione " alegra o dia" no final utilizando o operador +.
70. Crie uma string onde mostre sua idade, seu nome e sua data de nascimento.