Recebendo dados de usuario – Seção 3.12 - Python

*"""recebendo dados do usuario  
  
input() -> todo dado recebido via input é string  
Em python, string é tudo que estiver entre:  
- aspas simples  
- aspas duplas  
- aspas simples triplas  
- aspas duplas triplas  
Exemplos  
Aspas simples 'David'  
Aspas duplas "David"  
Aspas simples triplas '''David'''  
"""*#Aspas duplas triplas """David"""  
  
  
  
# Entrada de dados  
#print("Qual é seu nome? ")  
#nome = input()  
#Acima um modo de comando com duas linhas. Mas da pra colocar em uma linha. Abaixo  
nome = input('Qual seu nome? ')  
  
  
#Abaixo exemplo de print antigo  
#print('Seja bem vindo %s' % nome)  
  
  
#Saida de dados  
#Abaixo exemplo de print antigo  
#print('O %s tem %s anos' % (nome, idade))  
  
#Exemplo de print a partir da versão 3.  
#print('Seja bem vindo {0}.Format(nome))  
  
#Exemplo de print mais atual, a partir da versão 3.7  
print(f'Seja bem vindo {nome}')  
  
  
#print("Qual a sua idade")  
#idade = input()  
#Acima um modo de comando com duas linhas. Mas da pra colocar em uma linha. Abaixo  
#idade = input('Qual sua idade? ')  
#Abaixo essa mesma linha de comando com cast  
idade = int(input('Qual a sua idade?'))  
  
#Exemplo de print mais atual, a partir da versão 3.7  
print(f'O {nome} tem {idade} anos')  
  
print(f'O {nome} nasceu em {2021 - int(idade)}')  
  
"""  
Int(idade) é um cast  
Cast é a conversão de um tipo de dado para outro  
  
"""

Aula de numerico – seção 4.15 o tipo númerico

>>> 4/2

2.0

>>> 5/2

2.5

>>> int (5/2)

2

>>> 5 // 2

2

>>> 4\*4

16

>>> 4\*\*4

256

>>> 3\*\*3

27

>>> 3\*3

9

>>> int (5/2)

>>> 5%2

1

>>> 4%3

1

>>> 10%2

0

>>> 10%3

1

Comando **type** diz qual tipo de numero que é, se float ou int ou outro

>>> num = 42

>>> type(num)

<class 'int'>

>>>

>>> type(23)

<class 'int'>

>>> type(2.5)

<class 'float'>

>>>

Tipo Float – Seção 4.16

*"""  
tipo float, conhecido também como tipo real, decimal,  
casas decimais.  
Obs: O separador de casas decimais na programação é um  
ponto, não é virgula  
"""*#errado 1,44  
valor = 1, 44  
print(valor)  
print(type(valor))  
  
#certo 1.44  
valor = 1.44  
print(valor)  
print(type(valor))  
  
#Comando Type mostra o tipo, se é Int, Float, tuple,  
  
#É possivel. Fazer duas atribuições  
valor1, valor2 = 1, 44  
print(valor1)  
print(type(valor1))  
print(valor2)  
print(type(valor2))  
  
#podemos converter um float para int  
"""Ao converter valores float para inteiro, nós perdemos precisão. O resultado do comando abaixo vai dar 1, mas o numero original  
é 1.44"""  
res = int(valor)  
print(res)  
print(type(res))  
  
#podemos trabalhar com numero complexos  
variavel = 5j

Tipo Booleano Seção 4.17

*"""  
Tipo Booleano  
Algebra Booleana, criada por Georfe Boole  
2 constantes, verdadeira ou falso  
  
True - verdadeiro  
false - Falso  
  
OBS: Sempre com a letra maiuscula.  
Errado - true, false  
Certo - True, False  
"""*ativo = True  
print(ativo)  
"""Serve também para operações básicas"""  
# Negação (not)  
"""Fazendo a negação, se o valor booleano for verdade, o resultado será falso"""  
print(not ativo)  
  
# Operação Ou (or)  
"""É uma operação binária, ou seja, depende de dois valores. Ou um ou outro deve ser verdadeiro"""  
"""True or True > True  
True or False > True  
False or True > True  
False or False > False  
Veja abaixo"""  
print(ativo or logado)  
  
# Operação E (and)  
"""também é uma operação binária, ou seja, depende de dois valores. Ambos os valores devem ser verdadeiros."""  
""" True and True > True  
 True and False > False  
 False and True > False  
 False and False > False"""

Tipo String – Seção 4.18

*"""  
Tipo String  
  
Em Python, um dado é considerado um tipo String sempre que estiver entre aspas simples  
Exemplo: 'uma string', '234', 'a', 'True', '43.2'  
Sempre que estiver com aspas duplas  
Exemplo: "uma string", "234", "a", "True", "43.2"  
Ou sempre que tiver as duplas ou triplas, simples triplas  
'''  
'''uma string'''  
  
"""*"""Abaixo, lista de comandos que guardei como comentário"""  
"""  
nome = 'Geek University'  
print(nome)  
print(type(nome))  
  
nome = "Gina´s bar"  
print(nome)  
print(type(nome))  
  
nome = 'Angelina \nJolie'  
print(nome)  
print(type(nome))  
  
nome = 'Angelina Jolie'  
print(nome)  
print(type(nome))  
  
  
"""  
  
nome = 'Geek University'  
print(nome.upper())  
"""upper deixa tudo maiusculo"""  
"""lower deixa tudo minusculo"""  
  
nome = 'Geek University'  
print(nome.split())  
"""split transforma em uma lista de Strings cada palavra no resultado  
Exemplo ['Geek', 'University']"""  
  
#['G', 'e', 'e', 'k' ' ', 'U', 'n', 'i', 'v', 'e', 'r', 's', 'i', 't', 'y'  
#['0', '1', '2', '3', '4', '5', '6', '7' '8', '9', '10', '11', '12', '13',  
"""  
nome = 'Geek University'  
print(nome[0:4])  
  
nome = 'Geek University'  
print(nome[5:15]) #essa operação é chamado de Slice de String que serve para selecionar qual parte da string vc quer imprimir na tela  
  
print(nome.split()[0])  
#Com o comando .split, nós transformamos o Geek University em 'Geek' 'university', ou seja, em vez de separar por leatras, separamos por palavra. Neste caso o [0] é Geek e o [1] é University  
print(nome.split()[1])  
"""  
  
#podemos também inverter as letras, de traz pra frente  
  
nome = 'Geek University'  
print(nome[::-1])  
"[::-1] comece do primeiro elemento, vá até o ultimo e inverta"  
  
print(nome.replace('G', 'p'))  
"""(nome.replace('G', 'p')) vai substituir as letras"""  
print(nome.replace('e', 'i'))  
  
texto = 'socorram me subindo no onibus em marrocos'  
print(texto[::-1])  
"acima inverso"