DESEMPACOTANDO\_LISTAS (\*args e \*kwargs)

\***args**

def exemplo(a, b, \*args):

   print(a, b, \*args)

>>> exemplo(‘abacate’, ‘maça’, ‘uva’, ‘goiaba’)

abacate maça uva goiaba

exemplo 2:

def func(\*args):

    print(args)

lista = [1,2,3,4,5]

print(\*lista)

>>> 1 2 3 4 5

Exemplo 2b:

def func(\*args):

    print(args)

lista = [1,2,3,4,5]

print(\*lista, sep='-')

>>> 1-2-3-4-5

=None

exemplo 3:

def func(a1, a2, a3, a4, a5, nome=None):

    print(a1, a2, a3, a4, a5, nome)

func(1,2,3,4,5, nome='Luiz')

>>> 1 2 3 4 5 Luiz

Acessar valores pela posição

def func(\*argumentos):

    print(argumentos)

    print(argumentos[0])

    print(argumentos[-1])

func(1,2,3,4)

>>> (1, 2, 3, 4) valores

1 pos[0]

4 pos[-1] ultimo item

CAST DENTRO DAS FUNÇÕES

Tupla é uma Lista imutável. Todavia dentro das funções podemos fazer o casting de uma tupla e transforma-la em uma lista, tornando-se assim um objeto mutável.

Ex:

def func(\*args):

    args = list (args)

    args[0] = 20

    print(args)

func(1,2,3,4)

>>> [20, 2, 3, 4]

Note que foi feito o casting e alterado o primeiro valor (1 => 20).

def func(\*args):

    print(args)

lista = [1,2,3,4,5]

func(lista,7)

>>> ([1, 2, 3, 4, 5], 7)~ retorna uma lista dentro de uma tupla

Para solucionar isso podemos desempacotar a lista antes de imprimi-la.

def func(\*args):

    print(args)

lista = [1,2,3,4,5]

func(\*lista,7) ~ com \*lista desempacotamos a lista

>>> (1, 2, 3, 4, 5, 7)

\*\*KWARGS

def func(\*args, \*\*kwargs):

    print(args)

    print(kwargs)

lista = [1,2,3]

func(\*lista, sobrenome='Álecks', nome='Rubem')

>>> (1, 2, 3)

{'sobrenome': 'Álecks', 'nome': 'Rubem'}

Caso queira acessar só um argumento especifico podemos fazer da seguinte forma:

def func(\*args, \*\*kwargs):

    print(args)

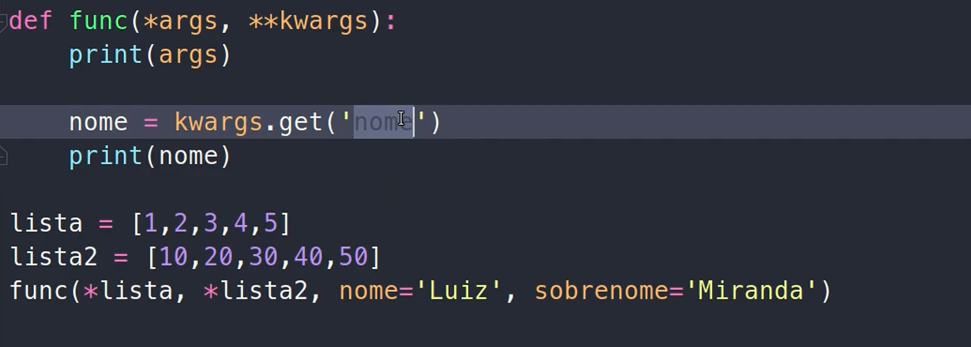
    print(kwargs['nome'])

lista = [1,2,3]

func(\*lista, sobrenome='Álecks', nome='Rubem')

>>> (1, 2, 3)

Rubem

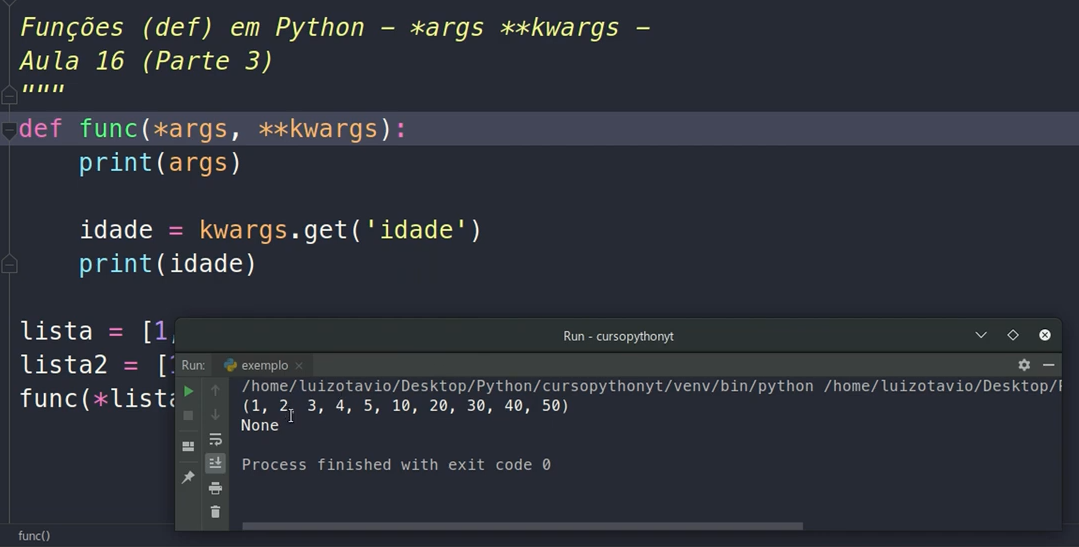
Se quisermos saber se foi enviada alguma chave predefinida:

quando não sabermos se a chave existe usamos .get(‘’), se a chave já foi definida poderia ser usado assim: kwargs[nome]

Vai retonar Luiz, que foi o valor atribuído.

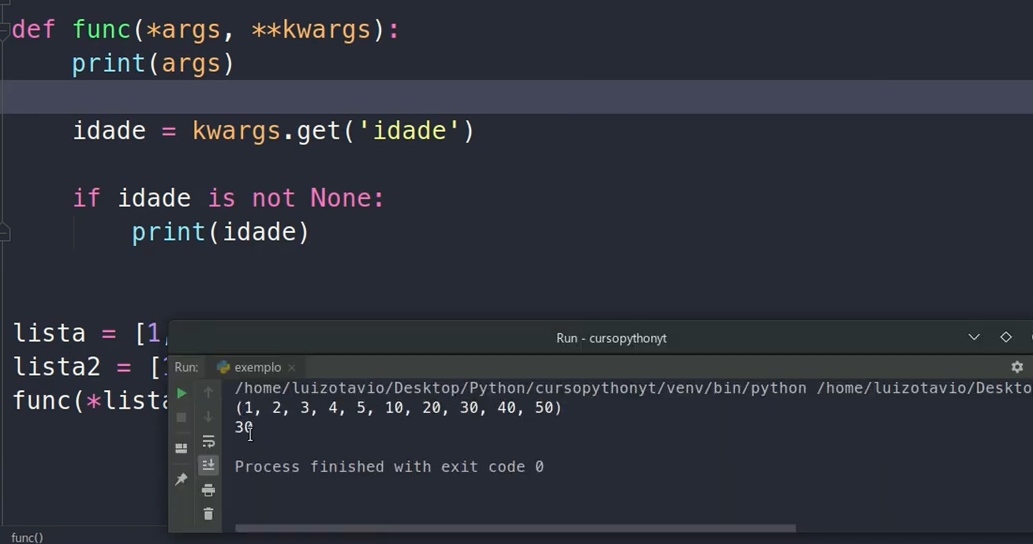
Exemplo2:

A chave ‘idade’ não existe



Vai retornar none pois esta chave não exite.

Depois de criarmos a chave de “idade” podemos melhorar o programa fazendo assim:



O programa retorna o valor de idade normalmente