

Computação I - Python

Aula 4 - Variáveis e Atribuição, Strings e Tuplas

Tuplas

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



Tuplas

- O que são Tuplas?

Tuplas

- Uma **tupla** é uma sequência heterogênea (permite que seus elementos sejam de tipos diferentes):

```
1 >>> a = (1, 2, 3, 4)
2 >>> b = (1.0, 2, '3', 4+0j)
3 >>> c = 1, 2, 3, 4
4 >>> d = (1, )
```

- Valores em uma tupla podem ser distribuídos em variáveis como uma atribuição múltipla:

```
1 >>> x = 1, 2, 3
2 >>> x
3      (1, 2, 3)
4 >>> a, b, c = x
5 >>> a
6      1
7 >>> b
8      2
9 >>> c
10     3
```

ATENÇÃO

Qual a diferença entre as funções abaixo ?

```
1 def teste1():
2     a = 10
3     a, b = 3 * a, a
4     return a, b
5
6 >>> teste1()
7 (30,10)
```

```
1 def teste2():
2     a = 10
3     a = 3 * a
4     b = a
5     return a, b
6
7 >>> teste2()
8 (30,30)
```

ATENÇÃO

Qual a diferença entre as funções abaixo ?

```
1 def teste1():  
2     a = 10  
3     a, b = 3 * a, a  
4     return a, b  
5  
6 >>> teste1()  
7 (30,10)
```

```
1 def teste2():  
2     a = 10  
3     a = 3 * a  
4     b = a  
5     return a, b  
6  
7 >>> teste2()  
8 (30,30)
```

O lado direito da atribuição é sempre avaliado antes que a atribuição seja feita.

Tuplas

- **Tupla Vazia:** tupla sem elementos.
- **Tupla unitária:** contém um único elemento, que deve ser sucedido por uma vírgula.
- Os parênteses são opcionais se não provocarem ambiguidade.
- Um valor entre parênteses sem vírgula no final é meramente uma expressão.

Qual o tipo de dado da variável A em cada um dos casos abaixo:

```
1 >>> A = ()
2
3 >>> A = (10)
4
5 >>> A = 10,
6
7 >>> A = (10,)
8
9 >>> A = 3*(10+3)
10
11 >>> A = 3*(10+3,)
```

Tuplas

- **Tupla Vazia:** tupla sem elementos.
- **Tupla unitária:** contém um único elemento, que deve ser sucedido por uma vírgula.
- Os parênteses são opcionais se não provocarem ambiguidade.
- Um valor entre parênteses sem vírgula no final é meramente uma expressão.

Qual o tipo de dado da variável A em cada um dos casos abaixo:

```
1 >>> A = ()
2     () # tupla vazia
3 >>> A = (10)
4     10 # inteiro
5 >>> A = 10,
6     (10,) # tupla unitaria
7 >>> A = (10,)
8     (10,) # tupla unitaria
9 >>> A = 3*(10+3)
10    39 # inteiro
11 >>> A = 3*(10+3,)
12    (13, 13, 13) # tupla
```

Tuplas

- Tuplas são muito similares às strings em relação às operações.
- O tamanho de uma tupla é dado pela função `len`.

```
1 >>> x = (1,2,3)
2 >>> len(x)
3      3
```

- **Indexação:** começando do 0 à esquerda, ou de -1 à direita.

```
1 >>> x[0]
2      1
```

- **Fatiamento:** idêntico às strings.

```
1 >>> x[0:2]
2      (1,2)  # NOVA TUPLA
```


Tuplas

● Concatenação e Replicação

```
1 >>> x*2
2     (1,2,3,1,2,3)
3 >>> x + (5,4)
4     (1,2,3,5,4)
```

● Imutabilidade : uma vez criada, uma tupla não pode ser alterada !

```
1 >>> x[0] = 9
2 Traceback (most recent call last):
3   File "<pyshell#2>", line 1, in <module>
4     x[0]=9
5 TypeError: 'tuple' object does not support item assignment
```

Tuplas

Joãozinho quer comprar o maior número de bombons possível com o dinheiro que tem. Faça funções para:

- a. calcular o número de bombons e o troco, dados o dinheiro e o preço de um bombom.

Tuplas

Joãozinho quer comprar o maior número de bombons possível com o dinheiro que tem. Faça funções para:

- a. calcular o número de bombons e o troco, dados o dinheiro e o preço de um bombom.

```
1 def bombom(dinheiro , preco):
2
3     """Os parametros de entrada sao do tipo (float , float).
4     O valor de retorno e do tipo tupla (float , float)"""
5
6     return dinheiro // preco , dinheiro % preco
7
8 >>> bombom(10,3)
9 (3,1)
```

Tuplas

Joãozinho quer comprar o maior número de bombons possível com o dinheiro que tem. Faça funções para:

- a. calcular o número de bombons e o troco, dados o dinheiro e o preço de um bombom.

```
1 def bombom(dinheiro , preco):  
2  
3     """Os parametros de entrada sao do tipo (float , float).  
4     O valor de retorno e do tipo tupla (float , float)"""  
5  
6     return dinheiro // preco , dinheiro % preco  
7  
8 >>> bombom(10,3)  
9 (3,1)
```

- b. calcular quanto Joãozinho terá que pedir para sua mãe para comprar um bombom a mais, dados o dinheiro que ele tem e o preço de um bombom. Utilize a função definida em a.

Tuplas

Joãozinho quer comprar o maior número de bombons possível com o dinheiro que tem. Faça funções para:

- a. calcular o número de bombons e o troco, dados o dinheiro e o preço de um bombom.

```
1 def bombom(dinheiro , preco):
2
3     """Os parametros de entrada sao do tipo (float , float).
4     O valor de retorno e do tipo tupla (float , float)"""
5
6     return dinheiro // preco , dinheiro % preco
7
8 >>> bombom(10,3)
9 (3,1)
```

- b. calcular quanto Joãozinho terá que pedir para sua mãe para comprar um bombom a mais, dados o dinheiro que ele tem e o preço de um bombom. Utilize a função definida em a.

```
1 def maisbombom(dinheiro , preco):
2
3     """Os parametros de entrada sao do tipo (float , float).
4     O valor de retorno e do tipo float"""
5
6     return preco - bombom(dinheiro , preco)[1]
7
8 >>> maisbombom(10,3)
9 2
```

Testes: bombom(10,3) e maisbombom(10,3)

Tuplas

- Resumo: neste vídeo aprendemos o que são Tuplas, como trabalhar com este tipo de dado e alguns exemplos.

Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

Computação I - Python

Aula 4 - Variáveis e Atribuição, Strings e Tuplas

Tuplas

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>