



Programação I

LUCAS SAMPAIO LEITE

Tuplas

- ❑ Tuplas são muito semelhantes às listas, exceto pelo fato de que tuplas são imutáveis;
- ❑ Exemplo: `lista[0] = “qualquer coisa”` não é possível em tuplas;
- ❑ Tuplas não permitem apagar, acrescentar nem realizar atribuições em posições específicas de tuplas já criadas.

De forma geral as tuplas são mais eficientes que as listas pois consomem menos recursos computacionais (memória), por serem estruturas mais simples (Borges, 2014).

Tuplas

❑ Declaração:

```
1  tupla = (0, 1, 2, 'muito parecido com listas')
2  print(tupla)
3
4  # Parenteses são opcionais .
5  tupla = 0, 1, 2, 'muito parecido com listas'
6  print(tupla)
```



```
(0, 1, 2, 'muito parecido com listas ')\n(0, 1, 2, 'muito parecido com listas ')
```

Tuplas

- ❑ Manipulação por índice, tamanho...

```
32 x = (1,2,3)
33 print(x)
34 print(x[0])
35 print(x[0:2])
36 print(len(x))
37 x[0]=2
```



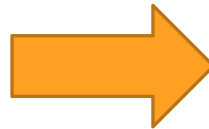
```
(1, 2, 3)
1
(1, 2)
3
Traceback (most recent call last):
  File "/home/lucas/Documents/vscode-projects/lucas.py", line 37,
in <module>
    x[0]=2
TypeError: 'tuple' object does not support item assignment
```

Tuplas

- ❑ Para criar-se uma Tupla com apenas um elemento, deve-se escrever:
`nome_da_variável = (elemento,)`.
 - ❑ É necessário se escrever a vírgula pois caso contrário, o interpretador irá entender que a variável `nome_da_variável` é apenas uma variável do mesmo tipo do elemento.
- ❑ Para uma tupla vazia, se escreve `nome_da_variável = ()` ou, `nome_da_variável = tuple();`

Tuplas

```
8  tupla = 1,  
9  print(tupla)  
10 print(type(tupla))  
11  
12 tupla = 1  
13 print(tupla)  
14 print(type(tupla))
```



```
(1,)  
<class 'tuple'>  
1  
<class 'int'>
```

Tuplas

- ❑ Os parênteses são opcionais se não provocarem ambigüidade. O que seria impresso?

```
tupla = ()  
print(tupla)  
print(type(tupla))
```

```
tupla = 3*(10+3)  
print(tupla)  
print(type(tupla))
```

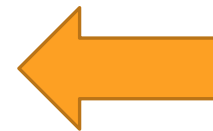
```
tupla = 3*(10+3,)  
print(tupla)  
print(type(tupla))
```

```
tupla = 3*10+3,  
print(tupla)  
print(type(tupla))
```


Tuplas

- ❑ Os parênteses são opcionais se não provocarem ambigüidade. O que seria impresso?

```
()  
<class 'tuple'>  
39  
<class 'int'>  
(13, 13, 13)  
<class 'tuple'>  
(33,)  
<class 'tuple'>
```



```
tupla = ()  
print(tupla)  
print(type(tupla))
```

```
tupla = 3*(10+3)  
print(tupla)  
print(type(tupla))
```

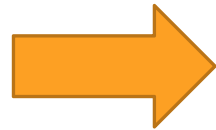
```
tupla = 3*(10+3,)  
print(tupla)  
print(type(tupla))
```

```
tupla = 3*10+3,  
print(tupla)  
print(type(tupla))
```


Tuplas

- ❑ É possível se converter listas em tuplas:

```
16  lista = [0,1,2,3,4]
17  print(lista)
18  tupla = tuple(lista)
19  print(tupla)
20  lista = list(tupla)
21  print(lista)
```



```
[0, 1, 2, 3, 4]
(0, 1, 2, 3, 4)
[0, 1, 2, 3, 4]
```

Tuplas

- ❑ O mesmo pode acontecer para uma string:

```
1  tupla = (0, 1, 2, 'muito parecido com listas')
2  print(tupla)
3  string = str(tupla)
4  print(string)
5  print(type(string))
```

```
(0, 1, 2, 'muito parecido com listas')
(0, 1, 2, 'muito parecido com listas')
<class 'str'>
```

A função `str()` irá transformar tudo o que representa a tupla (inclusive os parênteses) em uma string unicode.

Tuplas

- ❑ Embora uma tupla seja imutável, esta pode ser formada por elementos mutáveis (como listas) e então esses elementos podem ser atualizados.

```
32  tupla = [0, 1], 'a', 'b'
33  print(tupla)
34  tupla[0].append(2)
35  print(tupla)
36  tupla [0] = [0, 1, 2]
```



```
([0, 1], 'a', 'b')
```

```
([0, 1, 2], 'a', 'b')
```

```
Traceback (most recent call last):
```

```
  File "/home/lucas/Documents/vscode-projects/lucas.py", line 36,
in <module>
```

Tuplas

- ❑ É permitida a concatenação de tuplas.

```
1  tupla1 = (1, 2, 3)
2  print(tupla1)
3  tupla1 = (1,2,3) + (4,5,6)
4  print(tupla1)
```



```
(1, 2, 3)
(1, 2, 3, 4, 5, 6)
```

Operadores básicos sobre tuplas...

Expressão	Resultado	Descrição
<code>len((1,2,3))</code>	3	Número de elementos que a tupla contém
<code>(1, 2, 3) + (4, 5, 6)</code>	<code>(1, 2, 3, 4, 5, 6)</code>	Concatenação
<code>(1,) * 4</code>	<code>(1,1,1,1)</code>	Repetição
<code>3 in (1, 2, 3)</code>	True	Pertencimento
<code>for x in (1,2,3): print(x)</code>	1 2 3	Iteração

Lista de tuplas

```
1 cliente1 = (1, 'Lucas', '999.999.999-99', '9.999.999')
2 cliente2 = (2, 'Maria', '888.888.888-88', '8.888.888')
3
4 clientes = [cliente1, cliente2]
```

O que ganhamos com essa abordagem???

Quando usar tuplas?

- ❑ Em geral, tuplas podem ser substituídas com vantagem por listas (flexibilidade no lugar de eficiência);
- ❑ Entretanto, algumas construções em Python requerem tuplas ou seqüências imutáveis, por exemplo:
 - ❑ Tuplas (ao contrário de listas) podem ser usadas como chaves de dicionários;
 - ❑ Funções com número variável de argumentos acessam os argumentos por meio de tuplas;
 - ❑ Representar coleções de elementos que não devem ser modificados, como coordenadas geográficas, informações fixas de uma entidade, entre outros.

Exercícios

1. Escreva um programa em que dada uma tupla, remova os valores duplicados.

Exemplo: tupla = (1, 3, 5, 2, 3, 5, 1, 1, 3)

Resultado: A tupla original é: (1, 3, 5, 2, 3, 5, 1, 1, 3)

A tupla após a remoção de duplicatas: (1, 2, 3, 5)

Obs: Não converter a tupla para nenhum outro tipo.

Exercícios

2. Dada uma lista de números, escreva um programa Python para criar uma lista de tuplas tendo o primeiro elemento como o número e o segundo elemento como o cubo do número.

Exemplo:

```
Entrada: lista = [1, 2, 3]  
Resultado: [(1, 1), (2, 8), (3, 27)]  
Entrada: lista = [9, 5, 6]  
Resultado: [(9, 729), (5, 125), (6, 216)]
```

Exercícios

3. Contagem de ocorrências: Dado um texto armazenado em uma string, crie um programa que conte a quantidade de ocorrências de cada palavra e retorne uma tupla contendo a palavra e o número de ocorrências.
 - Exemplo: Para o texto: "A casa é bonita, a casa é azul", a saída seria (('A', 1), ('casa', 2), ('é', 2), ('bonita,', 1), ('a', 1), ('azul', 1)).
4. Registro de alunos: Crie um programa que registre os dados dos alunos de uma turma, incluindo nome, idade e notas. Utilize uma lista de tuplas para armazenar os registros dos alunos e permita a busca de um aluno pelo nome.

Dúvidas???



Fonte: <https://institutoseculoxxi.com.br/duvidas-entramos-em-contato-com-voce/>