



161

Introdução

- As tuplas são iguais às listas, com a exceção de que os dados, uma vez inseridos na tupla, não podem ser alterados, aconteça o que acontecer.
- A única exceção é quando os dados dentro da tupla são mutáveis, somente então os dados da tupla podem ser alterados.
- Por fim, para declarar uma tupla você pode utilizar parênteses ou deixar sem eles (parênteses são opcionais).
 - Entretanto é uma boa prática usá-los

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

162

Criando uma tupla

- Exemplo:

```
#tupla vazia
minhatupla = ()
print(minhatupla)

#tupla de inteiros
minhatupla = (1, 2, 3)
print(minhatupla)

#tupla com diferentes dados
minhatupla = (1, "Hello World", 3.4)
print(minhatupla)

#tupla com diferentes estruturas internamente
minhatupla = ("python", [8, 4, 6], (1, 2, 3))
print(minhatupla)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

163

Criando uma tupla (sem os parênteses)

- Uma tupla pode ser criada sem o uso de parênteses, é o que chamamos de empacotamento de tupla

```
minhatupla = 3, 4.6, "cachorro"
print(minhatupla)

# desempacotamento
a, b, c = minhatupla

print(a)      # 3
print(b)      # 4.6
print(c)      # cachorro
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

164

Criando uma tupla (sem os parênteses)

- Criar uma tupla com um elemento é um pouco complicado.
- Ter um elemento entre parênteses não é suficiente.
- Precisaremos de uma vírgula final para indicar que é, de fato, uma tupla.

```
minhatupla = ("hello world")
print(type(minhatupla)) # <class 'str'>

minhatupla = ("hello world",)
print(type(minhatupla)) # <class 'tuple'>

minhatupla = "hello world",
print(type(minhatupla)) # <class 'tuple'>
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

165

Acessando elemento na tupla

- O acesso aos elementos é igual ao acesso aos valores nas listas:

```
minhatupla = (1, 2, 3, 'texto')
for x in minhatupla:
    print(x)
print(minhatupla)
print(minhatupla[0])
print(minhatupla[:])
print(minhatupla[3])
print(minhatupla[-1])
print(minhatupla[-2][-4])
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

166

Mudando uma tupla

- “Mas você disse que tupla é imutável!”
- Sim, e é. Ao contrário das listas, as tuplas são imutáveis.
- Isso significa que os elementos de uma tupla não podem ser alterados depois de atribuídos.
 - Mas, se o próprio elemento for um tipo de dados mutável, como uma lista, seus itens aninhados podem ser alterados.
- Também podemos atribuir uma tupla a diferentes valores (redesignação).

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

167

Mudando uma tupla

- Exemplo:

```
tupla = (4, 2, 3, [6, 5])
tupla = 9
# Saída: TypeError: 'tuple' object does not support item assignment

tupla[3][0] = 9    # Saída: (4, 2, 3, [9, 5])
print(tupla)

# Redesignando o valor de uma tupla
tupla = ('p', 'y', 't', 'h', 'o', 'n')
print(tupla)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

168

Mudando uma tupla

- Concatenação:
 - Podemos usar o operador + para combinar duas tuplas.
 - Também podemos repetir os elementos em uma tupla por um determinado número de vezes usando o operador *.
 - Ambas as operações + e * resultam em uma nova tupla.
 - Ou seja, não estamos modificando uma tupla, mas sim, criando uma nova tupla

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

169

Mudando uma tupla

- Exemplo:

```
a = (1, 2, 3)
b = (4, 5, 6)
tuplax = a + b
print(tuplax)

tupla = ("python",) * 3
print(tupla)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

170

Excluindo uma tupla

- Conforme apresentado, não podemos alterar os elementos em uma tupla.
 - Logo, não podemos excluir ou remover itens de uma tupla.
- A exclusão de uma tupla inteiramente, no entanto, é possível usando a palavra-chave `del`.

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

171

Mudando uma tupla

- Exemplo:

```
minhatupla = ('p', 'y', 't', 'h', 'o', 'n')
del minhatupla
print(minhatupla)
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

172

Métodos de tupla

- Para manipulação de tuplas, existem apenas dois métodos: `count` e `index`, veja:

```
tupla = ('a', 'e', 'i', 'o', 'o',)  
print(tupla.count('o'))  
print(tupla.index('i'))
```

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

173

Por hoje é só!

Orientações

- Os códigos de exemplo desta aula
- Pratique os exercícios desta aula
- Somente com a prática constante você será fluente em Python!

Qualquer dúvida:

- www.arieldias.com
- contato@arieldias.com

Bons estudos! Até a próxima!

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida reprodução e uso sem autorização

174

Referências para esta aula

- GURU, 99. Python. Disponível em: <https://www.guru99.com/python-tutorials.html>. Acesso em: 02 fev. 2021.
- PROGRAMIZ. Python. Disponível em: <https://www.programiz.com/python-programming>. Acesso em: 02 fev. 2021.

Professor Ariel Dias - www.arieldias.com - Obra licenciada, proibida
reprodução e uso sem autorização