

Listas e tuplas juntas

Transcrição

Nesse vídeo veremos mais sobre tuplas e listas.

Tuplas dentro de uma lista

Agora queremos armazenar instrutores, cada um com o seu nome e sua idade:

```
>>> instrutor1 = ("Nico", 39)
>>> instrutor2 = ("Flávio", 37)
```

Agora podemos criar uma **lista de instrutores**:

```
>>> instrutor1 = ("Nico", 39)
>>> instrutor2 = ("Flávio", 37)
>>> instrutores = [instrutor1, instrutor2]
```

Isso mesmo! A lista pode armazenar tuplas dentro dela. Ao imprimi-la, vemos:

```
>>> instrutores
[('Nico', 39), ('Flávio', 37)]
```

Podemos acessar somente um tupla pelo seu índice na lista:

```
>>> instrutores[0]
('Nico', 39)
```

E podemos também acessar, através da lista, somente um elemento da tupla. Por exemplo, para acessar a idade do Nico, acessamos a sua tupla, e acessamos a sua idade passando mais um índice:

```
>>> instrutores[0][1]
39
```

E podemos fazer o contrário também, podemos colocar listas dentro de tuplas :)

Convertendo listas em tuplas

Agora temos a seguinte situação: precisamos ler de um arquivo, mas não sabemos quantas linhas esse arquivo tem. Então, vamos lendo linha por linha, e adicionando-as em uma lista:

```
>>> linhas = []
>>> linhas.append("linha 1")
```

```
>>> linhas.append("linha 2")
>>> linhas.append("linha 3")
```

Em algum momento o arquivo irá acabar, e quando esse momento chegar, não queremos mais adicionar linhas à lista. Então, devemos transformar a lista em uma lista imutável, no caso uma tupla. E o Python possui uma função específica para isso, a `tuple()`, que recebe por parâmetro a lista a ser convertida:

```
>>> linhas_tuple = tuple(linhas)
>>> type(linhas_tuple)
<class 'tuple'>
>>> linhas_tuple
('linha 1', 'linha 2', 'linha 3')
```

Convertendo tuplas em listas

Com uma simples função, transformamos uma lista em uma tupla! E o contrário também pode ser feito, com a função `list()`:

```
>>> linhas_list = list(linhas_tuple)
>>> type(linhas_list)
<class 'list'>
>>> linhas_list
['linha 1', 'linha 2', 'linha 3']
```