



Tuplas em Python

Em Python, as tuplas são estruturas de dados semelhantes às listas, porém, ao contrário das listas, as tuplas são imutáveis, o que significa que seus elementos não podem ser modificados após a criação. As tuplas são frequentemente usadas para armazenar coleções de elementos relacionados. Aqui estão algumas informações e exemplos sobre tuplas em Python:

1- Criando uma tupla:

As tuplas são criadas utilizando parênteses ou, opcionalmente, sem parênteses, separando os elementos por vírgulas. Por exemplo:

```
minha_tupla = (1, 2, 3, 4, 5)
outra_tupla = 1, 2, 3 # também é uma tupla válida
```

2- Acessando elementos de uma tupla:

Os elementos de uma tupla podem ser acessados usando índices, assim como nas listas. O índice começa a partir de zero. Por exemplo:

```
minha_tupla = (1, 2, 3, 4, 5)
primeiro_elemento = minha_tupla[0] # retorna 1
segundo_elemento = minha_tupla[1] # retorna 2
```



3- Imutabilidade:

Ao contrário das listas, as tuplas são imutáveis, o que significa que você não pode modificar seus elementos após a criação da tupla. Por exemplo:

```
minha_tupla = (1, 2, 3)
minha_tupla[0] = 10 # Isso resultará em um erro, pois as tuplas são imutáveis
```

4- Tamanho da lista:

É possível obter o tamanho (número de elementos) de uma lista usando a função `len()`. Por exemplo:

```
minha_lista = [1, 2, 3, 4, 5]
tamanho = len(minha_lista) # retorna 5
```

5- Desempacotamento de tuplas:

É possível atribuir os elementos de uma tupla a variáveis individuais em um processo chamado desempacotamento. Por exemplo:

```
minha_tupla = (1, 2, 3)
a, b, c = minha_tupla
print(a) # imprime 1
print(b) # imprime 2
print(c) # imprime 3
```

Embora as tuplas sejam imutáveis, elas são úteis para armazenar coleções de dados relacionados quando você não precisa alterar seus elementos posteriormente.