



Listas e Tuplas em Python:

Listas e tuplas são dois tipos de estruturas de dados em Python que permitem armazenar múltiplos valores em uma única variável. Ambas são sequências ordenadas de elementos, mas têm algumas diferenças importantes em termos de mutabilidade e sintaxe. Vamos explorar cada uma delas:

Listas:

- Uma lista é definida por colchetes `[]` e seus elementos são separados por vírgulas.
- As listas são mutáveis, o que significa que os elementos podem ser alterados, adicionados ou removidos após a criação da lista.
- Os elementos de uma lista podem ter tipos de dados diferentes, ou seja, uma lista pode conter números, strings, booleanos, outras listas e até mesmo objetos personalizados.
- Os elementos em uma lista são acessados por meio de um índice baseado em zero.
- As listas permitem operações como adicionar elementos (`append`), remover elementos (`remove`), contar elementos (`count`), ordenar elementos (`sort`), entre outros.

Exemplo de lista:

```
frutas = ['maçã', 'banana', 'laranja']
```



Tuplas:

- Uma tupla é definida por parênteses () ou apenas pela separação dos elementos por vírgulas.
- As tuplas são imutáveis, o que significa que, uma vez criadas, seus elementos não podem ser alterados. No entanto, é possível criar uma nova tupla com elementos modificados.
- Assim como as listas, os elementos de uma tupla podem ter tipos de dados diferentes.
- Os elementos em uma tupla são acessados por meio de um índice baseado em zero, assim como nas listas.
- As tuplas podem ser usadas para representar coleções de dados que não devem ser alterados, como coordenadas geográficas, dias da semana, entre outros.

Exemplo de tupla:

```
coordenadas = (10, 20)
```

Operações comuns em listas e tuplas:

- Acessar elementos: Utilize o índice entre colchetes para acessar um elemento específico da lista ou tupla.

```
lista = [1, 2, 3]
```

```
elemento = lista[0] # Acessa o primeiro elemento da lista (1)
```



Verificar o comprimento:

Utilize a função `len()` para obter o número de elementos em uma lista ou tupla.

```
lista = [1, 2, 3]
comprimento = len(lista) # Retorna 3
```

Percorrer elementos:

Utilize um loop `"for"` para percorrer todos os elementos de uma lista ou tupla.

```
lista = [1, 2, 3]
for elemento in lista:
    print(elemento) # Imprime 1, 2, 3 em linhas separadas
```

Concatenação:

É possível concatenar duas listas ou tuplas usando o operador `+`.

```
lista1 = [1, 2, 3]
```

```
lista2 = [4, 5, 6]
```

```
lista_concatenada = lista1 + lista2 # Retorna [1, 2, 3, 4, 5, 6]
```



Essas são as principais características e operações relacionadas a listas e tuplas em Python. Elas são amplamente utilizadas para armazenar e manipular coleções de dados de forma eficiente.