



Tuplas

Tupla é uma estrutura de dados semelhante a lista. Porém, ela tem a característica de ser imutável, ou seja, após uma tupla ser criada, ela não pode ser alterada.



```
1 faixas_karate = ('Branca', 'Amarela', 'Vermelha', 'Laranja', 'Verde', 'Roxa', 'Marrom', 'Preta')
2
3 print(type(faixas_karate)) # <class 'tuple'>
4 print(faixas_karate) # ('Branca', 'Amarela', 'Vermelha', 'Laranja', 'Verde', 'Roxa', 'Marrom', 'Preta')
```



Tuplas

Assim como é feito nas listas, podemos acessar um determinado valor na tupla pelo seu índice:

```
1 faixas_karate = ('Branca', 'Amarela', 'Vermelha', 'Laranja', 'Verde', 'Roxa', 'Marrom', 'Preta')
2
3 print(type(faixas_karate)) # <class 'tuple'>
4 print(faixas_karate[3]) # Laranja
```




Tuplas

Uma observação a ser feita no uso de uma tupla é que se ela tiver um único item, é necessário colocar uma vírgula depois dele, pois caso contrário, o objeto que vamos obter é uma string, porque o valor do item é do tipo string.

```
1 objeto_string = ('mouse')
2 objeto_tupla = ('mouse',)
3
4 print(type(objeto_string)) # <class 'str'>
5 print(type(objeto_tupla)) # <class 'tuple'>
```



Tuplas

O fato da tupla ser imutável faz com que os seus elementos não possam ser alterados depois dela já criada. Vamos usar a tupla vogais para mostrar um exemplo desse tipo. O Código exibirá o erro *TypeError: 'tuple' object does not support item assignment*.

```
1  vogais = ('a', 'e', 'i', 'o', 'u')
2
3  vogais[1] = 'E'
4
5  TypeError: 'tuple' object does not support item assignment
```




Tuplas

As tuplas devem ser usadas em situações em que não haverá necessidade de adicionar, remover ou alterar elementos de um grupo de itens. Exemplos bons seriam os meses do ano, os dias da semana, graduações de faixa, as estações do ano etc. Nesses casos, não haverá mudança nesses itens.



```
1 lista = ['Victor', 'Juliana', 'Samuel', 'Caio', 'Luana']
2 tupla = ('a', 'e', 'i', 'o', 'u')
3 print(lista) # ['Victor', 'Juliana', 'Samuel', 'Caio', 'Luana']
4 print(tupla) # ('a', 'e', 'i', 'o', 'u')
```



Dicionários

Os dicionários representam coleções de dados que contém na sua estrutura um conjunto de pares chave/valor, nos quais cada chave individual tem um valor associado. Esse objeto representa a ideia de um mapa, que entendemos como uma coleção associativa desordenada. A associação nos dicionários é feita por meio de uma chave que faz referência a um valor.

```
1 dados_faculdade = {  
2     'Nome': 'Universidade de Vassouras',  
3     'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo – Maricá/RJ',  
4     'Telefone': '(21) 3731-2977'  
5 }  
6  
7 print(dados_faculdade['Nome']) # Universidade de Vassouras
```




Dicionários

```
1 dados_faculdade = {  
2     'Nome': 'Universidade de Vassouras',  
3     'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo – Maricá/RJ',  
4     'Telefone': '(21) 3731-2977'  
5 }  
6  
7 print(dados_faculdade['Nome']) # Universidade de Vassouras
```

A estrutura de um dicionário é delimitada por chaves, entre as quais ficam o conteúdo desse objeto. Veja que é criada a variável `dados_faculdade`, à qual é atribuída uma coleção de dados que, nesse caso, trata-se de um dicionário. Na linha 7, imprimimos o conteúdo que é associado ao índice “Nome”, trazendo o resultado Universidade de Vassouras.



Dicionários

Nas listas e tuplas acessamos os dados por meio dos índices. Já nos dicionários, o acesso aos dados é feito por meio da chave associada a eles. Para adicionar elementos num dicionário basta associar uma nova chave ao objeto e dar um valor a ser associado a ela. No **Código anterior** vamos colocar a informação Instagram em `dados_faculdade`.

```
1 dados_faculdade = {  
2     'Nome': 'Universidade de Vassouras',  
3     'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo - Maricá/RJ',  
4     'Telefone': '(21) 3731-2977'  
5 }  
6  
7 print(dados_faculdade['Nome']) # Universidade de Vassouras  
8 dados_faculdade['Instagram'] = 'https://www.instagram.com/universidadedevassouras/'  
9  
10 print(dados_faculdade) # {'Nome': 'Universidade de Vassouras', 'Endereco':  
11 'Av. Governador Roberto Silveira, nº 437, Flamengo - Maricá/RJ', 'Telefone': '(21) 3731-2977', 'Instagram': 'https://www.instagram.com/universidadedevassouras/'}
```




Dicionários

Para remover um item do dicionário, podemos usar o método `pop()`, como vemos abaixo:

```
1 dados_faculdade = {
2     'Nome': 'Universidade de Vassouras',
3     'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo - Maricá/RJ',
4     'Telefone': '(21) 3731-2977'
5 }
6
7 print(dados_faculdade['Nome']) # Universidade de Vassouras
8 dados_faculdade['Instagram'] = 'https://www.instagram.com/universidadedevassouras/'
9
10 print(dados_faculdade) # {'Nome': 'Universidade de Vassouras', 'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo - Maricá/RJ', 'Telefone': '(21) 3731-2977', 'Instagram': 'https://www.instagram.com/universidadedevassouras/'}
11
12 dados_faculdade.pop('Instagram')
13
14 print(dados_faculdade) # {'Nome': 'Universidade de Vassouras', 'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo - Maricá/RJ', 'Telefone': '(21) 3731-2977'}
```



Dicionários

Também poderíamos usar a palavra-chave **del**, que remove uma chave e o valor associado a ela no dicionário. Isso se faz por meio da passagem no parâmetro

```
1 dados_faculdade = {  
2     'Nome': 'Universidade de Vassouras',  
3     'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo – Maricá/RJ',  
4     'Telefone': '(21) 3731-2977'  
5 }  
6  
7 print(dados_faculdade['Nome']) # Universidade de Vassouras  
8  
9 del dados_faculdade['Telefone']  
10  
11 print(dados_faculdade) # {'Nome': 'Universidade de Vassouras', 'Endereco': 'Av. Governador Roberto Silveira, nº 437, Flamengo – Maricá/RJ'}
```