

FAP 2024.1

Formação Acelerada em Programação

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO

MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Back end A5 (01 a 05)

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Tupla

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



Tuplas são estruturas de dados imutáveis em Python, o que significa que uma vez criadas, não podem ser alteradas. São úteis para armazenar coleções ordenadas de itens que não precisam ser modificados.

Exemplo:

tupla = (1,2,3,4,5)

Operações comuns com tuplas:

- Acessar elementos por índice: `tupla[0]` retorna o primeiro elemento.

```
semana5 > 🐍 tupla.py > ...
1  tupla = (1,2,3,4,5)
2  |
3  #Primeiro indice da tupla
4  print(tupla[0])
5
```

- Iterar sobre os elementos: `for item in tupla:`

```
8  for lista in tupla:
9  |     print("Tupla: ", lista)
```

Exemplo:

tupla = (1,2,3,4,5)

Operações comuns com tuplas:

- Concatenar tuplas: `tupla1 + tupla2`.

```
concatenando = tupla + tupla2

for lista in concatenando:
    print(lista)
```

Exercício:

Criar uma tupla.

Uf do Brasil.

Armazenar as UF na tupla.

Listar todas as UF.

Conjuntos

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



Conjuntos em Python são coleções de elementos únicos e não ordenados. São úteis para realizar operações como união, interseção e diferença entre coleções.

Exemplo:

Operações comuns com conjuntos:

- Adicionar elementos: `conjunto.add(valor)`.
- Remover elementos:
- `conjunto.remove(valor)`.

```
1 conjuntoA = {1, 2, 3, 4, 5}
2
3 conjuntoB = {5, 6, 7, 8}
4
5 print(conjuntoA)
6 conjuntoA.remove(2)
7 print(conjuntoA)
```

```
1 conjunto = {1, 2, 3, 4, 5}
2
3
4 print(conjunto)
```

```
semana5 > conjunto.py > ...
1 conjunto = {1, 2, 3, 4, 5}
2
3
4 print(conjunto)
5
6 conjunto.add(6)
7
8 print(conjunto)
```

Dicionário

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



Dicionários são estruturas de dados que mapeiam chaves a valores. São muito eficientes para buscas rápidas e associações de informações.

```
semana5 > 🐍 dicionario.py > ...  
1  dicionario = {"chave1":1, "chave2": 2, "chave3": 3}  
2  
3  print(dicionario)  
4
```


Exemplo:

Operações comuns com dicionários:

- Acessar valor por chave: `dicionario['chave']`.

```
semana5 >  dicionario.py > ...  
1  dicionario = {"chave1":1, "chave2": 2, "chave3": 3}  
2  
3  print(dicionario["chave1"])  
4
```

- Adicionar novo par chave-valor:
`dicionario['nova_chave'] = novo_valor.`

Exemplo:

Operações comuns com dicionários:

- Adicionar novo par chave-valor:

```
dicionario['nova_chave'] = novo_valor.
```

```
semana5 > 💻 dicionario.py > ...  
1  dicionario = {"chave1":1, "chave2": 2, "chave3": 3}  
2  
3  print(dicionario)  
4  
5  dicionario['chave4'] = 4  
6  
7  print(dicionario["chave4"])  
8
```

Exemplo:

Operações comuns com dicionários:

- Iterar sobre chaves: `for chave in dicionario:.`

```
9  for lista in dicionario:  
10 |     print(lista)
```

Criar um dicionário que armazene a UF e a descrição da UF.

Inserir, remover e listar os dados contidos no dicionário.

Começar o projeto Sistema de cadastro.

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Back end A5 (03/07)

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



Objeto

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



Em Python, tudo é um **objeto**. Além dos tipos de dados nativos (int, str, etc.), é possível criar classes para definir novos tipos de objetos. As classes definem atributos e métodos que operam sobre esses atributos.

Exemplo:

```
semana5 > objeto.py > ...
1  class Pessoa:
2      def __init__(self, nome, idade):
3          self.nome = nome
4          self.idade = idade
5
6      def saudacao(self):
7          return f'Olá, meu nome é {self.nome} e eu tenho {self.idade} anos.'
8
9  # Exemplo de uso da classe Pessoa
10  pessoa1 = Pessoa('Maria', 30)
11  print(pessoa1.saudacao()) # Saída: Olá, meu nome é Maria e eu tenho 30 anos.
```

Exercícios de fixação.

<https://github.com/isquicha/exercicios-logica-python/blob/main/exercicios/012.py>

INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO

