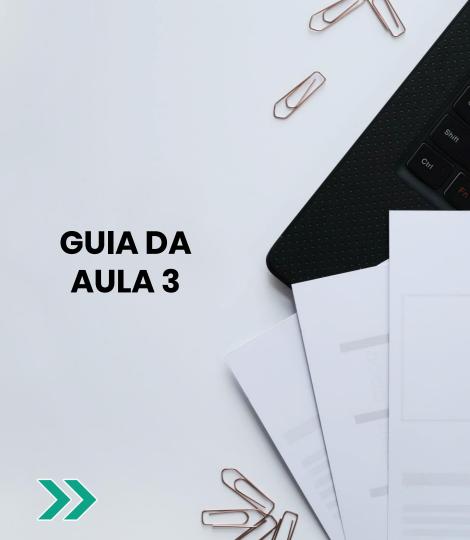


Profissão: Analista de dados





ESTRUTURA DE DADOS







Forme dicionários

Motivação

Definição

- Operações
- Métodos
- Conversão
- Revisitando a motivação



Acompanhe aqui os temas que serão tratados na videoaula





1. Motivação

Para se conectar a uma rede wi-fi, você precisa de duas informações: o nome da rede e a senha de acesso. Quando você vai acessar uma nova rede, você encontra uma lista de redes disponíveis:

```
In []: wifi_disponiveis = ['rede1', 'cnx_cnx', 'uai-fi', 'r3d3']
    print(wifi_disponiveis)
```

Você consegue identificar quais são os nome de redes e suas respectivas senhas? Talvez uma list não seja a melhor opção para armazenar esse tipo de dado.





2. Definição

Armazenam sequências no formato chave-valor. São do tipo dict:

```
In []: brasil = {'capital': 'Brasília', 'idioma': 'Português', 'populacao': 210}
    print(brasil)
    print(type(brasil))
```

Não é permite chaves duplicadas.

```
In []: carro = {
    'marca': 'Volkswagen',
    'modelo': 'Polo',
    'ano': 2021,
    'ano': 2004
}

print(carro)
```





2. Definição

Podemos criar dicionários compostos:

```
In [ ]:
         cadastro = {
              'andre': {
                  'nome': 'Andre Perez',
                  'ano nascimento': 1992,
                  'pais': {
                      'pai': {
                        'nome': '<nome-do-pai> Perez',
                        'ano nascimento': 1971
                      'mae': {
                        'nome': '<nome-da-mae> Perez',
                        'ano nascimento': 1973
                      },
         print(cadastro)
```





2. Definição

```
In [ ]: cadastro['andre']['pais']['mae']['ano_nascimento']
```





3. Operações

```
In [ ]: credito = {'123': 750, '789': 980}
```

Elementos são acessados pela sua chave.

```
In []: score_123 = credito['123']
    score_789 = credito['789']

    print(score_123)
    print(score_789)
```

Elementos são atualizados pela sua chave.

```
In []: credito['123'] = 435
    print(credito)
```





3. Operações

Para adicionar um novo elemento, basta criar um novo elemento chave-valor:

```
In []: credito['456'] = 1000
   print(credito)
```





4. Métodos

São métodos nativos do Python que nos ajudam a trabalhar no dia a dia com dicionários.

```
In [ ]:
         artigo = dict(
              titulo='Modulo 02 | Python: Estruturas de Dados',
              corpo='Topicos, Aulas, Listas, Conjuntos, Dicionários, ...',
              total caracteres=1530
In [ ]:
         # adicionar/atualizar um elemento pelo chave-valor: dict.update(dict)
         print(artigo)
         artigo.update({'total caracteres':
         7850}) print(artigo)
         artigo['total caracteres'] = 7850
In [ ]:
         # remover um elemento pelo chave: dict.pop(key)
         print(artigo)
         total caracteres = artigo.pop('total caracteres')
         print(artigo)
```





5. Conversão

Podemos converter as chaves e os itens de um dicionário em uma lista.

```
In [ ]:
         artigo = dict(
              titulo='Modulo 02 | Python: Estruturas de Dados',
              corpo='Topicos, Aulas, Listas, Conjuntos, Dicionários, ...',
              total caracteres=1530
In [ ]:
         chaves = list(artigo.keys())
         print(chaves)
         print(type(chaves))
In [ ]:
         valores = list(artigo.values())
         print(valores)
         print(type(valores))
```





6. Revisitando a motivação

```
In []: wifi_disponiveis = []

In []: rede = {'nome': 'rede1', 'senha': 'cnx_cnx'}
    wifi_disponiveis.append(rede)

In []: rede = {'nome': 'uai-fi', 'senha': 'r3d3'}
    wifi_disponiveis.append(rede)

In []: print(wifi_disponiveis)
```

