

Dicionários



Dicionários

Segundo Borges (2010), um dicionário é uma lista de associações composta por uma chave (de tipo imutável) e estruturas correspondentes às chaves que podem ser mutáveis ou não. Isso caracteriza os Dicionários como sendo mutáveis (pois os valores em si são mutáveis, exceto as chaves).

Dicionários

Dicionários são estruturas de dados similares às listas, porém, com propriedades de acesso diferentes. Um dicionário consiste em um conjunto de chaves e valores.

Sintaxe:

```
diccionario = { 'a': 1, 'b': 4, 'z': 'z' }  
  
diccionario = { chave: valor, chave: valor }
```

Assim como as listas, os dicionários são mutáveis. Mas suas chaves devem ser de um tipo imutável.

Para criar listas usamos colchetes “[]” para criar tuplas usamos parênteses “()” para criar dicionários usamos chaves “{}”

Dicionários

| Chave | Valor |
|-----------|----------|
| Curso | Python |
| Aula | Aula07 |
| Professor | Fernando |

```
dic = { 'curso': 'Python',  
        'aula': 'Aula07',  
        'professor': "Fernando" }
```

Listas: Acessamos os elementos pelo número do índice

Dicionários: Acessamos um elemento utilizando a chave

```
print(f'Professor {dic["professor"]} que ministra o curso {dic["curso"]}') )
```

Dicionários

Para adicionar/alterar elementos ao dicionário usamos dicionário [" valor"] = valor

Para apagar um elemento de um dicionário usamos: dicionario.clear()

Para apagar o dicionário usamos: del dicionário, deste modo não existe mais.

```
dic["aula"] = "Aula07 - Dicionários"  
dic.clear()  
print(dic)  
del dic
```

Obtendo itens, chaves e valores de um dicionário

key(), items() e values() retornam views (visualizações), que são iteradores dos tipos dict_keys dict_items e dict_values que devolvem um elemento de cada vez.

```
carros = {'marca': 'VW', 'modelo': 'Gol', 'ano_modelo': 2016}
print(f'Itens do dicionário carros : {carros.items()}')
print(f'Chaves do dicionário carros : {carros.keys()}')
print(f'Valores do dicionário carros : {carros.values()}')
```

Exemplo

```
produtos = { "Mouse": 98.75,  
             "Teclado": 125.65,  
             "Monitor": 134.78,  
             "Gabinete": 170.00,  
             "HD Externo": 510.50,  
             "Headset": 125.45 }  
  
while True:  
    produto = input("Informe o produto a pesquisar o preço ou fim para sair: ")  
    if produto == "fim":  
        break  
    if produto in produtos:  
        print(f"Produto {produto} custa {produtos[produto]}.")  
    else:  
        print(f"Produto {produto} não encontrado.")
```

Em Python podemos ter
dicionários contendo listas ou
até mesmo outros dicionários

Outras funções

- Copiando um dicionário para outro
- Adicionando conteúdo de um dicionário em outro dicionário
- Para retornar quantos conjuntos chave/valor existem no dicionário usamos a função `len()`.

```
dic = {'nome': "Fulano", 'sobrenome': 'de Tal'}  
local = {'UF': "SP", 'cidade': 'São Carlos'}  
dic2 = dic.copy()  
dic.update(local)  
print(len(dic))  
print(f'dic: {dic}')print(f'dic2: {dic2}')
```

Exemplo 2

```
aluno = {}
aluno['nome'] = str(input("Nome: "))
aluno['media'] = float(input(f"Media de {aluno['nome']}"))
if aluno['media'] >= 7:
    aluno['situacao'] = 'Aprovado'
elif aluno['media'] >= 5:
    aluno['situacao'] = 'Recuperação'
else:
    aluno['situacao'] = 'Reprovado'
print('=='*30)
for k, v in aluno.items():
    print(f' - {k} é igual a {v}')
```