

CC1612

Fundamentos de Algoritmos

Prof. Danilo H. Perico

Dicionários

Dictionaries

Dicionários

- Dicionários (em Python) são **vetores associativos**
- Vetores associativos são coleções **desordenadas** de dados, usadas para **armazenar** valores como um **mapa**: por meio de elementos formados pelo par **chave** e **valor**
- Assim, diferentemente das listas ou tuplas, que contém um único valor como elemento, o dicionário contém o par **chave:valor** (**key:value**)
 - **Chave (key)**: serve para deixar o dicionário otimizado
 - **Valor (value)**: valor do elemento associado a uma chave

Dicionários

- Dicionários diferem das listas essencialmente na maneira como os elementos são acessados:
 - **Listas**: valores são acessados por sua posição dentro da lista, via índice
 - **Dicionários**: valores são acessados por meio de suas **chaves** (*keys*)

Dicionários

- Um dicionário em Python funciona de forma semelhante ao dicionário de palavras:
 - As chaves (*keys*) de um dicionário devem ser exclusivas e com o tipo de dados imutáveis, como strings, inteiros ou tuplas
 - Porém, os valores (*values*) associados às chaves podem ser repetidos e de qualquer tipo

Dicionários

- Para criarmos um dicionário, devemos incluir uma sequência de elementos dentro de chaves `{ }`, separados por vírgula.
- A chave e o valor são separados por dois pontos `:`
- Cada elemento do dicionário é um par composto por chave (*key*) e valor (*value*).
- Sintaxe:

```
d = {  
    <key1>:<value1>,  
    <key2>:<value1>,  
    <key3>:<value2>  
}
```

Dicionários - Exemplos

Criando um dicionário com chaves inteiras

```
dicionario = {  
    1 : 'exemplo',  
    2 : 'de',  
    3 : 'dicionario'  
}
```

```
print(dicionario)
```

```
{1: 'exemplo', 2: 'de', 3: 'dicionario'}
```

Dicionários - Exemplos

Criando um dicionário com chaves de tipos mistos

```
teste = {  
    'nome' : 'fulano',  
    5 : 'cinco',  
    'lista' : [1, 2, 4]  
}
```

```
print(teste)
```

```
{'nome': 'fulano', 5: 'cinco', 'lista': [1, 2, 4]}
```


Dicionários - Acessando elementos

- Os valores são acessados por meio de suas chaves
- Utiliza-se o nome do dicionário e a chave dentro de colchetes `[]`

```
ingles = {  
    'um' : 'one',  
    'dois' : 'two',  
    'tres' : 'three',  
    'quatro' : 'four',  
    'cinco' : 'five'  
}
```

```
ingles['um']
```

'one'

```
ingles['quatro']
```

'four'

```
ingles_num = {  
    1 : 'one',  
    2 : 'two',  
    3 : 'three',  
    4 : 'four',  
    5 : 'five'  
}
```

```
ingles_num[3]
```

'three'

```
ingles_num[2]
```

'two'

Dicionários - Adicionando novos elementos

- Para adicionar um novo elemento a um dicionário existente, basta atribuir o novo valor e especificar a chave dentro de colchetes

Dicionário original

```
ingles_num = {  
    1 : 'one',  
    2 : 'two',  
    3 : 'three',  
    4 : 'four',  
    5 : 'five'  
}
```

```
ingles_num[7] = 'seven'
```

Acrescentando um novo elemento
chave = 7 e valor = 'seven'

```
print(ingles_num)
```

```
{1: 'one', 2: 'two', 3: 'three', 4: 'four', 5: 'five', 7: 'seven'}
```

Dicionários - Removendo elementos

- Para deletar um elemento de um dicionário utilizamos a palavra-chave **del**

Dicionário original

```
ingles_num = {  
    1 : 'one',  
    2 : 'two',  
    3 : 'three',  
    4 : 'four',  
    5 : 'five'  
}
```

```
del ingles_num[3]
```

```
print(ingles_num)
```

```
{1: 'one', 2: 'two', 4: 'four', 5: 'five'}
```

Removendo o elemento pela chave = 3

Dicionários - Removendo elementos

- Para deletar um elemento de um dicionário utilizamos a palavra-chave **del**

Dicionário original

```
ingles = {  
    'um' : 'one',  
    'dois' : 'two',  
    'tres' : 'three',  
    'quatro' : 'four',  
    'cinco' : 'five'  
}
```

```
del ingles['quatro']
```

Removendo o elemento pela chave =
'quatro'

```
print(ingles)
```

```
{'um': 'one', 'dois': 'two', 'tres': 'three', 'cinco': 'five'}
```

Dicionários - Alguns métodos

- **items()**: retorna todos os elementos do dicionário - pares chave:valor

```
dicionario = {  
    'um' : 'exemplo',  
    'dois' : 'de',  
    'tres' : 'dicionario'  
}
```

```
d = dicionario.items()
```

```
d = list(d)  
print(d)
```

```
[('um', 'exemplo'), ('dois', 'de'), ('tres', 'dicionario')]
```

Dicionários - Alguns métodos

- **keys()**: retorna todas as chaves do dicionário

```
diccionario = {  
    'um' : 'exemplo',  
    'dois' : 'de',  
    'tres' : 'dicionario'  
}
```

```
d = diccionario.keys()
```

```
d = list(d)  
print(d)
```

```
['um', 'dois', 'tres']
```

```
print(d[0])
```

```
um
```

Dicionários - Alguns métodos

- **values()**: retorna todas valores do dicionário

```
dicionario = {  
    'um' : 'exemplo',  
    'dois' : 'de',  
    'tres' : 'dicionario'  
}
```

```
d = dicionario.values()
```

```
d = list(d)  
print(d)
```

```
['exemplo', 'de', 'dicionario']
```

```
print(d[0])
```

```
exemplo
```

Dicionários - Iteração

- Podemos utilizar estruturas de repetição para iterar por um dicionário

```
dicionario = {  
    'um' : 'exemplo',  
    'dois' : 'de',  
    'tres' : 'dicionario'  
}
```

Iterando pelas chaves

```
for chave in dicionario:  
    print(chave)
```

um
dois
tres

Iterando pelos valores

```
for valor in dicionario.values():  
    print(valor)
```

exemplo
de
dicionario

Dicionários - Iteração

- Podemos utilizar estruturas de repetição para iterar por um dicionário

```
dicionario = {  
    'um' : 'exemplo',  
    'dois' : 'de',  
    'tres' : 'dicionario'  
}
```

```
for chave, valor in dicionario.items():  
    print(chave, valor)
```

```
um exemplo  
dois de  
tres dicionario
```