

Introdução ao Python

Estrutura de dados - Dicionários

Henrique Y. Shishido

Departamento de Computação
Universidade Tecnológica Federal do Paraná

Introdução

- Dicionários são usados para armazenar dados no formato `chave:valor`
- Os elementos de um dicionário são ordenados, mutáveis e não permitem elementos duplicados

Inicialização de um dicionário

- Um dicionário é inicializado envolvendo o conjunto de valores chave:valor em chaves { e }

```
>>> meuDict = {  
    "nome": "John Bull",  
    "endereco": "R. Goiás, 82",  
    "dataNasc": "20/12/1973",  
    "fone": "+559190900000",  
    "email": "johnb@email.com"  
}  
>>> print(meuDict)  
{'nome': 'John Bull', 'endereco': 'R. Goiás, 82', 'dataNasc': '20/12/1973', 'fone': '+559190900000', 'email': 'johnb@email.com'}
```

Valores duplicados

- Caso um dicionário seja inicializado com dois itens com chaves iguais, a última declaração será considerada

```
>>> meuDict = {  
    "nome": "John Bull",  
    "endereco": "R. Goiás, 82",  
    "idade": "25"  
    "idade": "70"  
}  
>>> print(meuDict)  
{'nome': 'John Bull', 'endereco': 'R. Goiás, 82', 'idade': '70'}
```

Tamanho de um dicionário

- Para determinar o número de itens existentes em um dicionário, pode-se usar a função `len()`:

```
>>> meuDict = {  
    "nome": "John Bull",  
    "endereco": "R. Goiás, 82",  
    "idade": "25"  
}  
>>> print(len(meuDict))  
3
```

Itens de um dicionário

- Os valores em um dicionário podem ser de diferentes tipos de dados

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> type(meuDict)  
<class 'dict'>
```

Acesso a itens de um dicionário

- Os itens de um dicionário podem ser acessados por meio da chave dentro de colchetes `[]` ou pelo método `get()`

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
#Usando colchetes meuDict["chave"]  
>>> print(meuDict["nome"])  
"John Kolz"  
  
#Usando o método get("chave")  
>>> print(meuDict.get("idade"))  
52
```

Chaves de um dicionário

- Para recuperar todas as chaves de um dicionário, utiliza-se o método `keys()`:

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> print(meuDict.keys())  
dict_keys(['nome', 'idade', 'fumante', 'filiacao'])
```


Valores de um dicionário

- O método `values()` retorna a lista de todos os valores de um dicionário

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> print(meuDict.values())  
dict_keys(['John Kolz', 52, True, ["Janete Lokenicz", "Peter Kolz"]])
```

Recuperando itens de um dicionário

- O método `items()` retorna cada item de um dicionário como tuplas em uma lista

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> print(meuDict.items())  
dict_items([('nome', 'John Kolz'), ('idade', 52), ('fumante', True), (  
    'filiacao', ['Janete Lokenicz', 'Peter Kolz'])])
```

Checando a existência de uma chave

- Para determinar se uma chave específica está presente em um dicionário, utiliza-se a palavra-chave `in`:

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> if "nome" in meuDict:  
>>>     print("A chave nome existe no meu dicionário")  
  
A chave nome existe no meu dicionário
```

Alterando valores em um dicionário

- A alteração de um valor de um item específico é realizado por meio da referência de sua chave:

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> meuDict["idade"] = 80  
>>> print(meuDict)  
{'nome': 'John Kolz', 'idade': 80, 'fumante': True, 'filiacao': [  
    'Janete Lokenicz', 'Peter Kolz']}
```

Atualizando dicionário

- Para atualizar o dicionário com um conjunto de itens, utiliza-se o método `update()`. O argumento passado deve ser um dicionário:

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]  
}  
  
>>> meuDict.update({"idade": 80, "fumante": False})  
>>> print(meuDict)  
{'nome': 'John Kolz', 'idade': 80, 'fumante': False, 'filiacao': [  
    'Janete Lokenicz', 'Peter Kolz']}
```

Adicionando novos itens no dicionário

- A adição de um novo item ao dicionário pode ser realizada de duas formas:

- Uso de uma nova chave e atribuindo um valor a ela. Exemplo:

```
meuDict["corPreferida"] = "amarelo"
```

- Uso do método update() passando novos pares chave:valor. Exemplo:

```
meuDict.update("corPreferida": "amarelo")
```

```
>>> meuDict = {
    "nome": "John Kolz",
    "idade": 52,
    "fumante": True,
    "filiacao": ["Janete Lokenicz", "Peter Kolz"]
}

>>> meuDict["nacionalidade"] = "Alemão"
# ou
>>> meuDict.update({"nacionalidade": "Alemão"})
>>> print(meuDict)
{'nome': 'John Kolz', 'idade': 52, 'fumante': True, 'filiacao': ['Janete Lokenicz', 'Peter Kolz'], 'nacionalidade': 'Alemão'}
```

Removendo itens do dicionário

- Há diversos meios para remover itens de um dicionário.

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
}  
  
#Remove o item de acordo com a chave  
>>> meuDict.pop("nome")  
{'idade': 52, 'fumante': True}  
  
#Remove o último item do dicionário  
>>> meuDict.popitem()  
{'nome': 'John Kolz', 'idade': 52}  
  
#Remove todos os itens do dicionário  
>>> meuDict.clear()  
{}
```

Iterando nos itens de dicionário

- O comando for pode ser usado para iterar sobre os elementos de um dicionário:

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
}  
  
#Imprime todas as chaves do dicionário  
>>> for item in meuDict:  
>>>     print(item)  
"nome",  
"idade",  
"fumante"  
  
#Imprime todos os valores do dicionário  
>>> for item in meuDict:  
>>>     print(meuDict[item])  
"John Kolz",  
52,  
True
```


Cópia de um dicionário

- Não é possível copiar um dicionário digitando `meuDict2 = meuDict1`, pois esse comando irá criar uma referência do `meuDict1` em `meuDict2`.
- Logo, é necessário usar o método :

```
>>> meuDict = {  
    "nome": "John Kolz",  
    "idade": 52,  
    "fumante": True,  
}  
  
>>> novoDict = meuDict.copy()  
>>> print(novoDict)  
{'nome': 'John Kolz', 'idade': 52, 'fumante': True}
```

Aninhamento de dicionários

- Um dicionário pode conter outros dicionários dentro de si.

```
>>> alunos = {  
    "aluno1": {  
        "nome": "Jefferson",  
        "idade": 15  
    },  
    "aluno2": {  
        "nome": "João",  
        "idade": 16  
    }  
}  
>>> print(alunos)  
{'aluno1': {'nome': 'Jefferson', 'idade': 15}, 'aluno2': {'nome': 'João',  
    'idade': 16}}
```