

Dicionários

Os dicionários são uma estrutura de dados dinâmica que armazena os dados na forma de pares chave-valor. Os dicionários, tal como as listas, são mutáveis. Os dicionários são indexados por chaves, que podem ser de qualquer tipo imutável, strings ou números podem ser chaves. As chaves devem ser únicas, enquanto os valores associados podem ser de qualquer tipo de dados e podem ser repetidos.

Definir um dicionário

Os dicionários são definidos utilizando chavetas {}. Cada valor do dicionário é definido no formato chave : valor

```
In [ ]: meu_dicionario = {'nome': 'Joaquim', 'idade': 30, 'cidade': 'Viseu', 'morada': ('viseu', 'viseu')}
print(meu_dicionario)
print(meu_dicionario['idade'])
print(meu_dicionario['morada'][0])
morada=meu_dicionario['morada']
print(morada[0])
```

Listar os elementos de um dicionário

Para listar os elementos de um dicionários podemos utilizar os métodos keys(), values() e items(). O método keys() devolve uma lista de todas as chaves do dicionário O método values() devolve uma lista de todos os valores armazenados no dicionário O método items() devolve uma lista de tuplos, em que cada um é um par chave-valor

```
In [ ]: meu_dicionario = {'nome': 'Joaquim', 'nif': '123123123', 'morada': 'viseu'}
chaves = meu_dicionario.keys()
print(chaves)
valores = meu_dicionario.values()
print(valores)
elementos = meu_dicionario.items()
print(elementos)
```

Também podemos aceder aos valores utilizando a chave como indice.

```
In [ ]: meu_dicionario = {'nome': 'Joaquim', 'nif': '123123123', 'morada': 'viseu'}
print(meu_dicionario['nome'])
```

```
In [ ]: meu_dicionario = {'nome': 'Joaquim', 'nif': '123123123', 'morada': 'viseu'}
for chave in meu_dicionario.keys():
    print(meu_dicionario[chave])
```

```
In [ ]: meu_dicionario = {'nome': 'Joaquim', 'nif': '123123123', 'morada': 'viseu'}
for elemento in meu_dicionario.items():
    print(f"{elemento[0]}:{elemento[1]}")
```

```
In [ ]: meu_dicionario = {'nome' : 'Joaquim', 'nif' : '123123123', 'morada':'viseu'}
        for chave, valor in meu_dicionario.items():
            print(f"{chave}:{valor}")
```

Método get - devolve o valor de uma chave ou um valor definido por omissão

```
In [ ]: meu_dicionario = {'nome':'Joaquim', 'nif':'123123123', 'morada':'viseu'}

        print(meu_dicionario.get("email","Não existe email"))
```

Pesquisar com base no valor de uma chave

```
In [ ]: meu_dicionario = {'nome' : 'Joaquim', 'nif' : '123123123', 'morada':'viseu'}

        #verificar se o dicionário tem uma chave nome
        if 'nome' in meu_dicionario.keys():
            print(f"Dicionário tem nome com o valor {meu_dicionario['nome']}")
        else:
            print("Dicionário não tem nome")
```

Operador in com dicionários

```
In [ ]: dicionario = {'nome':'Joaquim', 'idade':26}

        if 'nome' in dicionario:
            print("O dicionário tem a chave nome.")

        if 'profissao' not in dicionario:
            print('O dicionário não tem a chave profissao')
```

Concatenar dois dicionários

```
In [ ]: dicionario1 = {'a':1,'b':2,'c':3}
        dicionario2 = {'c':4,'d':5}

        dicionario1.update(dicionario2) #alterar os elementos repetidos e adiciona os no
        print(dicionario1)
```

Remover chaves

O método pop() remove a chave e devolve o valor correspondente. A instrução del remove a chave, se não existir ocorre um erro.

```
In [ ]: dicionario = {'a':1,'b':2,'c':3}
        print(dicionario)
        valor=dicionario.pop('b',None)
        print(valor)
        print(dicionario)
        valor=dicionario.pop('d',None)
        print(valor)
```

```
In [ ]: dicionario = {'a':1,'b':2,'c':3}
del dicionario['b']
print(dicionario)
```

Arrays de dicionários

OS arrays do numpy pode ser utilizados para guardar uma coleção de dicionários.

```
In [ ]: import numpy as np

def AdicionarAluno():
    nome=input("Nome:")
    idade=int(input("Idade:"))
    morada=input("Morada:")
    aluno={'nome':nome,'idade':idade,'morada':morada}
    return aluno

MAX_ITEMS=100
alunos=np.empty(MAX_ITEMS,dtype=object)

novo=AdicionarAluno()
alunos[0]=novo

print(alunos[0])
#mudar o nome
alunos[0]['nome']='maria'
print(alunos[0])
```

Listas de dicionários

A mesma coisa pode ser feita com listas (ver a próxima secção).

Os dicionários que são inseridos na lista não têm de ter a mesma estrutura!