

Dicionários e Uso de Matriz em Python





Dicionários

- Coleção não ordenada de valores que são referenciados pelas chaves correspondentes.

```
al_D={'joao':5.0, 'maria':10.0,'pedro':3.0}
```

- Trata-se de um outro tipo de dado assim como listas e tuplas.



Dicionários

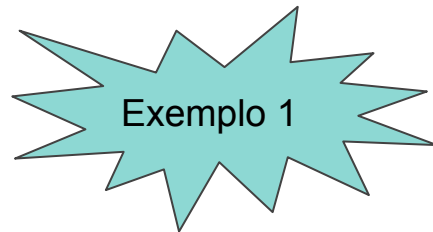
- Notação

`<nove-dicionario> = {chave1 : valor1, chave2 : valor2,...,chave_n : valor_n}`

- Apenas valores imutáveis podem ser usados como chaves: strings, números e tuplas.



Dicionários



- Vantagens
 - Acessa itens de interesse diretamente
 - Permite usar uma única estrutura de dados ao invés de listas separadas.

Índice	Valor
0	joao
1	maria
2	pedro

Índice	Valor
0	5
1	10
2	3

Chave	Valor
joao	5
maria	10
pedro	3



Dicionários

- Criando dicionário
al_D = {} (Vazio)
al_D={'joao':5.0,'maria':10,'pedro':3.0}
- Acessando dados
print(al_D['joao']) ⇒ 5
print(al_D['antonio']) ⇒ KeyERROR
- Alterando e acrescentando dados
al_D['joao'] =8
al_D['ana']=9.5

Chave	Valor
joao	5
maria	10
pedro	3

Chave	Valor
joao	8
maria	10
pedro	3
ana	9.5



Dicionários

- Removendo
`del(al_D['joao'])`
- Avaliando se está no dicionário
`print('maria' in al_D) ⇒ True`
`print('joao' in al_D) ⇒ Falso`
- Não há ordem para chaves e valores num dicionário
`qualquer_D={1:'janeiro','joao':80.5,(1,2,3):'sequencia'}`

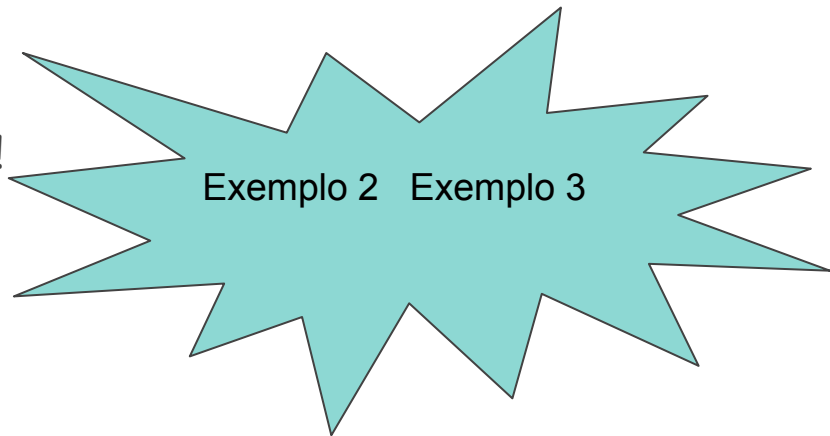
Chave	Valor
maria	10
pedro	3
ana	9.5



Dicionários

- Retornando as chaves em uma lista
`list(al_D.keys())` ⇒ ['maria','pedro','ana']
A ordem original não é garantida
- Retornando os valores em uma lista
`list(al_D.values())` ⇒ [10,3,9.5]
- Vários métodos para listas - PESQUISAR!!!
`copy(),clear(), get(),has_key(), etc..`

Chave	Valor
maria	10
pedro	3
ana	9.5

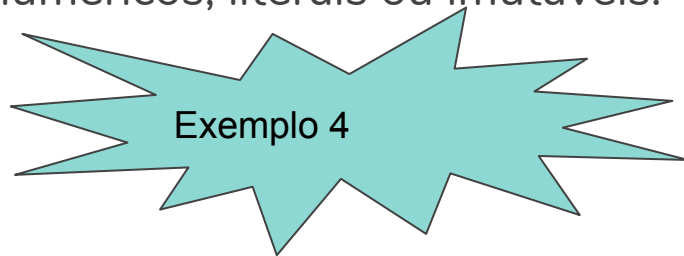
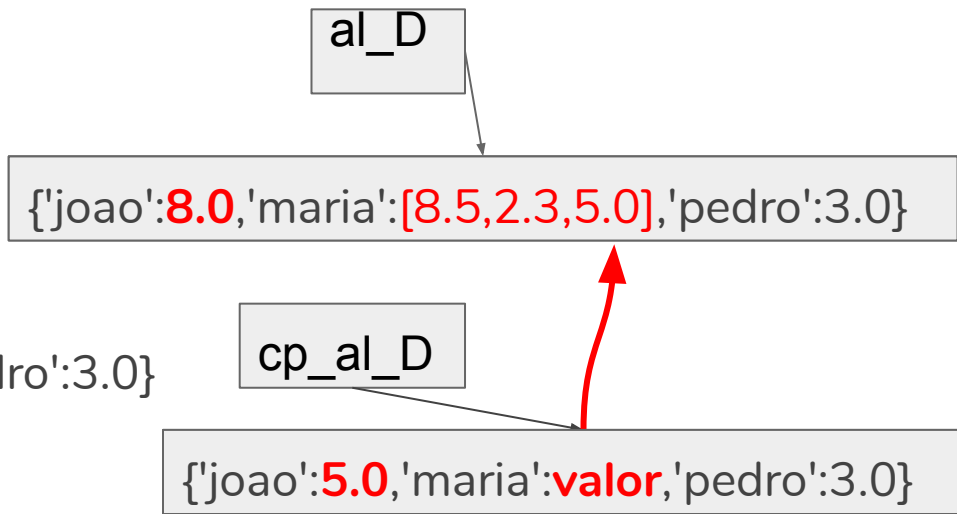




Dicionários

- `copy()`
`al_D={'joao':5.0,'maria':[8.5,2.3],'pedro':3.0}`
`cp_al_D= al_D.copy()`
`al_D['maria'].append(5.0)`
`al_D['joao']=8.0`

- `copy()` não cria uma cópia completamente independente do dicionário original
- A cópia é independente apenas para valores numéricos, literais ou imutáveis.
- A cópia não é independente para listas!!





Dicionários

- deepCopy()

from copy import deepCopy

```
al_D={'joao':5.0,'maria':[8.5,2.3],'pedro':3.0}
```

```
cp_al_D= deepcopy(al_D)
```

```
al_D['maria'].append(5.0)
```

```
cp_al_D['joao']=8.0
```

- Cria uma cópia completamente independente do dicionário original
- Necessidade de importar deepCopy
- deepcopy() é função
- copy() é método..

al_D

```
{'joao':5.0,'maria':[8.5,2.3,5.0],'pedro':3.0}
```

cp_al_D

```
{'joao':8.0,'maria':[8.5,2.3],'pedro':3.0}
```

Exemplo 5 Exemplo 6



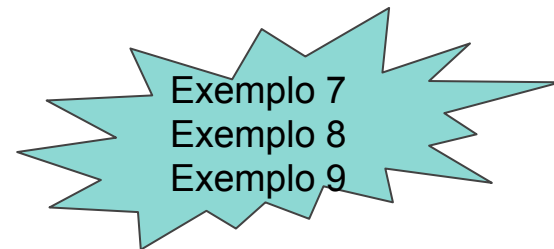
Uso de Matriz em Python

- Uma matriz pode ser entendida como uma estrutura de dados bi-dimensional, onde os dados endereçados por linhas e colunas.
- Todavia, pode-se considerar estruturas n-dimensionais.

	0	1	2			1	2		
0	10	20	30	0	10	20	30	20	30
1	40	50	60		40	50	60	50	60
2	70	80	90		70	80	90	80	90



Uso de Matriz em Python



- Uma matriz pode ser definida em Python como uma lista de listas.

```
M = [[10,20,30], [40,50,60],[70,80,90]]
```

```
print(M[1]) ⇒ [40,50,60]
```

```
print(M[1][2]) ⇒ 60
```

```
print(M[0][-1]) ⇒ 30
```

```
print(M) ⇒ [[10,20,30], [40,50,60],[70,80,90]]
```

	0	1	2
0	10	20	30
1	40	50	60
2	70	80	90



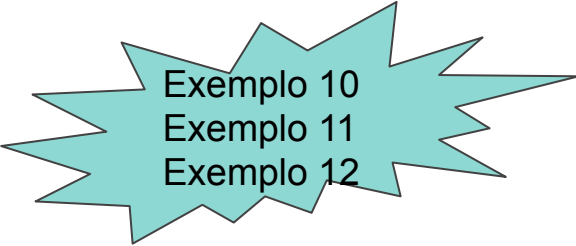
NumPy

- Pacote para cálculos científicos bastante útil para manipular estruturas n-dimensionais.
- Deve ser instalado: <https://scipy.org/install.html>
- Deve-se importar para utilização no código

```
import numpy as np
```



NumPy



Exemplo 10
Exemplo 11
Exemplo 12

- Operações com matrizes
 - `<Matriz>.add()` : Soma entradas na mesma posição de duas matrizes.
 - `<Matriz>..subtract()` : Subtrai entradas na mesma posição de duas matrizes.
 - `<Matriz>.divide()` : Divide entradas na mesma posição de duas matrizes.
 - `<Matriz>.multiply()`: Multiplica entradas na mesma posição de duas matrizes..
 - `<Matriz>.dot()` : Executa a multiplicação matricial.
 - `<Matriz>.sqrt()`:Calcula raiz quadrada de cada elemento da matriz.
 - `<Matriz>.sum(x,axis)`:Soma todas as entradas da matriz. Se `axis=0`, retorna a soma das colunas. Se `axis=1`, retorna a soma das linhas. `axis` é opcional.
 - `<Matriz>.T`:Retorna a matriz transposta.