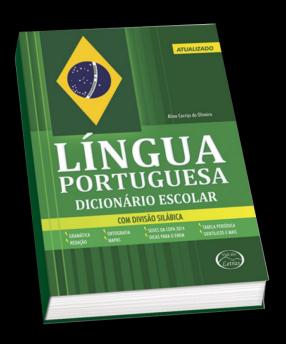
Prof. Alberto Costa Neto Programação em Python

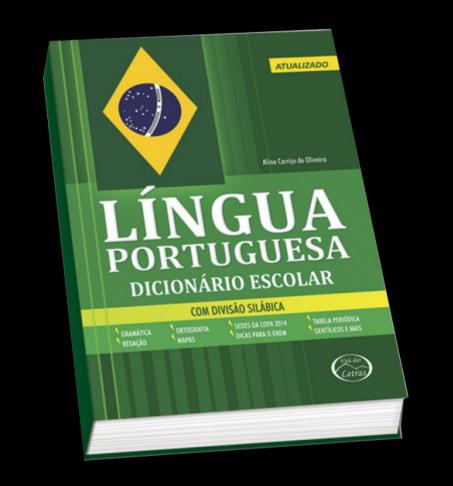
Dois tipos de Coleções

- List (Lista)
 - > Uma coleção linear de valores que ficam em ordem



- Dictionary (Dicionário)
 - > Um "saco" de valores, onde cada valor tem um rótulo (chave) identificando-o





metation / dik teifn/ noun [C,U] spoken words that shelse must write or type: We had a dicta tion in English today. > dyktando dictator /dik'tertə(r); US 'diktertər/ noun [C] a on to charge ruler who has total power in a country, especially one who used force to gain power and we Vou put un the dial who rules the country unfairly dyktator dictatorship noun [C,U] government by a dic tator; a country that is ruled by a dictator: a military dictatorship > dyktatura *dictionary /'dikfənri; US -neri/ noun [C] (pl. dictionaries) 1 a book that lists the words of a language in alphabetical order and that tells you what they mean, in the same or another language: to look up a word in a dictionary o a bilingual/monolingual dictionary o a French-Inglish dictionary > słownik 2 a book that ts the words connected with a particular suband tells you what they mean: a d

- São as coleções de dados mais poderosas de Python
- Permitem operações rápidas no estilo de banco de dados em Python
- Têm diferentes denominações em outras linguagens de programação
 - > Arrays Associativos Perl / PHP / Lua
 - > Properties, Map e Hashtable Java
 - > Property Bag C# / .Net

- Listas indexam suas entradas baseado-se na posição na lista
- Dicionários são como sacolas – sem noção de ordem
- Então a indexação é através das "lookup tags" (rótulos de busca/chaves)

```
>>> bolsa = dict()
>>> bolsa['dinheiro'] = 12
>>> bolsa['doce'] = 3
>>> bolsa['lenço'] = 75
>>> print(bolsa)
{'dinheiro': 12, 'doce': 3, 'lenço': 75}
>>> print(bolsa['doce'])
3
>>> bolsa['doce'] = bolsa['doce'] + 2
>>> print(bolsa)
{'dinheiro': 12, 'lenço': 75, 'doce': 5}
```

Literais Dicionário (Constantes)

• Usam chaves e têm uma lista de pares (chave : valor)

Para criar um dicionário vazio usamos chaves { }

```
>>> dic_vazio = { }
>>> print(dic_vazio)
{}
>>> dic = {'jose' : 1 , 'fred' : 42, 'maria': 100}
>>> print(dic)
{'jose': 1, 'fred': 42, 'maria': 100}
>>>
```

Comparando Listas e Dicionários

 Dicionários são como Listas exceto que usam chaves (rótulos de busca) no lugar de números para buscar valores

```
>>> lst = list()
>>> lst.append(21)
>>> lst.append(183)
>>> print(lst)
[21, 183]
>>> lst[0] = 23
>>> print(lst)
[23, 183]
```

```
>>> dic = dict()
>>> dic['idade'] = 21
>>> dic['curso'] = 182
>>> print(dic)
{'curso': 182, 'idade': 21}
>>> dic['idade'] = 23
>>> print(dic)
{'curso': 182, 'idade': 23}
```

```
>>> lst = list()
                                                Lista
>>> lst.append(21)
                                                    Valor
                                           Chave
>>> lst.append(183)
>>> print(lst)
                                              [0]
                                                     21
[21, 183]
                                                               lst
>>> lst[0] = 23
                                              [1]
                                                    183
>>> print(lst)
[23, 183]
>>> dic = dict()
                                              Dicionário
>>> dic['idade'] = 21
>>> dic['curso'] = 182
                                           Chave
                                                     Valor
>>> print(dic)
                                                     183
                                            ['curso']
{'curso': 182, 'idade': 21}
                                                              dic
>>> dic['idade'] = 23
                                            ['idade']
                                                      21
>>> print(dic)
{'curso': 182, 'idade': 23}
```

Chaves do tipo inteiro nos Dicionários

 Dicionários suportam até chaves do tipo inteiro. Neste caso, ficam com sintaxe de uso muito semelhante à das listas

```
>>> dic = dict()
>>> lst = list()
>>> lst.append(21)
                            >>> dic[1] = 21
>>> lst.append(183)
                            >>> dic[20] = 182
>>> print(lst)
                            >>> print(dic)
[21, 183]
                            {1: 21, 20: 182}
>>> lst[0] = 23
                            >>> dic[20] = 23
>>> print(lst)
                            >>> print(dic)
[23, 183]
                            {1: 21, 20: 23}
```

Copiando/Limpando Dicionários e Apagando itens

```
>>> dic = {'jose' : 1 , 'fred' : 42, 'maria': 100}
>>> dic1 = dic
>>> dic2 = dic.copy()
>>> valor = dic.pop('jose')
>>> print('Removido', valor, dic1)
Removido 1 {'fred': 42, 'maria': 100}
>>> dic.clear()
>>> print(dic1)
{ }
>>> print(dic2)
{'jose': 1, 'fred': 42, 'maria': 100}
>>>
```

Atualizando dicionário a partir de outro dicionário

```
>>> d1 = {'jose' : 1 , 'fred' : 42}
>>> d2 = {'jose' : 10, 'maria': 100}
>>> d3 = {}
>>> d3.update(d1)
>>> print(d3)
{'jose': 1, 'fred': 42}
>>> d3.update(d2)
>>> print(d3)
{'jose': 10, 'fred': 42, 'maria': 100}
>>>
```

- Ao aplicar update sobre d3, passando d1 como parâmetro, as chaves e valores de d1 serão incluídos em d3.
- Quando d3 já contém alguma chave contida em d1, o valor atual em d3 será substituído pelo valor de d1