

AULA 4 (Dicionários)

Professora: Lidiane Visintin

lidiane.visintin@ifc.edu.br

Professor: Rafael de Moura Speroni

rafael.speroni@ifc.edu.br

Objetivo:

Compreender o conceito de Dicionários.

Listas e Dicionários

 Em Python existem três tipos principais de variáveis compostas: Listas, Tuplas e Dicionários.

Listas e Dicionários

Vamos aos conceitos:

- Dicionários são uma estrutura de dados similar às listas, mas com propriedades de acesso diferentes. Cada dicionário é composto por um conjunto de chaves e valores.
- O dicionário consiste em relacionar uma chave a um valor.

Como imaginar um dicionário:

Preços de mercadorias

Produto	Preço
Alface	0,99
Batata	1,45
Tomate	4,99
Feijão	3,50

 A tabela exibida pode ser vista como um dicionário, em que chave seria o produto e o valor seu preço;

Dicionários

- Em Python, criamos dicionários utilizando chaves({ });
- Como ficaria no Python:

• Um dicionário é acessado por suas chaves, exemplo:

```
print(tabela["Alface"])
```

irá exibir 0.99;

ele diferencia letras maiúsculas de minúsculas, ou seja, é case sensitive

Diferente das listas em que o índice é um número Dicionários utilizam suas chaves como índice.

Como declarar um dicionário?

$$D = \{ \}$$

Este comando cria um dicionário denominado de D e as chaves ({ })
após o símbolo de igualdade indicam que é um dicionário vazio.

 Este comando cria um dicionário denominada de tabela e este dicionário contém 4 elementos.

Como acessar um elemento de um dicionário?

- Se acessarmos tabela ["Feijão"] o resultado que será exibido será 3.50, pois estamos acessando a última chave do dicionário, ou seja, o valor que está relacionado a chave "Feijão".
- tabela["Alface"] será igual a 0.99 e tabela["Batata"] será igual a 1.45.

Como alterar um elemento de um dicionário?

• Se atribuírmos tabela ["Alface"] = 1.50 o conteúdo de tabela ["Alface"] será atualizado. E ficaremos com:

O valor anterior será perdido

Como alterar um elemento de um dicionário?

• Se atribuírmos tabela ["Cenoura"] = 2.10 o conteúdo de tabela será atualizado. E ficaremos com:

Encontrando um elemento de um dicionário?

 Se buscarmos por uma chave que não existe no dicionário, isso nos retornará um erro, por exemplo:

```
print(tabela["Manga"])
```

Esse comando irá gerar um erro:

```
print(tabela["Manga"])
KeyError: 'Manga'
```

Encontrando um elemento de um dicionário?

 Para sabermos se a chave existe no dicionário podemos utilizar o comando in:

```
print("Manga" in tabela) resposta False;
print("Batata" in tabela) resposta True;
```

Obtendo as chaves e os valores de um dicionário?

 Podemos utilizar os métodos keys() e values(), para sabermos as chaves existentes no dicionário, ou os valores:

```
print(tabela.keys())
no terminal:
dict_keys(['Alface', 'Batata', 'Tomate', 'Feijão']);
print(tabela.values())
no terminal:
dict_values([1.5, 1.45, 4.99, 3.5])
```

Removendo elementos de um dicionário

 Para removermos um elemento do dicionário podemos utilizar a instrução del;

```
del tabela["Alface"]
```

irá ficar:

Como ler e mostrar um dicionário?

```
# cálculo de ocorrências de nomes
alunos =['Joao', 'Pedro', 'Lucas', 'Pedro', 'Ana']
dic = \{ \}
                                                   lista com o nome de
for p in alunos:
                                                   estudantes
     if p in dic:
                                               dicionário vazio
          dic[p] += 1
                                                    irá percorrer um a um dos
                                                    elementos da lista
     else:
          dic[p] = 1
                                              verifica se cada nome já está no
                                              dicionário.
print(dic)
```

Resultado: {'Joao': 1, 'Pedro': 2, 'Lucas': 1, 'Ana': 1}

Resumo

	Listas	Tuplas	Dicionários	Conjuntos
Ordem dos	Fixa	Fixa	Mantida a partir	Indeterminada
elementos			do Python 3.7	
Tamanho	Variável	Fixa	Variável	Variável
Elementos	Sim	Sim	Pode repetir	Não
repetidos			valores, mas as	
			chaves devem ser	
			únicas	
Pesquisa	Sequencial,	Sequencial,	Direta por chave	Direta por
	índice	índice		valor
	numérico	numérico		
Alterações	Sim	Não	Sim	Sim
Uso primário	Sequências	Sequências	Dados indexados	Verificação de
		Constantes	por chave	unicidade,
				operações
				com conjuntos

Referências

Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. Algoritmos: lógica para desenvolvimento de programação de computadores.. 27. ed.. Érica. 2014

Referências Complementares

DOWNEY, Allen B. Pense em Python. 2ª Ed. Novatec. 2016

MENEZES, Nilo Ney de Coutinho. Introdução a programação com Python. 3ª Ed. Novatec. 2019

CORMEN, Thomas H et al. Algoritmos: teoria e prática. 2. ed. Elsevier, Campus,. 2002

Referências na Internet

https://docs.python.org/3/

https://www.w3schools.com/python/default.asp