

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Computational Thinking

PROF. EDUARDO GONDO

Dicionário - Introdução

- ▶ é uma estrutura de dados
- ▶ assim como tuplas e listas, os **dicionários** representam conjunto de dados
- ▶ informações são colocadas na forma de chave(key)/valor(value)
- ▶ a chave serve para termos acesso ao valor e ela pode ser um número ou uma string ou qualquer outro objeto imutável
- ▶ já o valor pode ser qualquer tipo de objeto
- ▶ exemplos: dicionários do mundo real, agenda telefônica, whatsapp, servidor de email

Criação

```
1 dic = {}  
2 dic['carro'] = 'car'  
3 dic['caneta'] = 'pen'  
4 dic['rato'] = 'mouse'  
5  
6 #ou  
7  
8 agenda = {'ana':'ana@gmail.  
            com', 'beto':'beto@yahoo  
            .com', 'cida':'cida@zip.  
            com'}
```

criando dicionario vazio
adiciona pares de chave e
valor no dicionario

criando dicionario ja com
alguns valores
preenchidos

Alterando e removendo valores

- ▶ para alterar um valor no dicionário é da mesma forma que alteramos uma lista:

```
1 agenda["beto"] = "roberto@gmail.com"
```

- ▶ para remover uma entrada no dicionário podemos usar o método pop ou o comando del:

```
1 valor = agenda.pop('beto')  
2 del agenda['ana']
```

Acessando valores

- ▶ para recuperar um valor armazenado podemos usar o método `get` ou fazer algo parecido com as listas:

```
1 palavra = dic['rato']  
2 #ou  
3 palavra = dic.get('rato')
```

Mais alguns recursos

- ▶ `in` verifica se a chave está no dicionário

```
1 if 'ana' in agenda:
2     print("Ana esta no dicionario")
```

- ▶ podemos percorrer o conjunto de chaves de um dicionário

```
1 for chave in agenda.keys():
2     print(chave, '=>', agenda[chave])
```

- ▶ podemos percorrer os valores armazenados

```
1 for valor in agenda.values():
2     print(valor)
```

Resumo das operações

- ▶ acessar ou alterar um elemento do *dicionário*: use o nome do dicionário mais a chave (parecido com as *listas*)
- ▶ o método `pop(<key>)` retorna o valor associado à chave e remove o par chave/valor do dicionário
- ▶ também é possível remover um elemento do dicionário usando `del <dicionario>[key]`
- ▶ operador `in` verifica se uma determinada chave está no dicionário
- ▶ é possível percorrer o conjunto de chaves ou de valores usando os métodos `keys()` e `values()` do dicionário
- ▶ contudo é tão rotineiro iterar sobre o conjunto de chaves que podemos omitir a chamada do método `keys()`, faça um teste

Registro de informações

- ▶ podemos usar dicionários para armazenar registros de informações
- ▶ quase como se fosse um objeto, suponha que desejamos representar um veículo
- ▶ os dados de veículo seriam: marca, modelo, ano, placa e dono
- ▶ veja no código abaixo um exemplo:

```
1 dados = {}  
2 dados['marca'] = "Toyota"  
3 dados['modelo'] = "Yaris"  
4 dados['ano'] = 2020  
5 dados["placa"] = "HGR-8T34"  
6 dados["dono"] = "Marcos"
```


Registro de informações

- ▶ note que, podemos armazenar qualquer tipo de informação com um dicionário, muito parecido com classes e objetos
- ▶ combinando os dicionários com listas, onde cada posição da lista representa um dicionário, podemos fazer um repositório de dados
- ▶ ou seja, podemos fazer um sistema de cadastro com as opções de incluir, alterar, consultar e apagar
- ▶ porém com os dicionários fica muito mais fácil acessar as informações
- ▶ veja no próximo eslaide um exemplo de como fazer isso:

I Representação gráfica de lista de dicionários

Outras opções

- ▶ as estruturas de armazenamento como listas e dicionários podem ser combinadas de várias maneiras
- ▶ listaremos algumas:
 - ▶ dicionário de dicionários
 - ▶ dicionário de listas
 - ▶ lista de listas (matrizes)
 - ▶ lista de dicionários com listas dentro do dicionário
- ▶ caberá a você entender tais estruturas e combiná-las de modo que facilite o armazenamento das suas informações e a sua manipulação

I Problema 1

PROBLEMA 1 Escreva um programa que lê uma string e retorna uma contagem da ocorrência de todas as letras dessa string. Ignore se as letras são maiúsculas ou minúsculas e a contagem deve ser exibida em ordem alfabética.

I Problema 2

PROBLEMA 2 Escreva uma função que recebe um dicionário contendo palavras em **inglês** (chave) e **português** (valor). Sua função deverá retornar um outro dicionário invertendo as palavras, ou seja, seu dicionário deverá armazenar as palavras em português como chave e as em inglês como valor.

Problema 3

PROBLEMA 3 Faça um sistema de cadastro usando listas e dentro de cada lista um dicionário representando a informação a ser armazenada. Por exemplo, suponha que a queremos guardar um produto. Criamos uma lista e dentro de cada posição podemos colocar um dicionário conforme o exemplo abaixo:

```
1 prod = {"codigo": 123, "descricao": "camiseta branca", "
          quantidade": 100, "preco": 85.00}
```

Problema 4

PROBLEMA 4 Simulando o armazenamento de chaves PIX, como todos sabem o PIX se tornou muito popular para transferir e receber dinheiro. Em algum lugar (talvez no Banco Central ou Receita Federal), temos o armazenamento das chaves vinculado às informações de contas das pessoas. Crie um dicionário onde a chave é a chave PIX da pessoa (cpf/cnpj, email e telefone) e o valor é um outro dicionário que armazena os dados de uma conta: nome, banco, número da conta. Faça uma aplicação com duas opções de menu: cadastra e consulta chave PIX. No cadastra deverá ser informado a chave pix e as informações da conta e na consulta o usuário informa a chave Pix e o sistema retorna as informações da conta.

OBSERVAÇÃO: no cadastro, se a chave existir uma mensagem informando o problema deve ser exibida; já na consulta, se a chave não existir, também devemos informar o problema.

Problema 5

PROBLEMA 5 Faça uma aplicação simulando o whatsapp. Seu sistema deverá ter duas opções de menu: cadastra e consulta. No cadastra, você informa o número de telefone da pessoa e a mensagem enviada/recebida; o sistema deverá armazenar essas duas informações. Na consulta, o usuário informa o número de telefone e deverá ser exibido todas as mensagens já enviadas/recebidas do usuário. A ordem que as mensagens são exibidas é da última para a primeira. Use um dicionário para armazenar as informações onde a chave é o número do telefone e o valor é uma lista de objetos String representando as mensagens. **BÔNUS:** tente adicionar a data/hora nas mensagens. A data e hora pode ser o instante que cadastramos a mensagem. Observe que o dicionário não mais armazenará uma lista de String mas uma lista de dicionários.

Referência Bibliográfica

- ▶ Puga e Rissetti - Lógica de Programação e Estrutura de Dados
- ▶ Ascêncio e Campos - Fundamentos da Programação de Computadores
- ▶ Forbelone e Eberspacher - Lógica de programação: a construção de algoritmos e estruturas de dados
- ▶ Documentação do Python - <https://docs.python.org/3.8/>
- ▶ Python Programming For Beginners: Learn The Basics Of Python Programming (Python Crash Course, Programming for Dummies) (English Edition). Kindle
- ▶ Python: 3 Manuscripts in 1 book: - Python Programming For Beginners - Python Programming For Intermediates - Python Programming for Advanced (English Edition). Kindle

I Copyleft

Copyleft © 2025 Prof. Eduardo Gondo Todos direitos liberados.
Reprodução ou divulgação total ou parcial deste documento é liberada.