



Universidade Federal de Alagoas (UFAL)
Campus Arapiraca



Ciência da
Computação
UFAL Arapiraca

Algoritmos e Programação de Computadores (APC)

12

Dicionários e conjuntos

Alexandre de Andrade Barbosa

`alexandre.barbosa@arapiraca.ufal.br`

Objetivos

Objetivos

- Descrever os conteúdos:
 - Dicionários (dict)
 - Conjuntos (set)

Dicionários e conjuntos

Introdução

- Dicionários (dict) e conjuntos (set) também podem ser usados para armazenar mais de um valor
 - Dicionários armazenam pares Chave - valor
 - Conjuntos armazenam elementos sem repetição

Dicionários e conjuntos

Dicionários

Dicionários

- armazenam pares do tipo chave-valor
- os elementos armazenados podem ser acessados através de sua chave
- são mutáveis:
 - os elementos armazenados podem ser alterados
 - é possível adicionar e remover elementos
 - uma chave deve ser um tipo **imutável** (inteiro, string, tupla, ...)

Dicionários e conjuntos

Dicionários

```
1 # criando um dicionario vazio
2 dic = {}
3
4 # criando um dicionario com elementos
5 dic = {1: "um", 2: "dois"}
6 print(dic) # {1: "um", 2: "dois"}
7
8 # adicionando o valor associado a chave 3 (int)
9 dic[3] = "três"
10 print(dic) # {1: "um", 2: "dois", 3: "três"}
```

- Chaves: 1, 2 e 3
- Valores: "um", "dois" e "três"

Dicionários e conjuntos

Dicionários

A chave deve ser utilizada como referência

```
1 # dicionario com elementos
2 dic = {1: "um", 2: "dois"}
3 print(dic) # {1: "um", 2: "dois", 3:" três"}
4
5 # removendo elemento que possui chave "nome"
6 print(dic[2]) # "dois"
7 del dic[2]
8
9 # imprimindo dic após remoção
10 print(dic) # {1: "um", 3:" três"}
```

Dicionários e conjuntos

Dicionários

Dicionários

- Nos exemplos anteriores as chaves do dicionário eram int, qualquer tipo imutável pode ser utilizado com chave
- As chaves não precisam ser de um mesmo tipo, assim como os valores armazenados
- Um valor armazenado pode ser uma outra estrutura de dados
 - ex. um valor em um dicionário pode ser uma lista, ou vice-versa

Dicionários e conjuntos

Dicionários

```
1 # dicionario com chaves str
2 dic = {"um" : "one", "dois": "two", "três": "three"}
3 print(dic) # {"um" : "one", "dois": "two", "três": "three"}
4
5 # dicionario com uma lista como um de seus valores
6 disciplina = {"titulo": "Banco de dados",
7   "alunos" : ["joao", "jose", "maria"]}
8 print(disciplina) # todo o dicionario
9 print(disciplina["alunos"]) # apenas a lista
10 print(disciplina["alunos"][0]) # apenas um elemento da lista
```


Dicionários e conjuntos

Dicionários

Dicionários

- Chaves repetidas não são permitidas em um dicionário
- Embora seja possível definir o dicionário com chaves repetidas, sem que uma *Exception* ocorra
- Contudo, nenhuma valor repetido será mantido/armazenado no dicionário

Dicionários e conjuntos

Dicionários

```
1 # dicionario com elementos
2 dic = {1: "um", 1:"one", 2: "dois", 2: "two"}
3 print(dic) # {1: "one", 2: "two"}
```

- Observe que existem chaves repetidas 1 e 2, com os valores:
 - chave = 1 com valores 'um' e 'one'
 - chave = 2 com valores 'dois' e 'two'
- Nenhuma *Exception* ocorre
- Contudo, executando o código observamos apenas os últimos elementos fornecidos para cada chave (1: 'one' e 2: 'two')

Dicionários e conjuntos

Dicionários

Dicionários - métodos

- item **in** dict - verifica se 'item' é uma das chaves armazenadas
- pop(item) - retorna e remove o valor associado ao 'item' (chave)
- del(item) - remove o par chave-valor associado ao 'item' (chave)
- clear() - remove todos os pares armazenados
- items() - retorna uma lista com todos os pares de valores como tuplas
- values() - retorna uma lista com todos os valores armazenados
- keys() - retorna uma lista com todas as chaves armazenadas
- len(dict) - retorna a quantidade de pares armazenados no 'dict'

Dicionários e conjuntos

Dicionários

```
1 dic = {1: "um", 2: "dois", 3:"três"}
2
3 # retorna as chaves armazenadas no dicionário
4 print(dic.keys())
5
6 # retorna os valores armazenadas no dicionário
7 print(dic.values())
8
9 # retorna os itens (k,v) armazenadas no dicionário
10 print(dic.items())
```

Dicionários e conjuntos

Dicionários

Iterando sobre chaves, valores e itens

```
1 dic = {"um": "one", "dois": "two", "tres": "Three"}
2
3 for k in dic.keys() :
4     print(k)
5
6 for v in dic.values() :
7     print(v)
8
9 for k,v in dic.items() :
10    print("Chave: %s" % k)
11    print("Valor: %s" % v)
```

Dicionários e conjuntos

Dicionários

Dicionários

- Um dicionário pode ser utilizado para armazenar dados de uma entidade
- Por exemplo, não existe um tipo de dados para armazenar dados de uma entidade 'Pessoa'
- Podemos definir os campos de dados que desejamos armazenar em relação a uma pessoa e definimos estes como chaves
- Os respectivos valores para cada campo de dados serão os valores associados as suas chaves
- **Obs.:** Tópicos mais avançados tratam a definição de novos tipos com Classes

Dicionários e conjuntos

Listas, Tuplas e Dicionários

Construtores

```
1 # dicionario como um novo tipo
2 pessoa = {"nome": "Alexandre",
3           "profissao": "Professor",
4           "idade": 40}
5
6 print(pessoa["nome"])
7 print(pessoa["profissao"])
8 print(pessoa["idade"])
```

Dicionários e conjuntos

Listas, Tuplas e Dicionários

Construtores

```
1 lista = list((1,2))
2 lista.append(3)
3 print(lista)
4
5 tupla = tuple((1,2,3))
6 print(tupla)
7
8 dicionario = dict()
9 dicionario["um"] = 1
10 dicionario["dois"] = 2
11 print(dicionario)
```


Exercícios

Exercício

- 1 (Buracos no texto) Imagine que as letras "A", "D", "O", "P", "R" possuem apenas um buraco. Da mesma forma, a letra "B" possui dois buracos. Já as letras "C", "E", "F", "K" não possuem buracos. Nós devemos considerar que o número de buracos em um texto é igual ao número total de buracos nas letras do texto. Crie um programa que receba um texto e determine quantos buracos existem neste texto.

Dicionários e conjuntos

Listas, Tuplas e Dicionários

Construtores

```
1 # definino os valores de cada letra e quantidade de buracos
2 buracos = {"A":1,"B":2,"D":1,"O":1,"P":1,"Q":1,"R":1}
3
4 # considerando apenas maiusculas
5 texto = input().upper()
6
7 # calculando os buracos com um acumulador
8 cont = 0
9 for l in texto:
10     if l in buracos.keys() :
11         cont = cont + buracos[l]
12
13 # imprimindo
14 print(f"{texto} possui {cont} buracos no texto")
```

Dicionários e conjuntos

Conjuntos

Conjuntos

- permitem armazenamento de diversos itens
- os elementos armazenados NÃO podem ser acessados através de um índice ou chave
- um conjunto é 'desordenado' e não permite duplicatas
- são mutáveis:
 - os elementos armazenados podem ser alterados
 - é possível adicionar e remover elementos

Dicionários e conjuntos

Conjuntos

Conjuntos

```
1 conjunto = {1,2,3}
2 print(conjunto)
3
4 conjunto = {1,2,3,1}
5 print(conjunto)
6
7 conjunto = set((1,2,3,1))
8 print(conjunto)
```

Dicionários e conjuntos

Conjuntos

Conjuntos

```
1 conjunto = {1,2,3}
2 conjunto.add(4)
3 print(conjunto)
4
5 conjunto.remove(4)
6 print(conjunto)
```

Dicionários e conjuntos

Conjuntos

Conjuntos

```
1 conjunto = {1,2,3}
2
3 for x in conjunto:
4     print(x)
```

```
1 conjunto = {1,2,3}
2 print(type(conjunto))
3
4 dic = {1:" ", 2:"b" ,3:"c"}
5 print(type(dic))
```

Dicionários e conjuntos

Conjuntos

Conjuntos - métodos

- `add(item)` - adiciona o 'item' ao conjunto
- `pop()` - retorna e remove um elemento do conjunto
- `remove(item)` - remove o 'item' do conjunto
- `clear()` - remove todos os 'itens' armazenados
- `len(set)` - retorna a quantidade de pares armazenados no 'set'

Exercícios

Exercício

- 1 (Ingresso para o Cinema) A CineSistema possui um controle de venda de ingressos, em sua sala de projeção existem 10 fileiras de poltronas, identificadas por letras e em cada fileira existem 20 poltronas, identificadas por valores inteiros. Crie um programa que realize o controle da alocação das poltronas. Essa programa recebe a indicação da fileira (uma letra) e a indicação da numeração da poltrona (um valor inteiro) e indica se a poltrona está ocupada ou disponível para compra. Caso deseje comprar o ingresso é necessário confirmar. O programa continua em execução até que o usuário selecione a opção para sair.

Dicionários e conjuntos

Listas, Tuplas e Dicionários

```
1 # definindo as fileiras de poltronas
2 # [0]*30 repete 30 vezes o elemento 0 na lista
3 # 0 – poltrona vazia
4 # 1 – poltrona ocupada
5 sala = {"A": [0]*20, "B": [0]*20, "C": [0]*20,
6         "D": [0]*20, "E": [0]*20, "F": [0]*20,
7         "G": [0]*20, "H": [0]*20, "H": [0]*20, "J": [0]*20}
```

o programa continua no próximo slide...

Dicionários e conjuntos

Listas, Tuplas e Dicionários

```
1 while True:
2     for key in sala.keys():
3         print("{} : {}".format(key, sala[key]))
4
5     fila = input("Comprar ingresso para fila: ")
6     num = int(input("Comprar ingresso para poltrona nº: "))
7
8     fila = fila.upper() # considerar apenas maiúsculas
9     num = num - 1 # para o user inicio é 1
10
11    if (fila in sala) and (sala[fila][num] == 0) :
12        conf = input("Poltrona livre, confirma S-sim ou N-Não: ")
13        if (conf == 'S') or (conf == 's') :
14            sala[fila][num] = 1
15    else :
16        print("Poltrona {}:{} ocupada!".format(fila, num))
17
18    opc = input("Deseja outra poltrona S-sim ou N-Não: ")
19    if (opc == 'n') or (opc == 'N') :
20        break
```

Resumo

Resumo

Tipo	Declaração	Mutável	Duplicatas	Acesso
Lista	[]	Sim	Sim	Índices inteiros
Tupla	()	Não	Sim	Índices inteiros
Dicionário	{ }	Sim	Não	Chaves (tipo imutável)
Conjunto	{ }	Sim	Não	somente com iteração

Exercícios

Exercício

- 1 (Controle de estoque) Observe a solicitação no Huxley
<https://www.thehuxley.com/problem/71?quizId=8199>


Leituras recomendadas


 Python tutorial
5.5. Dictionaries
<https://docs.python.org/3/tutorial/datastructures.html#dictionaries>

 Python tutorial
5.4. Sets
<https://docs.python.org/3/tutorial/datastructures.html>

Vídeos recomendados

Para implementar em Python 3

 [Curso em Vídeo \(Canal no Youtube\)](#)
[Curso Python n19 - Dicionários](#)
<https://www.youtube.com/watch?v=ZWj8o692qGY>

 [Boson Treinamentos \(Canal no Youtube\)](#)
[O que são Sets \(Conjuntos\) - Curso de Programação com Python](#)
<https://www.youtube.com/watch?v=7zLXR8Zj9EM>

Perguntas?

Alexandre de Andrade Barbosa
alexandre.barbosa@arapiraca.ufal.br