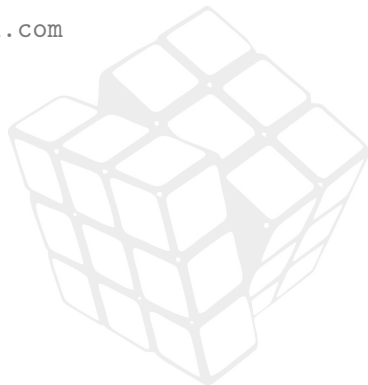


Algoritmos e Programação

Estrutura de Dados - Conjuntos

Gustavo Sávio
gsoprofessor@gmail.com

2017.2



- Um conjunto (set) é uma coleção não ordenada sem elementos duplicados.

```
1 # Criando um conjunto
2 frutas = {'maçã', 'laranja', 'maçã', 'laranja', 'banana'}
```

- Adicionando um elemento em um conjunto.

```
1 marcas_carros = {'fiat', 'chevrolet', 'honda'}
2
3 # Adicionando um elemento por vez
4 marcas_carros.add('hyundai')
5 marcas_carros.add('Ferrari')
6 print(marcas_carros)
```

- Adicionando elementos em um conjunto.

```
1  marcas_carros = {'fiat', 'chevrolet', 'honda'}  
2  
3  # Adicionando n elementos por vez  
4  marcas_carros.update(['hyundai', 'Ferrari'])  
5  print(marcas_carros)
```

- Removendo um elemento.

```
1  marcas_carros = {'fiat', 'chevrolet', 'honda', 'hyundai',  
    ↪    'Ferrari'}  
2  
3  # Remove um elemento e lança uma exceção se não encontrá-lo  
4  marcas_carros.remove('Ferrari')  
5  
6  # Remove um elemento e não lança uma exceção se não encontrá-lo  
7  marcas_carros.discard('Ferrari')  
8  print(marcas_carros)
```

- ▶ A estrutura de dados set possui as operações matemáticas propostas na teoria dos conjuntos
 - ▶ Interseção
 - ▶ União
 - ▶ Diferença
 - ▶ Diferença simétrica
 - ▶ Produto Cartesiano

► Interseção

```
1 conjuntoA = {1, 2, 3, 4, 5}
2 conjuntoB = {5, 6, 7, 8, 9, 10}
3
4 # interseção entre conjuntos
5 resultado = conjuntoA & conjuntoB
6 print(resultado)
7
8 # ou
9 resultado = conjuntoA.intersection(conjuntoB)
10 print(resultado)
```

► União

```
1 conjuntoA = {1, 2, 3, 4, 5}
2 conjuntoB = {5, 6, 7, 8, 9, 10}
3
4 # União
5 resultado = conjuntoA | conjuntoB
6 print(resultado)
7
8 # ou
9 resultado = conjuntoA.union(conjuntoB)
10 print(resultado)
```

► Diferença

```
1 conjuntoA = {1, 2, 3, 4, 5}
2 conjuntoB = {5, 6, 7, 8, 9, 10}
3
4 # Diferença
5 resultado = conjuntoA - conjuntoB
6 print(resultado)
7
8 # ou
9 resultado = conjuntoA.difference(conjuntoB)
10 print(resultado)
```

► Diferença Simétrica

```
1 conjuntoA = {1, 2, 3, 4, 5}
2 conjuntoB = {5, 6, 7, 8, 9, 10}
3
4 # Diferença Simétrica
5 resultado = conjuntoA ^ conjuntoB
6 print(resultado)
7
8 # ou
9 resultado = conjuntoA.symmetric_difference(conjuntoB)
10 print(resultado)
```

► Subconjunto

```
1 conjuntoA = {6, 7, 9}
2 conjuntoB = {5, 6, 7, 8, 9, 10}
3
4 # Subconjunto
5 resultado = conjuntoA <= conjuntoB
6 print(resultado)
7
8 # ou
9 resultado = conjuntoA.issubset(conjuntoB)
10 print(resultado)
```

► Produto cartesiano

```
1 import itertools
2
3 conjuntoA = {6, 7, 9}
4 conjuntoB = {5, 6, 7, 8, 9, 10}
5
6 # produto cartesiano
7 resultado = set(itertools.product(conjuntoA, conjuntoB))
8
9 print(resultado)
```

► Ordenando???

```
1 import collections
2
3 conjuntoA = {6, 7, 9}
4 conjuntoB = {5, 6, 7, 8, 9, 10}
5
6 # Hack
7 resultado = collections.Counter(conjuntoA.union(conjuntoB))
8 print(set(resultado))
```

► Copiando



```
1 conjuntoA = {6, 7, 9}
2
3 # Copiando
4 resultado = conjuntoA.copy()
5 print(resultado)
```

► Limpando

```
1 conjuntoA = {6, 7, 9}
2
3 # Limpando
4 conjuntoA.clear()
5 print(conjuntoA)
```

► Iterando

```
1 conjunto = {6, 7, 9}
2
3 # iterando com for
4 for valor in conjunto:
5     print(valor)
6
7 # usando enumerate
8 for indice, valor in enumerate(conjunto):
9     print(indice, ' ', valor)
```

-  Allen B. Downey; Think Python How to Think Like a Computer Scientist.
-  Docs Python 3. <https://docs.python.org/3/>