Conjuntos

Prof. Alberto Costa Neto

Programação em Python

Conjuntos

- Um objeto conjunto é uma coleção não ordenada de objetos distintos com suporte a *hash*, tendo os seguintes usos:
  - Testes de associação
  - Remover duplicatas de uma sequência
  - Computar operações matemáticas tais como interseção, união, diferença e diferença simétrica.

- Não suportam operações de slicing e acesso posicional (indexação) a elementos

Conjuntos

- Um Conjunto é tipo de estrutura de dados que representa um conjunto não ordenado de valores únicos.
- Suportam as operações len, in, for _ in set, max, min, sum

```
>>> x = {1,2,3,4,5,6,5}
>>> print(x)
{1, 2, 3, 4, 5, 6}
>>> y = {1,9,2}
>>> print(y)
{1, 2, 9}
>>> print(max(y))
9
```

```
>>> for iter in y:
...     print(iter)
...
1
2
9
>>>
```

Conjuntos

- Também podem ser gerados através de compreensão de conjuntos

```
>>> nums = {1,2,3,4,5,6,7,5,2,4,7,8,9}
>>> print(set(nums))
{1, 2, 3, 4, 5, 6, 7, 8, 9}

>>> impares = {x for x in nums if x%2==1}
>>> print(impares)
{1, 3, 5, 7, 9}
```

Métodos de conjuntos

- add: adiciona o valor no conjunto, se já não estiver lá
- discard: descarta o valor passado, retirando-o do conjunto. Não gera erro quando não consegue.
- remove: remove o valor passado do conjunto. Gera erro se não conseguir
- pop: retira um valor arbitrário do conjunto
- update: atualiza o conjunto com uma sequência de valores
- clear: limpa o conjunto (remove todos os valores)
- copy: cria uma cópia do conjunto

Métodos add e remove

```
>>> nums = {1,2,3,4,5}
>>> print(nums)
{1, 2, 3, 4, 5}
>>> nums.add(6)
>>> nums.add(4) # já está no conjunto, logo nada muda
>>> print(nums)
{1, 2, 3, 4, 5, 6}
>>> nums.remove(6)
>>> nums.remove(6) # gera erro pois o 6 não existe
KeyError: 6
>>> print(nums)
{1, 2, 3, 4, 5}
```

Métodos discard e pop

```
>>> nums = {1,2,3}
>>> nums.discard(3)
>>> nums.discard(3) # não gera erro
>>> print(nums)
{1, 2}
>>> nums.pop() # remove e retorna um valor arbitrário
1
>>> nums.pop() # remove e retorna um valor arbitrário
2
>>> nums.pop() # gera um erro, pois o set está vazio
KeyError: 'pop from an empty set'
```

Método update

```
>>> nums = {1}
>>> nums.update( (2,3) ) # atualizada com tupla
>>> print(nums)
{1, 2, 3}
>>> nums.update( [3,4,5] ) # atualiza com uma lista
>>> print(nums)
{1, 2, 3, 4, 5}
>>> nums.update( {5,6,7} ) # atualiza com um conjunto
>>> print(nums)
{1, 2, 3, 4, 5, 6, 7}
```

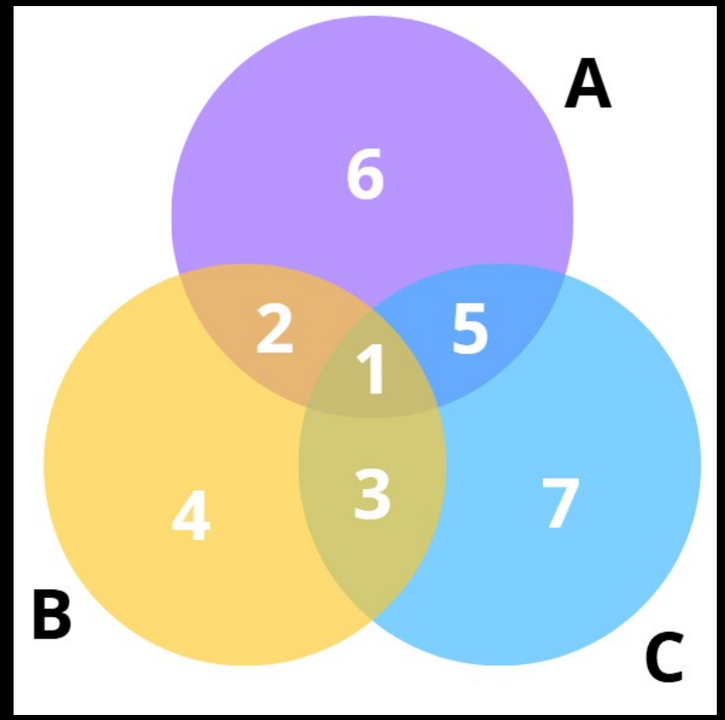
Operações matemáticas em Sets (métodos)

Um Set possui uma série de métodos que implementam as operações matemáticas sobre conjuntos

- União – union
- Intersecção - intersection
- Diferença – difference
- Diferença simétrica - symmetric_difference
- (Não) Está contido – not in / in
- Subconjunto – issubset
- Superconjunto – issuperset

União

- Retorna um novo conjunto com elementos do conjunto e de todos os outros



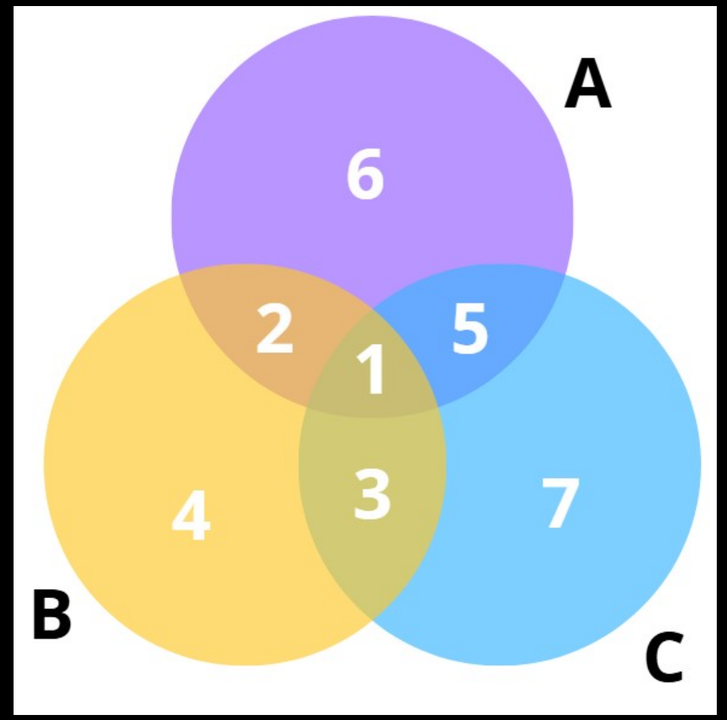

```
>>> A = {1,2,5,6}
>>> B = {1,2,3,4}
>>> C = {1,3,5,7}
>>> print(A.union(B))
{1, 2, 3, 4, 5, 6}
>>> print(B | C)
{1, 2, 3, 4, 5, 7}
```

```
>>> print(A.union(B).union(C))
{1, 2, 3, 4, 5, 6, 7}

>>> print(A | B | C)
{1, 2, 3, 4, 5, 6, 7}
```

Interseção

- Retorna um novo conjunto com elementos comuns do conjunto e de todos os outros

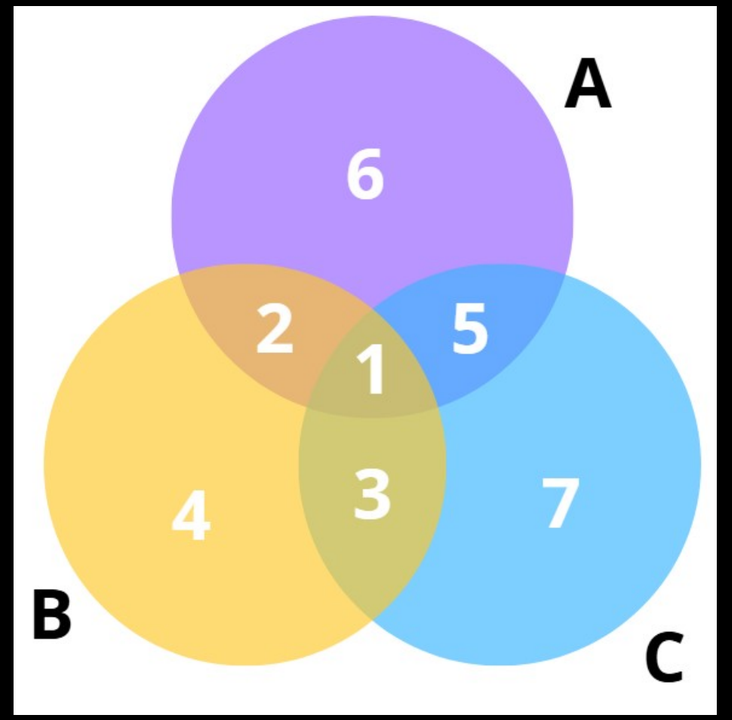



```
>>> A = {1,2,5,6}
>>> B = {1,2,3,4}
>>> C = {1,3,5,7}
>>> print(A.intersection(B))
{1, 2}
>>> print(A & B)
{1, 2}
```

```
>>> print(B & C)
{1, 3}
>>> print(A & C)
{1, 5}
>>> print(A & B & C)
{1}
```

Diferença

- Retorna um novo conjunto com elementos no conjunto que não estão nos outros.

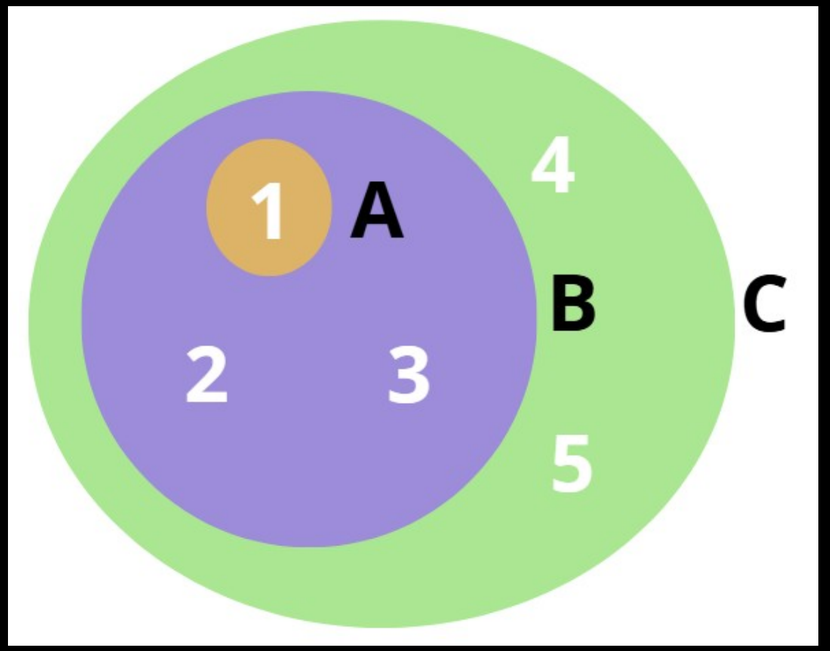



```
>>> A = {1,2,5,6}
>>> B = {1,2,3,4}
>>> C = {1,3,5,7}
>>> print(A.difference(B))
{5, 6}
>>> print(B - C)
{2, 4}
```

```
>>> print(B - A - C)
{4}
```

Subconjunto

- Testa se cada elemento do conjunto está contido no outro

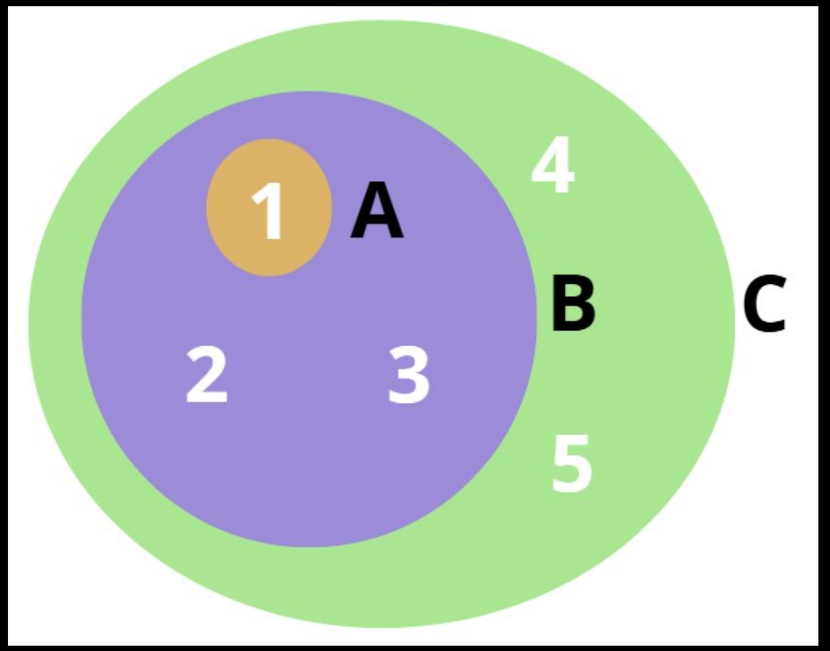



```
>>> A = {1}
>>> B = {1,2,3}
>>> C = {1,2,3,4,5}
>>> print(A.issubset(B))
True
>>> print(A < B)
True
```

```
>>> print(A < C)
True
>>> print(C < B)
False
>>> print(A < A)
False
>>> print(A <= A)
True
```

Superconjunto

- Testa se cada elemento do outro conjunto está contido no conjunto.

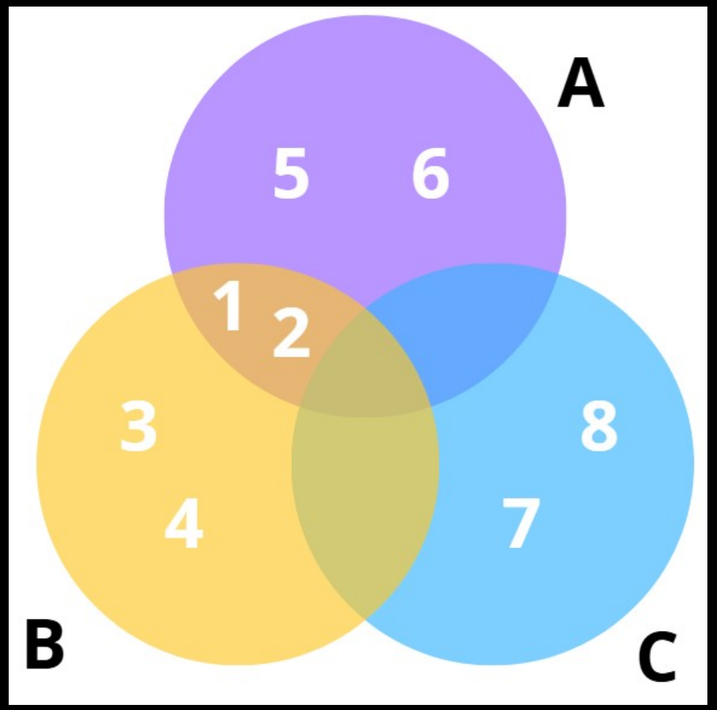



```
>>> A = {1}
>>> B = {1,2,3}
>>> C = {1,2,3,4,5}
>>> print(B.issuperset(A))
True
>>> print(B > A)
True
```

```
>>> print(C > A)
True
>>> print(B > C)
False
>>> print(A > A)
False
>>> print(A >= A)
True
```

Disjunto

- Retorna True se o conjunto não tem elementos em comum com outro. Conjuntos são disjuntos se e somente se a sua interseção é o conjunto vazio.




```
>>> A = {1,2,5,6}
>>> B = {1,2,3,4}
>>> C = {7,8}
>>> print(A.isdisjoint(B))
False
>>> print(C.isdisjoint(A))
True
```

Conjuntos Congelados (frozen)

- De forma semelhante a tuplas, frozenset são conjuntos imutáveis, ou seja, após serem criados não podem ser modificados

```
>>> nums = frozenset([1,2,3,4,5,6,7,5,2,4,7,8,9])
>>> print(nums)
frozenset({1, 2, 3, 4, 5, 6, 7, 8, 9})
>>> nums.add(10)
AttributeError: 'frozenset' object has no
attribute 'add'
```