Conjuntos

Guilherme Arthur de Carvalho

Analista de sistemas

@decarvalhogui

Objetivo Geral

Entender o funcionamento da estrutura de dados set.

Pré-requisitos

- Python 3
- VSCode

Percurso

Etapa 1 **Como criar conjuntos**

Etapa 2 Métodos da classe set

Etapa 1

Como criar conjuntos

Criando sets

Um set é uma coleção que não possui objetos repetidos, usamos sets para representar conjuntos matemáticos ou eliminar itens duplicados de um iterável.

Exemplo

```
set([1, 2, 3, 1, 3, 4])  # {1, 2, 3, 4}

set("abacaxi")  # {"b", "a", "c", "x", "i"}

set(("palio", "gol", "celta", "palio"))  # {"gol", "celta", "palio"}
```

Acessando os dados

Conjuntos em Python não suportam indexação e nem fatiamento, caso queira acessar os seus valores é necessário converter o conjunto para lista.

Exemplo

```
numeros = {1, 2, 3, 2}

numeros = list(numeros)

numeros[0]
```

Iterar conjuntos

A forma mais comum para percorrer os dados de um conjunto é utilizando o comando **for**.

Exemplo

```
carros = {"gol", "celta", "palio"}

for carro in carros:
    print(carro)
```

Função enumerate

Às vezes é necessário saber qual o índice do objeto dentro do laço **for**. Para isso podemos usar a função **enumerate**.

Exemplo

```
carros = {"gol", "celta", "palio"}

for indice, carro in enumerate(carros):
    print(f"{indice}: {carro}")
```

Percurso

~~Etapa 1~~ ~~Criação e acesso aos dados~~

Etapa 2 **Métodos da classe set**

Etapa 2

Métodos da classe set

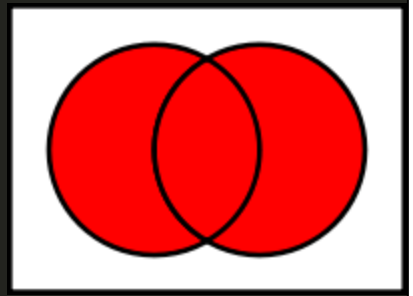
Percurso

~~Etapa 1~~ ~~Criação e acesso aos dados~~

~~Etapa 2~~ ~~Métodos da classe set~~

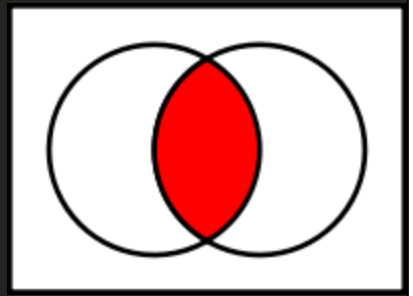
{}.union

```
conjunto_a = {1, 2}
conjunto_b = {3, 4}

conjunto_a.union(conjunto_b)  # {1, 2, 3, 4}
```



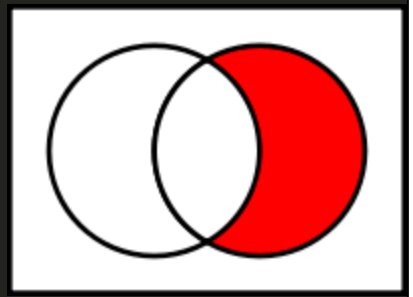
{}.intersection

```
conjunto_a = {1, 2, 3}
conjunto_b = {2, 3, 4}

conjunto_a.intersection(conjunto_b)  # {2, 3}
```



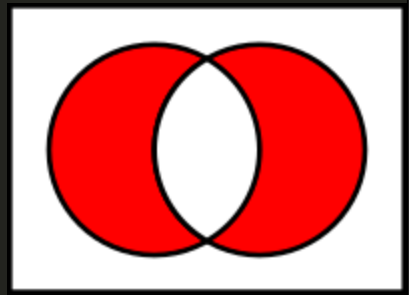
{}.difference

```
conjunto_a = {1, 2, 3}
conjunto_b = {2, 3, 4}

conjunto_a.difference(conjunto_b)  # {1}
conjunto_b.difference(conjunto_a)  # {4}
```



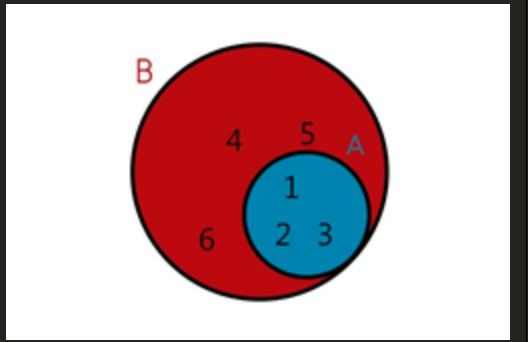
{}.symmetric_difference

```
conjunto_a = {1, 2, 3}
conjunto_b = {2, 3, 4}

conjunto_a.symmetric_difference(conjunto_b)  # {1, 4}
```



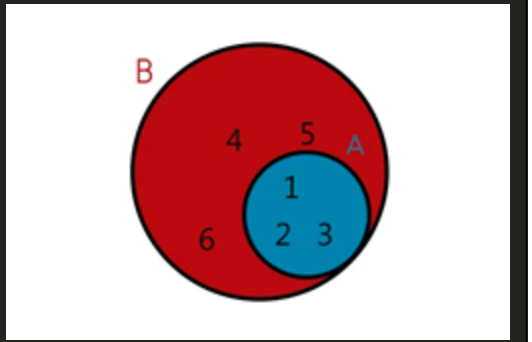
{}.issubset

```
conjunto_a = {1, 2, 3}
conjunto_b = {4, 1, 2, 5, 6, 3}

conjunto_a.issubset(conjunto_b)  # True
conjunto_b.issubset(conjunto_a)  # False
```



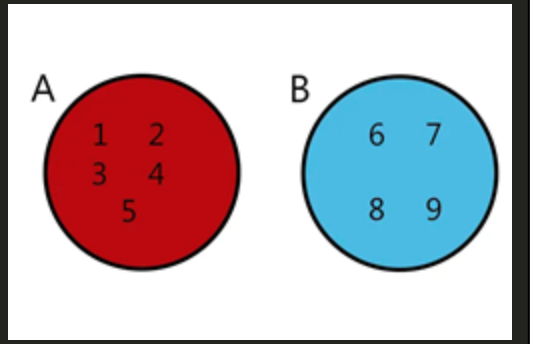
{}.issuperset

```
conjunto_a = {1, 2, 3}
conjunto_b = {4, 1, 2, 5, 6, 3}

conjunto_a.issuperset(conjunto_b)  # False
conjunto_b.issuperset(conjunto_a)  # True
```



{}.isdisjoint

```
conjunto_a = {1, 2, 3, 4, 5}
conjunto_b = {6, 7, 8, 9}
conjunto_c = {1, 0}

conjunto_a.isdisjoint(conjunto_b)  # True
conjunto_a.isdisjoint(conjunto_c)  # False
```



{}.add

```
sorteio = {1, 23}

sorteio.add(25)  # {1, 23, 25}
sorteio.add(42)  # {1, 23, 25, 42}
sorteio.add(25)  # {1, 23, 25, 42}
```

{}.clear

```
sorteio = {1, 23}

sorteio  # {1,23}
sorteio.clear()
sorteio  # {}
```

{}.copy

```
sorteio = {1, 23}

sorteio  # {1, 23}
sorteio.copy()
sorteio  # {1, 23}
```

{}.discard

```
numeros = {1, 2, 3, 1, 2, 4, 5, 5, 6, 7, 8, 9, 0}

numeros  # {1, 2, 3, 4, 5, 6, 7, 8, 9, 0}
numeros.discard(1)
numeros.discard(45)
numeros  # {2, 3, 4, 5, 6, 7, 8, 9, 0}
```

{}.pop

```
numeros = {1, 2, 3, 1, 2, 4, 5, 5, 6, 7, 8, 9, 0}

numeros  # {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
numeros.pop()  # 0
numeros.pop()  # 1
numeros  # {2, 3, 4, 5, 6, 7, 8, 9}
```

{}.remove

```
numeros = {1, 2, 3, 1, 2, 4, 5, 5, 6, 7, 8, 9, 0}

numeros  # {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
numeros.remove(0)  # 0
numeros  # {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

len

```
numeros = {1, 2, 3, 1, 2, 4, 5, 5, 6, 7, 8, 9, 0}

len(numeros)  # 10
```

in

```
numeros = {1, 2, 3, 1, 2, 4, 5, 5, 6, 7, 8, 9, 0}

1 in numeros  # True
10 in numeros  # False
```

Links Úteis

- https://github.com/digitalinnovationone/trilha-python-dio

Dúvidas?

> Fórum/Artigos

> Comunidade Online (Discord)