

Sets

Sets em Python são coleções não ordenadas de elementos únicos. Eles são usados quando a presença de um item na coleção é mais importante do que a ordem ou quantas vezes ele aparece. Utilizando sets, é possível realizar operações matemáticas de conjuntos como união, interseção, diferença, e diferença simétrica.

Caraterísticas dos Sets

- **Únicos:** Os sets automaticamente removem duplicados. Se adicionar o mesmo item mais de uma vez ao set, ele aparecerá na coleção apenas uma vez.
- **Não ordenados:** Os itens em um set não têm uma ordem fixa; não pode esperar que eles apareçam na mesma ordem em que foram adicionados.
- **Mutáveis:** pode adicionar ou remover itens de um set depois de sua criação.
- **Elementos imutáveis:** Os elementos que adiciona a um set precisam ser imutáveis (como números, strings, e tuplos) porque os sets precisam de uma forma de garantir a unicidade dos seus elementos. Tipos mutáveis como listas não podem ser adicionados a um set. Os elementos não têm de ser todos do mesmo tipo

Definir e adicionar

```
In [ ]: meu_set={1,2,3,4,5}
        print(meu_set)

        #adicionar
        meu_set.add(6)
        print(meu_set)

        meu_set.add("maria")
        print(meu_set)

        #elemento duplicado
        meu_set.add(1)
        print(meu_set)
```

Remover um elemento

```
In [ ]: meu_set={1,2,3,4,5}

        meu_set.remove(3)
        print(meu_set)
```

Listar

```
In [ ]: meu_set={1,2,3,4}
        for x in meu_set:
            print(x)
```

```
#este código não funciona
#for i in range(len(meu_set)):
#    print(meu_set[i]) #não podemos fazer isto com sets

for posicao, valor in enumerate(meu_set, start=1):
    print(f"Elemento da posição {posicao}: {valor}")
```

Verificar se existe

```
In [ ]: meu_set={1,2,3,4}

if 2 in meu_set:
    print("0 2 existe no set")
else:
    print("0 2 não existe no set")
```

Função enumerate

A função **enumerate()** pode ser utilizada com outros tipos de coleções

Listas

```
In [ ]: lista = ['maçã', 'banana', 'cereja']
for indice, valor in enumerate(lista, start=0):
    print(indice, valor)
```

Tuplos

```
In [ ]: tuplo = ('a', 'b', 'c')
for indice, valor in enumerate(tuplo, start=1):
    print(indice, valor)
```

Strings

```
In [ ]: string = "olá"
for indice, valor in enumerate(string, start=0):
    print(indice, valor)
```

Dicionários

```
In [ ]: dicionario = {'a': 1, 'b': 2, 'c': 3}
for indice, (chave, valor) in enumerate(dicionario.items(), start=1):
    print(indice, chave, valor)
```

Operações de conjuntos

```
In [ ]: meu_set={1,2,3,4,5}
outro_set={4,5,6,7,8}
```

União

```
In [ ]: #União
        uniao=meu_set.union(outro_set)
        print(uniao)
```

Interseção

```
In [ ]: intersecao = meu_set.intersection(outro_set)
        print(intersecao)
```

Diferença

```
In [ ]: diferenca = meu_set.difference(outro_set)
        print(diferenca)
```

Diferença simétrica

```
In [ ]: diferenca_simetrica = meu_set.symmetric_difference(outro_set)
        print(diferenca_simetrica)
```