



Explorando Conjuntos em Python

O que são Sets (`set`) em Python?

Os **sets** são coleções **sem elementos repetidos**!

Eles são úteis para:

- ✓ Representar **conjuntos matemáticos**.
- ✓ **Remover duplicatas** automaticamente.
- ✓ **Fazer operações de união, interseção e diferença** rapidamente.

1 Como Criar um Set?

✓ Usando chaves `{ }` (mais comum)

```
numeros = {1, 2, 3, 4, 5}
print(numeros)
# Saída: {1, 2, 3, 4, 5}
```

✓ Usando a função `set()`

```
numeros = set([1, 2, 2, 3, 4, 4, 5])
print(numeros)
# Saída: {1, 2, 3, 4, 5} (sem duplicatas!)
```

2 Características dos Sets

- ◆ Elementos não se repetem.
- ◆ A ordem não é garantida (diferente de listas e tuplas).
- ◆ Podemos adicionar e remover elementos, mas não modificar diretamente.

3 Como Adicionar e Remover Elementos

✓ `add()` → Adiciona um elemento ao set.

```
numeros.add(6)
print(numeros)
# Saída: {1, 2, 3, 4, 5, 6}
```

✓ `remove()` → Remove um elemento (gera erro se não existir).

```
numeros.remove(3)
print(numeros)
# Saída: {1, 2, 4, 5, 6}
```

✓ `discard()` → Remove um elemento **sem erro** caso não exista.

```
numeros.discard(10) # Não faz nada, pois 10 não está no set
```

✓ `pop()` → Remove um elemento **aleatório**.

```
elemento_removido = numeros.pop()
print(elemento_removido)
```

✓ `clear()` → Esvazia o set.

```
numeros.clear()
print(numeros)
# Saída: set()
```

4 Operações com Sets

✓ **União (|)** → Junta dois sets, eliminando duplicatas.

```
a = {1, 2, 3}
b = {3, 4, 5}
print(a | b) # União
# Saída: {1, 2, 3, 4, 5}
```

✓ **Interseção (&)** → Mantém apenas os elementos comuns.

```
print(a & b) # Interseção  
# Saída: {3}
```

✓ **Diferença (-)** → Elementos que estão em um set, mas não no outro.

```
print(a - b) # Diferença de A para B  
# Saída: {1, 2}
```

✓ **Diferença Simétrica (^)** → Elementos que estão em um dos sets, mas não nos dois.

```
print(a ^ b) # Diferença Simétrica  
# Saída: {1, 2, 4, 5}
```

Resumo Rápido

Operação	Código Exemplo	Saída
Criar set	<code>s = {1, 2, 3}</code>	<code>{1, 2, 3}</code>
Adicionar elemento	<code>s.add(4)</code>	<code>{1, 2, 3, 4}</code>
Remover elemento	<code>s.remove(2)</code>	<code>{1, 3}</code>
União	<code>`s1</code>	<code>s2`</code>
Interseção	<code>s1 & s2</code>	<code>{2}</code>
Diferença	<code>s1 - s2</code>	<code>{1}</code>
Diferença Simétrica	<code>s1 ^ s2</code>	<code>{1, 4}</code>