

E d u c a ç ã o  
P r o f i s s i o n a l  
P a u l i s t a

Técnico em  
Ciência de  
Dados

# Variáveis e tipos de dados

## Tuplas e conjuntos

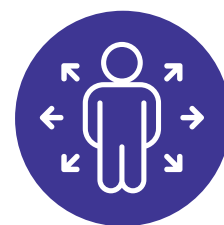
### Aula 2

Código da aula: [DADOS]ANO1C2B2S14A2



## Objetivo da Aula

Introduzir o conceito de conjuntos (set) em Python.



## Competências da Unidade (Técnicas e Socioemocionais)

- Ser proficiente em linguagens de programação, para manipular e analisar grandes conjuntos de dados;
- Usar técnicas para explorar e analisar dados, aplicar modelos estatísticos, identificar padrões, realizar inferências e tomar decisões baseadas em evidências;
- Colaborar efetivamente com outros profissionais, como cientistas de dados e engenheiros de dados; trabalhar em equipes multifuncionais colaborando com colegas, gestores e clientes.



## Recursos Didáticos

- Recurso audiovisual para a exibição de vídeos e imagens;
- Acesso ao laboratório de informática e/ou à internet;
- Software Anaconda/Jupyter Notebook instalado ou similar.



## Duração da Aula

50 minutos.

## Tipo de dados set

Em Python, um conjunto, representado pelo tipo de dados set, **é uma coleção desordenada e mutável de elementos distintos.**

A principal característica de um conjunto é que ele não permite elementos duplicados, o que significa que cada elemento em um conjunto é **único**. A sintaxe para criar um conjunto em Python é utilizando chaves `{}`. Por exemplo:

```
meu_conjunto = {1, 2, 3, 4, 5}
```

Elaborado especialmente para o curso com Jupyter Notebook.



### Atenção!

Não confunda dicionário e set, uma vez que ambos usam `{}`. Dicionários são estruturas chave-valor com acesso por chave, conjuntos são coleções desordenadas de elementos distintos.

## Tipo de dados set

Principais características e operações de conjuntos em Python:

- **Unicidade:** um conjunto não permite elementos duplicados. Se você tentar adicionar um elemento que já está presente, o conjunto permanecerá inalterado.

```
conjunto = {1, 2, 3, 3, 4}  
print(conjunto)
```

```
{1, 2, 3, 4}
```

Elaborado especialmente para o curso com Jupyter Notebook.



## Tipo de dados set

- **Mutabilidade:** os conjuntos são mutáveis, o que significa que você pode adicionar e remover elementos após a criação.

```
conjunto = {1, 2, 3}
conjunto.add(4)      # Adiciona o elemento 4 ao conjunto
conjunto.remove(2)   # Remove o elemento 2 do conjunto
print(conjunto)
```

```
{1, 3, 4}
```

Elaborado especialmente para o curso com Jupyter Notebook.

## Tipo de dados set

**Operações de conjuntos:** conjuntos suportam operações como união, interseção e diferença.

```
conjunto1 = {1, 2, 3}
conjunto2 = {3, 4, 5}

uniao = conjunto1.union(conjunto2)      # União
intersecao = conjunto1.intersection(conjunto2) # Interseção
diferenca = conjunto1.difference(conjunto2) # Diferença

print(uniao)          # Saída: {1, 2, 3, 4, 5}
print(intersecao)     # Saída: {3}
print(diferenca)      # Saída: {1, 2}
```

{1, 2, 3, 4, 5}  
{3}  
{1, 2}

Elaborado especialmente para o curso com Jupyter Notebook.

## Tipo de dados set

Pode-se também usar outros operadores.

```
conjunto1 = {1, 2, 3}
conjunto2 = {3, 4, 5}

# União
uniao = conjunto1 | conjunto2

# Interseção
intersecao = conjunto1 & conjunto2

# Diferença
diferenca = conjunto1 - conjunto2

print(uniao)          # Saída: {1, 2, 3, 4, 5}
print(intersecao)     # Saída: {3}
print(diferenca)      # Saída: {1, 2}
```

```
{1, 2, 3, 4, 5}
{3}
{1, 2}
```

Elaborado especialmente para o curso com Jupyter Notebook.



Vamos  
fazer uma  
**atividade**

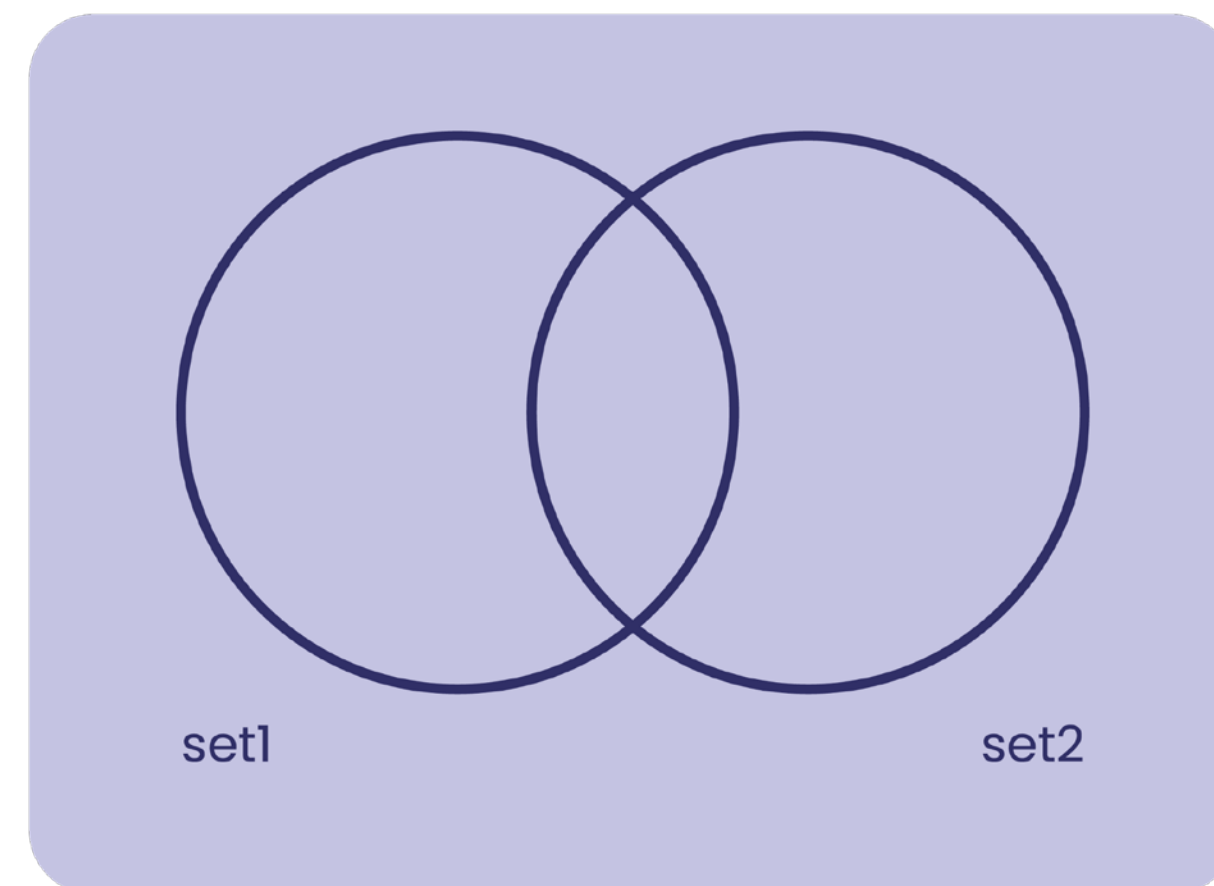
## Exercícios – Exemplos

Temos 2 conjuntos.

$\text{set1} = \{1, 2, 3, 4, 5\}$

$\text{set2} = \{4, 5, 6, 7, 8\}$

1. Realize as 3 operações apresentadas anteriormente, na mão. Não é para programar ainda.
2. Preencha também a imagem ao lado.



Elaborado especialmente para o curso.



**Individual**



**15 min**

Vamos  
fazer uma  
**atividade**

## Exemplo: resolução por Python

```
1 # Importando a biblioteca necessária
2 import matplotlib.pyplot as plt
3
4 # Criando dois conjuntos
5 set1 = {1, 2, 3, 4, 5}
6 set2 = {4, 5, 6, 7, 8}
7
8 # Realizando operações de conjunto
9 union = set1.union(set2)
10 intersection = set1.intersection(set2)
11 difference = set1.difference(set2)
12
13 # Plotando os conjuntos e suas operações
14 fig, ax = plt.subplots(figsize=(6, 4))
15 ax.set_title("Operações de Conjuntos")
16 ax.set_xlabel("Elementos")
17 ax.set_ylabel("Conjuntos")
18 ax.set_yticks([1, 2, 3])
19 ax.set_yticklabels(["Conjunto 1", "Conjunto 2", "Operações"])
20 ax.scatter(list(set1), [1]*len(set1), color="blue", label="Conjunto 1")
21 ax.scatter(list(set2), [2]*len(set2), color="red", label="Conjunto 2")
22 ax.scatter(list(union), [3]*len(union), color="green", label="União", s=100)
23 ax.scatter(list(intersection), [3]*len(intersection), color="orange", label="Interseção")
24 ax.scatter(list(difference), [3]*len(difference), color="purple", label="Diferença")
25 ax.legend()
26 plt.show()
```



**Individual**



**15 min**

Elaborado especialmente para o curso com Jupyter Notebook.

Vamos  
fazer uma  
**atividade**

## Exemplo: resolução por Python

```
print(f"União: {union}")  
print(f"Intersecção: {intersection}")  
print(f"Diferença: {difference}")
```

União: {1, 2, 3, 4, 5, 6, 7, 8}

Intersecção: {4, 5}

Diferença: {1, 2, 3}

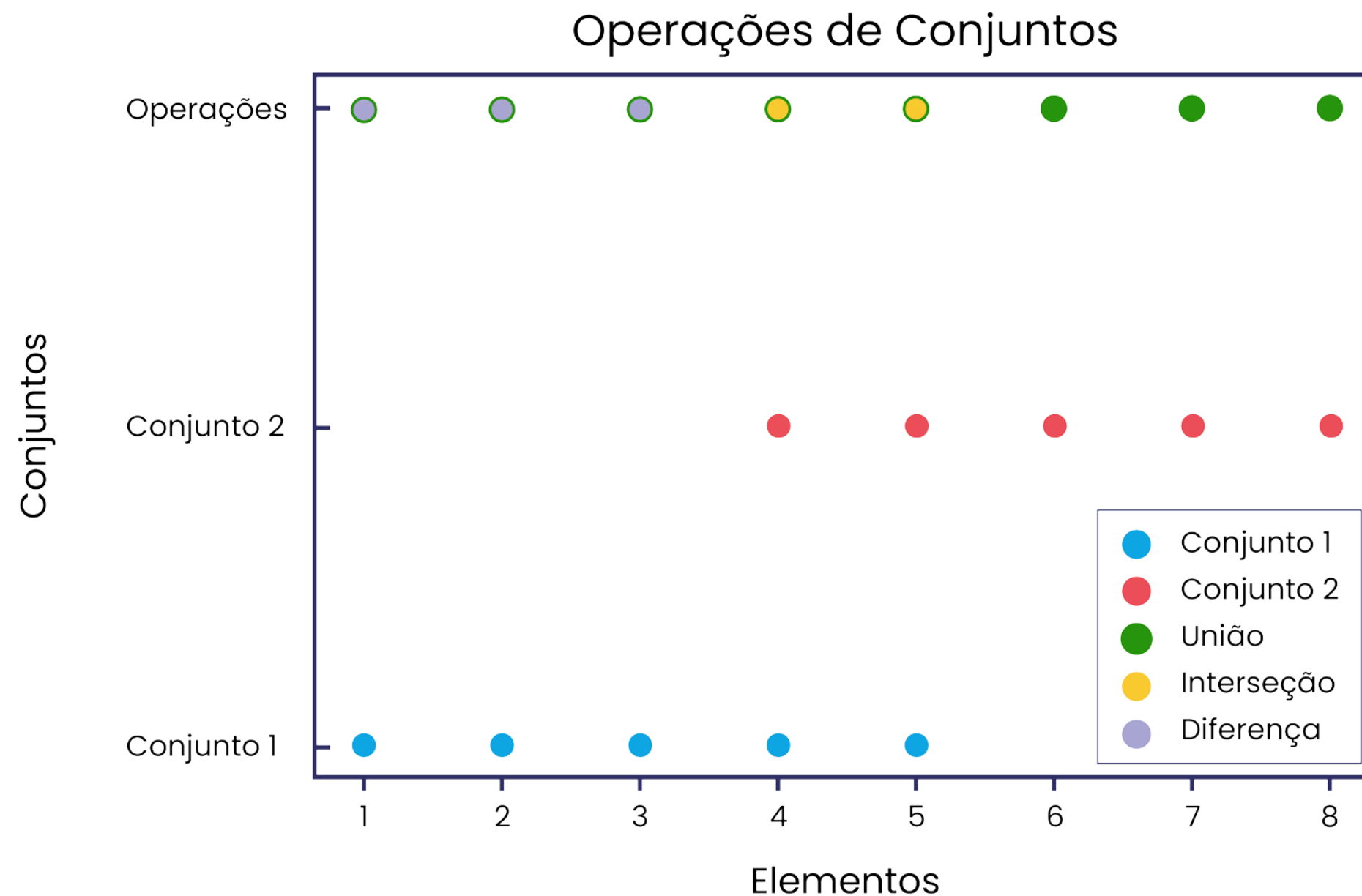
Elaborado especialmente para o curso com Jupyter Notebook.



**Individual**



**15 min**



Elaborado especialmente para o curso.

## Exposição

# Onde aprender mais?

Não há muito material em português sobre **conjuntos (set)**, mas podemos usar IA generativa para complementar nossos estudos.



### Curiosidade

A inteligência artificial generativa (IA generativa) refere-se a uma classe de modelos de inteligência artificial projetados para gerar novos dados, que se assemelham a um conjunto existente de dados.

Esses modelos são treinados para aprender as características e os padrões dos dados de treinamento e, em seguida, são capazes de criar novas instâncias que se assemelham a esses padrões.

# Exposição

## Onde aprender mais?

As IA generativas têm diversas aplicações, como:



### Geração de imagens

Criar imagens realistas de pessoas, objetos ou paisagens.



### Geração de texto

Produzir texto que se assemelha ao estilo e ao conteúdo de um conjunto de treinamento.



### Criação de música

Compor novas peças musicais com base em padrões aprendidos de músicas existentes.



### Geração de vídeos

Criar vídeos sintéticos com base em sequências de treinamento.



### Design de produtos

Gerar designs de produtos com base em exemplos existentes.



## Exposição

### Onde aprender mais?

Procure na internet alguns exemplos de IA generativa de texto. Em seguida, peça uma definição do que é conjunto (set) em Python.



#### Atenção!

Modelos de linguagem generativos, como o GPT (Generative Pre-trained Transformer), podem ser treinados para criar texto em um estilo semelhante ao de um autor específico. Esses modelos são capazes de gerar histórias, artigos ou até mesmo código de programação.



© Getty Images

O que nós  
**aprendemos  
hoje?**

## Hoje desenvolvemos:

- 1** O conhecimento do conceito de conjuntos (set) em Python;
- 2** A compreensão dos tipos de dados set;
- 3** A aplicação prática do conteúdo abordado nesta aula por meio de um exercício.

# Saiba mais

Quer rever o conceito de conjuntos e dicionário?

ALURA. Python Collections parte 2: conjuntos e dicionários, [s.d.]. Disponível em:

<https://www.alura.com.br/curso-online-python-collections-conjuntos-e-dicionarios>.

Acesso em: 27 mar. 2024.

# Referências da aula

PYTHON. *Built-in Types*, [s.d.]. Disponível em:  
<https://docs.python.org/3/library/stdtypes.html#set-types-set-frozenset>.  
Acesso em: 27 mar. 2024.

Identidade visual: imagens © Getty Images

Ed u c a ç ã o  
P r o f i s s i o n a l  
P a u l i s t a

Técnico em  
Ciência de  
Dados