



Estruturas Compostas



Alessandro Botelho Bovo

Objetivo da aula

- ▶ Praticar o uso combinado de **tuplas, listas, dicionários e conjuntos**.
- ▶ Resolver problemas reais evitando duplicações, organizando registros e permitindo comparações.
- ▶ Trabalhar sempre de forma explícita, sem atalhos como *comprehensions*.



Relembrando as estruturas básicas

- ▶ Lista (`list`) → estrutura ordenada, mutável, acessada por **índices numéricos**.
- ▶ Tupla (`tuple`) → estrutura ordenada, **imutável**, acessada por **índices numéricos**.
- ▶ Dicionário (`dict`) → coleção de **pares chave : valor**, mutável, acessada pela chave.
- ▶ Conjunto (`set`) → coleção de elementos **únicos e não ordenados**, permite operações matemáticas de **união, interseção e diferença**.



Lista de dicionários + conjunto

- **Problema:** Lista de alunos com cursos repetidos. Mostrar apenas os cursos únicos.

```
alunos = [  
    {'nome': 'Ana', 'curso': 'Engenharia'},  
    {'nome': 'Bruno', 'curso': 'Computação'},  
    {'nome': 'Carla', 'curso': 'Engenharia'},  
    {'nome': 'Daniel', 'curso': 'Química'}  
]
```

Cursos disponíveis:
Engenharia
Química
Computação



Mostrar apenas os cursos únicos

```
alunos = [  
    {'nome': 'Ana', 'curso': 'Engenharia'},  
    {'nome': 'Bruno', 'curso': 'Computação'},  
    {'nome': 'Carla', 'curso': 'Engenharia'},  
    {'nome': 'Daniel', 'curso': 'Química'}  
]
```

```
cursos = set()  
for aluno in alunos:  
    cursos.add(aluno['curso'])
```

```
print('Cursos disponíveis: ')  
for c in cursos:  
    print(c)
```

Cursos disponíveis:
Engenharia
Química
Computação



Lista de dicionários + tupla

- **Problema:** Guardar localização de lojas (nome + coordenadas). Achar a localização de uma loja.

```
lojas = [  
    {"nome": "Loja A", "localizacao": (23.3, 51.2)},  
    {"nome": "Loja B", "localizacao": (23.4, 51.5)},  
    {"nome": "Loja C", "localizacao": (23.6, 51.3)}  
]
```

```
Loja A fica em (23.3, 51.2)  
Loja B fica em (23.4, 51.5)  
Loja C fica em (23.6, 51.3)
```

```
Digite o nome de uma loja: Loja C  
Loja C fica em (23.6, 51.3)
```



Achar a localização de uma loja

```
lojas = [  
    {"nome": "Loja A", "localizacao": (23.3, 51.2)},  
    {"nome": "Loja B", "localizacao": (23.4, 51.5)},  
    {"nome": "Loja C", "localizacao": (23.6, 51.3)}  
]  
  
for loja in lojas:  
    print(loja["nome"], "fica em", loja["localizacao"])  
  
nome_loja = input('\nDigite o nome de uma loja: ')  
  
for loja in lojas:  
    if nome_loja == loja["nome"]:  
        print(loja["nome"], "fica em", loja["localizacao"])  
        break
```

```
Loja A fica em (23.3, 51.2)  
Loja B fica em (23.4, 51.5)  
Loja C fica em (23.6, 51.3)
```

```
Digite o nome de uma loja: Loja C  
Loja C fica em (23.6, 51.3)
```



Conjunto de tuplas

- ▶ **Problema:** Evitar repetição de pares (aluno, disciplina).

Ana -> Matemática

Bruno -> Física

Ana -> Química

Bruno -> Física

Matrículas registradas:

Bruno está matriculado em Física

Ana está matriculado em Química

Ana está matriculado em Matemática



Evitar repetição de pares (aluno, disciplina).

```
matriculas = set()

matriculas.add(("Ana", "Matemática"))
matriculas.add(("Bruno", "Física"))
matriculas.add(("Ana", "Química"))
matriculas.add(("Bruno", "Física")) # repetida

print("Matrículas registradas:")

for aluno, disc in matriculas:
    print(f'{aluno} está matriculado em {disc}')
```

```
Matrículas registradas:
Bruno está matriculado em Física
Ana está matriculado em Química
Ana está matriculado em Matemática
```



Dicionário com chave como tupla

- **Problema:** Representar distâncias entre cidades. Como achar a distância entre cidades.

```
distancias = {  
    ('Londrina', 'Curitiba'): 380,  
    ('Curitiba', 'São Paulo'): 408,  
    ('Londrina', 'Maringá'): 100  
}
```

```
Digite a cidade de origem: Londrina  
Digite a cidade de destino: Curitiba  
Distância Londrina-Curitiba: 380
```



Como achar a distância entre cidades

```
distancias = {  
    ('Londrina', 'Curitiba'): 380,  
    ('Curitiba', 'São Paulo'): 408,  
    ('Londrina', 'Maringá'): 100  
}  
  
origem = input('Digite a cidade de origem: ')  
destino = input('Digite a cidade de destino: ')  
  
print(f'Distância {origem}-{destino}:', distancias[(origem, destino)])
```

```
Digite a cidade de origem: Londrina  
Digite a cidade de destino: Curitiba  
Distância Londrina-Curitiba: 380
```



Conjunto + lista de dicionários

- **Problema:** Encontrar contatos em comum entre duas agendas.

```
agenda1 = [  
    {"nome": "Ana", "telefone": "1111-1111"},  
    {"nome": "Bruno", "telefone": "2222-2222"}  
]
```

```
agenda2 = [  
    {"nome": "Carla", "telefone": "3333-3333"},  
    {"nome": "Bruno", "telefone": "2222-2222"}  
]
```

Contatos em comum: {'Bruno'}



Encontrar contatos em comum entre duas agendas

```
agenda1 = [  
    {"nome": "Ana", "telefone": "1111-1111"},  
    {"nome": "Bruno", "telefone": "2222-2222"}  
]
```

```
agenda2 = [  
    {"nome": "Carla", "telefone": "3333-3333"},  
    {"nome": "Bruno", "telefone": "2222-2222"}  
]
```

```
nomes1 = set()  
for contato in agenda1:  
    nomes1.add(contato["nome"])
```

```
nomes2 = set()  
for contato in agenda2:  
    nomes2.add(contato["nome"])
```

```
comum = nomes1.intersection(nomes2)  
print("Contatos em comum:", comum)
```

Contatos em comum: {'Bruno'}



Lista de dicionários + operações com conjuntos

- **Problema:** Dois alunos cursaram diferentes disciplinas. Disciplinas em comum? Só Ana? Só Bruno?

```
aluno1 = {"nome": "Ana", "disciplinas": ["Matemática", "Física", "Química"]}
aluno2 = {"nome": "Bruno", "disciplinas": ["Biologia", "Física", "Química"]}
```

```
Em comum: {'Física', 'Química'}
Só Ana: {'Matemática'}
Só Bruno: {'Biologia'}
```



Disciplinas em comum? Só Ana? Só Bruno?

```
aluno1 = {"nome": "Ana", "disciplinas": ["Matemática", "Física", "Química"]}
aluno2 = {"nome": "Bruno", "disciplinas": ["Biologia", "Física", "Química"]}

d1 = set()
for d in aluno1["disciplinas"]:
    d1.add(d)

d2 = set()
for d in aluno2["disciplinas"]:
    d2.add(d)

print("Em comum:", d1.intersection(d2))
print("Só Ana:", d1.difference(d2))
print("Só Bruno:", d2.difference(d1))
```

```
Em comum: {'Física', 'Química'}
Só Ana: {'Matemática'}
Só Bruno: {'Biologia'}
```



Contato

Alessandro Botelho Bovo
alessandrobovo@utfpr.edu.br

