Listas, Tuplas e Dicionários

Listas

As listas representam colecções ordenadas e mutáveis. Em GIS, uma lista pode representar uma geometria, um conjunto de coordenadas, ou atributos simples.

```
lista = [1, 2, 3, 4, 5, 6, 8]
```

```
lista
```

```
[1, 2, 3, 4, 5, 6, 8]
```

```
type(lista)
```

```
list
```

```
lista
```

```
[1, 2, 3, 4, 5, 6, 8]
```

```
lista[0]
```

```
1
```

```
lista[3]
```

```
4
```

```
cidades = ['Maputo', 'Matola', 'Xai-Xai']
```

```
cidades
```

```
['Maputo', 'Matola', 'Xai-Xai']
```

```
cidades[2]
```

```
'Xai-Xai'
```

```
# Lista representando coordenadas de um ponto
coord = [-25.9653, 32.5832]   # [lat, lon]

# Lista de vários pontos
pontos = [
    [-25.96, 32.58],
    [-25.97, 32.59],
    [-25.95, 32.57]
]
```

```
coord
```

```
[-25.9653, 32.5832]
```

```
pontos
```

```
[[-25.96, 32.58], [-25.97, 32.59], [-25.95, 32.57]]
```

```
pontos[2][1] + 100
```

```
132.57
```

Operações básicas

```
coord
```

```
[-25.9653, 32.5832]
```

```
print(coord[1])
```

```
32.5832
```

```
coord[1] = 32.60
```

```
print(coord[1])
```

```
32.6
```

```
print(coord[0])   # latitude
coord[1] = 32.60  # modificar longitude
```

```
-25.9653
```

Tuplas

Tuplas são imutáveis. Muito úteis quando queremos garantir que os dados não mudam, por exemplo, coordenadas fixas.

```
ponto_fixo = (-25.9653, 32.5832)
```

```
type(ponto_fixo)
```

```
tuple
```

```
ponto_fixo[0] = -25.001
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-3942670190.py in <cell line: 0>()
----> 1 ponto_fixo[0] = -25.001

TypeError: 'tuple' object does not support item assignment
```

Next steps: Explain error

Dicionários

Representam dados estruturados em pares **chave: valor**. Em GIS, funcionam muito bem para representar atributos.

```
ponto = {
    "nome": "Ponto A",
    "lat": -25.9653,
    "lon": 32.5832,
    "categoria": "sensor"
}
```

```
print(ponto)
```

```
{'nome': 'Ponto A', 'lat': -25.9653, 'lon': 32.5832, 'categoria': 'sensor'}
```

Operações Básicas com Listas, Tuplas e Dicionários

Listas (mutáveis)

Coleções ordenadas que permitem alterar, adicionar e remover elementos. Muito usadas em GIS para representar coordenadas, séries de pontos, atributos simples.

```
# Criar
coords = [-25.96, 32.58]   # [lat, lon]
pontos = [10, 20, 30]
```

```
# Acessar elementos
coords[0]      # latitude
coords[1]      # longitude
```

```
32.58
```

```
# Modificar
coords[1] = 32.60
```

```
coords
```

```
[-25.96, 32.6]
```

```
coords.append(100)
```

```
coords
```

```
[-25.96, 32.6, 100]
```

```
coords.append(7)
```

```
coords
```

```
[-25.96, 32.6, 100, 7]
```

```
coords.insert(0, "A")
```

```
coords
```

```
['A', -25.96, 32.6, 100, 7]
```

```
coords[0]
```

```
'A'
```

```
# Adicionar
coords.append(100)     # altitude
coords.insert(0, "A")  # adiciona no início
```

```
coords
```

```
['A', -25.96, 32.6, 100, 7]
```

```
coords.remove(100)
```

```
coords
```

```
['A', -25.96, 32.6, 7]
```

```
coords.pop()
```

```
7
```

```
coords
```

```
['A', -25.96, 32.6]
```

```
coords.pop(1)
```

```
-25.96
```

```
coords
```

```
['A', 32.6]
```

```
# Remover
coords.remove(100)   # remove pelo valor
coords.pop()         # remove o último
coords.pop(0)        # remove pelo índice
```

```
'A'
```

```
# Tamanho
len(coords)
```

```
2
```

```
len("Ola Mundo        ")
```

```
17
```

Tuplas (imutáveis)

Estruturas ordenadas e que não podem ser alteradas. Úteis para dados fixos como coordenadas estáveis.

```
# Criar
ponto_fixo = (-25.96, 32.58)
```

```
# Acessar
ponto_fixo[0]
```

```
-25.96
```

```
# Descompactar
lat, lon = ponto_fixo
```

```
ponto_fixo
```

```
(-25.96, 32.58)
```

```
lat
```

```
-25.96
```

```
lon
```

```
32.58
```

```
# Tamanho
len(ponto_fixo)
```

```
2
```

Dicionários (chave → valor)

Estruturas ideais para representar atributos GIS, semelhantes a uma linha de tabela.

```
ponto = {
    "nome": "Ponto A",
```

```
    "lat": -25.96,
    "lon": 32.58,
    "categoria": "sensor",
    "altura": 120
}
```

```
ponto['altura']
```

```
120
```

```
ponto.get("altura")
```

```
120
```

```
# Acessar valores
ponto["lat"]
ponto.get("categoria")
```

```
'sensor'
```

```
# Modificar
ponto["lat"] = -25.97
```

```
ponto
```

```
{'nome': 'Ponto A',
 'lat': -25.97,
 'lon': 32.58,
 'categoria': 'sensor',
 'altura': 120}
```

```
# Adicionar nova
ponto["elevacao"] = 115
```

```
ponto
```

```
{'nome': 'Ponto A',
 'lat': -25.97,
 'lon': 32.58,
 'categoria': 'sensor',
 'altura': 120,
 'elevacao': 115}
```

```
# Remover
ponto.pop("categoria")
```

```
'sensor'
```

```
ponto
```

```
{'nome': 'Ponto A',
 'lat': -25.97,
 'lon': 32.58,
 'altura': 120,
 'elevacao': 115}
```

```
ponto.keys()
```

```
dict_keys(['nome', 'lat', 'lon', 'altura', 'elevacao'])
```

```
ponto.values()
```

```
dict_values(['Ponto A', -25.97, 32.58, 120, 115])
```

```
ponto.items()
```

```
dict_items([('nome', 'Ponto A'), ('lat', -25.97), ('lon', 32.58), ('altura', 120), ('elevacao', 115)])
```

```
type(ponto.keys())
```

```
dict_keys
```

```
type(ponto.items())
```

```
dict_items
```

```
type(ponto.values())
```

```
dict_values
```

```
# Listar chaves e valores
# ponto.keys()
# ponto.values()
# ponto.items()
```

```
dict_items([('nome', 'Ponto A'), ('lat', -25.97), ('lon', 32.58), ('elevacao', 115)])
```

```
# Tamanho
len(ponto)
```

```
5
```

Estrutura	Mutável?	Ordenada?	Quando usar em GIS
Lista	✔ Sim	✔ Sim	Conjuntos de coordenadas, séries de pontos
Tupla	✘ Não	✔ Sim	Coordenadas fixas e imutáveis
Dicionário	✔ Sim	✘ Não (em princípio)	Atributos de feições, propriedades de pontos

```
lista
```

```
[1, 2, 3, 4, 5, 6, 8]
```

```
lista[0]= 100
```

```
lista
```

```
[100, 2, 3, 4, 5, 6, 8]
```

```
lista2
```

```
['a', 'b', 'c']
```

```
lista2 = ['a', 'b', 'c']
```

```
lista2
```

```
['a', 'b', 'c']
```

```
pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0
```

```
!pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0
```

TPC 2

Tens um ficheiro Excel contendo informação sobre várias escolas, incluindo localização, infraestrutura e condições de saneamento.

O teu objectivo é realizar uma exploração inicial dos dados usando Pandas, aplicando operações fundamentais de leitura, seleção e estatísticas básicas.

1. **Importação dos dados**

Carrega o ficheiro Excel usando Pandas e mostra as primeiras 5 linhas para confirmar que foi carregado corretamente.

2. **Exploração geral**

Mostra:

- o número de linhas e colunas,
- os nomes das colunas,
- o tipo de dados (`dtypes`) de cada coluna.

3. **Estatísticas básicas**

Usando funções do `Pandas`:

1. Quantas escolas têm latrina
2. Calcula a longitude mínima e máxima.
3. Mostra a frequência de cada província.
4. Conta quantas escolas têm água disponível (Agua == "Sim")
5. Quantas escolas tem um distrito a sua escolha?
6. Quantas escolas com carteira tem um distrito a sua escolha?

Dica

1. Usa `.value_counts()` para contagens rápidas.
2. Usa `.mean()`, `.min()`, `.max()` para estatísticas básicas.

DEADLINE: ***Sexta-feira, 18horas***