```python
import pandas as pd
```

Exercícios 01

🎯 Parte 1 — Series e DataFrames

1.

Crie uma Series com os valores [100, 200, 300, 400, 500] e índices nomeados como: ['A', 'B', 'C', 'D', 'E'].

Acesse o valor do índice 'C'.

Verifique se o valor 400 está na Series.

```python
df = pd.DataFrame(index=['A', 'B', 'C', 'D', 'E'], data={'valores': [100, 200, 300, 400, 500]})
```

```python
df
```

border="1" class="dataframe">

	valores
A	100
B	200
C	300
D	400
E	500

```python
df.loc['C'] # Acessando o valor do índice 'C'
```

```
valores    300
Name: C, dtype: int64
```

```python
# df.iloc[0:4] == 400

# Verificando se o valor '400' está presente na Series 'valores'
df[df['valores'] == 400]
```

border="1" class="dataframe">

	valores
D	400

2.

Crie um dataFrame com os dados abaixo:

>

Produto		Preço	Quantidade
Arroz	20	2	
Feijão	8	4	
Macarrão	5	3	
Leite	4	6	

- Calcule a coluna `"Total"` (Preço * Quantidade).
- Exiba apenas a coluna Produto usando `.iloc[]`.
- Exiba as 2 primeiras linhas usando `.iloc[]`.

```
dados2 = {
    'produto':['Arroz', 'Feijão', 'Macarrão', 'Leite'],
    'preco':[20, 8, 5, 4],
    'quantidade':[2, 4, 3, 6]
}

df2 = pd.DataFrame(dados2)
```

```
df2
```

border="1" class="dataframe">

	produto	preco	quantidade
0	Arroz	20	2
1	Feijão	8	4
2	Macarrão	5	3
3	Leite	4	6

```
# Criando a coluna 'Total' no dataframe e calculando
df2['Total'] = df2['preco'] * df2['quantidade']
```

```
df2
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
0	Arroz	20	2	40
1	Feijão	8	4	32
2	Macarrão	5	3	15
3	Leite	4	6	24

```
# Exiba apenas a coluna Produto usando .iloc[].
df2.iloc[:, 0]
```

```
0       Arroz
1      Feijão
2    Macarrão
3       Leite
Name: produto, dtype: object
```

```
# Exiba as 2 primeiras linhas usando .iloc[].
df2.iloc[0:2]
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
0	Arroz	20	2	40
1	Feijão	8	4	32

🎯 Parte 2 — Filtros e Indexação

3.

Filtre o DataFrame acima para:

- Mostrar os produtos com Preço maior que 5.
- Mostrar os produtos com Quantidade menor ou igual a 3.
- Mostrar produtos cujo nome seja 'Feijão' ou 'Leite'.

```
df2 # Será o DataFrame utilizado nesse exercicio
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
0	Arroz	20	2	40
1	Feijão	8	4	32
2	Macarrão	5	3	15
3	Leite	4	6	24

```
# Mostrar os produtos com Preço maior que 5.
df2[df2['preco'] > 5]
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
0	Arroz	20	2	40
1	Feijão	8	4	32

```
# Mostrar os produtos com Quantidade menor ou igual a 3.
df2[df2['quantidade'] <= 3]
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
0	Arroz	20	2	40
2	Macarrão	5	3	15

```
# Mostrar produtos cujo nome seja 'Feijão' ou 'Leite'.
df2[(df2['produto'] == 'Feijão') | (df2['produto'] == 'Leite')]
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
1	Feijão	8	4	32
3	Leite	4	6	24

```
# Mostrar produtos cujo nome seja 'Feijão' ou 'Leite'.
# Mostrando de outra maneira para ser resolvido:
df2[df2['produto'].isin(['Feijão', 'Leite'])]
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
1	Feijão	8	4	32
3	Leite	4	6	24

4

Defina a coluna 'Produto' como índice do DataFrame.

- Filtre apenas os dados do produto 'Arroz'.

```
df2
```

border="1" class="dataframe">

	produto	preco	quantidade	Total
0	Arroz	20	2	40
1	Feijão	8	4	32
2	Macarrão	5	3	15
3	Leite	4	6	24

```
df2.set_index('produto', inplace=True)
```

```
df2
```

border="1" class="dataframe">

produto	preco	quantidade	Total
Arroz	20	2	40
Feijão	8	4	32
Macarrão	5	3	15
Leite	4	6	24

```
# Filtre apenas os dados do produto 'Arroz'.
df2.iloc[0]
```

```
preco         20
quantidade     2
Total         40
Name: Arroz, dtype: int64
```

🎯 Parte 3 — Dados Ausentes

5.

Crie um novo DataFrame com dados faltantes:

>

Nome	Idade	Cidade
Ana	25	São Paulo
Bruno	NaN	Rio de Janeiro
Carla	30	NaN

- Verifique onde existem dados ausentes.

- Preencha as idades ausentes com a média das idades.
- Preencha as cidades ausentes com `"Não informado"`.

```
import numpy as np

dados3 = {
    'Nome':['Ana', 'Bruno', 'Carla'],
    'Idade':[25, np.nan, 30],
    'Cidade':['São Paulo', 'Rio de Janeiro', np.nan]
}

df3 = pd.DataFrame(dados3)
```

```
df3
```

border="1" class="dataframe">

	Nome	Idade	Cidade
0	Ana	25.0	São Paulo
1	Bruno	NaN	Rio de Janeiro
2	Carla	30.0	NaN

```
# Verifique onde existem dados ausentes.
df3.isnull() # -> Vai mostrar True onde estiver ausente!
```

border="1" class="dataframe">

	Nome	Idade	Cidade
0	False	False	False
1	False	True	False
2	False	False	True

```
# Preencha as idades ausentes com a média das idades.
df3['Idade'] = df3['Idade'].fillna(df3['Idade'].mean())

# Preencha as cidades ausentes com "Não informado".
df3['Cidade'] = df3['Cidade'].fillna('Não Informado')
```

```
df3
```

border="1" class="dataframe">

	Nome	Idade	Cidade
0	Ana	25.0	São Paulo
1	Bruno	27.5	Rio de Janeiro
2	Carla	30.0	Não Informado

🎯 Parte 4 — Operações com DataFrames

6.

No DataFrame dos produtos:

- Aplique um desconto de 10% no preço e salve em uma nova coluna 'Preço_com_Desconto'.
- Calcule o preço médio dos produtos.
- Crie uma coluna booleana 'Caro' que indique se o produto custa mais que R$10.

```
df2 # Será utilizado esse DataFrame para esse exercício
```

border="1" class="dataframe">

produto	preco	quantidade	Total
Arroz	20	2	40
Feijão	8	4	32
Macarrão	5	3	15
Leite	4	6	24

```
# Aplique um desconto de 10% no preço e salve em uma nova coluna
'Preço_com_Desconto'.
desconto = df2['preco'] / 10 # Para calcular 10% de desconto em um produto, você
pode dividir o preço original do produto por 10.
df2['Preco_Com_Desconto'] = df2['preco'] - desconto
```

```
df2
```

border="1" class="dataframe">

produto	preco	quantidade	Total	Preco_Com_Desconto
Arroz	20	2	40	18.0
Feijão	8	4	32	7.2
Macarrão	5	3	15	4.5
Leite	4	6	24	3.6

```
# Calcule o preço médio dos produtos.
df2['preco'].mean() # Resultado: 9.25
```

```
np.float64(9.25)
```

```
# Crie uma coluna booleana 'Caro' que indique se o produto custa mais que R$10
df2['Caro'] = [True if x > 10 else False for x in df2['preco'] ]
```

```
df2
```

border="1" class="dataframe">

produto	preco	quantidade	Total	Preco_Com_Desconto	Caro
Arroz	20	2	40	18.0	True
Feijão	8	4	32	7.2	False
Macarrão	5	3	15	4.5	False
Leite	4	6	24	3.6	False

🎯 Parte 5 — Agrupamento com GroupBy

7.

Crie um novo dataFrame:

>

Categoria	Produto	Vendas
Alimento	Arroz	200
Alimento	Feijão	150
Bebida	Suco	180
Bebida	Refrigerante	120

- Agrupe por 'Categoria' e calcule a soma de 'Vendas'.
- Mostre a quantidade de produtos em cada categoria.

```
dados4 = {
    'Categoria': ['Alimento', 'Alimento', 'Bebida', 'Bebida'],
    'Produto': ['Arroz', 'Feijão', 'Suco', 'Refrigerante'],
    'Vendas': [200, 150, 180, 120]
}

df4 = pd.DataFrame(dados4)
```

```
df4
```

border="1" class="dataframe">

	Categoria	Produto	Vendas
0	Alimento	Arroz	200
1	Alimento	Feijão	150
2	Bebida	Suco	180
3	Bebida	Refrigerante	120

```
# Agrupe por 'Categoria' e calcule a soma de 'Vendas'.
df4.groupby('Categoria')['Vendas'].sum()
```

```
Categoria
Alimento    350
Bebida      300
Name: Vendas, dtype: int64
```

```
# Mostre a quantidade de produtos em cada categoria.
df4.groupby('Categoria')['Produto'].count()
```

```
Categoria
Alimento    2
Bebida      2
Name: Produto, dtype: int64
```

🎯 Parte 6 — Merge, Join e Concat

8.

Crie dois dataFrames:

```
clientes = pd.DataFrame({
    'ID': [1, 2, 3],
    'Nome': ['Ana', 'Bruno', 'Carla']
})

compras = pd.DataFrame({
    'ID': [2, 3, 4],
    'Produto': ['Arroz', 'Feijão', 'Leite']
})

df_clientes = pd.DataFrame(clientes)
df_compras = pd.DataFrame(compras)
```

```
df_clientes
```

border="1" class="dataframe">

	ID	Nome
0	1	Ana
1	2	Bruno
2	3	Carla

```
df_compras
```

border="1" class="dataframe">

	ID	Produto
0	2	Arroz
1	3	Feijão
2	4	Leite

- Faça um merge usando o ID (inner join).
- Faça um merge do tipo outer.
- Faça um concat vertical desses dois DataFrames (ignorar o ID).

```
# Faça um merge usando o ID (igual ao inner join do SQL).
pd.merge(df_clientes, df_compras, on='ID')
```

border="1" class="dataframe">

	ID	Nome	Produto
0	2	Bruno	Arroz
1	3	Carla	Feijão

```
# Faça um merge do tipo outer.
pd.merge(df_clientes, df_compras, how="outer")
```

border="1" class="dataframe">

	ID	Nome	Produto
0	1	Ana	NaN
1	2	Bruno	Arroz
2	3	Carla	Feijão
3	4	NaN	Leite

```
# Faça um concat vertical desses dois DataFrames (ignorar o ID).
pd.concat([df_clientes, df_compras])
```

border="1" class="dataframe">

	ID	Nome	Produto
0	1	Ana	NaN
1	2	Bruno	NaN
2	3	Carla	NaN
0	2	NaN	Arroz
1	3	NaN	Feijão
2	4	NaN	Leite

Parte 7 — Séries Temporais

9.

Crie uma sequência de datas diárias de '2024-01-01' a '2024-01-10':

- Crie um DataFrame com essa data e uma coluna aleatória chamada 'Vendas'.
- Defina a coluna de datas como índice.
- Filtre as vendas dos dias '2024-01-03' até '2024-01-07'.
- Reamostre os dados para frequência semanal somando as vendas.

```
# Crie uma sequência de datas diárias de '2024-01-01' a '2024-01-10':
datas = pd.date_range(start='2024-01-01', periods=10, freq='D')

# Crie um DataFrame com essa data e uma coluna aleatória chamada 'Vendas'.
vendas = [125, 136, 80, 92, 101, 101, 242, 96, 20, 120]

df5 = pd.DataFrame({'Data': datas, 'Vendas': vendas})
```

```
df5
```

border="1" class="dataframe">

	Data	Vendas
0	2024-01-01	125
1	2024-01-02	136
2	2024-01-03	80
3	2024-01-04	92
4	2024-01-05	101
5	2024-01-06	101
6	2024-01-07	242
7	2024-01-08	96
8	2024-01-09	20
9	2024-01-10	120

```
df5.set_index('Data', inplace=True)
```

```
df5.index
```

```
DatetimeIndex(['2024-01-01', '2024-01-02', '2024-01-03', '2024-01-04',
               '2024-01-05', '2024-01-06', '2024-01-07', '2024-01-08',
               '2024-01-09', '2024-01-10'],
              dtype='datetime64[ns]', name='Data', freq=None)
```

```
df5
```

border="1" class="dataframe">

	Vendas
Data	
2024-01-01	125
2024-01-02	136
2024-01-03	80
2024-01-04	92
2024-01-05	101
2024-01-06	101
2024-01-07	242
2024-01-08	96
2024-01-09	20
2024-01-10	120

```
# Filtre as vendas dos dias '2024-01-03' até '2024-01-07'.
df5.loc['2024-01-03':'2024-01-07']
```

border="1" class="dataframe">

Data	Vendas
2024-01-03	80
2024-01-04	92
2024-01-05	101
2024-01-06	101
2024-01-07	242

```
# Reamostre os dados para frequência semanal somando as vendas.
df5.resample('W').sum()
```

border="1" class="dataframe">

Data	Vendas
2024-01-07	877
2024-01-14	236

Parte 8 — Entrada e Saída de Dados

10.

(Opcional) Salve qualquer DataFrame criado em:

- Um arquivo CSV (.to_csv()).
- Um arquivo Excel (.to_excel()).

Depois tente ler novamente o arquivo salvo.