

Exercício

- Dada a string:
 - linha = "H2O; CO2; CH4; NH3; O2"
- a) Transforme em uma lista de 5 elementos.
- b) Em seguida, una os 5 elementos com o separador " | ".
- c) Escreva o código que, partindo da string original, percorre a lista e imprime cada molécula, uma por linha.

Exercício

```
linha = "H2O; CO2; CH4; NH3; O2"
```

a) split

```
lista = linha.split("; ")
```

```
print(lista)
```

b) join

```
resultado = " | ".join(lista)
```

```
print(resultado)
```

c) for com upper

```
for molecula in lista:
```

```
    print(molecula.upper())
```

PRÁTICAS EM CIÊNCIA DE DADOS / AULA

ppandas.

A biblioteca que transforma **arquivos confusos** em **perguntas respondidas**.

O que é pandas?

Em uma frase: pandas é o canivete suíço para dados tabulares em Python.

2008

ANO DE CRIAÇÃO

Criada por Wes McKinney em um fundo de hedge — para resolver problemas reais de análise financeira.

panel data

ORIGEM DO NOME

Vem de "panel data" (dados em painel), termo usado em econometria. Não tem nada a ver com o animal.

#1

EM PYTHON PARA DADOS

Padrão de facto da comunidade. Se você manipula dados em Python, vai cruzar com pandas mais cedo ou mais tarde.

Anatomia de um DataFrame

A estrutura central do pandas é como uma planilha — mas turbinada.

C O L U N A S
cada uma é uma Series

	país	continente	população
0	Brasil	América	212.688.125
1	Alemanha	Europa	83.190.556
2	China	Ásia	1.412.600.000
3	Japão	Ásia	125.300.000

Í N D I C E
identifica cada linha

INSPECIONANDO O OBJETO

df.shape	(4, 3) — linhas × colunas
df.columns	Index(['país', 'continente', 'população'])
df.dtypes	object, object, int64
df.index	RangeIndex(start=0, stop=4)
df.head(2)	primeiras 2 linhas
df.info()	resumo completo: tipos + memória

Lendo dados de qualquer lugar

pandas fala todos os dialetos. Você passa o arquivo, ele entrega um DataFrame.

.xlsx

```
pd.read_excel("file.xlsx")
```

Planilhas do Excel. Aceita múltiplas abas com `sheet_name=`.

.csv

```
pd.read_csv("file.csv")
```

Valores separados por vírgula. O formato mais universal — texto puro, versionável.

.json

```
pd.read_json("file.json")
```

Comum em APIs web. Para JSON aninhado, `json_normalize` é seu amigo.

SQL

```
pd.read_sql(query, conn)
```

Direto do banco de dados. Funciona com PostgreSQL, MySQL, SQLite...

As 5 operações que você vai usar 90% do tempo

Domine essas e você já se vira com a maioria dos problemas de análise de dados.

01

Selecionar

Pegar uma coluna ou um subconjunto delas.

```
df["populacao"]
```

02

Filtrar

Manter só as linhas que satisfazem uma condição.

```
df[df["populacao"] > 1e8]
```

03

Ordenar

Colocar as linhas em ordem por uma ou mais colunas.

```
df.sort_values("populacao")
```

04

Agregar

Resumir grupos de linhas em uma estatística.

```
df.groupby("continente").sum()
```

05

Transformar

Criar uma nova coluna a partir das existentes.

```
df["pct"] =  
df["populacao"]/total
```

Filtro: a operação mais útil

Como funciona uma máscara booleana, em duas etapas:

ETAPA 1

Produz uma Series de True/False

```
df["populacao"] > 100_000_000
```

Brasil → True

Alemanha → False

China → True

Japão → True

ETAPA 2

Use a Series como índice

```
df[ df["populacao"] > 100_000_000 ]
```

Sobram só as linhas onde a máscara é True:

Brasil	212.688.125
--------	-------------

China	1.412.600.000
-------	---------------

Japão	125.300.000
-------	-------------

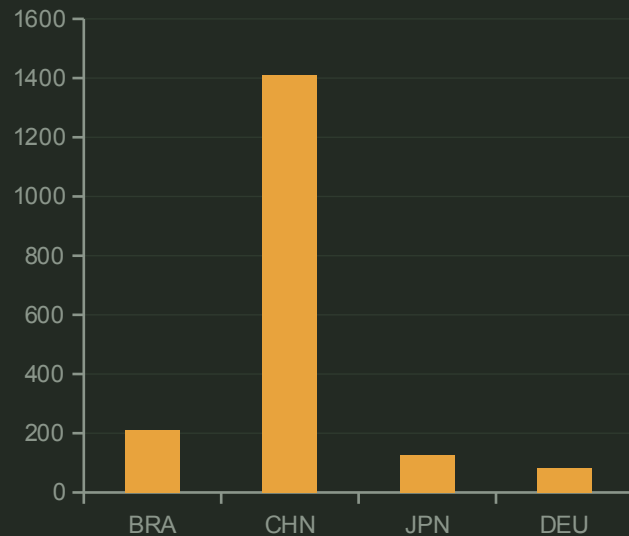
DICA Combine condições com & (e), | (ou), ~ (não). Cada condição entre parênteses.

Visualizando em uma linha de código

O método `.plot()` é uma porta amigável para o `matplotlib` — escolha o tipo no sufixo.

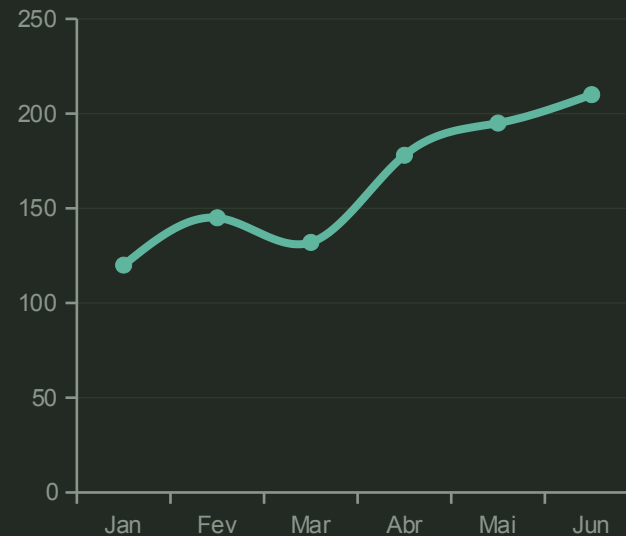
`df.plot.bar()`

Comparar valores entre categorias



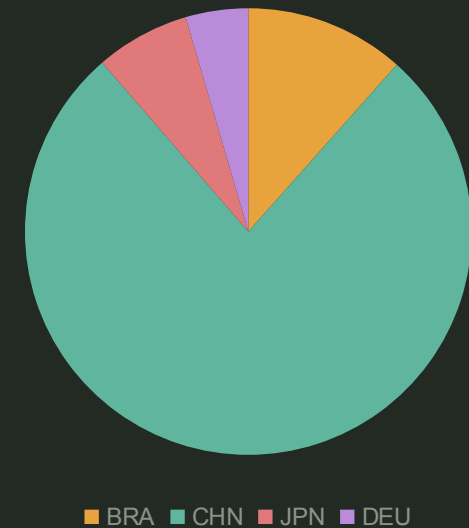
`df.plot.line()`

Mostrar evolução ao longo do tempo



`df.plot.pie()`

Mostrar partes de um todo



E AGORA?

Os próximos passos.

01

Pratique

Pegue um dataset que te interesse — futebol, música, política, qualquer coisa — e faça as 5 operações do slide 5 nele.

Sugestão: [kaggle.com/datasets](https://www.kaggle.com/datasets)

02

Explore

Conheça métodos que ficaram de fora: `groupby`, `merge`, `pivot_table`, `melt`, `str.contains`, `fillna`, `to_datetime`.

Documentação oficial em pandas.pydata.org

03

Aprofunde

Para análises mais sofisticadas: `matplotlib` e `seaborn` para gráficos, `scikit-learn` para machine learning, `polars` como alternativa moderna.

Cada um é um universo — vá pelo que o problema pedir.