

# Introdução a bibliotecas

# Pandas

Dentre as diversas bibliotecas disponíveis no repositório PyPI, pandas é um pacote Python que fornece estruturas de dados projetadas para facilitar o trabalho com dados estruturados (tabelas) e de séries temporais.

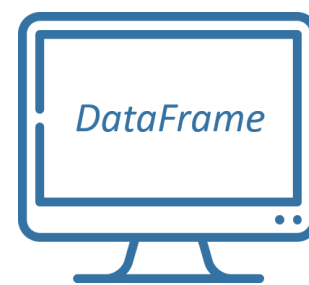
Como uma ferramenta de alto nível, pandas possui duas estruturas de dados que são as principais para a análise/manipulação de dados: a **Series** e o **DataFrame**. Nesta webaula vamos conhecer cada uma delas.



Fonte: Shutterstock.



Uma Series é um como um vetor de dados (unidimensional), capaz de armazenar diferentes tipos de dados.



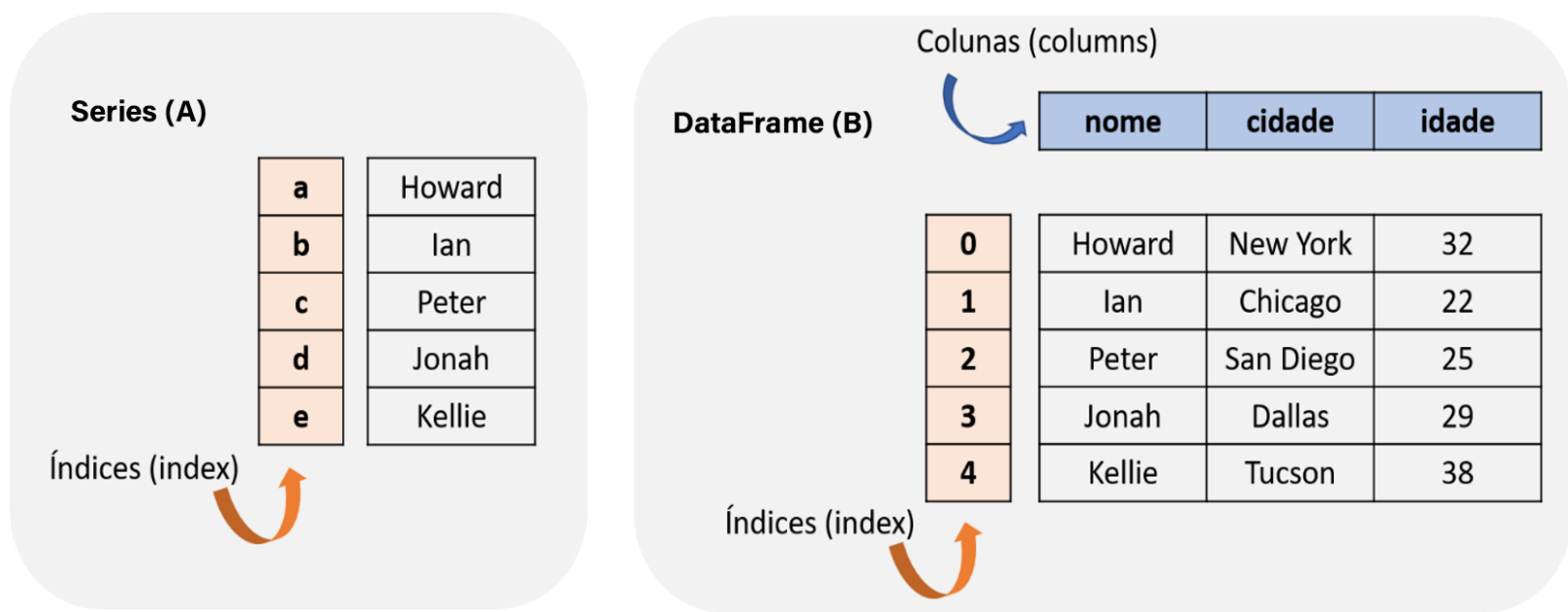
Um DataFrame é um conjunto de Series, ou, como a documentação apresenta, um contêiner para Series.

Ambas as estruturas possuem como grande característica a indexação das linhas, ou seja, cada linha possui um **rótulo** (nome) que a identifica, o qual pode ser uma string, um inteiro, um decimal ou uma data.

A figura a seguir ilustra uma **Series (A)** e um **DataFrame (B)**, veja que:

- » Uma Series possui somente "uma coluna" de informação e seus rótulos (índices).
- » Um DataFrame pode ter uma ou mais colunas e, além dos índices, também há um rótulo de identificação com o nome da coluna.

[Saiba mais](#)



Fonte: elaborada pela autora.

Para utilizar os recursos da biblioteca é preciso importar no projeto. Usa-se a seguinte convenção:

```
import pandas as pd
```

## Series

Para construir um objeto do tipo Series, é preciso utilizar o método `Series()` do pacote pandas. O método possui o seguinte construtor:

```
pandas.Series(data=None, index=None, dtype=None, name=None, copy=False, fastpath=False)
```

Como todos os parâmetros possuem valores padrão (default), o que permite instanciar um objeto de diferentes formas. Veja algumas dessas formas:

```
pd.Series(data=5) # Cria uma Series com o valor 5
pd.Series('Howard Ian Peter Jonah Kellie'.split()) # Cria uma Series com uma lista de nomes
```

## DataFrame

Para construir um objeto do tipo DataFrame é preciso utilizar o método `DataFrame()` do pacote pandas. O método possui o seguinte construtor:

```
pandas.DataFrame(data=None, index=None, columns=None, dtype=None, copy=False)
```

Veja alguns exemplos:

```
# Cria um DataFrame, de uma coluna a partir de uma lista
pd.DataFrame('Howard Ian Peter Jonah Kellie'.split(), columns=['nome'])
```

Resultado:

	nome
--	------

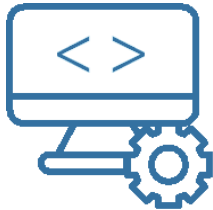
	nome
0	Howard
1	Ian
2	Peter
3	Jonah
4	Kellie

```
lista_nomes = 'Howard Ian Peter Jonah Kellie'.split()
lista_cpfs = '111.111.111-11 222.222.222-22 333.333.333-33 444.444.444-44
555.555.555-55'.split()
lista_emails = 'risus.varius@dictumPhasellusin.ca
Nunc@vulputate.ca fames.ac.turpis@cursusa.org non@felisullamcorper.org
eget.dictum.placerat@necluctus.co.uk'.split()
lista_idades = [32, 22, 25, 29, 38]
dados = list(zip(lista_nomes, lista_cpfs, lista_idades, lista_emails)) # cria
uma lista de tuplas # Cria um DataFrame a partir de uma lista de tuplas
pd.DataFrame(dados, columns=['nome', 'cpfs', 'idade', 'email'])
```

Resultado:

	nome	cpfs	emails	idades
0	Howard	111.111.111-11	risus.varius@dictumPhasellusin.ca	32
1	Ian	222.222.222-22	nunc@vulputate.ca	22
2	Peter	333.333.333-33	fames.ac.turpis@cursusa.org	25
3	Jonah	444.444.444-44	non@felisullamcorper.org	29
4	Kellie	555.555.555-55	eget.dictum.placerat@necluctus.co.uk	38

Saiba mais



Estruturas de dados são utilizadas para armazenar dados e diferentes estruturas possuem diferentes atributos e métodos. Com as estruturas de dados do pandas não é diferente, tais objetos possuem atributos e métodos específicos. Existem atributos e métodos que extraem informações estruturais da Series ou do DataFrame, por exemplo, o atributo *shape* ou *dtypes*. Por outro lado, existem recursos que permitem transformar os dados em informações, por exemplo, as estatísticas, como o método *mean()* ou *median()*.

Pesquise mais

Na documentação oficial da biblioteca pandas você encontrará o método *read\_html()*, que permite fazer a leitura de tabelas em páginas web.



pandas Team. **pandas.read\_html**. Disponível em:  
[https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read\\_html.html](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_html.html).  
Acesso em: 17 jun. 2020.



Fonte: Shutterstock.