Introdução ao pandas

Live de Python # 221

Roteiro

1. Antes do Pandas

Estruturas fundamentais

2. O pandas

Por isso viemos aqui

3. Series

O objeto construtor de tudo

4. DataFrames

Onde tudo fica bonito

picpay.me/dunossauro

apoia.se/livedepython

pix.dunossauro@gmail.com

Ajude o projeto <3

Ademar Peixoto, Adilson Herculano, Adriana Cavalcanti, Alexandre Harano, Alexandre Lima, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Andre Azevedo, André Rafael, Aquiles Coutinho, Arnaldo Turque, Aurelio Costa, Bruno Batista, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, Carina Pereira, Christiano Morais, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, David Couto, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Diogo Paschoal, Douglas Bastos, Douglas Braga, Douglas Zickuhr, Dutofanim Dutofanim, Eliel Lima, Elton Silva, Emerson Rafael, Eneas Teles, Erick Ritir, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Everton Silva, Fabiano Tomita, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Fabricio Araujo, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Flávio Meira, Flavkaze Flavkaze, Gabriel Barbosa, Gabriel Mizuno, Gabriel Nascimento, Gabriel Simonetto, Geandreson Costa, Guilherme Cabrera, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Gustavo Dettenborn, Gustavo Pereira, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Hugo Cosme, Igor Taconi, Israel Gomes, Italo Silva, Jair Andrade, Jairo Jesus, Jairo Lenfers, Janael Pinheiro, João Paulo, João Rodrigues, Joelson Sartori, Johnny Tardin, Jonatas Leon, Jônatas Silva, José Gomes, Joseíto Júnior, Jose Mazolini, José Pedro, Juan Gutierrez, Juliana Machado, Júlio Gazeta, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Miranda, Leonardo Mello, Leonardo Nazareth, Luancomputacao Roger, Lucas Adorno, Lucas Mello, Lucas Mendes, Lucas Nascimento, Lucas Oliveira, Lucas Simon, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Luiz Perciliano, Maicon Pantoja, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marcio Moises, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mateus Lisboa, Matheus Cortezi, Matheus Silva, Matheus Vian, Mauricio Nunes, Mrreinadogoes Mrreinadogoes, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nathan Branco, Nicolas Teodosio, Osvaldo Neto, Patricia Minamizawa, Patrick Felipe, Paulo Braga, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, P Muniz, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Regis Tomkiel, Renato Veirich, Ricardo Silva, Riverfount Riverfount, Robson Maciel, Rodrigo Alves, Rodrigo Cardoso, Rodrigo Freire, Rodrigo Oliveira, Rodrigo Quiles, Rodrigo Vaccari, Rodrigo Vieira, Rogério Nogueira, Rogério Sousa, Ronaldo Silva, Ronaldo Silveira, Rui Jr, Samanta Cicilia, Thalles Rosa, Thiago Araujo, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago Oliveira, Thiago Salgado, Thiago Souza, Tiago Minuzzi, Tony Dias, Valcilon Silva, Valdir Tegon, Victor Wildner, Vinícius Bastos, Vinicius Stein, Vitor Luz, Vladimir Lemos, Walter Reis, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Wilson Rocha, Xico Silvério, Yury Barros

Obrigado você

Antes do pandas

Uma visão geral sobre estruturas

Estruturas embutidas

Para começar nossa viagem do dia, gostaria de apresentar as listas do Python.

- Estrutura nativa
- Capacidade de armazenar uma quantidade incontável de objetos
- Mutável
- Armazena qualquer tipo de objetos

```
l = [1, 2, 3, 4, 5, 6, 7]
```

Listas

Elas pode ser acessadas por index, suportam fatiamentos e podem adicionar ou remover novos valores de dentro delas.

```
l = [1, 2, 3, 4, 5, 6, 7]
l[0]          # 1
l[0:3]        # [1, 2, 3]

l.append(8)   # [1, 2, 3, 4, 5, 6, 7, 8]
l.pop(0)      # [2, 3, 4, 5, 6, 7, 8]
```

Listas

- Contêm dados heterogêneos
- Consomem uma quantidade razoável de memória dependendo do problema

```
l = [[1, 2], 'a', (3, 2, 3), 7]

l[0]     # [1, 2]
l[0][0]  # 1
```

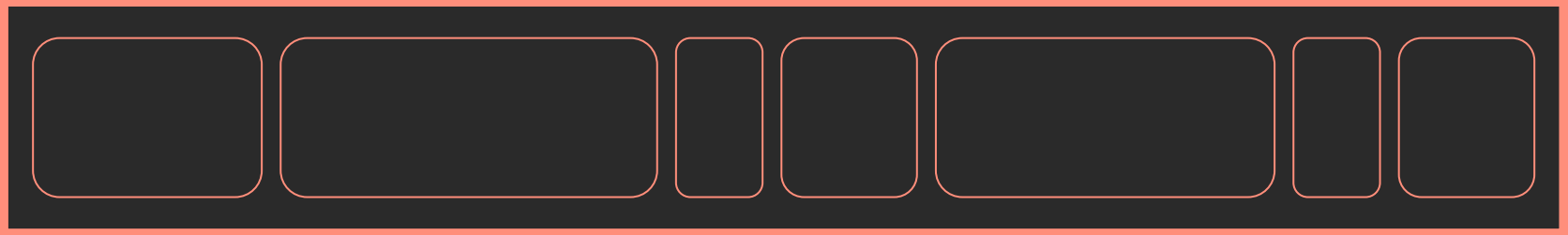
Arrays

Para trabalhar com dados homogêneos e economizar memória, temos o objeto **array**.

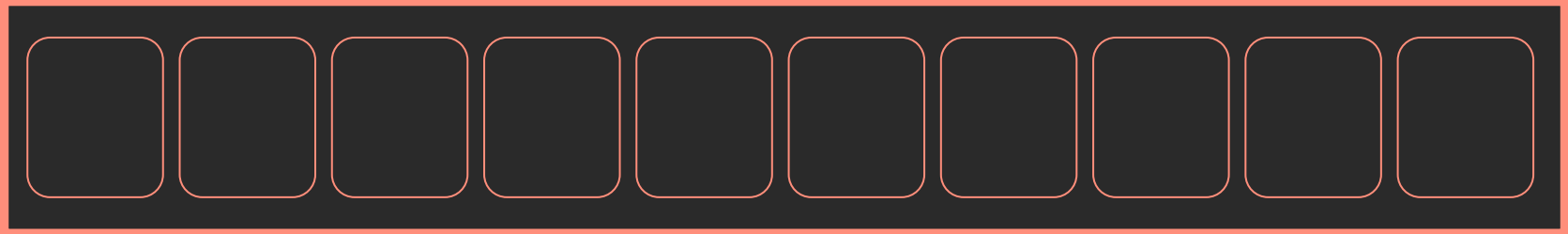
```
from array import array

a = array('i', [1, 2, 3, 4, 5, 6])

a.typecode  # 'i'
a.itemsize  #  4 bytes
```

Isso quer dizer Q?

Lista



Array



Diferenças de tamanho

```python
from array import array
from sys import getsizeof

l = list(range(1_000_000))
a = array('i', l)

getsizeof(l) / 1024 / 1024  # 7.629447937011719
getsizeof(a) / 1024 / 1024  # 3.8147735595703125
```

Afinal, que raios é um array?

Array é um termo em inglês que se refere ao que chamamos em português de vetor ou matriz.

```
# Array com 1 dimensão
vetor = [1, 2, 3, 4, 5]

# Array com 3 dimesões
matriz = [
  [1, 2, 3, 4, 5],
  [6, 7, 8, 9, 0],
  [1, 2, 3, 4, 5]
]
```

Quando estamos falando de matrizes e vetores estamos falando sobre o quê?

PERGUNTA!

Quando estamos falando de matrizes e vetores estamos falando sobre o quê?

MATEMÁTICA!

PERGUNTA!

Porém, contudo, entretanto, todavia

Operações algébricas não são suportadas na biblioteca padrão

```
l = [1, 2, 3]
l * 2  # [1, 2, 3, 1, 2, 3]
l / 2  # TypeError: unsupported operand type(s)

a = array('i', l)
a * 2  # array('i', [1, 2, 3, 1, 2, 3])
a / 2  # TypeError: unsupported operand type(s)
```

Numeric Python - NumPy

Para isso usamos o **Numpy**, uma biblioteca externa para trabalhar com arrays numéricos.

```python
from numpy import array

a = array([1, 2, 3])
a.dtype  # dtype('int64') - Homogêneo

# Algébrico
a * 2  # array([2, 4, 6])
a / 3  # array([0.33333333, 0.66666667, 1.])
a * 2 + 1 /2  # array([2.5, 4.5, 6.5])
```

pip install numpy

Bora instalar?

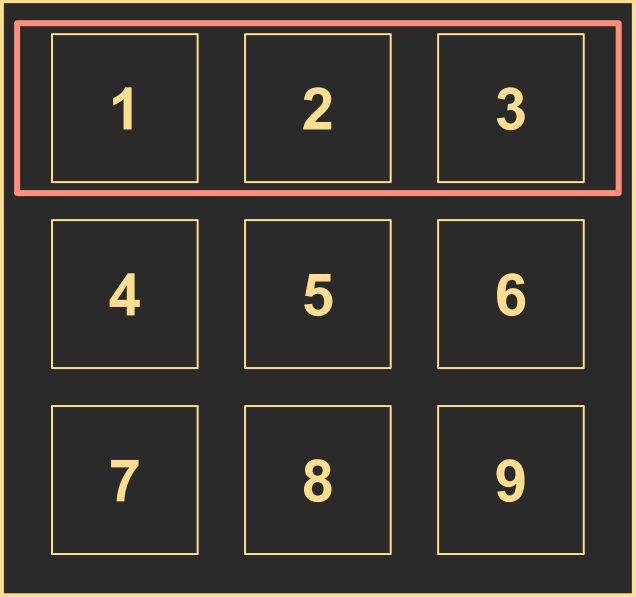
Indexação e matrizes

```python
from numpy import array

a = array([
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9],
])

# Indexável
a[0]    # array([1, 2, 3])
a[:,0]  # array([1, 4, 7])
```

1	2	3
4	5	6
7	8	9

Indexação e matrizes

```python
from numpy import array

a = array([
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9],
])

# Indexável
a[0]    # array([1, 2, 3])
a[:,0]  # array([1, 4, 7])
```

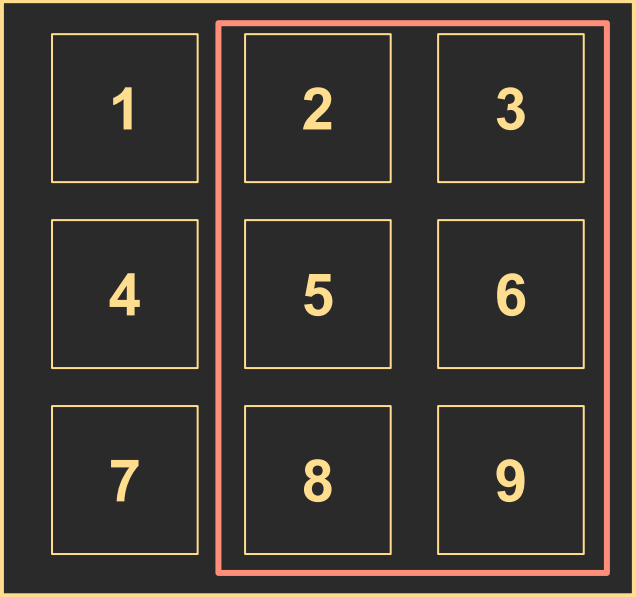


Indexação e matrizes

```python
from numpy import array

a = array([
  [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9],
])

# Indexável
a[0]    # array([1, 2, 3])
a[:,0]  # array([1, 4, 7])
```

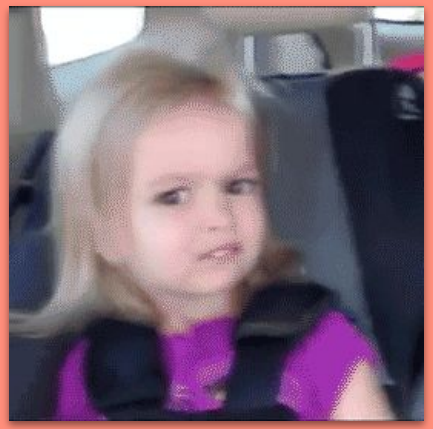
Indexação e matrizes

```
a[:,1:2]

array([[2, 3],
       [5, 6],
       [8, 9]])
```



```
a[:,0]
a[:,1:2]
a[2:,1:3]
```

???????



Ufa, até que enfim!

Pandas

Pandas

- Pandas é uma biblioteca que nasce em 2008, se torna de código aberto em 2009.
- É a primeira grande biblioteca do Python a inserir o conceito de DataFrame.
- Escrita em Python, Cython e C.
- Tem como base o Numpy
  - Expande os tipos e operações dos arrays
- Tem integração com os formatos de tabelas de mercado
  - csv
  - excel
  - bancos relacionais

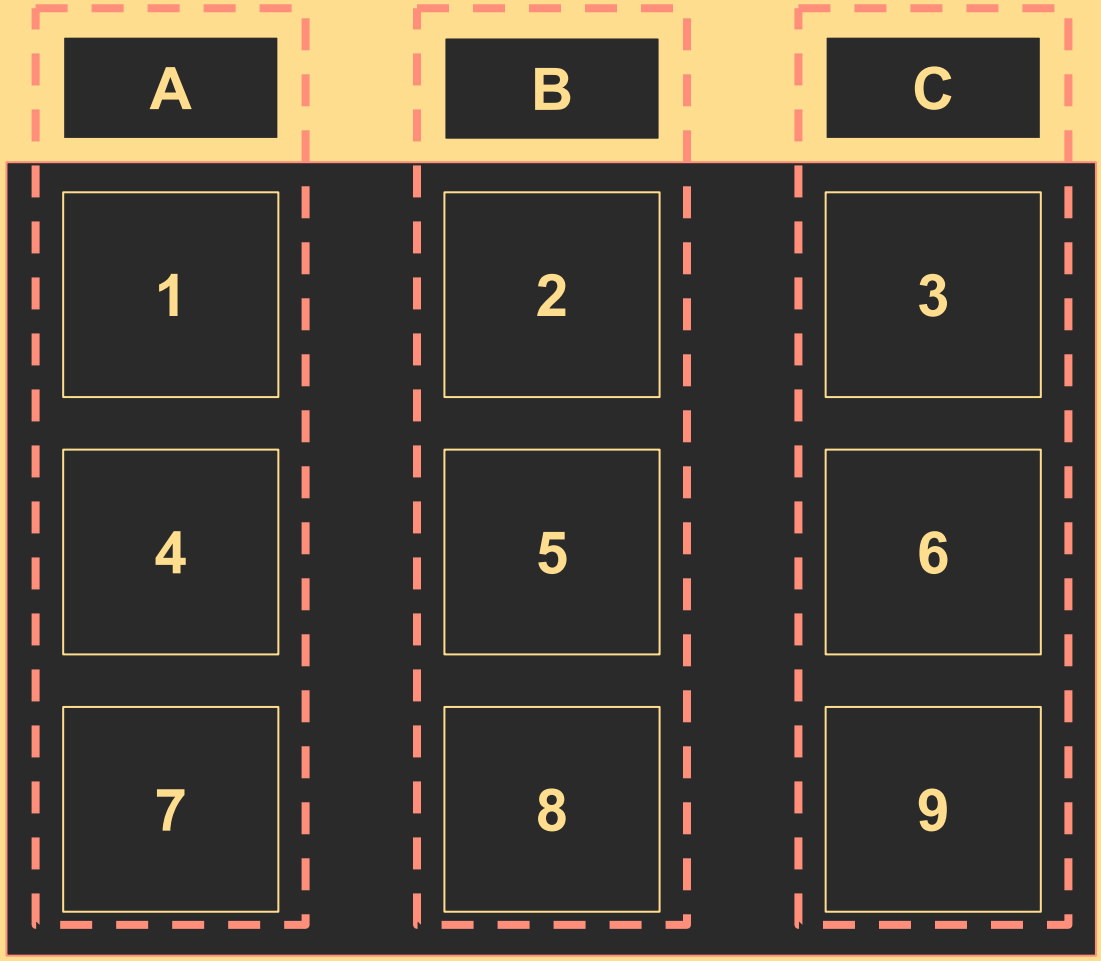
pip install pandas

Bora instalar!

Resolvendo problemas de indexação

```python
from pandas import DataFrame

l = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

df = DataFrame(l, columns=['A', 'B', 'C'])
```

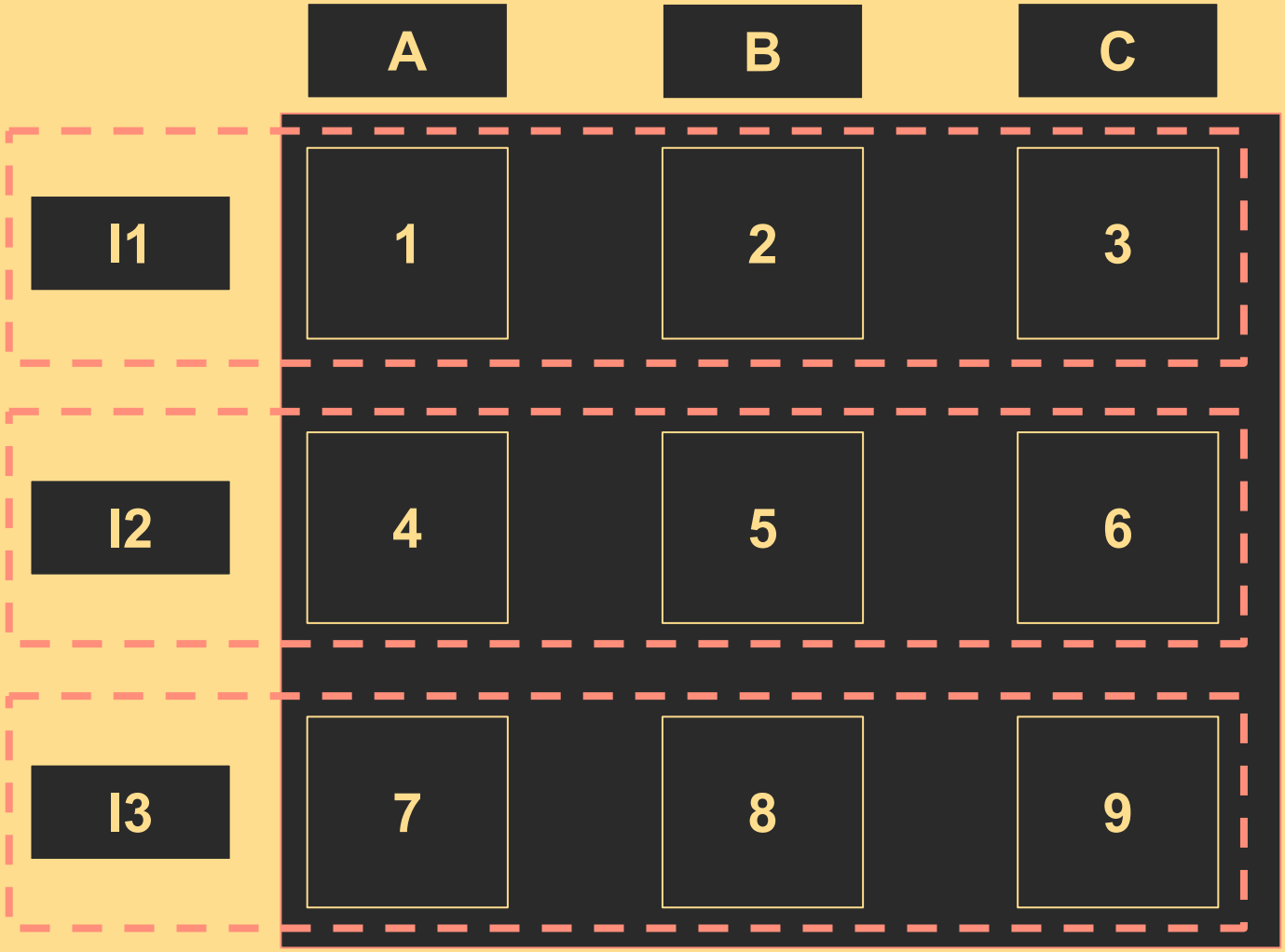
A nova estrutura [Colunas]



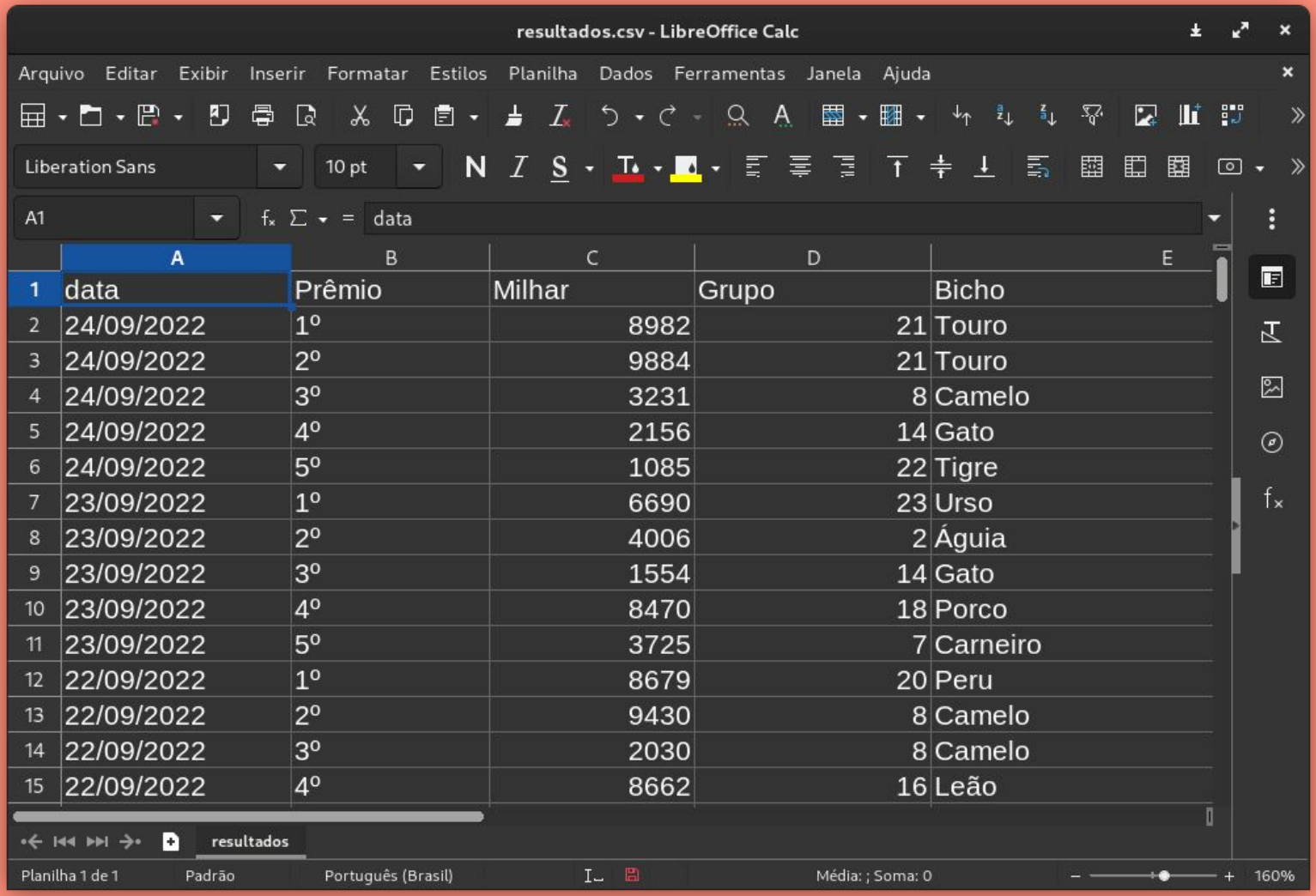
Resolvendo problemas de indexação

```python
from pandas import DataFrame

l = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

df = DataFrame(
  l,
  columns=['A', 'B', 'C'],
  index=['l1', 'l2', 'l3']
)
```

A nova estrutura [Indexes]



A primeira analogia



	A	B	C	D	E
1	data	Prêmio	Milhar	Grupo	Bicho
2	24/09/2022	1º	8982	21	Touro
3	24/09/2022	2º	9884	21	Touro
4	24/09/2022	3º	3231	8	Camelo
5	24/09/2022	4º	2156	14	Gato
6	24/09/2022	5º	1085	22	Tigre
7	23/09/2022	1º	6690	23	Urso
8	23/09/2022	2º	4006	2	Águia
9	23/09/2022	3º	1554	14	Gato
10	23/09/2022	4º	8470	18	Porco
11	23/09/2022	5º	3725	7	Carneiro
12	22/09/2022	1º	8679	20	Peru
13	22/09/2022	2º	9430	8	Camelo
14	22/09/2022	3º	2030	8	Camelo
15	22/09/2022	4º	8662	16	Leão

Ela não é necessariamente verdadeira

As "tabelas" do Pandas são chamadas de DataFrames. DataFrames são estruturas que são:

- Homogêneas por coluna (dados do mesmo tipo)
- Heterogêneas por linha (dados de tipos diferentes)
- Apresentam operações algébricas
- São bi-dimensionais

E as colunas são formadas por outra estrutura de dados chamada **Serie**

Series

A base de tudo

Series

Séries são estruturas como arrays unidimensionais, **vetores**. Contam, porém, com labels também.

```
1   # Exemplo de estrutura dom label
2   d = {
3      'Eduardo':  29,
4      'Fausto':   6,
5      'Joaquina': 31
6   }
```

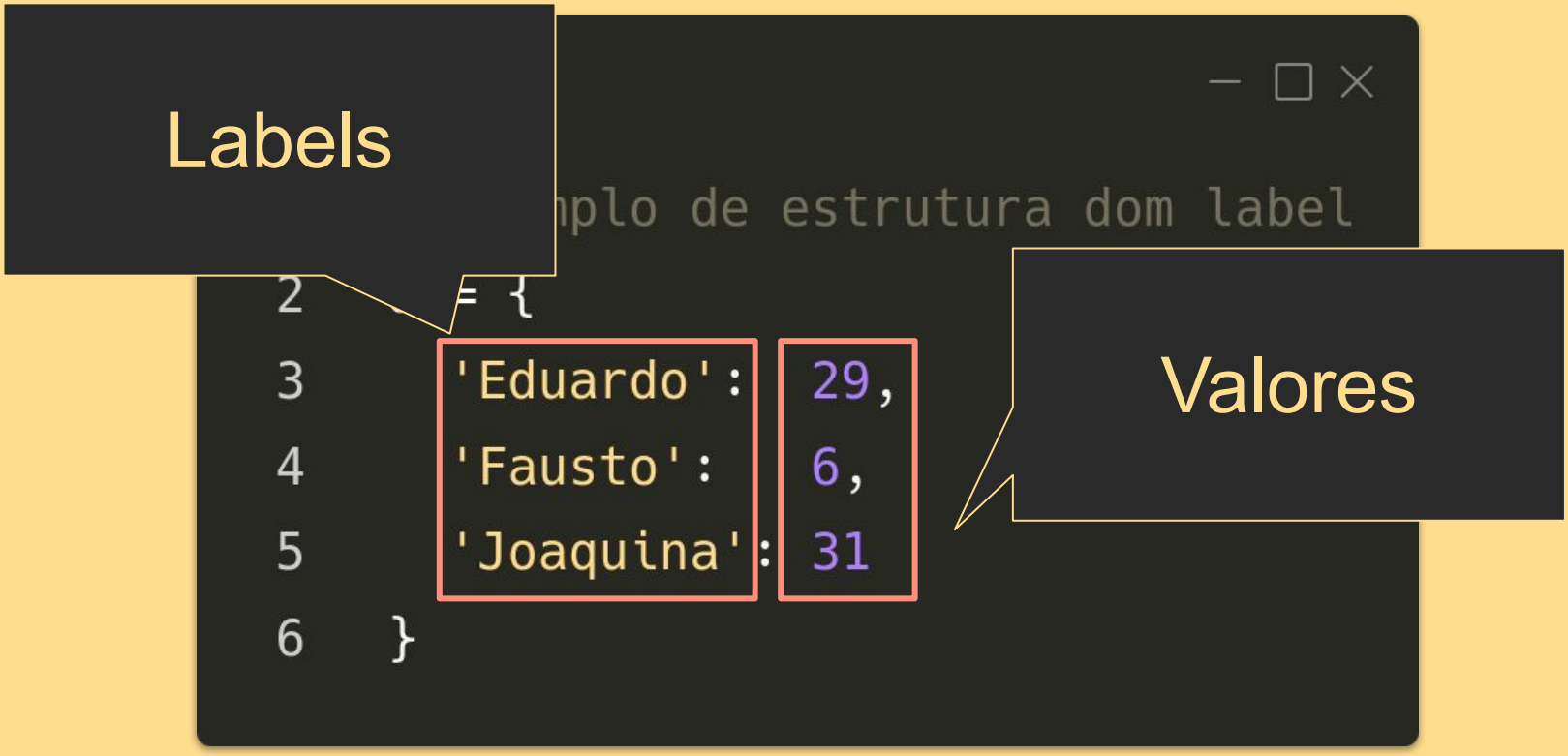
Series

Séries são estruturas como arrays unidimensionais, **vetores**. Contam, porém, com labels também.

```
1   # Exemplo de estrutura dom label
2   d = {
3       'Eduardo': 29,
4       'Fausto': 6,
5       'Joaquina': 31
6   }
```

Series

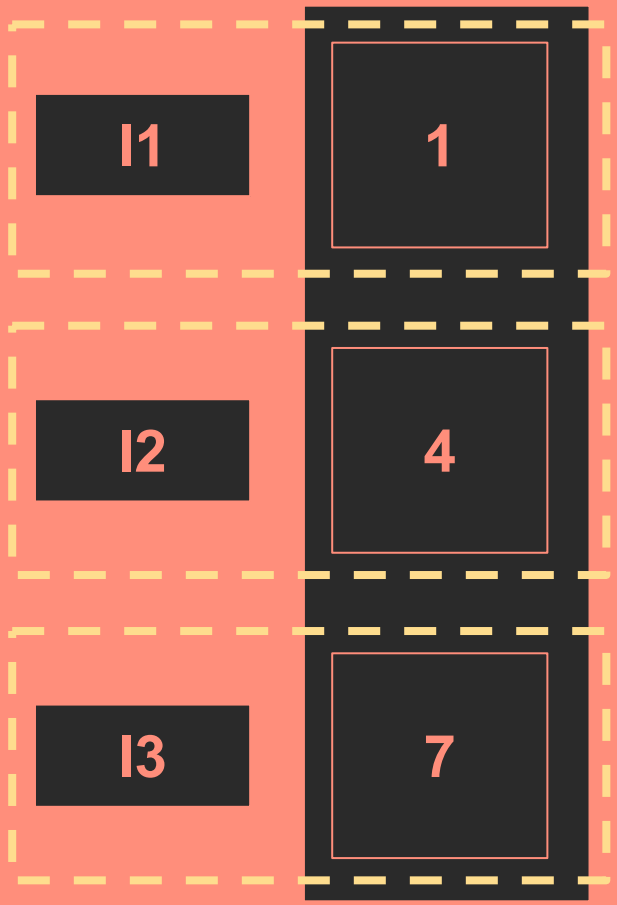
Séries são estruturas como arrays unidimensionais, **vetores**. Contam, porém, com labels também.



Series

- Representação de um array 1d no pandas
- Homogêneo
- Permite operações algébricas
- Filtros
- Gráficos (Plots)
- Agrupamento

```python
from pandas import Series
d = {
  'Eduardo':  29,
  'Fausto':   6,
  'Joaquina': 31
}

s = Series(d)
```



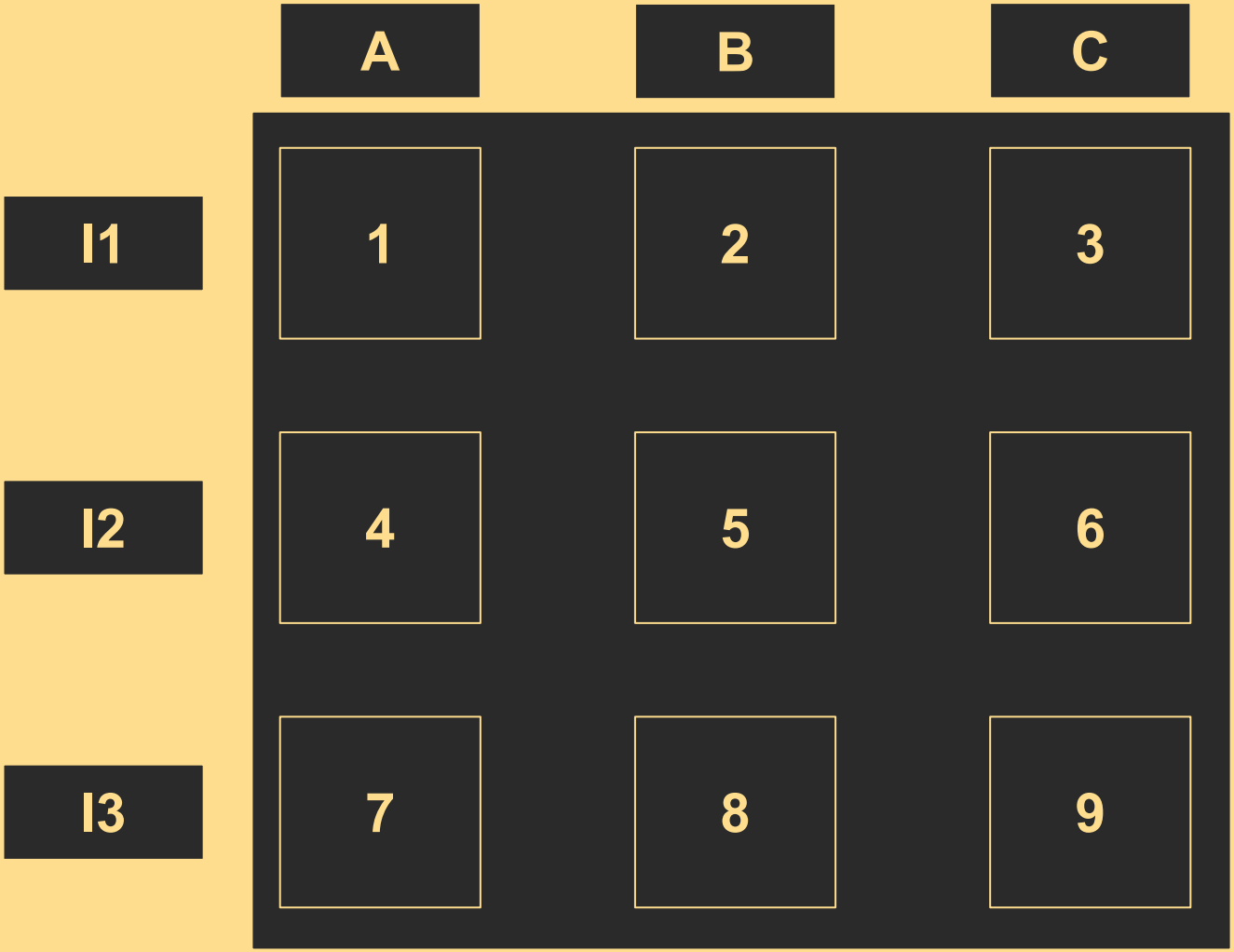
Data Frames

Agora SIM!

Resolvendo problemas de indexação

```python
from pandas import DataFrame

l = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

df = DataFrame(
  l,
  columns=['A', 'B', 'C'],
  index=['l1', 'l2', 'l3']
)
```

A estrutura



Ela não é necessariamente verdadeira

DataFrames são estruturas que são:

- Homogêneas por coluna (dados do mesmo tipo)
- Heterogêneas por linha (dados de tipos diferentes)
- Apresentam operações algébricas
- São bi-dimensionais

São uma serie de **Series** juntas

picpay.me/dunossauro

apoia.se/livedepython

pix.dunossauro@gmail.com

Ajude o projeto <3