

# pandas

Introdução às estruturas de dados do pandas  
Series

# pandas

## Estruturas de dados do pandas - Series

Uma Series é um objeto do tipo array unidimensional que contém uma sequência de valores e um array associado de rótulos de dados, chamado de índice. A Series mais simples é composta de apenas um array de dados.

Veja um exemplo:

```
In [1]: import pandas as pd

In [2]: series = pd.Series([10, 75, 84, 698, 42])

In [3]: series
Out[3]:
0      10
1      75
2      84
3     698
4      42
dtype: int64
```

Observe que a representação da Series mostra um índice à esquerda e os dados à direita. Este índice por padrão vai de zero à quantidade de dados menos um (0 até N-1).

# pandas

## Estruturas de dados do pandas - Series

Podemos obter informações de valores e índices utilizando `objeto.values` e `objeto.index`, veja:

```
In [1]: import pandas as pd
```

```
In [2]: series = pd.Series([10, 75, 84, 698, 42])
```

```
In [3]: series
```

```
Out[3]:
```

```
0      10
```

```
1      75
```

```
2      84
```

```
3     698
```

```
4      42
```

```
dtype: int64
```

```
In [4]: series.values
```

```
Out[4]: array([ 10,  75,  84, 698,  42], dtype=int64)
```

```
In [5]: series.index
```

```
Out[5]: RangeIndex(start=0, stop=5, step=1)
```

# pandas

## Estruturas de dados do pandas - Series

Podemos definir o índice para utilizarmos como um rótulo, facilitando identificação dos valores.

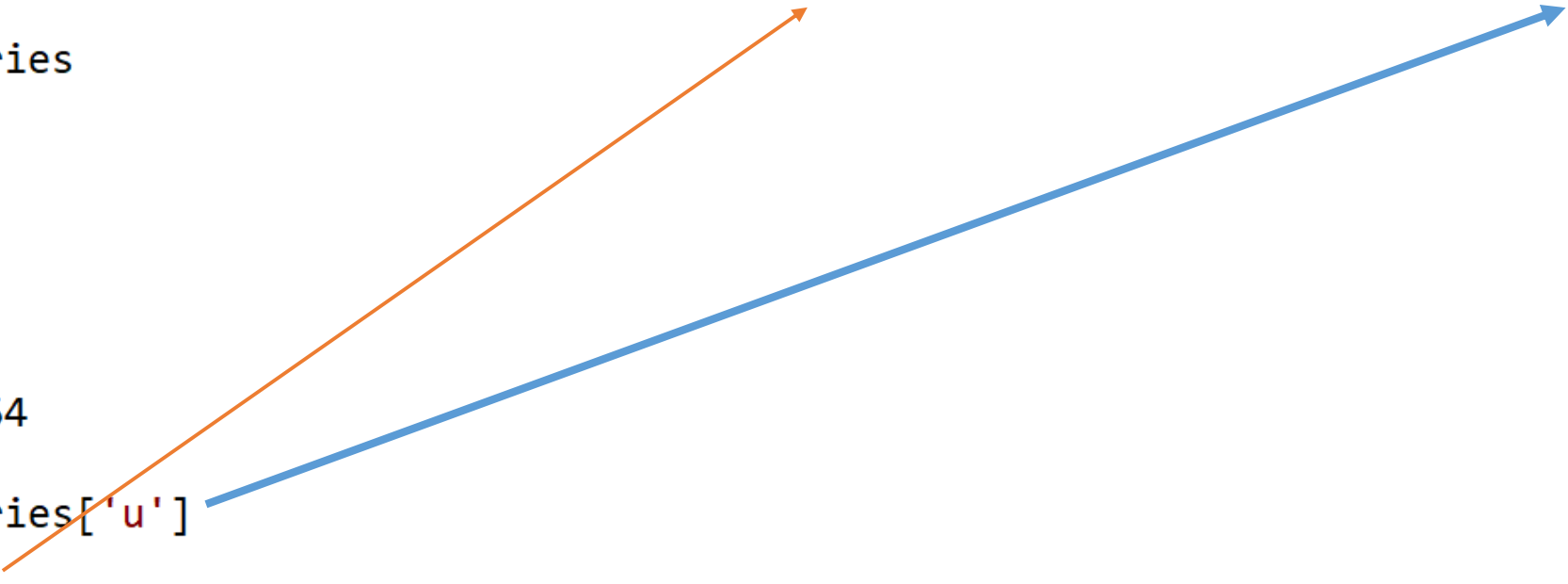
```
In [6]: series = pd.Series([10, 75, 84, 698, 42], index=['a', 'e', 'i', 'o', 'u'])  
  
In [7]: series  
Out[7]:  
a      10  
e      75  
i      84  
o     698  
u      42  
dtype: int64
```

# pandas

## Estruturas de dados do pandas - Series

Podemos utilizar o rótulo do índice para obter um determinado valor:

```
In [6]: series = pd.Series([10, 75, 84, 698, 42], index=['a', 'e', 'i', 'o', 'u'])  
  
In [7]: series  
Out[7]:  
a      10  
e      75  
i      84  
o     698  
u      42  
dtype: int64  
  
In [8]: series['u']  
Out[8]: 42
```



# pandas

## Estruturas de dados do pandas - Series

Podemos alterar um valor também informando o rótulo do índice:

```
In [7]: series
Out[7]:
a      10
e      75
i      84
o     698
u      42
dtype: int64
```

```
In [9]: series['i'] = 100

In [10]: series
Out[10]:
a      10
e      75
i     100
o     698
u      42
dtype: int64
```

# pandas

## Estruturas de dados do pandas - Series

Podemos usar operações do tipo Numpy, como filtragem de um array booleano, multiplicação escalar ou aplicação de funções matemáticas, preservando a ligação entre o índice e seu valor.

```
In [11]: series[series > 80]
Out[11]:
i    100
o    698
dtype: int64
```

```
In [12]: series * 3
Out[12]:
a      30
e     225
i     300
o    2094
u     126
dtype: int64
```

```
IPython: C:\Users\TecCard

In [23]: import numpy as np

In [24]: np.exp(series)
Out[24]:
a      2.202647e+04
e      3.733242e+32
i      2.688117e+43
o      1.372614e+303
u      1.739275e+18
dtype: float64
```

# pandas

## Estruturas de dados do pandas - Series

Podemos validar a existência de um índice na Series:

```
In [25]: 'a' in series  
Out[25]: True
```

```
In [26]: 'z' in series  
Out[26]: False
```



## Estruturas de dados do pandas - Series

### Criando Series à partir de um dicionário:

```
IPython: C:\Users\TecCard
In [30]: populacao_estados = {'São Paulo': 45538936, 'Minas Gerais': 21040662, 'Rio de Janeiro': 17159960, 'Bahia': 14812617, 'Paraná': 11348937, 'Rio Grande do Sul': 11329605, 'Pernambuco': 9496294, 'Ceará': 9075649, 'Pará': 8513497, 'Santa Catarina': 7075494, 'Maranhão': 7035055, 'Goiás': 6921161, 'Amazonas': 4080611, 'Paraíba': 3996496, 'Espírito Santo': 3972388, 'Rio Grande do Norte': 3479010, 'Mato Grosso': 3441998, 'Alagoas': 3322820, 'Piauí': 3264531, 'Distrito Federal': 2974703, 'Mato Grosso do Sul': 2748023, 'Sergipe': 2278308, 'Rondônia': 1757589, 'Tocantins': 1555229, 'Acre': 869265, 'Amapá': 829494, 'Roraima': 576568}
```

```
In [28]: populacao = pd.Series(populacao_estados)
```

```
In [29]: populacao
```

```
Out[29]:
```

São Paulo	45538936
Minas Gerais	21040662
Rio de Janeiro	17159960
Bahia	14812617
Paraná	11348937
Rio Grande do Sul	11329605
Pernambuco	9496294
Ceará	9075649
Pará	8513497
Santa Catarina	7075494
Maranhão	7035055
Goiás	6921161
Amazonas	4080611

Paraíba	3996496
Espírito Santo	3972388
Rio Grande do Norte	3479010
Mato Grosso	3441998
Alagoas	3322820
Piauí	3264531
Distrito Federal	2974703
Mato Grosso do Sul	2748023
Sergipe	2278308
Rondônia	1757589
Tocantins	1555229
Acre	869265
Amapá	829494
Roraima	576568

## Estruturas de dados do pandas - Series

Podemos criar um novo objeto Series à partir deste objeto informando um novo índice (veja o resultado quando informamos um índice inexistente, NaN demonstra dados ausentes):

```
In [35]: estados_sudeste = ['Espírito Santo', 'Minas Gerais', 'São Paulo', 'Rio de Janeiro']
```

```
In [36]: series_sudeste = pd.Series(populacao, index=estados_sudeste)
```

```
In [37]: series_sudeste
```

```
Out[37]:  
Espírito Santo    3972388  
Minas Gerais      21040662  
São Paulo         45538936  
Rio de Janeiro    17159960  
dtype: int64
```

```
In [38]: estados_sudeste = ['Espírito Santo', 'Minas Gerais', 'São Paulo', 'Rio de Janeiro', 'Wolkerslândia']
```

```
In [39]: series_sudeste = pd.Series(populacao, index=estados_sudeste)
```

```
In [40]: series_sudeste
```

```
Out[40]:  
Espírito Santo    3972388.0  
Minas Gerais      21040662.0  
São Paulo         45538936.0  
Rio de Janeiro    17159960.0  
Wolkerslândia      NaN  
dtype: float64
```

## Estruturas de dados do pandas - Series

Podemos tratar os dados ausentes com isnull e notnull:

```
In [42]: pd.isnull(series_sudeste)
Out[42]:
Espírito Santo      False
Minas Gerais        False
São Paulo            False
Rio de Janeiro      False
Wolkerslândia       True
dtype: bool

In [43]: pd.notnull(series_sudeste)
Out[43]:
Espírito Santo      True
Minas Gerais        True
São Paulo            True
Rio de Janeiro      True
Wolkerslândia       False
dtype: bool
```

```
In [44]: series_sudeste.isnull()
Out[44]:
Espírito Santo      False
Minas Gerais        False
São Paulo            False
Rio de Janeiro      False
Wolkerslândia       True
dtype: bool

In [45]: series_sudeste.notnull()
Out[45]:
Espírito Santo      True
Minas Gerais        True
São Paulo            True
Rio de Janeiro      True
Wolkerslândia       False
dtype: bool
```

## Estruturas de dados do pandas - Series

Podemos definir um nome para o objeto Series e para seu índice:

```
In [39]: series_sudeste.name = 'Estados do Sudeste'
```

```
In [40]: series_sudeste
```

```
Out[40]:
```

Espírito Santo	3972388.0
Minas Gerais	21040662.0
São Paulo	45538936.0
Rio de Janeiro	17159960.0
EvaldoLândia	NaN

Name: Estados do Sudeste, dtype: float64

```
In [51]: series_sudeste.index.name = 'Estados'
```

```
In [52]: series_sudeste
```

```
Out[52]:
```

Estados	
Espírito Santo	3972388.0
Minas Gerais	21040662.0
São Paulo	45538936.0
Rio de Janeiro	17159960.0
Wolkerslândia	NaN

Name: Estados do Sudeste, dtype: float64

## Estruturas de dados do pandas - Series

Podemos também alterar o índice:

```
In [53]: series_sudeste.index = ['Estado um', 'Estado dois', 'Estado três', 'Estado quatro', 'Estado cinco']

In [54]: series_sudeste
Out[54]:
Estado um          3972388.0
Estado dois       21040662.0
Estado três       45538936.0
Estado quatro     17159960.0
Estado cinco              NaN
Name: Estados do Sudeste, dtype: float64
```

## Estruturas de dados do pandas - Series

Neste endereço você encontrará uma lista de atributos e métodos do objeto Series:

<http://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.Series.html#pandas.Series>

# FIM