



# Importando a Base

## Documentação Oficial do Pandas

pandas documentation – pandas 1.2.1 documentation

Saw a typo in the documentation? Want to improve existing functionalities? The contributing guidelines will guide you through the process of improving pandas.

 <https://pandas.pydata.org/docs/>

## Lendo um CSV

- importar a biblioteca do pandas
- ler arquivo csv

```
importa pandas as pd
pd.read_csv('nome_do_arquivo')
```

## Ajuda do Jupyter

- com o cursor em uma função ou método
- apertando as teclas SHIFT + TAB
- ele abre uma ajuda com a documentação relativa ao método

## Modificando o Separador

- no arquivo acima está sendo utilizado como separador o ';'.
- por isso o pandas não separa corretamente as colunas

```
pd.read_csv('dados/aluguel.csv')
```

	Tipo;Bairro;Quartos;Vagas;Suites;Area;Valor;Condominio;IPFU
0	Quitinete;Copacabana;1;0;0;40;1700;500;60
1	Casa;Jardim Botânico;2;0;1;100;7000;;
2	Conjunto Comercial/Sala;Barra da Tijuca;0;4;0;...;
3	Apartamento;Centro;1;0;0;15;800;390;20
4	Apartamento;Higienópolis;1;0;0;48;800;230;...
...	...
32955	Quitinete;Centro;0;0;0;27;800;350;25
32956	Apartamento;Jacarepaguá;3;1;2;78;1800;800;40
32957	Apartamento;São Francisco Xavier;2;1;0;48;1400;...
32958	Apartamento;Leblon;2;0;0;70;3000;760;...
32959	Conjunto Comercial/Sala;Centro;0;0;0;250;6500;...;

32960 rows × 1 columns

- a documentação apresenta o atributo **sep**, em que o padrão é ','

```
Signature:  
pd.read_csv(  
    filepath_or_buffer: Union[str, pathlib.Path, IO[~AnyStr]],  
    sep=',',  
    delimiter=None,  
    header='infer',  
    names=None,  
    index_col=None,  
    usecols=None,  
    squeeze=False,  
    prefix=None,  
    mangle_dupe_cols=True,  
    dtype=None,  
    engine=None,  
    converters=None,
```

- ajustando o separador

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

```
In [3]: pd.read_csv('dados/aluguel.csv', sep = ';')
```

```
Out[3]:
```

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU
0	Quitinete	Copacabana	1	0	0	40	1700.0	500.0	60.0
1	Casa	Jardim Botânico	2	0	1	100	7000.0	NaN	NaN
2	Conjunto Comercial/Sala	Barra da Tijuca	0	4	0	150	5200.0	4020.0	1111.0
3	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0
4	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	NaN
...	...	...	...	...	...	...	...	...	...
32955	Quitinete	Centro	0	0	0	27	800.0	350.0	25.0
32956	Apartamento	Jacarepaguá	3	1	2	78	1800.0	800.0	40.0
32957	Apartamento	São Francisco Xavier	2	1	0	48	1400.0	509.0	37.0
32958	Apartamento	Leblon	2	0	0	70	3000.0	760.0	NaN
32959	Conjunto Comercial/Sala	Centro	0	0	0	250	6500.0	4206.0	1109.0

32960 rows × 9 columns

## Informações Sobre o Data Frame

- colocando o data frame dentro da variável dados:

```
dados = pd.read_csv('dados/aluguel.csv')
```

- informar o tipo de variável:

```
type(dados) ⇒ pandas.core.frame.DataFrame
```

- solicitando informações sobre o objeto:

```
dados.info()
```

```
dados.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32960 entries, 0 to 32959
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Tipo         32960 non-null   object 
 1   Bairro       32960 non-null   object 
 2   Quartos     32960 non-null   int64  
 3   Vagas        32960 non-null   int64  
 4   Suites       32960 non-null   int64  
 5   Area          32960 non-null   int64  
 6   Valor         32943 non-null   float64 
 7   Condominio    28867 non-null   float64 
 8   IPTU          22723 non-null   float64 
dtypes: float64(3), int64(4), object(2)
memory usage: 2.3+ MB
```

## Documentando o Notebook

- é possível ir documentando o notebook utilizando células **markdown**
- `#` cria um título h1, `##` h2 e `###` h3
- para adicionar um comentário dentro de uma célula de código em python, pode-se usar `#`
- escrevendo um comentário e apertando CTRL ; ⇒ cria um comentário com `#`
- apertando ESC depois a tecla **m** ⇒ passa a célula para markdown
- apertando a tecla **y** ⇒ passa para código novamente