



Selecionando Apenas Uma Variável

Lendo dados csv trocando separador e lendo os 10 primeiros

```
dados = pd.read_csv('../dados/aluguel.csv', sep = ';')
dados.head(10)
```

Selecionando apenas a variável 'Tipo'

```
# dados['Tipo'] ou
dados.Tipo # essa conotação só serve para um nome de variável válido
tipo_de_imovel = dados['Tipo']
type(tipo_de_imovel)
```

- existem alguns métodos para trabalhar com uma variável

```
df.value_counts() # mostra as variáveis disponíveis e qtas vezes elas aparecem
df.unique() # mostra as variáveis disponíveis
tipo_de_imovel.drop_duplicates()
# mostra apenas a 1ª ocorrência e elimina as outras
# é mostrado o índice da primeira ocorrência
```

Dropduplicates

- o `drop_duplicates()` possui argumentos que devemos conhecer
- ao pressionarmos "Shift + Tab", podemos analisar estes argumentos
- quando temos o `keep` configurado como `first`, todas as duplicatas são eliminadas
- o `inplace`, quando seu default esta marcado como `true`, ele modifica a variável a partir da execução do método
- não precisamos criar uma variável e atribuir a execução à ela

Fixando o método na variável

```
tipo_de_imovel.drop_duplicates(inplace =True )  
tipo_de_imovel
```

Organizando a Visualização

```
tipo_de_imovel = pd.DataFrame(tipo_de_imovel)  
tipo_de_imovel
```

	Tipo
0	Quitinete
1	Casa
2	Conjunto Comercial/Sala
3	Apartamento
7	Casa de Condomínio
16	Prédio Inteiro
17	Flat
29	Loja/Salão
80	Galpão/Depósito/Armazém
83	Casa Comercial
117	Casa de Vila
159	Terreno Padrão
207	Box/Garagem
347	Loft
589	Loja Shopping/ Ct Comercial
2157	Chácara
3354	Loteamento/Condomínio
4379	Sítio
4721	Pousada/Chalé
6983	Studio
9687	Hotel
23614	Indústria

Modificando o index

- como é mostrado o índice da primeira ocorrência, iremos reorganizar o index para fazer mais sentido

```
tipo_de_imovel.index
# observa os valores do index
```

```
Int64Index([    0,    1,    2,    3,    7,   16,   17,   29,   80,
            83,   117,   159,   207,   347,   589,  2157,  3354, 4379,
            4721,  6983,  9687, 23614],
           dtype='int64')
```

```
tipo_de_imovel.shape[0]
# mostra o n° de linhas do data frame
# => 22
```

```
range(tipo_de_imovel.shape[0])
# vendo o range do index
# => range(0, 22)
```

```
# pode-se usar o for para visualizar o range
for i in range(tipo_de_imovel.shape[0]):
    print(i)
```

- sabendo o range, é possível numerar manualmente

```
tipo_de_imovel.index = range(1, 23)
```

- lembrando que o range não pega o último valor
- mas numera-se automaticamente para facilitar manutenção futura

```
tipo_de_imovel.index = range(tipo_de_imovel.shape[0])
tipo_de_imovel
```

	Tipo
0	Quitinete
1	Casa
2	Conjunto Comercial/Sala
3	Apartamento
4	Casa de Condomínio
5	Prédio Inteiro
6	Flat
7	Loja/Salão
8	Galpão/Depósito/Armazém
9	Casa Comercial
10	Casa de Vila
11	Terreno Padrão
12	Box/Garagem
13	Loft
14	Loja Shopping/ Ct Comercial
15	Chácara
16	Loteamento/Condomínio
17	Sítio
18	Pousada/Chalé
19	Studio
20	Hotel
21	Indústria

- depois dessa definição, se aplicarmos:

```
tipo_de_imovel.index
```

- temos \Rightarrow RangeIndex(start=0, stop=22, step=1)
- pode-se ver acima, onde começa, onde termina e o passo do index

Renomeando a Coluna Index

```
tipo_de_imovel.columns.name = 'Id'
```

Resultado Final

Id	Tipo
0	Quitinete
1	Casa
2	Conjunto Comercial/Sala
3	Apartamento
4	Casa de Condomínio
5	Prédio Inteiro
6	Flat
7	Loja/Salão
8	Galpão/Depósito/Armazém
9	Casa Comercial
10	Casa de Vila
11	Terreno Padrão
12	Box/Garagem
13	Loft
14	Loja Shopping/ Ct Comercial
15	Chácara
16	Loteamento/Condomínio
17	Sítio
18	Pousada/Chalé
19	Studio
20	Hotel