Practical one pandas: -

Link of data set

https://www.kaggle.com/datasets/zahrayazdani81/carsdataset

```
import pandas as pd

# read dataSet csv

df = pd.read_csv("cardata.csv")
print(df.head(None))
```

[26] ✓ 0.0s

```
...      Car_Name  Year  Selling_Price  Present_Price  Kms_Driven Fuel_Type  \
0          ritz  2014           3.35           5.59       27000    Petrol
1           sx4  2013           4.75           9.54       43000    Diesel
2          ciaz  2017           7.25           9.85        6900    Petrol
3      wagon r  2011           2.85           4.15        5200    Petrol
4         swift  2014           4.60           6.87       42450    Diesel
..          ...   ...            ...            ...         ...       ...
296        city  2016           9.50          11.60       33988    Diesel
297        brio  2015           4.00           5.90       60000    Petrol
298        city  2009           3.35          11.00       87934    Petrol
299        city  2017          11.50          12.50        9000    Diesel
300        brio  2016           5.30           5.90        5464    Petrol

    Seller_Type Transmission  Owner
0        Dealer       Manual      0
1        Dealer       Manual      0
2        Dealer       Manual      0
3        Dealer       Manual      0
4        Dealer       Manual      0
..          ...          ...    ...
296      Dealer       Manual      0
297      Dealer       Manual      0
298      Dealer       Manual      0
299      Dealer       Manual      0
300      Dealer       Manual      0
```

لاحظ انه مش جايب الداتا كلها جنب بعض

```
# Set pandas option to display all columns beside each other
pd.set_option("display.width", 150)
```

استخدم دي

```python
# Set pandas option to display all columns beside each other
pd.set_option("display.width", 150)

# read dataSet csv
df = pd.read_csv("cardata.csv")
print(df.head(None))
```

7] ✓ 0.0s

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.590	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.540	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.850	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.150	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.870	42450	Diesel	Dealer	Manual	0
5	vitara brezza	2018	9.25	9.830	2071	Diesel	Dealer	Manual	0
6	ciaz	2015	6.75	8.120	18796	Petrol	Dealer	Manual	0
7	s cross	2015	6.50	8.610	33429	Diesel	Dealer	Manual	0
8	ciaz	2016	8.75	8.890	20273	Diesel	Dealer	Manual	0
9	ciaz	2015	7.45	8.920	42367	Diesel	Dealer	Manual	0
10	alto 800	2017	2.85	3.600	2135	Petrol	Dealer	Manual	0
11	ciaz	2015	6.85	10.380	51000	Diesel	Dealer	Manual	0
12	ciaz	2015	7.50	9.940	15000	Petrol	Dealer	Automatic	0
13	ertiga	2015	6.10	7.710	26000	Petrol	Dealer	Manual	0
14	dzire	2009	2.25	7.210	77427	Petrol	Dealer	Manual	0
15	ertiga	2016	7.75	10.790	43000	Diesel	Dealer	Manual	0
16	ertiga	2015	7.25	10.790	41678	Diesel	Dealer	Manual	0
17	ertiga	2016	7.75	10.790	43000	Diesel	Dealer	Manual	0
18	wagon r	2015	3.25	5.090	35500	CNG	Dealer	Manual	0
19	sx4	2010	2.65	7.980	41442	Petrol	Dealer	Manual	0
20	alto k10	2016	2.85	3.950	25000	Petrol	Dealer	Manual	0

Check the data for cleaning

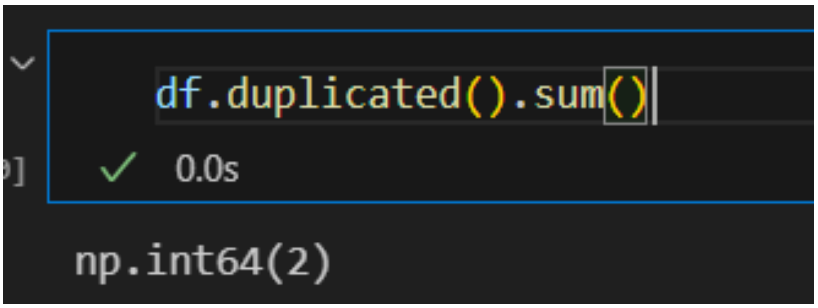
```python
df.info()
```

[58] ✓ 0.0s

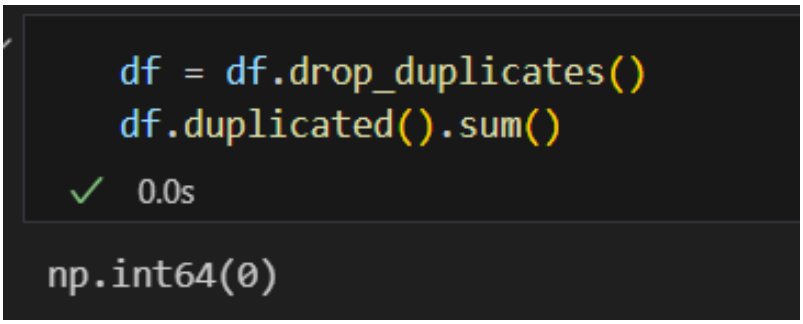
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301 entries, 0 to 300
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Car_Name       301 non-null    object
 1   Year           301 non-null    int64
 2   Selling_Price  301 non-null    float64
 3   Present_Price  301 non-null    float64
 4   Kms_Driven     301 non-null    int64
 5   Fuel_Type      301 non-null    object
 6   Seller_Type    301 non-null    object
 7   Transmission   301 non-null    object
 8   Owner          301 non-null    int64
dtypes: float64(2), int64(3), object(4)
memory usage: 21.3+ KB
```

No NaN or Na or any missing value

Check for duplicates

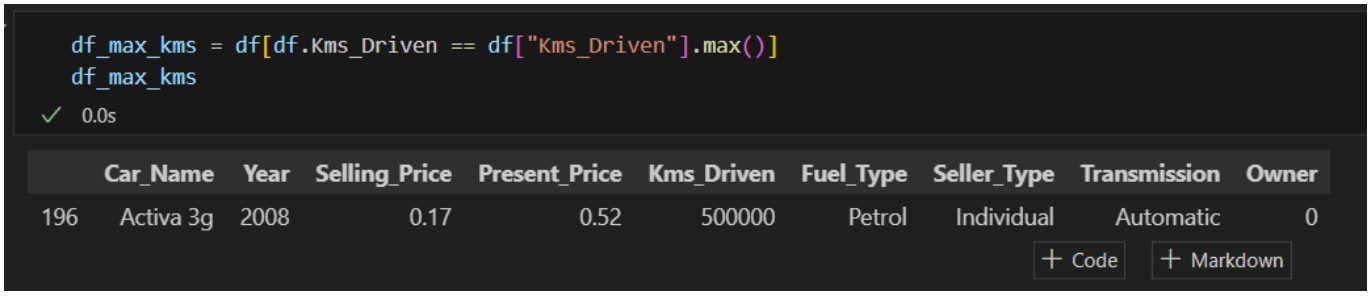


طلع فيه اتنين متكررين يلا نمسحهم



كدا مفيش متكرر

1- عاوزين نعرف أي العربيه الي بتمشي أكبر قدر ممكن



لا بس انا عاوز اسمها بس

```
df_max_kms = df[df.Kms_Driven == df["Kms_Driven"].max()]
df_max_kms["Car_Name"]
```

✓ 0.0s

```
196    Activa 3g
Name: Car_Name, dtype: object
```

2- عايز اقل واحده

```
# Find out which car can travel the lowest distance
df_min_kms = df[df.Kms_Driven == df["Kms_Driven"].min()]
df_min_kms
```

✓ 0.0s

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
133	Bajaj Avenger 220	2016	0.72	0.95	500	Petrol	Individual	Manual	0
165	Activa 3g	2016	0.45	0.54	500	Petrol	Individual	Automatic	0

طلع اتنين زي بعض

نقارن بين السعر و عايزين السعر الأقل

```
df_min_kms_price = df_min_kms[df_min_kms.Present_Price < df_min_kms["Present_Price"].max()]
df_min_kms_price
```

✓ 0.0s

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
165	Activa 3g	2016	0.45	0.54	500	Petrol	Individual	Automatic	0

3- عايز اعرف ايهم اعلي سعر

```
# find out the largest price
df_max_price = df[df.Present_Price == df["Present_Price"].max()]
df_max_price
```

✓ 0.0s

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
86	land cruiser	2010	35.0	92.6	78000	Diesel	Dealer	Manual	0

Source code:
https://github.com/MohamedElsyed2005/practical_one_pandas