Practical two pandas: -

Link table:

https://raw.githubusercontent.com/justmarkham/pandas-videos/master/data/chipotle.tsv

```
import pandas as pd

# read table from website

df = pd.read_table("https://raw.githubusercontent.com/justmarkham/pandas-videos/master/data/chipotle.tsv")
df
```

✓ 0.7s

	order_id	quantity	item_name	choice_description	item_price
0	1	1	Chips and Fresh Tomato Salsa	NaN	$2.39
1	1	1	Izze	[Clementine]	$3.39
2	1	1	Nantucket Nectar	[Apple]	$3.39
3	1	1	Chips and Tomatillo-Green Chili Salsa	NaN	$2.39
4	2	2	Chicken Bowl	[Tomatillo-Red Chili Salsa (Hot), [Black Beans...	$16.98
...	...	...	...	...	...
4617	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Black Beans, Sour ...	$11.75
4618	1833	1	Steak Burrito	[Fresh Tomato Salsa, [Rice, Sour Cream, Cheese...	$11.75
4619	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$11.25
4620	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Lettu...	$8.75
4621	1834	1	Chicken Salad Bowl	[Fresh Tomato Salsa, [Fajita Vegetables, Pinto...	$8.75

4622 rows × 5 columns

Data cleaning

```
df.info()
```

✓ 0.0s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4622 entries, 0 to 4621
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   order_id            4622 non-null   int64
 1   quantity            4622 non-null   int64
 2   item_name           4622 non-null   object
 3   choice_description  3376 non-null   object
 4   item_price          4622 non-null   object
dtypes: int64(2), object(3)
memory usage: 180.7+ KB
```

كلهم مش متساوين يعني فيه

Missing values

```
df.isna().sum()
```
✓ 0.0s

```
order_id                0
quantity                0
item_name               0
choice_description    1246
item_price              0
dtype: int64
```

فيه 1246 فيه

Missing values

```
# drop all rows that have na values
df = df.dropna()
```

⇨ After drop all rows

```
# drop all rows that have na values
df = df.dropna()
df.isna().sum()
```
✓ 0.0s

```
order_id              0
quantity              0
item_name             0
choice_description    0
item_price            0
dtype: int64
```

Check duplicated

```
# to know how many rows are duplicated
df.duplicated().sum()
```

[16] ✓ 0.0s

```
np.int64(41)
```

There 41 rows are duplicated

```
# to drop all rows are duplicated
df = df.drop_duplicates()
df.duplicated().sum()
```

[17] ✓ 0.0s

```
np.int64(0)
```

1- عاوزين نعرف كل منتج اشتري كام مره بس تلاحظ ان فيه تلاته كولوم واحد للاسم و التاني للكمية والتالت للسعر فلازم نعمل

Select => item_name , quantity and item_price

```
# select col item_name nad quantity
df1 = df[["item_name","quantity","item_price"]]
df1
```

[75] ✓ 0.0s

	item_name	quantity	item_price
1	Izze	1	$3.39
2	Nantucket Nectar	1	$3.39
4	Chicken Bowl	2	$16.98
5	Chicken Bowl	1	$10.98
7	Steak Burrito	1	$11.75

فيه علامه الدولار لازم تشيلها عشان تعرف تجمع الأرقام طب ازاي

```python
# select col item_name nad quantity
df1 = df[["item_name","quantity","item_price"]]
# remove dolar sign to make arithmetic operation => for int
# cast for str first to can remove $ then cast to intger
df1["item_price"] = df1["item_price"].str.replace("$"," ")
df1["item_price"] = df1["item_price"].astype(float)
df1
```

[88] ✓ 0.0s

	item_name	quantity	item_price
0	Chips and Fresh Tomato Salsa	1	2.39
1	Izze	1	3.39
2	Nantucket Nectar	1	3.39
3	Chips and Tomatillo-Green Chili Salsa	1	2.39
4	Chicken Bowl	2	16.98
...	...	...	...
4617	Steak Burrito	1	11.75
4618	Steak Burrito	1	11.75
4619	Chicken Salad Bowl	1	11.25
4620	Chicken Salad Bowl	1	8.75
4621	Chicken Salad Bowl	1	8.75

pandas.DataFrame.replace

```
DataFrame.replace(to_replace=None, value=_NoDefault.no_default, *,
inplace=False, limit=None, regex=False, method=_NoDefault.no_default)
```

بتتعامل مع النصوص بس ف لازم تحول الحاجه دي ل نص <= Replace

وبعد م بتشيل العلامه بتحولها لرقم تاني

طيب احنا عاوزين مجموع كل الكميه لكل عنصر معين و هنا هنستخدم Groupby وبعدها sum

```
# sum of all quantity to know how many each item was brought
df2 = df1.groupby("item_name").sum()
df2.head(10)
```

[89] ✓ 0.0s

item_name	quantity	item_price
6 Pack Soft Drink	55	356.95
Barbacoa Bowl	66	672.36
Barbacoa Burrito	91	894.75
Barbacoa Crispy Tacos	12	120.21
Barbacoa Salad Bowl	10	106.40
Barbacoa Soft Tacos	25	250.46
Bottled Water	211	302.56
Bowl	4	29.60
Burrito	6	44.40
Canned Soda	126	137.34

بعدها عاوزين نرتبهم علي حسب الكولوم بتاع الكميه و يكون تنازلي

sort_value(By = [“col_name”] , ascending =False)

```
DataFrame.sort_values(by, *, axis=0, ascending=True, inplace=False,
kind='quicksort', na_position='last', ignore_index=False, key=None) #
```

Sort by quantity

```
# sort the data frame by col quantity
df3 = df2.sort_values(by=["quantity"], ascending=False)
df3.head(10)
```

[90] ✓ 0.0s

item_name	quantity	item_price
Chicken Bowl	761	7342.73
Chicken Burrito	591	5575.82
Chips and Guacamole	506	2201.04
Steak Burrito	386	3851.43
Canned Soft Drink	351	438.75
Chips	230	494.34
Steak Bowl	221	2260.19
Bottled Water	211	302.56
Chips and Fresh Tomato Salsa	130	361.36
Canned Soda	126	137.34

Sorting by item_price

```
# sorting by item_price col
df4 = df2.sort_values(by=["item_price"], ascending=False)
df4.head(10)
```

[91] ✓ 0.0s

item_name	quantity	item_price
Chicken Bowl	761	7342.73
Chicken Burrito	591	5575.82
Steak Burrito	386	3851.43
Steak Bowl	221	2260.19
Chips and Guacamole	506	2201.04
Chicken Salad Bowl	123	1228.75
Chicken Soft Tacos	120	1108.09
Veggie Burrito	97	934.77
Barbacoa Burrito	91	894.75
Veggie Bowl	87	867.99