Pandas

---

df.dropna() >> if I have some missing values in my data frame and I want to delete them so we should use this method.

Now **dropna** has some default arguments which are **dropna(axis='index',how='any')** so that will delete any row containing any non value.

If we changed it to **dropna(axis='index',how='all')** >> that will delete any row if its all values are null.

If we want **dropna** to deal with columns we should change the axis **dropna(axis=coulmns,how='all')** >> here (any and all) have the same meaning.

df.dropna(axis='index', how='all', subset=['email']) >> that will return the row that at least has a value in the email field.

df.dropna(axis='index', how='all', subset=['last', 'email']) >> معناه انه مش هيمسح اي record ليه email or last name.

now if I want to know if the data frame has a (null , na ,null , non) values I can use **df.isna()** this method will return a data frame of (True and False)
True if there is a real value.
False if there is null value.

If I want to replace those values with a specific value so I should use **df.fillna(0)** and this will replace each null value with a 0.

If I want to change a column data type to another I should use **astype()**
df['age'] = df['age'].astype(float) >> that will change the data type of the 'age' column to a flout.

df['YearsCode']**.unique()** >> that will return all the unique values in the 'YearsCode' column.

If I want to rename the columns.

df.**rename**(columns = {‘old_name’ : ‘new_name’})

#to change columns data types.

df.astype({‘column_name’ : ‘new_data_type’, … })