**import** pandas **as** pd

**import** numpy **as** np

DATA**=** pd**.**Series(data**=**[1, 2, 3], index**=**['A', 'B', 'C']) ->

THE SAME THING

DATA**=** pd**.**Series(data**=**{'A': 1, 'B': 2, 'C': 3})

A 1

B 2

C 3

my\_series = pd.Series(pd.Series(data=[10, 20, 30])) -> it will start from 0 to 3

my\_df **=** pd**.**DataFrame(data**=**np**.**arange(0,20)**.**reshape(4,5), index**=**['A', 'B', 'C', 'D'],

columns**=**['col1', 'col2', 'col3', 'col4', 'col5'])

Index -> in the first column number of rows

Columns -> in the first-row number of columns, we can get any column by using my\_df[‘col1’]

pr.Series we use only data and index but in pd.DataFrame we use as many as we want columns

If we want to access any column we use **my\_df['col2']**

If it is 2 or more then we use **my\_df[['col2', 'col3']]**

**My\_df.iloc[0] it will give us all data in row 1**

**my\_df.loc['A'] it will give us all data in A -> loc and iloc**

Data [ Data % 3 == 0] if it is not divisible to 3 then we get NaN

my\_df[my\_df **%** 2 **==** 0]**.**fillna(value**=**0) here we can replace NaN numbers to 0

my\_df[my\_df % 2 == 0].fillna(value=my\_df['col2'].mean())

ExistingDATA [ “ ColumnNAME ” ] =[ 10, 20, 30, 40]

Data **.** drop ( columns **=** [ ' newCol ' ] ) -> it is not PERMAMENT!!!

Data **.** drop ( columns **=** [ ' newCol ' , ' col1 + col2 '], inplace **= True**)

Data = pd.DataFrame( { ' HOUSE': ['Falcon', 'Falcon', 'Parrot', 'Parrot', 'Cat', 'Cat', 'Cat'],

'PRICE': [380., 370., 24., 26., 50., 50., 150.]})

Data

DATA [ ' HOUSE ' ] . unique()) -> NO DOUBLE VALUES!!!

DATA [ ' HOUSE ' ] . value\_counts() -> WE CAN FIND HOW MANY THERE

DATA [' PRICE ' ] . mean() -> it gives us the average

DATA . groupby( by= ' HOUSE ' ) . mean () -> it will give us mean of speed of each animal without duplication

***head()*** shows us the first 5 values in the dataset by default, but you can specify more or less inside the parenthesis.

***info()*** helps us to identify missing data (414 non-null)

Data [ ' Type 2 ' ] **.** fillna ( Data [ ' Type 1 ' ] , inplace **= True** ) -> it will remove all missed values

pokemon\_df[pokemon\_df['Legendary'] == True] -> it will give us all TRUE values

What is the Difference between pokemon\_df['HP']**.**idxmax() and pokemon\_df['HP'].max()???

Data **.** groupby ( by **=** ' Type 1 ' ) **.** mean( ) [ ' HP ' ] -it will group by Type + give us means of all rows