Pandas

1. Pandas used for Data Analysis, Manipulation & Cleaning of large dataset

2. Data Structure
   - Series — One dimensional array with labels
   - DataFrame — Two dimensional array

3. Excersice

1 Initialization

```
!pip install pandas
import pandas
pandas.__version__

Requirement already satisfied: pandas in /usr
Requirement already satisfied: numpy>=1.15.4
Requirement already satisfied: pytz>=2017.2
Requirement already satisfied: python-dateut
Requirement already satisfied: six>=1.5 in
'1.1.5'
```

Syntax

```
import pandas as pd
a = [7,9,4]
dataset = pd.Series(a)
print(dataset)

0    7
1    9
2    4
dtype: int64
```

Syntax - Series without Index

```
import pandas as pd
a = [7,9,4]
dataset = pd.Series(a, index = ["No.1","No.2","No.3"])
print(dataset)

No.1    7
No.2    9
No.3    4
dtype: int64
```

Syntax - Series with Index

2 Loading Dataset - Variable

```
import pandas as pd
data = {"Name":["Champ","Mr.X","Mr.y"],"Color":["Red","Black","White"]}
dataset = pd.DataFrame(data)
print(dataset)

   Name  Color
0 Champ    Red
1  Mr.X  Black
2  Mr.y  White
```

Syntax - Dataframe

```
import pandas as pd
dataset = pd.read_csv("dataset.csv")
print(dataset.head(5))

   #             Name Type 1  ... Speed  Generation  Legendary
0  1        Bulbasaur  Grass  ...    45           1      False
1  2          Ivysaur  Grass  ...    60           1      False
2  3         Venusaur  Grass  ...    80           1      False
3  3  VenusaurMega Venusaur  Grass  ...    80           1      False
4  4       Charmander   Fire  ...    65           1      False

[5 rows x 12 columns]
```

Syntax - csv

3 Loading Dataset - From File

```
import pandas as pd
dataset = pd.read_csv("dataset.txt")
print(dataset.head(5))

   #(Name|Type 1|Type 2|HP|Attack|Defense|Sp. Atk|Sp. Def|Speed|Generation|Legendary
0  1|Bulbasaur|Grass|Poison|45|49|49|65|65|1|False
1  2|Ivysaur|Grass|Poison|60|62|63|80|80|1|False
2  3|Venusaur|Grass|Poison|80|82|83|100|100|1|False
3  3|VenusaurMega Venusaur|Grass|Poison|80|100|123|122|120|1|False
4  4|Charmander|Fire|65|52|43|60|50|65|1|False
```

Syntax - Txt

```
[12] import pandas as pd
     dataset = pd.read_excel("dataset.xlsx")
     print(dataset.head(5))

        #             Name Type 1  ... Speed  Generation  Legendary
     0  1        Bulbasaur  Grass  ...    45           1      False
     1  2          Ivysaur  Grass  ...    60           1      False
     2  3         Venusaur  Grass  ...    80           1      False
     3  3  VenusaurMega Venusaur  Grass  ...    80           1      False
     4  4       Charmander   Fire  ...    65           1      False

     [5 rows x 12 columns]
```

Syntax - XLSX

4 Details about Dataset

```
print(dataset1.shape) #No.Of Rows and Columns
print(dataset1.describe()) #Detailed info about Dataset

(800, 12)
                #          HP      Attack  ...     Sp. Def       Speed  Generation
count  800.000000  800.000000  800.000000  ...  800.000000  800.000000  800.000000
mean   362.813750   69.258750   79.001250  ...   71.902500   68.277500    3.32375
std    208.343798   25.534669   32.457366  ...   27.828916   29.060474    1.66129
min      1.000000    1.000000    5.000000  ...   20.000000    5.000000    1.00000
25%    184.750000   50.000000   55.000000  ...   50.000000   45.000000    2.00000
50%    364.500000   65.000000   75.000000  ...   70.000000   65.000000    3.00000
75%    539.250000   80.000000  100.000000  ...   90.000000   90.000000    5.00000
max    721.000000  255.000000  190.000000  ...  230.000000  180.000000    6.00000
```

Syntax

5 Data Slicing

```
print(dataset1.columns) #To get only Column Title
print(dataset1["Name"]) #To get data based on the specific Title
print(dataset1[["Name","Speed"]]) #To get data based on the multiple Title
print(dataset1["Name"][0:6]) #To get data based on the specific Title with Specific Count
```

Syntax - Title

```
print(dataset1.iloc[1]) #integer location-acquire complete info about certain index
print(dataset1.iloc[1:5]) #acquire complete info about range of index
print(dataset1.iloc[1,2]) #acquire specific cell data
```

Iloc - Integer Location

```
for index,row in dataset1.iterrows():
  print(index,row["Name"])

741 Golgopf
742 Pancham
```

Iterrows

```
print(dataset1.loc[dataset1["Speed"]>90])

       #                    Name Type 1  ... Speed  Generation  Legendary
6      6               Charizard   Fire  ...   100           1      False
7      6  CharizardMega Charizard X   Fire  ...   100           1      False
8      6  CharizardMega Charizard Y   Fire  ...   100           1      False
19    15       BeedrillMega Beedrill    Bug  ...   145           1      False
22    18                 Pidgeot  Normal  ...   101           1      False
```

Filter - Loc

```
dataset1.sort_values(["HP"],ascending=False)
```

	#	Name	Type 1	Type 2	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed
261	242	Blissey	Normal	NaN	255	10	10	75	135	55
121	113	Chansey	Normal	NaN	250	5	5	35	105	50
217	202	Wobbuffet	Psychic	NaN	190	33	58	33	58	33

Sorting

6 Editing Data

```
dataset1["Power"] = dataset1["HP"] + dataset1["Attack"]
print(dataset1.head(5))

   #             Name Type 1  Type 2  ...  Speed  Generation  Legendary  Power
0  1        Bulbasaur  Grass  Poison  ...     45           1      False     94
1  2          Ivysaur  Grass  Poison  ...     60           1      False    122
2  3         Venusaur  Grass  Poison  ...     80           1      False    162
3  3  VenusaurMega Venusaur  Grass  Poison  ...     80           1      False    180
4  4       Charmander   Fire     NaN  ...     65           1      False     91
```

Add Column

```
dataset1 = dataset1.drop(columns=["Power"]) #removing Column
print(dataset1.head(5))

   #             Name Type 1  ... Speed  Generation  Legendary
0  1        Bulbasaur  Grass  ...    45           1      False
1  2          Ivysaur  Grass  ...    60           1      False
2  3         Venusaur  Grass  ...    80           1      False
3  3  VenusaurMega Venusaur  Grass  ...    80           1      False
4  4       Charmander   Fire  ...    65           1      False

[5 rows x 12 columns]
```

Remove Column

7 Save Dataset

```
dataset1.to_csv("newDataset.csv")
dataset1.to_excel("newDataset.xlsx")
dataset1.to_csv("newDataset.csv",index=False, sep="\t")
```

csv - Excel - delimiter

8 Checking NaN Values in Dataset

```
dataset1.isna().any()

#             False
Name          False
Type 1        False
Type 2        False
HP            False
Attack        False
```

True - NaN Value | False - No NaN Value

9 Fix NaN Value

```
dataset1 = dataset1[dataset1["Type 2"].notna()]
```

Acquiring Dataset without NaN by removing that row (If String)

```
MeandatasetNotNan = dataset1["Speed"].fillna(dataset1["Speed"].mean())
MeandatasetNotNan
```

Filling NaN values with the Mean/Median of that specific Row (If Numerical Values)

10 Mapping Certain Value to another

```
print(dataset1.tail(5))
Generation = set(dataset1['Generation'])
dataset1['Generation'] = dataset1['Generation'].map({1: "one", 2: "two",3: "three", 4: "four",5: "five", 6: "six"}).astype(str)
print(dataset1.tail(5))

       #             Name  Type 1  ... Speed  Generation  Legendary
795  719         Diancie    Rock  ...    50         six       True
796  719  DiancieMega Diancie    Rock  ...   110         six       True
797  720  HoopaHoopa Confined  Psychic  ...    70         six       True
798  720  HoopaHoopa Unbound  Psychic  ...    80         six       True
799  721         Volcanion    Fire  ...    70         six       True

[5 rows x 12 columns]
       #             Name  Type 1  ... Speed  Generation  Legendary
795  719         Diancie    Rock  ...    50         nan       True
796  719  DiancieMega Diancie    Rock  ...   110         nan       True
797  720  HoopaHoopa Confined  Psychic  ...    70         nan       True
798  720  HoopaHoopa Unbound  Psychic  ...    80         nan       True
799  721         Volcanion    Fire  ...    70         nan       True
```

Here Number 1 is Mapped to "One", likewise for 2,3,4,5 and 6