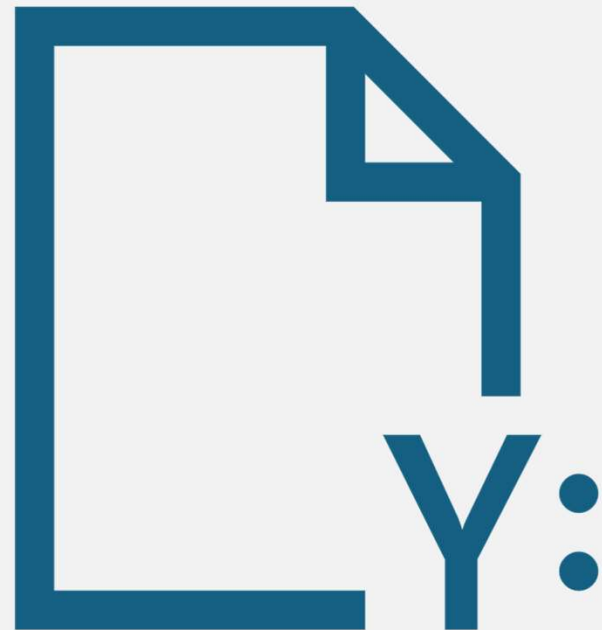


CSV(Comma Separated Values)

- In CSV file each value is separated with a comma
- We can handle csv files using a library called pandas





What is Pandas?

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.



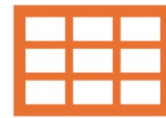
Reading a CSV file with pandas

```
import pandas
```

```
data=pandas.read_csv("weather_data.csv")
```

```
print(data["temp"])
```

Two primary data structures in pandas



Dataframe (like a sheet inside excel)



Series (a column inside a table/sheet)



Converting dataframe to dictionary

We can also convert it into json, pickle and many other formats

```
data.to_dict()
```



Converting series to a list

- `data["temp"].to_list()`

Some useful methods of a series object

mean()

median()

mode()

max()

min()

Get Data in a Row

- `print(data[(data.day=="Monday")])`

Challenge

- Get the data for the day on which highest temperature was recorded

Challenge

- Create a dictionary and convert it into a dataframe, then create a csv file from that dataframe



Convert dictionary to pandas dataframe and then create a csv file

- `my_dict={"Name":["Zeeshan","Fayaz","Ayaz"],
 "Gender":["M","M","M"]}`

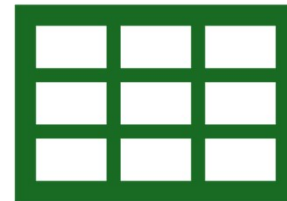
```
data=pandas.DataFrame(my_dict)
```

```
data.to_csv("my_data.csv")
```

Getting a row via Index



`data.loc[0]` //Returns row at 0th
index in dataframe



`data.loc[[0,1]]` //Returns rows
at 0th and 1st index

Print entire dataframe



```
import pandas as pd
```



```
df = pd.read_csv('data.csv')
```



```
print(df.to_string())
```

First 5 and last 5 rows

`data.head()`

`data.tail()`

You can change
the number

Cleaning the data

- **Remove Rows:**

- One way to deal with empty cells is to remove rows that contain empty cells.
- This is usually OK, since data sets can be very big, and removing a few rows will not have a big impact on the result.

```
import pandas as pd

df = pd.read_csv('data.csv')

new_df = df.dropna()

print(new_df.to_string())
```

Cleaning the data

- **Replace Empty values:**

- Another way of dealing with empty cells is to insert a new value instead.
- This way you do not have to delete entire rows just because of some empty cells.
- The fillna() method allows us to replace empty cells with a value:

```
import pandas as pd

df = pd.read_csv('data.csv')

df.fillna(130, inplace = True)
```


Cleaning the data

- **Replace Only For Specified Columns:**
 - The example above replaces all empty cells in the whole Data Frame.
 - To only replace empty values for one column, specify the *column name* for the DataFrame:
-

```
import pandas as pd

df = pd.read_csv('data.csv')

df["Calories"].fillna(130, inplace = True)
```

Cleaning the data

- Replace Using Mean, Median, or Mode

```
import pandas as pd

df = pd.read_csv('data.csv')

x = df["Calories"].median()

df["Calories"].fillna(x, inplace = True)
```

```
import pandas as pd

df = pd.read_csv('data.csv')

x = df["Calories"].mode()[0]

df["Calories"].fillna(x, inplace = True)
```

```
import pandas as pd

df = pd.read_csv('data.csv')

x = df["Calories"].mean()

df["Calories"].fillna(x, inplace = True)
```

Removing Duplicates

```
df.drop_duplicates(inplace = True)
```