1. Read CSV Files

A simple way to store big data sets is to use CSV files (comma separated files).

CSV files contains plain text and is a well know format that can be read by everyone including Pandas.

```
# https://www.w3schools.com/python/pandas/data.csv
```

In [12]:
```
import pandas as pd

df = pd.read_csv('D:\Subject Materials\Python\Pandas\data.csv')
df
#print(df)
#print(df.to_string())
```

Out[12]:

	Duration	Pulse	Maxpulse	Calories
0	60	110	130	409.1
1	60	117	145	479.0
2	60	103	135	340.0
3	45	109	175	282.4
4	45	117	148	406.0
...	...	...	...	...
164	60	105	140	290.8
165	60	110	145	300.0
166	60	115	145	310.2
167	75	120	150	320.4
168	75	125	150	330.4

169 rows × 4 columns

Tip: use to_string() to print the entire DataFrame.

max_rows

The number of rows returned is defined in Pandas option settings.

You can check your system's maximum rows with the pd.options.display.max_rows statement.

In [14]:
```
print(pd.options.display.max_rows)   # Default Values
#print(df)

# In my system the number is 60, which means that if the DataFrame contains more
# the print(df) statement will return only the headers and the first and last 5 r
```

```
60
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
```

In [17]:
```
pd.options.display.max_rows = 200
print(pd.options.display.max_rows)
print(df)
```

```
200
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
```

2. Pandas Read JSON

Big data sets are often stored, or extracted as JSON.

JSON is plain text, but has the format of an object, and is well known in the world of programming, including Pandas.

In our examples we will be using a JSON file called 'data.json'.

https://www.w3schools.com/python/pandas/data.js
(https://www.w3schools.com/python/pandas/data.j

In [18]:
```
df = pd.read_json('https://www.w3schools.com/python/pandas/data.js')

print(df.to_string())
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.5
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       NaN
```

Dictionary as JSON

JSON = Python Dictionary : i.e Elements are of (Key/Value) Pair

JSON objects have the same format as Python dictionaries.

In [19]:
```
data = {
  "Duration":{
    "0":60,
    "1":60,
    "2":60,
    "3":45,
    "4":45,
    "5":60
  },
  "Pulse":{
    "0":110,
    "1":117,
    "2":103,
    "3":109,
    "4":117,
    "5":102
  },
  "Maxpulse":{
    "0":130,
    "1":145,
    "2":135,
    "3":175,
    "4":148,
    "5":127
  },
  "Calories":{
    "0":409,
    "1":479,
    "2":340,
    "3":282,
    "4":406,
    "5":300
  }
}

df = pd.DataFrame(data)

print(df)
```

```
   Duration  Pulse  Maxpulse  Calories
0        60    110       130       409
1        60    117       145       479
2        60    103       135       340
3        45    109       175       282
4        45    117       148       406
5        60    102       127       300
```

3. Pandas Read txt file

Read Text Files with Pandas using read_csv()

In [31]:
```
# read text file into pandas DataFrame
#df = pd.read_csv("D:\Subject Materials\Python\Pandas\DataSet.txt", sep=" ")
#df = pd.read_csv("D:\Subject Materials\Python\Pandas\DataSet.txt", sep=" ", hea
df = pd.read_csv("D:\Subject Materials\Python\Pandas\DataSet.txt", sep=" ", heade
# display DataFrame
print(df)
```

```
     Team1   Team2
0  Batsman  Bowler
1   Sachin  Bumrah
2    Virat   Siraj
3    Rahul   Shami
4    Dhoni  Ashwin
5    Raina  Jadeja
```

Viewing the Data

One of the most used method for getting a quick overview of the DataFrame, is the head() method.

The head() method returns the headers and a specified number of rows, starting from the top.

In [33]:
```
df = pd.read_csv('D:\Subject Materials\Python\Pandas\data.csv')
print(df)
```

```
147         60    112         146      361.9
148         30    103         127      185.0
149         60    110         150      409.4
150         60    106         134      343.0
151         60    109         129      353.2
152         60    109         138      374.0
153         30    150         167      275.8
154         60    105         128      328.0
155         60    111         151      368.5
156         60     97         131      270.4
157         60    100         120      270.4
158         60    114         150      382.8
159         30     80         120      240.9
160         30     85         120      250.4
161         45     90         130      260.4
162         45     95         130      270.0
163         45    100         140      280.9
164         60    105         140      290.8
165         60    110         145      300.0
166         60    115         145      310.2
```

In [40]:
```
print(df.head())  # By Default only 5 Records are Listed
print(df.head(10)) # Now The first 10 Records are Listed
```

```
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0
5        60    102       127     300.0
6        60    110       136     374.0
7        45    104       134     253.3
8        30    109       133     195.1
9        60     98       124     269.0
```

In [42]:
```
print(df.tail())  # By Default only Last 5 Records are Listed
print(df.tail(10)) # Now The Last 10 Records are Listed
```

```
     Duration  Pulse  Maxpulse  Calories
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
     Duration  Pulse  Maxpulse  Calories
159        30     80       120     240.9
160        30     85       120     250.4
161        45     90       130     260.4
162        45     95       130     270.0
163        45    100       140     280.9
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
```

Info About the Data

The DataFrames object has a method called info(), that gives you more information about the data set.

In [47]:
```
df = pd.read_csv('D:\Subject Materials\Python\Pandas\data.csv')
print(df.info())

print("\n")

df = pd.read_csv('D:\Subject Materials\Python\Pandas\data.csv')
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Duration  169 non-null    int64  
 1   Pulse     169 non-null    int64  
 2   Maxpulse  169 non-null    int64  
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Duration  169 non-null    int64  
 1   Pulse     169 non-null    int64  
 2   Maxpulse  169 non-null    int64  
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 5.4 KB
None
```