What is Pandas?

Pandas is a python library used to analyze data. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories.Pandas can clean messy data sets, and make them readable and relevant. While workin on a data science or ML project we need a clean data

Importing Pandas

In [1]:
```
import pandas as pd    # Here pandas is the name of library and pd is its alias name, n
```

Checking version of Pandas

In [2]:
```
print(pd.__version__)
```

```
1.2.4
```

Series

A Pandas Series is like a column in a table. It is a one-dimensional array holding data of any type.

In [5]:
```
# Create a simple Pandas Series from a list:
a = [1, 2, 3]

var = pd.Series(a)

print(var)
```

```
0    1
1    2
2    3
dtype: int64
```

Labels

If nothing else is specified, the values are labeled with their index number. First value has index 0, second value has index 1 etc. This label can be used to access a specified value.

In [6]:
```
# Return the first value of the Series:
print(var[0])
```

```
1
```

Creating Labels

With the index argument, you can name your own labels.

In [8]:
```
a = [1, 2, 3]
var = pd.Series(a, index = ["x", "y", "z"])
print(var)
```

```
x    1
y    2
```

```
z    3
dtype: int64
```

In [9]:
```
# When you have created labels, you can access an item by referring to the label.
print(var["y"])
```

```
2
```

Key/Value Objects as Series

In [11]:
```
# You can also use a key/value object, like a dictionary, when creating a Series.
# Create a simple Pandas Series from a dictionary:
```

In [16]:
```
calories = {"Day_1": 300, "Day_2": 320, "Day_3": 340, "Day_4": 360, "Day_5": 380, "Day_
var= pd.Series(calories)
print(var)
# Note: The keys of the dictionary become the labels.
```

```
Day_1    300
Day_2    320
Day_3    340
Day_4    360
Day_5    380
Day_6    400
dtype: int64
```

In [17]:
```
# To select only some of the items in the dictionary, use the index argument and specif
```

In [20]:
```
# Create a Series using only data from "day1" and "day2":
var = pd.Series(calories, index=["Day_1", "Day_2"])
print(var)
```

```
Day_1    300
Day_2    320
dtype: int64
```

DataFrames

Data sets in Pandas are usually multi-dimensional tables, called DataFrames.Series is like a column, a DataFrame is the whole table, where DataFrame is both rows and column.

In [22]:
```
# Create a DataFrame from two Series:
data = {
    "calories" :[300, 320, 340],
    "duration" :[50, 55, 60]
}
var = pd.DataFrame(data)
print(var)
```

```
   calories  duration
0       300        50
1       320        55
2       340        60
```

In [24]:
```
df=pd.DataFrame(data)
print(df)
```

```
   calories  duration
0       300        50
```

```
1       320        55
2       340        60
```

Locate Row

Pandas use the loc attribute to return one or more specified row(s)

In [25]:
```
print(df.loc[0])
```

```
calories    300
duration     50
Name: 0, dtype: int64
```

In [28]:
```
# Return row 0 and 1:
print(df.loc[[0, 1]])
```

```
   calories  duration
0       300        50
1       320        55
```

Index Naming

With the index argument, you can name your own indexes.

In [30]:
```
df= pd.DataFrame(data, index=["Day_1", "Day_2", "Day_3"])
print(df)
```

```
       calories  duration
Day_1       300        50
Day_2       320        55
Day_3       340        60
```