

## Concept 1.3: Pandas - So much more than a cute animal

- Introducing Pandas data structures: Series, DataFrames and Index objects.

# convention for importing pandas

```
import pandas as pd
days = pd.Series(['Monday', 'Tuesday', 'Wednesday'])
print(days) # prints
```

```
0      Monday
1      Tuesday
2     Wednesday
dtype: object
```

# creating series with a numpy array

```
days_list = np.array(['Monday', 'Tuesday', 'Wednesday'])
numpy_days = pd.Series(days_list)
print(numpy_days) # prints
0      Monday
1      Tuesday
2     Wednesday
dtype: object
```

# using strings as index

```
days = pd.Series(['Monday', 'Tuesday', 'Wednesday'],
                  index=['a', 'b', 'c'])
```

# create series from a dictionary

```
days1 = pd.Series({'a': 'Monday', 'b': 'Tuesday', 'c': 'Wednesday'})
```

# days and days1 prints this

```
a      Monday
b      Tuesday
c     Wednesday
dtype: object
```

Series can be accessed using the specified index as shown below

```
days[0] = Monday
```

```
days[1:] = b      Tuesday
           c      Wednesday
dtype: object
```

```
days['c'] = Wednesday
```

A DataFrame can be described as a table (2 dimensions) made up of many series with the same index..

```
print(pd.DataFrame())
# prints an empty dataframe
Empty DataFrame
Columns: []
Index: []

# create a dataframe from a dictionary
df_dict = {'Country': ['Ghana', 'Kenya', 'Nigeria', 'Togo'],
           'Capital': ['Accra', 'Nairobi', 'Abuja', 'Lome'],
           'Population': [10000, 8500, 35000, 12000],
           'Age': [60, 70, 80, 75]}
df = pd.DataFrame(df_dict, index=[2, 4, 6, 8])

df_list = [['Ghana', 'Accra', 10000, 60],
           ['Kenya', 'Nairobi', 8500, 70],
           ['Nigeria', 'Abuja', 35000, 80],
           ['Togo', 'Lome', 12000, 75]]
df1 = pd.DataFrame(df_list, columns=['Country', 'Capital', 'Population', 'Age'],
                  index=[2, 4, 6, 8])

# df and df1 prints this
   Country  Capital  Population  Age
2   Ghana    Accra    10000    60
4   Kenya  Nairobi    8500    70
6   Nigeria   Abuja   35000    80
8    Togo     Lome    12000    75

# select the row in the at index 3
df.iloc[3] = Country      Togo
              Capital      Lome
              Population  12000
              Age         75
              Name: 8, dtype: object
# select row with index label 6
df.loc[6] = Country      Nigeria
            Capital      Abuja
            Population  35000
            Age         80
            Name: 6, dtype: object
# select the Capital column
```

```

df['Capital'] = 2      Accra
                  4      Nairobi
                  6      Abuja
                  8      Lome
                  Name: Capital, dtype: object
df.at[6, 'Country'] = Nigeria
df.iat[2, 0] = Nigeria

```

The describe function gives the summary of the numeric columns in a dataframe displaying count, mean, standard deviation, interquartile range, minimum and maximum values.

```

df['Population'].sum() = 65500
df.mean() = Population    16375.00
              Age         71.25
              dtype: float64
df.describe() =

```

	Population	Age
count	4.0	4.0
mean	16375.00	71.3
std	12499.2	8.5
min	8500.0	60.0
25%	9625.0	67.5
50%	11000.0	72.5
75%	17750.0	76.3
max	35000.0	80.0

- The missing data enigma: Importance, types and handling missing data.