Section (2)
Data Mining

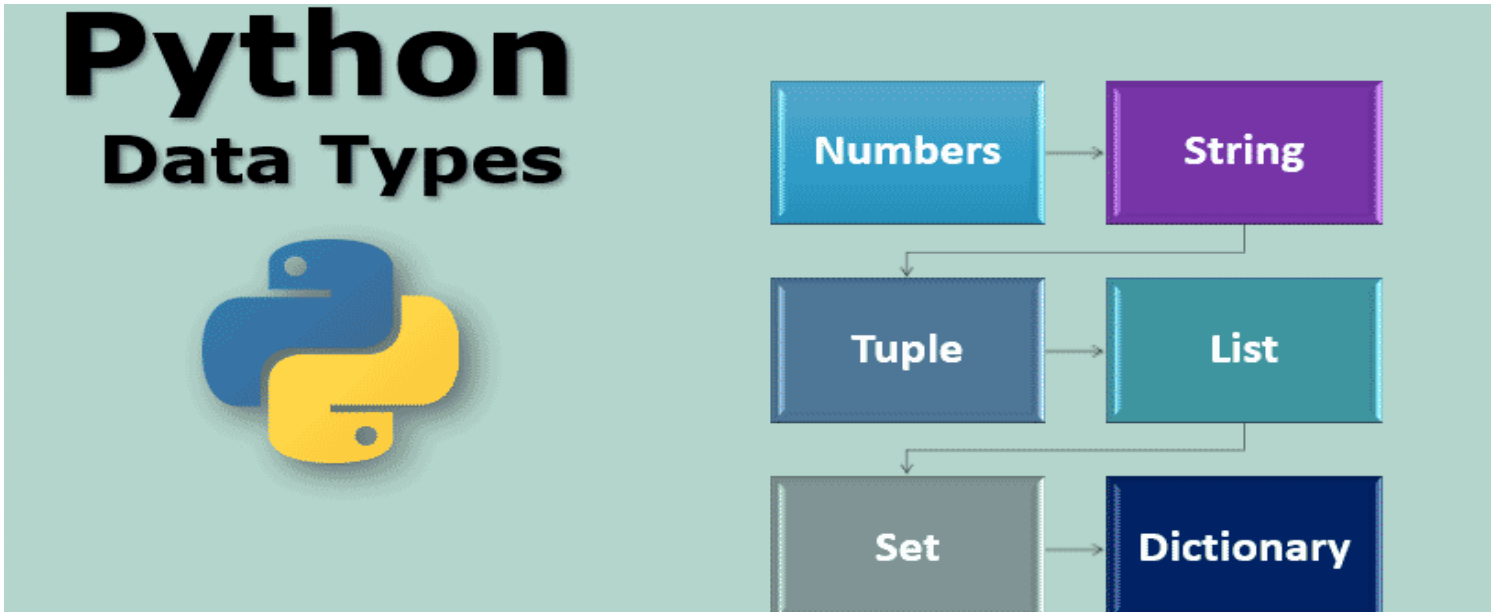
Nagwa El-Araby - Mohammed El-Barbeer - Aml Magdy
Information System Department

2019 - 2020

PYTHON FOR DATA SCIENCE

In python, there are many data structures available. They are:

- strings
- Lists
- dictionaries
- sets



1-STRINGS

- Python string is a sequence of characters
- How to create string in python :

You can create Python string using a **single or double quote**.

```
mystring = "Hello Python3.6"
print(mystring)
```

Output:

```
Hello Python3.6
```

How to extract Nth letter or word?

You can use the syntax below to get first letter.

```
mystring = 'Hi How are you?'
mystring[0]
```

Output

```
'H'
```

mystring[0] *refers to first letter as indexing in python starts from 0. Similarly, mystring[1] refers to second letter.*

1-STRINGS(CONT.)

To get first word

```
mystring.split(' ')[0]
```

Output : Hi

How it works -

1. mystring.split(' ') tells Python to use space as a delimiter.

Output : ['Hi', 'How', 'are', 'you?']

2. mystring.split(' ')[0] tells Python to pick first word of a string.

2- LIST

- Unlike String, List can contain different types of objects such as integer, float, string etc.

x = [142, 124, 234, 345, 465]
y = [‘A’, ‘C’, ‘E’, ‘M’]
z = [‘AA’, 44, 5.1, ‘KK’]

We can extract list item using Indexes. Index starts from 0 and end with (number of elements-1).

- Example :

k = [124, 225, 305, 246, 259]

k[0]: 124
k[-1]: 259

Explanation :
k[0] picks first element from list.
Negative sign tells Python to search list item from right to left.
k[-1] selects the last element from list.

2- LIST (CONT.)

Combine / Join two lists

The '+' operator is concatenating two lists.

```
X = [1, 2, 3]
Y = [4, 5, 6]
Z = X + Y
print(Z)
```

```
[1, 2, 3, 4, 5, 6]
```

Sum of values of two list

```
X = [1, 2, 3]
Y = [4, 5, 6]
import numpy as np
Z = np.add(X, Y)
print(Z)
```

```
print(Z)
[5 7 9]
```

2- LIST (CONT.)

Modify / Replace a list item

Suppose you need to replace third value to a different value.

```
X = [1, 2, 3]
X[2]=5
print(X)
```

```
print(X)
[1, 2, 5]
```

We can add a list item by using **append method**.

```
X = ['AA', 'BB', 'CC']
X.append('DD')
print(X)
```

Result : ['AA', 'BB', 'CC', 'DD']

Similarly, we can remove a list item by using **remove method**.

```
X = ['AA', 'BB', 'CC']
X.remove('BB')
print(X)
```

Result : ['AA', 'CC']

3- DICTIONARIES

- It works like an address book wherein you can find an address of a person by searching the name.
- In this example. name of a person is considered as **key** and address as **value**.
- It is important to note that the key must be unique while values may not be.
- Keys should not be duplicate because if it is a duplicate, you cannot find exact values associated with key.

List

index	value
0	"Eggs"
1	"Milk"
2	"Cheese"
3	"Yogurt"
4	"Butter"
5	"More Cheese"

Dictionary

key	value
'Eggs'	2.59
'Milk'	3.19
'Cheese'	4.80
'Yogurt'	1.35
'Butter'	2.59
'More Cheese'	6.19

3- DICTIONARIES (CONT.)

Create a dictionary

It is defined in curly braces {}. Each key is followed by a colon (:)

```
teams = {'Dave' : 'team A',
         'Tim' : 'team B',
         'Babita' : 'team C',
         'Sam' : 'team B',
         'Ravi' : 'team C'
        }
```

Find Values

```
teams['Sam']
```

Output : 'team B'

Delete an item

```
del teams['Ravi']
```

4- SETS

- A set in Python holds a sequence of values. It is sequenced but does not support indexing.

Sets are unordered collections of simple objects.

```
X = set(['A', 'B', 'C'])
```

Q. Does 'A' exist in set X?

```
'A' in X
```

Result : True

Q. How to add 'D' in set X?

```
X.add('D')
```

Q. How to remove 'C' from set X?

```
X.remove('C')
```

Q. How to create a copy of set X?

```
Y = X.copy()
```

Q. Which items are common in both sets X and Y?

```
Y & X
```

Popular python packages for Data Analysis & Visualization

1. **Pandas**. provides high-performance, easy-to-use data structures and data analysis tools
2. **NumPy**. For numerical computing. It allows us to do some operations on an entire column or table in one line.
3. **Scipy**. For mathematical and scientific functions such as integration, interpolation, signal processing, etc.
4. **Scikit-learn**. A collection of machine learning algorithms.
5. **Matplotlib**. For data visualization. It's a leading package for graphics in Python.
6. **Statsmodels**. For statistical and predictive modeling. It allows users to impute missing values, statistical tests and take table output to HTML format.
7. **Pandasql**. It allows SQL users to write SQL queries in Python. It is very helpful for people who loves writing SQL queries to manipulate data.

Python Packages Commands

Install Package

```
!pip install pandas
```

Uninstall Package

```
!pip uninstall pandas
```

Show Information about Installed Package

```
!pip show pandas
```

List of Installed Packages

```
!pip list
```

Upgrade a package

```
!pip install --upgrade pandas
```

How to import a package

1. import pandas as pd

It imports the package pandas under the alias pd. A function DataFrame in package pandas is then submitted with pd.DataFrame.

2. import pandas

It imports the package without using alias but here the function **DataFrame** is submitted with full package name **pandas.DataFrame**

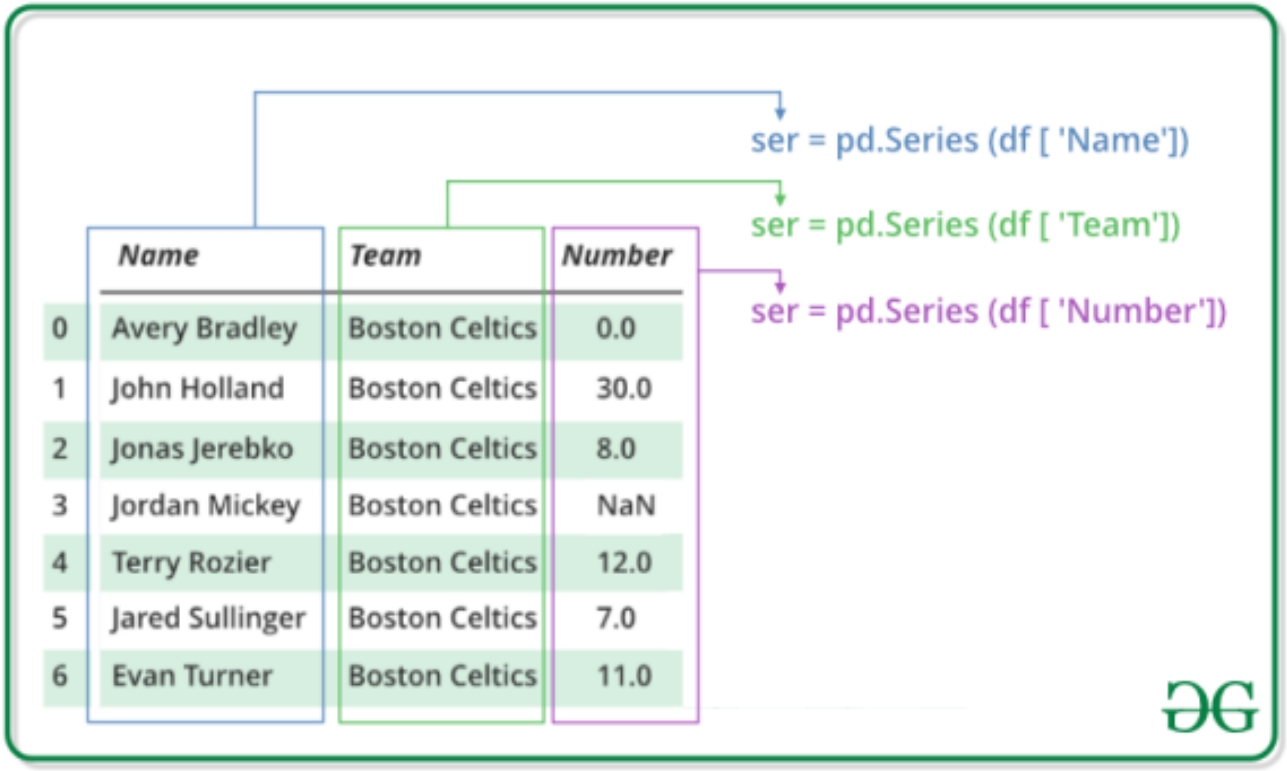
3. from pandas import *

It imports the whole package and the function **DataFrame** is executed simply by typing **DataFrame.** It sometimes creates confusion when same function name exists in more than one package.

Pandas Data Structures: Series and Data-Frame

In pandas package, there are two data structures: series and data frame.

1. Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, python objects, etc.). The axis labels are collectively called index. Pandas Series is nothing but a column in an excel sheet.



	Name	Team	Number
0	Avery Bradley	Boston Celtics	0.0
1	John Holland	Boston Celtics	30.0
2	Jonas Jerebko	Boston Celtics	8.0
3	Jordan Mickey	Boston Celtics	NaN
4	Terry Rozier	Boston Celtics	12.0
5	Jared Sullinger	Boston Celtics	7.0
6	Evan Turner	Boston Celtics	11.0

Pandas Data Structures: Series and Data-Frame (cont.)

Creating a series from array: In order to create a series from array, we have to import a numpy module and have to use array() function.

```
import pandas as pd
import numpy as np
data = np.array(['g','e','e','k','s'])
ser = pd.Series(data)
print(ser)
```

Output:

```
0    g
1    e
2    e
3    k
4    s
dtype: object
```

Pandas Data Structures: Series and Data-Frame(cont.)

You can get a particular element of a series using index value. See the

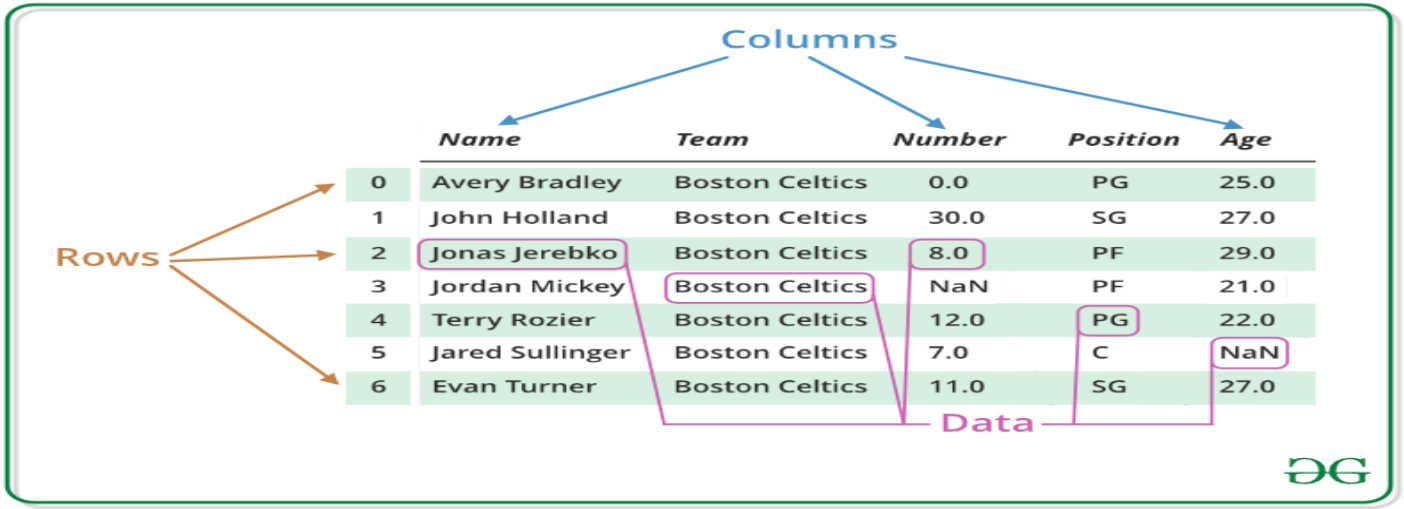
```
s1[0]
```

```
g
```

```
s1[:3]
```

```
0    g
1    e
2    e
dtype: object
```

2- Data Frame: It is a 2-dimensional data structure that can store data of different data types such as characters, integers, floating point values. You can think of data frame as Excel Spreadsheet.



	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Data Frame Example :

```
import pandas as pd
df = pd.DataFrame({
'col1' : ['A', 'B', 'D', 'E', 'F', 'C'],
'col2' : [2, 1, 9, 8, 7, 4],
'col3': [0, 1, 9, 4, 2, 3],
})
```

Output

	col1	col2	col3
0	A	2	0
1	B	1	1
2	D	9	9
3	E	8	4
4	F	7	2
5	C	4	3

Important Pandas Functions

Functions	Python (pandas)
Installing a package	!pip install name
Loading a package	import name as other_name
Checking working directory	import os os.getcwd()
Setting working directory	os.chdir()
List files in a directory	os.listdir()
Remove an object	del object

Important Pandas Functions (cont.)

Functions	Python (pandas)
Drop Variables	df.drop(['x1', 'x2'], axis = 1)
Filter Data	df.query('x1 >= 100')
Structure of a DataFrame	df.info()
Summarize dataframe	df.describe()
Get row names of dataframe "df"	df.index
Get column names	df.columns
View Top N rows	df.head(N)
View Bottom N rows	df.tail(N)

Important Pandas Functions (cont.)

Functions	Python (pandas)
Get dimension of data frame	df.shape
Get number of rows	df.shape[0]
Get number of columns	df.shape[1]
Length of data frame	len(df)
Get random 3 rows from dataframe	df.sample(n=3)
Get random 10% rows	df.sample(frac=0.1)
Check Missing Values	pd.isnull(df.x)
Sorting	df.sort_values(['x1', 'x2'])
Rename Variables	df.rename(columns={'x1': 'newvar'})

Assignment (1)

- For the following code :

```
df = pd.DataFrame({
'col1' : ['A', 'B', 'D', 'E', 'F', 'C'],
'col2' : [2, 1, 9, 8, 7, 4],
'col3': [0, 1, 9, 4, 2, 3],
})
```

- Get number of rows
- Get random 50% rows
- Sort by 'col1' , DESC
- Filter data to get obtain result (col2 >=7)
- Get the oldest 3 rows

Assignment Solution

- Assignment Solution :

```
import pandas as pd
df = pd.DataFrame({
'col1' : ['A', 'B', 'D', 'E', 'F', 'C'],
'col2' : [2, 1, 9, 8, 7, 4],
'col3': [0, 1, 9, 4, 2, 3],
})

print(df.shape[0])
print(df.sample(frac=0.5))
print(df.sort_values(['col1'], ascending=False))
print(df.query('col2 >= 7'))
print(df.head(3))
```


Thank You