3.4 Python Libraries for Consuming Data

3.4 Python Libraries for Consuming Data

Numpy

- NumPy is a Python library used for working with arrays.
- It also has functions for working in the domain of linear algebra, Fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant.
- It is an open-source project and you can use it freely.
- NumPy stands for Numerical Python.
- https://github.com/numpy/numpy.

Why Use Numpy

- In Python, we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

Why is NumPy Faster Than Lists?

- NumPy arrays are stored at one continuous place in memory, unlike lists.
  - So processes can access and manipulate them very efficiently.
- This behaviour is called the locality of reference in computer science.
- This is the main reason why NumPy is faster than lists.
- Also, it is optimized to work with the latest CPU architectures.
- NumPy is a Python library and is written partially in Python.
  - Most of the parts that require fast computation are written in C or C++.

Import Numpy

Install NumPy:
conda install numpy
pip install numpy

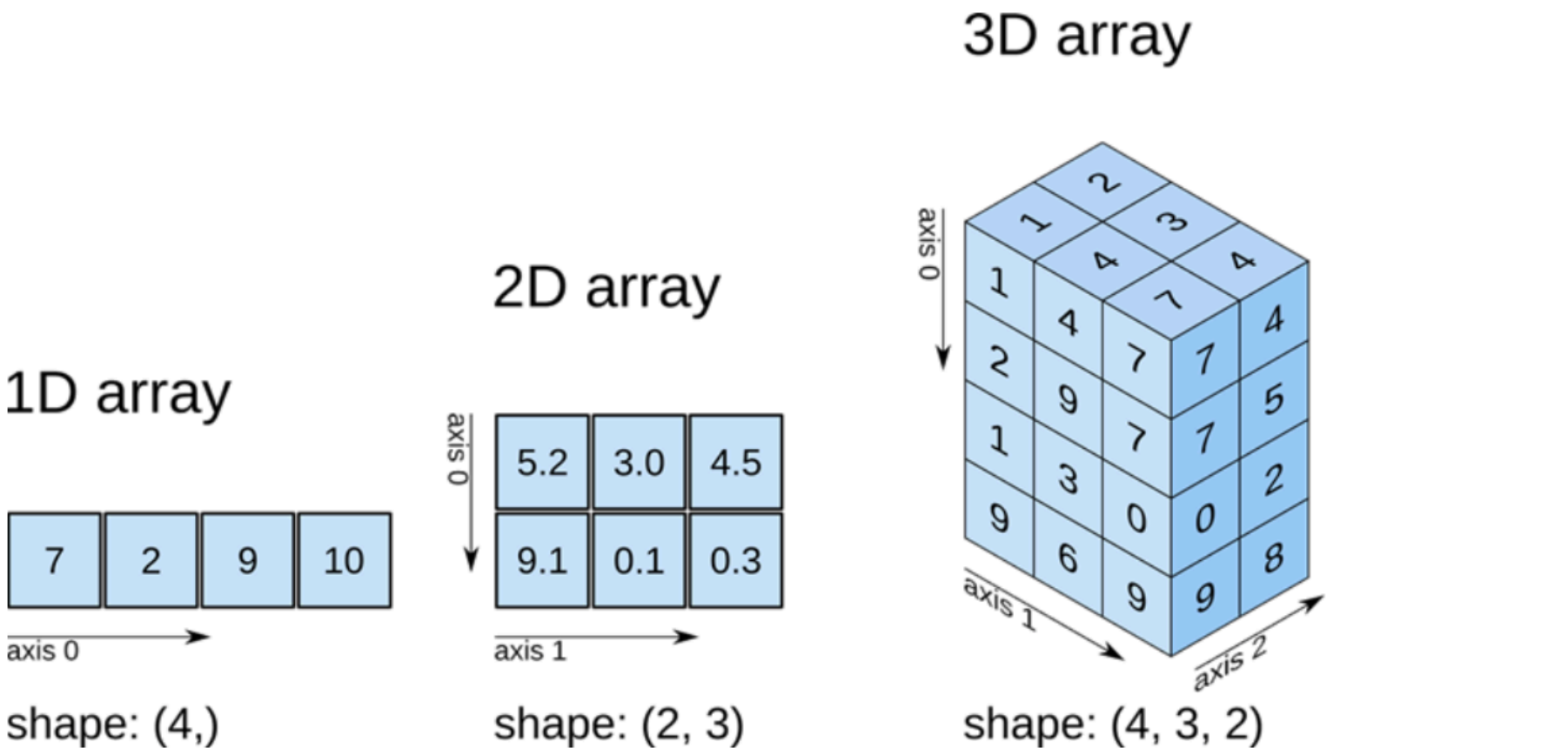
Import: import numpy as np
Check Version: print(np.__version__)

Create Numpy Array

- type()
  - This built-in Python function tells us the type of the object passed to it.
- Like in the above code it shows that arr is numpy.ndarray type.
- An ndarray is a (usually fixed-size) multidimensional container of items of the same type and size.
- The number of dimensions and items in an array is defined by its shape, which is a tuple of N non-negative integers that specify the sizes of each dimension.

- The type of items in the array is specified by a separate data-type object (dtype), one of which is associated with each ndarray.

x = np.array([[1, 2, 3], [4, 5, 6]], np.int32)



Accessing Numpy Array

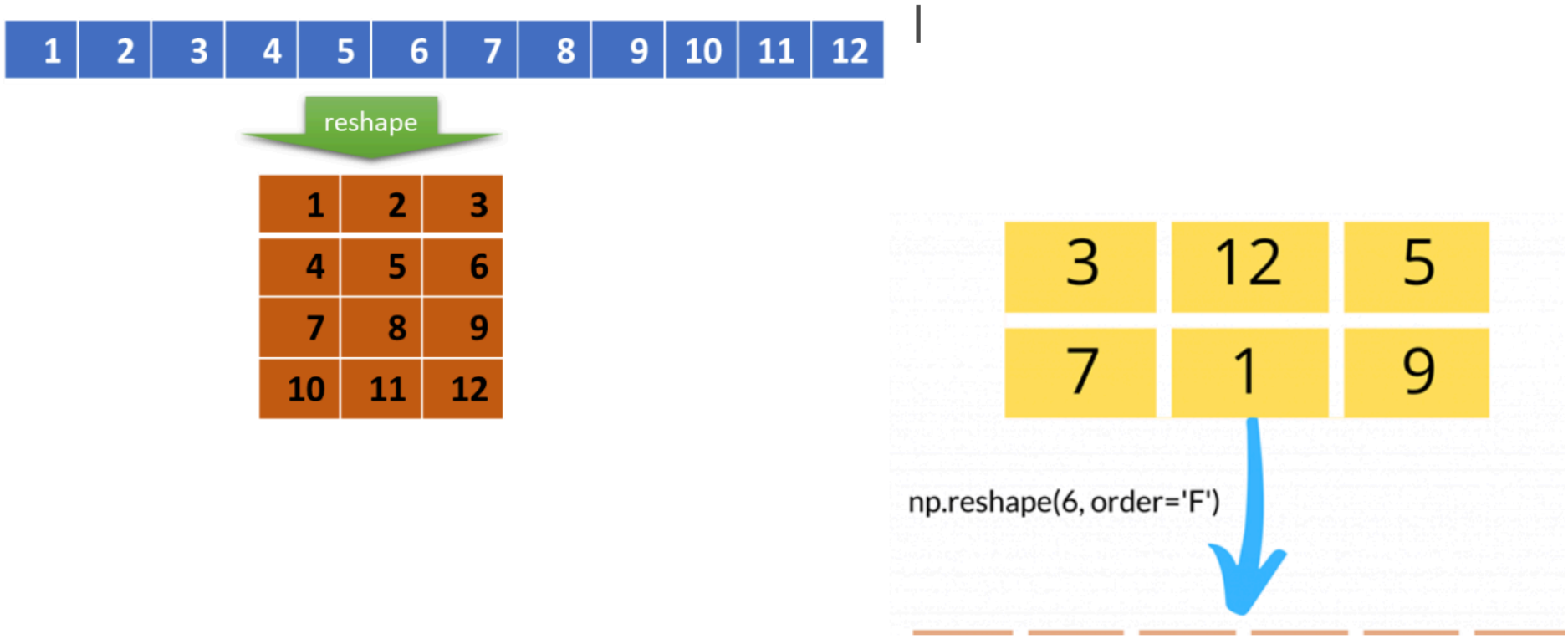
- The indexes in NumPy arrays start with 0, meaning that the first element has index 0, and the second has index 1 etc.

arr = np.array([1, 2, 3, 4, 5])

- index → 0, 1, 2, 3, 4

Reshape ndarray

arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])
newarr = arr.reshape(4, 3)



Slicing

- Slicing in python means taking elements from one given index to another given index.
  - We pass slice instead of index like this: [start:end].
  - We can also define the step, like this: [start:end:step].
  - If we don't pass start its considered 0
  - If we don't pass end its considered length of array in that dimension

- If we don't pass step its considered 1

```
arr = np.array([1, 2, 3, 4, 5, 6, 7])
print(arr[1:5])
```

```
>> [2 3 4 5]
```

- The result includes the start index, but excludes the end index.

print(a[0,1:4])

	0	1	2	3	4
0	11	12	13	14	15
1	16	17	18	18	20
2	21	22	23	24	25
3	26	27	28	29	30
4	31	32	33	34	35

- *print*(a[0,1:4])

- *print*(a[1:4,0])

	0	1	2	3	4
0	11	12	13	14	15
1	16	17	18	18	20
2	21	22	23	24	25
3	26	27	28	29	30
4	31	32	33	34	35

print(a[::2,::2])

	0	1	2	3	4
0	11	12	13	14	15
1	16	17	18	18	20
2	21	22	23	24	25
3	26	27	28	29	30
4	31	32	33	34	35

print(a[:,1])

	0	1	2	3	4
0	11	12	13	14	15
1	16	17	18	18	20
2	21	22	23	24	25
3	26	27	28	29	30
4	31	32	33	34	35

Search, Sort & Filter

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.
- https://github.com/pandas-dev/pandas

Import Pandas

Install pandas:
conda install pandas
pip install pandas

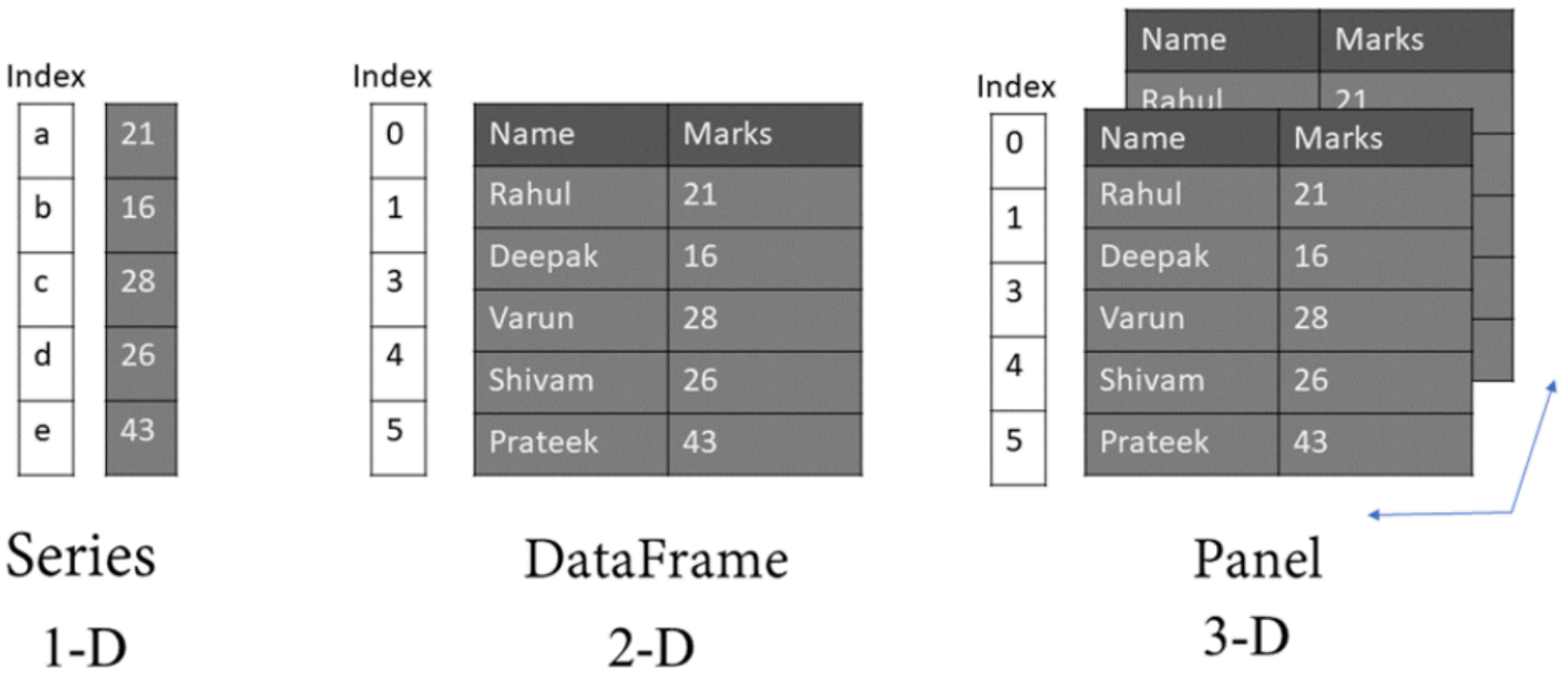
Import: import pandas as pd

Check Version: print(pd.__version__)

Check version: `print(pd.__version__)`

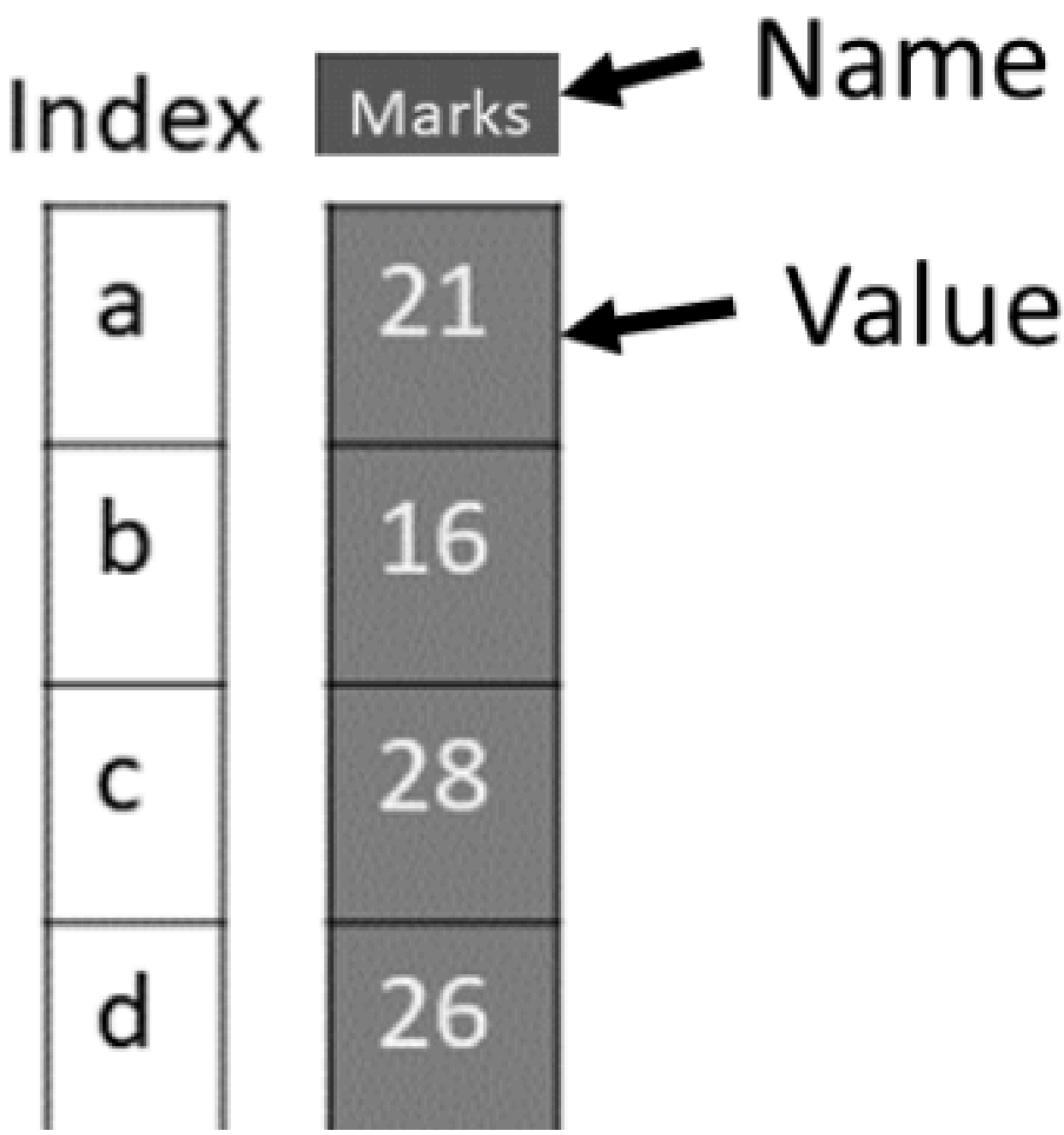
Data Structures in Pandas

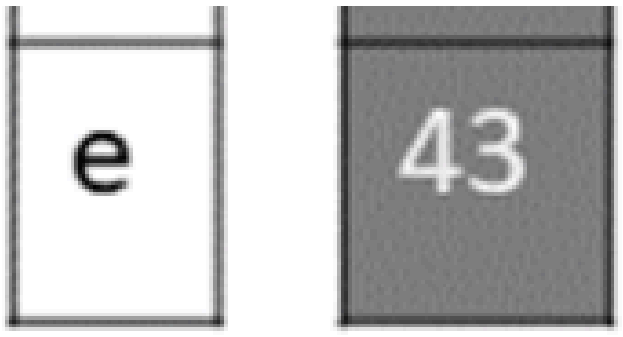
Series — 1D
DataFrame — 2D
Panel — 3D



Series

- The Series is the object of the pandas library designed to represent one-dimensional data structures.
  - Similar to an array but with some additional features.
- A series consists of two components.
  - One-dimensional data (Values)
  - Index





Working with Series

DataFrame

A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

DataFrame Operations

Operation	Syntax	Result
Select column	df[col]	Series
Select row by label	df.loc[label]	Series
Select row by integer location	df.iloc[loc]	Series
Slice rows	df[5:10]	Dataframe
Select rows by boolean vector	df[bool_vec]	Dataframe

Reading From CSV File

- A simple way to store big data sets is to use CSV files (comma separated files).
- CSV files contain plain text and are a well know format that can be read by everyone including Pandas.
- In our examples, we will be using a CSV file called data.csv.

```
import pandas as pd
df = pd.read_csv('data.csv')
print(df.to_string())
```

Analyzing DataFrames

- 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.

```
print(df.head(10))
```

Pandas - Cleaning Data

- Data cleaning means fixing bad data in your data set.
- Bad data could be

- Bad data could be:
  - Empty cells
  - Data in wrong format
  - Wrong data
  - Duplicates

Handling Empty Cells

Empty cells can potentially give you a wrong result when you analyse data.

1. Remove Rows with empty cells: dropna()

- By default, the dropna() method returns a new DataFrame, and will not change the original.
  - If you want to change the original DataFrame, use the inplace = True argument:

2. Replace Empty Values: fillna()

- 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.

id	module	marks
123X	Science	90
567V	Maths	NaN
890M	Arts	NaN

dropna()

id	module	marks
123X	Science	90

id	module	marks
123X	Science	90
567V	Maths	NaN
890M	Arts	NaN

fillna(88)

| id | module | marks |

123X	Science	90
567V	Maths	88
890M	Arts	88

Fixing Data

- Suppose we have a dataFrame mentioned in slide 24.
- Let's say that the duration cannot be more than 120.
- Then we can fix it by setting any duration value higher than 120 to 120.

```
for x in df.index:
  if df.loc[x, "Duration"] > 120:
    df.loc[x, "Duration"] = 120
```

Handle Duplicates

- Duplicate rows are rows that have been registered more than one time.

10	60	'2020/12/11'	103	147	329.3
11	60	'2020/12/12'	100	120	250.7
12	60	'2020/12/12'	100	120	250.7
13	60	'2020/12/13'	106	128	345.3

- To discover duplicates, we can use the duplicated() method.
- The duplicated() method returns a Boolean values for each row:

```
print(df.duplicated())
```

- To remove duplicates, use the drop_duplicates() method.

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

◄ 3.3 Quiz

Jump to...

GET IN TOUCH

University of Moratuwa
Centre for Open & Distance Learning
CODL

011 308 2787/8

011 265 0301 ext. 3850,3851

open@uom.lk

University Website

CODL Website

Data retention summary