Data Science for Managerial Decisions (MB 511)
A Short Course in Data Science using Python

Instructor
Anant Prakash Awasthi

Course

- <mark>Getting and setting</mark>
- <mark>A Quick Introduction</mark>
- <mark>Industry Products written in Python</mark>

- <mark>Data Types & Structure</mark>
- Loops and Decision Making
- Functions & Libraries
- Error Handling
- Working with Operating System and Dates
- Data Exchange
- Data Management
- Data Visualization
- Feature Engineering
- Quick Introduction to AI
- Model Development and Validation
- Model Deployment

- Case Study 1 – Data Management
- Case Study 2 - Bank Marketing
- Case Study 3 - Student Performance
- Case Study 4 - AI4I 2020 Predictive Maintenance

EMBA Program
MB-511

Getting and setting

Python from Python Website

VS Code

Managed Python Distribution

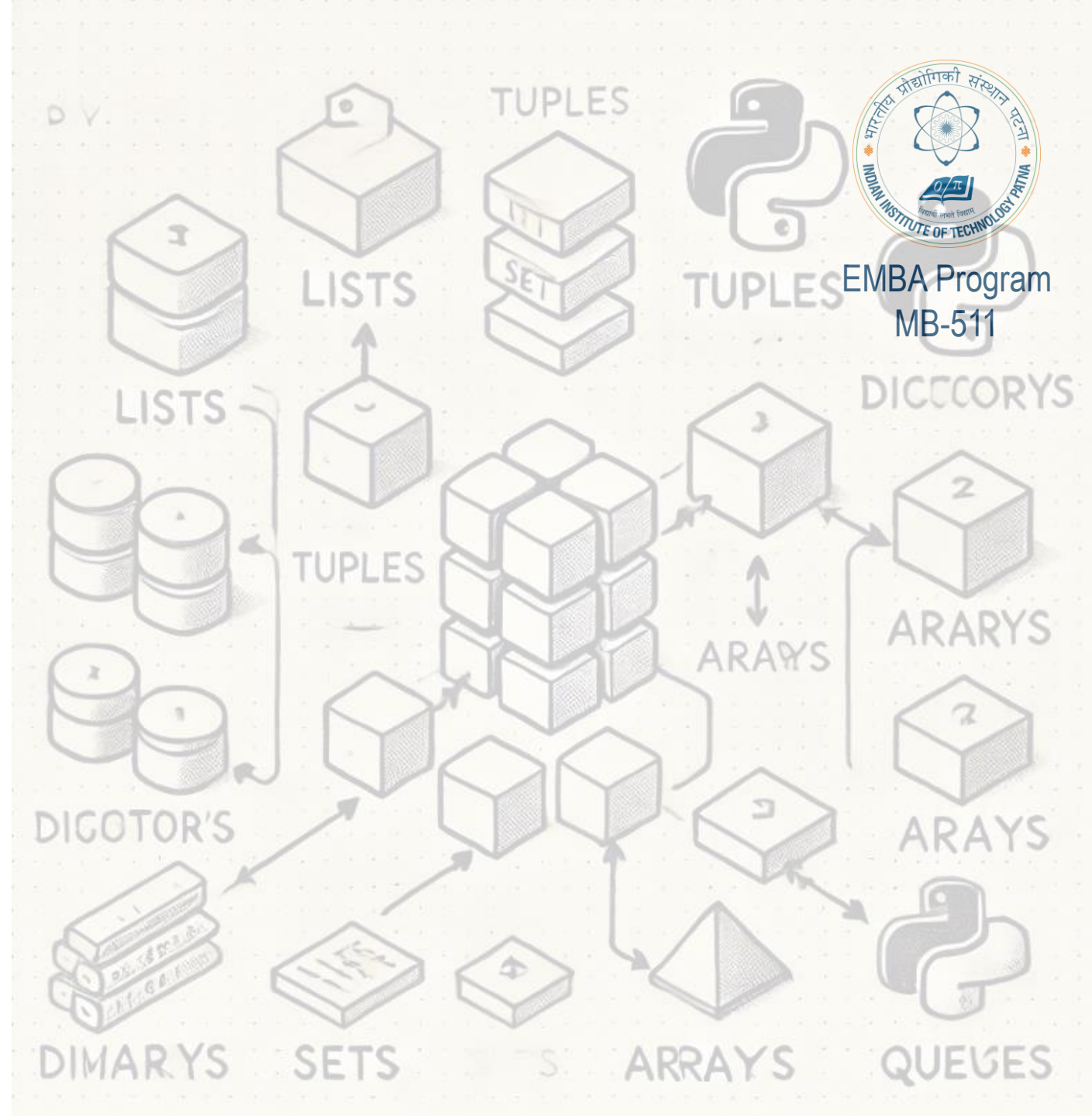## Collaboration and Version Control
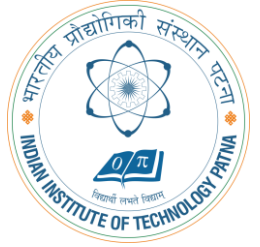
Data Structures in Python

Introduction to Lists

A list in Python is a versatile data structure that allows you to store a collection of items in an ordered sequence. Lists can contain elements of any data type (integers, strings, floats, objects), and the elements can be changed (i.e., lists are mutable). Lists are defined using square brackets [] and elements are separated by commas.

Key Features:

- Ordered: Items have a defined order and can be accessed by their index.
- Mutable: You can change, add, or remove elements after the list has been created.
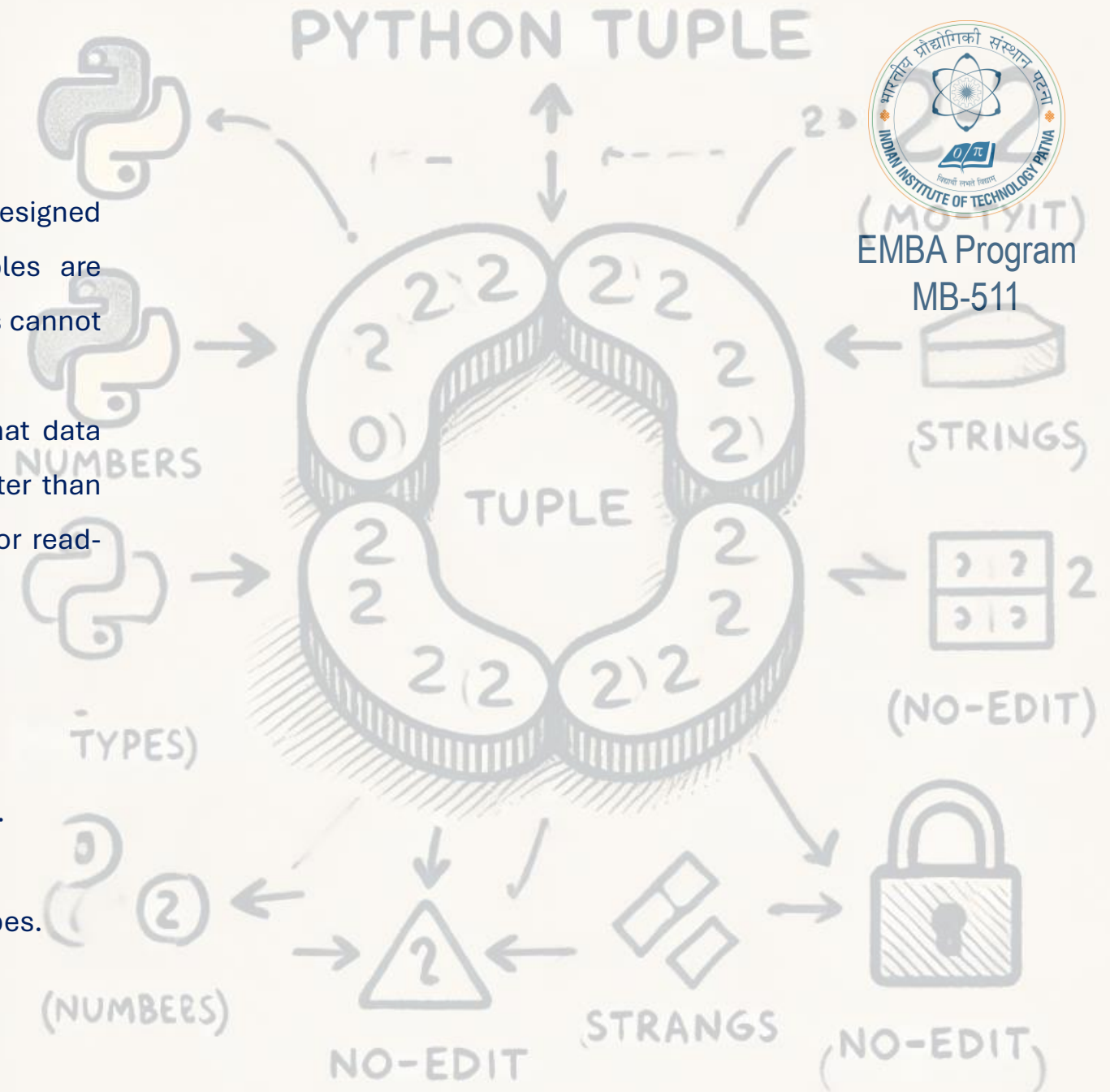- Allow Duplicates: Lists can contain multiple items with the same value.
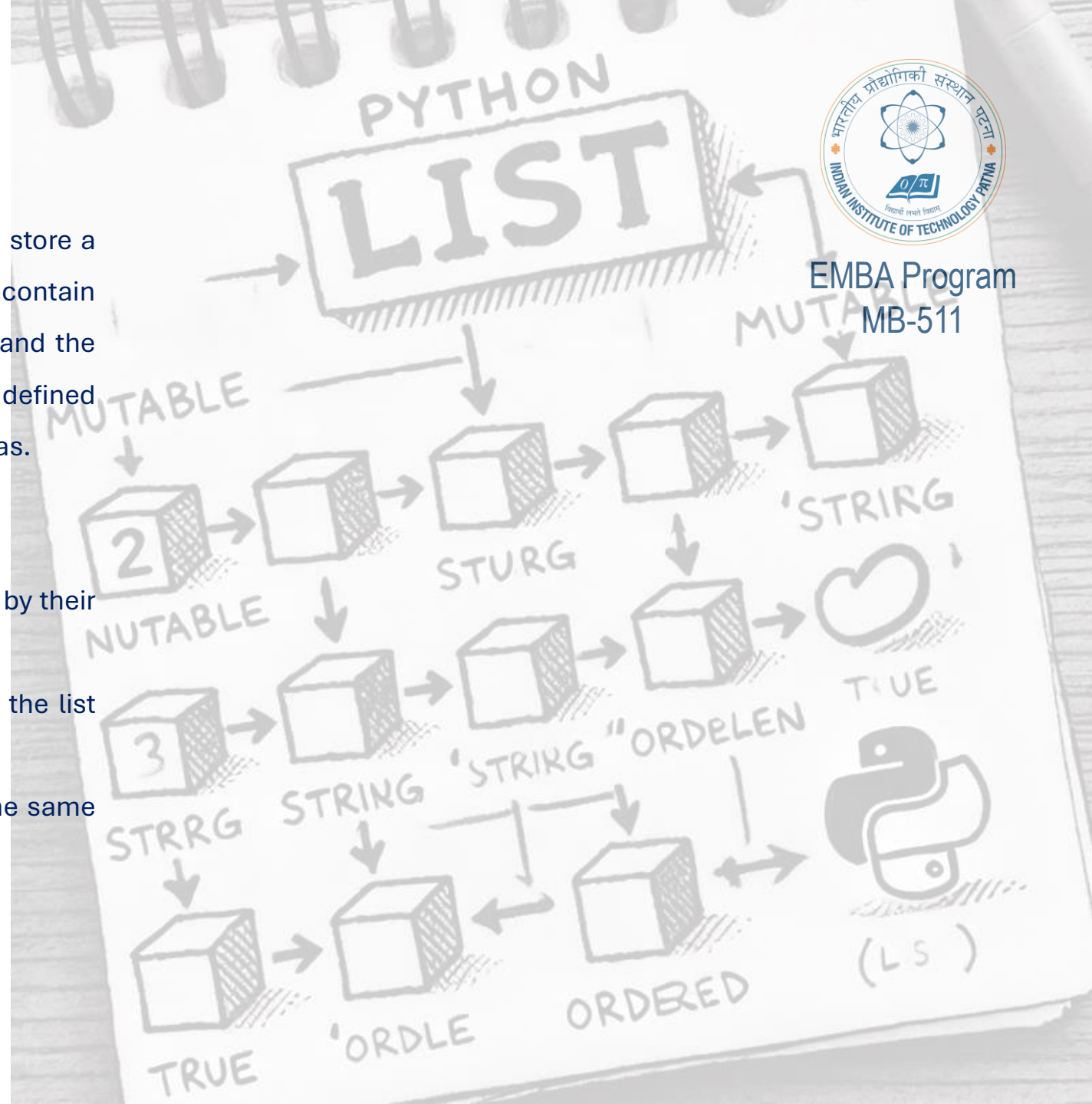
Introduction to Tuples

A tuple is one of the fundamental data structures in Python, designed to hold an ordered collection of items. Unlike lists, tuples are immutable, meaning that once a tuple is created, its elements cannot be changed, added, or removed.

Tuples are useful in scenarios where you want to ensure that data remains constant throughout the program. They are also faster than lists due to their immutability, making them a good choice for read-only data.

Key Features:

- Immutable: Once created, elements cannot be changed.

- Ordered: Elements are stored in the order they are inserted.

- Allow Duplicates: Tuples can contain duplicate values.

- Heterogeneous: Tuples can store items of different data types.

Introduction to Sets

A set in Python is a collection of unique elements. Unlike lists or tuples, sets do not allow duplicate values, and their elements are unordered, meaning the items have no defined position. Sets are useful for situations where the uniqueness of elements is important, such as eliminating duplicate entries from a list.

Key Features:

- Unordered: The items have no index.

- Mutable: You can add or remove items.

- No duplicates: A set automatically removes any duplicate values.

- Iterable: You can loop through the set.

Introduction to Dictionary

A dictionary in Python is a built-in data structure that stores data as key-value pairs. Each key is unique, and it is associated with a value. Unlike lists or tuples, which use numeric indexes, dictionaries allow you to access values by their corresponding keys.

Key Features:

- Mutable: You can add, remove, or modify key-value pairs after the dictionary has been created.

- Unordered: Dictionaries do not maintain a specific order (though from Python 3.7 onwards, insertion order is maintained).

- Unique Keys: Each key in a dictionary must be unique, but the values can be duplicated.

- Flexible: Keys can be of any immutable data type (strings, numbers, tuples), and values can be of any data type (including other dictionaries, lists, etc.).
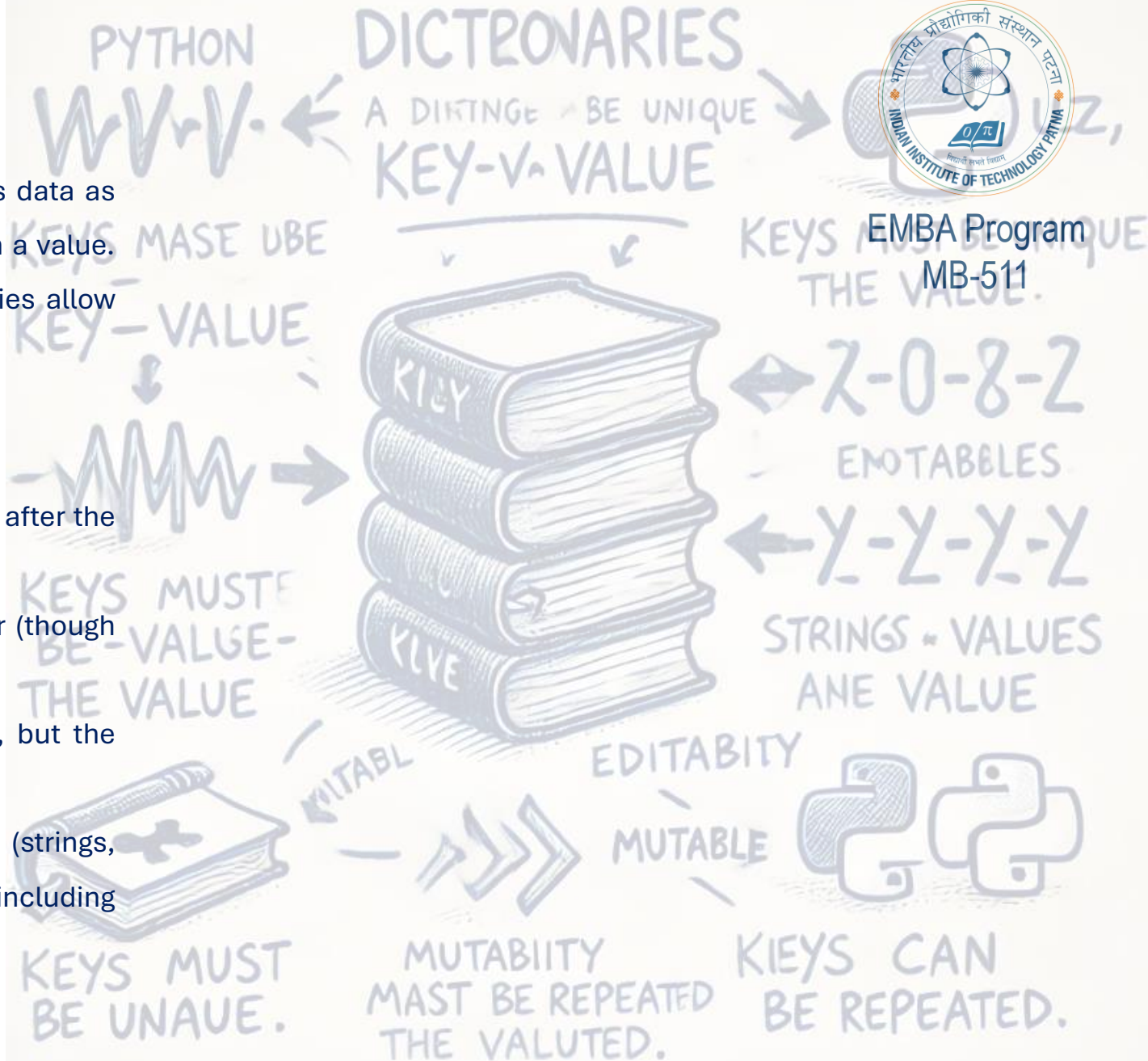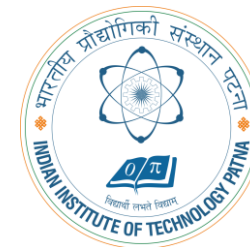
Have a question?

Feel Free to Reach out at

- **+91-88846-92929 (WhatsApp)**
- **anant.awasthi@outlook.com (E-Mail)**