Introduction to Programming with Python

Amin Sakhaei

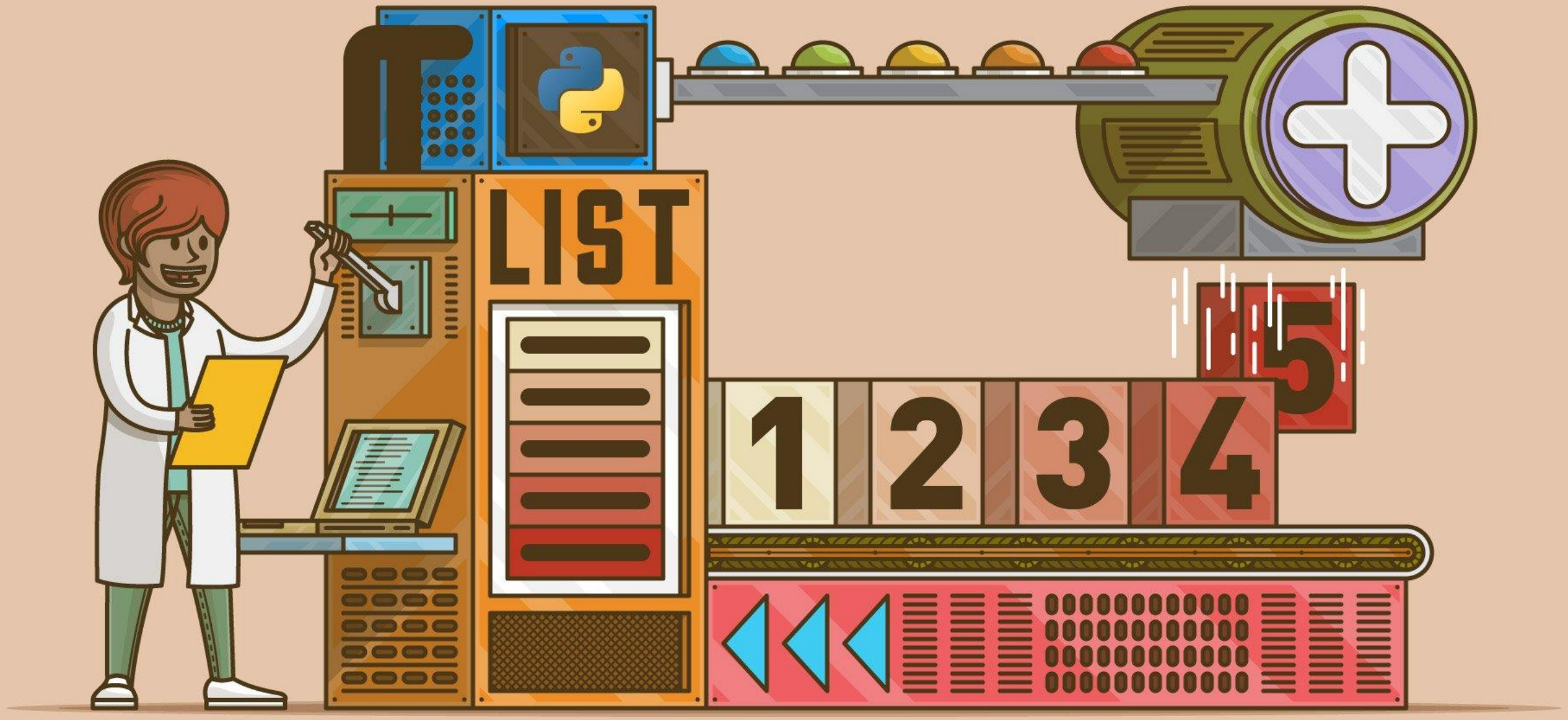
Pouya Taghipour

Amirkabir University of Technology

Department of Biomedical Engineering

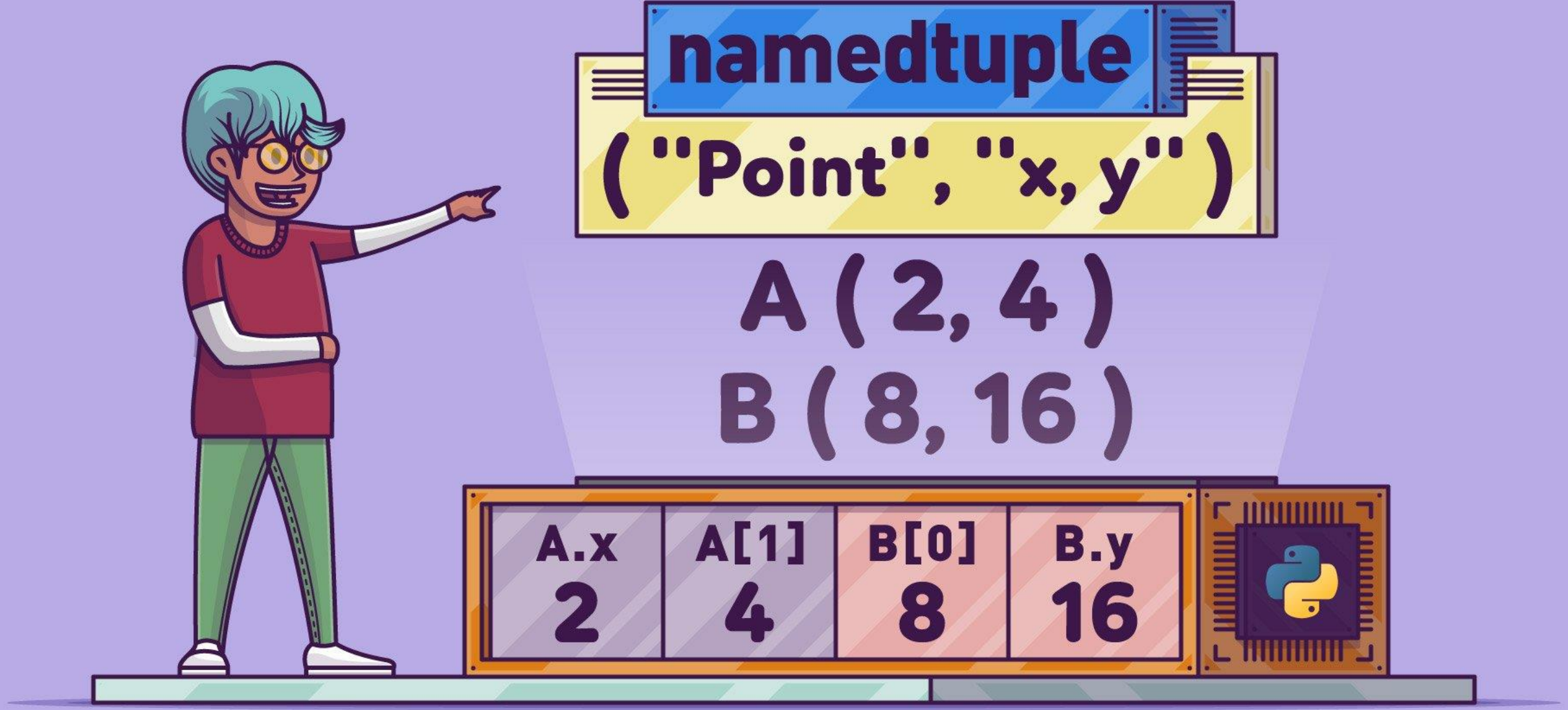
Summer 2023

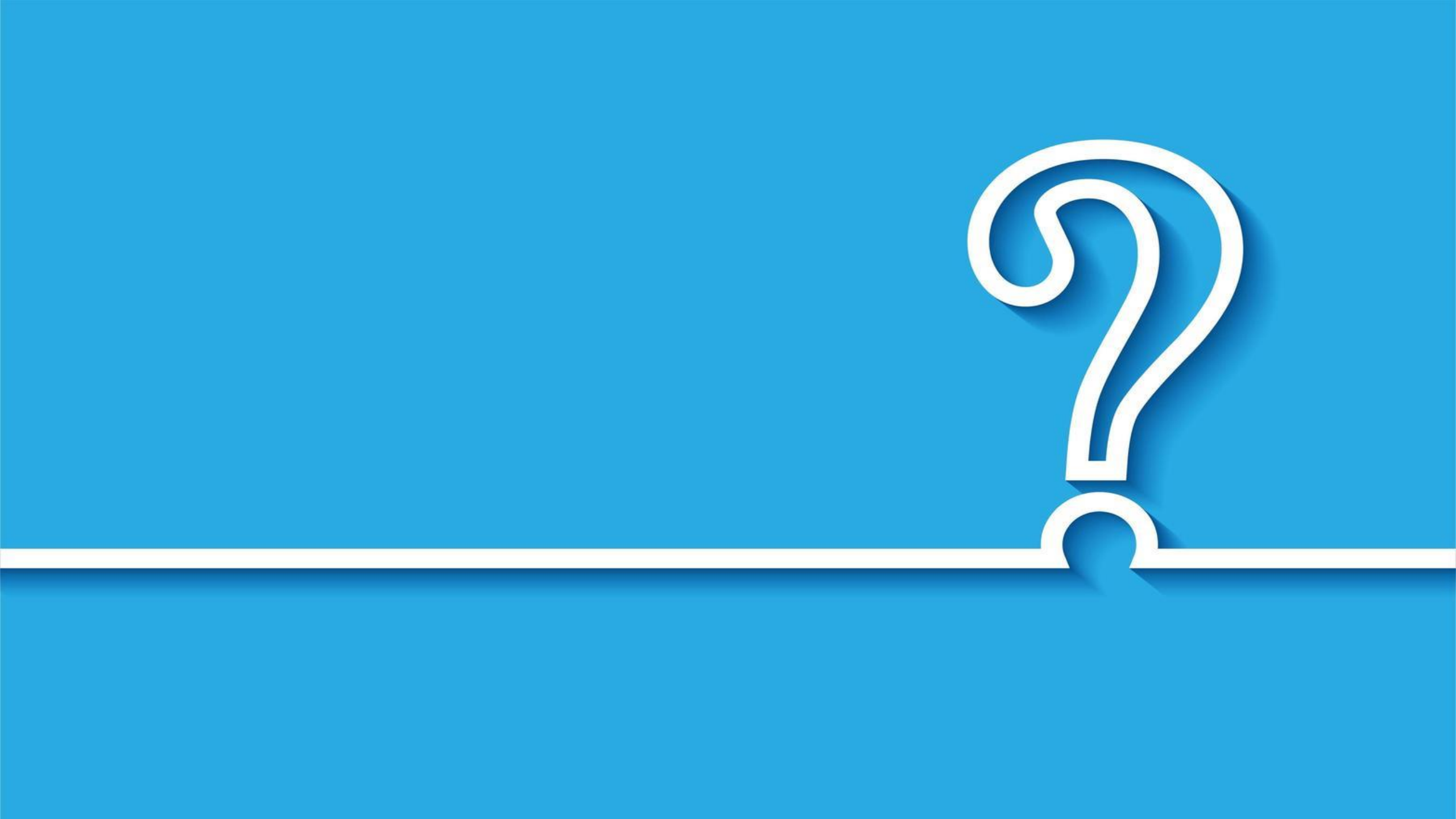
Previously on the


LIST
1
2
3
4
5

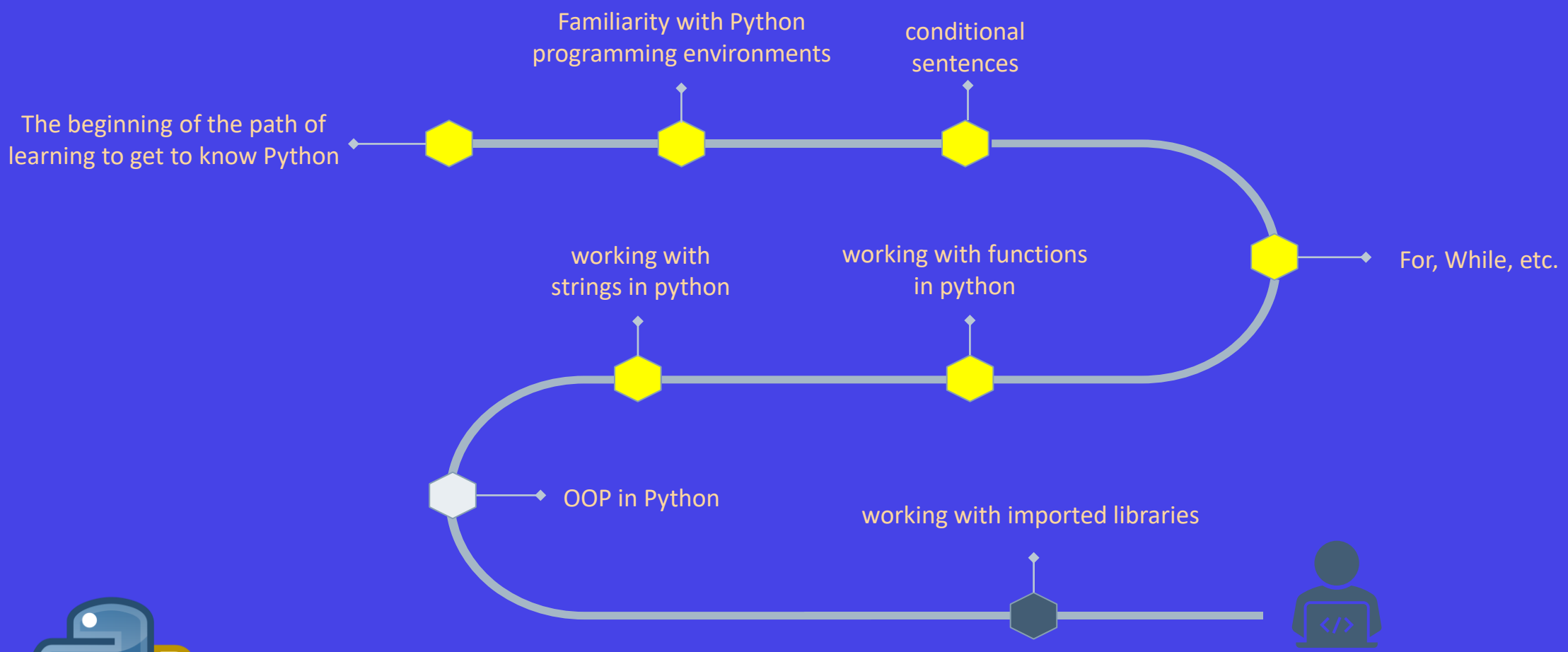

5
2
3
4
1
REVERSING.

List	**Tuple**
Set	**Dictionary**

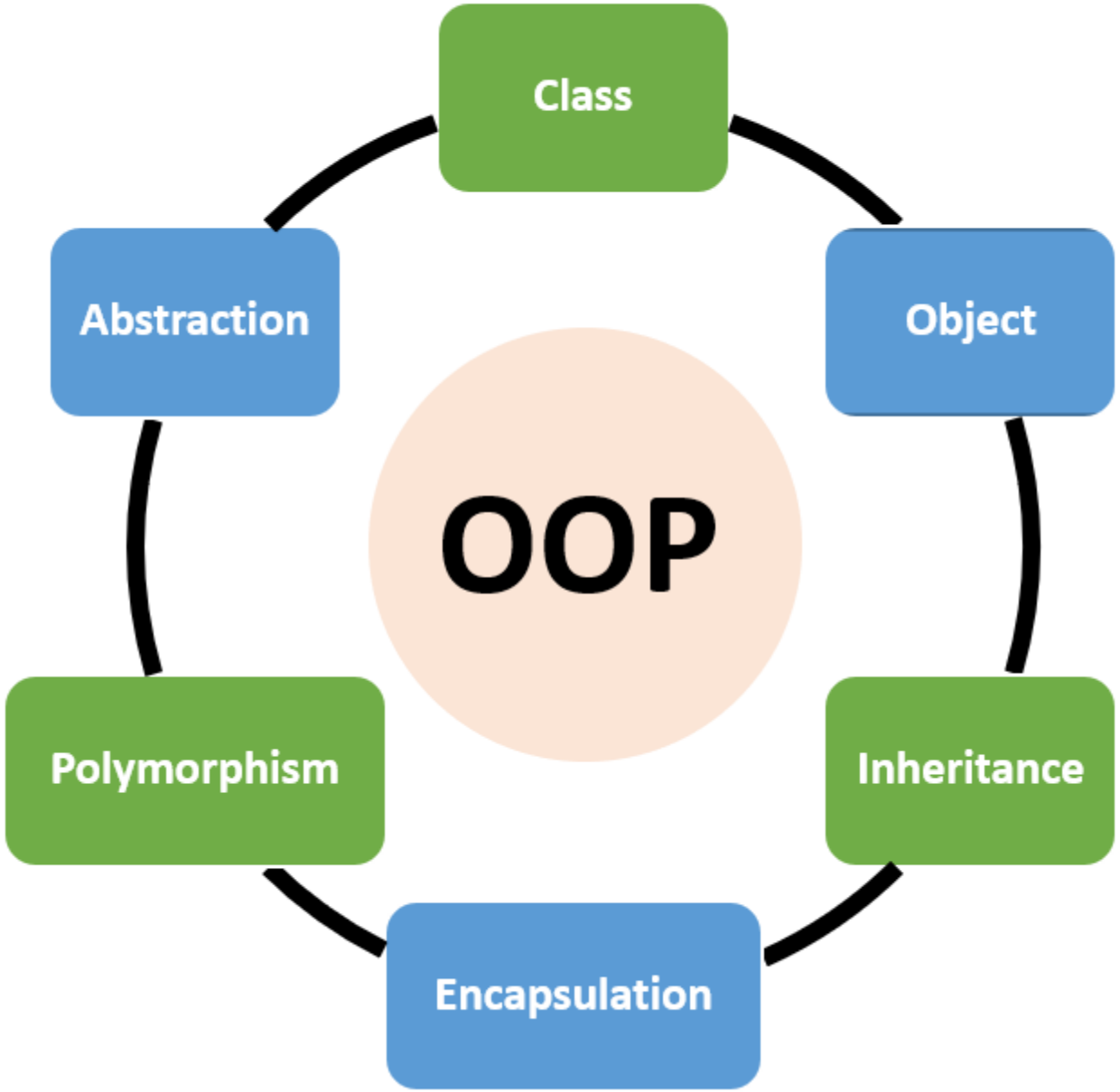

namedtuple
("Point", "x, y")
A (2, 4)
B (8, 16)
A.x
2
A[1]
4
B[0]
8
B.y
16



Course Outline



Object-Oriented Programming


Class
Object
Inheritance
Encapsulation
Polymorphism
Abstraction
OOP

Basic Paradigms

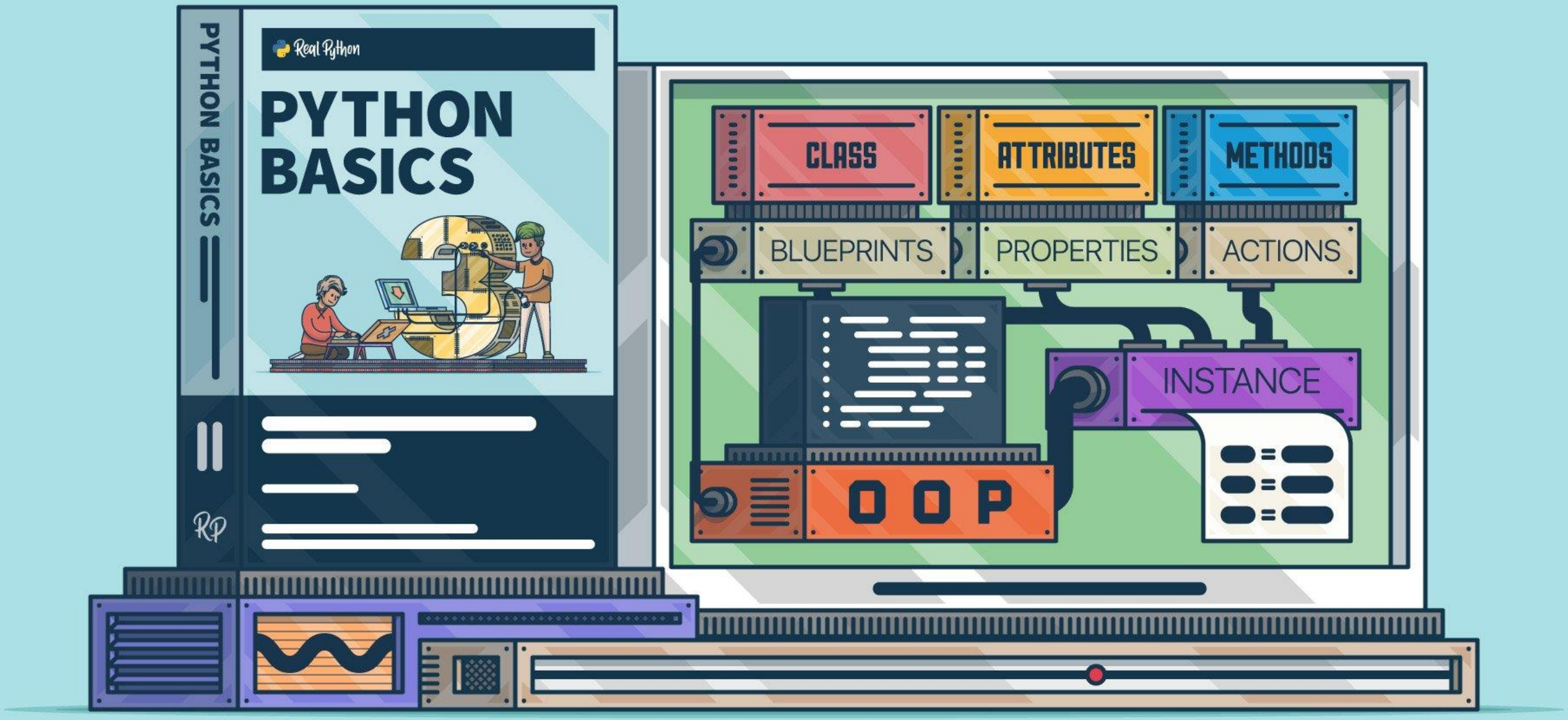
1. Procedural:
   - Organizing code with functions and data manipulation.
   - Using functions for sequential tasks.
   - Handling data with functions.
2. Object-Oriented:
   - Combining data and functionality.

Introduction to OOP

- OOP is a programming paradigm that structures code around objects and classes.
- Python is a versatile language for OOP.

Benefits

- Code Reusability
- Modularity
- Real-World Modeling
- Ease of Maintenance


PYTHON BASICS
Real Python
PYTHON BASICS
RP
CLASS
ATTRIBUTES
METHODS
BLUEPRINTS
PROPERTIES
ACTIONS
INSTANCE
OOP

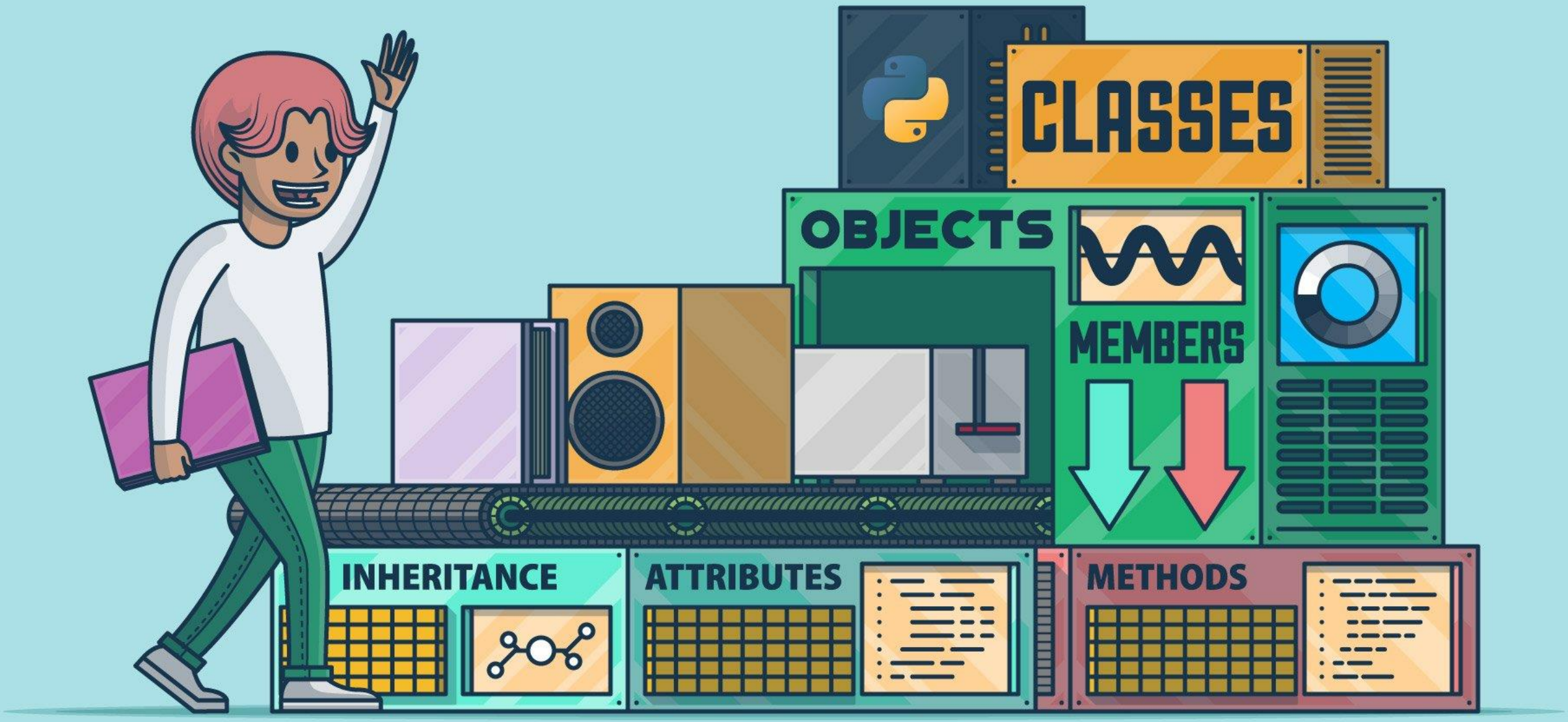
Key OOP Concepts

- Objects and Classes
- Attributes and Methods
- Encapsulation
- Inheritance
- Polymorphism


CLASSES
INSTANCES
ATTRIBUTES

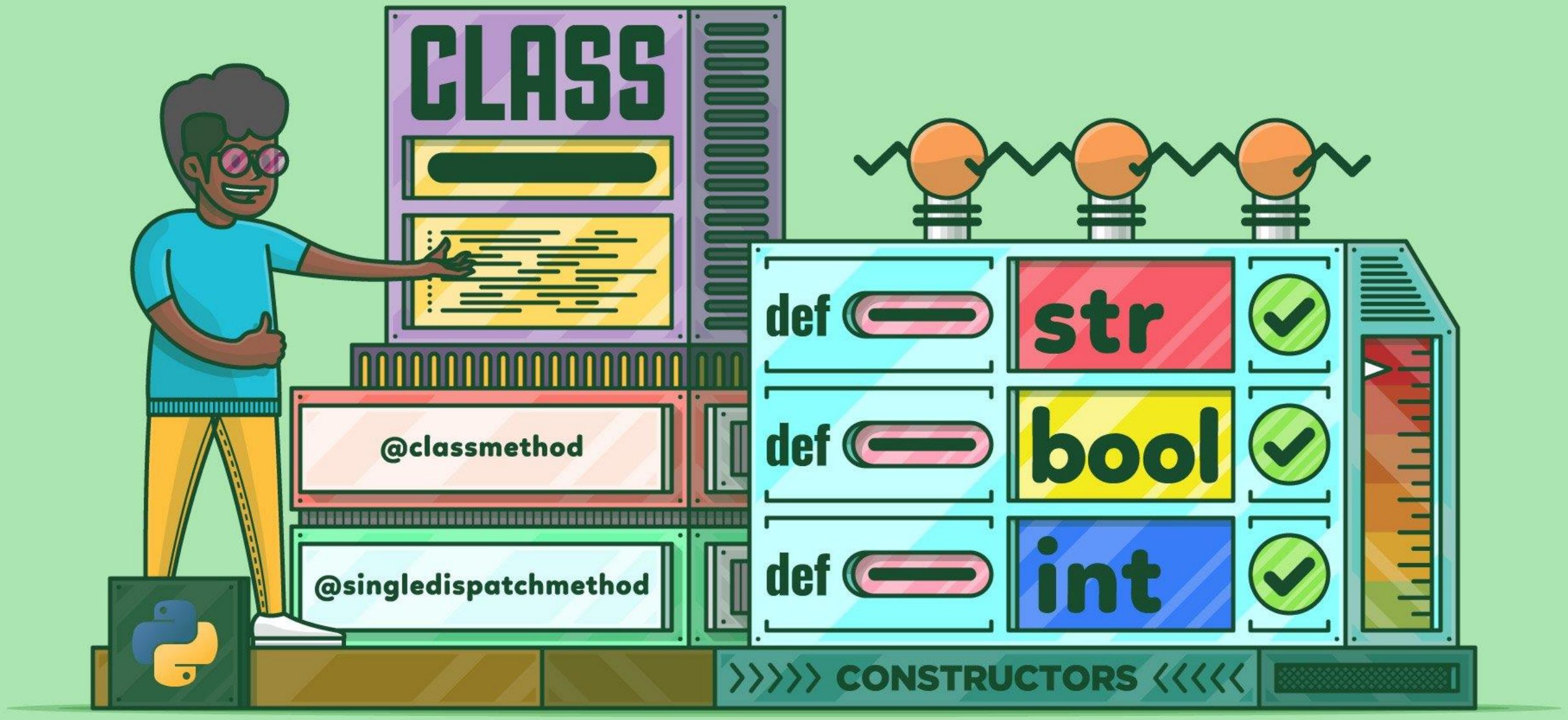
Objects and Classes

- Objects: Instances of classes, represent real-world entities.
- Classes: Define structure and behavior, act as blueprints.


CLASSES
OBJECTS
MEMBERS
INHERITANCE
ATTRIBUTES
METHODS

Attributes and Methods

- Attributes: Data or properties of an object.
- Methods: Functions that define object behavior.


CLASS
@classmethod
@singledispatchmethod
def
str
def
bool
def
int
CONSTRUCTORS

Encapsulation

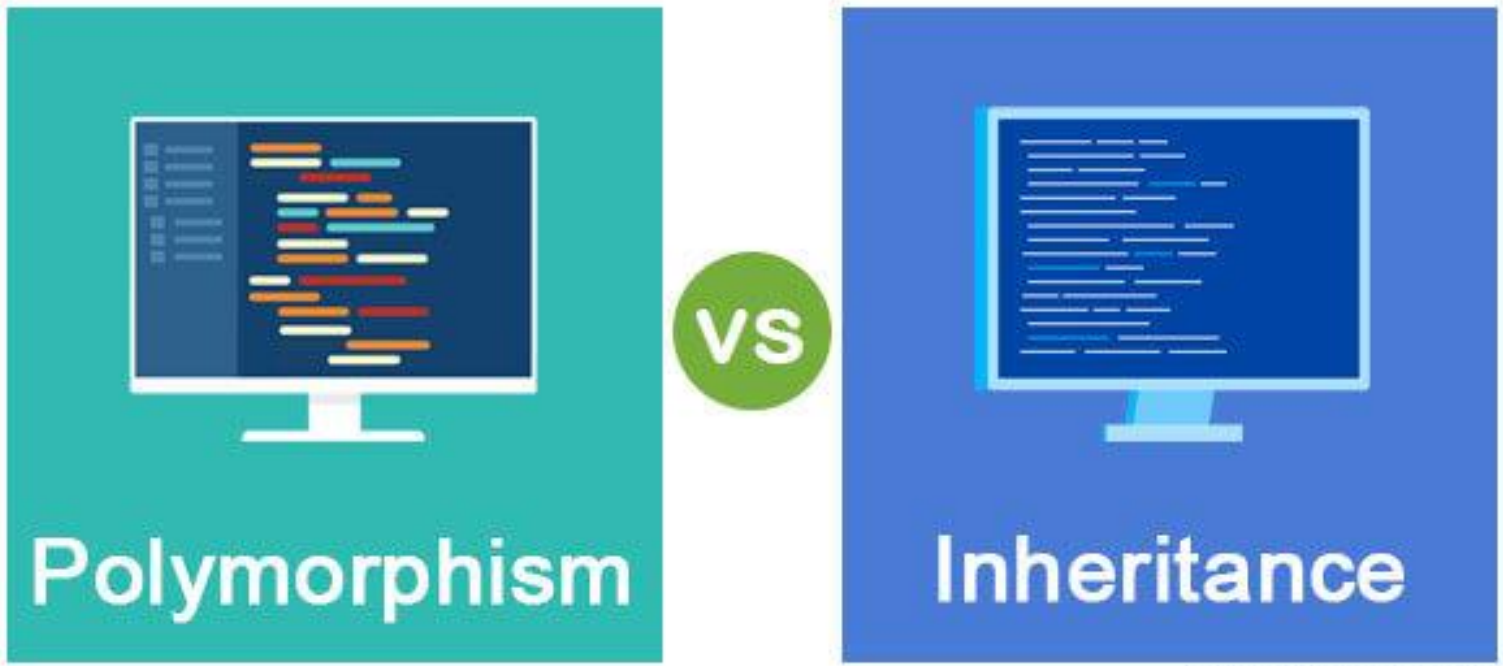
- Definition: Bundling data (attributes) and methods that operate on the data into a single unit (object).
- Benefits: Data hiding, code organization, and data protection.

Inheritance

- Definition: Creating new classes (subclasses) based on existing classes (superclasses).
- Benefits: Code reuse, hierarchical structure.

Polymorphism

- Definition: Objects of different classes can respond to the same method name.
- Examples: Method overriding, interface implementation.


Polymorphism
VS
Inheritance

What we do next?

Contact US

Pouya Taghipour

Pouya_Ta

P.taghipour8@gmail.com

@PouyaTghpr

Amin Sakhaei

AminSakhaei

sakhaiea@gmail.com

@Aminsakhaei

```
print("Have a nice day!")
```