Bootcamp 134 | Python

Course 08 | OOP

Amir Hossein Chegouniyan

Head of the Technical Team at Dariche Tejarat

Lecturer of Python – Django at Maktab Sharif

Amirhossein-chegounian

Content

- Overview of Object-Oriented Programming
- Defining Classes and Creating Objects
- The __init__ Method
- Instance Methods and Attributes
- Class Variables vs. Instance Variables
- Class Methods and Static Methods
- Inheritance

Overview of Object-Oriented Programming

- What is OOP?:
  - Explain the basic idea of OOP and how it models real-world objects.
- Benefits of OOP:
  - Emphasize key advantages like code organization, reusability, and modularity.
- Key Concepts in OOP:
  - Introduce core ideas — Classes, Objects, Methods, and Attributes.

Defining Classes and Creating Objects

- What is a Class?:
  - Explain a class as a blueprint or template for creating objects.
- Creating a Class:
  - Basic syntax of defining a class using the class keyword
- Creating Objects:
  - Demonstrate creating an instance of a class (object) and interacting with it

The __init__ Method

- Purpose of __init__:
  - Introduce __init__ as the constructor method, responsible for initializing object attributes.
- Using __init__ to Set Attributes:
  - Walk through a simple example to set up an object's attributes when it's created.

Instance Methods and Attributes

- Defining Methods:
  - Explain methods as functions defined within a class to define object behavior.
- Accessing Instance Attributes:
  - Show how to define and access instance attributes within methods.
- Calling Methods on Objects:
  - Demonstrate calling a method to interact with an object.

Class Variables vs. Instance Variables

- Instance Variables:
  - Explain that instance variables are specific to each instance (object) of a class.
- Class Variables:
  - Show how class variables are shared across all instances of a class.
- Examples:
  - Compare class and instance variables with a simple example (e.g., shared attributes vs. unique attributes).

Class Methods and Static Methods

- Class Methods:
  - Explain class methods as methods that operate on the class itself, not on individual instances.
- Using @classmethod Decorator:
  - Introduce the @classmethod decorator and show a simple example.
- Static Methods:
  - Introduce static methods as methods that don’t access instance or class attributes.
- Using @staticmethod Decorator:
  - Demonstrate the @staticmethod decorator with a simple example.

Inheritance

- What is Inheritance?:
  - Explain inheritance as a way to create new classes from existing classes.
- Creating a Child Class:
  - Demonstrate how a child class can inherit attributes and methods from a parent class.
- Overriding Methods:
  - Briefly show how a child class can override methods from the parent class.

Any question?

Next course

- Object Lifecycle
- Data Hiding and Encapsulation
- Properties and the @property Decorator
- Magic (Dunder) Methods & Operator Overloading
- Inheritance and Multiple Inheritance
- Composition vs. Inheritance
- Abstraction
- Mixins