Classes in Python

- A **class** is a blueprint for modeling and creating objects, defining their state and behavior with attributes and methods.
- Classes can be used to represent real-world objects or entities that are relevant to the context of a computer program.
    - Examples: a house, a bank account, an employee, a customer, a car, a product.
- The **class header** declares the class name. This name will be used to create objects.
- The **class body** contains the attributes and methods, defining the data and behavior of the objects.
- To define a class, write the class keyword followed by a space, the name of the class, and a colon. Then, start a new line with 4 spaces of indentation to start writing the class body.
- Class names:
    - They are usually nouns.
    - They should follow the Pascal Case naming convention (also known as Upper Camel Case). This involves starting every word with an uppercase letter, without spaces between the words.
    - Examples: **H**ouse, **B**ank**A**ccount, **S**mall**H**ouse.
- The class body must be indented with 4 spaces.
- Class body structure (top to bottom):
    1. Class attributes
    2. __init__() method with Instance Attributes
    3. Methods
- The __init__() method is a special method that is called automatically when object of the class is created, to set up its initial state.

Example

This is an example of a class header, the first line of a class definition. This class represents a player in a video game.