

Code for Teachers

A practical approach to programming

Chapter 6: Objects

6-1: Introducing Objects

Lesson Objectives

- Thinking/Coding in Objects
- Glimpsing the power of thinking in Object-Oriented Programming (OOP)
- Have some fun creating Objects

Objects

- Humans conceive of the world as a composite of objects and their interactions
- “I went to the store”, “The rocket is fast”
- Both describe objects and either an interaction or a state, or both
- Computers don’t think at all; they just process information
- OOP gives us a way of structuring that information in a way that makes sense to humans

Objects in Code

- Objects in code are manifestations of *Classes*
 - Classes are blueprints for individual objects
- When we create a new object, we do so by invoking the class's *constructor* function
- Constructors build a new instance of an Object and return it
- In Python, Classes have the `__init__()` function that acts as a constructor
- Objects have *properties* and *methods*
 - Properties: an Object's characteristics
 - Methods: an Object's abilities

Example: Car

- Cars are great as a Class because all cars have the same attributes
 - Year
 - Make
 - Model
- All cars can do the same things
 - Accelerate/decelerate

Beyond the Constructor

- Adding methods to objects makes working with information easy
- For example the “magic” `__str__()` method tells Python how to `print()` your object.
 - Every time you `print()`, you’re asking Python to call an object’s `__str__()` method.
- What if the cars had `acceleration` and `top_speed` properties?
 - We could race them!



theforeverstudent.com



@mttaggart



mttaggart