Introduction to Objects and Classes

This lesson introduces you to the concept of objects and classes and the benefits of using them.

WE'LL COVER THE FOLLOWING

A Brief Encounter

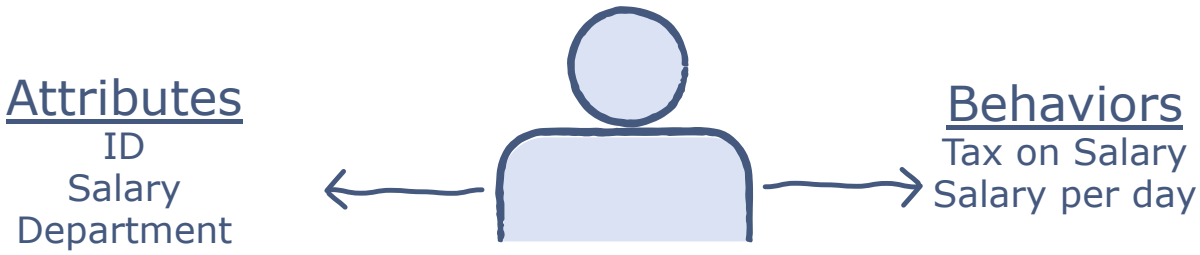
We can see *objects* everywhere in our surroundings. These objects have certain properties that define them. There are certain behaviors that these objects perform on their own, and there are actions that can be performed on them as well.

Let's take the example of a company employee. An employee has the following properties or **attributes**:

- ID
- Salary
- Department

The following actions or **behaviors** can be performed on an employee:

- Calculation of tax on salary
- Calculation of salary per day

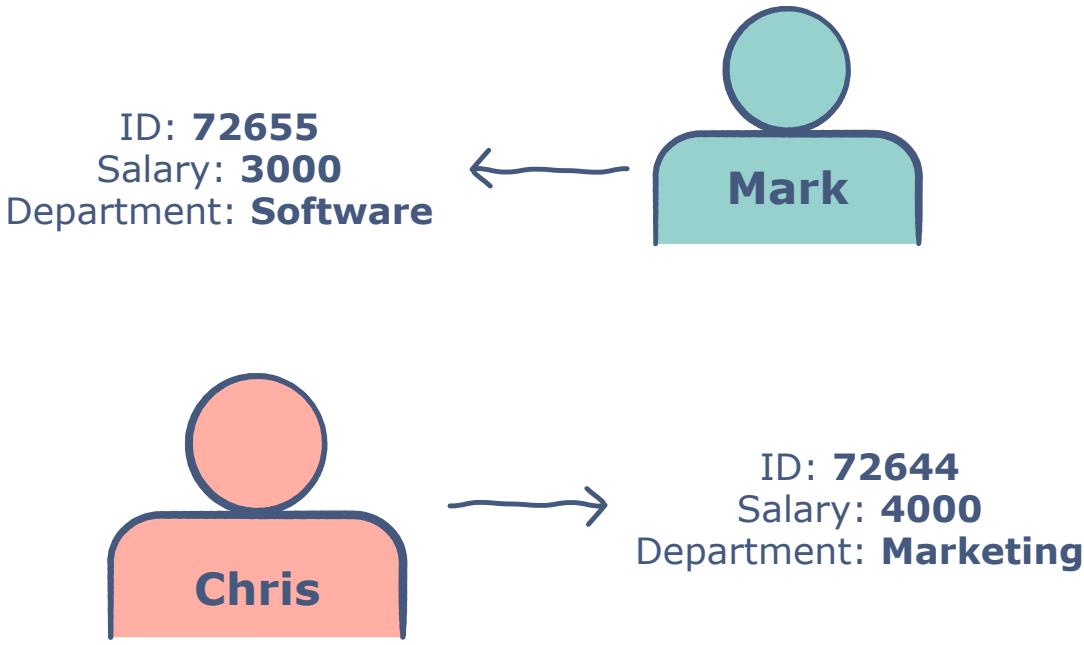


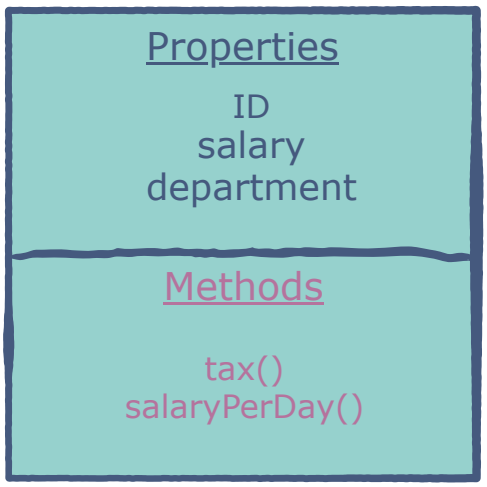
In a company, each worker has a different name, salary, and department but the type of each worker is *employee*. So, there is a generic blueprint for each worker working in the company, but each one of them has different attributes associated with them.

Having the *same* blueprint but *different* properties are the basis for **classes** and **objects**.

Objects and Classes

Suppose, at Educative there are two employees: **Mark** and **Chris**. The properties of Mark and Chris are given in the image below:





employee class

> You might see attributes referred to as **properties** or **members** by different people. For consistency, we'll be using the convention of *properties* throughout the course.

Properties

Properties are variables that contain information regarding the object of a class. An employee object will have an `ID`, a `salary`, and the `department` as its *properties*. New properties could be added to be a part of an object of the employee class.

> You might see behaviors referred to as **member functions**, or **methods** by different people. For consistency, we'll be using the convention of *methods* throughout the course.

Methods

Methods are like functions that have access to properties (and other methods) of a class. Methods can accept parameters and return values. They are used to perform an action on an object of a class. In the example above, we have `tax()` and `salaryPerDay()` as class methods.

Benefits of Objects and Classes

The concepts of objects and classes allow us to create complex applications in Python. This is why classes and objects are considered the basic building

blocks behind all OOP's principles.

- They are also instrumental for compartmentalizing code. Different components can become separate classes which would interact through interfaces. These ready-made components will also be available for use in future applications.
- The use of classes makes it easier to maintain the different parts of an application since it is easier to make changes in classes (more on this later).

In the next lesson, we will discuss how to declare a class in Python.