

12-Python-Classes and Objects

Classes and Objects

Object-oriented Programming

1 December 2025

Classes

If we spend some time to think about it, one of the big goals of programming is describing something that exists in the physical space using a programming language. For example, if we wanted to make a program that describes a human being, we would need to "design" a way to describe a human.

A class is a way to describe an object.

Being and Doing

We can split up these concepts into being and doing. Let's say that we describe these things as objects. We'll use a human as our example of an object. So, an object exists and it does, two verbs that can describe humans. When an object exists, we can describe its state at a particular time. Humans can be described using their name, their height, their eye colour, etc. All these things are pieces of information.

Humans can also do things. Humans can walk, they can speak, think, etc.

These two things, being and doing, can be described using object-oriented programming.

Let's describe a human using Python and classes.

Classes in Python

Let's create a human class in Python. We need to name the class and in Python, and a number of other languages, including Java, we use a capital to start out the name of our class.

We'll name our class, `Human`.

Another thing we should do is create a **constructor**. The constructor is a special *method* (a method is a function that works on an object; more about methods later) that says how an object is created. The method has a name called `__init__()`. It has at least one parameter, `self`. `self` refers to the object itself.

The constructor method **runs whenever an object is created**.

For our example, let's use the suggestions above for the being part of humans. When an object exists, we call its variables *properties*. These are the data that describe an object: name, height, eye colour.

```
class Human:  
    def __init__(self):  
        """This is the human constructor"""  
        self.name = ""          # when a human is born, it has no name  
        self.height = 10 cm     # I'm taking a guess here  
        self.eye_colour = ""   # we'll decide this later, as well  
        print("A human is born!")
```

Creating an Object and Changing Properties

So, now that we've described how a human exists. Let's create a human (please don't clip this out of context).

In addition to creating a new human, we're going to modify some of its properties. Remember, that when a human is created, it doesn't have a name nor any eye colour.

```
# Assuming the code above is part of this  
  
if __name__ == "__main__":  
    # create a human  
    human_one = Human()           # this runs the constructor  
    print(f"{human_one.name} is a person who has {human_one.eye_colour}  
coloured eyes.")  
  
    # give it a name  
    human_one.name = "Mr. Ubial"  
  
    # give it an eye colour  
    human_one.eye_colour = "brown"  
  
    # use some of its properties  
    print(f"{human_one.name} is a person who has {human_one.eye_colour}  
coloured eyes.")
```

This would be the output. Notice that the first line is from the constructor!

Also, the second `print()` function call shows that our new object doesn't have a `name` or an `eye_colour`.

Also, to note about the code above, we can change the properties of an object in the same way as a variable by using the assignment operator (`=`). We can also access the value in the property using the name of that property.

- > A human is born!
- > is a person who has coloured eyes.
- > Mr. Ubial is a person who has brown coloured eyes.

Doing and Methods

Methods are things that the objects can do. Remember that humans can walk, talk, and think. We can use methods to describe what objects can do.

This is what the class looks like with our new methods.

```
import random

# ...some code here maybe...

# Rewriting the class above to include more methods aside from the constructor

class Human:
    def __init__(self):
        """This is the human constructor"""
        self.name = ""          # when a human is born, it has no name
        self.height = 10 cm     # I'm taking a guess here
        self.eye_colour = ""   # we'll decide this later, as well
        print("A human is born!")

    def walk(self):
        """Make the human walk!””
        print("You see the human walk a couple of steps. 🚶 ”)

    def talk(self, words: str):
        """How a human verbalizes.

Params:
    words – the words that the human should say”

if words:
    print(words)
    return
else:
    print(f"Hello, world! My name is {self.name}.")
```

```
def think(self) -> str:  
    """See what the person is thinking...  
  
    Returns:  
        what the person is thinking"""\n    thoughts = [  
        "What am I going to eat for dinner?",  
        "How do large language models actually work?",  
        "What's the meaning of life?"  
    ]  
  
    return random.choice(thoughts)  
  
# rewriting the __name__ == "__main__"  
  
if __name__ == "__main__":  
    # Create a human  
    human_one = Human()  
  
    # Modify its properties  
    human_one.name = hello  
    # Use the methods
```