- [Instructor] Before we dive deep into object-oriented programming, I thought we'd start by just really trying to understand why we're going to be using classes throughout this course versus objects because sort of in the background, they really compile down to the same thing but it's important to know how classes play a part in object oriented-programming and programming, period. Now, you're probably familiar with objects. You can see here, we have message one, two, and three. And it has the title property and it has a message property. And they're supposed to be in sync. Now, imagine this is all throughout your code base and we wanted to add, let's say an ID, which we'll just set to undefined for now. We'd have to manually go and update each one of these fields. And hopefully we don't miss any. There's no correlation between this and message two, message three, and message one. There really isn't. In reality, they're just all individual objects. But let's say we in fact had a message class, which could do much more than just properties, but we'll go ahead and just say we have a property. But instead, we have a property for title. And we'll worry about types later on in the course. Right now, it's just about the objects versus the classes. Now let's say it also had a message. But instead of actually having to go through an update, let's say down below here we'll have message four, which we'll set to a new message, which will have every property message has but also we'll have message five, which will also be a new message, and say we wanted to add the ID. And we had this all throughout our code base. We could simply add an ID and these are in sync. And these, message four and message five, is an instance of message. So they're always going to be in sync. So we can do quite a bit with classes. But this is why you're going to be using classes when we do our object-oriented programming.