

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high-quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

ANA BELL:

OK. We have our class. We've done pretty much everything up until here, and we're going to add this special function here. So we're implementing this underscore underscore eq underscore underscore. And implementing this is going to allow us to compare two types of cars.

So the way I've decided to compare two types of cars is I'm saying the two cars are going to be equal if they have the same number of wheels, they have the same color, and if they have the same number of doors. OK? So if all of these are equal, then return true and else return false.

Inside the actual program here, I'm creating one car, four wheels, two doors. I'm changing its color to red. I'm creating another car-- four wheels, two doors. By default, this new car, or your car, is going to have the color empty string, because that's how a new car gets initialized, right?

So between my car and your car, the difference is going to be the color. OK? So the color is not going to be the same, but they have the same number of wheels and the same number of doors.

So because I implemented the equal method in my code, this does not throw an error. This lets me proceed. It compares four with four, which is good, two with two, which is good. And then the colors don't match, so it's going to say false, which most of the guys are getting it right.