* - JavaScript is a prototype based object oriented programming language .
* **That means, at its core, when we work with JavaScript, we're working with objects, and these objects are based on prototype .**
* which is a very technical and confusing way of saying JavaScript works very much the same way we humans work with objects in the real world .
* So the quickest path to understanding JavaScript is through understanding objects and how they work .
* Because an object in JavaScript is pretty much the same as an object in real life, except it's written in code instead of created as a physical object .
* Let me show you what I mean using my backpack .
* This backpack is an object .
* This object has many identifiable properties like color, size, et cetera .
* We can give all of these properties recognizable names and then use them to describe this individual backpack objects .
* This backpack has the color gray .
* That's a property .
* If we fill it all the way up, we'll discover its volume is 30 liters .
* Another property .
* It has a bunch of pockets, 15 I think, that's a property .
* It has seven straps as a property, et cetera, et cetera .
* Object properties define the specifics of this one particular object .
* That way, I can have several backpacks, and why they're all backpacks, each has its own set of properties which makes that backpack a unique object .
* Now, I said **JavaScript is a prototype based object oriented programming language .**
* That means, each object is a unique instance of an object prototype .
* I have a lot of backpacks .
* The reason I can say they're all backpacks is they share common properties which define them as belonging to the backpack category .
* They all have two straps on one side, I can put my arms through .
* The area behind those straps is flat so it fits comfortably against my back .
* And they all have an opening at the top .
* This properties describe the prototype of a backpack and the particular combination and configuration of these properties define each individual backpack .
* This object prototype makes it easy to identify an existing bag as a backpack .
* You can check to see if it has the properties that match the prototype, and makes it easy to create a new backpack .
* Simply describe it by filling the property values .
* This prototype based object orientation allows us to do in programming, what we as humans do every time we encounter a new object and catalog it in our minds .
* We look at it, note its properties and think, this thing has the same collection of properties as this other thing I already know .
* So even though it is a different thing in appearance, the similarities in properties means its the same object type .
* And there's more, ***objects can have features built into them allowing us to change their property values .***
* This backpack has several such features .
* A lid that can be opened and closed, a zipper to open and close the left and right sides, Strap adjusters to change the strap length, et cetera .
* In JavaScript, these property changing features inside an object are called **methods** .
* Which makes sense if you think about it, there's a method for opening and closing the backpack, a method for lengthening the straps and so on .
* ***These methods act on the current object only .***
* So if I open this backpack using its top access method nothing happens to this other backpack because it's an entirely separate object .
* Oh, and one final thing, ***objects can contain other objects .***
* So inside this backpack object, there is another object, a headlamp .
* And because the headlamp object is inside the backpack object, it is quite literally a property of the backpack object for as long as it's in there .
* This headlamp has its own set of properties, color, size, battery status, et cetera, and has its own set of methods, I can turn the light on and off, I can adjust the strap, I can charge it and so on .
* The headlamp object like the backpack object is a unique and separate object .
* It can be included inside another object, or it can be separate .
* There can be one or many of them and changing one object does not change the others .
* This mental model of objects described by their unique properties based on an object prototype whose individual properties can be modified using internal methods, is what I want you to keep front of mind as we explore JavaScript together .