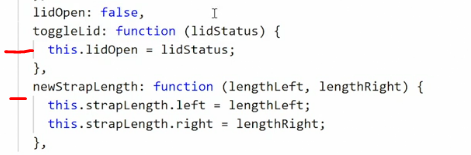
* - [Instructor] In addition to properties, objects can contain their own functions .
* These functions typically perform actions on the properties of the object .
* And ***when a function is inside an object, it is called a method .***
* So when you hear me talk about methods, I'm referring to functions sitting inside objects .
* ***Each method is added to the object as a property .***
* You can see here, we have a bunch of properties: Name, volume, color, and so on .
* And then we have two properties towards the bottom, toggle lid and new strap length that contain functions .
* So these are methods .
* There are actually **two syntaxes for these methods** .



* The one you see here is a function expression meaning we are explicitly saying inside toggle lid, there's a function that has these parameters and this is the function body .
* We can also do a shorthand for this by taking the function expression away and just saying toggle lid, parentheses, and then the parameters .

Graphical user interface, application

Description automatically generated

* It works pretty much the same way, but this shorthand is somewhat **harder to read .**
* So the convention is to use a function expression .
* So it says function, it's very clear of what's going on .
* Now, let's look at how one of these methods work .
* So we have this toggle lid method .
* Inside the toggle lid method sits a function that receives one parameter .
* A parameter is a piece of data we can pass to the function .
* That parameter is then used inside the function to set the new value for the lid open properties .
* So we're saying this .
* .
* .
* So this current objects, lid open property using dot notation, set the value of that equal to whatever sits inside lid status, which is passed as parameter .
* Now we haven't covered functions yet so here's a quick crash course .
* And by the way, you can also skip ahead to the chapter on functions if you want to go more in depth right away, but it's not necessary .
* We'll get to functions at a logical point in the progression of this course .
* A function like you see here is a program that does something, typically change a value somewhere .
* Its most basic form is a function expression like the one you see here where we say function, meaning it's a function .
* Then we use parentheses to capture any parameters passed to the function .
* These are the values the function can use .
* And then we wrap curly brackets around the statements of the function, so what the function does .
* Function, parameter, curly brackets around the statements and that sits here .
* To make a function run, we call it by stating its name and then adding two parentheses behind it .
* This tells JavaScript this is a function and I want to run it .
* It's called a function call .
* If we want to pass values to the function, we put those values inside the parentheses and they become the parameters of the function .
* Let me show you how this works .
* We can do it first in the browser so we'll try to use this toggle lid function to change the lid open value .
* First, I'll log out the lid open values .
* I'll say backpack lid open, and you can see the value currently is false .

Text

Description automatically generated with medium confidence

* Then we'll use the toggle lid method .
* So I'll say backpack .
* And because it is a property, I can just call it using dot notation so I'll say toggle lid .
* Now the toggle lid method is a method, meaning we need to tell the browser hey, we want to run this as a function so I'll put parentheses around it .
* Then I need to pass the value .
* Remember inside this function, we are grabbing a parameter and then we're using that value to reset the value here .
* So I want to set the value to true since this is a Boolean value .

Graphical user interface, text, application, email

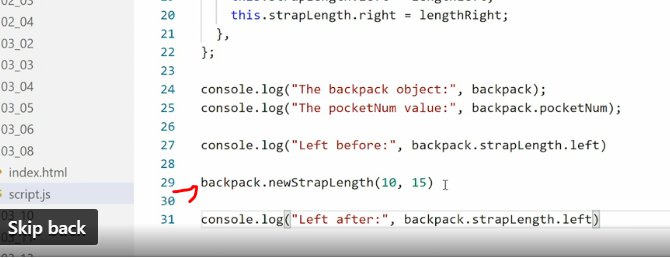
Description automatically generated

* Hit return and we get this undefined because the function itself is not returning anything to the browser .
* However, something changed in the backpack lid open .
* So if I arrow key up twice, I get to the call for backpack lid open and you see the value is now true .

Graphical user interface, text, application

Description automatically generated

* Now important, I didn't change the value in script .
* JS .
* This value only changed inside the browser and that's kind of the point of having these methods .
* We can now pass an object to the browser and then change its properties at will using the methods and those changes only happen in the user's computer in the user's browser at that moment .
* We can also access these methods and use them inside our script .
* And down here, we have the new strap length method as well .
* This one has two parameters, length left and length right because we are changing both of these properties up here .
* So to access this, we first need to see that this actually happens .
* So let me console log out the value of the strap length left before then run the method and then console log it out again after to see that the change happens .
* So let's say console log left, left before, and then we grab backpack .
* straplength .
* left .
* Then we used a new strap length method .
* So I'll call it by saying backpack .
* newstraplength .
* And here I need to pass in two values, one for left and one for right, so I'll say 10 comma 15 .
* And then I'll just copy this line at the top here and paste it in and then say left after .



* So you remember how the browser reads JavaScript from the top to the bottom? What will happen here is the browser will read down and go, blah, blah, blah, blah, blah .
* Oh okay, you want the console log outs, the backpack strap length left, here's the value .
* Then it encounters backpack's new strap length and says oh, you want to run this method? Cool, I'll run the method .
* The method will then change the values up here .
* Strap length left and right to 10 and 15 because of this function .
* It grabs the two values, passes them in, reassigns these values .
* Then we go down and say console log again .
* Same value, backpack strap length left .
* We have now changed those values so we get a different output .
* Save this, run it in the browser and you see now we have left before 26 and left after 10 .

Graphical user interface, text, application

Description automatically generated

* This in a very simplified form is how methods work where you define them inside the object, and we call them using dot notation, pass parameters to them and changes happen accordingly and we can then stack the order of when we want to do things to get the output we want .