**Practice: Pass values between functions**

Selecting transcript lines in this section will navigate to timestamp in the video

- Passing data to and returning data from functions changes a function from a basic, single purpose tool to an entire factory for data manipulation. In this practice assignment, you'll get to play with passing data back and forth between functions to create more efficient and reusable code. This is sort of a lead-up to how we work with components in JavaScript frameworks. So getting your mind on this track here and now will help you in the future as you continue on your learning journey. Here's what I want you to do. Writing markup for images quickly becomes clunky, especially if you wrap your images in a figure tag and add a figcaption or you use responsive images markup. In the practice files, you'll find a basic html page with a main container. Using everything you've learned so far, I want you to

- create two functions, one dependent on the other, to output some data from a custom object.

The object is already defined in the JavaScript file. And all you have to do is pull the different properties and add them wherever you think they make sense.

This is a practice assignment, so be as creative or as uncreative as you like. Like I said, create two functions.

The main function creates a new article element, populates it with some content from the object properties and returns that new element with its content to where the function was called. The second function is called by the main function. Its purpose is to create a new figure element populated with an image pointing to the image URI defined in the object, add a figcaption with an image description, and return that whole figure back to the main function. Finally, use querySelector and the append method to append the article with all its content, including that figure, to the main in the document. The key to all this is to pass the objects to the main function, then pass it to the figure function, return everything from the figure function into the main function, and then return everything back to where the main function was called. The goal of this practice example is for you to get familiar with the idea of breaking big tasks into smaller tasks. You see, marking up images can quickly become a big task. And once you've created a function for image markup, you can reuse that function anywhere you want to display an image by just passing in the arguments you want, which in our case would be the image URI and the description.