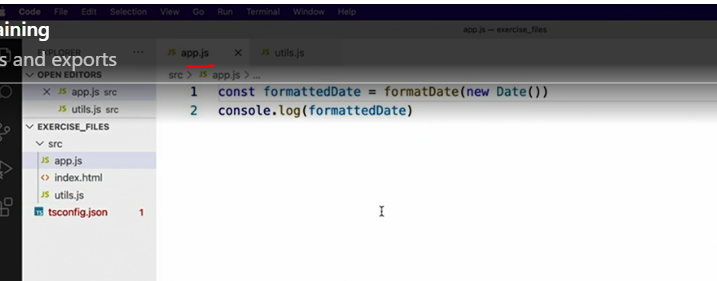
* - [Instructor] In the previous video, I very quickly showed you an application that did not use modules and then that same application after it had been converted to use modules.
* In this video, I'll back up to that pre-modules application and walk you through exactly how to convert it to a module-based application.
* This is the entry point of the application, App.
* js.

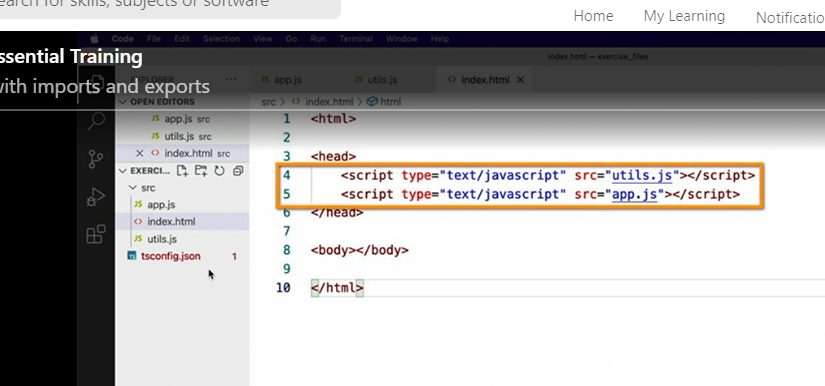


* The code in App.
* js relies on a utility function, format date, defined in the util.
* js file.

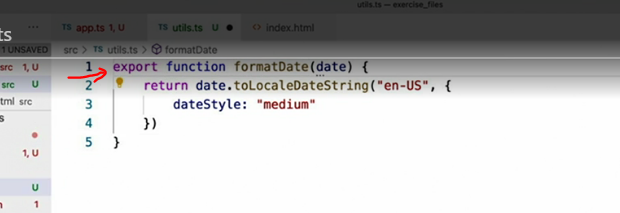
Graphical user interface, text, application

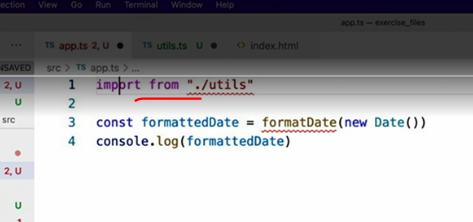
Description automatically generated

* And it gains access to that function simply by the fact that we happen to be loading both of these files into the same browser window.



* The moment we convert these files to TypeScript, however, TypeScript complains about the fact that it doesn't know about function, format, date.
* To make TypeScript happy again, we'll need to tell it where it can find this function.
* That is, we'll have to explicitly import this function from the other file.
* However, before something can be imported by a module, it must first be exported, so let's do that.



* In order to make a function or variable defined in a module available to other modules, we simply export it.
* That is, we add the export keyword just before it's definition like this.
* Then we can jump back to app.
* ts, the module that wants to use this function, and import the function using this syntax.
* I start with the import keyword and then I say which module I want to import from.
* Since this is a local file, I'll just provide the path to the file without the extension.
* This adds the reference to the target module.
* 
* Now, I need to declare what exports I want to import by listing them in side braces, like I'm defining a JSON object.

Text

Description automatically generated

* And that's it, I can now use this function as if it were defined locally.
* And that is your quick intro into modules and how to use them with TypeScript.
* But be aware that there's plenty more you can learn, however, it's outside of the scope of this course since it's all just JavaScript and really just doesn't have much to do with TypeScript.
* I highly recommend that you search our outstanding video library for JavaScript modules to find many great videos to help you use modules to their fullest potential.