**Modules (Files)**

We know that it’s a good practice to separate your code into different file, or modules, which can then be combined and used in a big one.

There are 2 ways of implementing modules in JavaScript: ES6 and Node.js. These are different *runtime environments* in which our code is executed, and each environment has different syntax for implementing modules.

1. **JavaScript Runtime Environments**

A runtime environment is where your program will be executed. It determines what global objects your program can access and it can also impact how it runs.

**A Browser’s Runtime Environment**

This is common as we try to run an html file on our browser. Here, we are executing this code in the *browser’s runtime environment*. The window object provides access to huge amount of data and functionalities.

Applications created for and executed in the browser are known as front-end applications. For a long time, JavaScript code could only be executed in a browser and was used exclusively for creating front-end applications. In order to create back-end applications that could run on a computer WITHOUT a browser, you would need to use other programming languages such as Java or PHP.

**The Node Runtime Environment**

On the other hand, the Node runtime environment was created to give us access to server’s file system, database, and network (full-stack).

For example, suppose you created a file called **my-app.js**. We can check to see the directory that this file is located in using the Node runtime environment variable **process**:

// my-app.js

console.log(process.env.PWD);

🡪 In summary: In each of these environments, different data values and functions are available, and these differences help distinguish front-end applications from back-end applications.

* Front-end JavaScript applications are executed in a browser’s runtime environment and have access to the window object.
* Back-end JavaScript applications are executed in the Node runtime environment and have access to the file system, databases, and networks attached to the server.

1. [Implementing Modules in Node](https://www.codecademy.com/courses/learn-intermediate-javascript/articles/implementing-modules-in-node)
2. Implementing Modules in ES6