

But when using node our JS code is executed outside of a browser / on the server. With this we get info about the OS, we can work with files, can work with network e.g. we can build a webserver that listens for HTTP requests on a given port.

* File System Module:

In this module we've a comprehensive set of methods for working with files & directories.

All operations defined here comes in two forms: ①

(i) Synchronous / Blocking

(ii) Asynchronous / Non-blocking

We have access i.e. Asynchronous method

Also, we have accessSync i.e. ~~Non~~ Synchronous Method.

In real world application we should Asynchronous Methods because these are non-blocking. Cruz, A node process has a single thread. If you're using node to build the backend for your application you might have several clients connecting to that backend, if you keep that single-thread busy you won't be able to serve many clients, so also use Asynchronous Methods.

Class: It is a container for properties & fns which we call methods.
Also, it is a container for a bunch of related methods & properties.

Date _____
Page _____

→ readDir (directory method):

→ readDirSync (directory method, Synchronous form):

It will return all the files & folders in the current folder.

→ readDir (Asynchronous form): This method takes a fn as their last argument, node will call this fn when that asynchronous operation completes.

We call this function a callback.

Callback fn: It has two parameters (i) Error & (ii) Result.
So, we need to pass a fn as callback fn with an error & the result which is a string array.

So, we need to check if we've error/result, only one of these argument will have a value & the other will be not.

Imp →

In order to work with files/folders & directories in node, we need to require the FS module & then use one/more of the module methods defined in this module.

All these methods comes in pairs, Asynchronous & Synchronous.
(Always prefer to use asynchronous methods).