

# NODEJS

## ● Path Module

```
const path= require("path");
//path
console.log(path.dirname('/Users/sumeetpravinjumde/Documents/Work/
Coding/Nodejs/repl_env/pathmodule/pathmod.js'));
// entension
console.log(path.extname('/Users/sumeetpravinjumde/Documents/Work/
Coding/Nodejs/repl_env/pathmodule/pathmod.js'));
// base name
console.log(path.basename('/Users/sumeetpravinjumde/Documents/Work/
Coding/Nodejs/repl_env/pathmodule/pathmod.js'));
// return an object having all info
console.log(path.parse('/Users/sumeetpravinjumde/Documents/Work/Coding/
Nodejs/repl_env/pathmodule/pathmod.js'));
// access all info of file
const mypath= path.parse('/Users/sumeetpravinjumde/Documents/Work/
Coding/Nodejs/repl_env/pathmodule/pathmod.js');
console.log(mypath.name);
console.log(mypath.ext);

// console
// sumeetpravinjumde@Sumeets-Air pathmodule % node pathmod.js
// /Users/sumeetpravinjumde/Documents/Work/Coding/Nodejs/repl_env/
pathmodule
// .js
// pathmod.js
// {
//   root: '/',
//   dir: '/Users/sumeetpravinjumde/Documents/Work/Coding/Nodejs/repl_env/
pathmodule',
//   base: 'pathmod.js',
//   ext: '.js',
//   name: 'pathmod'
// }
// pathmod
// .js
```

## ● Os Modules

```
const { copyFileSync } = require("fs");
const os = require("os");
console.log(os.arch());
// console
// sumeetpravinjumde@Sumeets-Air osmodule % node osmod.js
```

```
// arm64
// sumeetpravinjumde@Sumeets-Air osmodule %
const freeMemory = os.freemem();
console.log(`${freeMemory}`);
const totalmem = os.totalmem();
console.log(`${totalmem}`);
console.log(os.hostname());
console.log(os.platform());
console.log(os.tmpdir());
console.log(os.type());

// console result
// Node.js v17.0.1
// sumeetpravinjumde@Sumeets-Air osmodule % node osmod.js
// arm64
// 336494592
// 8589934592
// Sumeets-Air
// darwin
// /var/folders/mm/bl1r88_d2p7_vklkh7_lyd80000gn/T
// Darwin
```

## ● Async and Sync

// Async vs Sync File System

// We should all ways prefer async rather than sync be case Async doesn't let other lines of code wait till it complete the other tasks

// Sync way to read

```
const fs = require("fs");
const data = fs.readFileSync("fsAsyncdemo/Async.txt","utf-8");
console.log(data);
```

// Aync Way to read

```
fs.readFile("fsAsyncdemo/Async.txt","utf-8",(err,data)=>{
  console.log(data);
  console.log(err);
});
console.log("after the data");
```

## ● Asynchronous FS

```
const fs = require("fs");
```

// throws Error for this methods so we have to use callback for async function

```

// fs.writeFile("fsAsyncdemo/Async.txt","this is Asynchronous data");
fs.writeFile("fsAsyncdemo/Async.txt","this is Asynchronous data",(err)=>
{
    console.log("files are created");
    console.log(err);
});
// To Append the data
fs.appendFile("fsAsyncdemo/Async.txt"," this is appended statement",(err)=>
{
    console.log("statement is appended");
    console.log(err);
});

// Read file
fs.readFile("fsAsyncdemo/Async.txt","UTF-8",(err,data)=>{
    console.log(data);
});

// Rename
fs.readFile("fsAsyncdemo/Async.txt","fsAsyncdemo/Async.txt",(err)=>{
    console.log("rename is done");
    console.log(err);
});
// Delete file
fs.readFile("fsAsyncdemo/Async.txt",(err)=>{
    console.log("file deleted");
    console.log(err);
});

// remove folder
fs.rmdir("./Async",(err)=>{
    console.log("folder deleted");
    console.log(err);
});

// Notes for Asnc function
// CODE
// const fs = require("fs");
// fs.writeFile("fsAsyncdemo/Async.txt","this is Asynchronous data");

// sumeetpravinjumde@Sumeets-Air fsAsync % cd ..

```

```

// sumeetpravinjumde@Sumeets-Air repl_env % cd fsAsync
// sumeetpravinjumde@Sumeets-Air fsAsync % mkdir fsAsyncdemo
// sumeetpravinjumde@Sumeets-Air fsAsync % type nul > fsAsync.js
// sumeetpravinjumde@Sumeets-Air fsAsync % node fsAsync.js
// node:internal/validators:223
//   throw new ERR_INVALID_CALLBACK(callback);
//   ^
// TypeError [ERR_INVALID_CALLBACK]: Callback must be a function. Received
// undefined
//   at maybeCallback (node:fs:177:3)
//   at Object.writeFile (node:fs:2101:14)
//   at Object.<anonymous> (/Users/sumeetpravinjumde/Documents/Work/
// Coding/Nodejs/repl_env/fsAsync/fsAsync.js:3:4)
//   at Module._compile (node:internal/modules/cjs/loader:1095:14)
//   at Object.Module._extensions.js (node:internal/modules/cjs/
// loader:1147:10)
//   at Module.load (node:internal/modules/cjs/loader:975:32)
//   at Function.Module._load (node:internal/modules/cjs/loader:822:12)
//   at Function.executeUserEntryPoint [as runMain] (node:internal/modules/
// run_main:81:12)
//   at node:internal/main/run_main_module:17:47 {
//   code: 'ERR_INVALID_CALLBACK'
// }
// Node.js v17.0.1
// sumeetpravinjumde@Sumeets-Air fsAsync % node fsAsync.js

// This error because While using aSync function you have to use call back
// Code
// fs.writeFile("fsAsyncdemo/Async.txt","this is Asynchronous data",(err)=>
// {
//   console.log("files are created");
//   console.log(err);
// });
// console
// Node.js v17.0.1
// sumeetpravinjumde@Sumeets-Air fsAsync % node fsAsync.js
// files are created
// null
// sumeetpravinjumde@Sumeets-Air fsAsync %

```

## ● WebServer

```

// To create nodejs Server follow the steps
// require to get current http requires
// 1.It requires http module
// server takes an fun with req and res callback fun

```

// listen is port number where you want server to start so localhost is 127.0.0.1  
and port is whatever you want i have taken port as 3000

```
const http = require("http");
const server = http.createServer((req,res)=>{
  res.end('Hello World Sumeet');
})
server.listen(3000,'127.0.0.1',()=>{
  console.log('listening to port 3000')
});
```

## ● Events Module

Nodejs has built in module called as "Events"

where you can create-,fire- and listen for- your own events.

Ex.1 - registering for the events to be fired only one time using once.

Ex.2 - create an event emitter instance and register a couple of callbacks

Ex.3 - registering for the event with callback parameters

```
const event = require("events");
const EventEmitter = new event();
// declaring event
eventEmitter.on('saymyname',()=>{
  console.log("your name is sumeet");
})
eventEmitter.on('saymyname',()=>{
  console.log("your name is jumde");
})
// event with multiple parameters
eventEmitter.on('checkpage',(sc,msg)=>{
  console.log(` status code is ${sc} and page is ${msg}`)
})
// calling event , we can call multiple event with same event name
eventEmitter.emit('saymyname');
// calling event with multipul arguments
eventEmitter.emit("checkpage",200,"ok");
```

## ● Stream module

```
const fs =require("fs");
const http = require ("http");
const server = http.createServer();
server.on('request',(req,res)=>{
  // normay way
  // fs.readFile('input.txt',(err,data)=>{
  // if(err) return console.log(err);
```

```
// res.end(data.toString());  
// })  
//Stream Way  
const readablestream = fs.createReadStream("input.txt");  
readablestream.on('data',(chunkdata)=>{  
  res.write(chunkdata);  
});  
readablestream.on('end',()=>{  
  res.end();  
})  
readablestream.on('error',(err)=>{  
  console.log(err);  
  res.end("file not found");  
})  
})  
server.listen(8000,'127.0.0.1');
```