**File System (fs) Module**

The syntax for including the fs module in your application:

**var fs=require("fs");**

**Write node Example with File system methods.**

1. **To create folder**
2. **Create one file inside that folder**
3. **Append some data to that file.**
4. **Read data from the file**
5. **Rename that file**
6. **Delete File**

var ps=require("fs");

ps.mkdirSync("node");

ps.writeFileSync("node/write.txt","Hello");

ps.appendFileSync("node/write.txt","Hi");

data=ps.readFileSync("node/write.txt");

console.log(data);

console.log(data.toString());

ps.renameSync("node/write.txt"," node/readwrite.txt")

ps.unlinkSync("node/readwrite.txt");

**Read data from file and sort that data in ascending order.**

//string format

var ps=require("fs");

ps.writeFileSync("s1.txt","50 -1 99 20 0 56 78 59");

data=ps.readFileSync("S1.txt","utf-8");

data=data.split(" ");

data.sort();

console.log(data);

**Output:**

[

'-1', '0', '20',

'50', '56', '59',

'78', '99'

]

//integer format

var ps=require("fs");

ps.writeFileSync("s1.txt","50 -1 99 20 0 56 78 59");

data=ps.readFileSync("S1.txt","utf-8");

data=data.split(" ");

d=data.sort();

let p=[];

for(i=0;i<d.length;i++){

    p[i]=parseInt(d[i]);

}

console.log(p)

**Output:**

[

-1, 0, 20, 50,

56, 59, 78, 99

]

**Write a node.js script to copy contents of one file to another file. Data should be fetched from Source.txt and insert to destination.txt.**

var ps=require("fs");

ps.writeFileSync("source.txt","ABC");

ps.appendFileSync("source.txt","DEF");

data=ps.readFileSync("Source.txt","utf-8");

ps.writeFileSync("destination.txt",data);

data1=ps.readFileSync("destination.txt","utf-8");

console.log(data.toString());

**Output:**

ABCDEF

**Reading the file data from file using Asynchronous mode.**

var ps=require("fs");

ps.readFile("abc.txt",function(err,data)

{

    if(err)

    {

        return console.error(err);

    }

    console.log(data.toString());

    console.error("completed");

}

);

console.log("Program ended");

**Output:**

Program ended

Today is a good dayToday is a good day

completed

**Writing data to file, appending data to file and then reading the file data using Asynchronous mode.**

var fs=require("fs");

fs.writeFile("abc.txt","Today is a good day",(err)=> {if(err){console.log("completed")}});

fs.appendFile("abc.txt","Today is a good day",function(err)

{

if(err){console.log("completed")

}

});

fs.readFile("abc.txt",(err,data)=>{

if(err){

console.error(err);

}

console.log(data.toString())

});

console.log("File Operations ended")

**Output:**

File Operations ended

Today is a good dayToday is a good day

**Defining an array of object with properties name and age. Write this object in a file named student.txt then read the file and display the object on console.**

const student =

    [

        {

            name: "ABC",

            age: 30

        },

        {

            name: "XYZ",

            age: 32

        }

    ]

var ps=require("fs");

ps.writeFileSync("student.txt",JSON.stringify(student));

data=ps.readFileSync("student.txt","utf-8");

b=JSON.parse(data);

console.log(b);

**Output:**

[ { name: 'ABC', age: 30 }, { name: 'XYZ', age: 32 } ]

**Create JSON object which contains array of objects. Calculate perimeter of square and perimeter of circle by using side value and diameter value respectively. And**

const shape =

    [

        {

            name: "circle",

            diameter: 8

        },

        {

            name: "square",

            side: 10

        }

    ]

var ps=require("fs");

ps.writeFileSync("shape.txt",JSON.stringify(shape));

data=ps.readFileSync("shape.txt","utf-8");

b=JSON.parse(data);

if( b[0].name == 'circle'){

    var perimeter = (b[0].diameter/2) \* 3.14 \* 2 ;

    console.log(perimeter);

}

if ( b[1].name == 'square'){

    var peri = (b[1].side) \*4  ;

    console.log(peri);

}

ps.appendFileSync("shape.txt","\nPerimeter of circle = "+ JSON.stringify(perimeter)+ "\nPerimeter of square = "+JSON.stringify(peri));

**Output:**

[{"name":"circle","diameter":8},{"name":"square","side":10}]

Perimeter of circle = 25.12

Perimeter of square = 40

**Write node.js script to create a class named person by assigning name and age in form of members. Create one function named elder which returns elder person object. Details of elder person should be printed in console as well as in file.**

class person

{

    constructor(name,age)

    {

        this.age=age;

        this.name=name;

    }

    elder(P)

    {

        if(this.age>P.age)

        {

            return this;

        }

        else{

            return P;

        }

    }

}

var p1= new person("xyz",23);

var p2= new person("abc",34);

var p3=p1.elder(p2);

const jsonstr=JSON.stringify(p3);

var ps=require("fs");

ps.writeFileSync("d2.txt",jsonstr);

**Output:**

person { age: 34, name: 'abc' }

**Write node.js script to create a class named time and assign members hour, minute and second. Create two objects of time class and add both the time objects so that it should return the value in third time object. The third time object should have hour , minute and second such that after addition if seconds exceed 60 then minute value should be incremented and if minute exceed 60 then hour value should be incremented. The value should be printed in console as well as in file.**

class time

{

    constructor(hour,min,sec)

    {

        this.hour=hour;

        this.min=min;

        this.sec=sec;

    }

    timer(p)

    {

        var t=new time();

        t.hour=this.hour+p.hour;

        t.min=this.min+p.min;

        t.sec=this.sec+p.sec;

        if(t.sec>60)

        {

            t.sec%=60;

            t.min++;

        }

        if(t.min>60)

        {

            t.min%=60;

            t.hour++;

        }

        return t;

    }

}

var t1= new time(1,50,50);

var t2= new time(2,30,50);

var t3=t1.timer(t2);

console.log(t3);

const jsonstr=JSON.stringify(t3);

var ps=require("fs");

ps.writeFileSync("time.txt",jsonstr);

**Output:**

**time { hour: 4, min: 21, sec: 40 }**

**OS Module : Operating System**

**Get information about the computer's operating system:**

The syntax for including the os module in your application:

**var os=require("os");**

**Write node.js script to create a folder named “AA” at temp folder. Also, create file named “temp.txt” inside “AA” folder. Now, check if available physical memory of the system is greater than 1 GB then print message “Sufficient Memory” in the file, else print message “Low Memory” in file.**

var ps=require("fs");

var os=require("os");

console.log(os.arch());

console.log(os.hostname());

console.log(os.platform());

console.log(os.tmpdir());

f = os.tmpdir();

freemem=os.freemem()/1024/1024/1024;

ps.mkdirSync(f+"/AA");

if(freemem > 1){

ps.writeFileSync(f+"/AA/temp.txt","Sufficient memory")

}

else{

    ps.writeFileSync(f+"/AA/temp.txt","Low memory")

}

**Output:**

x64

ITICT406-182

win32

C:\Users\LJIET\AppData\Local\Temp

**Write node.js script to create a folder named “AA” at temp folder. Also, create file named “temp1.txt” inside “AA” folder. Now, check if available physical memory of the system is greater than 1 GB then print message “Sufficient Memory” in the file, else print message “Low Memory” in file.**

var ps=require("fs");

var os=require("os");

console.log(os.platform());

f = os.tmpdir();

p = os.platform();

if(p ==  "win32"){

ps.writeFileSync(f+"/AAAA/temp1.txt","You are working on windows 32 bit")

}

else{

    ps.writeFileSync(f+"/AAAA/temp.txt","You are working on windows 64 bit")

}

**Path Module**

The Path module provides a way of working with directories and file paths.

The syntax for including the path module in your application:

**var os=require("path");**

**Write node.js script to check whether the file extension is .txt or not.**

var pm=require("path");

path=pm.dirname("D:/LJ/abc.html");

console.log(path);

path=pm.basename("D:/LJ/abc.txt");

console.log(path);

ext = pm.extname("D:/LJ/abc.txt")

console.log(ext);

path=pm.parse("D:/LJ/abc.html");

console.log(path);

if(path.ext == ".txt"){

    console.log("Text Document");

}else{

    console.log("Not a text Document");

}

**Output:**

D:/LJ

abc.txt

.txt

{

root: 'D:/',

dir: 'D:/LJ',

base: 'abc.html',

ext: '.html',

name: 'abc'

}

Not a text Document