

# CREATE A BIG FILE

→ in every new folder, installation of minimist is required

JS 5\_CreateBigFile.js U ×

Lecture 23 > JS 5\_CreateBigFile.js > ...

```
1 // npm installl minimist
2 // node 5_CreateBigFile.js --dest=big.data
3
4 let minimist = require("minimist");
5 let fs = require("fs");
6
7 let args = minimist(process.argv);
8
9 let arr = [];
10 for(let i = 0; i < 50; i++){
11     arr.push(i);
12 }
13
14 let str = arr.join(",");
15 console.log(str);
```

The join() method returns an array as a string.

The elements will be separated by a specified separator. The default is comma (,).

join() does not change the original array.

→ Lecture 23 git:(main) x node 5\_CreateBigFile.js --dest=big.data

0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49

JS 5\_CreateBigFile.js U ×

Lecture 23 > JS 5\_CreateBigFile.js > [⌘] str

```
1 // npm install minimist
2 // node 5_CreateBigFile.js --dest=big.data
3
4 let minimist = require("minimist");
5 let fs = require("fs");
6
7 let args = minimist(process.argv);
8
9 let arr = [];
10 for(let i = 0; i < 50; i++){
11     arr.push(i);
12 }
13
14 let str = arr.join("\n");
15 console.log(str);
```

Syntax

array.join(separator)

using "\n" as separator

→ Lecture 23 git:(main) x node 5\_CreateBigFile.js --dest=big.data

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10

```
JS 5_CreateBigFile.js U × ... big.data U ×
Lecture 23 > JS 5_CreateBigFile.js > ...
1 // npm install minimist
2 // node 5_CreateBigFile.js --dest=big.data
3
4 let minimist = require("minimist");
5 let fs = require("fs");
6
7 let args = minimist(process.argv);
8
9 let arr = [];
10 for(let i = 0; i < 50; i++){
11     arr.push(i);
12 }
13
14 let str = arr.join("\n");
15
16 fs.writeFileSync(args.dest, str, "utf-8");
17 fs.appendFileSync(args.dest, str, "utf-8");
18 fs.appendFileSync(args.dest, str, "utf-8");
19 fs.appendFileSync(args.dest, str, "utf-8");
20
21
22
23
176 28
177 29
178 30
179 31
180 32
181 33
182 34
183 35
184 36
185 37
186 38
187 39
188 40
189 41
190 42
191 43
192 44
193 45
194 46
195 47
196 48
197 49
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
→ Lecture 23 git:(main) x node 5_CreateBigFile.js --dest=big.data
```

The 'fs' module of Node.js implements the File I/O operation. Methods in the fs module can be synchronous as well as asynchronous. The Asynchronous function has a callback function as the last parameter which indicates the completion of the asynchronous function.

Asynchronous methods are preferred over synchronous methods as asynchronous methods never block a program during its execution, whereas the latter does. Blocking the main thread is malpractice in Node.js, thus synchronous functions should only be used for debugging or when no other options are available.

The fs.writeFileSync() is a synchronous method. The fs.writeFileSync() creates a new file if the specified file does not exist. Also the 'readline-sync' module is used to enable user input at runtime.

### Syntax

<https://www.geeksforgeeks.org/node-js-fs-writefilesync-method/>

fs.writeFileSync (file, data, options)

## fs.appendFileSync() function

The `fs.appendFileSync()` method is used to synchronously append the given data to a file. A new file is created if it does not exist. The optional option parameter can be used to modify the behavior of the operation.

Syntax:

`fs.appendFileSync(path, data, options)`

<https://www.geeksforgeeks.org/node-js-fs-appendfilesync-function/>

JS 5\_CreateBigFile.js U X

Lecture 23 > JS 5\_CreateBigFile.js > ...

```
1 // npm install minimist
2 // node 5_CreateBigFile.js --dest=big.data
3
4 let minimist = require("minimist");
5 let fs = require("fs");
6
7 let args = minimist(process.argv);
8
9 let arr = [];
10 for(let i = 0; i < 5000000; i++){
11     arr.push(i);
12 }
13
14 let str = arr.join("\n");
15
16 fs.writeFileSync(args.dest, str, "utf-8");
17 fs.appendFileSync(args.dest, str, "utf-8");
18 fs.appendFileSync(args.dest, str, "utf-8");
19 fs.appendFileSync(args.dest, str, "utf-8");
20
```

around 1.7 GB file is created.

To install minimist globally, type:-  
`npm install minimist -g` → global installation