

# Getting Started

## FUNCTIONS

[https://www.w3schools.com/js/js\\_functions.asp](https://www.w3schools.com/js/js_functions.asp)

JS FirstFunction.js U X

Javascript > Lecture 21 > JS FirstFunction.js > IsPrime

```
1 let clargs = process.argv;
2
3 let n = parseInt(clargs[2]);
4
5 for(let i = 2; i <= n; i++){
6   let isPrime = IsPrime(i);
7   if(isPrime == true){
8     console.log(i);
9   }
10 }
11
12 function IsPrime(x){
13   let isPrime = true;
14   for(let div = 2; div * div <= x; div++){
15     if(x % div == 0){
16       isPrime = false;
17       break;
18     }
19   }
20   return isPrime;
21 }
```

// function is called

← // function is created

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 21 git:(main) x node FirstFunction.js 20
2
3
5
7
11
13
17
19
```

# ARRAYS

JS FirstArray.js U X

JavaScript > Lecture 21 > JS FirstArray.js > ...

```
1
2
3 let clargs = process.argv;
4
5 let n = parseInt(clargs[2]);
6
7 let arr = [];
8 for(let i = 0; i < n; i++){
9   let val = parseInt(clargs[i + 3]);
10  arr.push(val);
11 }
12 OR arr[i] = val
13 for(let i = 0; i < arr.length; i++){
14   console.log(arr[i]);
15 }
16
17 console.log(arr);
18 console.log(arr.length);
```

node FirstArray.js 5 10 20 30 40 50  
0 1 2 3 4 5 6 7  
number conversion

Array creation

val variable is created and it will get the input from  $i+n$  place

pushing the value val in the array.

Same thing

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

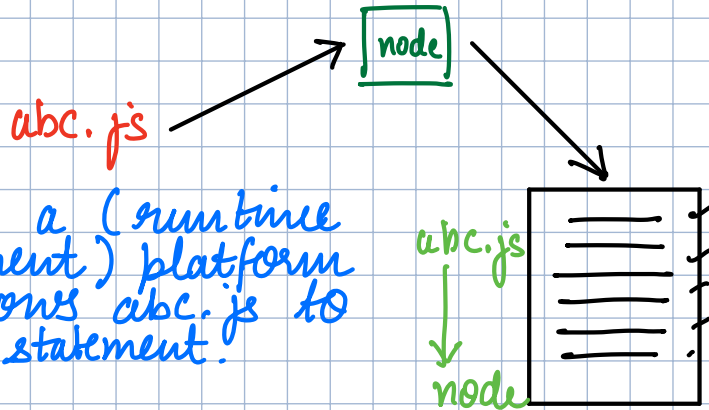
```
+ Lecture 21 git:(main) x node FirstArray.js 5 10 20 30 40 50
10
20
30
40
50
5
[ 10, 20, 30, 40, 50 ]
```

single line array view  
(similar to array list)

[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp)

# DEPENDENCIES (external module)

Node is a process that runs the javascript file.



- Node is a (runtime environment) platform that allows abc.js to run its statement.

- Node is a compiler / interpreter / runtime environment (because it has libraries in it).

## # NPM: Node Package Manager

All the libraries in the node is present in NPM.

NPM is a downloader of library/libraries.

To install a library:-

— on the terminal; type

example: "npm install minimist" [minimist to read command line.]  
library

Syntax: "npm install libraries\_name"

<https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
→ Lecture 21 git:(main) x npm install minimist
npm WARN saveError ENOENT: no such file or directory, open '/Users/hemakshipandey/package.json'
npm WARN enoent ENOENT: no such file or directory, open '/Users/hemakshipandey/package.json'
npm WARN hemakshipandey No description
npm WARN hemakshipandey No repository field.
npm WARN hemakshipandey No README data
npm WARN hemakshipandey No license field.

+ minimist@1.2.5
added 1 package from 1 contributor and audited 5 packages in 0.659s
found 0 vulnerabilities
```

To include an installed library in javascript

```
JS FirstModule.js U X
Javascript > Lecture 21 > JS FirstModule.js > ...
1 // npm install minimist
2 let parser = require('minimist');
3 let args = parser(process.argv);
4
5 console.log(args);
6
7 //node FirstModule.js -x 4 -y 10
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 21 git:(main) x node FirstModule.js -x 4 -y 10
{
  _: [
    '/usr/local/bin/node',
    '/Users/hemakshipandey/Desktop/Pepcoding-FJP1-Development/Java
  ],
  x: 4,
  y: 10
}
```

```
JS FirstModule.js U X
Javascript > Lecture 21 > JS FirstModule.js > ...
1 // npm install minimist
2 let parser = require('minimist');
3 let args = parser(process.argv);
4
5 let name = args.name;
6 let age = args.age;
7
8 if(age > 30){
9   console.log("Hello " + name + " . You must go home.");
10 }else{
11   console.log("Heya " + name + " . Where is the party tonight.");
12 }
13
14 //node FirstModule.js -x 4 -y
15
16 //node FirstModule.js --name="Hemakshi Pandey" --age=23
17
18
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 21 git:(main) x node FirstModule.js --name="Hemakshi Pandey" --age=23
Heya Hemakshi Pandey . Where is the party tonight.
→ Lecture 21 git:(main) x
```

// This represent  
args.name  
and args.age  
respectively

JS FirstModule.js U X

Javascript > Lecture 21 > JS FirstModule.js > ...

```
1 // npm installl minimist
2 let parser = require('minimist');
3 let args = parser(process.argv);
4
5 if(args.age > 30){
6     console.log("Hello " + args.name + " . You must go home.");
7 }else{
8     console.log("Heya " + args.name + " . Where is the party tonightargs..");
9 }
10
11 //node FirstModule.js -x 4 -y
12
13 //node FirstModule.js --name="Hemakshi Pandey" --age=23
14
15
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
+ Lecture 21 git:(main) x node FirstModule.js --name="Hemakshi Pandey" --age=23
Heya Hemakshi Pandey . Where is the party tonight.
+ Lecture 21 git:(main) x node FirstModule.js --name="Hemakshi Pandey" --age=35
Hello Hemakshi Pandey . You must go home.
+ Lecture 21 git:(main) x
```

JS FirstModule.js U X

Javascript > Lecture 21 > JS FirstModule.js > ...

```
1 // npm install minimist
2 let parser = require('minimist');
3 let args = parser(process.argv);
4
5 console.log(args);
6
7 if(args.age > 30){
8     console.log("Hello " + args.name + " . You must go home.");
9 }else{
10     console.log("Heya " + args.name + " . Where is the party tonight.");
11 }
12
13 //node FirstModule.js -x 4 -y
14
15 //node FirstModule.js --name="Hemakshi Pandey" --age=23
16
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
+ Lecture 21 git:(main) x node FirstModule.js --name="Hemakshi Pandey" --age=23
{
  _: [
    '/usr/local/bin/node',
    '/Users/hemakshipandey/Desktop/Pepcoding-FJP1-Development/Javascript/Lecture 21/FirstModule.js'
  ],
  name: 'Hemakshi Pandey',
  age: 23
}
Heya Hemakshi Pandey . Where is the party tonight.
```

} object

# FILE MANIPULATION

```
JS FirstFiles.js U x f1.txt U
Javascript > Lecture 21 > JS FirstFiles.js > ...
1
2 // read a file capitalize every word, write the file
3
4 let minimist = require("minimist");
5 let args = minimist(process.argv);
6
7 // node FirstFile.js --source=f1.txt --dest=f2.txt
8 let fs = require("fs");
9 let stext = fs.readFileSync(args.source, "utf-8");
10 console.log(stext);
```

→ create f1.txt in the folder and enter some text to it.

→ file system = fs (preinstalled in node)

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
→ Lecture 21 git:(main) x node FirstFiles.js --source=f1.txt --dest=f2.txt
cricket is a great game.
it is also called gentlemen's game
it's origin came in england
england recently lost to india
→ Lecture 21 git:(main) x
```

```
JS FirstFiles.js U x f1.txt U
Javascript > Lecture 21 > JS FirstFiles.js > ...
1
2 // read a file capitalize every word, write the file
3
4 let minimist = require("minimist");
5 let args = minimist(process.argv);
6
7 // node FirstFile.js --source=f1.txt --dest=f2.txt
8 let fs = require("fs");
9 let stext = fs.readFileSync(args.source, "utf-8");
10 //console.log(stext);
11
12 let words = stext.split(" ");
13 console.log(words);
```

→ to convert in text form

```
→ Lecture 21 git:(main) x node FirstFiles.js --source=f1.txt --dest=f2.txt
[
  'cricket',    'is',
  'a',          'great',
  'game.\nit',  'is',
  'also',       'called',
  'gentlemen's', 'game\nit's',
  'origin',     'came',
  'in',         'england\nengland',
  'recently',   'lost',
  'to',         'india'
]
```

→ conversion of text file to array

JS FirstFiles.js U × ≡ f1.txt U

Javascript > Lecture 21 > JS FirstFiles.js > ...

```
1
2 // read a file capitalize every word, write the file
3
4 let minimist = require("minimist");
5 let args = minimist(process.argv);
6
7 // node FirstFile.js --source=f1.txt --dest=f2.txt
8 let fs = require("fs");
9 let stext = fs.readFileSync(args.source, "utf-8");
10 //console.log(stext);
11
12 let words = stext.split(" ");
13 //console.log(words);
14
15 for(let i = 0; i < words.length; i++){
16     words[i] = words[i].toUpperCase();
17 }
18
19 let dtext = words.join(" ");
20 console.log(dtext);
21
```

capitalize every word

conversion of text file back from array.

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 21 git:(main) x node FirstFiles.js --source=f1.txt --dest=f2.txt
CRICKET IS A GREAT GAME.
IT IS ALSO CALLED GENTLEMEN'S GAME
IT'S ORIGIN CAME IN ENGLAND
ENGLAND RECENTLY LOST TO INDIA
→ Lecture 21 git:(main) x
```

JS FirstFiles.js U ×

≡ f1.txt U ×

≡ f2.txt U ×

Javascript > Lecture 21 > JS FirstFiles.js > ...

```
1
2 // read a file capitalize every word, write the file
3
4 let minimist = require("minimist");
5 let args = minimist(process.argv);
6
7 // node FirstFile.js --source=f1.txt --dest=f2.txt
8
9 // args.source (f1.txt) and args.dest (f2.txt)
10
11 let fs = require("fs");
12 let stext = fs.readFileSync(args.source, "utf-8");
13 //console.log(stext);
14
15 let words = stext.split(" "); //string has split
16 //console.log(words);
17
18 for(let i = 0; i < words.length; i++){
19     words[i] = words[i].toUpperCase();
20 }
21
22 let dtext = words.join(" "); //array has join
23 //console.log(dtext);
24 fs.writeFileSync(args.dest, dtext, "utf-8");
25
```

Javascript > Lecture 21 > ≡ f1.txt

```
1 cricket is a great game.
2 it is also called gentlemen's game
3 it's origin came in england
4 england recently lost to india
```

Javascript > Lecture 21 > ≡ f2.txt

```
1 CRICKET IS A GREAT GAME.
2 IT IS ALSO CALLED GENTLEMEN'S GAME
3 IT'S ORIGIN CAME IN ENGLAND
4 ENGLAND RECENTLY LOST TO INDIA
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 21 git:(main) x node FirstFiles.js --source=f1.txt --dest=f2.txt
→ Lecture 21 git:(main) x
```

zsh + v