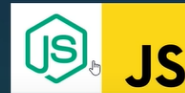


What is Node.js?

Front-end (browser)

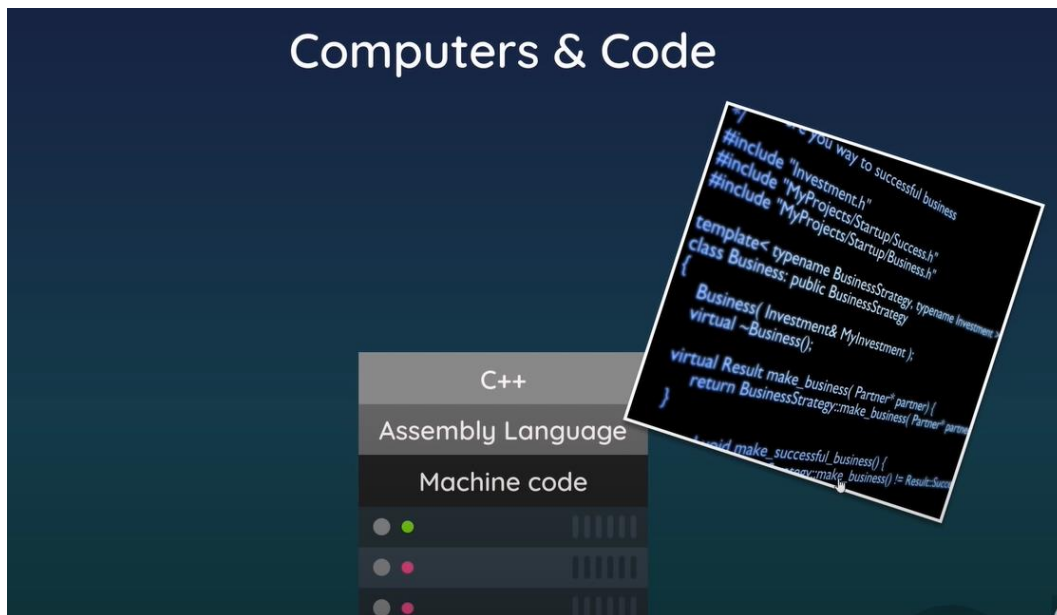


Back-end (server)



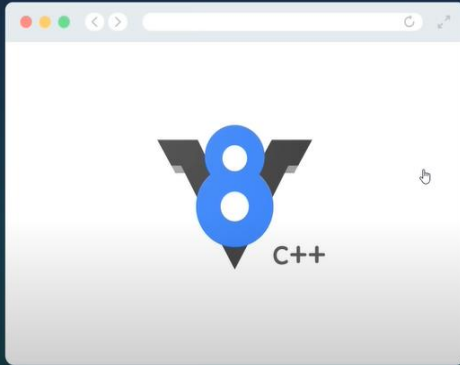
- We cannot run javascript inside computer or browser but node js helps us to run javascript inside computer or browser.

Computers & Code



- V8 engine helps javascript to run in browser. V8 engine works on browser. V8 engine is written in C++.

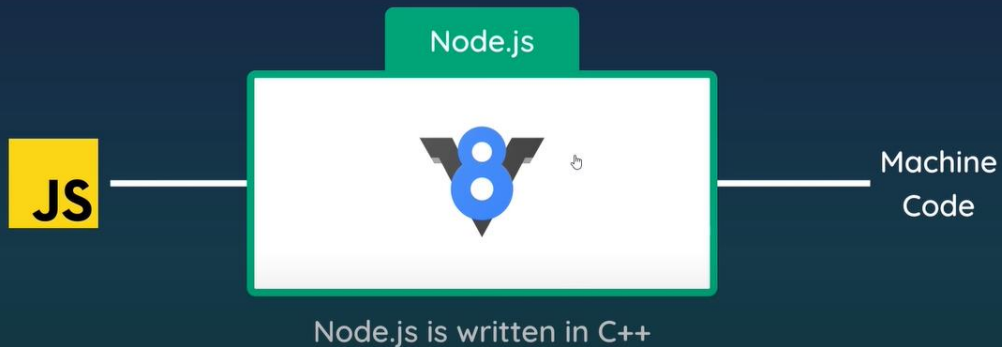
JavaScript in Browsers



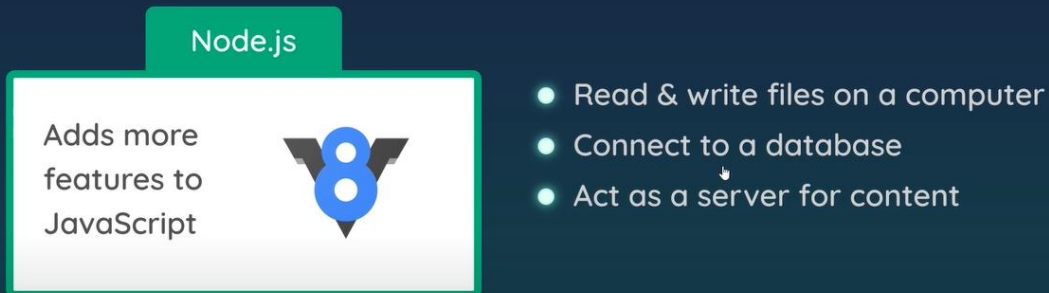
- V8 engine compiles JavaScript into machine code

- Node js written in C++. So it can run javascript in computer or browser. Because V8 engine is installed inside node js.

Enter Node.js

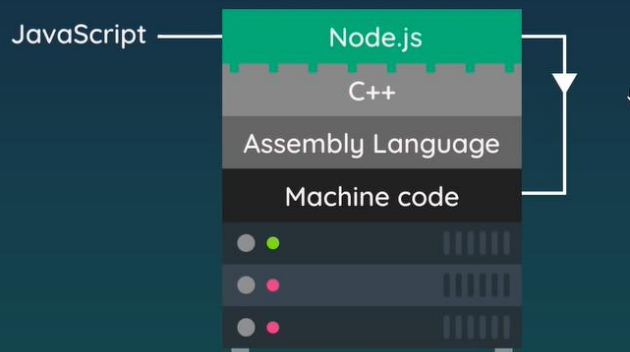


Node is more than v8...

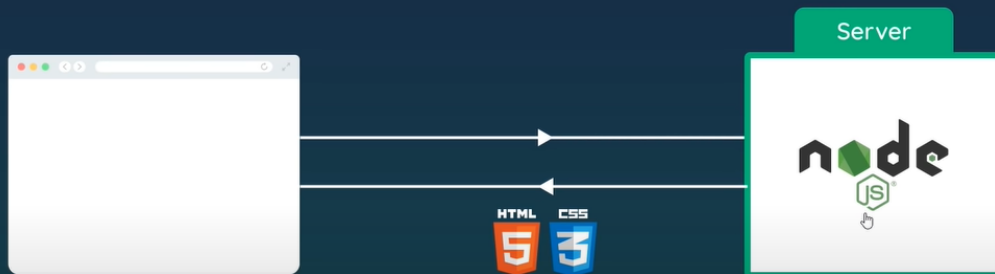


- **Now we can run javascript file in our computer with help of node js. So that computers can understand javascript.**

Computers & Code



The Role of Node.js



But why use Node.js?

- No need to learn an extra language for server
- Can share code between front and back end
- Node.js has a massive community behind it
- Huge amount of third-party packages & tools to help

In this course...

- How to install Node & use it to run JavaScript
- How to read & write files on your computer
- How to create a server using Node.js to create a website
- How to create an Express app / website
- How to use MongoDB (a NoSQL database)
- How to use template engines to easily create HTML views
- Put everything together to make a simple blog site

```
Command Prompt
C:\Users\Shaun>node -v
v14.1.0

C:\Users\Shaun>node
Welcome to Node.js v14.1.0.
Type ".help" for more information.
> 5 + 5
10
> var name = 'mario'
undefined
> name
'mario'
>
(To exit, press ^C again or ^D or type .exit)
>

C:\Users\Shaun>cd documents

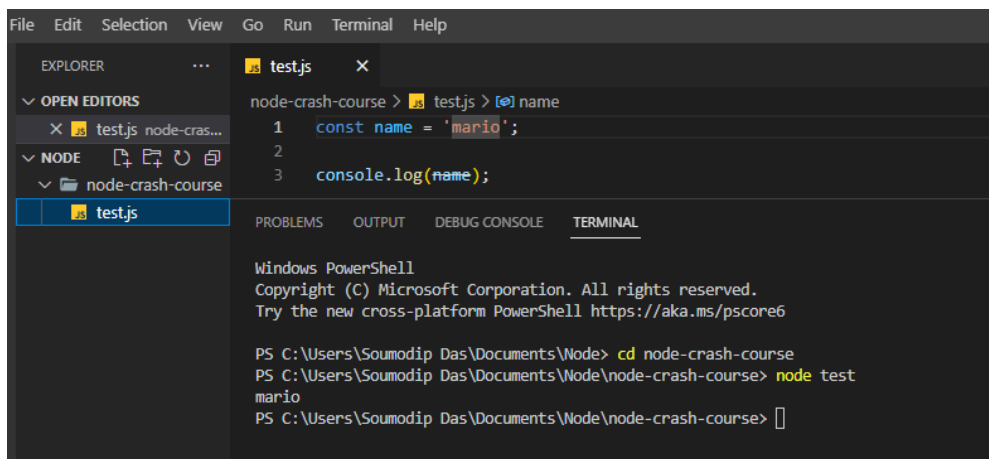
C:\Users\Shaun\Documents>cd tuts

C:\Users\Shaun\Documents\Tuts>mkdir node-crash-course

C:\Users\Shaun\Documents\Tuts>cd node-crash-course

C:\Users\Shaun\Documents\Tuts\node-crash-course>code .
```

- For accessing this we need to create a new file inside documents folder in this PC.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer pane shows a file named `test.js` inside the `node-crash-course` folder. The main editor area displays the content of `test.js`:

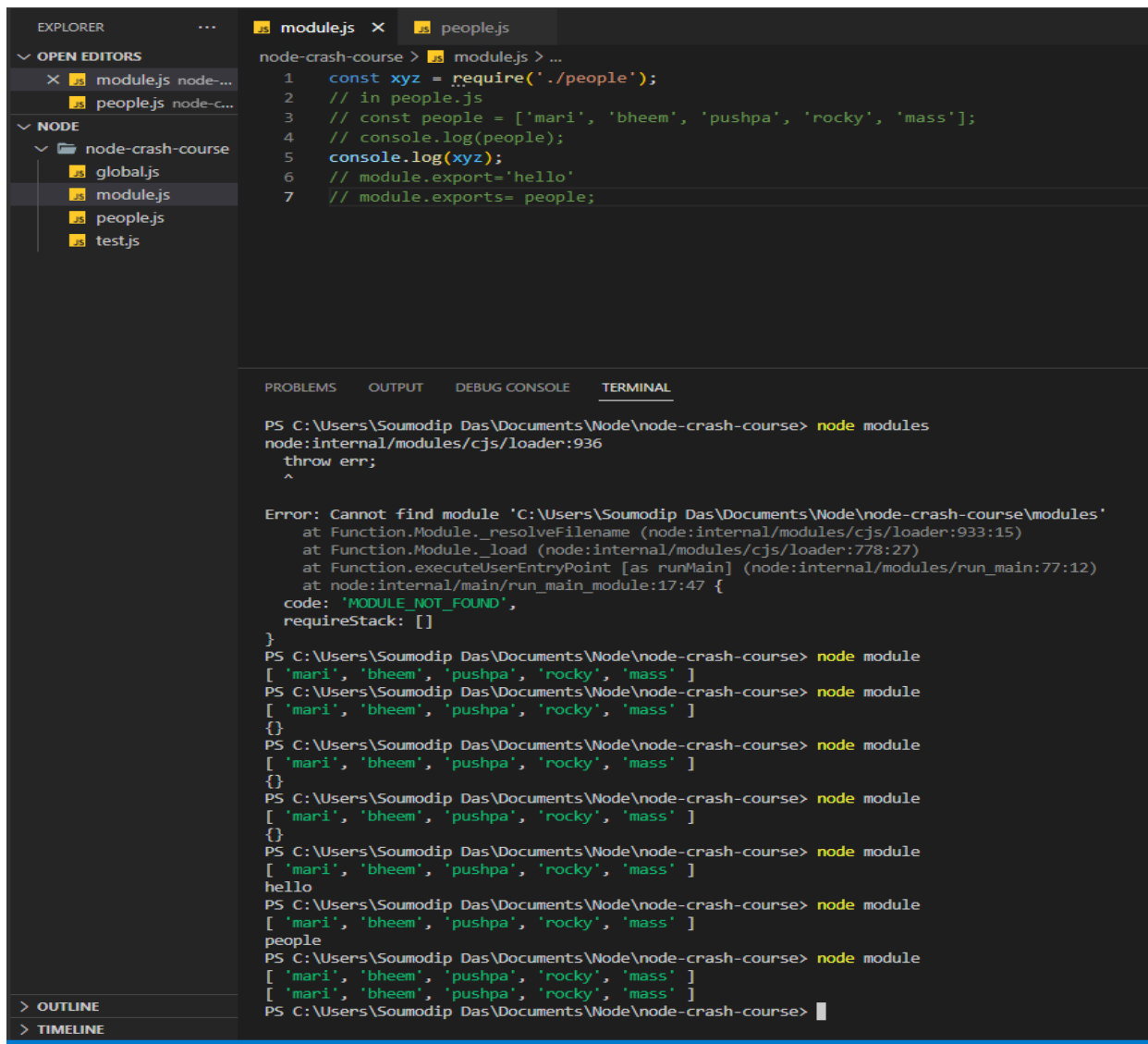
```
node-crash-course > test.js > name
1  const name = 'mario';
2
3  console.log(name);
```

At the bottom, the Terminal pane shows a Windows PowerShell session:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Soumodip Das\Documents\Node> cd node-crash-course
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node test
mario
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course>
```


Modules & Require



```
node-crash-course > module.js > ...
1  const xyz = require('./people');
2  // in people.js
3  // const people = ['mari', 'bheem', 'pushpa', 'rocky', 'mass'];
4  // console.log(people);
5  console.log(xyz);
6  // module.export='hello'
7  // module.exports= people;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node modules
node:internal/modules/cjs/loader:936
  throw err;
  ^

Error: Cannot find module 'C:\Users\Soumodip Das\Documents\Node\node-crash-course\modules'
    at Function.Module._resolveFilename (node:internal/modules/cjs/loader:933:15)
    at Function.Module._load (node:internal/modules/cjs/loader:778:27)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:77:12)
    at node:internal/main/run_main_module:17:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
{}
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
{}
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
{}
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
hello
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
people
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course>
```

EXPLORER

OPEN EDITORS

- module.js node-...
- people.js node-c...

NODE

- node-crash-course
 - global.js
 - module.js
 - people.js
 - test.js

```
node-crash-course > module.js > ...
1  const {people,ages} = require('./people');
2  // in people.js
3  // const people = ['mari', 'bheem', 'pushpa', 'rocky', 'mass'];
4  // const ages = [10, 12, 52, 25, 41];
5  // console.log(people);
6  // module.exports=
7  // {
8  //   people, ages
9  // };
10 // console.log(xyz.people, xyz.ages);
11 // module.export='hello'
12 // module.exports= people;
13 console.log(people,ages);
14 const os = require('os');
15 console.log(os.platform(), os.homedir());
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
{
  people: [ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ],
  ages: [ 10, 12, 52, 25, 41 ]
}
PS C:\Users\Soumodip Das\Documents\Wode\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ] [ 10, 12, 52, 25, 41 ]
PS C:\Users\Soumodip Das\Documents\Wode\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
PS C:\Users\Soumodip Das\Documents\Wode\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ] [ 10, 12, 52, 25, 41 ]
PS C:\Users\Soumodip Das\Documents\Wode\node-crash-course> node module
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ]
[ 'mari', 'bheem', 'pushpa', 'rocky', 'mass' ] [ 10, 12, 52, 25, 41 ]
{
  arch: [Function: arch] {
    [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
  },
  cpus: [Function: cpus],
  endianness: [Function: endianness] {
    [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
  },
  freemem: [Function: getFreeMem] {
    [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
  },
  getPriority: [Function: getPriority],
  homedir: [Function: __node_internal_checkError] {
    [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
  },
  hostname: [Function: __node_internal_checkError] {
    [Symbol(Symbol.toPrimitive)]: [Function (anonymous)]
  },
  loadavg: [Function: loadavg],
```

The File System


```
node-crash-course > file.js > ...
1  const fs = require('fs');
2  //reading files
3  fs.readFile('./docs/blog1.txt', (err, data) => {
4    if (err) {
5      console.log(err);
6    }
7    // console.log(data);
8    console.log(data.toString());
9  });
10 console.log('last line');
11
12 //writing files
13
14
15 //directories
16
17
18 //deleting files
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

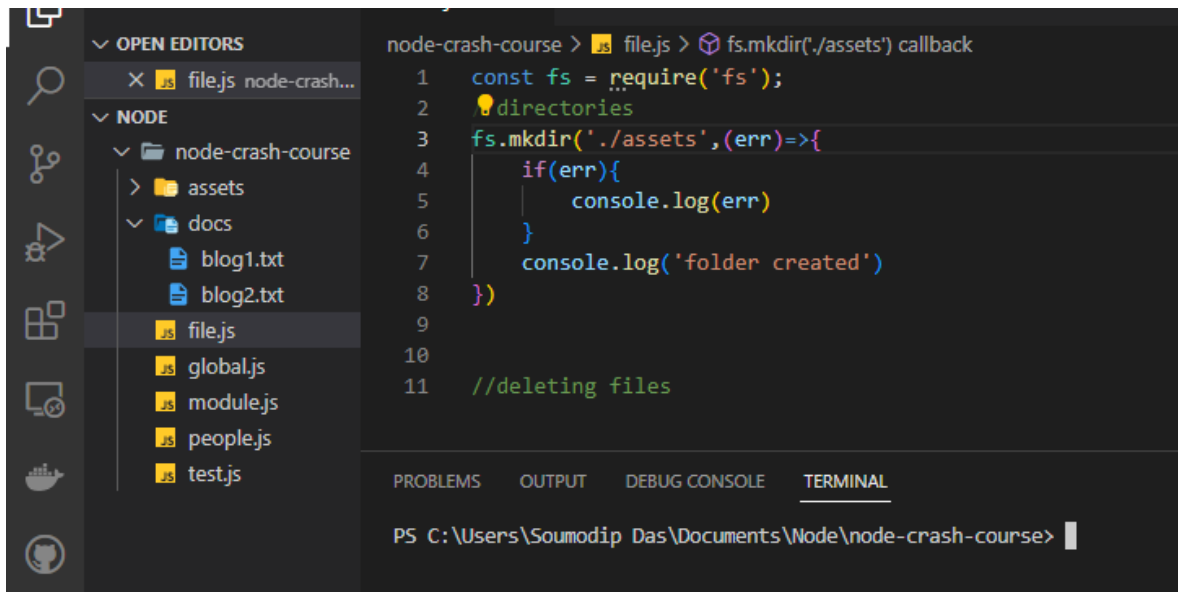
```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
<Buffer 68 65 6c 6c 6f 2c 20 72 6f 63 6b 79 20 62 68 61 69 20 61 6e 64 20 70 75 73 68 70 61 20>
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
hello, rocky bhai and pushpa
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
last line
hello, rocky bhai and pushpa
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> 
```

```
node-crash-course > file.js > ...
1  const fs = require('fs');
2  //writing files
3  fs.writeFile('./docs/blog1.txt', 'hello, mari', () => {
4    console.log('file was written');
5  });
6  fs.writeFile('./docs/blog2.txt', 'hello, rocky', () => {
7    console.log('file was written');
8  });
9  //directories
10
11
12 //deleting files
```

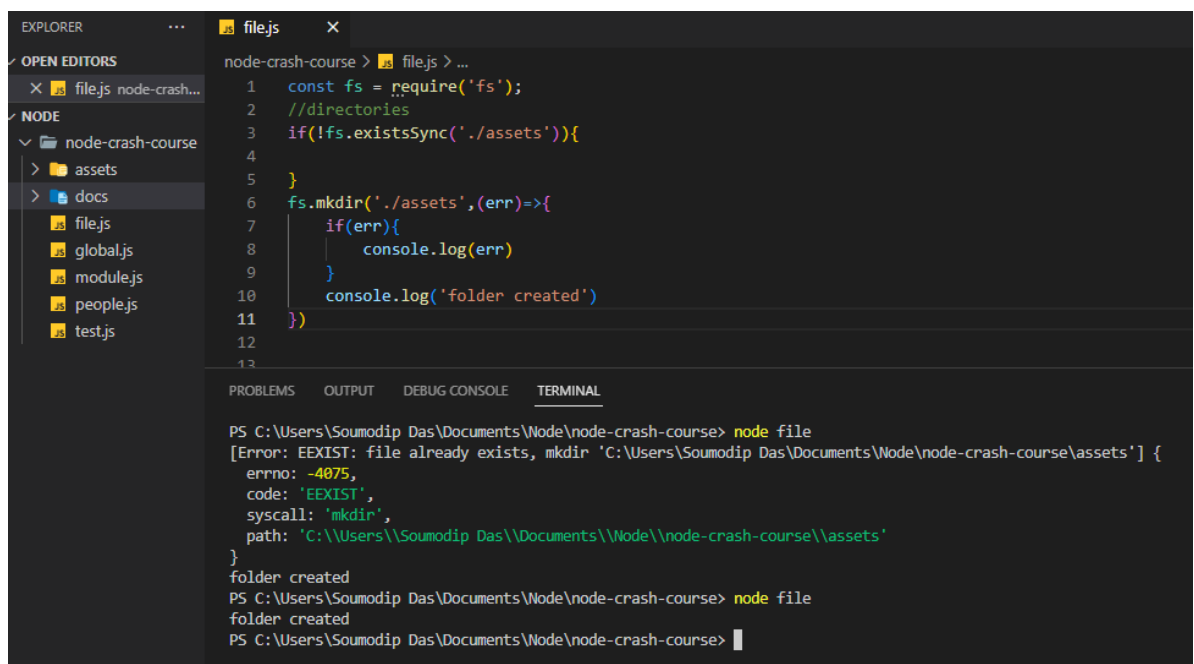
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

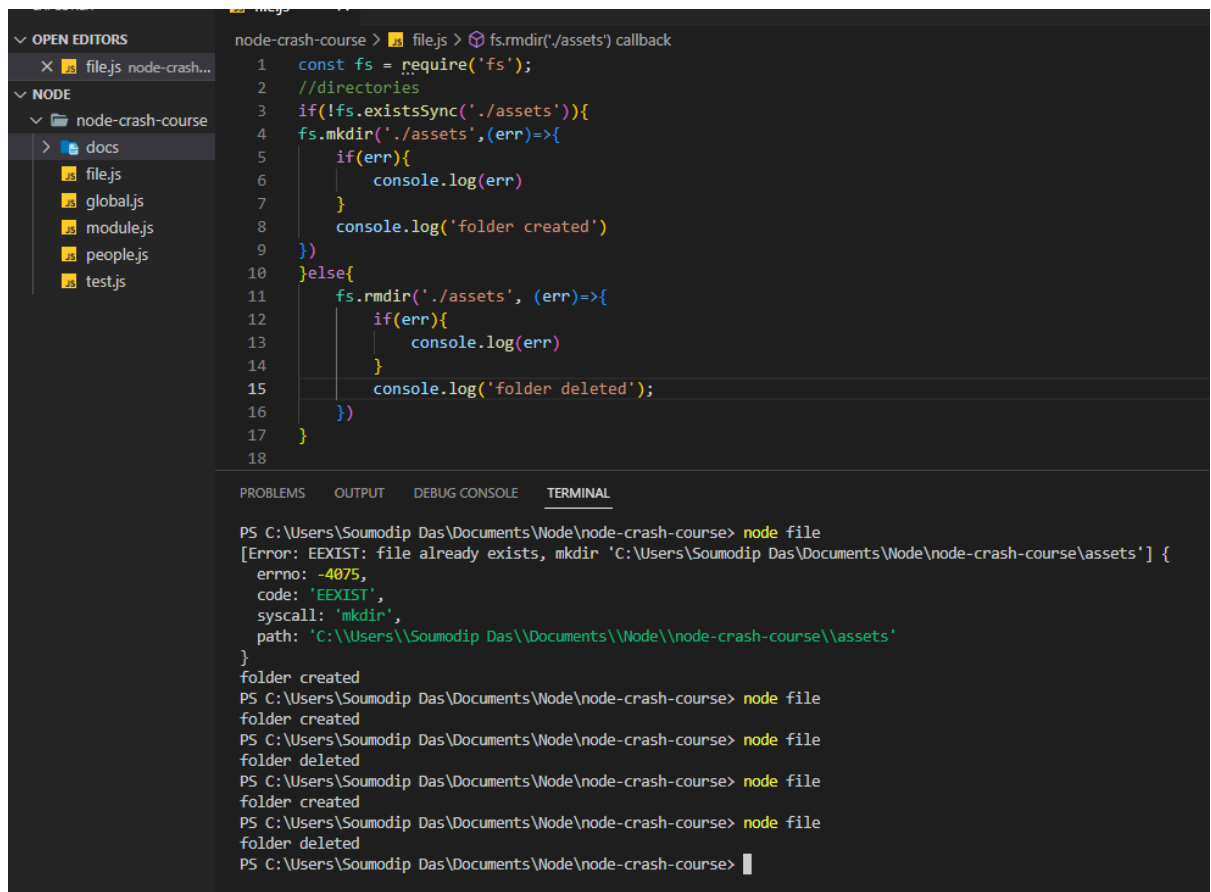
```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
<Buffer 68 65 6c 6c 6f 2c 20 72 6f 63 6b 79 20 62 68 61 69 20 61 6e 64 20 70 75 73 68 70 61 20>
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
hello, rocky bhai and pushpa
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
last line
hello, rocky bhai and pushpa
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
file was written
file was written
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> 
```

- **Blog2 automatically created in docs**



- We can see a new folder name assets has been created





The image shows a Visual Studio Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'node-crash-course' with files 'file.js', 'global.js', 'module.js', 'people.js', and 'test.js'. The code editor shows a JavaScript file named 'file.js' with the following code:

```
node-crash-course > file.js > fs.rmdir('./assets') callback
1  const fs = require('fs');
2  //directories
3  if(!fs.existsSync('./assets')){
4    fs.mkdir('./assets',(err)=>{
5      if(err){
6        console.log(err)
7      }
8      console.log('folder created')
9    })
10 }else{
11   fs.rmdir('./assets', (err)=>{
12     if(err){
13       console.log(err)
14     }
15     console.log('folder deleted');
16   })
17 }
18 }
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
[Error: EEXIST: file already exists, mkdir 'C:\Users\Soumodip Das\Documents\Node\node-crash-course\assets'] {
  errno: -4075,
  code: 'EEXIST',
  syscall: 'mkdir',
  path: 'C:\Users\Soumodip Das\Documents\Node\node-crash-course\assets'
}
folder created
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
folder created
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
folder deleted
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
folder created
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course> node file
folder deleted
PS C:\Users\Soumodip Das\Documents\Node\node-crash-course>
```

- First assets was deleted , then if we again say node files, then the assets file has been again created

The screenshot shows the VS Code interface with the Explorer sidebar on the left and the Editor on the right. The Explorer sidebar shows the file structure of a project named 'node-crash-course', including a 'docs' folder with 'blog1.txt', 'blog2.txt', and 'delete.txt'. The Editor displays the 'file.js' file with the following code:

```
node-crash-course > file.js > ...
1  const fs = require('fs');
2  //deleting files
3  if(fs.existsSync('./docs/delete.txt')){
4      fs.unlink('./docs/delete.txt', (err)=>{
5          if (err){
6              console.log(err)
7          }
8          console.log('file deleted')
9      })
10 }
```

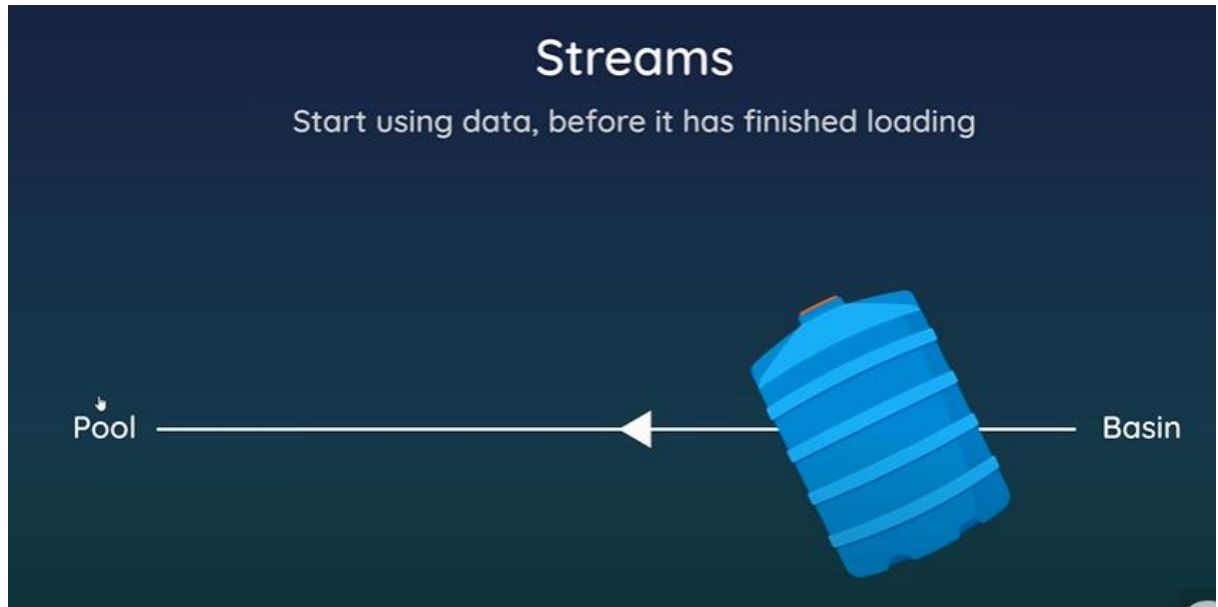
The TERMINAL panel at the bottom shows the command 'node file' being executed, resulting in the output 'file deleted'.

- If we run node file then delete.txt file will be deleted.

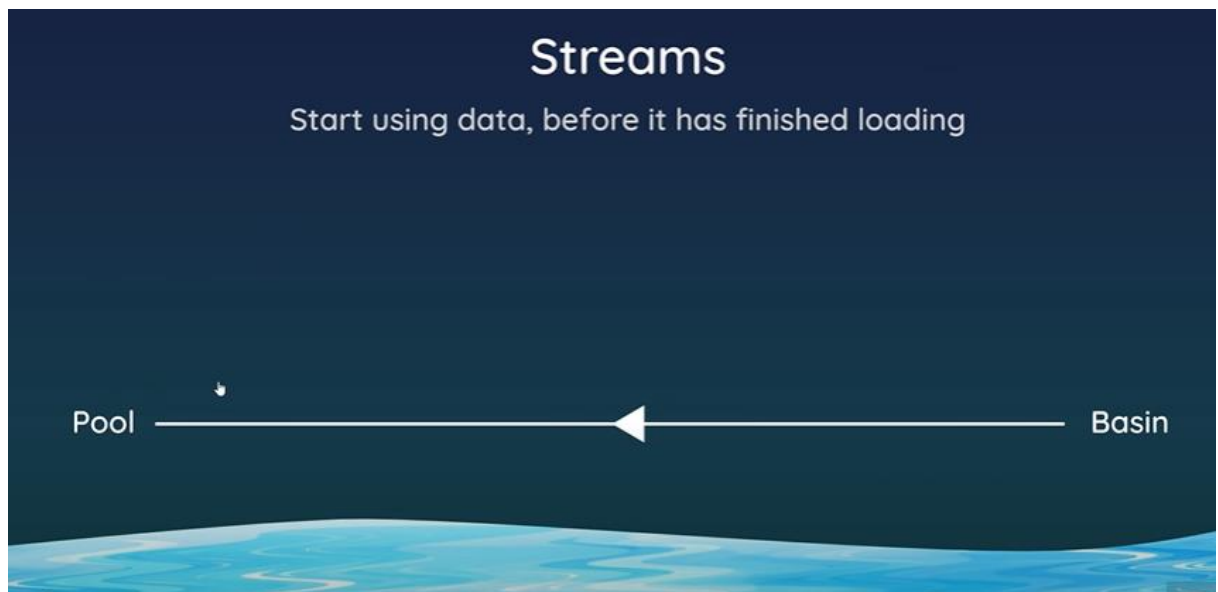
This screenshot is identical to the one above, showing the same VS Code interface with the 'file.js' file and the terminal output 'file deleted'.

Streams & Buffers

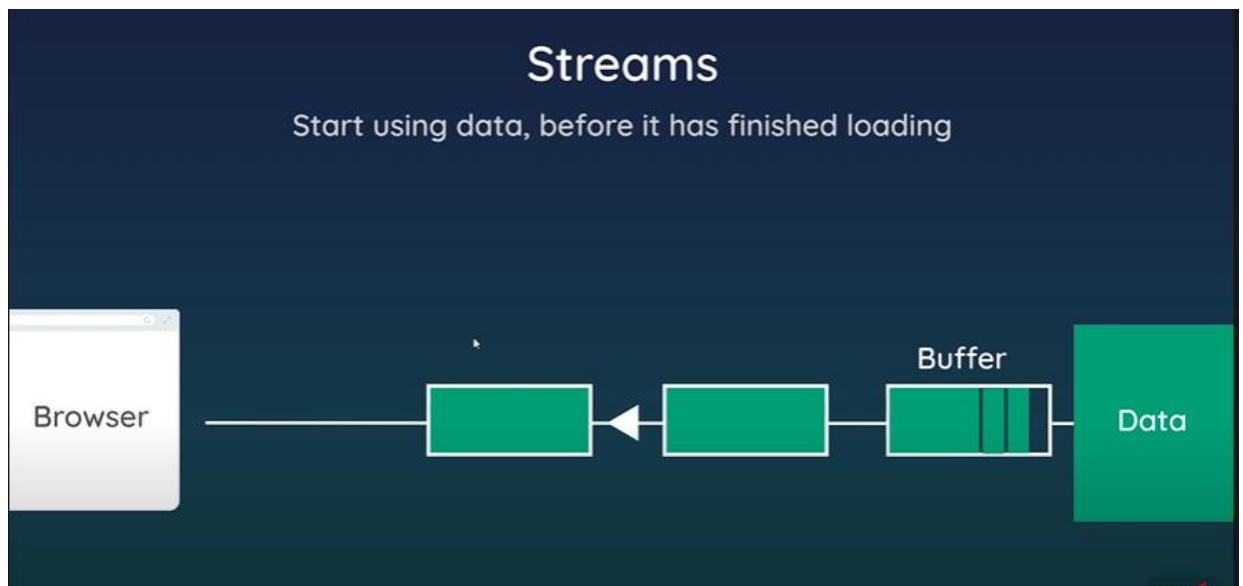
1st Option



2nd Option



- Same like data and browser



- For Read Stream

```
EXPLORER  ...  streams.js X
OPEN EDITORS
  streams.js node-...
NODE
  node-crash-course
    docs
      blog1.txt
      blog2.txt
      blog3.txt
    file.js
    global.js
    module.js
    people.js
    streams.js
    test.js

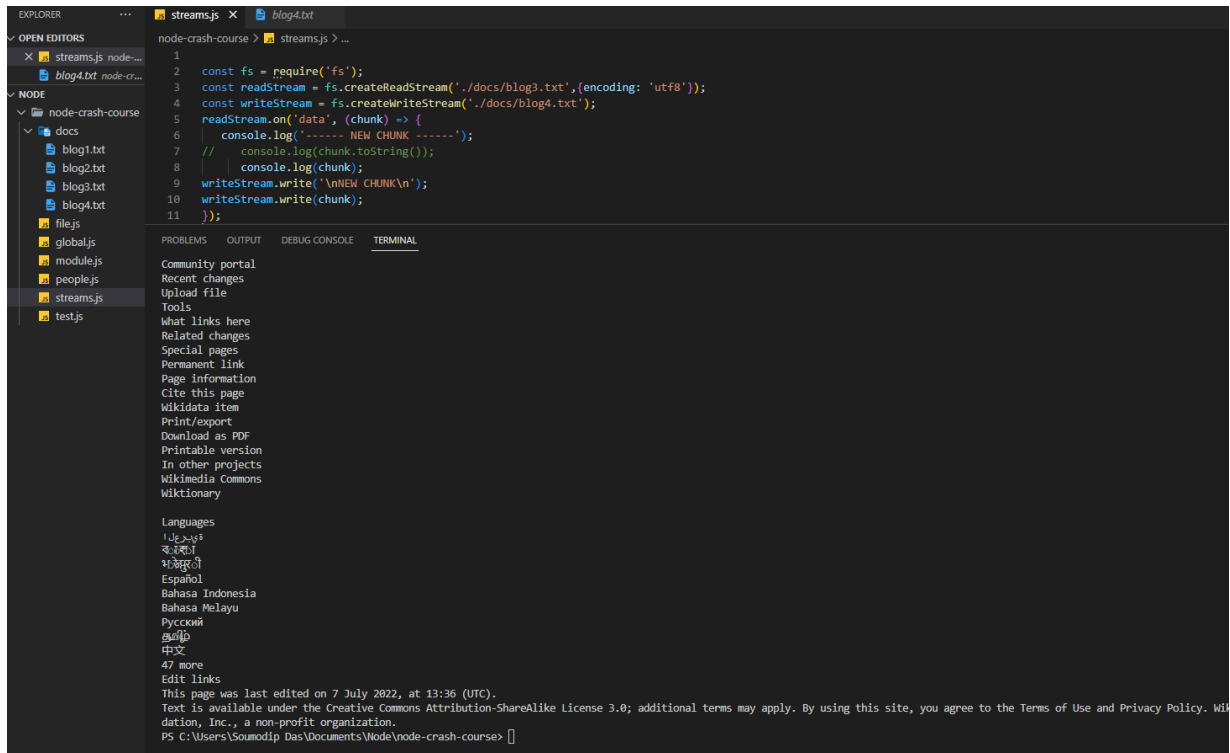
streams.js X
node-crash-course > streams.js ...
1
2 const fs = require('fs');
3 const readStream = fs.createReadStream('./docs/blog3.txt', {encoding: 'utf8'});
4 // const writeStream = fs.createWriteStream('./docs/blog4.txt');
5 readStream.on('data', (chunk) => {
6   console.log('----- NEW CHUNK -----');
7   // console.log(chunk.toString());
8   console.log(chunk);
9   // writeStream.write('\nNEW CHUNK\n');
10  // writeStream.write(chunk);
11 });
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Most of these variants are usually found in lowercase.

555: the Thai variation of LOL. "5" in Thai is pronounced "ha", three of them being "hahaha".
asg: Swedish abbreviation of the term asgarv, meaning intense laughter.
g: Danish abbreviation of the word griner, which means "laughing" in Danish.[49]
ha3: Malaysian variation of LOL. ha3 means pronouncing ha three times, "hahaha".
jajajá: in Spanish, the letter "j" is pronounced /x/. [50]
jejeje: in the Philippines is used to represent "hehehe". "j" in Filipino languages is pronounced as /h/, derived from the Spanish /x/. Its origins can be traced to SMS language. It is widely used in a Filipino youth subculture known as Jejemons.[51][52]
kkkk: in Somali and Ethiopian languages spoken in the Horn of Africa, iterations of the letter "k", usually ranging between 2 and 8 k's, are used as a variation of LOL. These iterations are also used by Shona, Ndebele and other Zimbabwean languages speakers, with the longer variant being "kikiki" (emulating a laughing sound).
mdr: Esperanto version, from the initials of "multe da ridoj", which translates to "lot of laughs" in English.
mdr: French version, from the initials of "mort de rire" which roughly translated means "died of laughter", although many French people also use LOL instead as it is the most widely used on the internet.[53][54]
mkm: in Afghanistan "mkm" (being an abbreviation of the phrase "ma khanda mikonam"). This is a Dari phrase that means "I am laughing".
ptdr: French variant from pété de rire - literally meaning "broken with laughter"
rs: in Brazil "rs" (being an abbreviation of "risos", the plural of "laugh") is often used in text based communications in situations where in English LOL would be used, repeating it ("rsrsrsrsrs") is often done to express longer laughter or laughing harder. Also popular is "kkk" (which can also be repeated indefinitely), due to the pronunciation of the letter k in Portuguese sounding similar to the ca in card, and therefore representing the laugh "cacacacaca" (also similar to the Hebrew version above).[55]
wkwkwk: in Indonesian, "wkwkwk" is often used to express laughter. A new variant, "Akwoakwoak" or "Awkwoakwoak" is often used beside the original version. Both the original and the new variant are derived from the sound of Donald Duck laughing, something like "Wakakaka".
חחחחח: Hebrew version of LOL. The letter ח is pronounced [/x/ /x/] and ה is pronounced [/h/ /h/]. Putting them together (usually three or more in a row) makes the word khakhakha or hahaha (since vowels in Hebrew are generally not written), which is in many languages regarded as the sound of laughter.
.....: Arabic script repetition of the "ا" character meaning "Hahahaha" or "Hehehehe". "ا" is equivalent to the letter "H". The equivalents of the "a" or "e" short vowels are Arabic diacritics and are optional to write.
ㅋㅋㅋ ("kkk" or "kekeke") [41] and ㅎㅎㅎ ("hhh") are usually used to indicate laughter in Korean. 'ㅋ', is a Korean Jamo consonant representing a "k" sound, and 'ㅎ' represents an "h" sound. Both "ㅋㅋㅋ" and "ㅎㅎㅎ" represent laughter which is not very loud. However, if a vowel symbol is written, louder laughter is

- For Write Stream



- We can see that a new txt file named blog4 has been created.

