JSON

→ JSON stands for JavaScript Object Notation.

→ JSON is a lightweight format for storing and transporting data.

→ JSON is often used when data is sent from a server to a web page

→ JSON is self-describing and easy to understand.

WRITING JSON

Javascript > Lecture 27 > JS FirstWritingJSON.js > ...

```
1   //npm init
2   // npm install minimist
3   // node FirstWritingJSON.js --dest=teams.json
4
5   let minimist = require("minimist");
6   let fs = require("fs");
7
8   let args = minimist(process.argv);
9   console.log(args.dest)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→  Lecture 27 git:(main) ✗ node FirstWritingJSON.js --dest=teams.json
teams.json
→  Lecture 27 git:(main) ✗
```

FirstWritingJSON.js U

Javascript > Lecture 27 > JS FirstWritingJSON.js > ...

```
11  // JSON = JavaScript Object Notation
12  // JSON means saving data in the same format as of javascript objects
13
14  let s1 = {
15      name: "Hemakshi",
16      age: 20
17  }; // an object
18
19  let s2 = {
20      name: "Morgan",
21      age: 22
22  } // another object
23
24  let stdntsWay1 = [s1, s2];   an array of object which is again an object = JSO (Javascript object)
25
26  let json = JSON.stringify(stdntsWay1); // JSO to JSON (Javascript object notation)
27  fs.writeFileSync(args.dest, json, "utf-8");
```

teams.json U

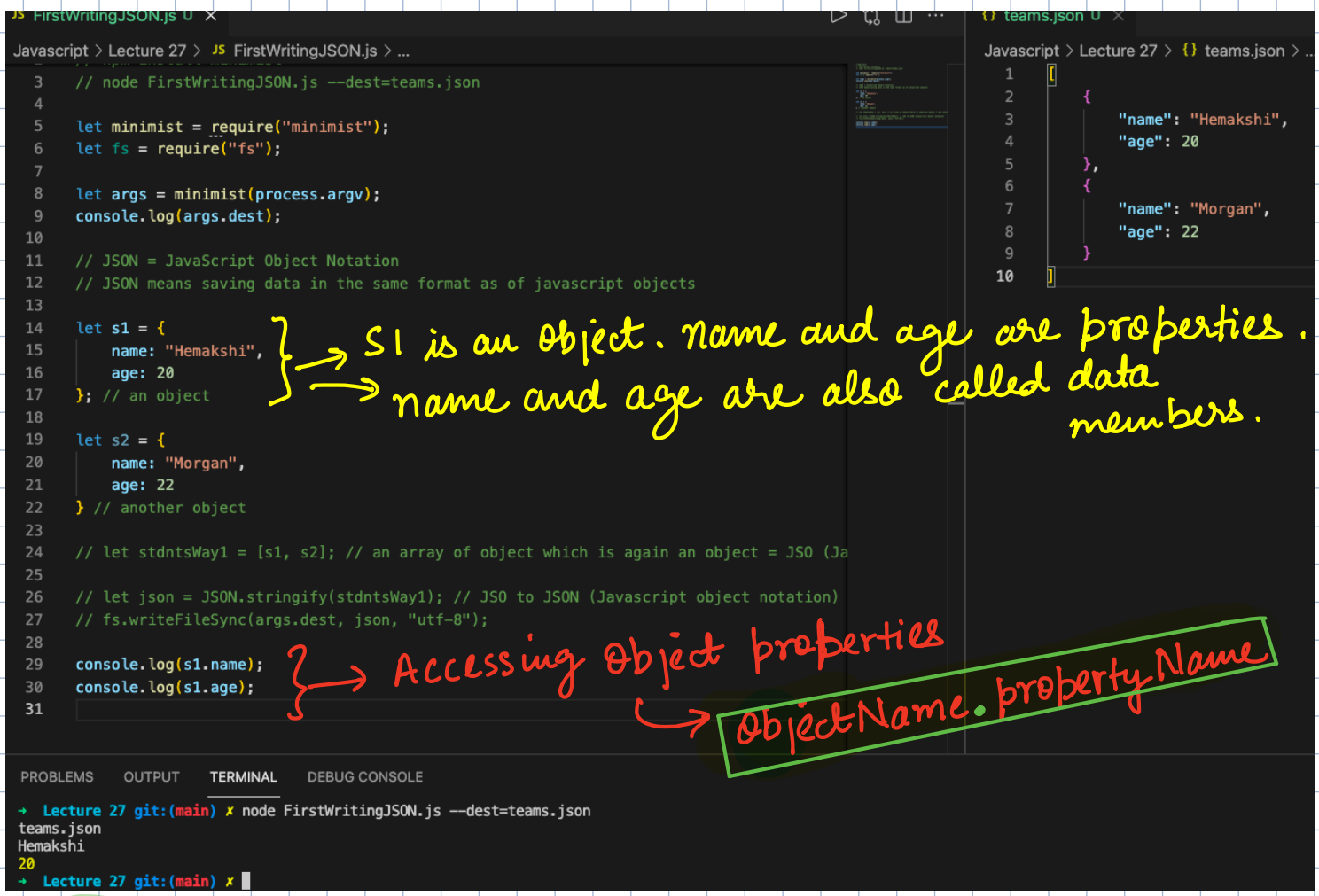
Javascript > Lecture 27 > {} teams.json > ...

```
1  [{"name":"Hemakshi","age":20},{"name":"Morgan","age":22}]
```

JSON.stringify() → a common use of JSON is to exchange data to/from a web server. When send data to a web server, the data has to be string. Convert a Javascript object into a string with JSON.stringify()

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE zsh

```
→  Lecture 27 git:(main) ✗ node FirstWritingJSON.js --dest=teams.json
teams.json
→  Lecture 27 git:(main) ✗
```



JavaScript objects

The name : values pairs in JavaScript object are called properties.

```
let person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
```

Accessing Object Properties

you can access object properties in two ways :-

```
let person = { firstName: "John",
               lastName : "Doe",
               age : 50,
               eyeColor: "blue"
             };
```

① objectName . propertyName ;

eg → person . lastName ;

② objectName ["propertyName"];

double quotes.

eg → person ["lastName"];

Array of objects

```
32
33    let ages_Arr = [10, 20, 30];
34    console.log(ages_Arr[0]);
35    console.log(ages_Arr[1]);
36    console.log(ages_Arr[2]);
37
38    let names_Arr = ["Hemakshi", "Morgan", "Audrey"];
39    console.log(names_Arr[0]);
40    console.log(names_Arr[1]);
41    console.log(names_Arr[2]);
42
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→  Lecture 27 git:(main) ✗ node FirstWritingJSON.js --dest=teams.json
teams.json
10
20
30
Hemakshi
Morgan
Audrey
→  Lecture 27 git:(main) ✗
```

Javascript > Lecture 27 > JS FirstWritingJSON.js > ...

```
let arrOfObjects = [
        {
            name: "Hemakshi",
            age: 20
        },
        {
            name: "Morgan",
            age: 22
        },
        {
            name: "Audrey",
            age: 21
        }
]

console.log(arrOfObjects[0].name);
console.log(arrOfObjects[0].age);

console.log(arrOfObjects[1].name);
console.log(arrOfObjects[1].age);

console.log(arrOfObjects[2].name);
console.log(arrOfObjects[2].age);
```

0

1

2

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→  Lecture 27 git:(main) ✗ node FirstWritingJSON.js --dest=teams.json
Hemakshi
20
Morgan
22
Audrey
21
→  Lecture 27 git:(main) ✗
```

FirstJSON.js U

Javascript > Lecture 27 > FirstJSON.js > ...

```
// npm install minimist
// node FirstJSON.js

let minimist = require("minimist");
let fs = require("fs");

let args = minimist(process.argv);

// JSON = JavaScript Object Notation
// JSON means saving data in the same format as of javascript objects

let s1 = {
    name: "Hemakshi",
    age: 20
}; // s1 is an object. name and age are properties
// name and age are also called data memeber

console.log(s1.name);
console.log(s1.age);

let ages_Arr = [10, 20, 30];
console.log(ages_Arr[0]);
console.log(ages_Arr[1]);
console.log(ages_Arr[2]);

let names_Arr = ["Hemakshi", "Morgan", "Audrey"];
console.log(names_Arr[0]);
console.log(names_Arr[1]);
console.log(names_Arr[2]);
```

FirstJSON.js U

Javascript > Lecture 27 > FirstJSON.js > ...

```
let arrOfObjects = [
        {
            name: "Hemakshi",
            age: 20
        },
        {
            name: "Morgan",
            age: 22
        },
        {
            name: "Audrey",
            age: 21
        }
]

console.log(arrOfObjects[0].name);
console.log(arrOfObjects[0].age);

console.log(arrOfObjects[1].name);
console.log(arrOfObjects[1].age);

console.log(arrOfObjects[2].name);
console.log(arrOfObjects[2].age);
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 27 git:(main) ✗ node FirstJSON.js
Hemakshi
20
10
20
30
Hemakshi
Morgan
Audrey
Hemakshi
20
Morgan
22
Audrey
21
```

FirstWritingJSON.js U

Javascript > Lecture 27 > FirstWritingJSON.js > teams > rank

```
// node FirstWritingJSON.js --dest=teams.json

let minimist = require("minimist");
let fs = require("fs");

let args = minimist(process.argv);

let teams = [
    {
        name: "India",
        rank: 1

    },
    {
        name: "Australia",
        rank: 3

    },
    {
        name: "England",
        rank: 2

    }
];

let json = JSON.stringify(teams); // JSO -> JSON
fs.writeFileSync(args.dest, json, "utf-8");
```

JSO

JSON.stringify

teams.json U

Javascript > Lecture 27 > {} teams.json > {}

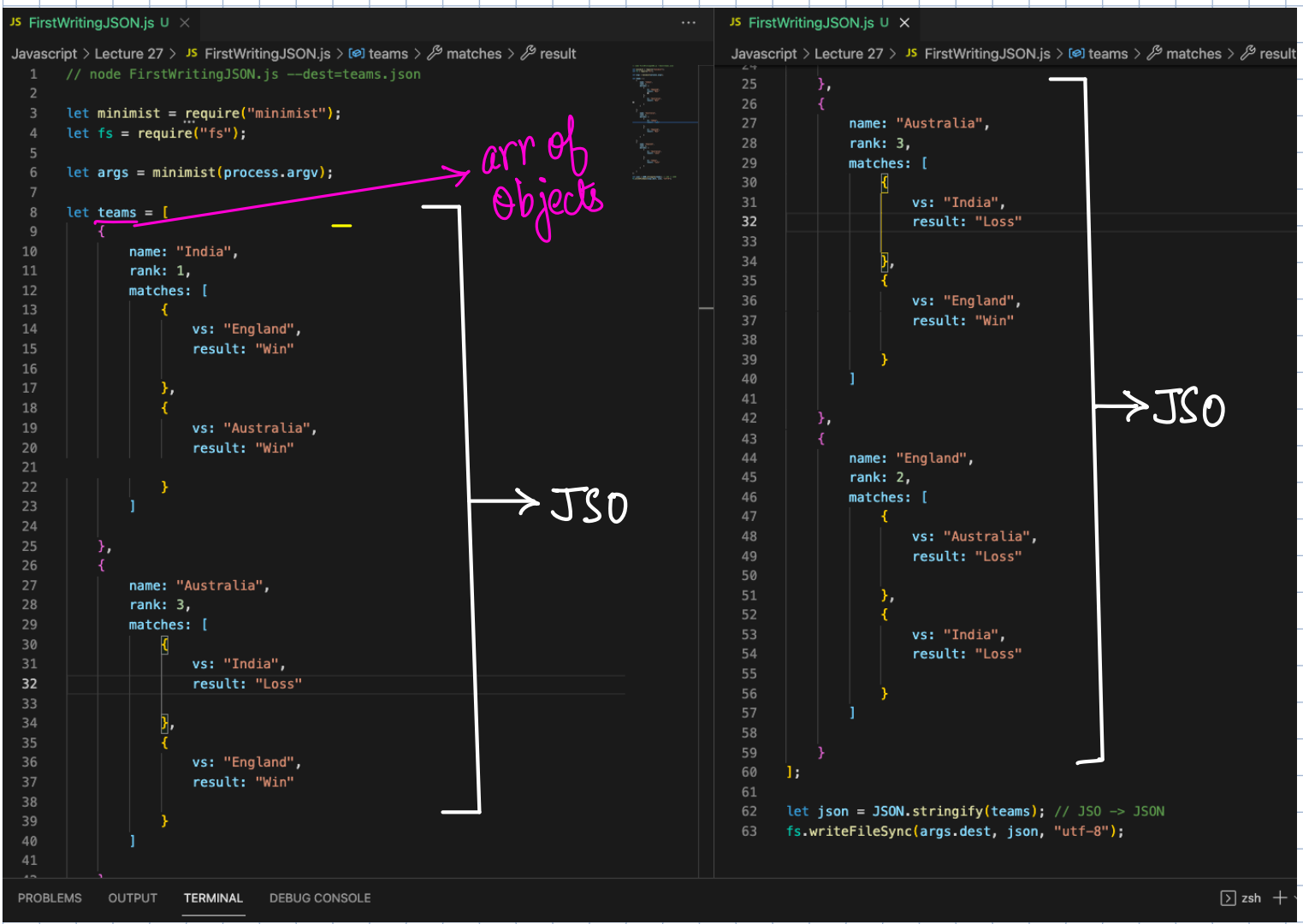
```
[
    {
        "name": "India",
        "rank": 1
    },
    {
        "name": "Australia",
        "rank": 3
    },
    {
        "name": "England",
        "rank": 2
    }
]
```

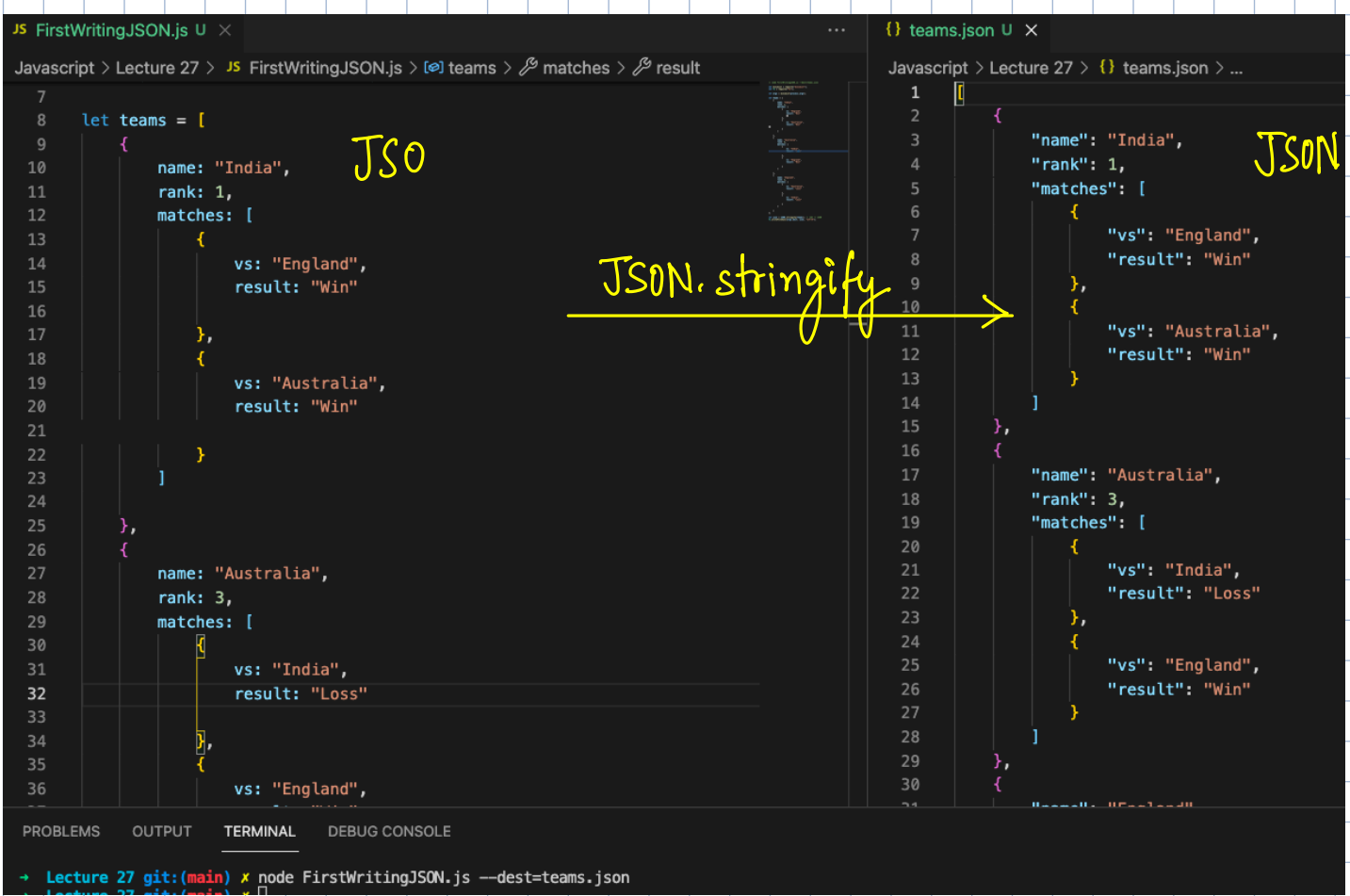
JSON

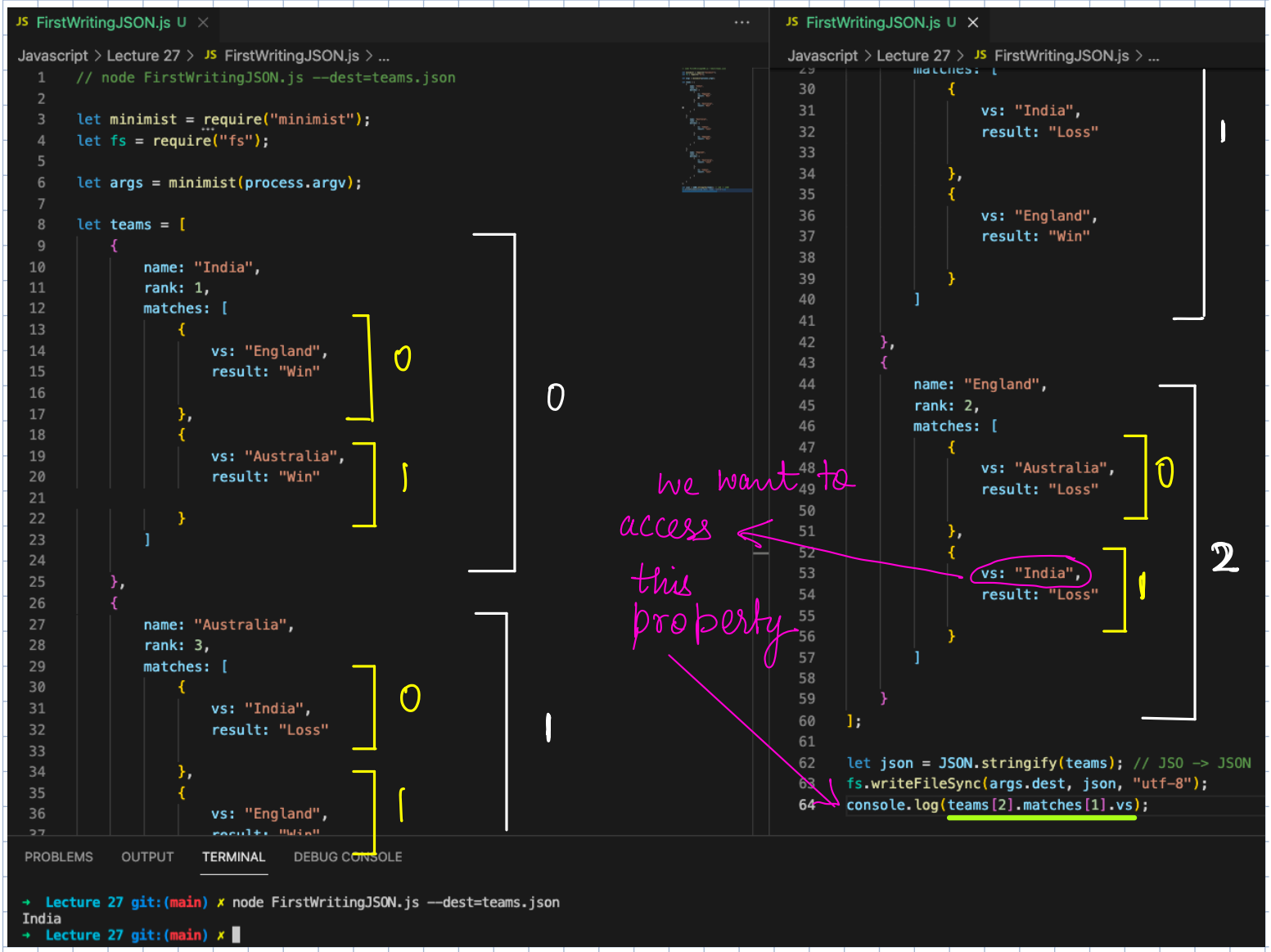
Writing JSON

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 27 git:(main) ✗ node FirstWritingJSON.js --dest=teams.json
→ Lecture 27 git:(main) ✗
```


FirstWritingJSON.js U
Javascript > Lecture 27 > FirstWritingJSON.js > teams > matches > result
// node FirstWritingJSON.js --dest=teams.json
let minimist = require("minimist");
let fs = require("fs");
let args = minimist(process.argv);
let teams = [
{
name: "India",
rank: 1,
matches: [
{
vs: "England",
result: "Win"
},
{
vs: "Australia",
result: "Win"
}
]
},
{
name: "Australia",
rank: 3,
matches: [
{
vs: "India",
result: "Loss"
},
{
vs: "England",
result: "Win"
}
]
},
{
name: "England",
rank: 2,
matches: [
{
vs: "Australia",
result: "Loss"
},
{
vs: "India",
result: "Loss"
}
]
}
];
let json = JSON.stringify(teams); // JSO -> JSON
fs.writeFileSync(args.dest, json, "utf-8");
arr of objects
JSO
JSO
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
zsh


FirstWritingJSON.js U
teams.json U
Javascript > Lecture 27 > FirstWritingJSON.js > teams > matches > result
Javascript > Lecture 27 > teams.json > ...
JSO
JSON.stringify
JSON
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE
→ Lecture 27 git:(main) ✗ node FirstWritingJSON.js --dest=teams.json



READ JSON

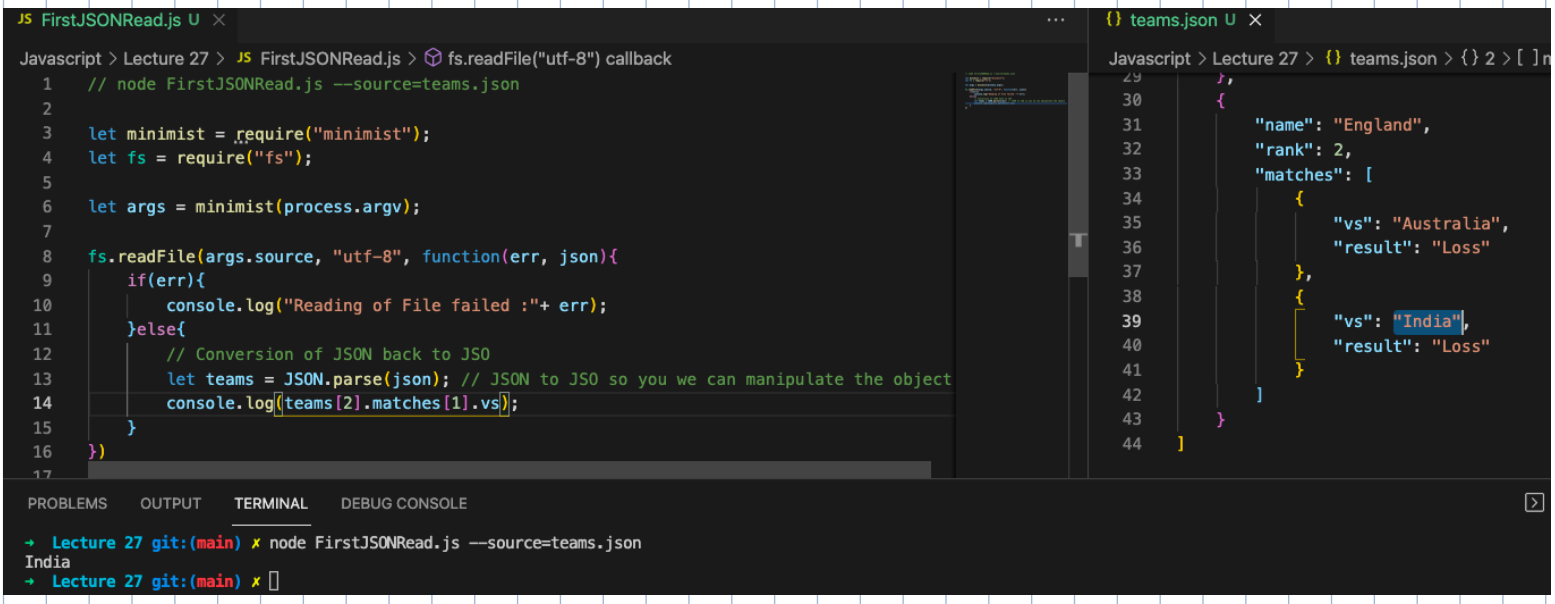
Javascript > Lecture 27 > FirstJSONRead.js > fs.readFile("utf-8") callback

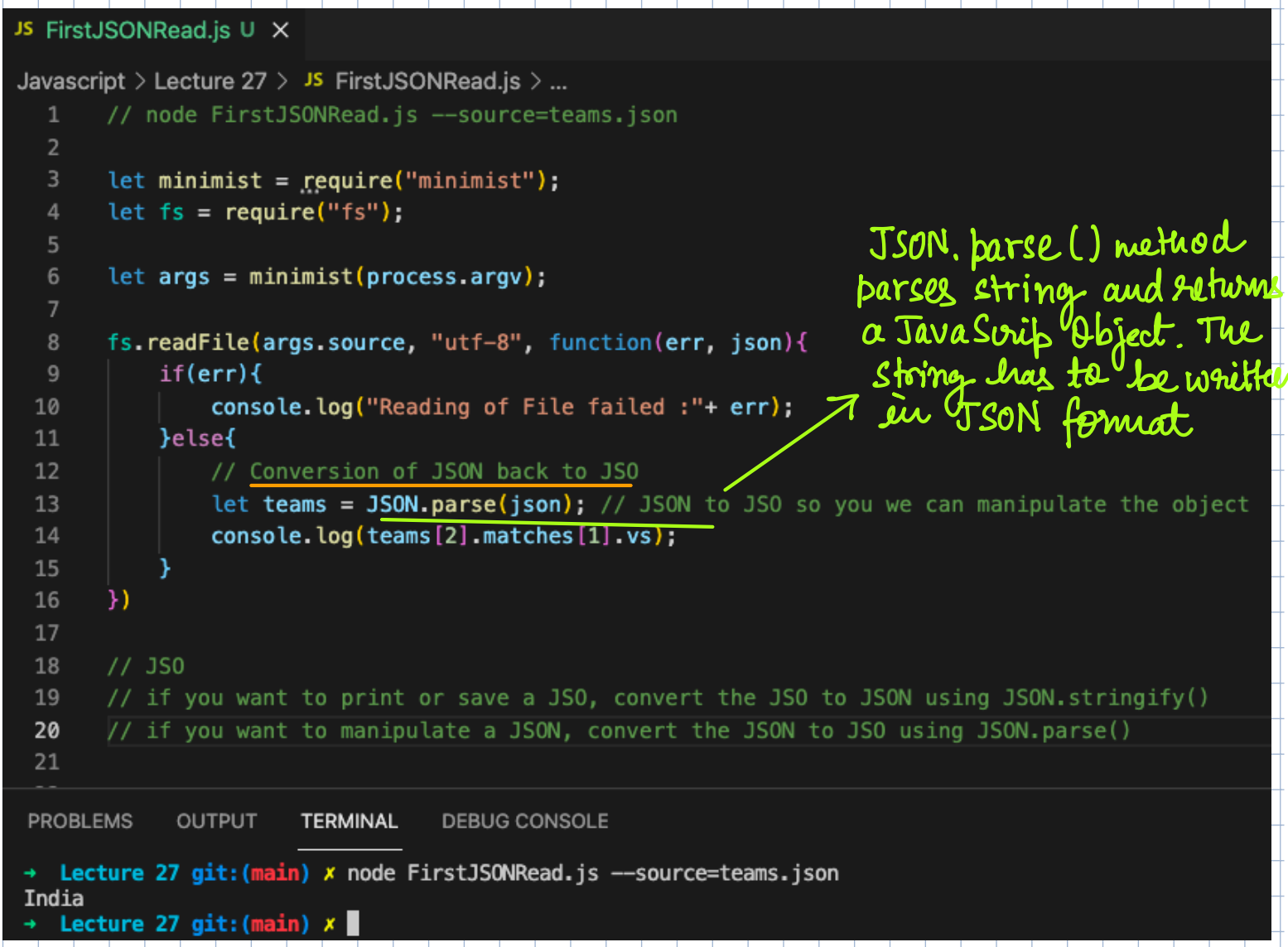
```
1   // node FirstJSONRead.js --source=teams.json
2
3   let minimist = require("minimist");
4   let fs = require("fs");
5
6   let args = minimist(process.argv);
7
8   fs.readFile(args.source, "utf-8", function(err, json){
9       if(err){
10          console.log("Reading of File failed :"+ err);
11      }else{
12          console.log(json);
13      }
14  })
15
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

err

```
Lecture 27 git:(main) ✗ node FirstJSONRead.js --source=team.json
Reading of File failed :Error: ENOENT: no such file or directory, open 'team.json'
Lecture 27 git:(main) ✗
```





JSON.parse() method parses string and returns a JavaScrip Object. The string has to be written in JSON format

→ If you want to print or save a JSO, convert the JSO to JSON using JSON.stringify().

→ If you want to manipulate a JSON, convert the JSON to JSO using JSON.parse().