FOLDER CREATION

JS FirstFolderCreation.js U

Javascript > Lecture 29 > JS FirstFolderCreation.js > ...

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

→ we will make three different folder for India, Australia and England

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 29 git:(main) ✗ node FirstFolderCreation.js --source=teams.json --dest=root
teams.json
root
→ Lecture 29 git:(main) ✗
```

Javascript > Lecture 29 > JS FirstFolderCreation.js > ...

```
1   // node FirstFolderCreation.js --source=teams.json --dest=root
2   //npm install minimist
3   //npm init
4
5   let minimist = require("minimist");
6   let fs = require("fs");
7   const { EBADMSG } = require("constants");
8
9   let args = minimist(process.argv);
10
11  let teamsJSON = fs.readFileSync(args.source, "utf-8");
12  let teams = JSON.parse(teamsJSON);
13
14  console.log(teams.length);
```

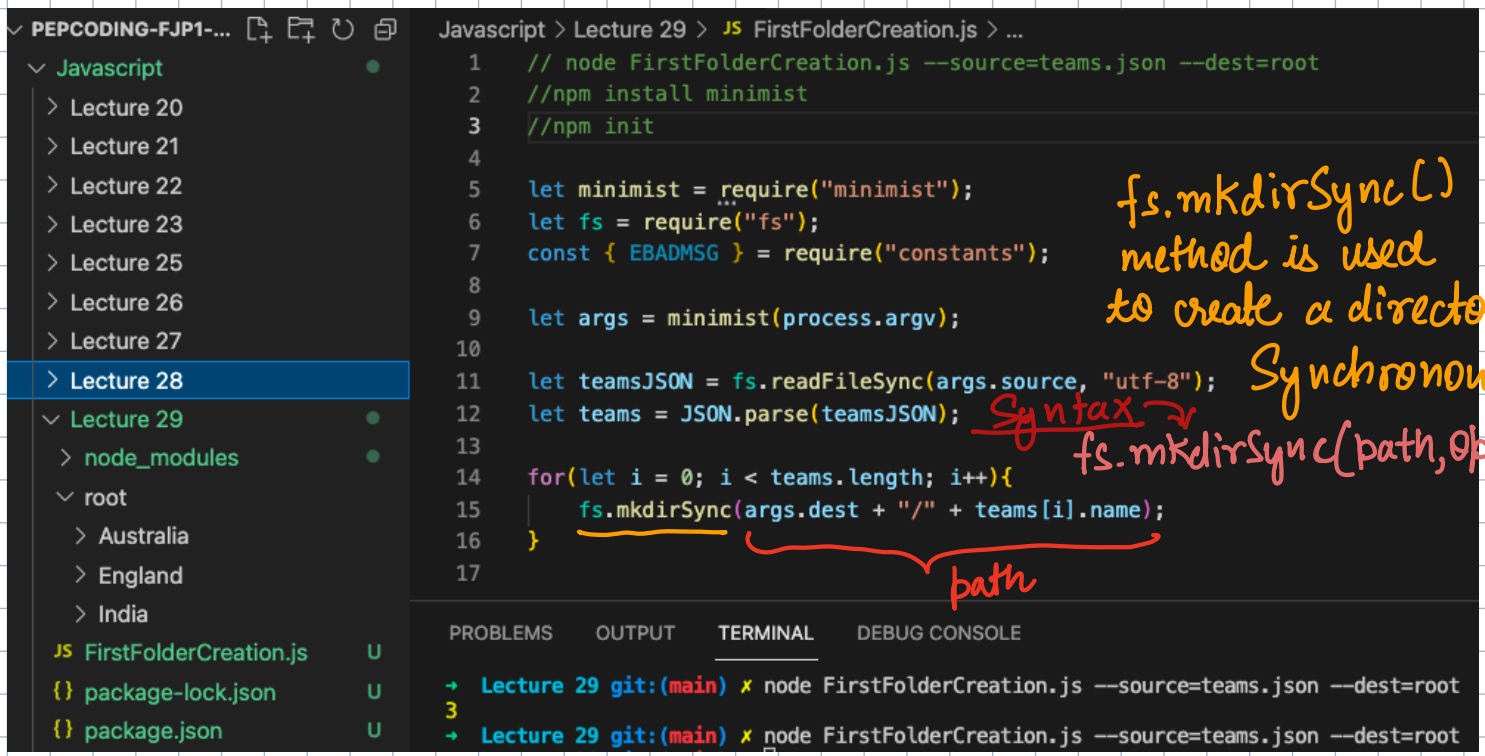
name of folder

reading teams.json file

conversion of JSON to JSO

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→ Lecture 29 git:(main) ✗ node FirstFolderCreation.js --source=teams.json --dest=root
3
→ Lecture 29 git:(main) ✗
```


PEPCODING-FJP1-...
Javascript
Lecture 20
Lecture 21
Lecture 22
Lecture 23
Lecture 25
Lecture 26
Lecture 27
Lecture 28
Lecture 29
node_modules
root
Australia
England
India
FirstFolderCreation.js U
package-lock.json U
package.json U
Javascript > Lecture 29 > FirstFolderCreation.js > ...
// node FirstFolderCreation.js --source=teams.json --dest=root
//npm install minimist
//npm init
let minimist = require("minimist");
let fs = require("fs");
const { EBADMSG } = require("constants");
let args = minimist(process.argv);
let teamsJSON = fs.readFileSync(args.source, "utf-8");
let teams = JSON.parse(teamsJSON);
for(let i = 0; i < teams.length; i++){
fs.mkdirSync(args.dest + "/" + teams[i].name);
}
fs.mkdirSync() method is used to create a directory Synchronously.
Syntax
fs.mkdirSync(path,Options)
path
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Lecture 29 git:(main) ✗ node FirstFolderCreation.js --source=teams.json --dest=root
3
Lecture 29 git:(main) ✗ node FirstFolderCreation.js --source=teams.json --dest=root


Javascript > Lecture 29 > FirstFolderCreation.js > ...
// node FirstFolderCreation.js --source=teams.json --dest=root
//npm install minimist
//npm init
let minimist = require("minimist");
let fs = require("fs");
let path = require("path"); // better method for path
let args = minimist(process.argv);
let teamsJSON = fs.readFileSync(args.source, "utf-8");
let teams = JSON.parse(teamsJSON);
for(let i = 0; i < teams.length; i++){
let folderName = path.join(args.dest, teams[i].name);
console.log(folderName);
}
The path.join.method is used to join a number of path segments using the platform specific delimeter to form a single path. The final path is normalized after joining takes place. The path-segments are specified using comma-separated values.
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Lecture 29 git:(main) ✗ node FirstFolderCreation.js --source=teams.json --dest=root
root/India
root/Australia
root/England
Lecture 29 git:(main) ✗

```
// node FirstFolderCreation.js --source=teams.json --dest=root
//npm install minimist
//npm init

let minimist = require("minimist");
let fs = require("fs");
let path = require("path");  // we will not put the slash by ourselves
// to make folder paths, never append slashes yourself
// use path.join()

let args = minimist(process.argv);

let teamsJSON = fs.readFileSync(args.source, "utf-8");
let teams = JSON.parse(teamsJSON); // conversion of JSON to JSO

for(let i = 0; i < teams.length; i++){
    let folderName = path.join(args.dest, teams[i].name);
    fs.mkdirSync(folderName);
}
```

WRITING PDF

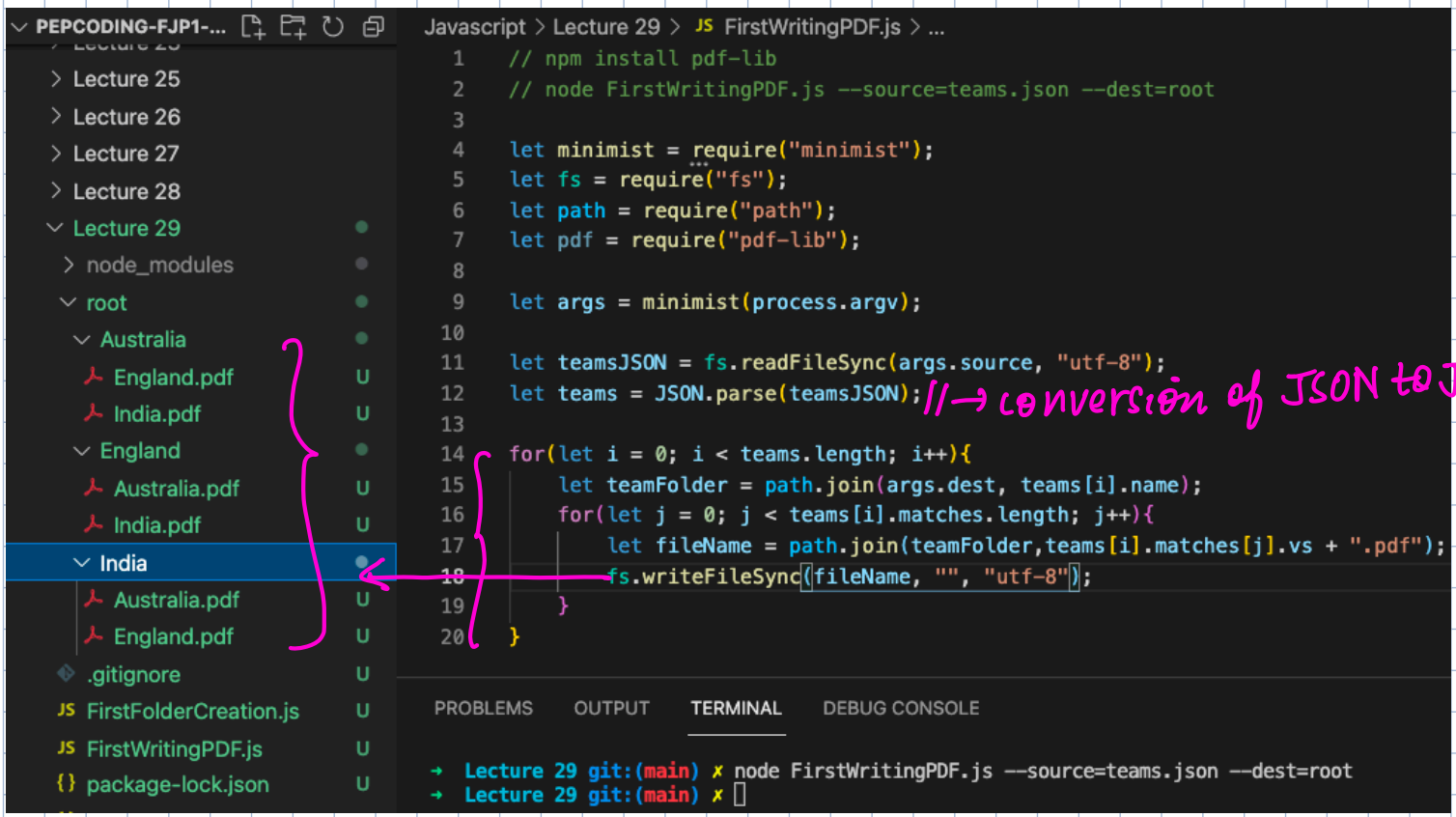
Javascript > Lecture 29 > JS FirstWritingPDF.js > ...

```
// npm install pdf-lib
// node FirstWritingPDF.js --source=teams.json --dest=root

let minimist = require("minimist");
let fs = require("fs");
let path = require("path");
let pdf = require("pdf-lib");

let args = minimist(process.argv);
console.log(args.source);
console.log(args.dest)
```

→ for every match we have to make a pdf scorecard.

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
→  Lecture 29 git:(main) ✗ node FirstWritingPDF.js --source=teams.json --dest=root
teams.json
root
→  Lecture 29 git:(main) ✗
```



Javascript > Lecture 29 > FirstWritingPDF.js > createScoreCard

```
1   // npm install pdf-lib
2   // node FirstWritingPDF.js --source=teams.json --dest=worldCup
3
4   let minimist = require("minimist");
5   let fs = require("fs");
6   let path = require("path");
7   let pdf = require("pdf-lib");
8
9   let args = minimist(process.argv);
10
11  let teamsJSON = fs.readFileSync(args.source, "utf-8");
12  let teams = JSON.parse(teamsJSON);
13
14  fs.mkdirSync(args.dest);
15
16  for(let i = 0; i < teams.length; i++){
17      let teamFolder = path.join(args.dest, teams[i].name);
18      fs.mkdirSync(teamFolder); // team name folder creation
19      for(let j = 0; j < teams[i].matches.length; j++){
20          let matchFileName = path.join(teamFolder,teams[i].matches[j].vs + ".pdf");
21          createScoreCard(teams[i].name, teams[i].matches[j], matchFileName);
22      }
23  }
24
25  function createScoreCard(teamName, match, matchFileName){
26    // this function creates pdf for match in appropriate folder with correct details
27    //here we will use pdf-lib to create the pdf
28
29  }
```

matches pdf in team folder

Template

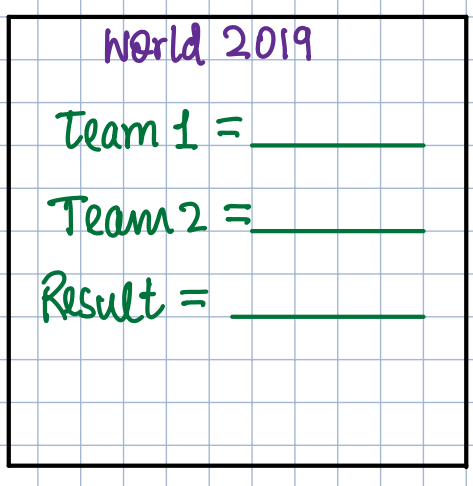
Worldcup 2019

Team 1	
Team 2	
Result	

```
// npm install pdf-lib
// node FirstWritingPDF.js --source=teams.json --dest=worldCup

let minimist = require("minimist");
let fs = require("fs");
let path = require("path");
let pdf = require("pdf-lib");

let args = minimist(process.argv);

let teamsJSON = fs.readFileSync(args.source, "utf-8");
let teams = JSON.parse(teamsJSON); //conversion to JSON to JSO -> array of teams object

fs.mkdirSync(args.dest); // creation of worldCup folder

for(let i = 0; i < teams.length; i++){
    let teamFolder = path.join(args.dest, teams[i].name);
    fs.mkdirSync(teamFolder); // team name folder creation of each team (India, Australia, England)
                              //in worldCup folder -> eg worldCup\India
    for(let j = 0; j < teams[i].matches.length; j++){
        let matchFileName = path.join(teamFolder,teams[i].matches[j].vs + ".pdf");
        createScoreCard(teams[i].name, teams[i].matches[j], matchFileName);
    }
}

function createScoreCard(teamName, match, matchFileName){
  // this function creates pdf for match in appropriate folder with correct details
  //here we will use pdf-lib to create the pdf
  console.log(teamName);
  console.log(match.vs);
```

```
    }
}

function createScoreCard(teamName, match, matchFileName){
  // this function creates pdf for match in appropriate folder with co
  //here we will use pdf-lib to create the pdf
  console.log(teamName);
  console.log(match.vs);
  console.log(match.result);
  console.log(matchFileName);
  console.log("-------------------------");

}
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

→ Lecture 29 git:(main) ✗ node FirstWritingPDF.js --source=teams.json --dest=root
India
England
Win
root/India/England.pdf
-------------------------
India
Australia
Win
root/India/Australia.pdf
-------------------------
Australia
India
```



Template

pdf

→ we will load the template and fill it's content according to matches and generates separates pdf's inside each folder.