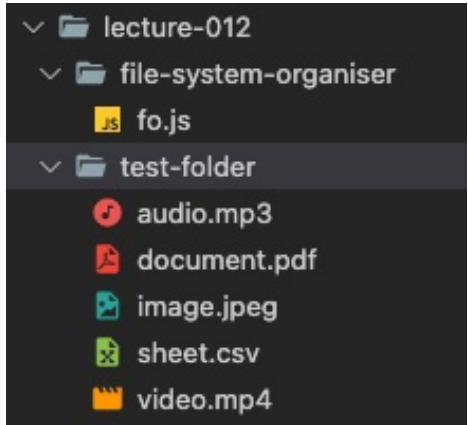


Lecture-012

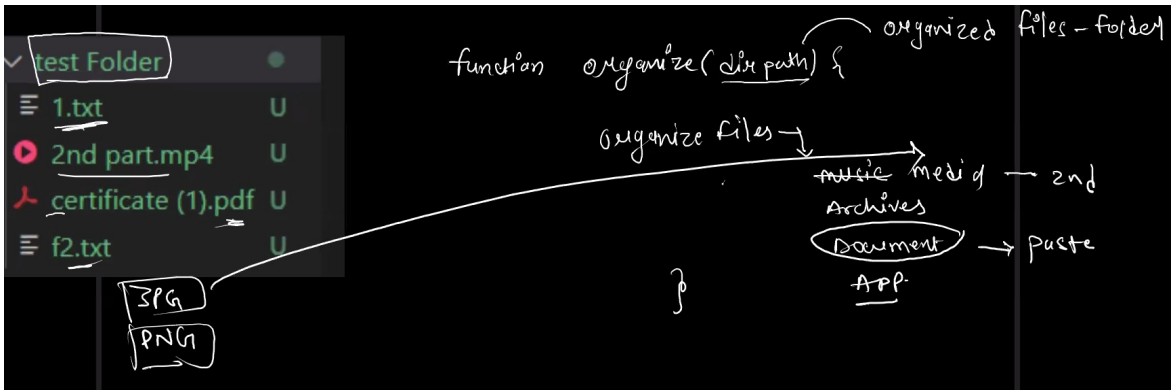
project - file system organiser

preq : fs module, path module.
modularity, modularity, methords etc...

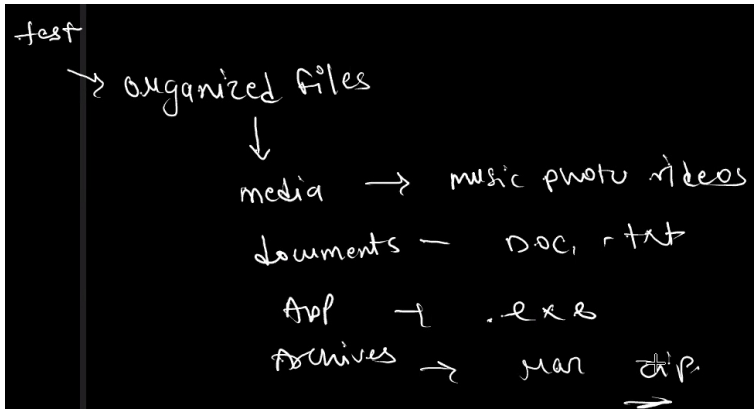
test-folder setup : fill folder with files to be organised



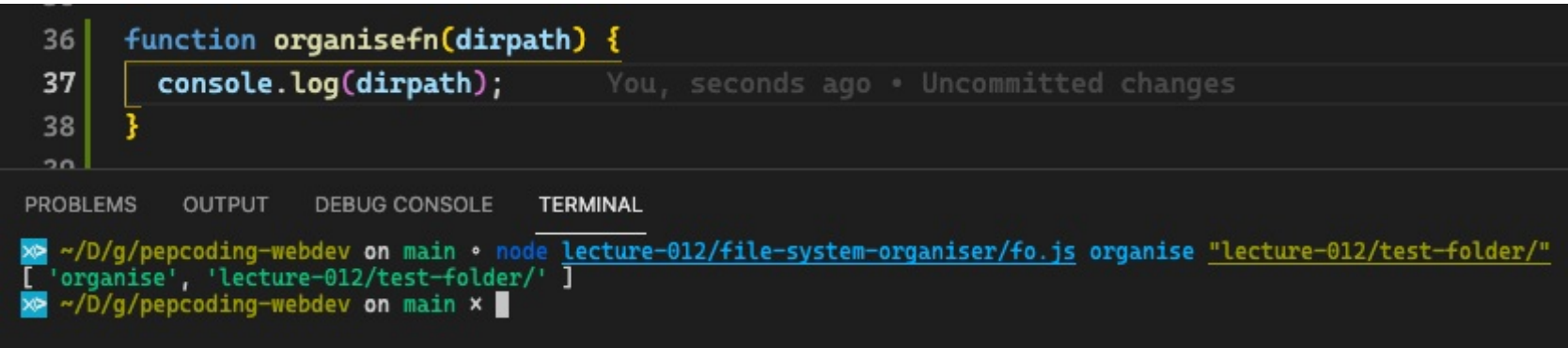
organise fn | setup ./test/organised/.../...



structure



initiating the organise fn : which is called in the switch case



checking if file exists

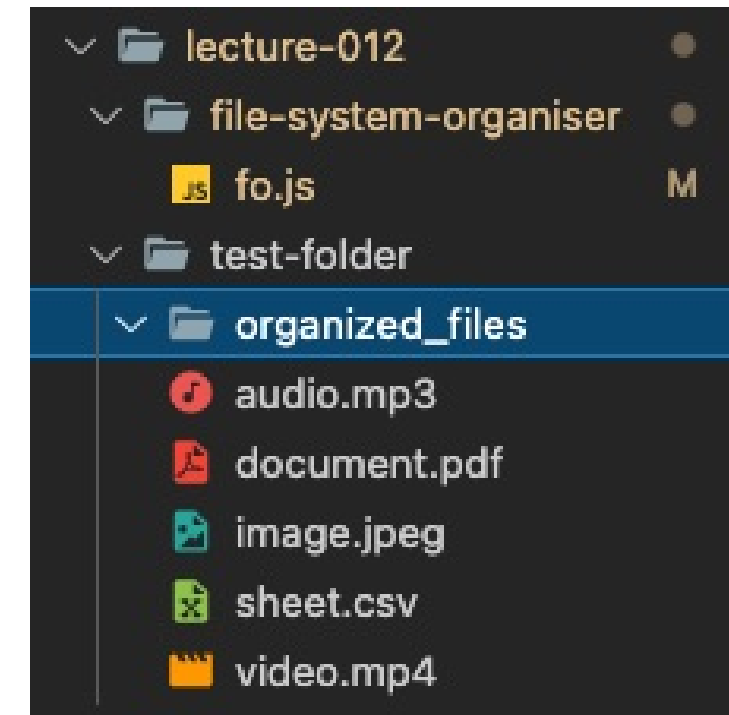


Disclaimer : might take time to improve notes b/c having troubles understanding

we are writing this function to organise the files

```
function organizeFn(dirpath) {  
  // input of a directory Path  
  let destPath;  
  
  if (dirpath == undefined) {  
    console.log("Please Enter a valid Directory Path");  
    //check whether dirpath is passed or not  
    return;  
  } else {  
    let doesExist = fs.existsSync(dirpath);  
    // this will tell whether the dirpath exists or not  
    if (doesExist == true) {  
      destPath = path.join(dirpath, "organized_files");  
      if (fs.existsSync(destPath) == false) {  
        fs.mkdirSync(destPath); // we will only create a folder if it does not already exists  
      } else {  
        console.log("The folder : " + destPath + " ... already exists");  
      }  
    } else {  
      console.log("Please Enter a valid Path");  
    }  
  }  
}
```

folder structure after run



to organise files to sub folders

```
let types = {
  media: ["mp4", "mp3", "jpg", "jpeg", "png", "gif"],
  archives: ["zip", "rar", "7z"],
  documents: ["doc", "docx", "pdf", "csv", "txt"],
  executables: ["exe", "msi", "apk", "dmg"],
};
```

code to log extensions of files inside the dirpath

```
74 // we are writing this function to categorize our files
75 function organizeHelper(src, dest) {
76   let childNames = fs.readdirSync(src); // get all the files and folders inside your src
77   //console.log(childNames)
78   for (let i = 0; i < childNames.length; i++) {
79     let childAddress = path.join(src, childNames[i]); // path is identified for the files
80     let isFile = fs.lstatSync(childAddress).isFile(); // we check here to identify only the files
81     // console.log(childAddress + " " + isFile);
82
83     if (isFile == true) {
84       // if the file is a file
85       let fileCategory = getCategory(childNames[i]);
86       // we are calling the categorize function to get the category of the file (extension of the file)
87     }
88   }
89 }
90
91 function getCategory(name) {
92   let ext = path.extname(name); // we are getting the extension of the file
93   ext = ext.slice(1); // we are removing the dot from the extension
94   console.log(ext);
95 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
~/D/g/pepcoding-webdev on main x node lecture-012/file-system-organiser/fo.js organise ../lecture-012/test-folder/
The folder : lecture-012/test-folder/organized_files ...already exists
mp3
pdf
jpeg
csv
mp4
```


driver code

```
// we are writing this function to categorize our files
function organizeHelper(src, dest) {
  let childNames = fs.readdirSync(src); // get all the files and folders inside your src
  //console.log(childNames)

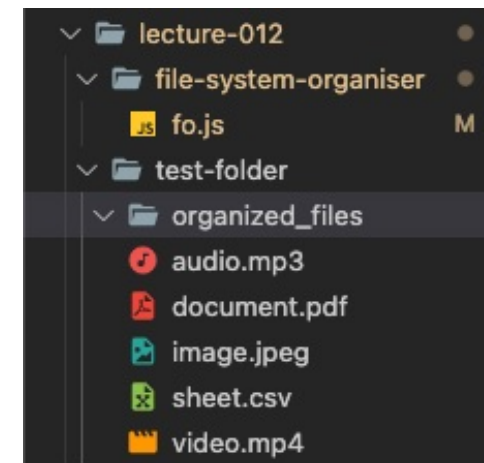
  for (let i = 0; i < childNames.length; i++) {
    let childAddress = path.join(src, childNames[i]); // path is identified for the files
    let isFile = fs.lstatSync(childAddress).isFile(); // we check here to identify only the files
    //console.log(childAddress + " " + isFile)

    if (isFile == true) {
      let fileCategory = getCategory(childNames[i]);
      console.log(childNames[i] + " belongs to " + fileCategory);
    }
  }
}
```

spitting out category types according to types obj

```
91 function getCategory(name) {
92   let ext = path.extname(name);
93   ext = ext.slice(1); // we will take out the extension names of the files
94   //console.log(ext)
95
96   for (let type in types) {
97     let cTypeArr = types[type];
98     //console.log(cTypeArr)
99
100    for (let i = 0; i < cTypeArr.length; i++) {
101      if (ext == cTypeArr[i])
102        // we matched the extensions with the values presnet in ctypeArr
103
104      return type;
105    }
106  }
107  return "others";
108 }
109
```

for ref.



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
~ /D/g/pepcoding-webdev on main x node lecture-012/file-system-organiser/fo.js organise "./lecture-012/test-folder/"
The folder : lecture-012/test-folder/organized_files ...already exists
audio.mp3 belongs to media
document.pdf belongs to documents
image.jpeg belongs to media
sheet.csv belongs to documents
video.mp4 belongs to media
```

send files

lecture-003

lecture-004

lecture-005

lecture-006

lecture-007

lecture-008

lecture-009

lecture-010

lecture-011

lecture-012

file-system-organiser

fo.js

test-folder

organized_files

documents

media

audio.mp3

document.pdf

image.jpeg

sheet.csv

video.mp4

temp

_config.yml

.gitignore

README.md

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

if (isFile == true) {
 let fileCategory = getCategory(childNames[i]);
 console.log(childNames[i] + " belongs to " + fileCategory);
 sendFiles(childAddress, dest, fileCategory);
}

function getCategory(name) {
 let ext = path.extname(name);
 ext = ext.slice(1); // we will take out the extension names of the files
 //console.log(ext)
 for (let type in types) {
 let cTypeArr = types[type];
 //console.log(cTypeArr)
 for (let i = 0; i < cTypeArr.length; i++) {
 if (ext == cTypeArr[i])
 // we matched the extensions with the values presnet in ctypeArr
 return type;
 }
 }
 return "others";
}

function sendFiles(srcFilePath, destFilePath, fileCategory) {
 let categoryPath = path.join(destFilePath, fileCategory);
 if (fs.existsSync(categoryPath) == false) {
 fs.mkdirSync(categoryPath);
 }
}

PROBLEMS

OUTPUT

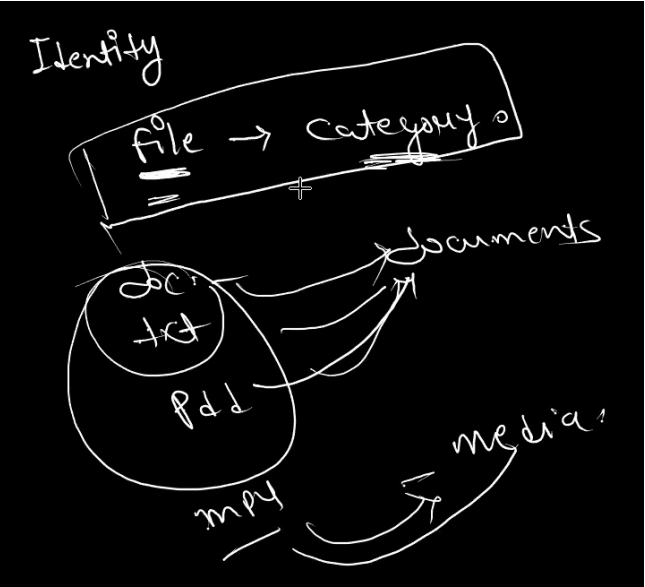
DEBUG CONSOLE

TERMINAL

>> ~/D/g/pepcoding-webdev on main x node lecture-012/file-system-organiser/fo.js organise ./lecture-012/test-folder/
The folder : lecture-012/test-folder/organized_files ...already exists
audio.mp3 belongs to media
document.pdf belongs to documents
image.jpeg belongs to media
sheet.csv belongs to documents
video.mp4 belongs to media

> OUTLINE

send files fn



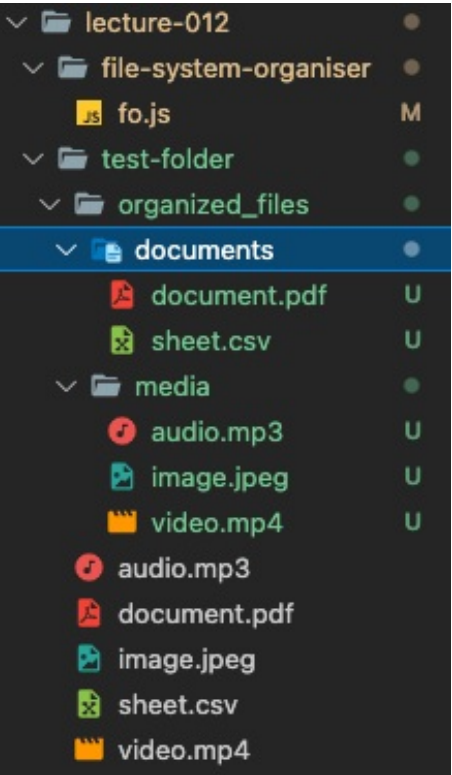
copy files from src to organised files

```
108 function sendFiles(srcFilePath, dest, fileCategory) {
109   let categoryPath = path.join(dest, fileCategory);
110   if (fs.existsSync(categoryPath) == false) {
111     fs.mkdirSync(categoryPath);
112   }
113
114   let fileName = path.basename(srcFilePath); // we will get the file name
115   let destFilePath = path.join(categoryPath, fileName); // we will get the destination path
116   fs.copyFileSync(srcFilePath, destFilePath); // we will copy the file to the destination path
117 }
118
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
~/D/g/pepcoding-webdev on main x node lecture-012/file-system-organiser/fo.js organise "../lecture-012/test-folder/"
The folder : lecture-012/test-folder/organized_files ...already exists
audio.mp3 belongs to media
document.pdf belongs to documents
image.jpeg belongs to media
sheet.csv belongs to documents
video.mp4 belongs to media
~/D/g/pepcoding-webdev on main x
```

copied!



unlink deletes file copies from src path

```
108 function sendFiles(srcFilePath, dest, fileCategory) {
109   let categoryPath = path.join(dest, fileCategory);
110   if (fs.existsSync(categoryPath) == false) {
111     fs.mkdirSync(categoryPath);
112   }
113
114   let fileName = path.basename(srcFilePath); // we will get the file name
115   let destFilePath = path.join(categoryPath, fileName); // we will get the destination path
116   fs.copyFileSync(srcFilePath, destFilePath); // we will copy the file to the destination path
117   fs.unlinkSync(srcFilePath); // we will delete the file from the source path
118   console.log(fileName + " copied to " + categoryPath);
119 }
120
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
audio.mp3 copied to lecture-012/test-folder/organized_files/media
document.pdf belongs to documents
document.pdf copied to lecture-012/test-folder/organized_files/documents
image.jpeg belongs to media
image.jpeg copied to lecture-012/test-folder/organized_files/media
sheet.csv belongs to documents
sheet.csv copied to lecture-012/test-folder/organized_files/documents
video.mp4 belongs to media
video.mp4 copied to lecture-012/test-folder/organized_files/media
~/D/g/pepcoding-webdev on main x
```

deleted files from src

