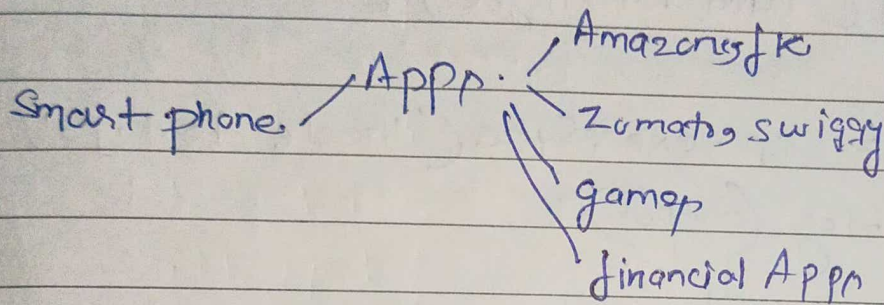
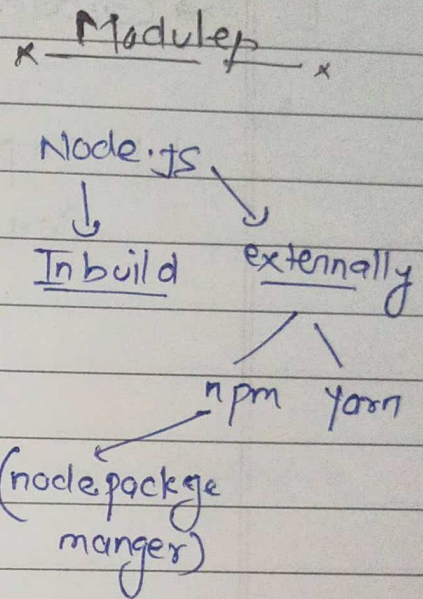


25-Nov-2021

Node Modules (folder)



App - 1 Universal App.



Inbuilt Modules

1) child process :-

file name  
cp.js (keyword)  
require ('child-process') (Module name)

cmd

calc → (open Calculator)

Code (open VS Code)

(Similar child process work)

```
const cp = require('child-process')
console.log('Trying to open Calculator');
cp.execSync('calc')
```

```
console.log('Calculator opened');
```

Terminal → node cp.js



```
console.log('Trying to open vs code')
```

```
cp.execSync('code')
```

```
console.log('vs code opened')
```

```
cp.execSync('start chrome')
```

(create  
for test)

```
cp.execSync('start chrome https://www. --- .com')
```

→ test.js

```
console.log('I am a test file')
```

```
let output = cp.execSync('node test.js')
```

```
console.log(output)
```

data is coming in Buffer form

↓ (means)

Binary form

```
console.log('Output is ' + output);
```

(Just Concat)

Child process → to create Sub process.



## OS module

Page No. \_\_\_\_\_

os.js (file)

Operating System

far communication with operating system

variable module (package)  
`const os = require('os')`

(archite.)  
`console.log(os.arch())`

① `console.log(os.arch())`

② `console.log(os.platform())`

③ `console.log(os.networkInterfaces())`

④ `console.log(os.cpus())`

Imp File System  
fs.js (file)

// Node Modules are used to perform different tasks as required

// File System Modules :-

// 1:- read, write, update, delete

\*

→ `fs.txt`

`const fs = require('fs')` (readFileSync is a method to read a file)

`let data = fs.readFileSync('fs.txt')`

`console.log('This is the content' + data)`



## \* Writing to a file

(अगर पहले से ही file 

Page No.
Date

 change हो गया हो तो वही data change हो जाएगा)  
(create हो जाएगा)

```
fs.writeFileSync('f2.txt', 'Hello from f2')
```

`writeFileSync(' ', ' ')` expect Two things  
(method is used to put data (write) to a file.  
If the file passed doesn't exist it creates a new file and writes to it.

## \* Update a file (append)

```
fs.appendFileSync('f2.txt', 'this is f2 data')
```

`appendFileSync` adds data to the previously written file.

Basically it appends a file with the new data passed.

## \* Delete a file by passing its path

```
# unlinkSync()
```

```
fs.unlinkSync('f2.txt')
```

```
console.log('file removed')
```



# Directories

Page No. \_\_\_\_\_

Date \_\_\_\_\_

① Create // `mkdirSync()` is used to create a new directory.  
`fs.mkdirSync('myDirectory')`

② Delete // `rmdirSync()` is used to delete a Directory.  
`fs.rmdirSync('myDirectory')`

③ path  
to check whether a directory exist or not we can use `existsSync` method

```
let doesExist = fs.existsSync('myDirectory')
console.log(doesExist) // return
                        ① True
                        ② false
```

If the directory exist the method return True otherwise false

my Directory  
├── js.txt  
└── js.txt

④ stats of a path (what are the details of a folder contents inside of a folder) (`lstatSync` provides up with all the statistics of a directory)

```
let statsofPath = fs.lstatSync('myDirectory')
console.log(statsofPath)
```

⑤ `isFile()` is a method whether a passed path is a file or not.

```
console.log('isFile?', statsofPath.isFile()) // false
```

⑥ `isDirectory()` is a method that checks whether a passed path is a directory or not.

```
console.log('isDirectory?', statsofPath.isDirectory()) // True
```



Read Directory (`readDirSync()`) is used to check the content of a particular folder/directory.

Page No. \_\_\_\_\_

Date \_\_\_\_\_

`readDirSync()`

⑦ `let folderPath = 'D:\\FTP2 Dev\\Node Modules\\my Directory'`

`let folderContent = fs.readDirSync(folderPath)`

`console.log(folderContent)`

`console.log('directory content' + folderContent)`

⑧ Copy files

my directory → fs.txt

my Directory2 (src folder, dest folder)

basename

↓  
fs.txt

`let srcFolder = 'D:\\FTP2 Dev\\Node Modules\\my Directory'`

`let destFolder = 'D:\\ _____ '`

# `const path = require('path')`

`let toBeCopiedFileName = path.basename(srcFilePath)`

// basename method gets the actual name of the file or directory

`console.log(toBeCopiedFileName) → o/p → fs.txt`

`let destPath = path.join(destFolder, toBeCopiedFileName)`  
→ (2nd JS ch 10 p 2)

`console.log(destPath)`

`fs.copyFileSync(srcFilePath, destPath)`

`console.log("file copied")`