

Node JS

GEEKYSHOWS YOUTUBE CHANNEL LEARNING NOTES

Source Code - https://github.com/satyam-seth-learnings/nodejs_learning/tree/main/Geekyshows

YouTube Link - https://youtube.com/playlist?list=PLbGui_ZYuhiiSVvVP_9w57-aU7kx_H9bu

SATYAM SETH

04-09-2022

INDEX				
S. No.	Date	Title	Page No.	Teachers Sign/ Remarks
1	2-4-22	Introduction to Node JS	1	SS-08 -08
2	3-4-22	REPL	2	07-18 -18
3	3-4-22	Run First Project on Node JS Environment	2	- -
4	3-4-22	Wrapper Module	3	08-08 -08
5	20-5-22	Path Module	4	08-08 -08
6	22-5-22	How to use ES6 import statement	6	08-08 -08
7	22-5-22	File System Module	7	08-08 -08
8	24-5-22	OS Module	10	08-08 -08
9	24-5-22	Event Module	12	08-08 -08
10	25-5-22	HTTP Module and Create Web Server	13	08-08 -08
11	25-5-22	Nodemon	13	08-08 -08
12	26-5-22 26-5-22	DNS Module	14	08-08 -08
13	27-5-22	Mongoose Crash Course	16	08-08 -08
14	30-5-22	MongoDB User Authentication and	21	
15	-	Authorization and Roles	-	
16	31-5-22	Introduction to Express JS	24	
17	31-5-22	Install and Uninstall Express JS	24	
18	11-6-22	Setup Babel with Express JS	25	
19	11-6-22	Express Application Generation tool	27	
20	11-6-22	First Express JS Application	29	
21	18-6-22	Routing	30	
22	19-6-22	Router Router	35	
23	20-6-22	Route Parameter and Query String	39	
24	22-6-22	Controllers	41	
25	22-6-22	Views	41	

ARFAT
Date 2-4-22
Page 1

What is Node.js -

- Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.
- Node.js allows you to run JavaScript on the server.
- Node.js also provides library of various JavaScript modules which helps to develop web applications.
- Node.js was developed by Ryan Dahl in 2009.
- Node.js is not a language or a framework.

Advantages of Node.js -

- Open Source
- Efficient, Fast and Highly Scalable
- Event Driven
- Very Popular

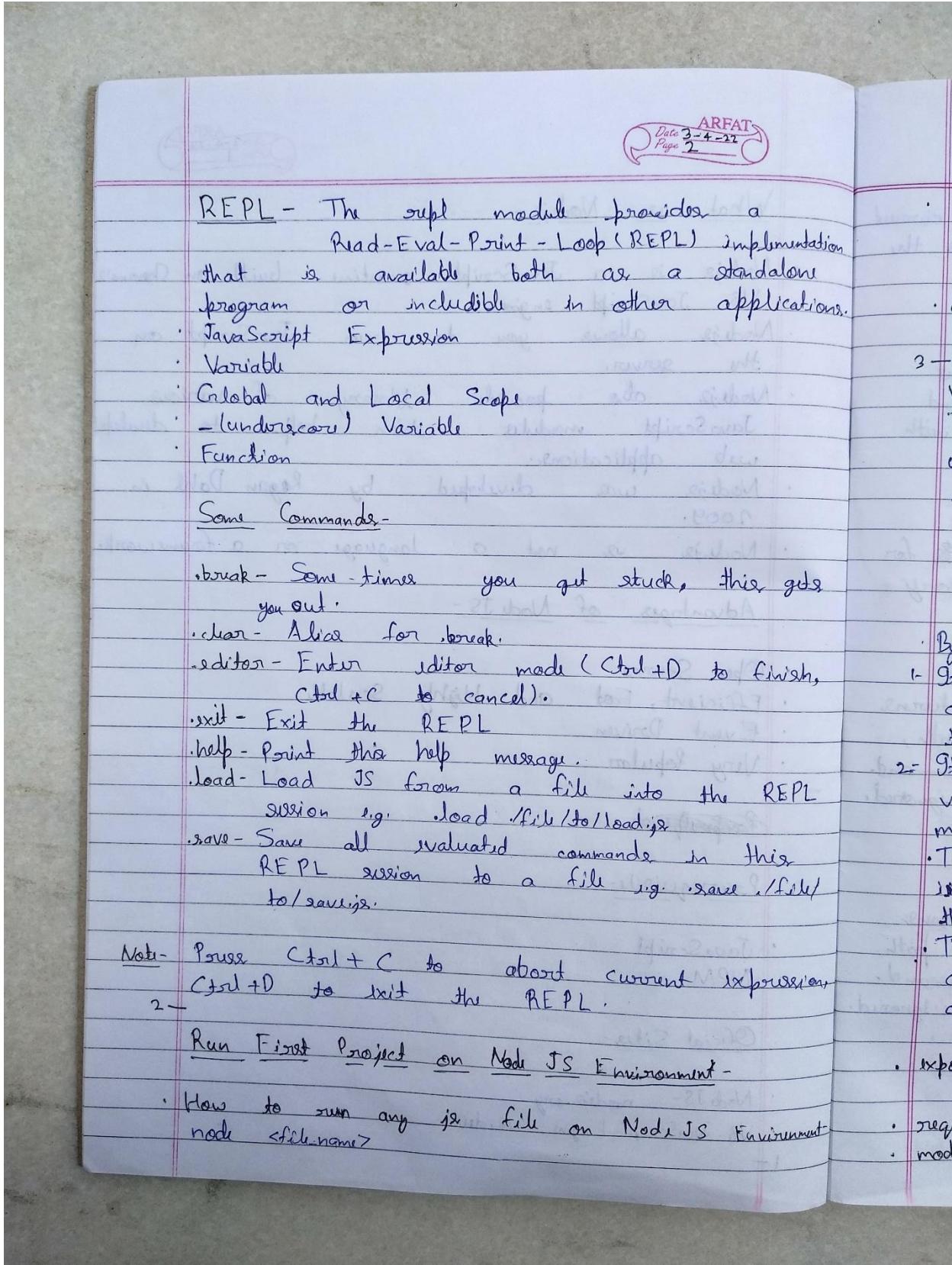
~~Prerequisite~~

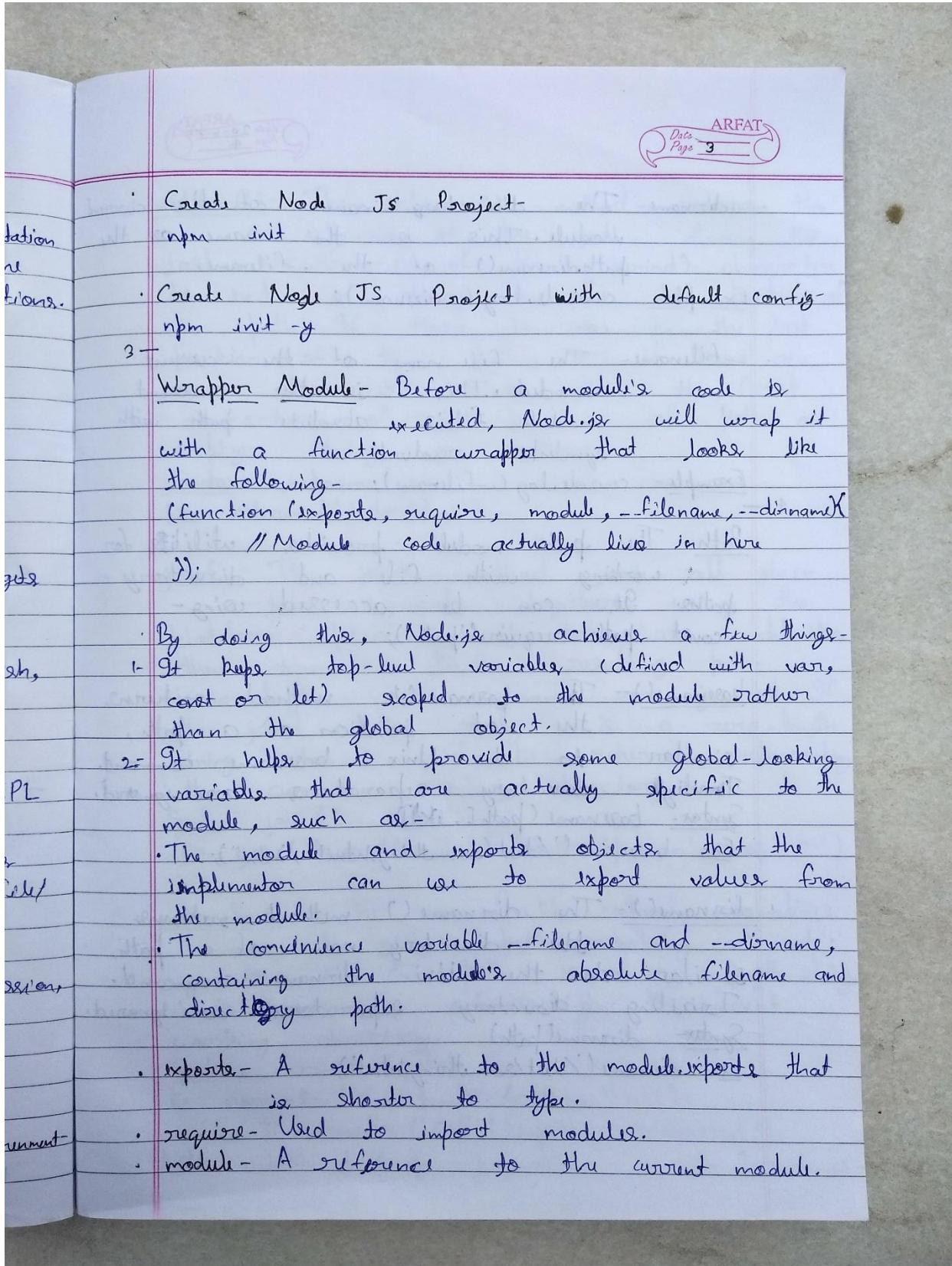
Prerequisite -

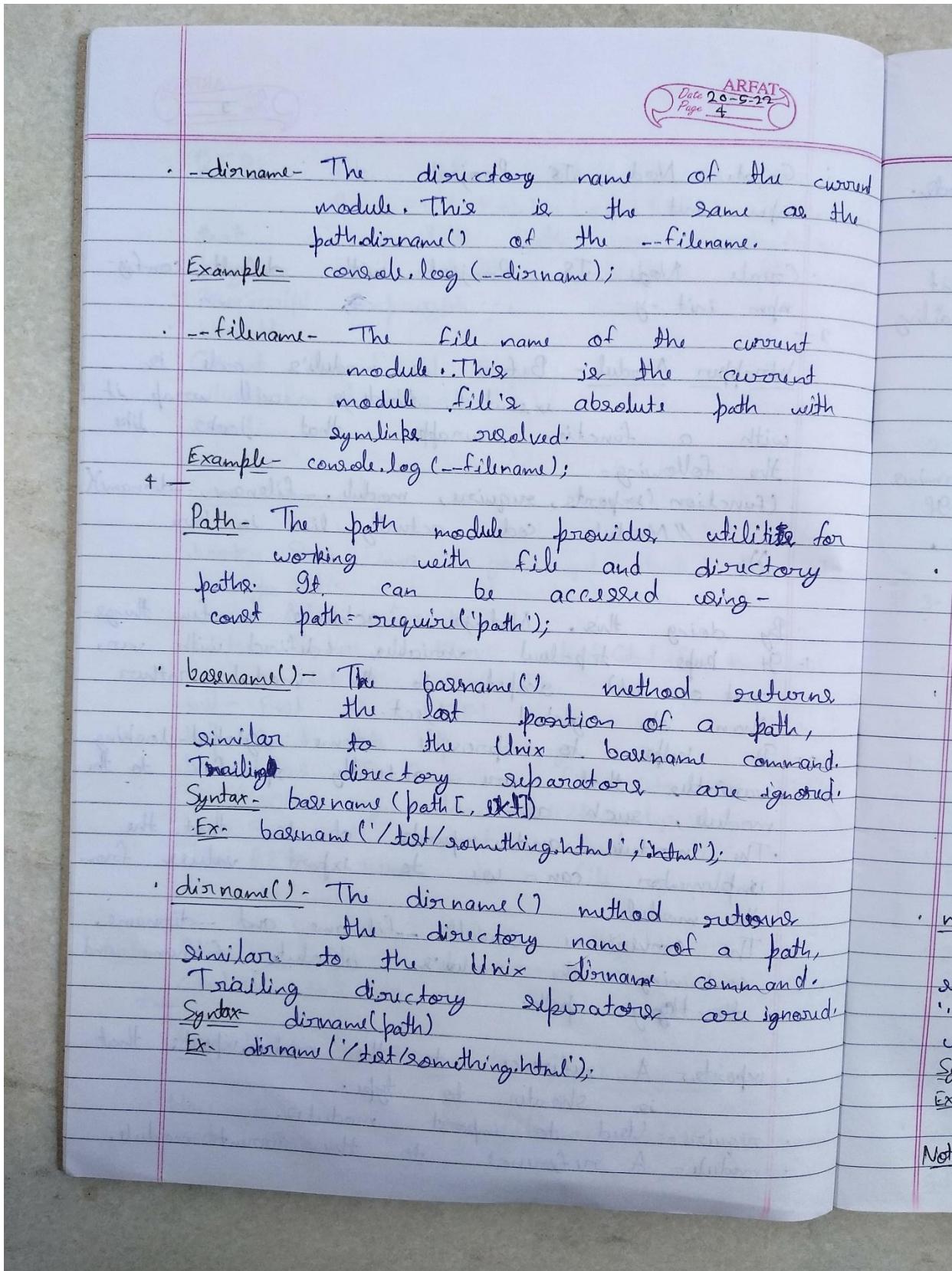
- JavaScript
- NPM

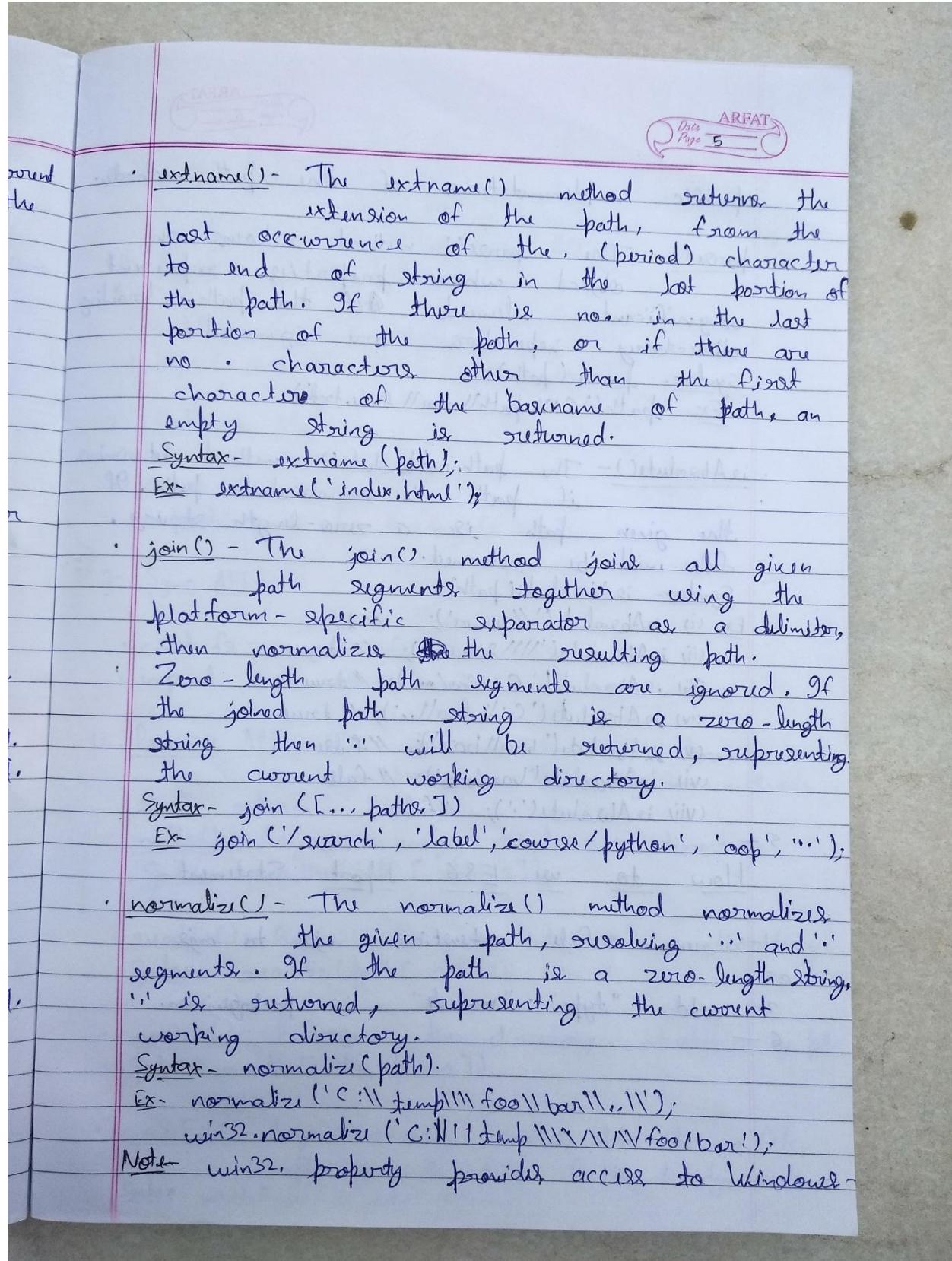
Official Site -

- Node.js - nodejs.org
- V8 JavaScript Engine - v8.dev









ARFAT
Date _____
Page 6

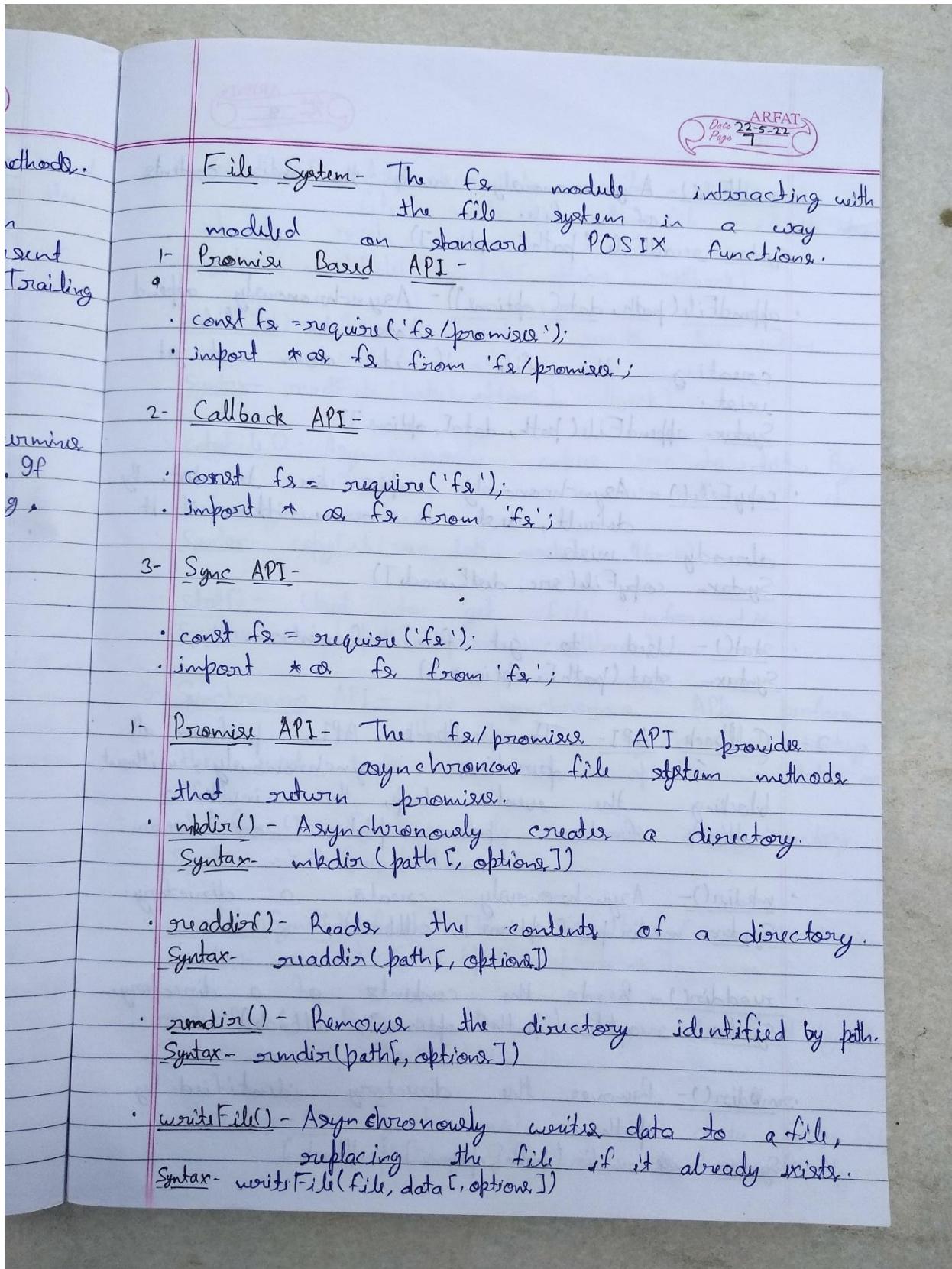
specific implementations of the path methods.

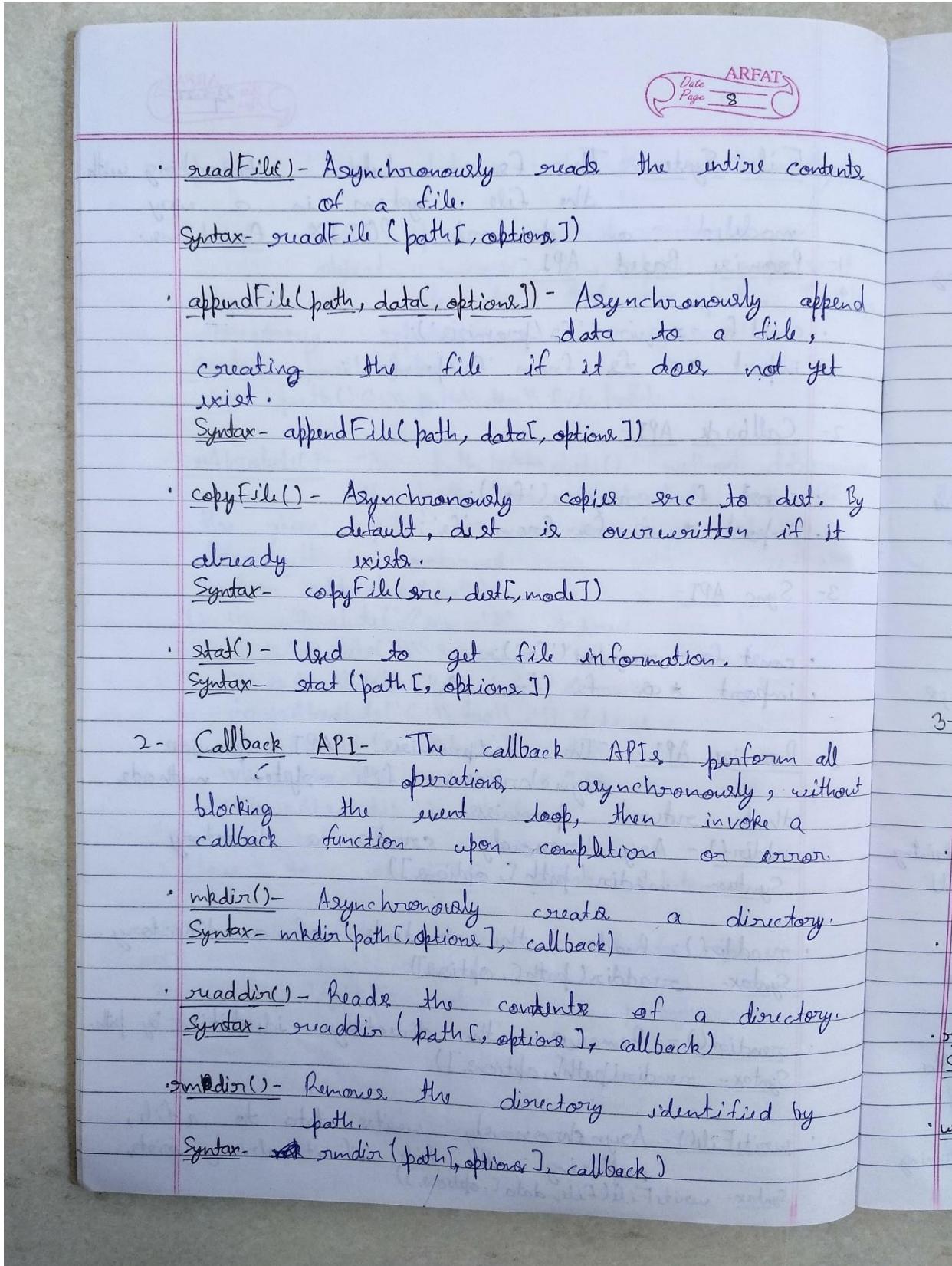
- parse() - The `parse()` method returns an object whose properties represent significant elements of the path. Trailing directory separators are ignored.
 Syntax - `parse(path);`
 Ex - `path('C:\\path\\dir\\file.txt');`
- isAbsolute() - The `path.isAbsolute()` method determines if path is an absolute path. If the given path is a zero-length string, false will be returned.
 Syntax `isAbsolute(path);`
 Ex. (i) `isAbsolute('//server');` // true
 (ii) `isAbsolute('///server');` // true
 (iii) `isAbsolute('C:/foo/..');` // true
 (iv) `isAbsolute('C:\\foo\\..');` // true
 (v) `isAbsolute('bar\\bar');` // false
 (vi) `isAbsolute('bar\\bar');` // false
 (vii) `isAbsolute('.');` // false

5 -

How to use ESG Import Statement -

- 1- rename file extension .js to .mjs
 or
- 2- add "type": "module" in package.json.





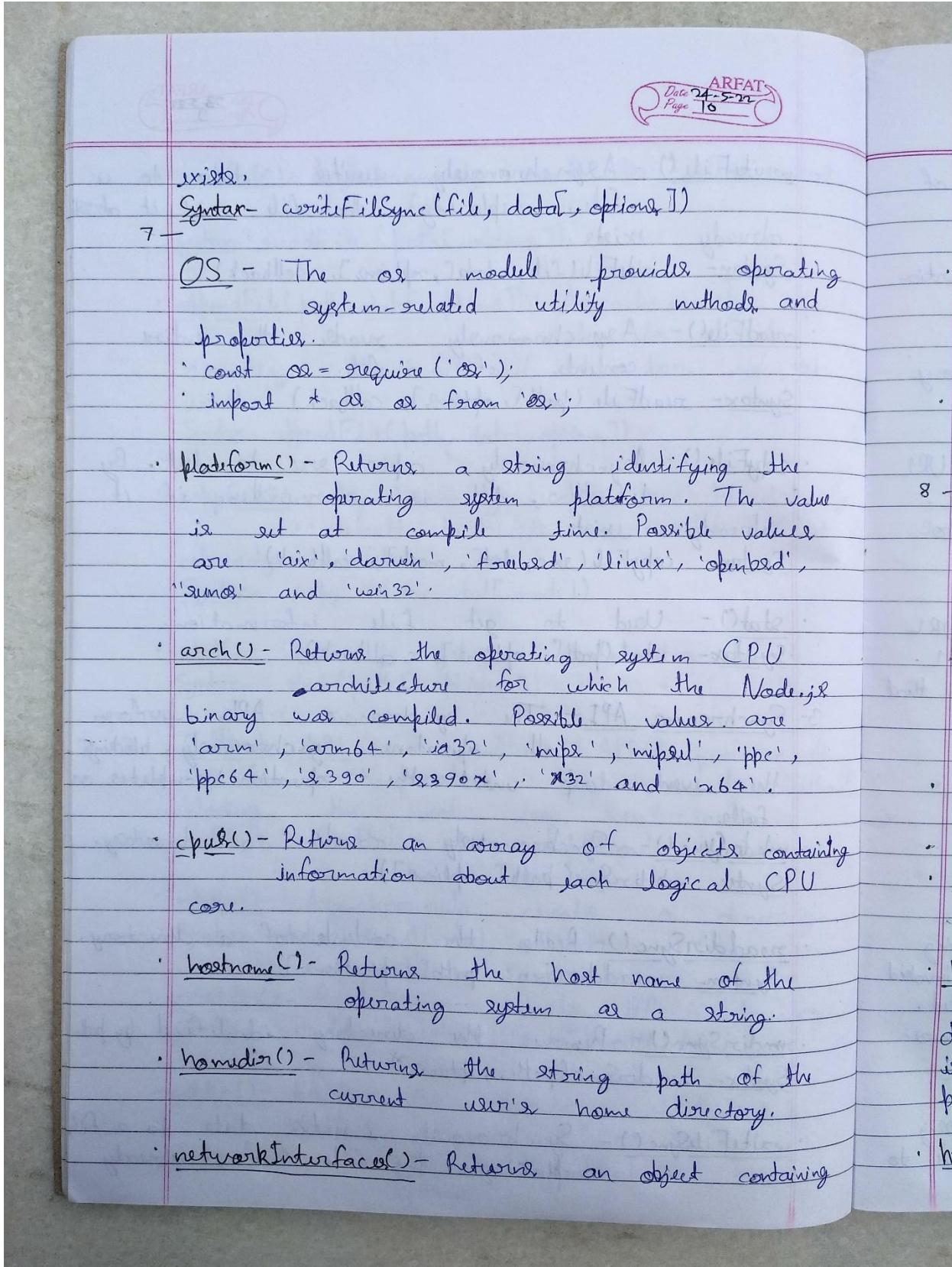
Date 23-5-22
Page 9 ARFAT

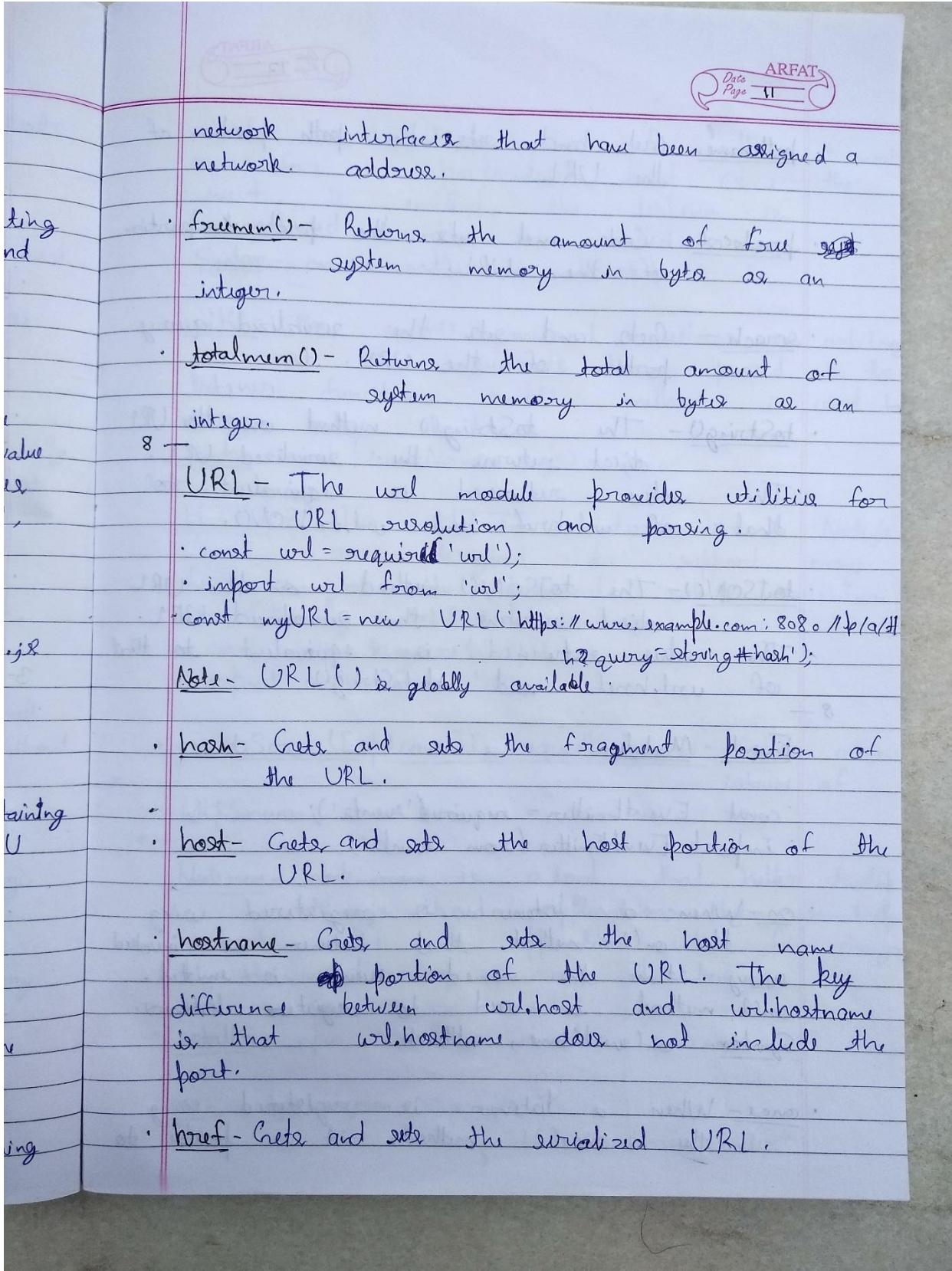
intents

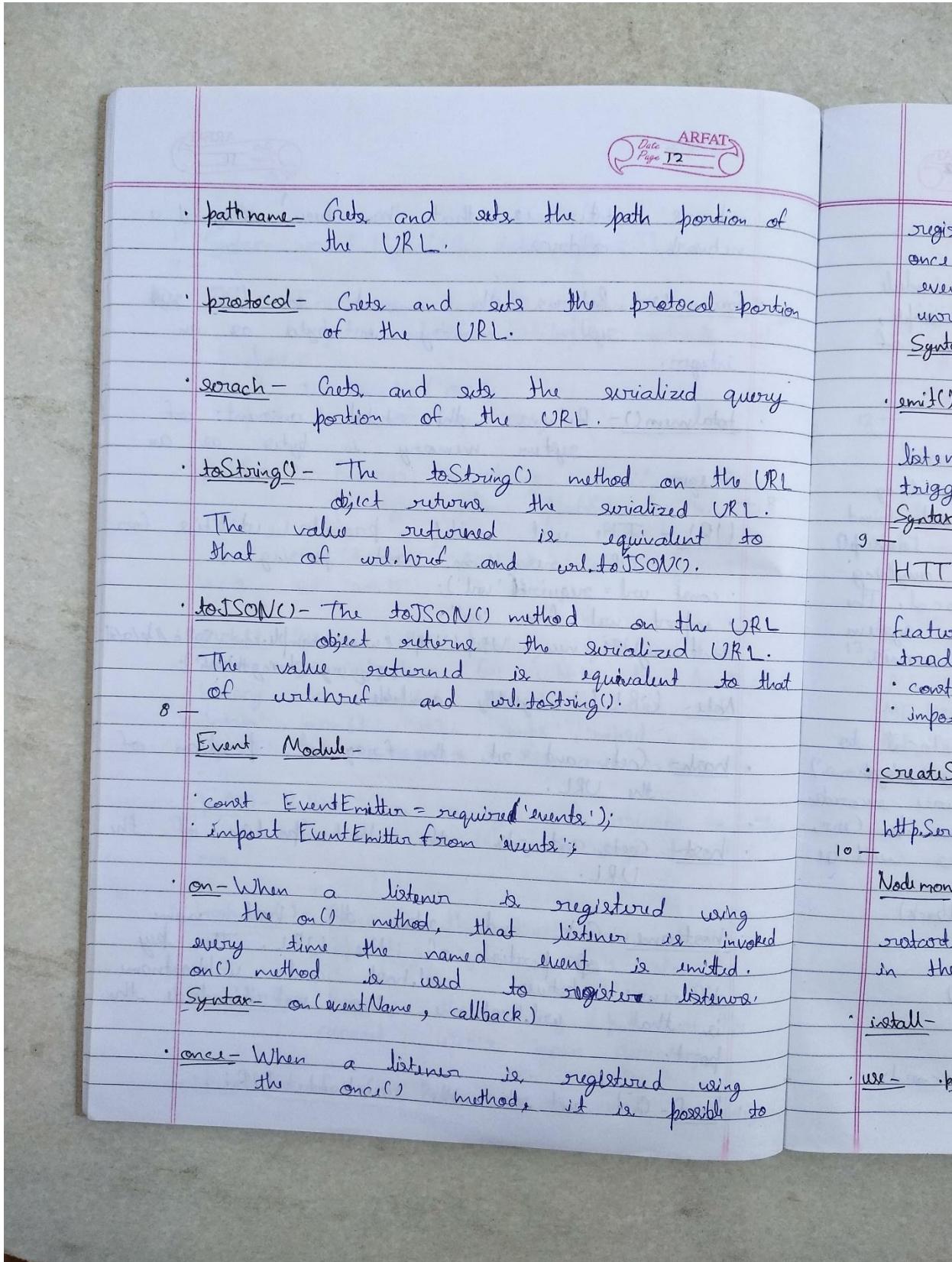
- writeFile() - Asynchronously writes data to a file, replacing the file if it already exists.
Syntax - `writeFile(file, data[, options], callback)`
- readFile() - Asynchronously reads the entire contents of a file.
Syntax - `readFile(path[, options], callback)`
- copyFile() - Asynchronously copies src to dest. By default, dest is overwritten if it already exists.
Syntax - `copyFile(src, dest[, mode], callback)`
- stat() - Used to get file information.
Syntax - `stat(path[, options], callback)`

3- Synchronous API - The synchronous APIs perform all operations synchronously, blocking the event loop until the operation completes or fails.

- mkdirSync() - Synchronously creates a directory.
Syntax - `mkdirSync(path[, options])`
- readdirSync() - Reads the contents of a directory.
Syntax - `readdirSync(path[, options])`
- rmdirSync() - Removes the directory identified by path.
Syntax - `rmdirSync(path[, options])`
- writeFileSync() - Synchronously writes data to a file, replacing the file if it already







ARFAT
Date: 25-5-22
Page 13

register a listener that is called at most once for a particular event. Once the event is emitted, the listener is unregistered and then called.

Syntax - `once(eventName, callback)`

• emit() - The `emit()` method allows an arbitrary set of arguments to be passed to the listener functions. `emit()` method is used to trigger the event.

Syntax - `emit(eventName, args)`

9 - HTTP - The HTTP interfaces in Node.js are designed to support many features of the protocol which have been traditionally difficult to use.

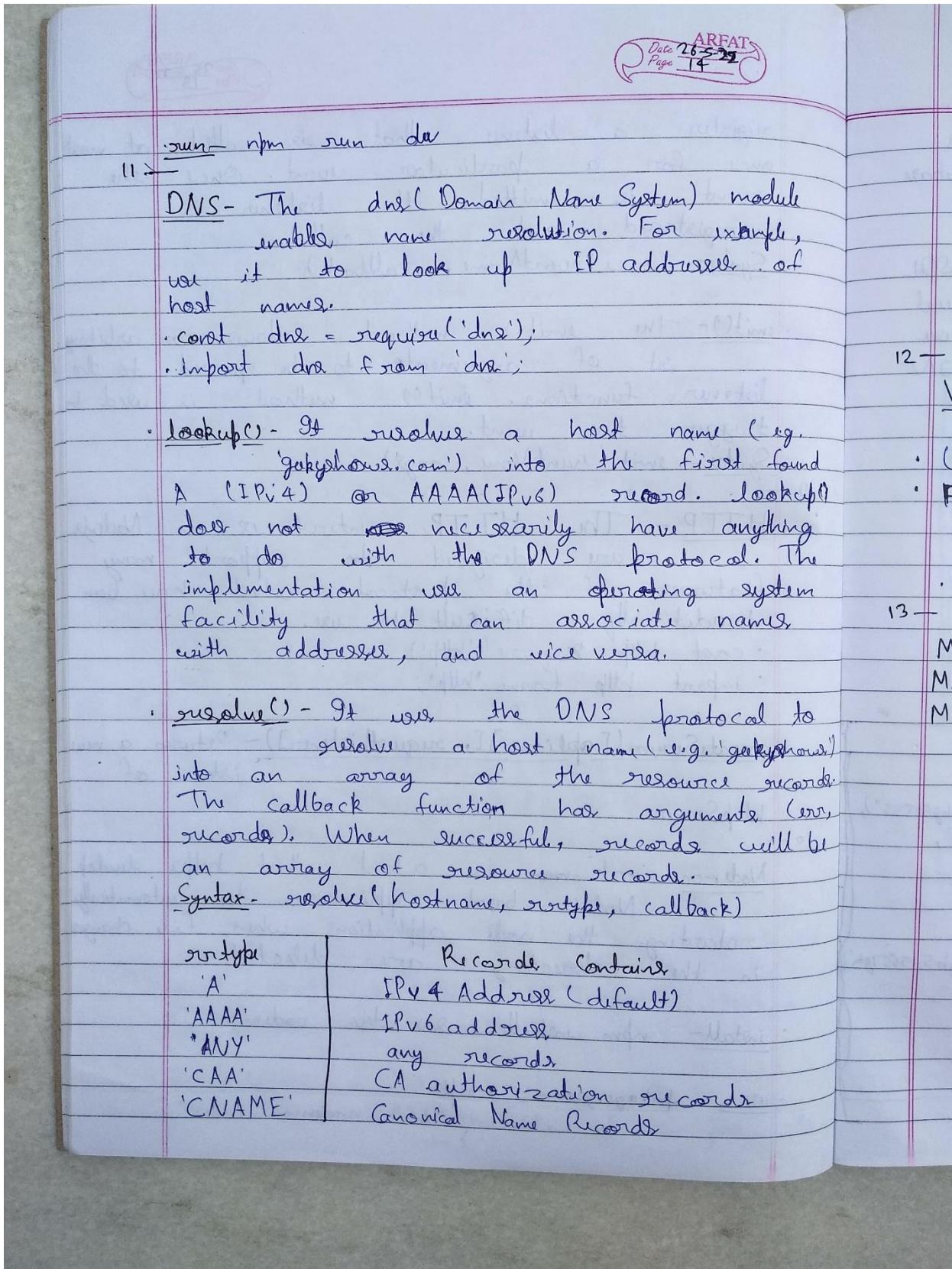
- `const http = require('http');`
- `import http from 'http';`

• createServer([options], [requestListener]) - Returns a new instance of `http.Server`.

10 - Nodemon - `nodemon` is a tool that helps develop Node.js based applications by automatically restarting the node application when file changes in the directory are detected.

- install - `npm install --save-dev nodemon`
- use - `package.json` -

```
scripts: {
  "dev": "nodemon index"
```



		ARFAT Date _____ Page 15
'MX'	Mail Exchange Records	
'NAPTR'	Name Authority Pointer Records	
'NS'	Name Server Records	
'PTR'	Pointer Records	
'SOA'	Start of Authority Records	
'SRV'	Service Records	
'TXT'	Text Records	
12 -	<u>What Next -</u>	
	<ul style="list-style-type: none"> • Update Yourself • Framework - Angular, ReactJS, VueJS, ExpressJS, Koa etc. • http://nodejs.org/en/docs 	
13 -	<p>MERN - MongoDB + Express JS + React JS + Node</p> <p>MEAN - MongoDB + Express JS + Angular JS + Node</p> <p>MEVN - MongoDB + Express JS + Vue JS + Node</p>	