Node js

Date ...../...../..... No ....

cmd → node -v → ✓

↳ cls → clear cmd

node → علشان تكتب فيه كأنك console ذي ما هو

Process يعمل exit → process.exit()

Blocking Vs Non Blocking

```
const user = getuser(1)
console.log(user);
```

```
getuser(1, (user) => {
    console.log(user)
});
```

the fast

أسرع

Node js uses event driven, non blocking I/O (input output) model that makes it lightweight and efficient

---

Section 3

* You should import fs → const fs = require('fs')

fs.writeFileSync('notes.txt', ← any name you want
'الكلام');

↓ بعدها node app.js → يتحول الـ terminal ← Run الملف ده

---

* افرض انت عامل كذا ملف وعاوز تشوف output بتاعهم كلهم

← هتعمل require('اسم الملف') في اول سطر في app مثلاً

وبعدها ← node app.js

output → الملف الاول

↳ ثم app output

* علشان تبعت دالة او متغير لملف تاني لازم تعملها module.export = الاسم

Date ……/……/…… No ……

Importing npm Modules

npm init → وتعمل ok لكل الاختيارات → Package.json
↓
Java Script object notation

npm موقع → Validator Package → npm i Validator

↳ Chalk → npm i Chalk → حط (type: module) ES6

Getting input from user

node app.js add

هيطبعلي الـ array كل كلمة جديدة تبع add

→ console.log (process.argv)

↓ الناتج

[..., مسار الناتج كامل ---

..., --- الناتج

add]

node app.js --title = "عنوانه" ← لو عاوز تحطلي حاجة وليها وصف خاص بيها

→ npm i yargs → const yargs = require ("yargs");

yargs.command ({

command: "add"

describe: ___

builder: {

title: {

___

___

}}

Date ..../..../.... No ....

```
const book = {
  title: '—',
  author: '—'
}
```

Storing data with JSON

JSON. Stringify (object (book) , array)
↓ to convert object or array to JSON

JSON . Parse (   )
↓ to convert from JSON to object.

debugging Node.js

[1] Console . log → Parameters على الخطأ وتفضل تجرب في الموضوع في الدالة

[2] debugger → in terminal → node inspect app.js add
↓
-- inspect - brk ---

على حسب الـ version ← بتاع النود Node

edge : //inspect /# devices ← صفحة في متصفح وبعدها
Chrom :

← ممكن نعمل Add Folder to work space ← وهيفتح folder الملف اللي انا عاوزه

* اضغط على ▷ ← هيعمل run ← الـ debugger

لو debugger وقف ← اكتب في console vsc ← restart

لو عاوز تقفل ← اقفله عادي واخرج واكتب ← CTRL + C اضغط مرتين

Andrew

شرح كودي
فيديو 30

Section 6 Weather app

Date ..../..../.... No ....

Asychronus Basics

مثال

```
console.log ('Start 1')
Settime out( () => {
    console.log ('Start 2')
}, 2000)
Set time out ___
    console.log ('Start 3').
}, 0 )

console.log ( 'start 4')
```

out Put

Start 1
Start 4
Start 3
Start 2

Call Stack هيدخل ال Node APIs ثم Call Queue ثم يروح لل Call Stack لما يكون فاضي

Call back function → is a function we Provide it as an argument to another function

```
Set time out( () => {
    console.log( ___ )
}, 3000 )
```

object destructuring

```
const Product = {
    label: 'Red notebook',
    Price: 3,
    Stock: 201,
    SalePrice: undefined
}
```

بدل مثلا

```
const label = Product.label
  "   Stock =   "  .stock
```

```
const {label, Stock} = Product
```

url / Products ? Search = games

Date ..../..../.... No ....

Query String:

the client, when he write the url should put query string then the server will use this data and send response back

heroku

heroku create → ب
git remote → heroku origin ← هتطلعك
git push heroku main

ولكن تام اردت التعديل

لو فى اى تغير ↓ → git اعمل
git status
git add .
git commit -m " "
git push

↳ git push heroku main

MongoDB

/users/softzone/mongodb/bin/mongod.exe
--dbpath=/users/softzone/mongodb-data

REST API

Representational State Transfer - Application Programming interface

Date ..../..../.... No ....

Promise chaining

Promise

```
Const add = (a,b) => {
  return new Promise ((resolve, reject)
  => {
    setTimeout (() => {
      resolve (a+b)
    }, 2000)
  })
}
```

Promise chaining

```
add (1,2).then ((Sum) => {
  . console.log (Sum)
  return add (sum, 4)
}).then ((Sum2) => {
  console.log (Sum2)
}).catch ((e) => {
  console.log (e)
})
```

Async/Await

Async function return Promise

It is readable with async and await and you can use all sums but Promise chaining you can't use alot of one Sum in then()

```
Const doWork = async () => {
Const sum = await add (1,99)
  ~  ~2 =  ~  ~ (Sum, 50)
  ~  ~3 =  ~  ~ (Sum2, 3)
  return Sum3
}
```

Date / / No

npm bcryptjs → bcrypt.hash(password, 8)

npm jsonwebtoken

npm Pagination

npm multer → for uploading files

↳ npm sharp → for images → crop and others

t: Sendgrid.com

Testing

, Super test

, Mocha → JS test framework running on Node.JS

, Jest → ↙

↳ npm i jest

① ↳ "test": "jest" → Package.json

② make new file with → name.test.js

Why test → Safe times

↳ Create reliable Softwares

↳ Gives flexibility to developers, Refactoring...