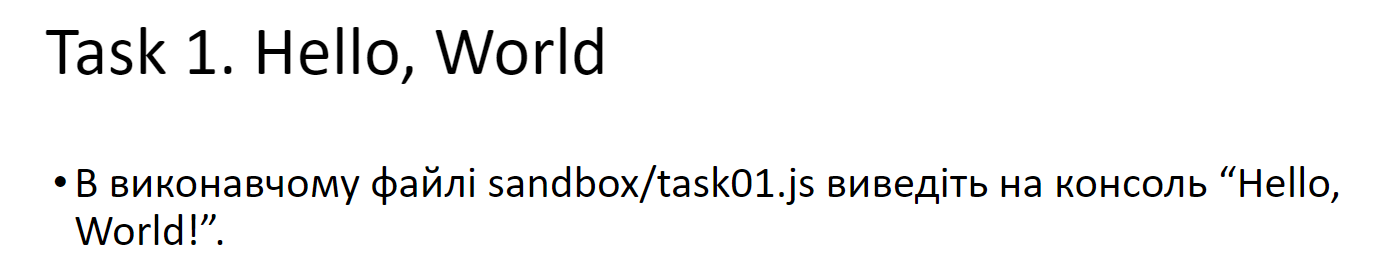
**Лабораторна робота №1**

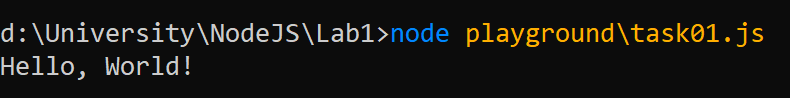
**Робота з файлами, JSON, об’єктами, модулями, npm, lodash, nodemon, yargs**

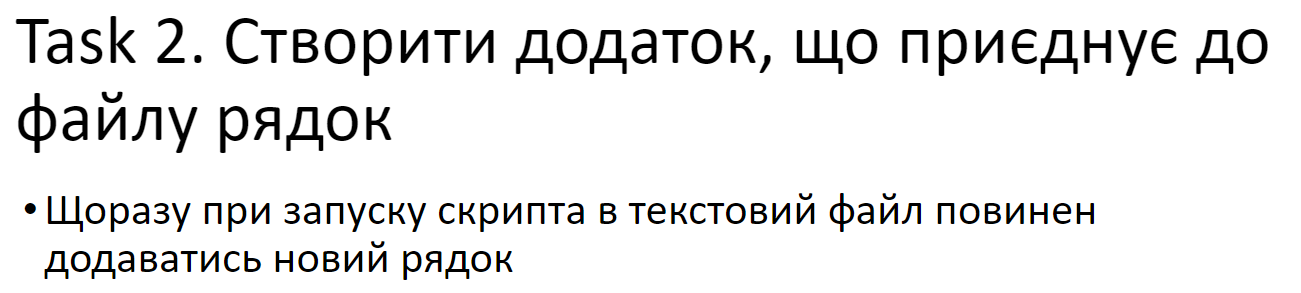
Регенель Тимофій, ВТ-22-2

Репозиторій: https://github.com/TimothySpaceman/nodejs-lab1



console.log("Hello, World!")





const fs = require('fs')

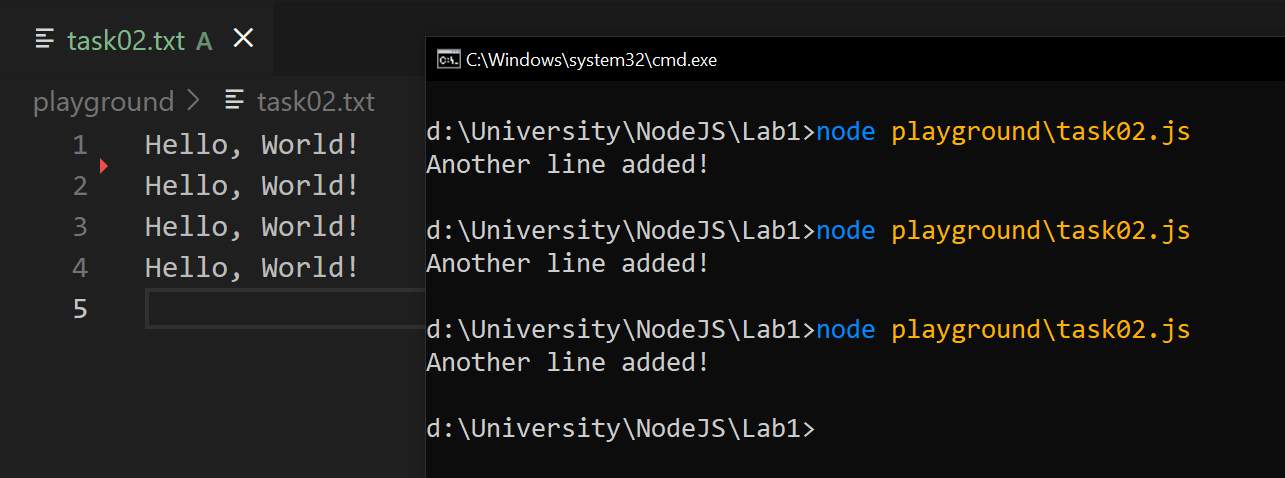
const fileName = "playground/task02.txt"

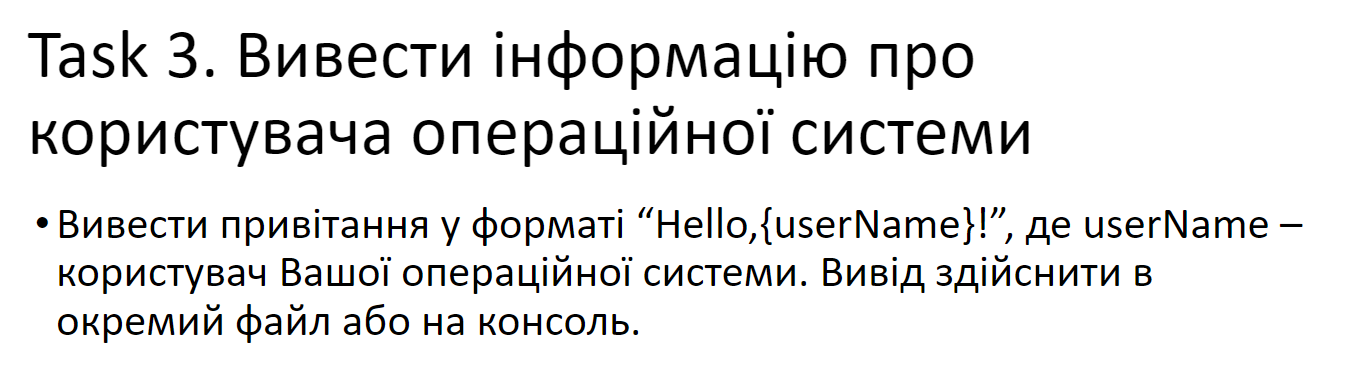
const message = "Hello, World!\n"

fs.appendFile(fileName, message, (err) => {

console.log(err || "Another line added!")

})



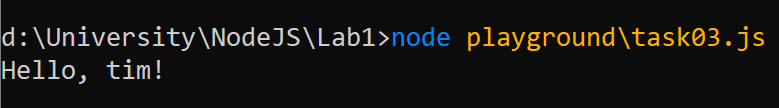


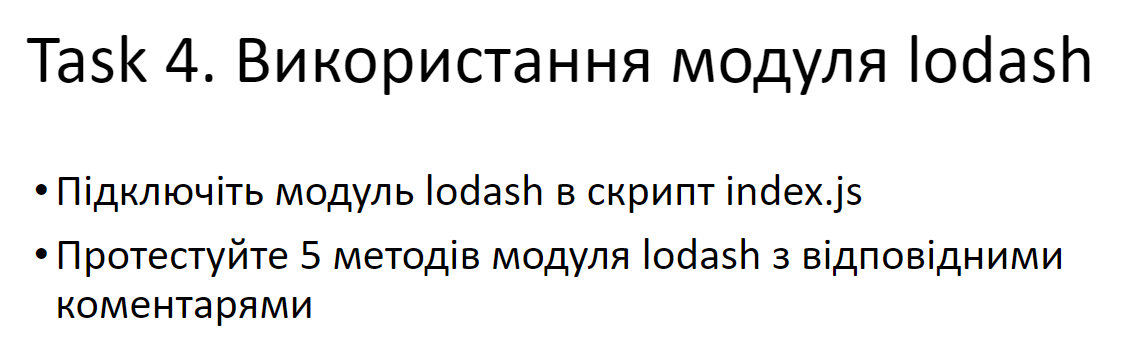
const os = require('os')

const username = os.userInfo().username

const message = `Hello, ${username}!`

console.log(message)





const \_ = require("lodash")

//1 - chunk()

//Splits array into chunks the length of size

{

const arr = ["a", "b", "c", "d", "e"]

const chunkLength = 2

const chunks = \_.chunk(arr, chunkLength)

console.group("Method 1 - chunk()")

console.log("Array", arr)

console.log("Chunked array", chunks, "\n")

console.groupEnd()

}

//2 - inRange()

//Checks x number is in the range (a, b)

{

const tests = [

{

x: 3,

a: -5,

b: 5

},

{

x: 3,

a: 0,

b: 3

},

{

x: -2,

a: 4

}

]

console.group("Method 2 - inRange()")

tests.forEach(t => {

console.log("x: ", t.x)

console.log("a: ", t.a)

console.log("b: ", t.b)

console.log("inRange: ", \_.inRange(t.x, t.a, t.b), "\n")

})

console.groupEnd()

}

//3 - at()

//Takes paths and creates an array with corresponding values

{

const data = {

id: 73580962008,

username: "Tim",

stats: {

bestTurns: 92,

moneySpent: 2930000,

achievements: [

"Bought an aircraft carrier",

"Promoted the construction of a private subway",

"Introduced Minecraft in the curriculum"

]

}

}

const paths = ["username", "stats.bestTurns", "stats.achievements[0]"]

const result = \_.at(data, paths)

console.group("Method 3 - at()")

console.log("Data: ", data)

console.log("Paths: ", paths)

console.log("Result: ", result, "\n")

console.groupEnd()

}

//4 - findKey()

//Returns the key of the first element that meets the criteria

{

const data = {

darkTheme: true,

fontSize: 18,

autoSave: false,

autoComplete: true

}

const criteria = x => !x

const key = \_.findKey(data, criteria)

console.group("Method 4 - findKey()")

console.log("Data: ", data)

console.log("Criteria: ", criteria.toString())

console.log("findKey(): ", key, "\n")

console.groupEnd()

}

//5 - invert()

//Swaps keys and values of an object

{

const data = {

id: 195,

email: "example@gmail.com",

username: "TimothySpaceman"

}

const invertedData = \_.invert(data)

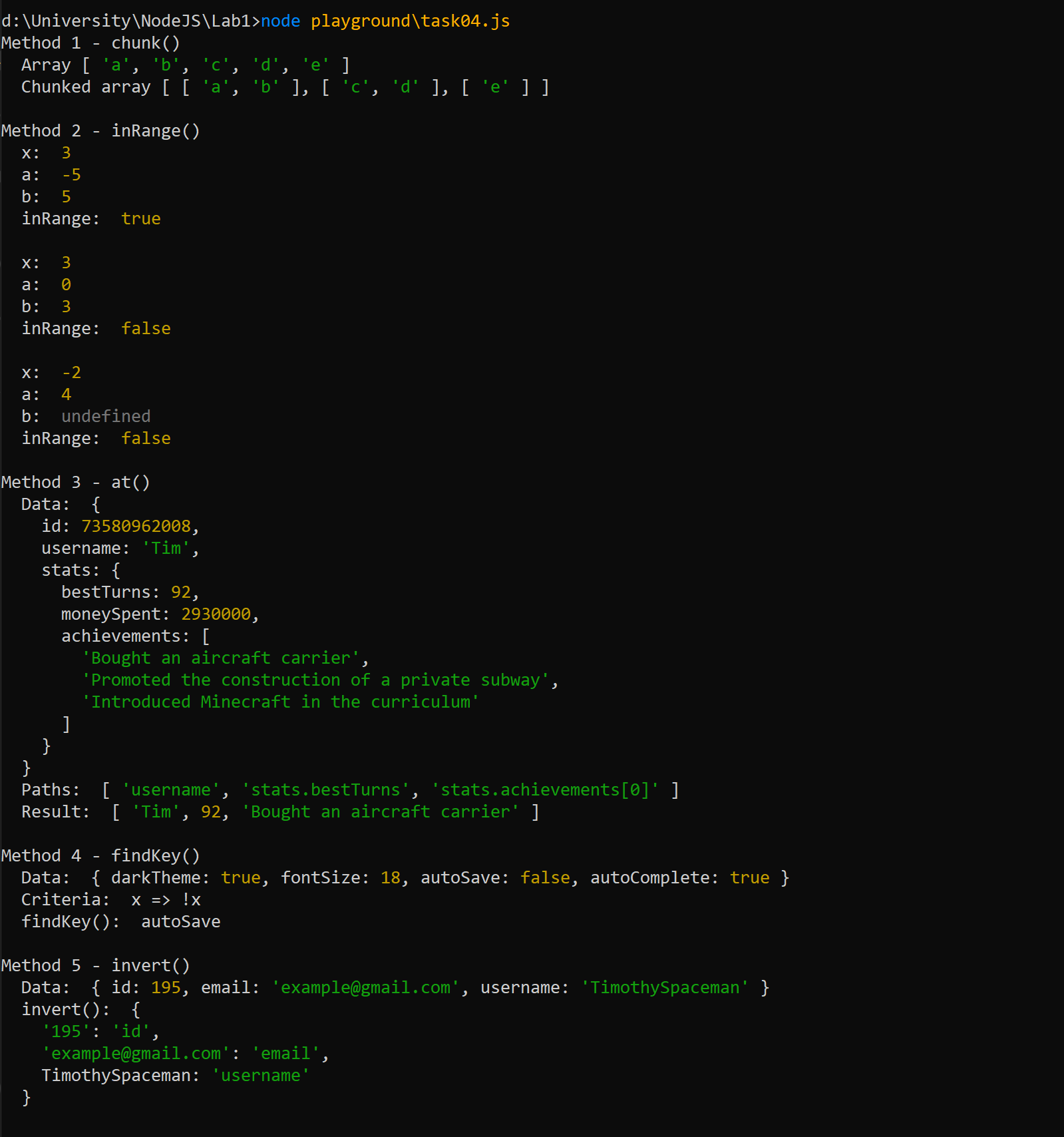
console.group("Method 5 - invert()")

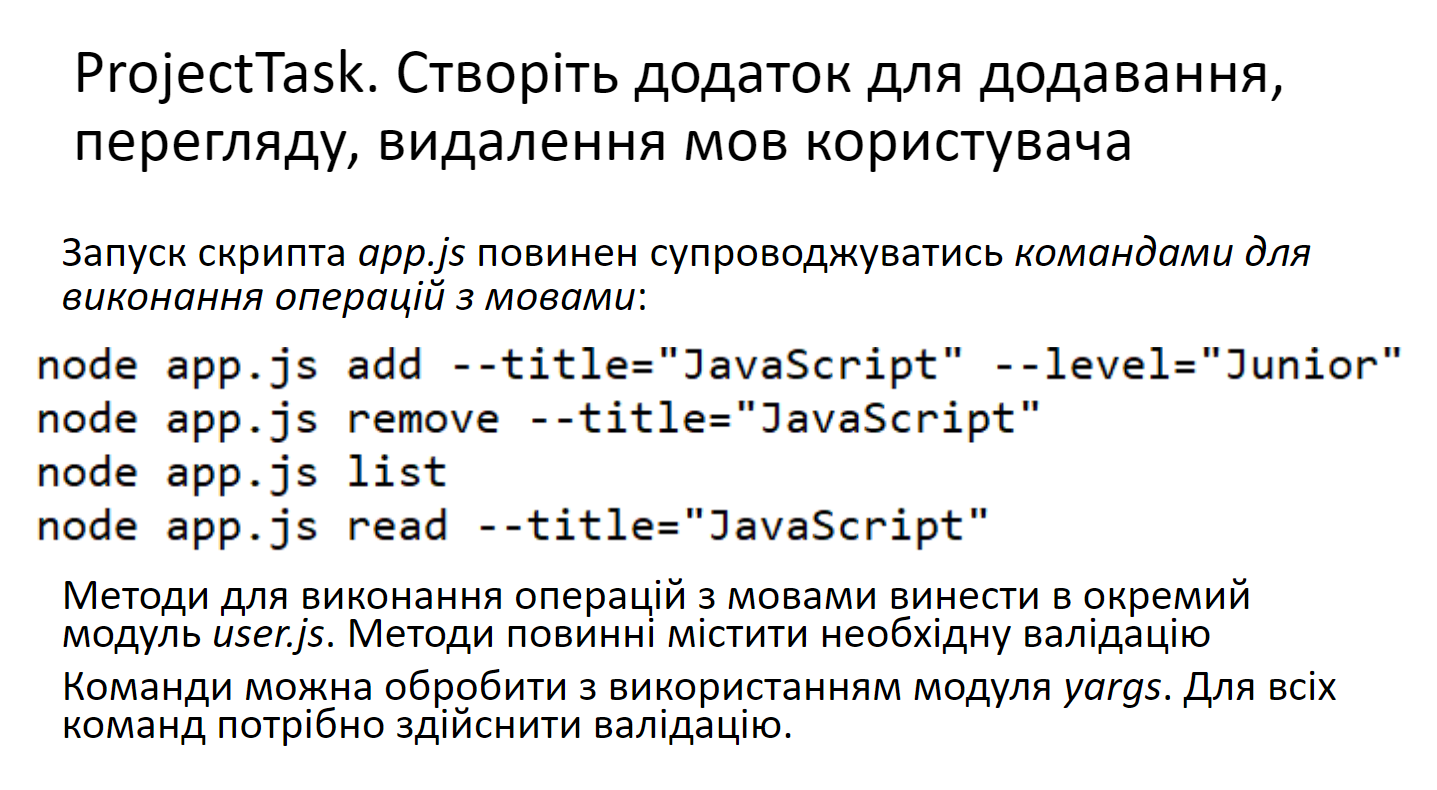
console.log("Data: ", data)

console.log("invert(): ", invertedData, "\n")

console.groupEnd()

}





user.js

const fs = require("fs")

const \_ = require("lodash")

const fileName = "user.json"

module.exports.load = () => {

try {

const stringData = fs.readFileSync(fileName, "utf8")

return JSON.parse(stringData)

} catch (err) {

console.log("Error reading the file: ", err)

}

}

module.exports.save = (data) => {

try {

const stringData = JSON.stringify(data)

fs.writeFileSync(fileName, stringData)

} catch (err) {

console.log("Error writing the file: ", err)

}

}

module.exports.add = ({ title, level }) => {

if (!title) {

return "Title is not provided"

}

if (!level) {

return "Level is not provided"

}

const data = module.exports.load()

if (\_.findIndex(data.languages, l => l.title == title) >= 0) {

return "Language is already in the list"

}

data.languages.push({ title: title, level: level })

module.exports.save(data)

return `Successfully added ${title} to the list`

}

module.exports.remove = ({ title }) => {

if (!title) {

return "Title is not provided"

}

const data = module.exports.load()

const index = \_.findIndex(data.languages, l => l.title == title)

if (index < 0) {

return "Language is not in the list"

}

data.languages = data.languages.filter(l => l.title != title)

module.exports.save(data)

return `Successfully removed ${title} from the list`

}

module.exports.set = ({ title, level }) => {

if (!title) {

return "Title is not provided"

}

if (!level) {

return "Level is not provided"

}

const data = module.exports.load()

const index = \_.findIndex(data.languages, l => l.title == title)

if (index < 0) {

return "Language is not in the list"

}

data.languages[index].level = level

module.exports.save(data)

return `Successfully changed ${title} level to ${level}`

}

module.exports.list = () => module.exports.load().languages

module.exports.read = ({ title }) => {

if (!title) {

return "Title is not provided"

}

const data = module.exports.load()

const index = \_.findIndex(data.languages, l => l.title == title)

if (index < 0) {

return "Language is not in the list"

}

return data.languages[index]

}

app.js

const { argv } = require("yargs")

const u = require("./user.js")

var yargs = require("yargs").strict().command(

"add",

"",

{

title: {

demand: true,

type: "string"

},

level: {

demand: true,

type: "string"

}

},

(argv) => console.log(u.add(argv))

).command(

"set",

"",

{

title: {

demand: true,

type: "string"

},

level: {

demand: true,

type: "string"

}

},

(argv) => console.log(u.set(argv))

).command(

"remove",

"",

{

title: {

demand: true,

type: "string"

}

},

(argv) => console.log(u.remove(argv))

).command(

"list",

"",

{},

() => console.log(u.list())

).command(

"read",

"",

{

title: {

demand: true,

type: "string"

}

},

(argv) => console.log(u.read(argv))

).argv

