

Лабораторна робота №1

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

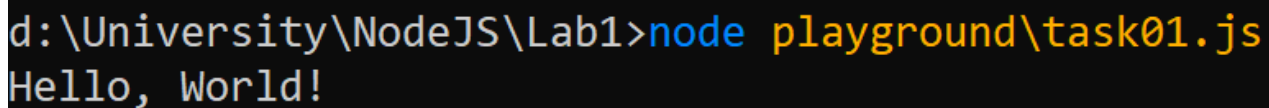
Регенель Тимофій, ВТ-22-2

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

Task 1. Hello, World

- В виконавчому файлі `sandbox/task01.js` виведіть на консоль "Hello, World!".

```
console.log("Hello, World!")
```



```
d:\University\NodeJS\Lab1>node playground\task01.js
Hello, World!
```

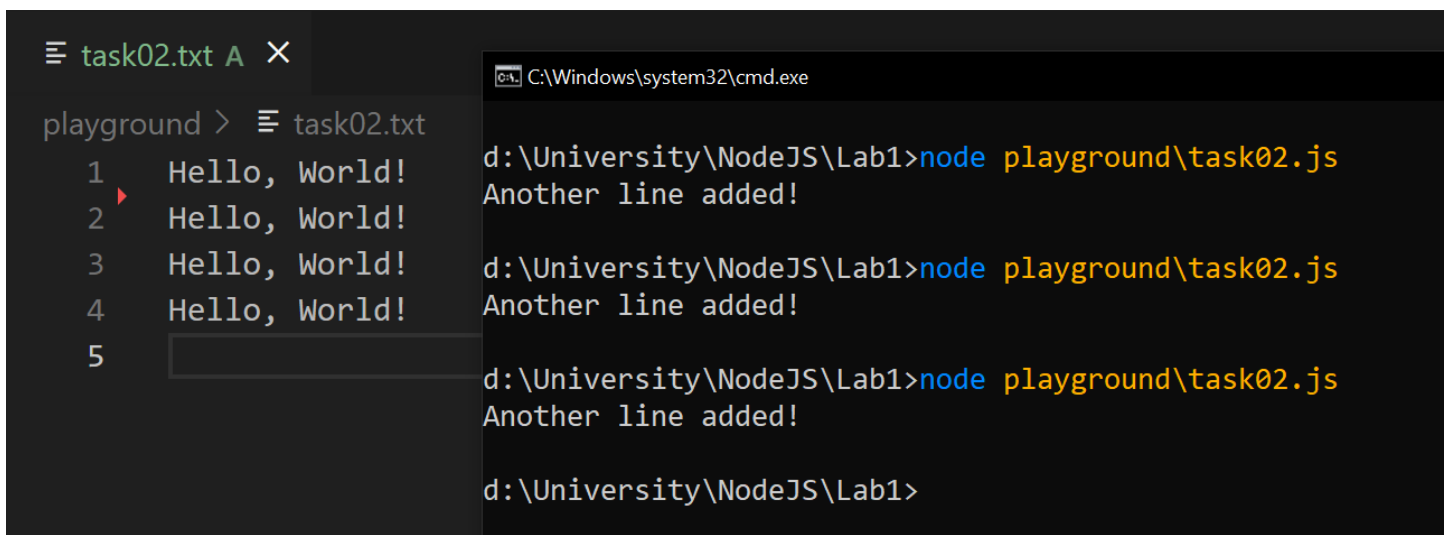
Task 2. Створити додаток, що приєднує до файлу рядок

- Щоразу при запуску скрипта в текстовий файл повинен додаватись новий рядок

```
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!")
})
```



The screenshot shows a code editor on the left with a file named `task02.txt` open. The file contains five lines, each with the text "Hello, World!". The editor is in "playground" mode. On the right, a terminal window titled `C:\Windows\system32\cmd.exe` shows the command `node playground\task02.js` being executed multiple times. Each execution results in the output "Another line added!".

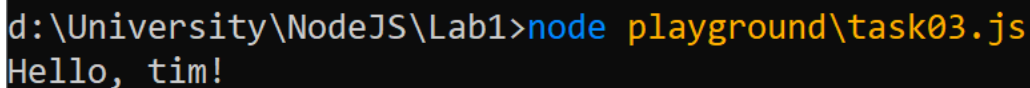
Task 3. Вивести інформацію про користувача операційної системи

- Вивести привітання у форматі "Hello,{userName}!", де userName – користувач Вашої операційної системи. Вивід здійснити в окремий файл або на консоль.

```
const os = require('os')

const username = os.userInfo().username
const message = `Hello, ${username}!`

console.log(message)
```



```
d:\University\NodeJS\Lab1>node playground\task03.js
Hello, tim!
```

Task 4. Використання модуля lodash

- Підключіть модуль lodash в скрипт index.js
- Протестуйте 5 методів модуля lodash з відповідними коментарями

```
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()  
  }  
}
```

```
d:\University\NodeJS\Lab1>node playground\task04.js
```

```
Method 1 - chunk()
```

```
Array [ 'a', 'b', 'c', 'd', 'e' ]  
Chunked array [ [ 'a', 'b' ], [ 'c', 'd' ], [ 'e' ] ]
```

```
Method 2 - inRange()
```

```
x: 3  
a: -5  
b: 5  
inRange: true
```

```
x: 3  
a: 0  
b: 3  
inRange: false
```

```
x: -2  
a: 4  
b: undefined  
inRange: false
```

```
Method 3 - at()
```

```
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'  
    ]  
  }  
}  
Paths: [ 'username', 'stats.bestTurns', 'stats.achievements[0]' ]  
Result: [ 'Tim', 92, 'Bought an aircraft carrier' ]
```

```
Method 4 - findKey()
```

```
Data: { darkTheme: true, fontSize: 18, autoSave: false, autoComplete: true }  
Criteria: x => !x  
findKey(): autoSave
```

```
Method 5 - invert()
```

```
Data: { id: 195, email: 'example@gmail.com', username: 'TimothySpaceman' }  
invert(): {  
  '195': 'id',  
  'example@gmail.com': 'email',  
  TimothySpaceman: 'username'  
}
```

ProjectTask. Створіть додаток для додавання, перегляду, видалення мов користувача

Запуск скрипта *app.js* повинен супроводжуватись командами для виконання операцій з мовами:

```
node app.js add --title="JavaScript" --level="Junior"
node app.js remove --title="JavaScript"
node app.js list
node app.js read --title="JavaScript"
```

Методи для виконання операцій з мовами винести в окремий модуль *user.js*. Методи повинні містити необхідну валідацію

Команди можна обробити з використанням модуля *yargs*. Для всіх команд потрібно здійснити валідацію.

```
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

```

```

d:\University\NodeJS\Lab1>node app.js add --title "JS" --level "Intermediate"
Language is already in the list

d:\University\NodeJS\Lab1>node app.js add --title "JSX" --level "Intermediate"
Successfully added JSX to the list

d:\University\NodeJS\Lab1>node app.js set --title "JSX" --level "Advanced"
Successfully changed JSX level to Advanced

d:\University\NodeJS\Lab1>node app.js read --title "JSX"
{ title: 'JSX', level: 'Advanced' }

d:\University\NodeJS\Lab1>node app.js list
[
  { title: 'JS', level: 'Advanced' },
  { title: 'TS', level: 'Intermediate' },
  { title: 'C#', level: 'Intermediate' },
  { title: 'Python', level: 'Hate' },
  { title: 'JSX', level: 'Advanced' }
]

d:\University\NodeJS\Lab1>node app.js remove --title "JSX"
Successfully removed JSX from the list

d:\University\NodeJS\Lab1>node app.js remove --title "JSX"
Language is not in the list

d:\University\NodeJS\Lab1>node app.js list
[
  { title: 'JS', level: 'Advanced' },
  { title: 'TS', level: 'Intermediate' },
  { title: 'C#', level: 'Intermediate' },
  { title: 'Python', level: 'Hate' }
]

d:\University\NodeJS\Lab1>

```