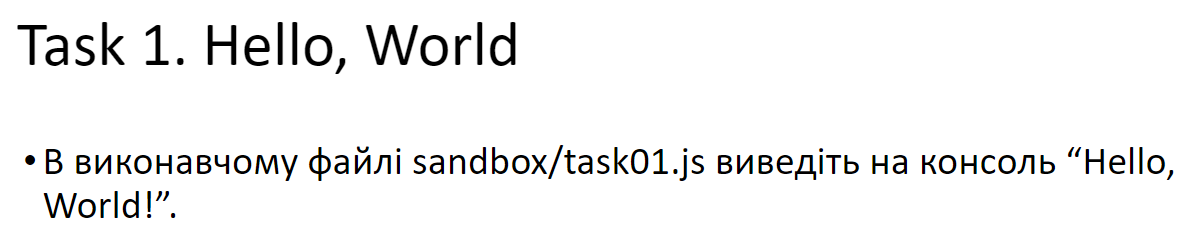
Node.js

Лабораторна робота №1

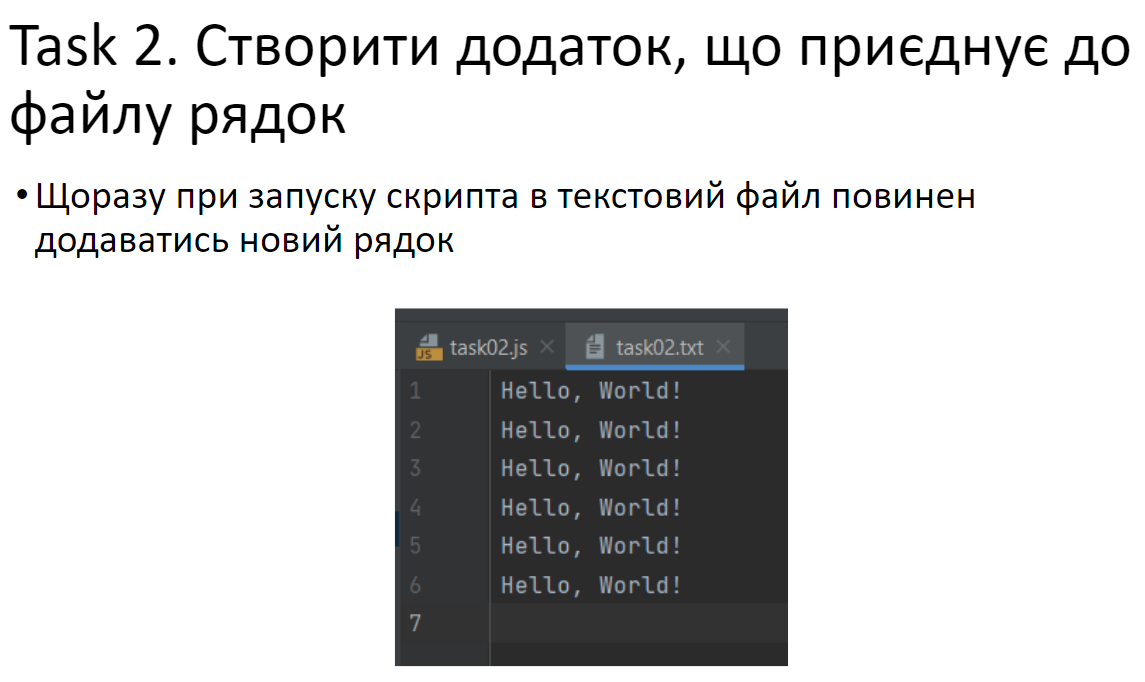
Робота з файлами, JSON, об’єктами, модулями, npm, lodash, nodemon, yargs



Лістинг програми:  
console.log("Hello, World!");

Результат виконання програми:





Лістинг програми:  
const fs = require('fs');

let newText = 'Hello, World!';

let previousContent = '';

try {

previousContent = fs.readFileSync('task02.txt', 'utf-8');

} catch (err) {

}

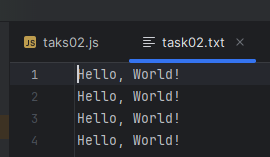
let newContent = previousContent + '\n' + newText;

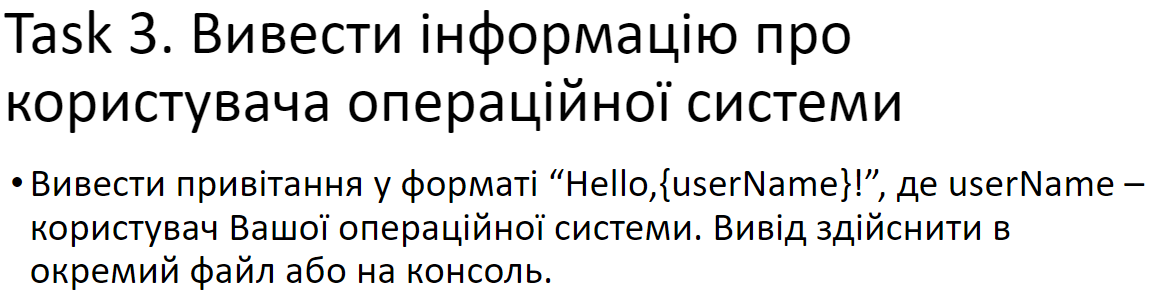
fs.writeFileSync('task02.txt', newContent, 'utf-8');

console.log('Створенно новий рядок:', newText);

Результат виконання програми:







Лістинг програми:

const os = require('os');

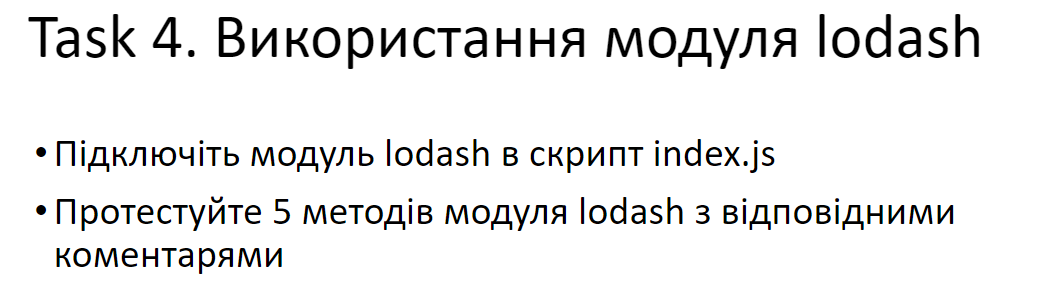
let userName = os.userInfo().username;

let greeting = `Hello, ${userName}!`;

console.log(greeting);

Результат виконання програми:





Лістинг програми:  
let \_ = require('lodash');

let isEmpty = \_.isEmpty([]);

console.log('Є пустим:', isEmpty);

let capitalizedString = \_.capitalize('hello');

console.log('Рядок з великої літери:', capitalizedString);

let rangeArray = \_.range(1, 6);

console.log('Масив в діапазоні [1;6]:', rangeArray);

let originalArray = [1, 2, 3, 4, 5];

let shuffledArray = \_.shuffle(originalArray);

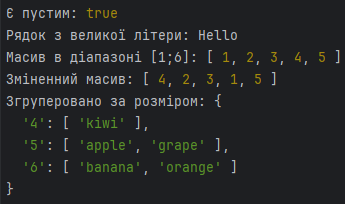
console.log('Зміненний масив:', shuffledArray);

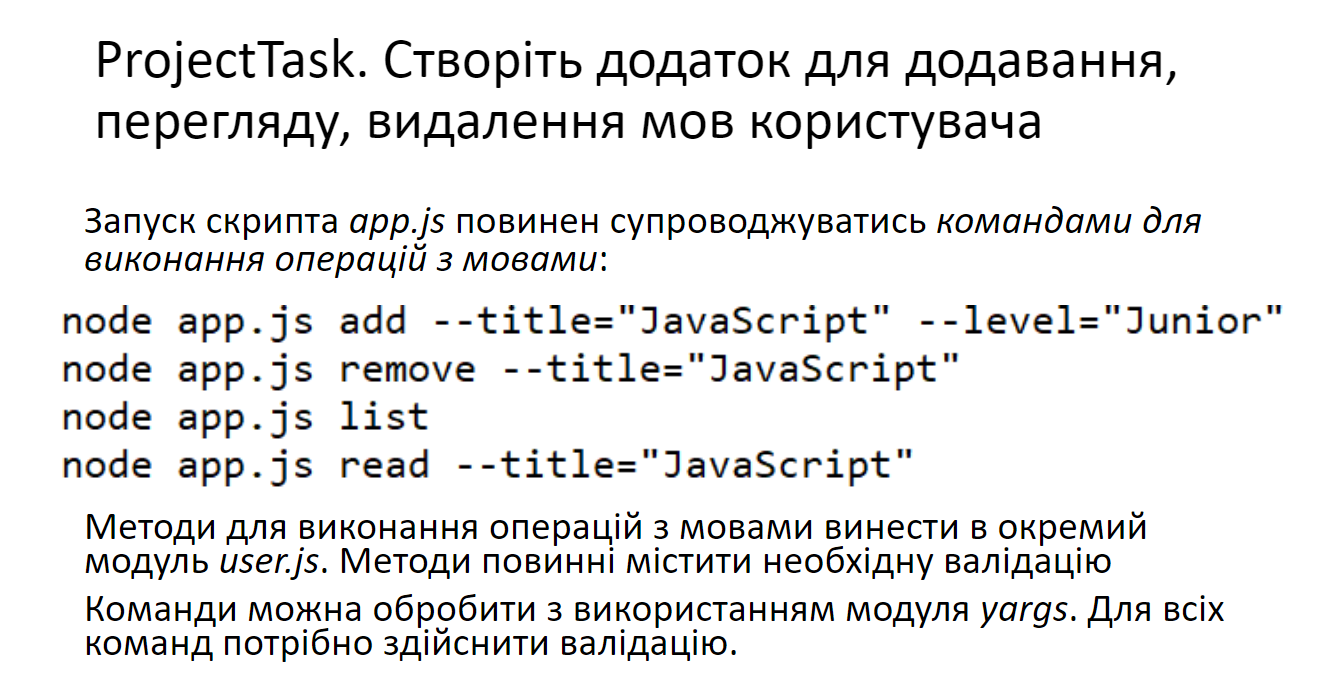
let fruits = ['apple', 'banana', 'orange', 'kiwi', 'grape'];

let groupedByLength = \_.groupBy(fruits, 'length');

console.log('Згруперовано за розміром:', groupedByLength);

Результат виконання програми:





Лістинг програми(User.js):

const fs = require('fs');

const getUserData = () => {

try {

let data = fs.readFileSync('user.json', 'utf-8');

return JSON.parse(data);

} catch (error) {

return { firstname: '', lastname: '', languages: [] };

}

};

const saveUserData = (userData) => {

fs.writeFileSync('user.json', JSON.stringify(userData, null, 2), 'utf-8');

};

const addUser = (title, level) => {

const userData = getUserData();

userData.languages.push({ title, level });

saveUserData(userData);

console.log('Language added successfully.');

};

const removeUser = (title) => {

let userData = getUserData();

let index = userData.languages.findIndex(lang => lang.title === title);

if (index !== -1) {

userData.languages.splice(index, 1);

saveUserData(userData);

console.log('Language removed successfully.');

} else {

console.log('Language not found.');

}

};

const listUsers = () => {

let userData = getUserData();

console.log('User Data:', userData);

};

const readUser = (title) => {

let userData = getUserData();

let language = userData.languages.find(lang => lang.title === title);

if (language) {

console.log('Language Details:', language);

} else {

console.log('Language not found.');

}

};

module.exports = { addUser, removeUser, listUsers, readUser };

Лістинг програми(app.js):

const yargs = require('yargs');

const userModule = require('./user');

yargs

.command({

command: 'add',

describe: 'Add a new programming language',

builder: {

title: {

describe: 'Language title',

demandOption: true,

type: 'string',

},

level: {

describe: 'Language level',

demandOption: true,

type: 'string',

},

},

handler: (argv) => {

userModule.addUser(argv.title, argv.level);

},

})

.command({

command: 'remove',

describe: 'Remove a programming language',

builder: {

title: {

describe: 'Language title',

demandOption: true,

type: 'string',

},

},

handler: (argv) => {

userModule.removeUser(argv.title);

},

})

.command({

command: 'list',

describe: 'List all programming languages',

handler: () => {

userModule.listUsers();

},

})

.command({

command: 'read',

describe: 'Read details about a programming language',

builder: {

title: {

describe: 'Language title',

demandOption: true,

type: 'string',

},

},

handler: (argv) => {

userModule.readUser(argv.title);

},

})

.parse();

Результат виконання програми:

