

# Node.js

Лабораторна робота №1

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

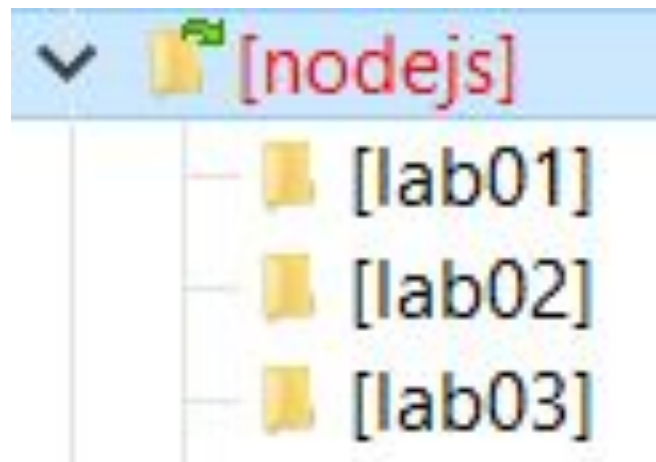
# Встановлення Node.js

- Завантажте та встановіть дистрибутив Node.js із сайту <https://nodejs.org/>
- Рекомендується встановлювати LTS-версію

# Засоби розробки

- Середовище JetBrains WebStorm

# Структура папок з лабораторними роботами:

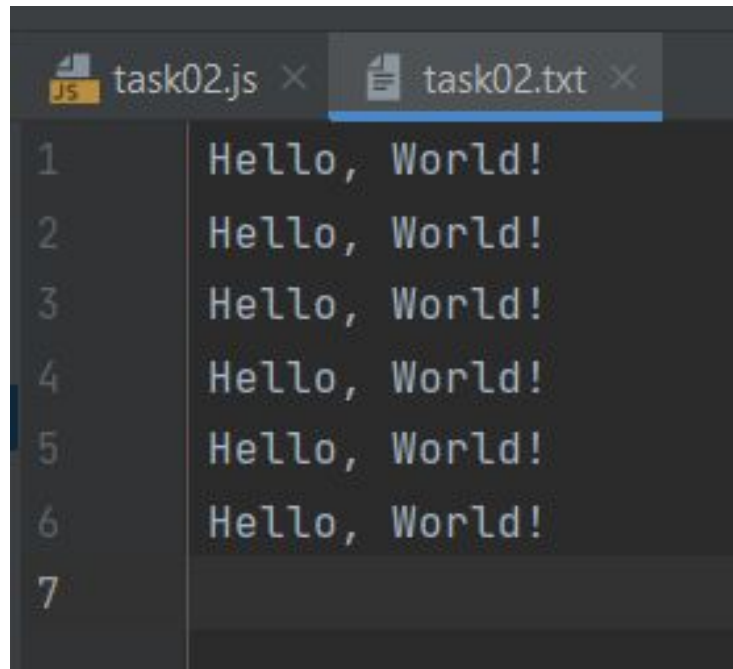


# Task 1. Hello, World

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

## Task 2. Створити додаток, що приєднує до файлу рядок

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



```
task02.js × task02.txt ×  
1 Hello, World!  
2 Hello, World!  
3 Hello, World!  
4 Hello, World!  
5 Hello, World!  
6 Hello, World!  
7
```

## 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.

# Ініціалізація проекту, package.json

- В проекті даної лабораторної роботи здійсніть ініціалізацію пакетного менеджера: `npm init`
- Виконайте кроки по заповненню інформації про додаток
- Переконайтесь в наявності файлу `package.json`



# Модуль lodash через npm

- Встановіть модуль lodash із додаванням відповідної інформації в package.json
- Інформацію по встановленню модулів можна знайти на сайті [npmjs.com](https://npmjs.com)
- Після встановлення в каталозі проекту повинна з'явитись папка node-modules
- Перевірте наявність інформації про встановлений модуль в файлі package.json
- При копіюванні файлів проекту (наприклад, на репозиторій), папку node-modules копіювати НЕ ПОТРІБНО, оскільки вся інформація про проект міститься в package.json

## Task 4. Використання модуля lodash

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

# Передача параметрів в консоль

- Вводити дані можна через консоль при запуску додатку
- Ці дані записуються в `process.argv`

Створіть файл *user.json* з даними імені користувача та мовами, якими він володіє



The image shows a code editor window with a tab labeled 'user.json'. The editor contains the following JSON code:

```
1 {  
2     "firstName": "Jane",  
3     "lastName": "Smith",  
4     "languages": [  
5         {"title": "JS", "level": "Pro"},  
6         {"title": "NodeJS", "level": "Beginner"}  
7     ]  
8 }
```

The code is syntax-highlighted: property names are in purple, string values are in green, and array elements are in blue. The editor has a line number margin on the left and a gutter with fold icons. The last line of the file is highlighted in yellow.

# 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.

# Файлова структура лабораторної роботи, необхідна для задачі

- Тренувальні вправи даної роботи перенести в папку playground
- На репозиторій папку node\_modules не комітити

