# Lab1

Saturday, March 13, 2021 4:06 PM

1 zoom call for each slot (3 slots) with different teacher. Several rooms in each slot, one per group.
Global chat, teacher will come. Mapping of groups to slots : choose by myself

# 01UDFOV/01TXYOV – WEB APPLICATIONS I

## GETTING STARTED WITH NODE.JS

During this first lab, you will start to become acquainted with JavaScript (in Node.js) by putting in practice what you saw in the lectures of the first two weeks.

## EXERCISE 0 - PREPARATION

First of all, check that your *Node.js* installation is working within Visual Studio Code. Create a function that, given an array of strings, for each string computes a new string made of the first two and the last two characters of the original string. The new string should replace the old one in the same array.

e.g., '*spring*' yields '*spng*'

If the word is shorter than two characters, return the empty string.

## EXERCISE 1 – FUNCTIONAL PROGRAMMING

Implement a program to manage a series of tasks (i.e., actions that the user wants to do in the future). In particular, a task is made of the following fields:

- a unique numerical *id* (required);
- a textual *description* (required);
- whether it is *urgent* (default: false);
- whether it is *private* (default: true);
- a *deadline* (i.e., a date with or without a time. This field is optional).

By means of *constructor functions*, create some Task objects and add them to a TaskList object, i.e., an object that is able to store a list of tasks (as an array, internally). Then, implement the following methods:

- *sortAndPrint*: the method should sort the content of the TaskList by deadline, in ascending order (the tasks without a deadline should be listed at the end). After sorting, the method should print the content of the TaskList;

```
****** Tasks sorted by deadline (most recent first): ******
Id: 3, Description: phone call, Urgent: true, Private: false, Deadline: March 8,
2021 4:20 PM
Id: 2, Description: monday lab, Urgent: false, Private: false, Deadline: March 16,
2021 10:00 AM
Id: 1, Description: laundry, Urgent: false, Private: true, Deadline: <not defined>
```

- *filterAndPrint:* starting from the entire list of tasks, the method should filter out the tasks that are not urgent. After filtering, the method should print the content of the filtered TaskList (without any particular order).

```
****** Tasks filtered, only (urgent == true): ******
Id: 3, Description: phone call, Urgent: true, Private: false, Deadline: March 8,
2021 4:20 PM
```

***Hints***
*To implement the described functionality, you can manipulate the array of tasks by using the JavaScript functional programming paradigm.*

## EXERCISE 2 – DATABASE INTERACTION

Extend the program developed in Exercise 1 to use a database. Consider the database "tasks.db", that contains a collection of tasks stored in the same format described in Exercise 1. The program should:

- load all the tasks included in the database into a TaskList and print them;
- load and print, through a parametric query, a TaskList containing only the list of tasks whose deadline is after a given date;
- load and print, through a parametric query, a TaskList containing only the list of tasks that contain a given word.

***Hints***
*The file "tasks.db" is included in the repository available on GitHub: https://github.com/polito-WA1-AW1-2021/lab1-node.git*

*As you saw in the lectures, you can connect to an SQLite database with one of the following modules:*

1. *sqlite3 (https://www.npmjs.com/package/sqlite3) – the basic library*
2. *sqlite (https://www.npmjs.com/package/sqlite) – a Promise-based API to sqlite3*