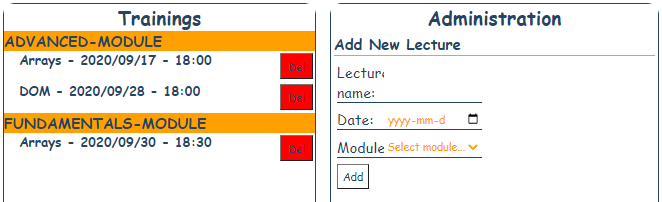
# Problem 01. SoftUni Administration

Exam problems for the [“JavaScript Advanced” course @ SoftUni](https://softuni.bg/courses/javascript-advanced). Submit your solutions in the SoftUni Judge system at <https://judge.softuni.bg/Contests/Compete/Index/2590#0>

### Use the given skeleton to solve this problem.

**Note: USE THE GOOGLE CHROME BROWSER!**

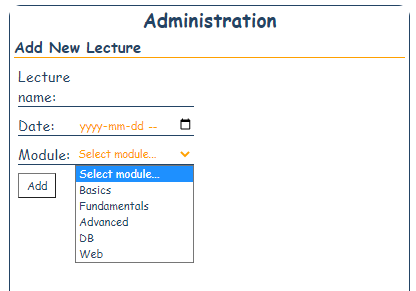
### Note: You have NO permission to change directly the given HTML *(index.html file)*.



### Your Task

**Write the missing JavaScript code** to make the **SoftUni Administration Functionality** works as follows:

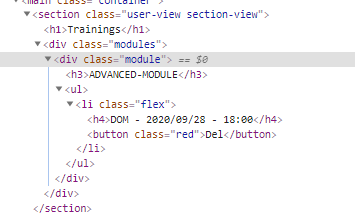
The **Module field** is a dropdown menu.



When the **Add** button is **clicked**, first you need to **validate the inputs**. If any of the input fields is **invalid (empty or default value)**, the function **doesn’t** make anything. Be careful with the **validation of the Module dropdown** (it's default value is **NOT** an empty string).

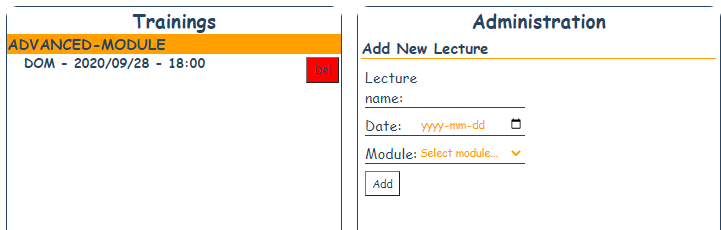
**After validating** the input fields, you need to **add** the new **lecture** in the **Trainings** section.

The HTML structure after adding new lecture looks like this:



The added lecture should have **button** with text "**Del**". **The date** and the **lecture name** must be **formatted** as shown on the screenshots. Be careful with the **HTML-Structure** and with the **class names**!

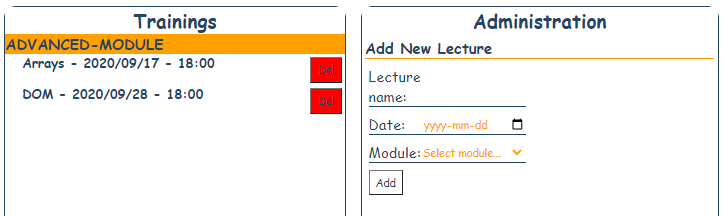
Duplicated names of the lectures are allowed.



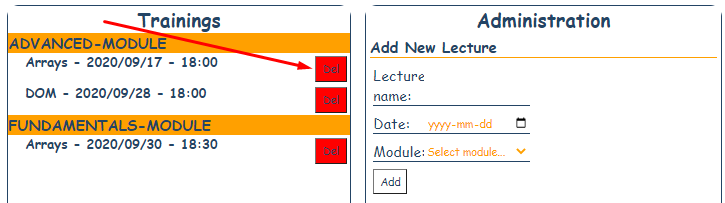
### If there is another Lecture with the same Module:

The lectures should be **sorted by DATE**! The date is separated with **slashes** (/) and the **separator** between the **date** and **time** is a **minus sign.**

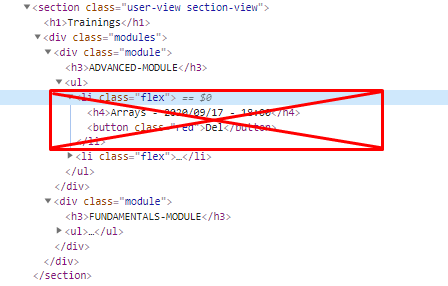
**Hint: The format of the date allows alphabetical sort.**



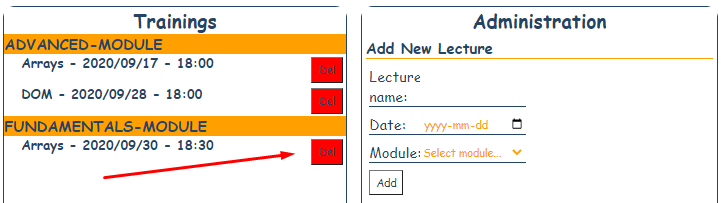
### Deleting Lecture/Module:

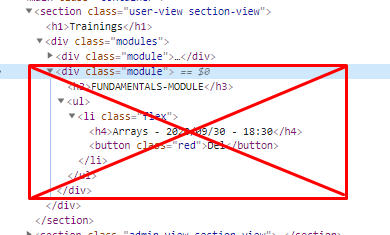


When the “**Delete**” button is clicked, the **Lecture (whole list element)** should be **removed** from the HTML.



If there are **NO** Lectures left in the module (after delete) the whole **module** should be **also deleted**.





function solve() {

    const formControls = document.querySelectorAll('.form-control input');

    const lecture = formControls[0];

    const date = formControls[1];

    const moduleName = document.querySelector('select');

    const modulesOutput = document.querySelector('.modules');

    let state = {};

    function createElement(type, text, attributes = []) {

        let element = document.createElement(type);

        if (text) {

            element.textContent = text;

        }

        attributes

            .map(attr => attr.split('='))

            .forEach(([name, value]) => {

                element.setAttribute(name, value);

            })

        return element;

    }

    function add(e) {

        e.preventDefault();

        // Правя валидации

        if (lecture.value === '' || date.value === '' || moduleName.value.includes('Select module')) {

            return;

        }

        //Създавам си ли-то

        let lectureName = lecture.value;

        let dateString = date.value;

        const lectureTitle = lectureName + ' - ' + dateString.split('-').join('/').split('T').join(' - ');

        const delBtn = createElement('button', 'Del', ['class=red']);

        const lectureH4 = createElement('h4', lectureTitle);

        const li = createElement('li', undefined, ['class=flex']);

        li.appendChild(lectureH4);

        li.appendChild(delBtn);

        //Създавам си модула

        let module = undefined;

        let ul = undefined;

        //Ако го създавам за първи път...

        if (!state[moduleName.value]) {

            let h3 = createElement('h3', `${moduleName.value.toUpperCase()}-MODULE`);

            ul = createElement('ul');

            lis = [];

            module = createElement('div', undefined, ['class=module']);

            module.appendChild(h3);

            module.appendChild(ul);

            state[moduleName.value] = { module, ul, lis: [] };

        } else { // Ако вече имам такъв модул

            module = state[moduleName.value].module;

            ul = state[moduleName.value].ul;

        }

        //Слагам си ли-то в литата на модула

        state[moduleName.value].lis.push({ li, date: date.value });

        //Сортирам ли-тата и апендвам към ul-a

        state[moduleName.value].lis.sort((a, b) => {

            return a.date.localeCompare(b.date)

        }).forEach(({ li }) => {

            ul.appendChild(li);

        })

        modulesOutput.appendChild(module);

    }

    //Tриене

    function del(e) {

        let li = e.target.parentNode;

        let ul = li.parentNode;

        let module = ul.parentNode;

        //Изтривам ли-то

        li.remove();

        //Ако ул-а е празен трия целия модул

        if (ul.children.length === 0) {

            module.remove();

        }

    }

    // //Взимам си стейта и апендвам всички модули в .modules

    // function render() {

    //     Object.entries(state).forEach(([name, module]) => {

    //         modulesOutput.appendChild(module.module)

    //     })

    // }

    //Закачам евентлисънър на мейн метода

    document.getElementsByTagName('main')[0].addEventListener('click', (e) => {

        //ако таргета е бутон само тогава правя нещо

        if (e.target.tagName === 'BUTTON') {

            //Ако не съдържа клас red добавям

            if (!e.target.classList.contains('red')) {

                add(e);

                // render()

            } else {  //Иначе трия

                del(e);

            }

        }

    })

};