# DOM: Exercises

Problems for exercises and homework for the ["Free JS for Front-End Course @ SoftUni"](https://softuni.bg/trainings/2946/js-for-front-end-march-2020). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/Practice/Index/2237#0>

## Subtraction

A HTML page holds **two text fields** with **id=**"firstNumber" and **id=**"secondNumber". Write a function to **subtract** the values from these text fields and display the result in a div with **id=**"result".

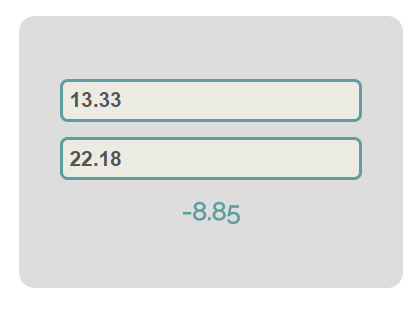
### HTML, CSS and JavaScript Code

Implement the aboveto provide the following functionality:

* Your function should take the **values** of the input fields, **convert** them to numbers, **subtract** the second number from the first and then write the result in the **<div>** with **id="result"**
* Your function should be able to work with **any 2 numbers** in the inputs, not only the ones given in the example.

Submit in the judge the code (implementation) of the above function. It may hold other functions in its body.

### Example



### Hints

We see that the **textboxes** and **div** have **id** attributes on them.

|  |
| --- |
| index.html |
| **<input type="text" id="firstNumber" value="13.33" disabled=""></input>**  **<input type="text" id="secondNumber" value="22.18" disabled=""></input>**  **<div id="result"></div>** |

We can take the numbers directly from the input field by using the **getElementById()** function. After we have taken the elements from the DOM it’s time to do the actual work. We get the values of the two **textboxes**, the value of a textbox as one would expect is **text**, in order to get a **number** we need to use a function to **parse** **them**.

All that’s left now is to write the result in the div. We use the same function to get the **result** element by **id** and change it’s **text content** to the resulting **subtraction.**

|  |
| --- |
| subtract.js |
| **function subtract() {**  **let num1 = Number(document.getElementById("firstNumber").value);**  **let num2 = Number(document.getElementById("secondNumber").value);**  **document.getElementById("result").textContent = num1 - num2;**  **}** |

Our code is ready for submitting now, paste the contents of the **.js** file in the judge.

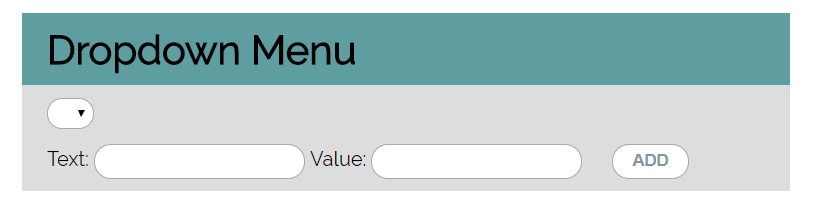
## Fill Dropdown

Your task is to take values from **input** fields with **id=**"**newItemText**" and **id=**"**newItemValue**"and create and append an **<option>** to the **<select>** with **id**="**menu**"**.**

Again you should create a separate file called **dropdown.js.** In it you should have the following function

|  |
| --- |
| dropdown.js |
| **function fillDropdown() {**  **// TODO**  **}** |

### Example



### Hints

* Your function should take the values of **newItemText** and **newItemValue**. After that you should create a new **option** element and set it’s **textContent** and it’s **value** to the newly taken ones.
* Once you have done all of that you should **append** the newly created **option** as a **child** to the select item with **id=**"**menu**"**.**
* Finally you should **clear** the value of the **input** fields.

## Accordion

An **HTML** file is given and your task is to show **more**/ **less** information by clicking a **button** (it is not an actual button, but a **span** that has an **onlick** event attached to it). When **[More]** link is clicked, it **reveals** the contents of a **hidden** div and change the text of the link to **[Less]**. When the same link is clicked **again** (now reading Less), **hide** the div and **change** the text of the link back. Link action should be **toggleable** (you should be able to click the button infinite amount of times).

### Example





### Hints

* To **change** the text content of a button you could use **getElementsByClassName()**. Which however returns a **collection** and we need only **one** element from it so the correct way is to **use** it like this: **getElementsByClassName(`button`)[0]** and it will return the needed span element.
* After that we should change the **display style** of the div **with id=**"**extra**". If the display style is "**none**" we should **change** it to "**block**" and the **opposite**.
* Alongside all of this we should **change** the text content of the **button** to Less/More.

## Order the Names

Write afunctionthat **orders** names **alphabetically**.



You will receive a **name of a student as an input**. When the [ADD] button is **clicked**, you should

add the given student name in the SoftUni Database, while **keeping** the **alphabetial order**.

For instance, if we register **David,** his name should appear in the **D** column.





If you **receive more than one name with the same starting letter**, you should **join all names** by a

comma and a space (", ").



## Chat Room

Write afunction to create the functionality of a **chat room.**



**Note:** Do not forget to **add event listener** to the **send button!**

The newdiv **element with class=**"**message my-message**"should contain the current message that is about to be send.

The **current** div should be appended to the div with an id="chat\_messages".

**The input should be cleared on each click of the send button**.





## \*Distance Converter

Your task is to convert from **one** distance unit to **another** by adding a **click** event listener to a button. When it is clicked, **read** the value in the input field, **get** the selected option from the input and output units drop downs and **calculate** and **display** the converted value in the disabled output field.

You should have the following **distanceConverter.js** file:

|  |
| --- |
| distanceConverter.js |
| **function attachEventsListeners() {**  **// TODO: Аttach click event to convert button**  **}** |

Multiply the incoming distance by the following conversion rates to convert to meters. Divide to convert from meters to the required output unit.

1 km = 1000 m

1 m = 1 m

1 cm = 0.01 m

1 mm = 0.001 m

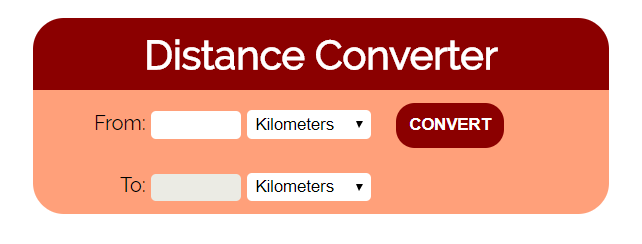
1 mi = 1609.34 m

1 yrd = 0.9144 m

1 ft = 0.3048 m

1 in = 0.0254 m

### Example



### Hint

To see which option is selected, read the properties of its parent: value gives you the value of the selected option (as displayed in the HTML), selectedIndex gives you the 0-based index of the selected option. E.g. if miles are selected, #inputUnits.value is "**mi**", #inputUnits.selectedIndex is **4**. Option text is irrelevant.