**Exercise: DOM Manipulations**

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/2239#0>

## Increment Counter

You are tasked with creating a piece of **HTML** dynamically using JavaScript and **appending** it to a given element using a passed in **selector**.

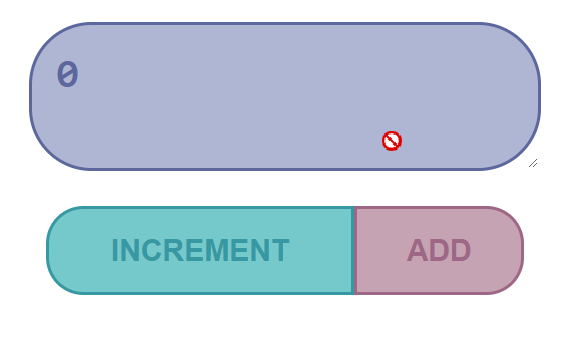
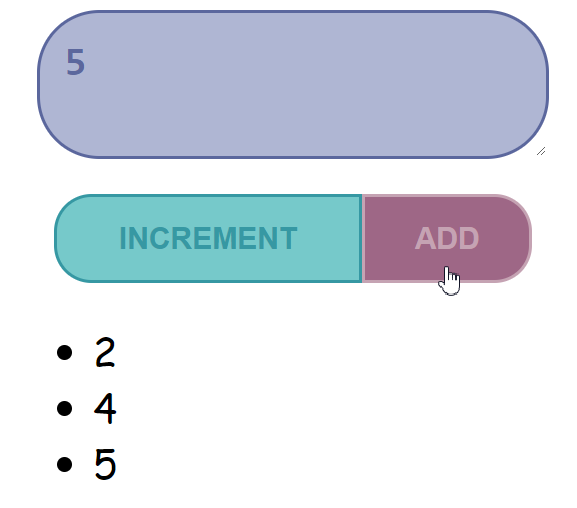
Your function will receive a **string** value representing a **selector** (for example "#wrapper" or ".root"), all elements created should be appended to the **selector**.

The HTML you create should contain 4 elements:

* <textarea> with class="counter", value="0" and the disabled attribute.
* <button> with class="btn", id="incrementBtn" and text "**Increment**".
* <button> with class="btn", id="addBtn" and text "**Add**".
* Unordered list <ul> with class="results".

When the [Increment] is clicked the value of the **textarea** should go up by **one** (if it was 0 it should become 1 e.t.c.). When the [Add] is clicked a new list item (<li>) with text equal to the current value of the textarea should be added to the unordered list.

### Screenshots



### Hints

We’ll start off by creating the needed elements and parsing the **selector**. Adding multiple elements to the DOM can be expensive, instead of repeatedly adding to the DOM we can create a DocumentFragment and **add** the elements to it instead. When we have built our hierarchy we can **append the** DocumentFragment to the DOM, which will add all of the fragment’s elements to the specified selector.  
  
The next have to **add values**, and **attributes** to the **elements** and **events** to the **buttons**:

|  |
| --- |
| incrementCounter.js |
| function increment(selector) {  const container = document.querySelector(selector);    const fragment = document.createDocumentFragment();    const textArea = document.createElement('textarea');    textArea.value = 0;    textArea.className = 'counter';    textArea.setAttribute('disabled', true);    let list = document.createElement('ul');    list.className = 'results';    const incrementBtn = document.createElement('button');    incrementBtn.textContent = 'Increment';    incrementBtn.className = 'btn';    incrementBtn.setAttribute('id', 'incrementBtn');    const addBtn  = document.createElement('button');    addBtn.textContent = 'Add';    addBtn.className = 'btn';    addBtn.setAttribute('id', 'addBtn');    // add our elements to the DOM:    fragment.appendChild(textArea);    fragment.appendChild(incrementBtn);    fragment.appendChild(addBtn);    fragment.appendChild(list);    container.appendChild(fragment);  ...} |

The last step is to **addEventListener()** to created buttons.

|  |
| --- |
| incrementCounter.js |
| ...  document.getElementById('incrementBtn').addEventListener('click', plusOne);    document.getElementById('addBtn').addEventListener('click', listIncr);    function plusOne() {      textArea.value = +textArea.value + 1;    }    function listIncr() {      const li = document.createElement('li');      li.textContent = textArea.value;      list.appendChild(li);    } |

## Sections

You will receive an **array** of strings. For each string, create a **div** with a **paragraph** with the **string** in it. Each paragraph is initially **hidden ("display:none")**. Add a **click** **event** **listener** to **each div** that **displays** the **hidden** paragraph. Finally, you should **append** all divs to the element with an **id="content"**.

**Example**

 🡪 

## JavaScript Quizz

Write a function that has the functionality of a quiz.



There are three sections that contain 1 **question** and **2** possible **answers.**

**The right answer is only one!**

When one of the **list elements** is **clicked,** the next section **must appear** (if any…).

After all three questions have been answered, the result div must **appear.** (Use **'none'** and **'block**' to hide and show the question sections).

If all questions are answered correctly, you should print the following message:   
"You are recognized as top JavaScript fan!"

Otherwise, just print "You have {rightAnswers} right answers".

The right answers are (onclick**,** JSON.stringify()andA programming API for HTML and XML documents).











## \* Numpad Calculator

Write a function that implements a calculator that has the following functionalities:



When one of **the buttons is clicked**, its value should be shown in the "Expression" **field** (#expressionOutput).

For instance, if we click on the button with value 9, the expected result should be:



If the **current еxpression** field (#expresisonOutput) contains the whole math expression (**left operand**, **operator** and **right operand**: **Example: 9 + 2**), the expected result should be:



When the **equal sign "**=**" is pressed,** the **result** of that **expression** should appear in the Result field (#resultOutput)



Otherwise, if we create an invalid expression such as "99 +" (**without second/right operand**) and we hit the equal sign "=", the expected result should be:



The "Clear" button should **clear** bothExpressionandResult **fields (**#expressionOutputand#resultOutput**)**

For instance, if we have the following expression:



And we press "Clear", the expected result should be:

