**Lab: DOM Introduction**

Problems for in-class lab for the [" HYPERLINK "https://softuni.bg/courses/js-advanced"JavaScript HYPERLINK "https://softuni.bg/courses/js-advanced"Advanced" course @ HYPERLINK "https://softuni.bg/courses/js-advanced"SoftUni](https://softuni.bg/courses/js-advanced). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/2760/DOM-Introduction-Lab>

**Environment Specifics**

Please, be aware that every JS environment may **behave differently** when executing code. Certain things that work in the browser are not supported in **Node.js**, which is the environment used by **Judge**.

The following actions are **NOT** supported:

* **.forEach()** with **NodeList** (returned by **querySelector()** and **querySelectorAll()**)
* **.forEach()** with **HTMLCollection** (returned by **getElementsByClassName()** and **element.children**)
* Using the **spread-operator** (**...**) to convert a **NodeList** into an array
* **append()** in Judge (use only **appendChild()**)
* **prepend()**
* Always turn the collection into an **JS array** (forEach, forOf, et.)

If you want to perform these operations, you may use **Array.from()** to first convert the collection into an array.

* **Edit Element**

Create function **edit()** that takes **three** parameters.

**Input/Output**

**First** parameter is a **reference** to an **HTML** element, other two parameters are string – **match** and **replacer.**

You have to **replace** all occurrences of **match** inside the **text content** of the given elements with **replacer.**

* **Collect List Items**

Write a JS function that scans a given **HTML list** and **appends** all collected list items’ text to a **text area** on the same page when the user **clicks** on a button.

**Input/Output**

There will be no input/output, your program should instead **modify** the DOM of the given HTML document.

**Examples**

 

* **Sum Numbers**

Write a JS function that **reads** two numbers from input fields in a **web page** and puts their **sum in another field** when the user **clicks** on a button.

**Input/Output**

There will be no input/output, your program should instead **modify** the DOM of the given HTML document.

**Examples**



* **Show More**

Write a JS function that **expands** a hidden section of text when a link is **clicked**. The link should **disappear** as the rest of the text shows up.

**Input/Output**

There will be no input/output, your program should instead **modify** the DOM of the given HTML document.

**Examples**

 

* **Colorize Table**

Write a JS function that **changes the color** of all **even** rows when the user **clicks** a button. Apply the color "**Teal**" to the target rows.

**Input/Output**

There will be no input/output, your program should instead **modify** the DOM of the given HTML document.

**Examples**

 

* **Sum Table**

Write a JS function that finds the first table in a document and sums the values in the last column. The result is then displayed in an element with ID "sum".

**Input/Output**

There will be no input/output, your program should instead **modify** the DOM of the given HTML document.

**Examples**

 

* **Extract Parenthesis**

Write a JS function that when **executed**, extracts all parenthesized text from a target paragraph by given element ID. The result is a string, joined by "; " (semicolon, space).

**Input**

Your function will receive a **string parameter**, representing the target element ID, from which text must be extracted. The text should be extracted from the DOM.

**Output**

**Return a string** with all matched text, separated by "; " (semicolon, space).

**Examples**



|  |
| --- |
| **Sample call** |
| **let text = extract("content");** |
| **Result (stored in variable text)** |
| **Bulgaria; Kazanlak; Rosa demascena Mill;** |