Lab: Document Object Model

Problems for in-class lab for the "JavaScript Advanced" course @ SoftUni. Submit your solutions in the SoftUni judge system at https://judge.softuni.bg/Contests/328/.

1. 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.

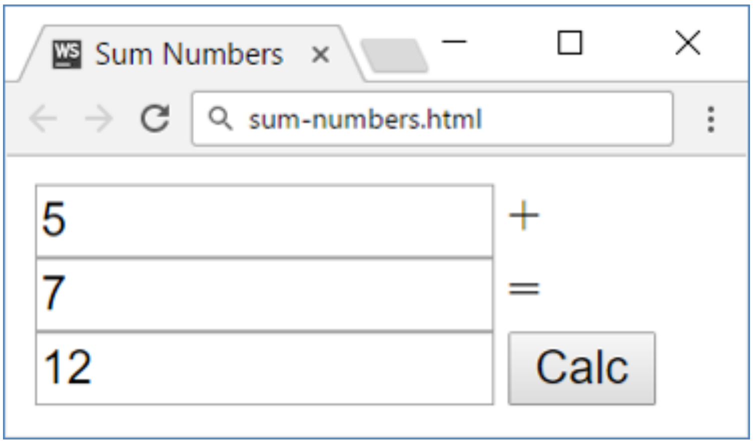
Input/Output

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

Sample HTML

```
<input type="text" id="num1" /> +
<input type="text" id="num2" /> =
<input type="text" id="sum" readonly="readonly" />
<input type="button" value="Calc" onclick="calc()" />
<script>
    function calc() {
       // TODO: sum = num1 + num2
    }
</script>
```

Examples



2. 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

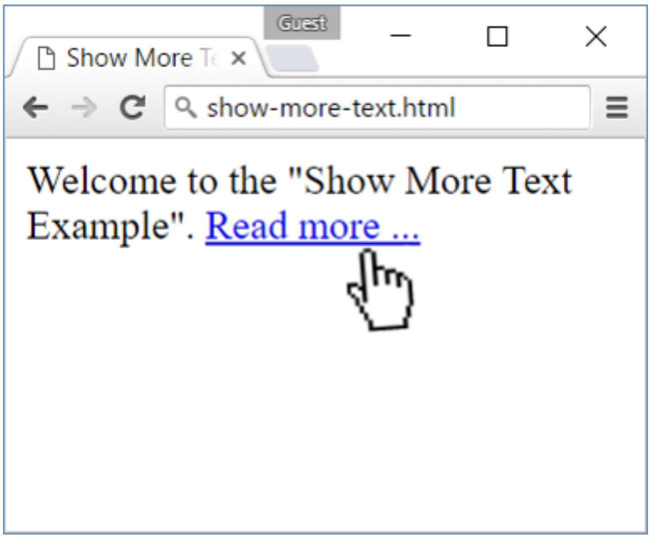
There will be no input/output, your program should instead **modify** the DOM of the given HTML document.

Sample HTML

```
Welcome to the "Show More Text Example".
<a href="#" id="more" onclick= "showText()">Read more …</a>
```

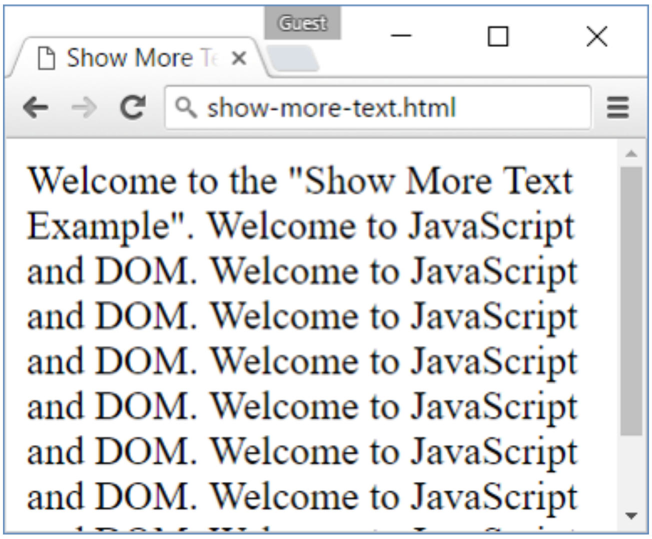
```
<span id="text" style= "display:none">Welcome to JavaScript and DOM.</span>
<script>
  function showText() {
    // TODO
  }
</script>
```

Examples



Welcome to the "Show More Text Example". Read more ...

→



Welcome to the "Show More Text Example". Welcome to JavaScript and DOM. Welcome to JavaScript and DOM. Welcome to JavaScript and DOM. Welcome to JavaScript and DOM. Welcome to JavaScript and DOM. Welcome to JavaScript and DOM. Welcome to JavaScript

3. 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.

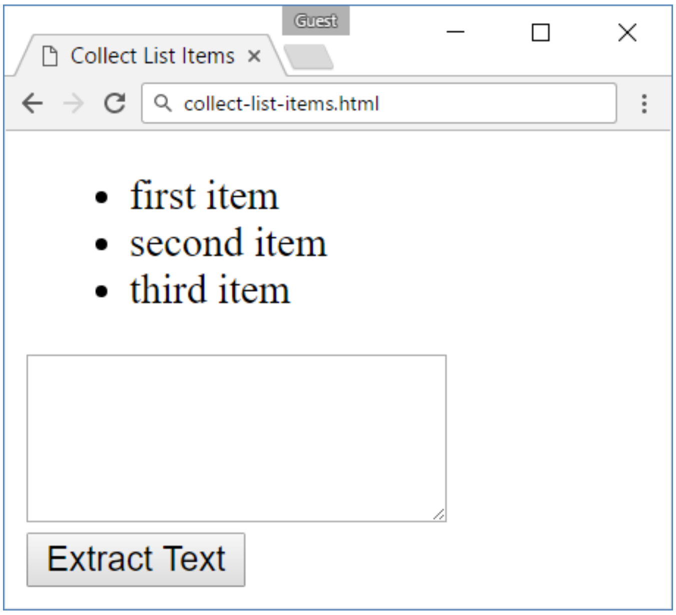
Input/Output

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

Sample HTML

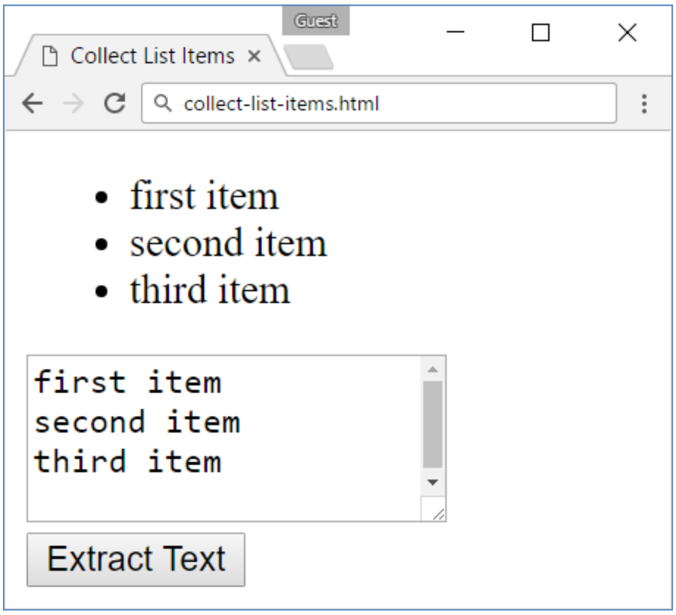
```
<ul id="items">
  <li>first item</li>
  <li>second item</li>
  <li>third item</li>
</ul>
<textarea id="result"></textarea>
<br>
<button onclick="extractText()">Extract Text</button>
<script>
  function extractText() {
    // TODO
  }
</script>
```

Examples



- first item
- second item
- third item

→



- first item
- second item
- third item

```
first item
second item
third item
```

4. 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.

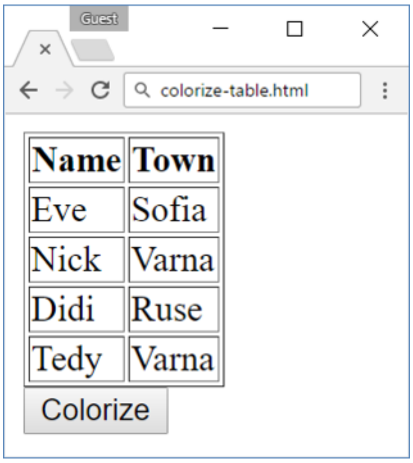
Input/Output

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

Sample HTML

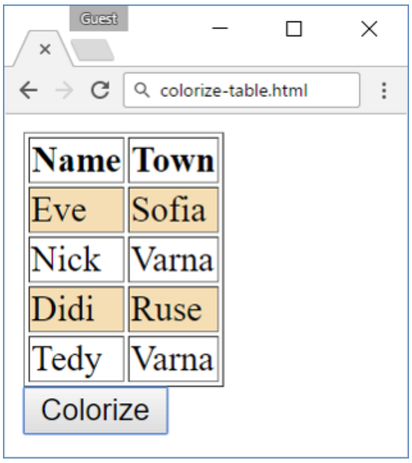
```
<table>
  <tr><th>Name</th><th>Town</th></tr>
  <tr><td>Eve</td><td>Sofia</td></tr>
  <tr><td>Nick</td><td>Varna</td></tr>
  <tr><td>Didi</td><td>Ruse</td></tr>
  <tr><td>Tedy</td><td>Varna</td></tr>
</table>
<button onclick="colorize()">Colorize</button>
<script>
  function colorize() {
    // TODO
  }
</script>
```

Examples



Name	Town
Eve	Sofia
Nick	Varna
Didi	Ruse
Tedy	Varna

→



Name	Town
Eve	Sofia
Nick	Varna
Didi	Ruse
Tedy	Varna

5. Countdown

Write a JS program that **implements** a web countdown timer that supports **minutes** and **seconds**. The user should be able to set the time by calling you function with the number of seconds required. The time begins to count down as soon as the function is called. Using the sample code, your function will be called as soon as the page finishes loading and will begin to count down from 10 minutes.

Input

Your function will receive a **number parameter**, representing the starting number of **seconds**.

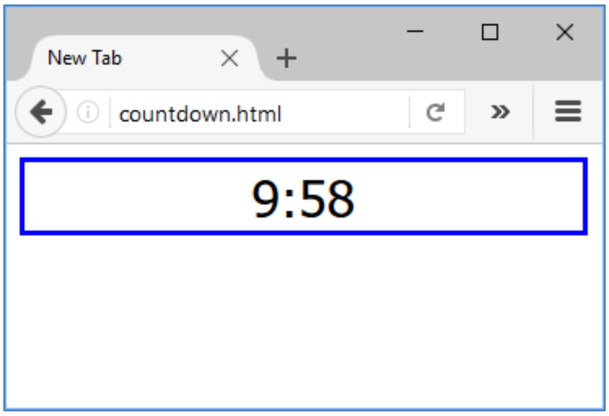
Output

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

Sample HTML

`<input type="text" id="time" style="border:3px solid blue; text-align:center; font-size:2em;" disabled="true"/>`
`<script>window.onload = function() { countdown(600); }</script>`

Examples



6. 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).

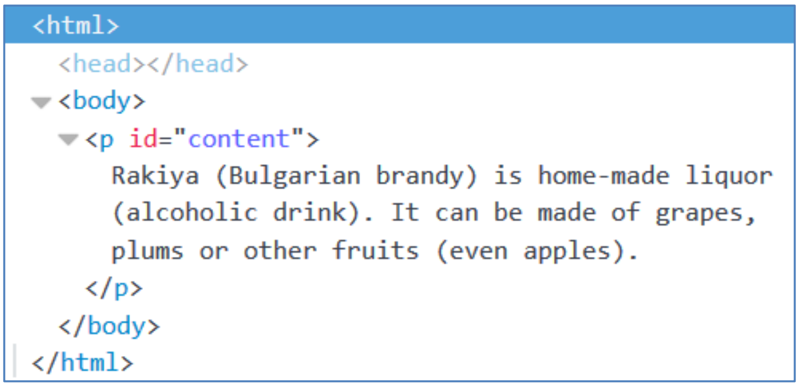
Input

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



```
<html>
  <head></head>
▼ <body>
  ▼ <p id="content">
      Rakiya (Bulgarian brandy) is home-made liquor
      (alcoholic drink). It can be made of grapes,
      plums or other fruits (even apples).
    </p>
  </body>
</html>
```

Sample call
`let text = extract("content");`
Result (stored in variable text)
`Bulgarian brandy; alcoholic drink; even apples`

7. * 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".

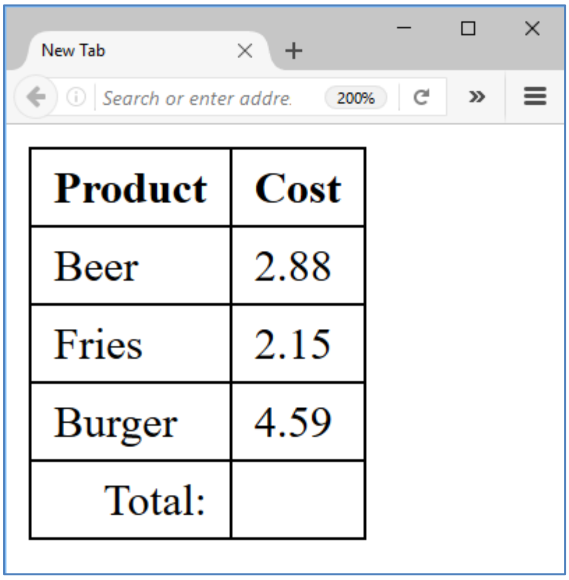
Input/Output

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

Sample HTML

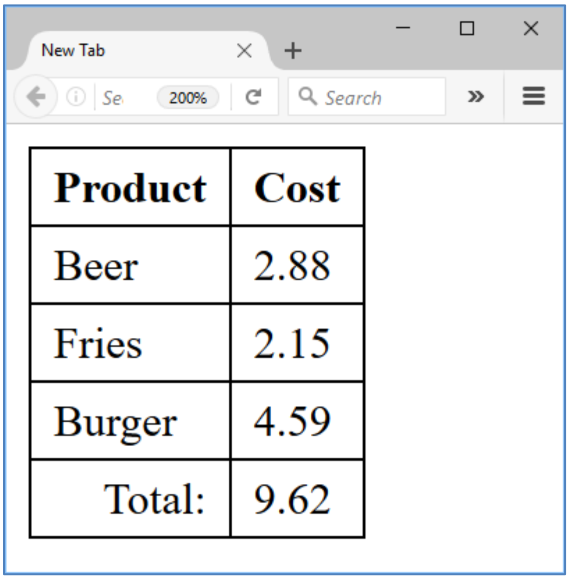
```
<table>
  <tbody>
    <tr><th>Product</th><th>Cost</th></tr>
    <tr><td>Beer</td>   <td>2.88</td></tr>
    <tr><td>Fries</td>  <td>2.15</td></tr>
    <tr><td>Burger</td> <td>4.59</td></tr>
    <tr><td>Total:</td> <td id="sum"></td></tr>
  </tbody>
</table>
<button onclick="sum()">Sum</button>
```

Example



Product	Cost
Beer	2.88
Fries	2.15
Burger	4.59
Total:	

→



Product	Cost
Beer	2.88
Fries	2.15
Burger	4.59
Total:	9.62