Exercises: DOM Manipulation

Problems for exercises and homework for the "JavaScript Advanced" course @ SoftUni. Submit your solutions in the SoftUni judge system at https://judge.softuni.bg/Contests/641/ .

1. Subtraction

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

HTML and JavaScript Code

You are given the following **HTML** code:

subtract.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Subtraction</title>
</head>
<body>
<div id="wrapper">
    <input type="text" id="firstNumber" value="13.33" disabled>
    <input type="text" id="secondNumber" value="22.18" disabled>

    <div id="result"></div>
</div>
<script src="subtract.js"></script>
<script>
    window.onload = function () {
        subtract();
    }
</script>
</body>
</html>
```

It comes together with the following **JavaScript** code:

subtract.js

```
function subtract() {
    // TODO
}
```

Implement the above to provide the following functionality:

- Your function should take the values of "**firstNumber**" and "**secondNumber**", **convert** them to numbers, **subtract** the second number from the first and then write the result in the ** ** 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 JS 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.

```
<input type="text" id="firstNumber" value="13.33" disabled>
<input type="text" id="secondNumber" value="22.18" disabled>
<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**.

```
let num1 = Number(document.getElementById("firstNumber").value);
let num2 = Number(document.getElementById("secondNumber").value);
```

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.**

```
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.

2. Fill Dropdown

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

HTML and JavaScript Code

You are given the following **HTML** code:

dropdown.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Fill Dropdown</title>
    <script src="dropdown.js"></script>
```

```
</head>
<body>
<h1>Dropdown Menu</h1>
<div>
    <select id="menu"></select>
</div>
  <label for="newItemText">Text: </label><input type="text" id="newItemText" />
  <label for="newItemValue">Value: </label><input type="text" id="newItemValue" />
  <input type="button" value="Add" onclick="addItem()">
</body>
</html>
```

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

dropdown.js

```
function addItem() {
    // 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 two **input** fields.

3. Accordion

An **html** file is given and your task is to show **more**/show **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).

HTML and JavaScript Code

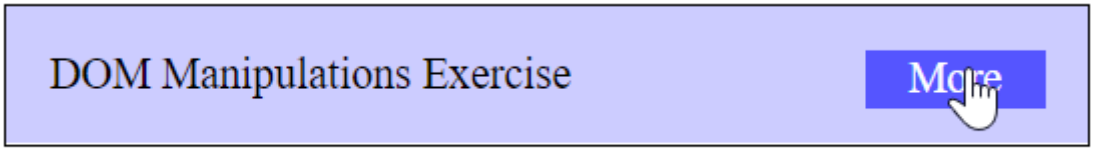
You are given the following **HTML** code:

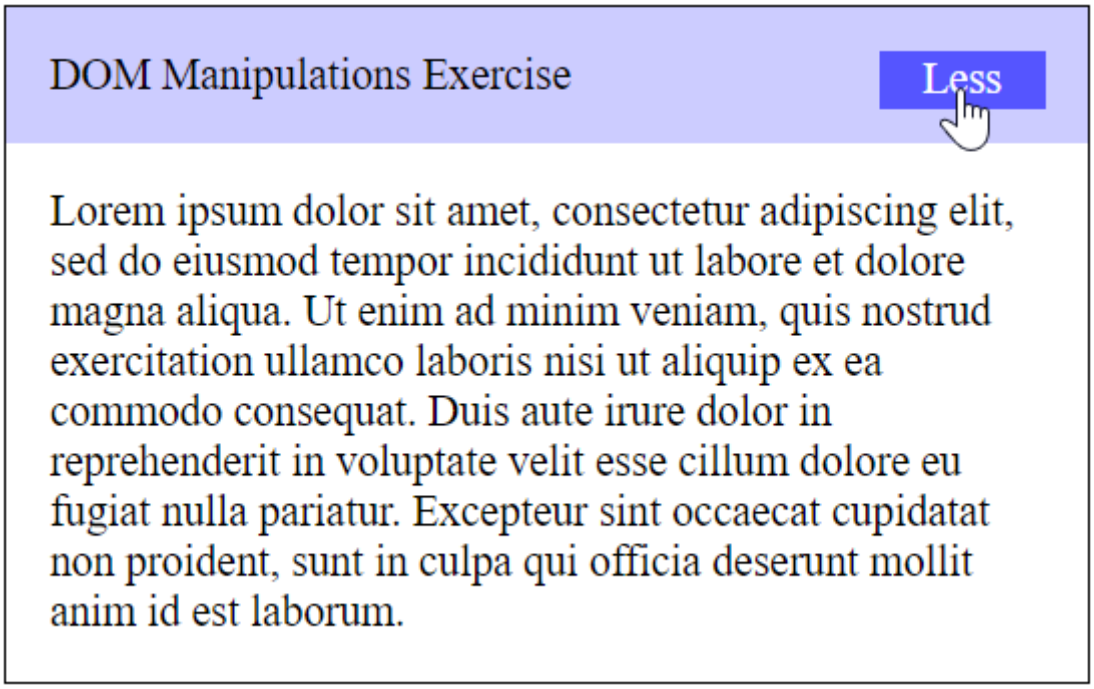
accordion.html

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <title>Accordion</title>
  <style>
    #accordion {
      border: 1px solid black;
      display: inline-block;
      width: 400px;
    }

    #accordion p {
      margin: 1em;
    }

    .button {
      float: right;
      background: #5555ff;
      padding: 0.1em 1em 0.1em 1em;
      color: white;
      cursor: pointer;
    }

    #extra {
      display: none;
    }

    .head {
      background: #ccccff;
      padding: 1em;
    }
  </style>
</head>
<body>
<div id="accordion">
  <div class="head">DOM Manipulations Exercise <span class="button"
onclick="toggle()">More</span></div>
  <div id="extra">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.</p>
  </div>
</div>
  <script>
    function toggle() {
      // TODO
    }
  </script>
</body>
</html>
```

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.

4. 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 id "**content**".

HTML and JavaScript Code

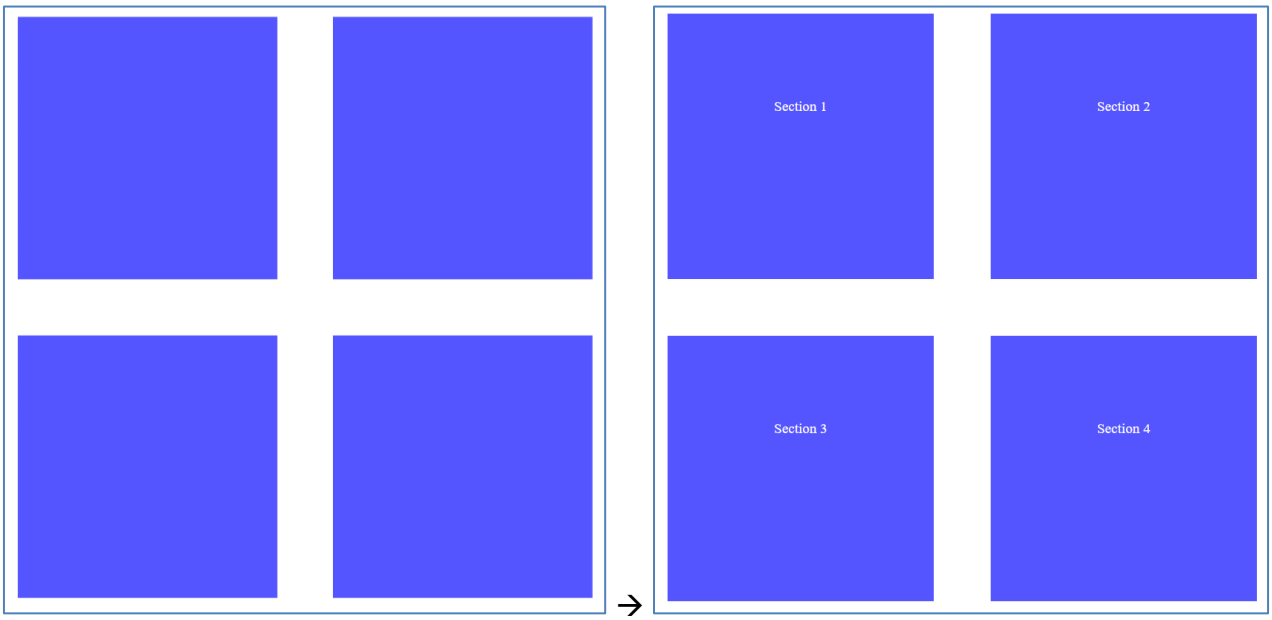
You are given the following **HTML** code:

sections.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
  <title>Sections</title>
  <style>
    #content {
      width: 1000px;
    }

    #content div {
      float: left;
      width: 300px;
      height: 300px;
      margin: 2em;
      background: #5555ff;
      text-align: center;
    }

    #content div p {
      color: white;
      margin: 6em 3em 6em 3em;
    }
  </style>
</head>
<div id="content">
</div>
<body onload="create(['Section 1', 'Section 2', 'Section 3', 'Section 4']);">
<script>
  function create(sentences) {
    // TODO:
  }
</script>
</body>
</html>
```

Example



5. Notification

Write a JS function that receives a string **message** and **displays** it inside a div with id "**notification**" for 2 seconds. The div starts **hidden** and when the function is called, **reveal** it. After 2 seconds, **hide** the div. In the example document, a notification is shown when you click the button.

HTML and JavaScript Code

You are given the following **HTML** code:

notification.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Notification</title>
  <style>
    body { width: 600px; text-align: center; }
    .header {
      background-color: #5555ff;
      color: white;
      position: relative;
      left: 0;
      top: 0;
      padding: 0.5em;
    }
    #container {
      position: relative;
    }
    .post {
      margin: 48px;
      text-align: left;
    }
    #notification {
      float: right;
      background: #119911;
      color: #ffffff;
      padding: 0.5em 2em 0.5em 2em;
      margin: 1em;
      display: none;
      position: absolute;
      top: 0;
      right: 0;
    }
  </style>
</head>
<body>
<div id="container">
  <header class="header">
    <h1>Welcome to our site</h1>
  </header>
  <div id="content">
    <article class="post">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
```

```
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p>
      <p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum
dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
    </article>
    <button onclick="notify('Something happened!')">Get notified</button>
  </div>
  <div id="notification"></div>
</div>
<script>
  function notify(message) {
    // TODO:
  }
</script>
</body>
</html>
```

Example

When we click the "Get notified" **button** a div appears in our upper-right corner. It should **disappear** in 2 seconds.

6. Time Converter

Create a JS app to convert between diferent time units. Your task is to add a **click** event listener to **all** buttons. When a button is **clicked**, read the **corresponding** input field and **convert** the value to the **three other** units of time and **display** it in the input fields.

HTML and JavaScript Code

You are given the following **HTML** code:

timeConverter.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Time Converter</title>
```

```
  <script src="timeConverter.js"></script>
  <style>
    label, input {
      display: inline-block;
      width: 5em;
    }
    label {
      text-align: right;
    }
  </style>
</head>
<body onload="attachEventsListeners()">
<h1>Time Converter</h1>
<div>
  <label for="days">Days: </label>
  <input type="text" id="days">
  <input id="daysBtn" type="button" value="Convert">
</div>
<div>
  <label for="hours">Hours: </label>
  <input type="text" id="hours">
  <input id="hoursBtn" type="button" value="Convert">
</div>
<div>
  <label for="minutes">Minutes: </label>
  <input type="text" id="minutes">
  <input id="minutesBtn" type="button" value="Convert">
</div>
<div>
  <label for="seconds">Seconds: </label>
  <input type="text" id="seconds">
  <input id="secondsBtn" type="button" value="Convert">
</div>
</body>
</html>
```

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

timeConverter.js

```
function attachEventsListeners() {
  // TODO: attach click events to all buttons
}
```

Example

One day is equal to 24 hours/1440 minutes/86400 seconds. Whichever button we **click** the input fields should change depending on the added value on the left. (If we make it 48 hours and click convert the days field should change to 2 and etc.).

7. * 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.

HTML and JavaScript Code

You are given the following **HTML** code:

distanceConverter.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Distance Converter</title>
  <script src="distanceConverter.js"></script>
  <style>
    label, input {
      display: inline-block;
      width: 5em;
    }

    label {
      text-align: right;
    }
  </style>
</head>
<body onload="attachEventsListeners()">
<h1>Distance Converter</h1>
<div>
  <label for="inputDistance">From:</label>
  <input type="text" id="inputDistance">
  <select id="inputUnits">
    <option value="km">Kilometers</option>
    <option value="m">Meters</option>
    <option value="cm">Centimeters</option>
    <option value="mm">Millimeters</option>
    <option value="mi">Miles</option>
    <option value="yrd">Yards</option>
    <option value="ft">Feet</option>
    <option value="in">Inches</option>
  </select>
  <input type="button" id="convert" value="Convert">
</div>
<div>
  <label for="outputDistance">To:</label>
  <input type="text" id="outputDistance" disabled="disabled">
  <select id="outputUnits">
    <option value="km">Kilometers</option>
    <option value="m">Meters</option>
```

```
    <option value="cm">Centimeters</option>
    <option value="mm">Millimeters</option>
    <option value="mi">Miles</option>
    <option value="yrd">Yards</option>
    <option value="ft">Feet</option>
    <option value="in">Inches</option>
  </select>
</div>
</body>
</html>
```

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

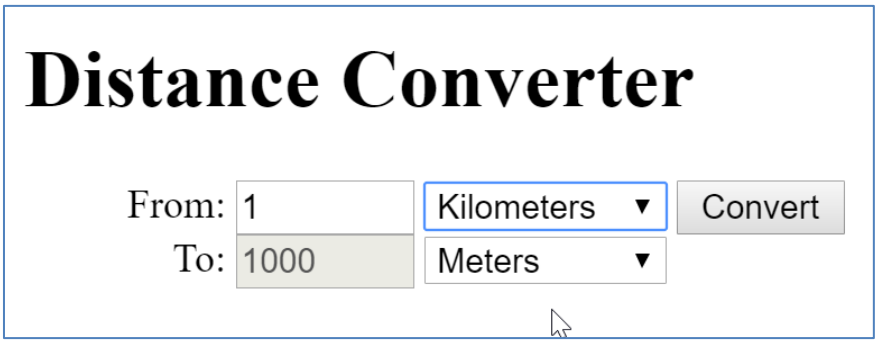
distanceConverter.js

```
function attachEventsListeners() {
  // TODO: attach 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.