Лабораторна робота №3

Тема: Обробка подій JavaScript.

Хід роботи:

Завдання на лабораторну роботу:

1. ***Завдання на лабораторну роботу:***

Завдання 1. Користувач вводить своє ім’я в текстове поле. При натисканні на кнопкувідображається привітання:

Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 1</title>
    <link rel="stylesheet" href="./styles/css/task_1.css">5
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <table class="button_section__table">
                <tr>
                    <td>
                        <label for="textGreeting" class="table__label_greeting">Enter your
name:</label>
                    </td>
                    <td>
                        <input type="text" name="textGreeting" id="textGreeting"
class="table__text_greeting" placeholder="Your name">
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="button" value="Greeting" id="buttonGreeting"
class="table__button_greeting">
                    </td>
                </tr>
            </table>
        </section>
    </div>

    <script src="./js/task_1.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

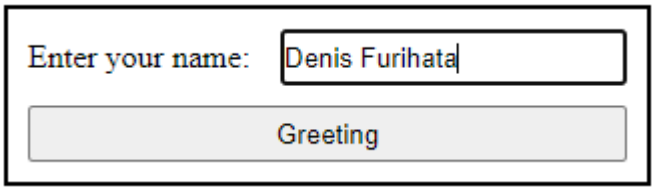
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 30vw

.button_section__table
  border-collapse: collapse

.table__text_greeting
  margin-left: 15px
  height: 28px

.table__button_greeting
  height: 28px
```

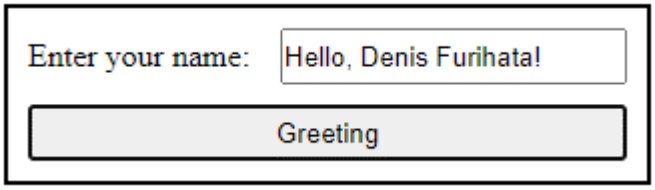
```
  width: 100%
  margin: 10px 0 0 0
```

- ***JS:***

```
'use strict';

const textGreeting = document.querySelector('#textGreeting');
const buttonGreeting = document.querySelector('#buttonGreeting');
buttonGreeting.addEventListener('click', () => {
    textGreeting.value = 'Hello, ' + textGreeting.value + '!';
})
```

Результат програми:





Завдання 2. Користувач заповнює два текстові поля. При натисканні на кнопку, значення вцих полях міняються місцями:

Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 2</title>
    <link rel="stylesheet" href="./styles/css/task_2.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <table class="button_section__table">
                <tr>
                    <td>
                        <label for="textFirstText" class="table__label_first_text">Enter
first text:</label>
                    </td>
                    <td>
                        <input type="text" name="textFirstText" id="textFirstText"
class="table__text_first_text" placeholder="Some text">
                    </td>
                </tr>
```

```
                <tr>
                    <td>
                        <label for="textSecondText" class="table__label_second_text">Enter
second text:</label>
                    </td>
                    <td>
                        <input type="text" name="textSecondText" id="textSecondText"
class="table__second_text" placeholder="Some text">
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="button" value="Swap values in two text inputs"
id="buttonSwap" class="table__button_swap">
                    </td>
                </tr>
            </table>
        </section>
    </div>

    <script src="./js/task_2.js"></script>
</div>
</body>
</html>
```

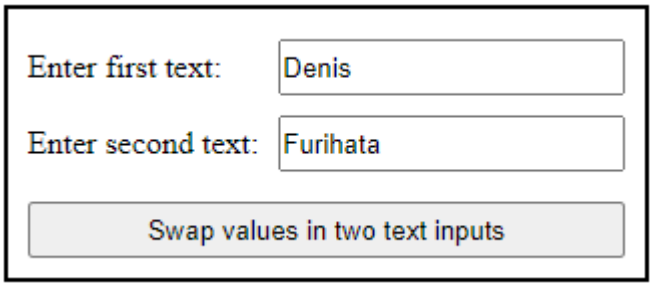
- ***SASS:***

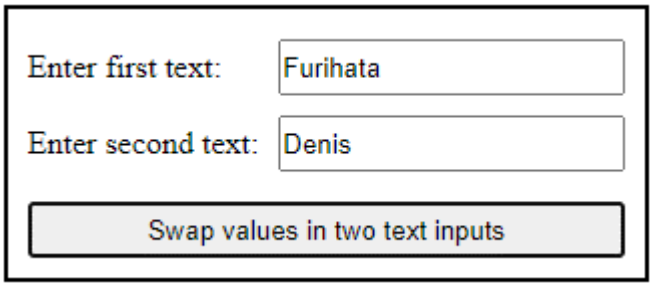
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 30vw

.button_section__table
  border-collapse: collapse

.table__text_first_text, .table__second_text
  margin: 5px 0 5px 10px
  height: 28px

.table__button_swap
  height: 28px
  width: 100%
  margin: 10px 0 0 0
```

- ***JS:***

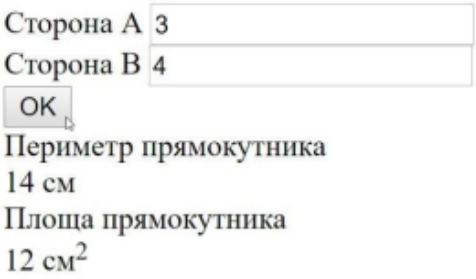
```
'use strict';

let textFirstText = document.querySelector('#textFirstText');
let textSecondText = document.querySelector('#textSecondText');
const buttonSwap = document.querySelector('#buttonSwap');
buttonSwap.addEventListener('click', () => {
    [textFirstText.value, textSecondText.value] = [textSecondText.value,
textFirstText.value]
});
```

Результат програми:





Завдання 3. Дано сторони прямокутника (вводяться користувачем). При натисканні накнопку обчислити периметр та площу прямокутника:



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 3</title>
    <link rel="stylesheet" href="./styles/css/task_3.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <table class="button_section__table">
                <tr>
                    <td>
                        <label for="textSideA" class="table__label_side_a">Enter side
A:</label>
                    </td>
                    <td>
                        <input type="text" name="textSideA" id="textSideA"
class="table__text_side_a" placeholder="Rectangle side A">
                    </td>
                </tr>
                <tr>
                    <td>
                        <label for="textSideB" class="table__label_side_b">Enter side
B:</label>
```

```
                    </td>
                    <td>
                        <input type="text" name="textSideB" id="textSideB"
class="table__text_side_b" placeholder="Rectangle side B">
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="button" value="Get perimeter and area of rectangle"
id="buttonPerimeterArea" class="table__button_perimeter_area">
                    </td>
                </tr>
            </table>
            <div class="button_section__output_perimeter_area"></div>
        </section>
    </div>

    <script src="./js/task_3.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

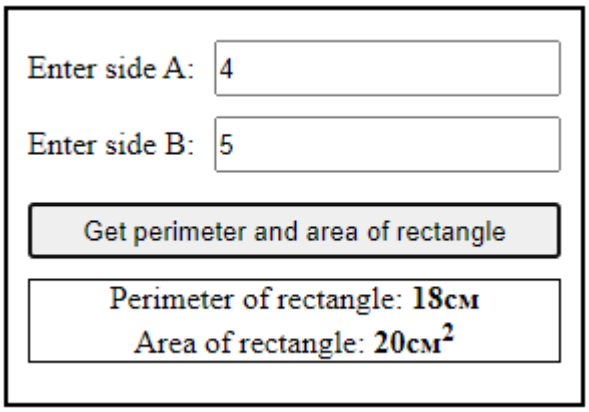
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 30vw

.button_section__table
  border-collapse: collapse

.table__text_side_a, .table__text_side_b
  margin: 5px 0 5px 10px
  height: 28px

.table__button_perimeter_area
  height: 28px
  width: 100%
  margin: 10px 0 0 0

.output_perimeter_area__section
  border: 1px solid black
  margin: 10px 0
  text-align: center
```

- ***JS:***

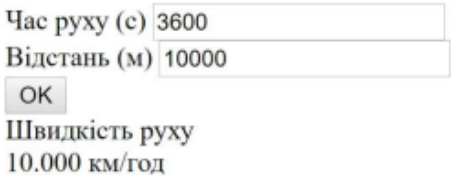
```
'use strict';

const buttonPerimeterArea = document.querySelector('#buttonPerimeterArea');
buttonPerimeterArea.addEventListener('click', () => {
    const textSideA = parseInt(document.querySelector('#textSideA').value);
    const textSideB = parseInt(document.querySelector('#textSideB').value);
    if (isNaN(textSideA) || isNaN(textSideB)) {
        document.querySelector('.button_section__output_perimeter_area').innerHTML =
            `<section class="output_perimeter_area__section">
                <p>Values were entered <strong>incorrectly</strong>!</p>
            </section>`;
```

```
    } else {
        const rectanglePerimeter = Math.round((textSideA * 2 + textSideB * 2) * 100) / 100;
        const rectangleArea = textSideA * textSideB;
        document.querySelector('.button_section__output_perimeter_area').innerHTML =
            `<section class="output_perimeter_area__section">
                <p>Perimeter of rectangle: <strong>${rectanglePerimeter}см</strong></p>
                <p>Area of rectangle: <strong>${rectangleArea}см<sup>2</sup></strong></p>
            </section>`;
    }
});
```

Результат програми:



Завдання 4 . Дано час руху в секундах та відстань в метрах (вводяться користувачем), що подолав велосипедист. При натисканні на кнопку обчислюється йогошвидкість руху в км/год:



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 4</title>
    <link rel="stylesheet" href="./styles/css/task_4.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <table class="button_section__table">
                <tr>
                    <td>
                        <label for="textTimeOfMovement"
class="table__label_time_of_movement">Enter time of movement(sec.):</label>
                    </td>
                    <td>
                        <input type="text" name="textTimeOfMovement"
id="textTimeOfMovement" class="table__text_time_of_movement" placeholder="Time of
movement">
```

```
                    </td>
                </tr>
                <tr>
                    <td>
                        <label for="textDistance" class="table__label_distance">Enter
distance(m.):</label>
                    </td>
                    <td>
                        <input type="text" name="textDistance" id="textDistance"
class="table__text_distance" placeholder="Distance">
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="button" value="Get speed of movement"
id="buttonSpeedMovement" class="table__button_speed_movement">
                    </td>
                </tr>
            </table>
            <div class="button_section__output_speed_movement"></div>
        </section>
    </div>

    <script src="./js/task_4.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

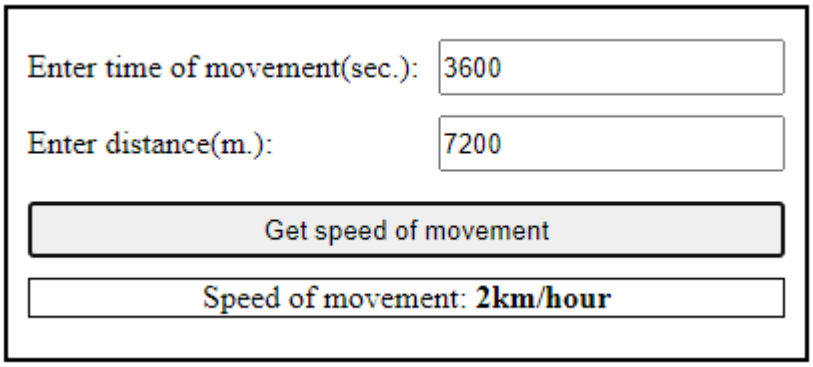
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 30vw

.button_section__table
  border-collapse: collapse

.table__text_time_of_movement, .table__text_distance
  margin: 5px 0 5px 10px
  height: 28px

.table__button_speed_movement
  height: 28px
  width: 100%
  margin: 10px 0 0 0

.output_speed_movement__section
  border: 1px solid black
  margin: 10px 0
  text-align: center
```

- ***JS:***

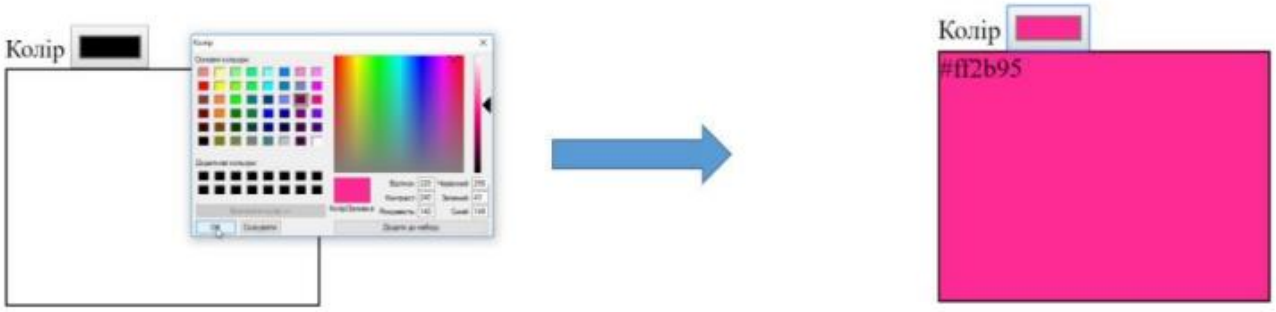
```
'use strict';

document.querySelector('#buttonSpeedMovement').addEventListener('click', () => {
    const timeOfMovement = parseFloat(document.querySelector('#textTimeOfMovement').value);
    const distance = parseFloat(document.querySelector('#textDistance').value);
```

```
    if (isNaN(timeOfMovement) || isNaN(distance)) {
        document.querySelector('.button_section__output_speed_movement').innerHTML =
            `<section class="button_section__output_speed_movement">
                <p>Incorrect type of entered values!</p>
            </section>`;
    } else {
        console.log(timeOfMovement, distance);
        const speedOfMovement = Math.round((distance / timeOfMovement) * 100) / 100;
        document.querySelector('.button_section__output_speed_movement').innerHTML =
            `<section class="output_speed_movement__section">
                <p>Speed of movement: <strong>${speedOfMovement}km/hour</strong></p>
            </section>`;
    }
});
```

Результат програми:



Завдання 5. Дано блок і поле для вибору кольору. При здійсненні вибору кольору в блоціз'являється 16-ве значення кольору, а сам блок заливається даним кольором:



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 5</title>
    <link rel="stylesheet" href="./styles/css/task_5.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <table class="button_section__table">
                <tr>
                    <td>
                        <label for="colorChooseColor"
class="table__label_choose_color">Choose some color:</label>
                    </td>
```

```
                    <td>
                        <input type="color" name="colorChooseColor" id="colorChooseColor"
class="table__color_choose_color">
                    </td>
                </tr>
            </table>
            <div class="button_section__output_color"></div>
        </section>
    </div>

    <script src="./js/task_5.js"></script>
</div>
</body>
</html>
```

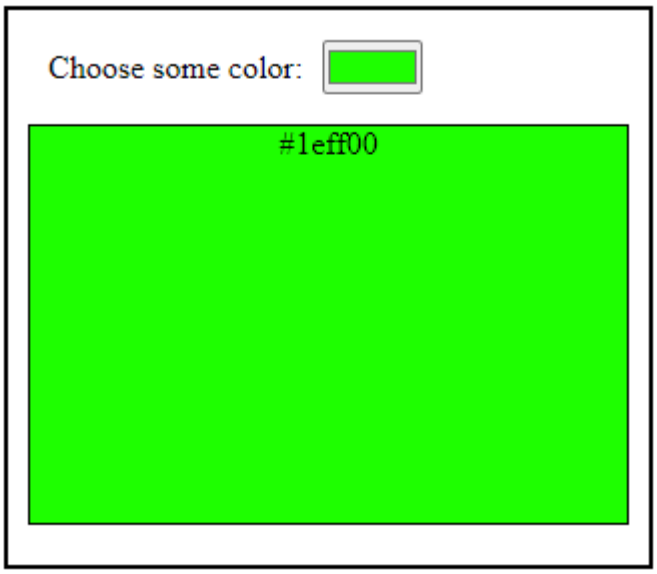
- ***SASS:***

```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 30vw

.button_section__table
  border-collapse: collapse

.table__label_choose_color, .table__color_choose_color
  margin: 5px 0 5px 10px

.button_section__output_color
  border: 1px solid black
  margin: 10px 0
  text-align: center
  width: 300px
  height: 200px
```

- ***JS:***

```
'use strict';

const colorChooseColor = document.querySelector('#colorChooseColor');

colorChooseColor.addEventListener('input', () => {
    const color = colorChooseColor.value;
    const outputColor = document.querySelector('.button_section__output_color');

    outputColor.innerHTML = `<p class="output_color__p">${color}</p>`;
    outputColor.style.backgroundColor = color;
});
```

Результат програми:



Завдання 6. Дано випадний список міст та відповідні зображення міст. Посилання на зображення. При виборі міста зі списку на сторінці з'являється його зображення:

Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 6</title>
    <link rel="stylesheet" href="./styles/css/task_6.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <table class="button_section__table">
                <tr>
                    <td>
                        <label for="selectChooseImg" class="table__label_choose_img">Choose
some image to see it:</label>
                    </td>
                    <td>
                        <select name="selectChooseImg" id="selectChooseImg"
class="table__select_choose_img">
                            <option value="default">--> Select image <--</option>
                            <option value="new_york">New York</option>
                            <option value="los_angeles">Los Angeles</option>
                            <option value="chicago">Chicago</option>
                        </select>
```

```
                    </td>
                </tr>
            </table>
            <div class="button_section__output_img"></div>
        </section>
    </div>

    <script src="./js/task_6.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

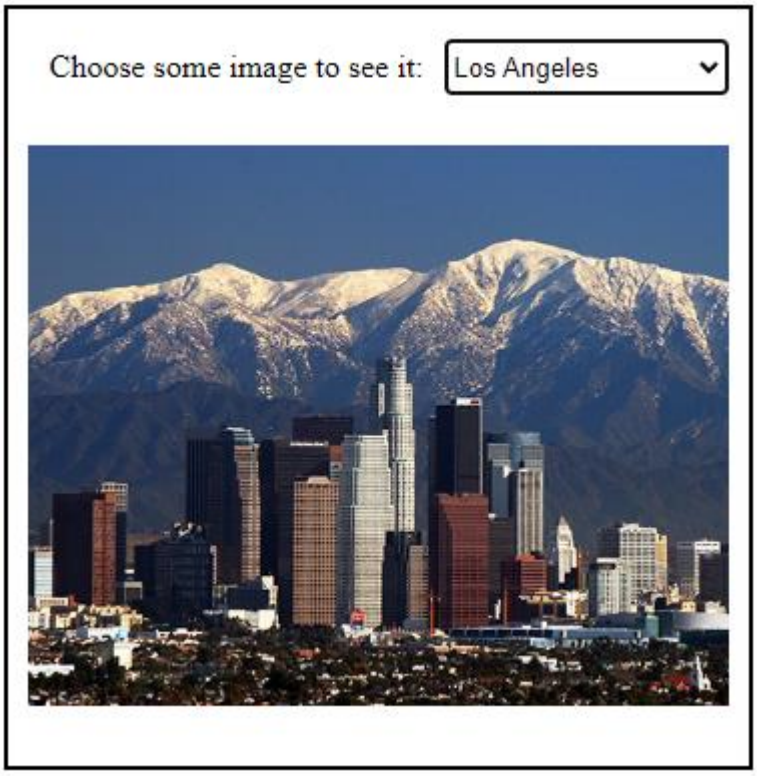
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 30vw

.button_section__table
  border-collapse: collapse

.table__label_choose_img, .table__select_choose_img
  margin: 5px 0 5px 10px
  height: 28px

.button_section__output_img
  margin: 10px 0
  width: 100%
  height: 300px
```

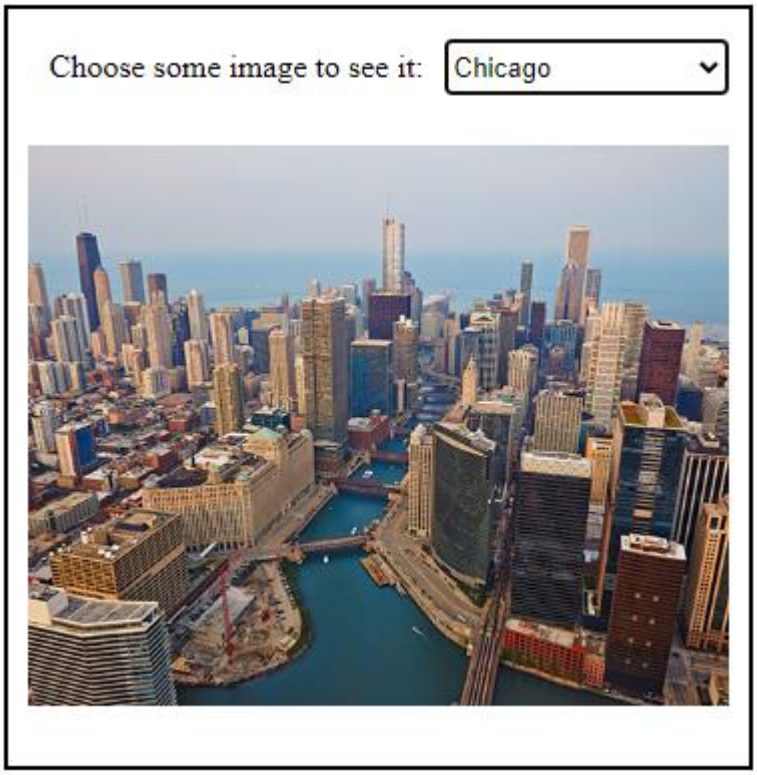
- ***JS:***

```
'use strict';

const selectChooseImg = document.querySelector('#selectChooseImg');

selectChooseImg.addEventListener('input', () => {
    const img = selectChooseImg.value;
    const outputImg = document.querySelector('.button_section__output_img');

    switch (img) {
        case 'los_angeles': {
            outputImg.style.background = "url('../../.././img/los-angeles.jpg') no-repeat
center";
            outputImg.style.backgroundSize = 'contain';
            break;
        }
        case 'new_york': {
            outputImg.style.background = "url('../../.././img/new-york.jpg') no-repeat
center";
            outputImg.style.backgroundSize = 'contain';
            break;
        }
        case 'chicago': {
            outputImg.style.background = "url('../../.././img/chicago.jpg')  no-repeat
center";
            outputImg.style.backgroundSize = 'contain';
            break;
```

```
        }
    }
});
```

Результат програми:



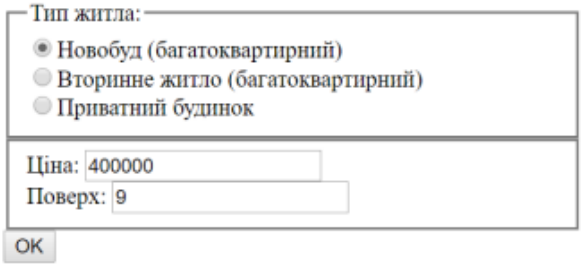


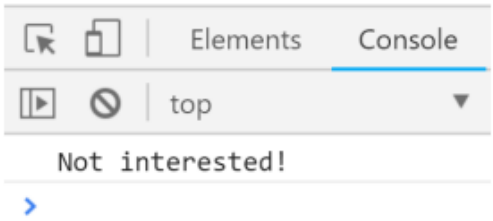
Завдання 7.Нам потрібно придбати житло (квартиру або будинок) при таких умовах:

1) Ціна не перевищує 600 тис. грн.
2) Новобуд (з 2-го по 6-тий поверх) або приватний будинок

За даною пропозицією визначте, чи можемо ми придбати дану нерухомість

Приклад роботи скрипта:





Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 7</title>
    <link rel="stylesheet" href="./styles/css/task_7.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <fieldset class="button_section__dwelling_fieldset">
                <legend>Type of dwelling:</legend>
                <table class="button_section__table">
                    <tr>
                        <td>
```

```
                            <input type="radio" name="radioBuildingType"
id="radioNewBuilding"
                                   class="table__number_new_building" value="New building">
                        </td>
                        <td>
                            <label for="radioNewBuilding"
class="table__label_new_building">New
                                building(multi-apartment)</label>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <input type="radio" name="radioBuildingType"
id="radioSecondaryHousing"
                                   class="table__number_secondary_housing" value="Secondary
housing">
                        </td>
                        <td>
                            <label for="radioSecondaryHousing"
class="table__label_secondary_housing">Secondary
                                housing(multi-apartment)</label>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            <input type="radio" name="radioBuildingType"
id="radioPrivateHouse"
                                   class="table__number_private_house" value="Private
house">
                        </td>
                        <td>
                            <label for="radioPrivateHouse"
class="table__label_private_house">Private house</label>
                        </td>
                    </tr>
                </table>
            </fieldset>
            <fieldset class="button_section__actions_fieldset">
                <table class="button_section__table">
                    <tr class="table__price_tr">
                        <td>
                            <label for="numberPrice"
class="table__label_price">Price:</label>
                        </td>
                        <td>
                            <input type="number" name="numberPrice" id="numberPrice"
class="table__number_price">
                        </td>
                    </tr>
                    <tr class="table__floor_tr">
                        <td>
                            <label for="numberFloor"
class="table__label_floor">Floor:</label>
                        </td>
                        <td>
                            <input type="number" name="numberFloor" id="numberFloor"
class="table__number_floor">
                        </td>
                    </tr>
                    <tr class=table__check_suggestions_tr"">
                        <td colspan="2">
                            <input type="button" value="Check suggestions"
id="buttonCheckSuggestions"
                                   class="table__button_check_suggestions">
                        </td>
                    </tr>
                </table>
            </fieldset>
            <div class="button_section__output_status"></div>
        </section>
```

```
    </div>

    <script src="./js/task_7.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

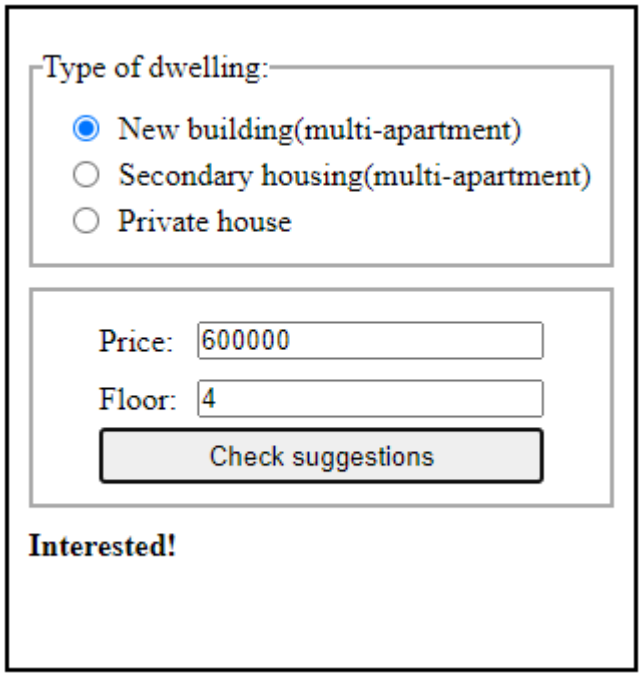
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 35vw

.button_section__dwelling_fieldset, .button_section__actions_fieldset
  border: 2px solid darkgrey
  padding: 5px
  margin: 10px 0
  width: 100%

.button_section__table
  border-collapse: collapse
  margin: 5px
  display: flex
  align-items: center
  flex-direction: column

.table__label_new_building, .table__label_secondary_housing, .table__label_private_house
  margin: 0 0 0 10px
  height: 28px

.table__button_check_suggestions
  width: 100%
  height: 28px

.table__number_price, .table__number_floor, .table__number_new_building,
.table__number_secondary_housing, .table__number_private_house
  margin: 5px 0 5px 10px

.button_section__output_status
  margin: 10px 0
  width: 100%
  height: 50px
```

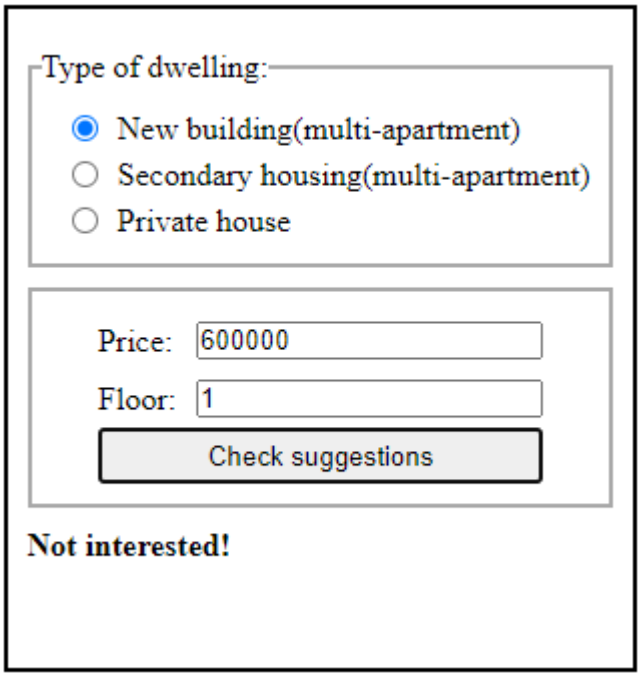
- ***JS:***

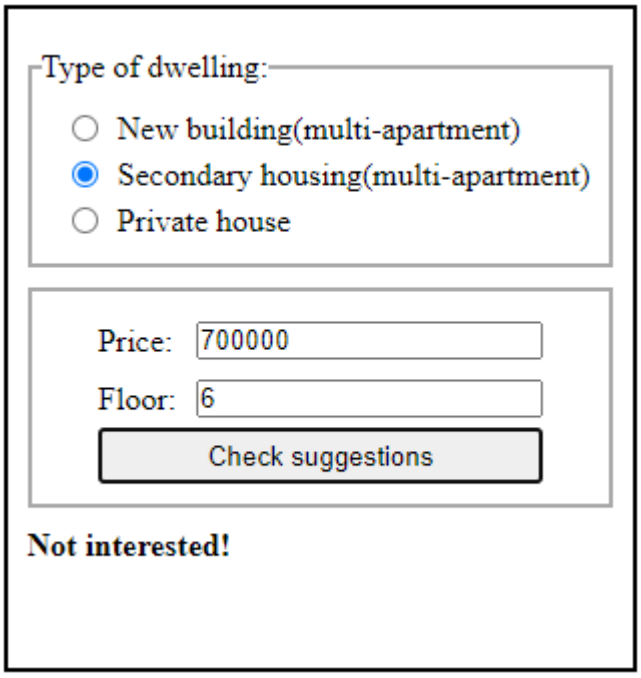
```
'use strict';
const radioBuildingType = document.querySelectorAll('input[name="radioBuildingType"]');

const hiddenFloor = () => {
    document.querySelector('.table__floor_tr').setAttribute('hidden', '');
};

const unHiddenFloor = () => {
    document.querySelector('.table__floor_tr').removeAttribute('hidden');
};

radioBuildingType[0].addEventListener('change', unHiddenFloor)
radioBuildingType[1].addEventListener('change', unHiddenFloor)
```

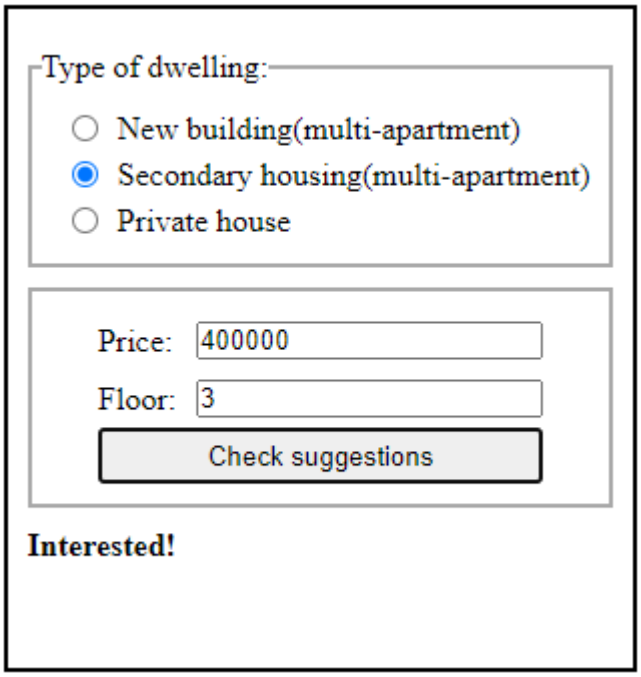
```
radioBuildingType[2].addEventListener('change', hiddenFloor)

const buttonCheckSuggestions = document.querySelector('#buttonCheckSuggestions');

buttonCheckSuggestions.addEventListener('click', () => {
    const numberPrice = document.querySelector('#numberPrice').value;
    const numberFloor = document.querySelector('#numberFloor').value;
    const outputField = document.querySelector('.button_section__output_status');

    if (!radioBuildingType[0].checked && !radioBuildingType[1].checked &&
!radioBuildingType[2].checked) {
        outputField.innerHTML = `<p>Choose type of dwelling!</p>`;
        console.log('Choose type of dwelling!');
    } else {
        if (numberPrice > 600000 || numberPrice < 0) {
            outputField.innerHTML = `<p><strong>Not interested!</strong></p>`;
            console.log('Not interested!');
        } else {
            if (radioBuildingType[0].checked || radioBuildingType[1].checked) {
                if (numberFloor < 1 || numberFloor > 47) {
                    outputField.innerHTML = `<p><strong>Not interested!</strong></p>`;
                    console.log('Not interested!');
                } else {
                    if (numberFloor >= 2 && numberFloor <= 6) {
                        outputField.innerHTML =
                            `<p><strong>Interested!</strong></p>`;
                        console.log('Interested!');
                    } else {
                        outputField.innerHTML =
                            `<p><strong>Not interested!</strong></p>`;
                        console.log('Not interested!');
                    }
                }
            } else {
                outputField.innerHTML =
                    `<p><strong>Interested!</strong></p>`;
                console.log('Interested!');
            }
        }
    }
});
```

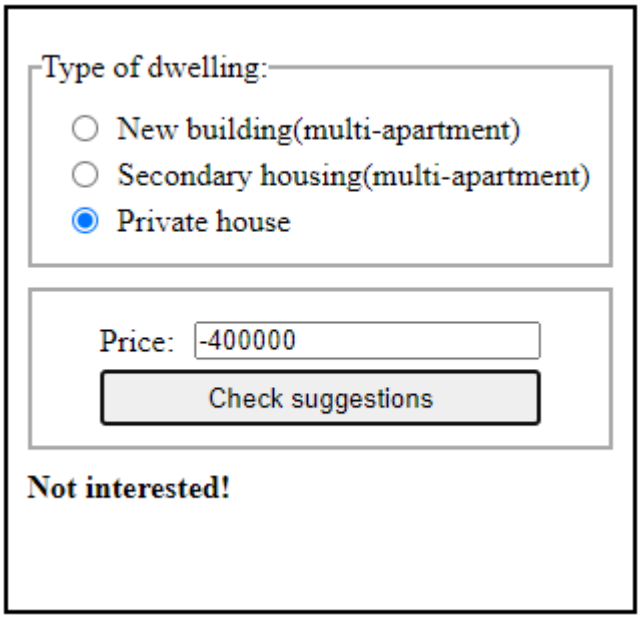
Результат програми:

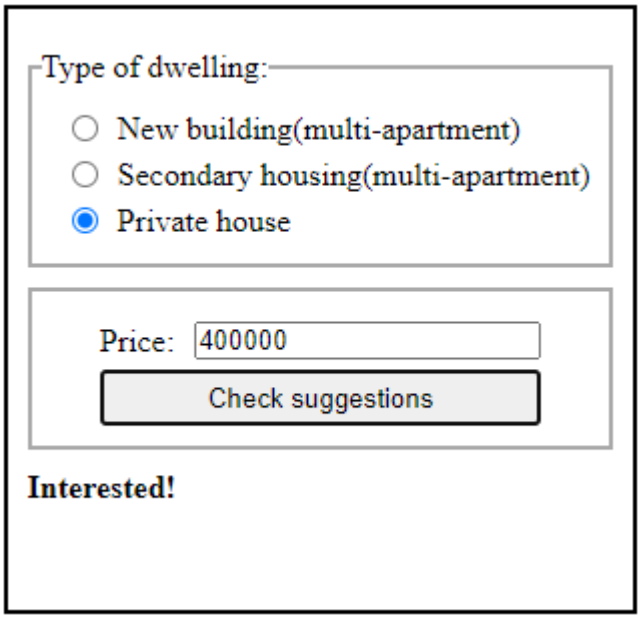



Type of dwelling:
New building(multi-apartment)
Secondary housing(multi-apartment)
Private house
Price: 600000
Floor: 1
Check suggestions
Not interested!


Type of dwelling:
New building(multi-apartment)
Secondary housing(multi-apartment)
Private house
Price: 700000
Floor: 6
Check suggestions
Not interested!


Type of dwelling:
New building(multi-apartment)
Secondary housing(multi-apartment)
Private house
Price: 400000
Floor: 3
Check suggestions
Interested!

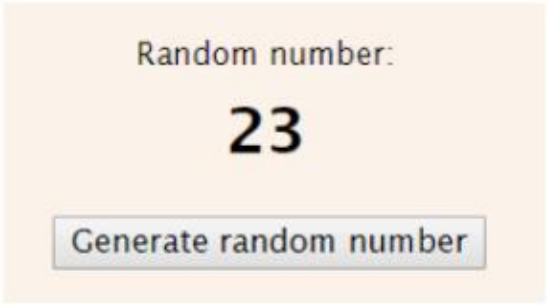

Type of dwelling:
New building(multi-apartment)
Secondary housing(multi-apartment)
Private house
Price: -400000
Check suggestions
Not interested!



2. *Домашнє завдання:*

Завдання 8:

Створити html-структуру для генерування випадкових чисел. При натисканні на кнопку генерується випадкове число у довільному проміжку яке виводиться у дів-блок.



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 8</title>
    <link rel="stylesheet" href="./styles/css/task_8.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <fieldset class="button_section__random_number_fieldset">
                <label for="buttonRandomNumber" class="table__label_random_number">Random
number</label>
                <div class="button_section__div_output_number"><h3
class="div_output_number__h3">0</h3></div>
                <input type="button" value="Generate random number[-100; 100]"
name="buttonRandomNumber" id="buttonRandomNumber" class="table__button_random_number">
```

```
            </fieldset>
        </section>
    </div>

    <script src="./js/task_8.js"></script>
</div>
</body>
</html>
```

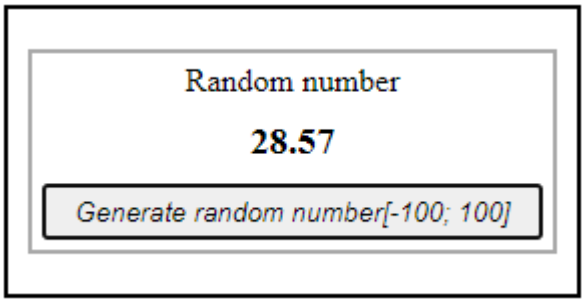
- ***SASS:***

```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 35vw

.button_section__random_number_fieldset
  text-align: center
  border: 2px solid darkgrey
  padding: 5px
  margin: 10px 0
  width: 100%

.table__button_random_number
  width: 250px
  height: 28px
  font-style: italic

.button_section__div_output_number
  margin: 10px 0
  width: 100%
```

- ***JS:***

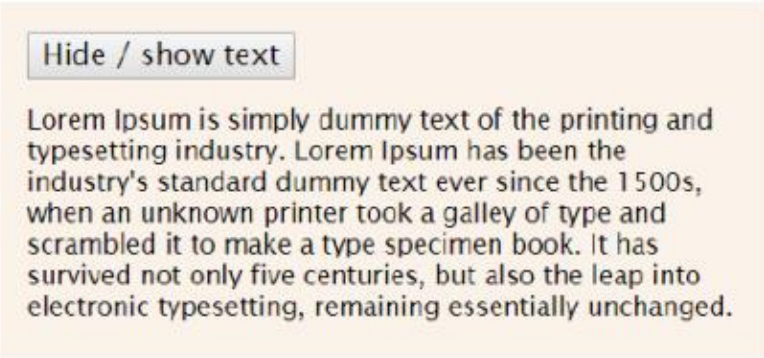
```
'use strict';

const getRandomFloat = (fromNumber, toNumber) => Math.round((Math.random() * (toNumber -
fromNumber) + fromNumber) * 100) / 100;

const buttonRandomNumber = document.querySelector('#buttonRandomNumber');

buttonRandomNumber.addEventListener('click', () => {
    document.querySelector('.div_output_number__h3').innerHTML = getRandomFloat(-100, 100);
})
```

Результат програми:



Завдання 9:

Створити html сторінку з кнопкою та текстом. При натисканні на кнопку текст має ховатись або відображатись:



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 9</title>
    <link rel="stylesheet" href="./styles/css/task_9.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <fieldset class="button_section__show_hide_fieldset">
                <input type="button" value="Show/hide text" name="buttonShowHideText"
                       id="buttonShowHideText" class="table__button_show_hide_text">
                <div class="button_section__div_text">
                    <p>
                        Lorem ipsum dolor sit amet, consectetur adipisicing elit.
Accusantium, animi
                        blanditiis consequuntur dicta eaque earum eius enim et eum facere,
id ipsam
                        laboriosam officiis perspiciatis, quidem rem saepe unde vitae.
                    </p>
                    <p>
                        Adipisci aperiam, cumque distinctio dolorem ducimus, eum eveniet
expedita fuga illum
                        in inventore, labore minima nam nobis non nostrum placeat quam
quibusdam quisquam
                        rerum totam ullam veniam. A, aut, cupiditate.
                    </p>
                    <p>
                        Ad, consectetur dicta doloribus et iure laudantium ratione vitae!
Animi dolores
                        excepturi labore neque officia porro quam sapiente tempora
voluptatibus. Autem
                        cupiditate dignissimos ea facere incidunt iure omnis, reprehenderit
unde.
                    </p>
                    <p>
                        Aspernatur cum debitis dignissimos esse id, illo itaque iusto
laboriosam maiores
                        natus, nihil odit officia quibusdam quis quod recusandae
repudiandae tempora totam
                        velit veniam? Aliquam pariatur quaerat rem sit suscipit.
                    </p>
```

```
                    <p>
                        Architecto, aspernatur beatae commodi cum deserunt distinctio ea
error eum ex
                        excepturi impedit inventore ipsa ipsum iusto laborum modi molestiae
nemo odit optio
                        praesentium quam repellat saepe sed voluptatibus voluptatum!
                    </p>
                </div>
            </fieldset>
        </section>
    </div>

    <script src="./js/task_9.js"></script>
</div>
</body>
</html>
```

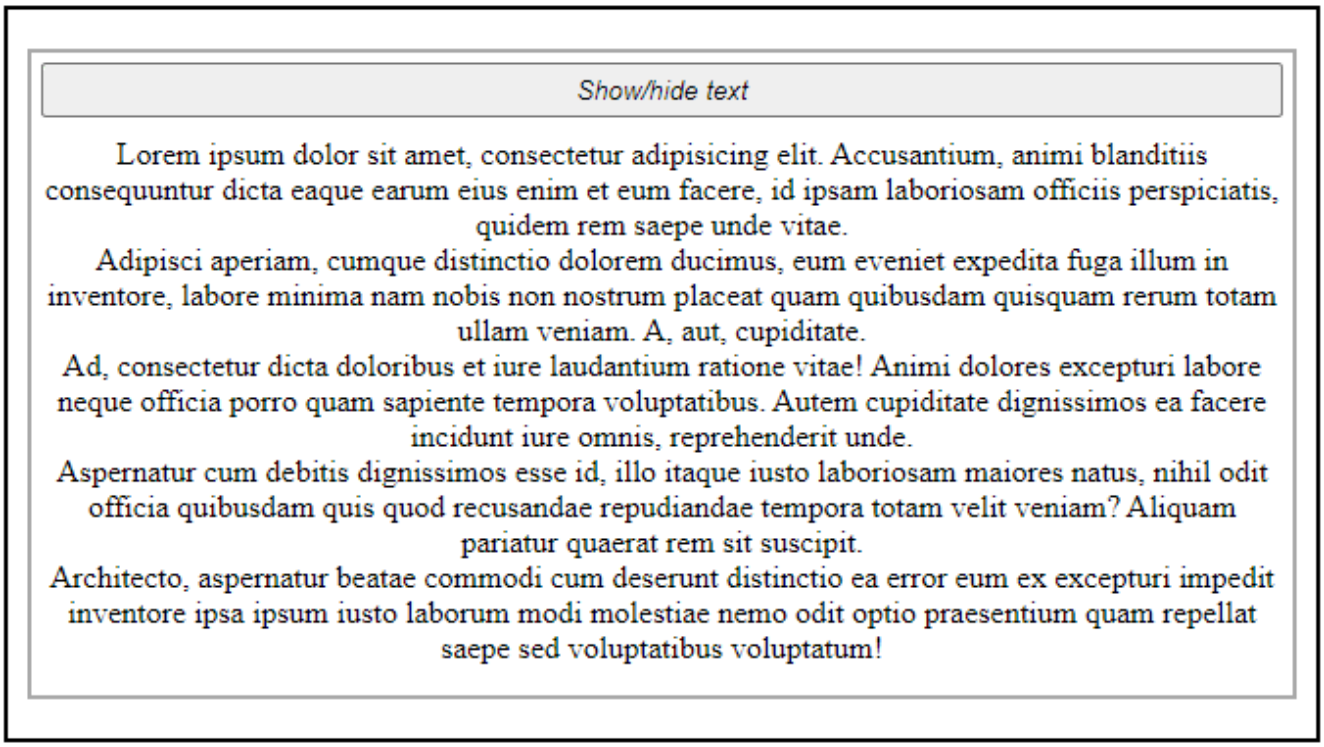
- ***SASS:***

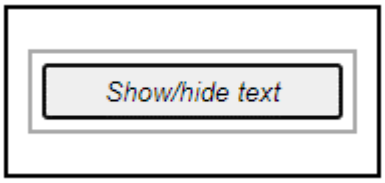
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid black
  margin: 10px 0
  padding: 10px
  max-width: 35vw

.button_section__show_hide_fieldset
  text-align: center
  border: 2px solid darkgrey
  padding: 5px
  margin: 10px 0
  width: 100%

.table__button_show_hide_text
  width: 100%
  min-width: 150px
  height: 28px
  font-style: italic

.button_section__div_text
  margin: 10px 0
  width: 100%
```

- ***JS:***

```
'use strict';

const buttonShowHideText = document.querySelector('#buttonShowHideText');

buttonShowHideText.addEventListener('click', () => {
    const text = document.querySelector('.button_section__div_text');

    if(text.hasAttribute('hidden')){
        text.removeAttribute('hidden');
    } else{
        text.setAttribute('hidden', '');
    }
});
```

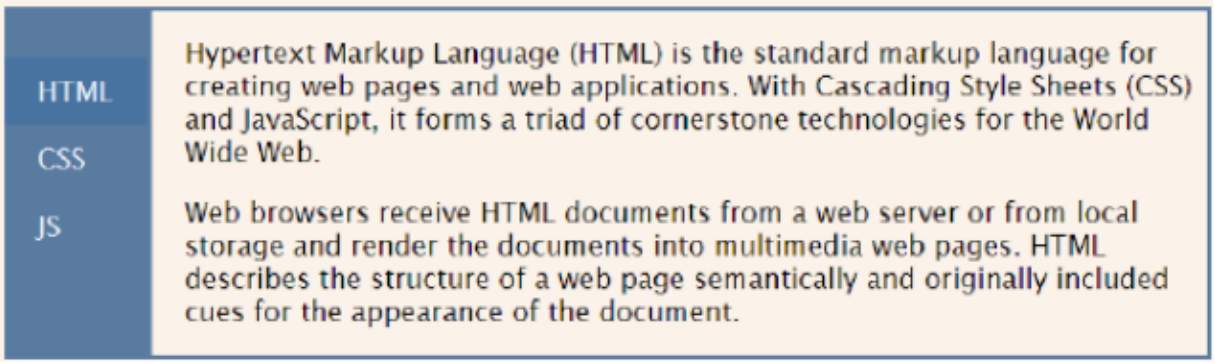
Результат програми:





Завдання 10:

Створити html сторінку із вкладками. У кожної вкладки є свій вміст. В один момент активною може бути тільки одна вкладка.



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 10</title>
    <link rel="stylesheet" href="./styles/css/task_10.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
```

```
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <aside class="button_section__aside">
                <div class="aside__HTML" id="asideHTML">HTML</div>
                <div class="aside__CSS" id="asideCSS">CSS</div>
                <div class="aside__JS" id="asideJS">JS</div>
            </aside>
            <div class="button_section__div_HTML_text" id="divHTMLText">
                <h1>HTML</h1>
                <p>
                    The HyperText Markup Language or HTML is the standard markup
                    language for documents designed to be displayed in a web browser. It
can be assisted by technologies
                    such as Cascading Style Sheets (CSS) and scripting languages such as
JavaScript.
                </p>
                <p>
                    Web browsers receive HTML documents from a web server or from local
storage and render the documents
                    into multimedia web pages. HTML describes the structure of a web page
semantically and originally
                    included cues for the appearance of the document.
                </p>
                <p>
                    HTML elements are the building blocks of HTML pages. With HTML
constructs, images and other objects
                    such as interactive forms may be embedded into the rendered page. HTML
provides a means to create
                    structured documents by denoting structural semantics for text such as
headings, paragraphs, lists,
                    links, quotes and other items. HTML elements are delineated by tags,
written using angle brackets.
                    Tags such as "img" and "input" directly introduce content into the
page. Other tags such as "p"
                    surround and provide information about document text and may include
other tags as sub-elements.
                    Browsers do not display the HTML tags but use them to interpret the
content of the page.
                </p>
                <p>
                    HTML can embed programs written in a scripting language such as
JavaScript, which affects the
                    behavior and content of web pages. Inclusion of CSS defines the look
and layout of content. The
                    World Wide Web Consortium (W3C), former maintainer of the HTML and
current maintainer of the CSS
                    standards, has encouraged the use of CSS over explicit presentational
HTML since 1997.[2] A form of
                    HTML, known as HTML5, is used to display video and audio, primarily
using the
                    "canvas" element, in collaboration with javascript.
                </p>
            </div>
            <div class="button_section__div_CSS_text" id="divCSSText">
                <h1>CSS</h1>
                <p>
                    Cascading Style Sheets (CSS) is a style sheet language used for
describing the presentation of a
                    document written in a markup language such as HTML or XML (including
XML dialects such as SVG,
                    MathML or XHTML).[1] CSS is a cornerstone technology of the World Wide
Web, alongside HTML and
                    JavaScript.[2]
                </p>
                <p>
                    CSS is designed to enable the separation of presentation and content,
including layout, colors, and
                    fonts.[3] This separation can improve content accessibility; provide
more flexibility and control in
```

```
                    the
                    specification of presentation characteristics; enable multiple web
pages to share formatting by
                    specifying the relevant CSS in a separate .css file, which reduces
complexity and repetition in the
                    structural content; and enable the .css file to be cached to improve
the page load speed between the
                    pages that share the file and its formatting.
                </p>
                <p>
                    Separation of formatting and content also makes it feasible to present
the same markup page in
                    different
                    styles for different rendering methods, such as on-screen, in print, by
voice (via speech-based
                    browser
                    or screen reader), and on Braille-based tactile devices. CSS also has
rules for alternate formatting
                    if
                    the content is accessed on a mobile device.[4]
                </p>
                <p>
                    The name cascading comes from the specified priority scheme to
determine which style rule applies if
                    more than one rule matches a particular element. This cascading
priority scheme is predictable.
                </p>
                <p>
                    The CSS specifications are maintained by the World Wide Web Consortium
(W3C). Internet media type
                    (MIME
                    type) text/css is registered for use with CSS by RFC 2318 (March 1998).
The W3C operates a free CSS
                    validation service for CSS documents.[5]
                </p>
                <p>
                    In addition to HTML, other markup languages support the use of CSS
including XHTML, plain XML, SVG,
                    and
                    XUL.
                </p>
            </div>
            <div class="button_section__div_JS_text" id="divJSText">
                <h1>JS</h1>
                <p>
                    JavaScript (/ˈdʒɑːvəskrɪpt/), often abbreviated as JS, is a programming
language that is one of the
                    core technologies of the World Wide Web, alongside HTML and CSS. As of
2022, 98% of websites use
                    JavaScript on the client side for webpage behavior, often incorporating
third-party libraries. All
                    major web browsers have a dedicated JavaScript engine to execute the
code on users' devices.
                </p>
                <p>
                    JavaScript is a high-level, often just-in-time compiled language that
conforms to the ECMAScript
                    standard.[10] It has dynamic typing, prototype-based object-
orientation, and first-class functions.
                    It is multi-paradigm, supporting event-driven, functional, and
imperative programming styles. It has
                    application programming interfaces (APIs) for working with text, dates,
regular expressions,
                    standard data structures, and the Document Object Model (DOM).
                </p>
                <p>
                    The ECMAScript standard does not include any input/output (I/O), such
as networking, storage, or
                    graphics facilities. In practice, the web browser or other runtime
system provides JavaScript APIs
```

```
                    for I/O.
                </p>
                <p>
                    JavaScript engines were originally used only in web browsers, but are
now core components of some
                    servers and a variety of applications. The most popular runtime system
for this usage is Node.js.
                </p>
                <p>
                    Although Java and JavaScript are similar in name, syntax, and
respective standard libraries, the two
                    languages are distinct and differ greatly in design.
                </p>
            </div>
        </section>
    </div>

    <script src="./js/task_10.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid royalblue
  margin: 10px 0
  max-width: 60vw
  display: flex
  align-items: center
  height: 700px

.button_section__aside
  background-color: royalblue
  height: 100%
  max-width: 30vw
  display: flex
  flex-direction: column
  align-items: center
  justify-content: center

.aside__HTML, .aside__CSS, .aside__JS
  height: 75px
  width: 100%
  font-size: 24px
  display: flex
  align-items: center
  justify-content: center
  &:hover
    background-color: #6389df
    cursor: pointer

.button_section__div_HTML_text, .button_section__div_CSS_text, .button_section__div_JS_text
  background-color: bisque
  height: 100%
  padding: 15px
  display: flex
  flex-direction: column
  justify-content: center
```

```
  p
    margin: 10px 0
```

- ***JS:***

```
'use strict';

const chooseSelector = (chosenElem) => chosenElem.style.backgroundColor = '#6389df';

const unChooseSelector = (chosenElem) => chosenElem.style.backgroundColor = '#4169E1FF';

const hiddenSelector = (selector) => selector.style.display = 'none';
// const hiddenSelector = (selector) => selector.setAttribute('hidden', '');

const unhiddenSelector = (selector) => selector.style.display = 'flex';
// const unhiddenSelector = (selector) => selector.removeAttribute('hidden');

const divHTMLText = document.querySelector('#divHTMLText');
const divCSSText = document.querySelector('#divCSSText');
const divJSText = document.querySelector('#divJSText');

const asideHTML = document.querySelector('#asideHTML');
const asideCSS = document.querySelector('#asideCSS');
const asideJS = document.querySelector('#asideJS');

chooseSelector(asideHTML);
hiddenSelector(divCSSText);
hiddenSelector(divJSText);

asideHTML.addEventListener('click', () => {
    chooseSelector(asideHTML);
    unChooseSelector(asideCSS);
    unChooseSelector(asideJS);

    unhiddenSelector(divHTMLText);
    hiddenSelector(divCSSText);
    hiddenSelector(divJSText);
});

asideCSS.addEventListener('click', () => {
    chooseSelector(asideCSS);
    unChooseSelector(asideHTML);
    unChooseSelector(asideJS);

    unhiddenSelector(divCSSText);
    hiddenSelector(divHTMLText);
    hiddenSelector(divJSText);
});

asideJS.addEventListener('click', () => {
    chooseSelector(asideJS);
    unChooseSelector(asideHTML);
    unChooseSelector(asideCSS);

    unhiddenSelector(divJSText);
    hiddenSelector(divHTMLText);
    hiddenSelector(divCSSText);
});
```

Результат програми:

HTML

CSS

JS

HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as "img" and "input" directly introduce content into the page. Other tags such as "p" surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.[2] A form of HTML, known as HTML5, is used to display video and audio, primarily using the "canvas" element, in collaboration with javascript.

HTML

CSS

JS

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML).[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.[2]

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.[4]

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.[5]

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

HTML

CSS

JS

JS

JavaScript (/ˈdʒɑːvəskrɪpt/), often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[10] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

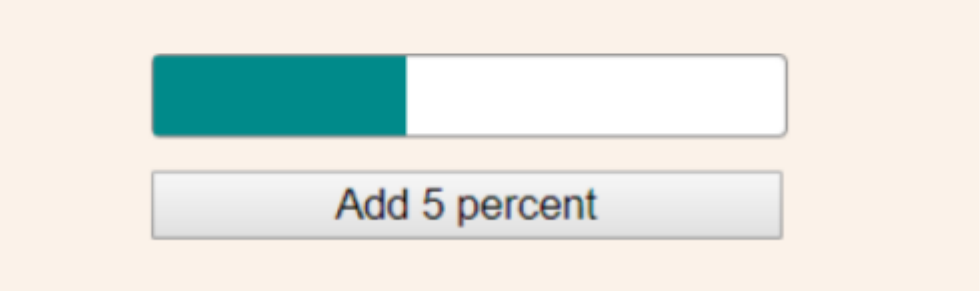
The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

Завдання 11:

Створити html сторінку з прогресбаром и кнопкою яка змінює його заповненість.



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 11</title>
    <link rel="stylesheet" href="./styles/css/task_11.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <progress class="button_section__progress_bar" id="progressBar" max="100"
value="5">5%</progress>
            <div class="button_section__buttons">
                <input class="button_section__button_add" id="buttonAddProgressBar"
```

```
type="button" value="Add 5 percents">
                <input class="button_section__button_remove" id="buttonRemoveProgressBar"
type="button" value="Remove 5 percents">
            </div>
        </section>
    </div>

    <script src="./js/task_11.js"></script>
</div>
</body>
</html>
```

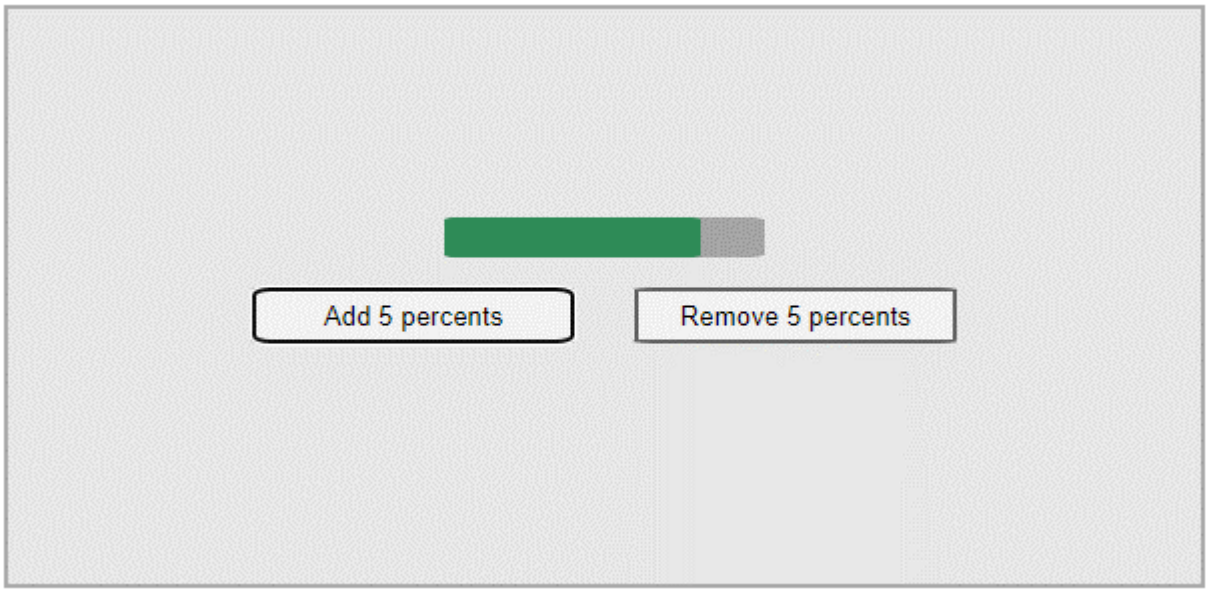
- ***SASS:***

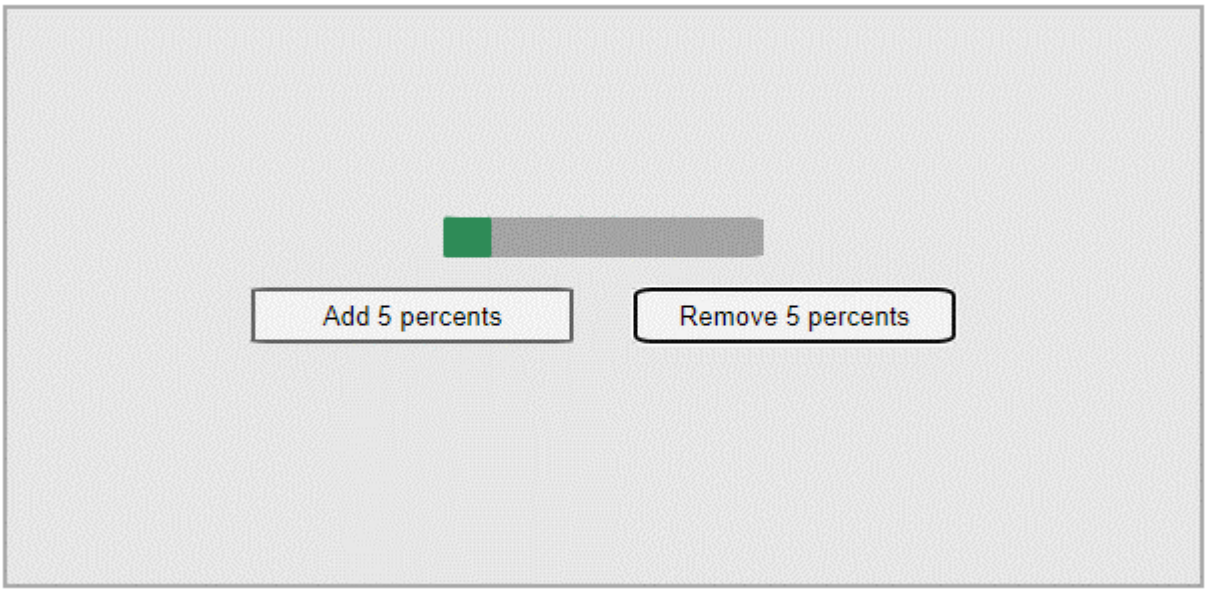
```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 2px solid darkgrey
  background-color: #e8e8e8
  margin: 10px 0
  max-width: 60vw
  display: flex
  flex-direction: column
  justify-content: center
  align-items: center
  width: 600px
  height: 30vh

.button_section__progress_bar
  height: 20px
  background-color: white

  &::-webkit-progress-bar
    background-color: #a5a5a5
    border-radius: 5%

  &::-webkit-progress-value
    background-color: seagreen
    border-radius: 5%

  &[value]::-webkit-progress-value::before
    content: '80%'
    position: absolute
    right: 0
    top: -125%

.button_section__buttons
  width: 90%
  display: flex
  justify-content: center


.button_section__button_add, .button_section__button_remove
  margin: 15px
  height: 28px
  width: 30%
  background-color: #f3f3f3
  border: 2px solid #626262
  border-radius: 5%
  cursor: pointer
```

- ***JS:***

```
'use strict';

const progressBar = document.querySelector('#progressBar');
const buttonAddProgressBar = document.querySelector('#buttonAddProgressBar');

buttonAddProgressBar.addEventListener('click', () => {
    progressBar.value += 5;
});

const buttonRemoveProgressBar = document.querySelector('#buttonRemoveProgressBar');

buttonRemoveProgressBar.addEventListener('click', () => {
    progressBar.value -= 5;
});
```

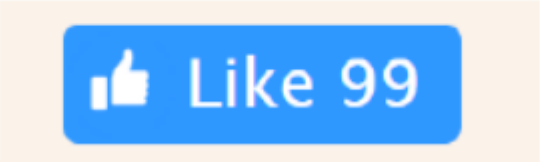
Результат програми:





Завдання 12:

Створити кнопку для соцмереж. Кожне натискання змінює к-сть реакцій.



Виконання:

- ***HTML:***

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Task 12</title>
    <link rel="stylesheet" href="./styles/css/task_12.css">
    <link rel="icon" href="../../../../img/icons/racoon_animal_icon_123571.ico">
</head>
<body>
<div class="wrapper">
    <div class="container">
        <section class="button_section">
            <button class="button_section__button_like" id="buttonLike" value="0">
                <img src="../../../../img/icons/racoon_animal_icon_123571.ico"
alt="raccoon_ico"
                     class="button_section__raccoon_ico">
                Like 0
            </button>
        </section>
    </div>

    <script src="./js/task_12.js"></script>
</div>
</body>
</html>
```

- ***SASS:***

```
*
  margin: 0
  padding: 0
  box-sizing: border-box

.wrapper
  width: 100%
  height: 100vh
  display: flex
  align-items: center
  flex-direction: column

.button_section
  border: 1.5px solid darkgrey
  background-color: #e8e8e8
  margin: 10px 0
  max-width: 20vw
  display: flex
  flex-direction: column
  justify-content: center
  align-items: center
  width: 250px
  height: 10vh

.button_section__button_like
```

```
  width: 150px
  height: 50px
  font-size: 20px
  color: white
  background-color: #c1c1c1
  cursor: pointer
  border: 2px solid darkgrey
  border-radius: 10px

  &:hover
    background-color: #a2a2a2

  display: flex
  justify-content: space-evenly
  align-items: center

.button_section__raccoon_ico
  width: 40px
  height: 40px
```

- ***JS:***

```
'use strict';

const buttonLike = document.querySelector('#buttonLike');

buttonLike.addEventListener('click', () => {
    buttonLike.value = `${parseInt(buttonLike.value) + 1}`;
    buttonLike.innerHTML = `<img src="../../../../img/icons/racoon_animal_icon_123571.ico"
alt="raccoon_ico"
                            class="button_section__raccoon_ico">
                            Like ${buttonLike.value}`;
});
```

Результат програми:

Репозиторій лабораторної роботи:

- https://github.com/AndriiBabushko/Frontend

Висновок: під час виконання лабораторної роботи було отримано навички обробки подій в мові JavaScript.