

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

### **Лабораторна робота №7**

з дисципліни "Основи клієнтської розробки"

Тема: "JavaScript. Події. Обробники подій. Спливання.  
Делегування подій"

Тематика: «Рецепти»

Виконав:

студент групи ІП-93

Домінський Валентин

Олексійович

Перевірила:

ст. вик. Кафедри ІСТ

Хмелюк Марина Сергіївна

Київ 2021

## Завдання:

1. Використати одну з подій миші. Написати функцію-обробник. Призначити функцію-обробник події через атрибут і через властивість.

Використати метод `addEventListener`, призначити одній події різні обробники (написати функції-обробники).

Призначити обробником події об'єкт за допомогою `addEventListener`, застосувати метод `handleEvent`, вивести елемент, на якому спрацював обробник, використовуючи `event.currentTarget`.

Видалити об'єкт, використовуючи `removeEventListener`

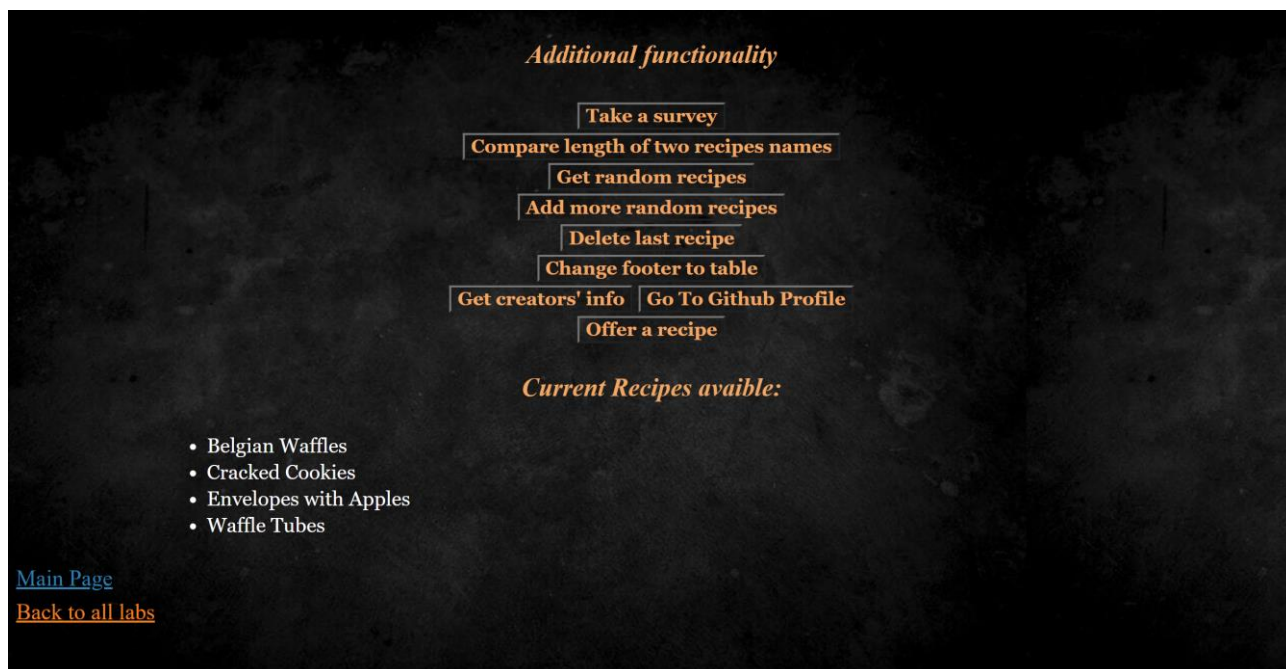
2. Створити список або використати існуючий. Реалізувати підсвічування елементів списку при кліку миші. Використовувати `event.target`. Обробник `onclick` застосувати для списку, а не для кожного елемента.

Створити меню (кілька кнопок), додати один обробник для всього меню і атрибуту `data-*` для кожної кнопки, в відповідності з методами, які вони викликають.

Застосувати прийом проектування «Поведінка» (додавання елементам поведінки `behavior` за допомогою атрибута `data-*`).

## Результати:

Додаткова сторінка, де використані всі методи, які є в завданні:



## Вихідний код:

### AdditionalInfo.html:

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

<head>
  <meta name="author" content="Valentyn Dominskyi">
  <meta name="keywords" content="recipes, cookbook">
  <meta name="description" content="cookbook site with recipes">
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Additional page</title>
</head>

<body>
  <header>
    <h3 id="AdditionalFunc">Additional functionality</h3>
  </header>

  <div class="AdditionalPage">
    <div class="AdditionalPageButtons">
      <button type="button" onclick="startDialog()">Take a survey</button>
      <br>
      <button type="button" id="LengthComparer">Compare length of two recipes
names</button>
      <br>
      <button type="button" onclick="insertRandomRecipes()">Get random
recipes</button>
      <br>
      <button type="button" id="AddRecipeButton">Add more random recipes</button>
      <br>
    </div>
  </div>
</body>
</html>
```

```
<button type="button" id="DeleteRecipeButton">Delete last recipe</button>
<br>
<button type="button" id="ChangeFooter">Change footer to table</button>
<br>
<div id="Menu">
  <button data-action="getInfo">Get creators' info</button>
  <button data-action="getGithub">Go To Github Profile</button>
</div>
<button data-toggle-id="offerRecipe">Offer a recipe</button>
<form id="offerRecipe" hidden>
  Recipe, which You want to see: <input type="text">
</form>
</div>
```

```
<h3>Current Recipes avaible:</h3>
<ul id="HighlightList">
  <li>Belgian Waffles</li>
  <li>Cracked Cookies</li>
  <li>Envelopes with Apples</li>
  <li>Waffle Tubes</li>
</ul>
```

```
<p class="RandomRecipe "></p>
<p class="RandomRecipe "></p>
<p class="RandomRecipe "></p>
</div>
<footer>
  <a href="index.html" target="_blank">Main Page</a>
  <br>
  <a href="../index.html">Back to all labs</a>
</footer>
```

```
<div>
  <script src="scripts/GetInfo.js"></script>
```

```

<script src="scripts/Dialog.js"></script>
<script src="scripts/GetRandomRecipe.js"></script>
<script src="scripts/ChangeRecipePlaceholder.js"></script>
<script src="scripts/ChangeFooter.js"></script>
<script src="scripts/LengthComparer.js"></script>
<script src="scripts/ListHighlight.js"></script>
<script src="scripts/Delegater.js"></script>
<script src="scripts/Behaviour.js"></script>
<script>
    let recipeButton = document.getElementById("AddRecipeButton");
    recipeButton.addEventListener("click", addRecipePlaceholder);
    recipeButton.addEventListener("click", insertRandomRecipes);

    let deleteRecipeButton = document.getElementById("DeleteRecipeButton");
    deleteRecipeButton.addEventListener("click", deleteRecipePlaceholder);

    highlightListElement();
</script>
</div>
</body>

</html>

```

## Style.css:

```

header {
    text-align: center;
}

```

```

body {
    font-size: 23px;
    color: #ffffff;
    font-style: normal;
    line-height: 1.3;
}

```

```
text-align: left;
background-image: url("images/Background.jpg");
}
```

```
h3 {
font-size: 25px;
color: #f3a866;
font-style: italic;
line-height: 1.5;
text-align: center;
}
```

```
p#ParagraphWithScale {
text-align: center;
block-size: 250px;
}
```

```
p#Frame {
text-align: right !important;
margin-top: -506.5px !important;
}
```

```
#ImageWithScale {
transform: scale(0.5) translate(0%, -50%) !important;
}
```

```
p.SidebarHeader {
font-size: 5;
color: orange;
font-family: "Lucida Console", sans-serif;
text-align: center;
}
```

```
ul,
```

```
ol,  
dl {  
    font-size: 20px;  
    color: #ffffff;  
    font-family: Georgia, serif;  
    margin: auto;  
    padding-left: 10%;  
}
```

```
ul,  
ol {  
    padding-left: 15%;  
}
```

```
ul.CircleType {  
    list-style-type: circle;  
}
```

```
ul.DiscType {  
    list-style-type: disc;  
}
```

```
ul.SquareType {  
    list-style-type: square;  
}
```

```
ol.UpperRomanType {  
    list-style-type: upper-roman;  
}
```

```
ol.LowerRomanType {  
    list-style-type: lower-roman;  
}
```



```
ol.AType {  
    list-style-type: upper-alpha;  
}
```

```
ol.DecimalType {  
    list-style-type: decimal;  
}
```

```
table {  
    font-size: x-large;  
    margin-left: auto;  
    margin-right: auto;  
    text-align: center;  
    padding: auto;  
    border: 5px solid #56018F;  
    vertical-align: middle;  
    color: rgb(255, 249, 240);  
}
```

```
th {  
    background-color: purple;  
}
```

```
tr {  
    vertical-align: middle;  
    text-align: center;  
}
```

```
td {  
    border: 1.5px solid #6701ac;  
}
```

```
a {  
    font-size: 22px;
```

```
font-style: normal;
line-height: 1.5;
text-align: center;
}
```

```
a:link {
  color: #2E82B3;
}
```

```
a:visited {
  color: #F08221;
}
```

```
a:active {
  color: #C728F7;
}
```

```
div.RecipeDescription {
  padding-left: 10%;
  padding-right: 10%;
}
```

```
div.RecipeDescription>p:first-child:first-letter {
  font-size: 200%;
  border: white solid 1px;
  padding: 4px;
  margin: 4px;
}
```

```
div.Sidebar {
  float: right;
  width: 25%;
  height: auto;
  border: 2px solid white;
```

```
padding: 1%;  
margin: auto 2%;  
margin-bottom: 3%;  
}
```

```
div.AdditionalPageButtons {  
    text-align: center;  
}
```

```
div.AdditionalPageButtons button {  
    font-family: Georgia, serif;  
    background: transparent;  
    overflow: hidden;  
    font-weight: 800;  
    font-size: 1.15rem;  
    color: #f3a866;  
}
```

```
p#ThemeMode {  
    position: absolute;  
    top: 10px;  
    left: 1000px;  
}
```

```
.highlight {  
    background-image: linear-gradient(to right, #493240, #cf037e, #493240);  
    border-radius: 6px;  
    padding: 3px 6px;  
}
```

## Behaviour.js:

```
'use strict'
```

```
document.addEventListener('click', function(event) {  
    let id = event.target.dataset.toggleId;
```

```

    if (!id) {
        return;
    }

    let textArea = document.getElementById(id);
    textArea.hidden = !textArea.hidden;
});

```

## ChangeFooter.js:

```

'use strict'

const FOOTER_SELECTOR = "footer";

const FOOTER_STRING = '<a href="index.html" target="_blank">Main Page</a><br><a href="../index.html">Back to all labs</a>';

const TABLE_SELECTOR = "table";

const TABLE_STRING = '<table><tr><td><a href="index.html" target="_blank">Main Page</a></td></tr><tr><td><a href="../index.html">Back to all labs</a></td></tr></table>';

let changeFooterButton = document.getElementById("ChangeFooter");
changeFooterButton.addEventListener('click', {
    handleEvent(event) {
        changeFooterToTable();
        changeFooterButton.removeEventListener('click', changeFooterToTable)
        alert("You " + event.type + "ed on " + event.currentTarget.nodeName);
    }
});

function changeFooterToTable() {
    let footer = document.querySelector(FOOTER_SELECTOR);
    footer.outerHTML = TABLE_STRING;
}

function changeTableToFooter() {
    let table = document.querySelector(TABLE_SELECTOR);
    let footer = document.createElement(FOOTER_SELECTOR);
    footer.innerHTML = FOOTER_STRING;

```

```
    table.replaceWith(footer);  
}
```

## ChangeRecipePlaceholder.js:

```
'use strict'
```

```
const PARAGRAPH_TAG = "p";  
const PARAGRAPH_CLASS = "RandomRecipe";  
const DIV_SELECTOR = "div.AdditionalPage";  
const LAST_CHILD_SELECTOR = ":last-child";
```

```
function addRecipePlaceholder() {  
    let paragraph = getParagraph();  
    let newParagraph = createNewParagraph();  
  
    addParagraph(paragraph, newParagraph);  
}
```

```
function getParagraph() {  
    let paragraph = document.querySelector(RECIPE_SELECTOR + LAST_CHILD_SELECTOR);  
    if (!paragraph) {  
        let startDiv = findStartDiv();  
        addParagraphToDiv(startDiv);  
    }  
  
    return paragraph;  
}
```

```
function findStartDiv() {  
    let div = document.querySelector(DIV_SELECTOR);  
    return div;  
}
```

```
function addParagraphToDiv(div) {  
    let newParagraph = createNewParagraph();  
    div.append(newParagraph);  
}
```

```
}
```

```
function createNewParagraph() {  
    let newParagraph = document.createElement(PARAGRAPH_TAG);  
    newParagraph.className = PARAGRAPH_CLASS;  
  
    return newParagraph;  
}
```

```
function addParagraph(paragraph, newParagraph) {  
    paragraph.after(newParagraph);  
}
```

```
function deleteRecipePlaceholder() {  
    let paragraph = getParagraph();  
    paragraph.remove();  
}
```

## Delegater.js:

```
'use strict'
```

```
class Menu {  
    constructor(btn) {  
        this._btn = btn;  
        btn.onclick = this.onClick.bind(this);  
    }  
    getInfo() {  
        showInfo(SURNAME, NAME);  
    }  
    getGithub() {  
        goToGithub();  
    }  
    onClick(event) {  
        let action = event.target.dataset.action;  
        if (action) {  
            this[action]();  
        }  
    }  
}
```

```
    }  
};  
}
```

```
let menu = document.getElementById("Menu");
```

```
new Menu(menu);
```

## **Dialog.js:**

```
'use strict'
```

```
let questions = {  
  0: "How old are You?",  
  1: "Why do You attend Our site?",  
  2: "What do You like the most about this site?",  
  3: "How did you find out about Our site?",  
};
```

```
const START_SURVEY_MESSAGE = "This button will start a survey. Do You want to start?";  
const END_SURVEY_MESSAGE = "Thank You for Your answers!";  
const CANCEL_SURVEY_MESSAGE = "No problem. Do what You want";
```

```
function startDialog() {  
  let confirmation = getConfirmation();  
  
  if (confirmation) {  
    startSurvey();  
    return;  
  }  
  
  cancelSurvey();  
}
```

```
function getConfirmation() {  
  let confirmation = confirm(START_SURVEY_MESSAGE);  
  return confirmation;
```

```
}
```

```
function startSurvey() {  
    let answers = {};  
    for (let i = 0; i < 4; i++) {  
        answers[i] = prompt(questions[i]);  
    }  
  
    alert(END_SURVEY_MESSAGE);  
}
```

```
function cancelSurvey() {  
    alert(CANCEL_SURVEY_MESSAGE);  
}
```

## **GetInfo.js:**

```
'use strict'
```

```
const SURNAME = "Dominskyi";  
const NAME = "Valentyn";  
const DEFAULT_POSITION = "Student";  
const GITHUB_INVITATION = "Wanna see github profile?";  
const GITHUB_ACCOUNT = "https://github.com/VsIG-official";
```

```
function showInfo(surname, name, position = DEFAULT_POSITION) {  
    alert(`Surname - ${surname}, name - ${name} and position - ${position}`);  
}
```

```
function goToGithub() {  
    let confirmation = confirm(GITHUB_INVITATION);  
  
    if (confirmation) {  
        location.href = GITHUB_ACCOUNT;  
    }  
}
```



## GetRandomRecipe.js:

```
'use strict'
```

```
let links = {
```

```
  0: "Belgian-Waffles.html",
```

```
  1: "Cracked-Cookies.html",
```

```
  2: "Envelopes-With-Apples.html",
```

```
  3: "Waffle-Tubes.html",
```

```
}
```

```
const RECIPE_SELECTOR = "div > p.RandomRecipe";
```

```
function insertRandomRecipes() {
```

```
  let paragraphs = getAllParagraphs();
```

```
  let paragraphsNum = 0;
```

```
  paragraphs.forEach(paragraph => {
```

```
    paragraphsNum++;
```

```
    changeHTML(paragraph, paragraphsNum);
```

```
  });
```

```
}
```

```
function getAllParagraphs() {
```

```
  let paragraphs = document.querySelectorAll(RECIPE_SELECTOR);
```

```
  return paragraphs;
```

```
}
```

```
function getRandomRecipe() {
```

```
  let randomNumber = Math.round(Math.random() * 3);
```

```
  let randomRecipe = links[randomNumber];
```

```
  return randomRecipe;
```

```
}
```

```
function changeHTML(paragraph, paragraphsNum) {
```

```
  let randomRecipe = getRandomRecipe();
```

```

    let number = document.createTextNode(paragraphsNum);

    paragraph.innerHTML = `

```

## LengthComparer.js:

```

'use strict'

document.getElementById("LengthComparer").onclick = function() {
    compareTwoTexts()
};

function compareTwoTexts() {
    let texts;
    texts = getTwoTexts();

    if (!texts[0] || !texts[1]) {
        return;
    }

    if (isTextsChosen()) {
        let textToShow = compareLengthes(texts[0], texts[1]);

        showText(textToShow);
        return;
    }

    return;
}

function getTwoTexts() {
    let firstText = prompt("Write first recipes ' name", "");
    if (!firstText) {
        return;
    }
}

```

```

    let secondText = prompt("And for the second one", "");
    if (!secondText) {
        return;
    }

    return [firstText, secondText];
}

function isTextsChosen() {
    let isChosen = confirm("Are You sure, that You want to compare these strings?");
    return isChosen;
}

function compareLengthes(firstText, secondtext) {
    let text;
    if (firstText.length > secondtext.length) {
        text = firstText;
    } else {
        text = secondtext;
    }
    return text;
}

function showText(text) {
    alert(text);
}

```

## ListHighlight.js:

```

'use strict'

let selectedLi;

function highlightListElement() {
    let list = document.getElementById("HighlightList");

```

```
list.onclick = function(event) {  
    let li = event.target.closest('li');  
  
    if (!li) {  
        return;  
    }  
  
    if (!list.contains(li)) {  
        return;  
    }  
  
    highlightElement(li);  
};  
}  
  
function highlightElement(li) {  
    if (selectedLi) {  
        selectedLi.classList.remove('highlight');  
    }  
  
    selectedLi = li;  
    selectedLi.classList.add('highlight');  
}
```

### Висновки:

Я глибше познайомився з javascript, подіями. Навчився їх обробляти та делегувати.

Дізнався про класи та поведінки

### Посилання:

[Github](#)

