# Лабораторная работа 4 Добавление обработчиков событий на JavaScript

## Цель работы:

Научиться добавлять обработчики событий на элементы веб-страницы с использованием JavaScript, чтобы взаимодействовать с пользователем и обрабатывать действия на странице.

## Описание задания:

Создать простую веб-страницу с элементами, которые реагируют на события, такие как клик, наведение мыши, ввод текста и отправка формы. В ходе работы вы освоите добавление и использование основных событий JavaScript.

## Инструменты:

- Текстовый редактор (например, Visual Studio Code)  
- Браузер (Google Chrome, Firefox и т.д.)

## Ход работы:

### 1. Создание HTML-файла:

Создайте новый файл с расширением `.html`, например, `index.html`.  
Добавьте базовую структуру HTML:  
```html  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <title>Обработчики событий на JavaScript</title>  
 <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
 <!-- Контент страницы будет здесь -->  
 <script src="script.js"></script>  
</body>  
</html>  
```

### 2. Добавление элементов для работы с событиями:

Добавьте элементы, которые будут реагировать на события:  
```html  
<button id="clickButton">Кликни меня</button>  
<div id="hoverDiv">Наведи на меня</div>  
<input type="text" id="inputField" placeholder="Введите текст">  
<form id="myForm">  
 <input type="text" placeholder="Введите что-нибудь">  
 <button type="submit">Отправить</button>  
</form>  
<p id="output"></p>  
```

### 3. Создание JavaScript-файла:

Создайте файл `script.js` в той же папке, что и `index.html`.

### 4. Добавление обработчика события для клика:

В `script.js` добавьте код для обработки события `click`, который изменит текст кнопки после нажатия:  
```javascript  
document.getElementById("clickButton").addEventListener("click", function() {  
 this.textContent = "Спасибо за клик!";  
});  
```

### 5. Добавление обработчика события для наведения мыши:

Добавьте обработчики событий `mouseover` и `mouseout` для изменения стиля элемента при наведении:  
```javascript  
const hoverDiv = document.getElementById("hoverDiv");  
  
hoverDiv.addEventListener("mouseover", function() {  
 hoverDiv.style.backgroundColor = "lightblue";  
});  
  
hoverDiv.addEventListener("mouseout", function() {  
 hoverDiv.style.backgroundColor = "";  
});  
```

### 6. Добавление обработчика события для ввода текста:

Используйте событие `input` для отслеживания ввода текста и отображения его в реальном времени:  
```javascript  
const inputField = document.getElementById("inputField");  
const output = document.getElementById("output");  
  
inputField.addEventListener("input", function() {  
 output.textContent = "Вы ввели: " + inputField.value;  
});  
```

### 7. Добавление обработчика события для отправки формы:

Используйте событие `submit` для предотвращения стандартного действия отправки формы и вывода сообщения:  
```javascript  
document.getElementById("myForm").addEventListener("submit", function(event) {  
 event.preventDefault();  
 alert("Форма отправлена!");  
});  
```

### 8. Проверка работы:

Откройте `index.html` в браузере и проверьте работу всех событий:  
- Клик по кнопке должен изменить ее текст.  
- Наведение на `div` должно изменить его цвет.  
- Ввод текста в поле должен отображаться в `p` элементе.  
- Отправка формы должна показывать сообщение.

## Контрольные вопросы:

- Что такое событие в JavaScript и какие типы событий существуют?

- Как добавить обработчик события на элемент?

- Чем отличается addEventListener от атрибутов HTML, таких как onclick?

- Что делает метод preventDefault() и в каких случаях его полезно использовать?

## Дополнительное задание:

Добавьте обработчики событий для двойного клика (dblclick) и для нажатия клавиши (keydown), чтобы получить больше опыта в работе с событиями.