

Події. Частина 1

№ уроку: 4 **Курс:** JavaScript Базовий

Засоби навчання: Visual Studio Code
Web Browser

Огляд, мета та призначення уроку

Навчитися обробляти події DOM-елементів за допомогою JavaScript. Вивчити основні події.

Вивчивши матеріал цього заняття, учень зможе:

- Створювати обробники подій, використовуючи HTML-атрибути.
- Використовувати властивості у JavaScript-кодi для обробки подій.
- Працювати з методом `addEventListener` для встановлення подій.

Зміст уроку

1. Що таке подія.
2. Основні події.
3. Обробка подій через HTML-атрибути.
4. Обробка подій через властивості.
5. Обробка подій за допомогою `addEventListener`.
6. Об'єкт як обробник подій.

Резюме

- **Подія** – це сигнал, який повідомляє про настання певного стану певного об'єкта. Наприклад, подія натискання на кнопку, подія завантаження вмісту сторінки.
- **Подію можна обробити.** Для цього необхідно створити спеціальний блок коду – обробник події. У JavaScript обробником події є функція, яка встановлюється як значення для певної властивості або використовується як параметр в методі `addEventListener`.
- Події виникають у DOM-вузлах та деяких об'єктах. Наприклад, можна обробити клік на кнопку та клік на елемент DIV.
- **Список основних подій:**
 - Click – натискання лівою кнопкою мишки.
 - Contextmenu – натискання правою кнопкою мишки.
 - Focus – елемент управління `input` отримає фокус (користувач клацнув по елементу).
 - Keydown – клавіша на клавіатурі затиснута.
 - Keyup – клавіша на клавіатурі відпущена.
 - Mouseover – курсор увійшов в периметр елемента.
 - Mouseout – курсор вийшов за межі елемента.
 - Mousedown – затиснута клавіша мишки.
 - Mouseup – відпущена клавіша мишки.
 - Mousemove – курсор переміщається по елементу.
 - Submit – користувач надіслав форму (натиснув на кнопку `submit` для `form`).
- Обробник може бути встановлений за допомогою атрибута в HTML-розмітці. Це один зі способів встановлення обробників, який варто уникати. Оскільки у такому випадку

Page | 1

відбувається змішування логіки представлення (HTML) та бізнес-логіки (JavaScript). Варто завжди прагнути до такої структури коду, щоб розмітка відокремлювалася від сценаріїв. Такий код легше супроводжувати та розширювати.

Атрибут, через який встановлюється обробник події, отримує назву з використанням приставки on та імені події. Наприклад onclick, onmousemove.

```
<div onclick="JavaScript код"></div>
```

- Щоб встановити обробник події через JavaScript-код, необхідно визначити функцію та додати її як значення для властивості з ім'ям події, яку необхідно обробити.

```
function myFunc() {  
  alert("Hello");  
}
```

```
div.onclick = myFunc;
```

Такий підхід має кілька недоліків. Один з них – на одну подію одного елемента можна додати лише один обробник. Під час додавання другого оброблювача перший обробник видаляється.

- Якщо в обробнику використовувати ключове слово this, можна отримати доступ до об'єкта, який ініціював подію. Наприклад, якщо в обробнику кнопки на подію click використовувати this, можна отримати доступ до самої кнопки, за якою виконали клік, отримати значення властивостей цієї кнопки. Це зручно використовувати тоді, коли той самий обробник використовується для різних елементів на сторінці.

- **Оптимальним способом встановлення обробника події є функція addEventListener.**

```
addEventListener(event, handler, options)
```

event – ім'я події, без приставки on (click, mousedown тощо).

handler – функція, що встановлюється як обробник.

options – додаткові параметри, які будуть розглянуті докладніше у наступних уроках.

У такий спосіб можна додати кілька обробників на одну подію того самого елемента сторінки. Також цей спосіб підтримує маршрутизацію подій, яка буде розглянута в окремому уроці.

- Щоб видалити обробник, який був встановлений через addEventListener, потрібно використовувати метод removeEventListener.

Закріплення матеріалу

- Що таке подія?
- Назвіть події, які пов'язані з клавіатурою.
- Назвіть події, які пов'язані з рухом курсора мишки.
- Як встановити обробник через розмітку?
- Які ви знаєте способи встановлення обробників через JavaScript-код?
- У чому перевага використання addEventListener?

Самостійна діяльність учня

Виконайте завдання у директорії Tasks\006 Events. Part 1. Текст завдань розташований у коментарях, у тегах script.

Рекомендовані ресурси

Вступ до подій

https://developer.mozilla.org/ru/docs/Learn/JavaScript/Building_blocks/Events

Оброблення подій

https://developer.mozilla.org/ru/docs/Web/Guide/Events/Event_handlers

Довідник з подій

<https://developer.mozilla.org/ru/docs/Web/Events>