

Події. Частина 3

№ уроку: 6 **Курс:** JavaScript Базовий

Засоби навчання: Visual Studio Code
Web Browser

Огляд, мета та призначення уроку

Вивчити фази поширення подій – занурення, на цілі, спливання. Навчитися використовувати фази подій для ефективної обробки подій елементів.

Вивчивши матеріал цього заняття, учень зможе:

- Використовувати перехоплення та спливання подій.
- Використовувати одну функцію для обробки подій кількох елементів.
- Зупиняти спливання подій.
- Зупиняти дію за замовчуванням, яка пов'язана з подією.

Зміст уроку

1. Фази проходу подій.
2. Реєстрація оброблювача події на різні фази.
3. Скасування спливання.
4. Скасування дії за замовчуванням.
5. Приклади використання.

Резюме

- **Фази проходу подій:**
 - **Capturing phase** – фаза занурення або фаза перехоплення. Подія йде згори донизу, від кореневого елемента до того, який ініціював подію.
 - **Target phase** – фаза цілі. Подія досягла елемента, через який подія виникла.
 - **Bubbling phase** – фаза спливання. Подія рухається від цільового елемента до кореневого.
- Фаза перехоплення використовується досить рідко. Переважно використовується фаза спливання, оскільки вона дає змогу зробити один обробник події на контейнері, у якому розташовані кілька елементів, що ініціюють подію. Цей підхід називається **делегуванням подій** – дочірні елементи делегують оброблення події батьківському елементу. Водночас один обробник на контейнері оброблятиме події від усіх вкладених елементів. Такий підхід дає змогу заощадити оперативну пам'ять, оскільки потрібен лише один обробник замість кількох. Також це спрощує розширення застосунку, оскільки з додаванням нових елементів немає потреби створювати для них обробники.
- Об'єкт події надає властивість **event.target**, яка містить найглибший елемент, що викликав подію (цільовий елемент).

- **Event.target** – цільовий елемент, у процесі проходження події його значення не змінюється. Значення `this` завжди містить поточний елемент, до якого дійшла подія. На етапі перехоплення та на етапі спливання значення `this` та `event.target` будуть відрізнятися. У фазі мети значення `this` і `event.target` будуть рівними.
- За замовчуванням всі обробники події реєструються на фазі спливання. Для реєстрації на фазі перехоплення необхідно додати в **addEventListener** третій параметр – `true`.
- Щоб припинити перехід події до наступного елемента, необхідно викликати метод **event.stopPropagation()**.
- Щоб скасувати дію за замовчуванням, яка пов'язана з подією, необхідно викликати метод **event.preventDefault()**. Наприклад, для події `submit` форми відбувається відправлення мережного запиту. Якщо в обробнику події буде викликаний метод `preventDefault()`, то відправлення запиту буде скасовано.
- Деякі події не спливають, наприклад, подія `focus` не підтримує цієї фази.

Закріплення матеріалу

- Назвіть фази переходів.
- Чим відрізняється властивість `event.target` від ключового слова `this` у функції обробника події?
- Як встановити обробник, який спрацює на фазі перехоплення?
- Як можна зупинити перехід події вузлами DOM-дерева?
- Як можна скасувати дію за замовчуванням, яка пов'язана з подією?
- Наведіть приклади використання перехоплення подій для розв'язку прикладного завдання.

Самостійна діяльність учня

Виконайте завдання у директорії `Exercises\Tasks\008 Events. Part 3`. Текст завдань розташований у коментарях, у тегах `script`.

Рекомендовані ресурси

Спливання та перехоплення подій

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

`event.stopPropagation()`

<https://developer.mozilla.org/ru/docs/Web/API/Event/stopPropagation>

Делегування подій

<https://davidwalsh.name/event-delegate>