

Події. Частина 2

№ уроку: 5 **Курс:** JavaScript Базовий

Засоби навчання: Visual Studio Code
Web Browser

Огляд, мета та призначення уроку

Розглянути приклади використання подій курсора мишки, клавіатури та подій життєвого циклу сторінки. Навчитися отримувати додаткову інформацію, яка пов'язана з подією.

Вивчивши матеріал цього заняття, учень зможе:

- Обробляти події мишки.
- Обробляти події натискання на клавіші клавіатури.
- Обробляти подію завантаження вмісту сторінки й інші події, які пов'язані зі сторінкою.
- Працювати з додатковою інформацією про подію, яка виникла.

Зміст уроку

1. Аргумент функції обробника події.
2. Події DOMContentLoaded, load.
3. Приклади обробки подій курсора мишки.
4. Приклади обробки подій клавіатури.

Резюме

- HTML-сторінка має три основні події:
 - **DOMContentLoaded** – браузер завантажив розмітку та побудував DOM-дерево, але зовнішні ресурси, наприклад картинки, можуть завантажуватися. Ця подія спрацьовує на об'єкті document.
 - **load** – HTML та зовнішні ресурси завантажені. Ця подія спрацьовує на об'єкті window.
- **Об'єкт події** – об'єкт, який створює браузер у момент виникнення події. Цей об'єкт передається як перший аргумент у функцію обробник події та може бути використаний для отримання додаткової інформації про подію. Наприклад, розташування курсора, затискання клавіші на клавіатурі чи мишки.
- Працюючи з подіями мишки, можна назвати дві категорії подій – **прості** та **комплексні**. Комплексні події будуються з урахуванням простих подій.
- **Прості події:**
 - mousedown – кнопка мишки натиснута;
 - mouseup – кнопка мишки відпущена;
 - mousemove – рух мишки над елементом;
 - mouseover – курсор з'явився над елементом;

- mouseout – курсор йде з елемента;
 - contextmenu – затиснута права кнопка мишки.
- **Комплексні події:**
 - click – mousedown та mouseup над цим же елементом;
 - dblclick – подвійний клік на елементі.
 - Під час обробки події, яка пов'язана з кліком, можна дізнатися клавiшу миші, що використовувалася для натискання. Для цього необхідно отримати значення властивості which об'єкта події, яке передавалося в метод обробника події. Значення 1 – затиснута ліва клавiша, 2 – середня клавiша (колесо), 3 – права клавiша мишки.
 - Також об'єкт події містить властивості, що визначають, які клавiші були затиснуті на клавiатурі під час події – **shiftKey**, **altKey**, **ctrlKey**, **metaKey** (Cmd для Mac).
 - Для отримання координат курсора використовують властивості **clientX**, **clientY**, **pageX** і **pageY**.

Client – координати щодо вікна.
Page – координати щодо документа (документ може скролитись, а частина документа перебувати за межами видимої частини вікна).
 - Якщо завдання реагувати на взаємодію користувача з клавiатурою (обробляти натискання на стрілочки, комбінації клавiш або певні клавiші), використовуються події **keydown** та **keyup**.
 - Під час обробки події в об'єкті події є дві властивості, які пов'язані з клавiшею, **code** і **key**.
Key – символ клавiші (змінюється відповідно до мови користувача).
Code – фізичний код клавiші.

Закріплення матеріалу

- У чому різниця між подіями load і DOMContentLoaded?
- Як можна попередити користувача, що він має намір залишити сторінку?
- Як дізнатися, яка клавiша мишки була затиснута під час обробки події click?
- Як дізнатися, чи затиснута клавiша shift під час обробки події click?
- У чому різниця між властивостями clientY та pageY?
- Як отримати інформацію про те, яка клавiша на клавiатурі була затиснута?

Самостійна діяльність учня

Виконайте завдання в директорії Exercises \ Tasks \ 007 Events Part 2. Текст завдань розташований у коментарях, у тегах script.

Рекомендовані ресурси

DOMContentLoaded

https://developer.mozilla.org/ru/docs/Web/API/Window/DOMContentLoaded_event

Подія load

https://developer.mozilla.org/ru/docs/Web/API/Window/load_event

Інтерфейс об'єкта подій мишки

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

Інтерфейс об'єкта подій клавіатури

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