

Перевірка даних форми

№ уроку: 13 Курс: JavaScript Базовий

Засоби навчання: Visual Studio Code
Web Browser

Огляд, мета та призначення уроку

Навчитися використовувати HTML5-атрибути та JavaScript-код для перевірки правильності даних, які введені у форму.

Вивчивши матеріал цього заняття, учень зможе:

- Використовувати події submit, input і change.
- Використовувати HTML5-атрибути для перевірки введених даних у форму.
- Локалізувати повідомлення, яке видає вбудований механізм перевірки даних.
- Навчитися базових принципів роботи з регулярними виразами.
- Використовувати користувацький код для перевірки введених даних у форму.

Зміст уроку

1. Подія submit.
2. Події blur і focus.
3. Події copy, paste, cut.
4. Використання вбудованої перевірки даних HTML5.
5. Додавання своїх правил перевірки даних під час використання вбудованої перевірки даних HTML5.
6. Використання регулярних виразів.
7. Використання користувацького коду для перевірки правильності введених даних.

Резюме

- Під час валідації форми використовується кілька подій:
 - **submit** – подія може бути оброблена на формі та спрацює в момент відправлення форми (натискання на кнопку submit у формі).
 - **focus** – елемент керування отримав фокус.
 - **blur** – елемент керування втратив фокус.
 - **input** – значення елемента керування змінилося.
 - **change** – значення було змінено, і елемент управління втратив форму.
 - **copy** – було виконано операцію копіювання.
 - **paste** – було виконано операцію вставлення.
 - **cut** – значення було вирізане з елемента управління.
- **Перевірка даних форми**, або **Валідація форми** – перевірка значень, введених в елементи управління форми на відповідність бізнес-правилам і виведення повідомлень

про помилку, якщо правила були порушені. **Приклади правил:** поле, обов'язкове для введення, у полі має бути значення email-адреси, поле має містити значення щонайменше 10 символів тощо.

- Валідація форми має обов'язково супроводжуватися помилками, які пов'язані з неправильно введеними даними.
- Валідація форми може проводитись:
 - На сервері.
 - На клієнті:
 - ✓ За допомогою HTML5-атрибутів.
 - ✓ За допомогою JavaScript-коду.
- **Перевірка форми** – обов'язкова дія, яка виконується на стороні клієнта, навіть якщо перевірка даних є на стороні сервера. Перевірка введених даних забезпечує правильність даних, захист користувацьких даних, захист серверної сторони. На стороні клієнта перевірка форми відіграє в роль, яка пов'язана зі зручністю використання, оскільки не потрібно чекати відповіді сервера, щоб вказати, де була допущена помилка. Валідація на стороні сервера необхідна, щоб забезпечити цілісність та безпеку даних.
- Під час використання вбудованої перевірки через атрибути HTML5 правила перевірки вказуються за допомогою спеціальних атрибутів. Сама перевірка проводиться в момент зміни значення поля введення та в момент відправлення даних на сервер подією submit.
- Під час використання вбудованої перевірки елемент управління отримує клас valid, якщо введене значення відповідає обмеженням, заданим через атрибути, і invalid, якщо значення порушує задані правила.
- Типи input, які підтримують перевірку введених значень:
 - **type="url"** – перевіряє значення на відповідність URL-адресі.
 - **type="email"** – перевіряє на відповідність адреси електронної пошти.
- **Атрибути валідації:**
 - **pattern** – значення має відповідати регулярному виразу;
 - **min** – значення має бути більшим за зазначене;
 - **max** – значення має бути меншим за зазначене;
 - **required** – значення обов'язкове;
 - **step** – значення має бути кратним кроку;
 - **maxlength** – кількість символів значення не має перевищувати вказану;
 - **minlength** – кількість символів має бути більшою за зазначену або дорівнювати їй.
- **setCustomValidity** – метод, який використовується для додавання користувацького повідомлення, що пов'язане з перевіркою правильності введених даних. Під час виклику цього методу на елементі керування значення, яке введене в елемент керування, вважається не валідним. А параметр, який передається в метод setCustomValidity, визначає текст повідомлення про помилку, яке буде показано користувачеві. Щоб усунути помилку з елемента управління, потрібно викликати метод setCustomValidity з переданням порожнього рядка як параметра.

Цей метод може використовуватися для додавання своїх правил до валідації форми через HTML5-атрибути та для локалізації повідомлень.

- **pattern** – атрибут, який використовується для перевірки введеного значення за допомогою регулярного виразу.
 - **a** – відповідає символу a;
 - **abc** – відповідає символу a, далі b, далі c;
 - **\d** – будь-яка цифра;
 - **\D** – будь-який символ, що не є цифрою;
 - **\w** – будь-який символ латинського алфавіту, цифри та символ нижнього підкреслення;
 - **\W** – всі символи, окрім символів латинського алфавіту, цифр і символ нижнього підкреслення;
 - **[]** – група символів, яка вказана у дужках, наприклад, **[abc]** відповідає символу a, або b, або c;
 - **[a-z]** – будь-який символ від a до z (у нижньому регістрі);
 - **[0-9]** – будь-який символ від 0 до 9;
 - **[0-9a-zA-Z]** – будь-який символ від 0 до 9 або будь-який символ латинського алфавіту в нижньому або верхньому регістрі.
- **Квантифікатори** – вказують кількість символів, яка має зустрітись у значенні, що перевіряється.
 - ***** – 0 або більше разів;
 - **+** – 1 або більше разів;
 - **?** – 1 або 0 разів;
 - **{n}** – n разів;
 - **{n,m}** – від n до m разів.

Приклади:

- **\d+** – відповідає рядку, який складається з одного і більше цифрових символів, наприклад, 1, 22, 123, 7777 тощо.
- **\d{2,4}** – відповідає рядку, який складається з двох, трьох або чотирьох цифрових символів, наприклад, 10, 88, 123, 5671 тощо.
- **[a-z]{3}\d{2}** – три будь-які символи латинського алфавіту та дві будь-які цифри, наприклад, abc12, rgb43.
- **abcdd** – символи abc і три цифри, наприклад, abc111, abc123 тощо.
- **https?** – символи http та 1 або 0 символів. s – http або https.

Закріплення матеріалу

- Які події можна використовувати для перевірки зміни даних у полі введення?
- Які є способи перевірки даних форми?
- Який атрибут потрібно використовувати для вказівки обмеження на введення лише електронних адрес в полі введення?
- Що робить метод `setCustomValidity`?

- Що таке регулярне вираження?
- Які HTML5-атрибути підтримують перевірку через регулярні вирази?

Самостійна діяльність учня

Виконайте завдання у директорії Tasks\016 Form Validation. Текст завдань розташований у коментарях, у тегах script.

Рекомендовані ресурси

Constraint Validation. HTML5-атрибути для валідації форм

https://developer.mozilla.org/ru/docs/Web/Guide/HTML/HTML5/Constraint_validation

Регулярні вирази

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Regular_Expressions

Перевірка даних форми

https://developer.mozilla.org/ru/docs/Learn/Forms/Form_validation

