

Використання форм (HTML-ФОРМИ)

Форма HTML являє собою документ, створений з використанням елементів HTML. Призначенням форми є збір інформації від користувачів. Після того як користувач заповнить форму і запускає процес її обробки, інформація з неї потрапляє в програму, що працює на сервері. Інша програма під назвою Common Gateway Interface (CGI) обробляє її. Таким чином користувач може інтерактивно взаємодіяти з сервером Web через Internet.

Форми так само зручні і для розробників сайту при розробці CMS (система управління вмістом (англ. Content Management System)), яка дозволяє підтримувати головну властивість сайту - актуальність

Створення простої форми

Теги `<form>` і `</ form>` задають початок і кінець форми. Починаючий форму тег `<form>` містить два атрибути: `action` і `method`. Атрибут `action` містить адресу URL сценарію, який має бути викликаний для обробки форми. Атрибут `method` вказує браузеру, який вид HTTP запити необхідно використовувати для відправки форми; можливі значення `POST` і `GET`.

Головна відмінність методів `POST` і `GET` полягає у способі передачі інформації. У методі `GET` параметри передаються через адресний рядок, тобто по суті в HTTP-заголовку запити, в той час як у методі `POST` параметри передаються через тіло HTTP-запити і ніяк не відбиваються на вигляді рядка.

Для відправки форми на сервер використовується кнопка типу `submit`, також форму можна відправити натиснувши клавішу `Enter` в межах форми. Якщо кнопка типу `submit` відсутній, клавіша `Enter` імітує її використання.

Коли форма відправляється на сервер для обробки даних вказаним обробником, що знаходиться по адресі, заданим атрибутом `action` елементу `<form>` Дані відправляються у вигляді пари «ім'я=значення», де ім'я визначається атрибутом `name` (Задати ім'я можна наступним елементам `input`, `textarea`, `button` чи `select`, а значення буде, тим, що користувач ввів в поле форми.

Для тега `form` можна використовувати псевдо-класи CSS `:valid`, `:invalid`.

Функція html-форми - обмін даними між користувачем та сервером або зв'язку між користувачем і адміністратором.

Тег `<form>` може містити один або більше з наступних елементів форми:

- `<input>`
- `<textarea>`
- `<button>`
- `<select>`
- `<option>`

- <optgroup>
- <fieldset>
- <label>

Якщо форму вкласти в іншу форму, то будуть виконані умови та відправлені дані зовнішньої форми.

На одній сторінці допускається використання будь-якої кількості форм, але необхідно пам'ятати, що у разі надсилання даних будуть відправлені дані тільки однієї форми.

Приклад використання HTML тега <form> для зворотнього зв'язку.

```
<p>Приклад форми для входу</p>
<hr>
<form>
  <label>Імя:</label><br>
  <input type="text" name="login"><br>

  <label>Email:</label><br>
  <input type="email" name=""><br>

  <label>Ваш сайт:</label><br>
  <input type="url"><br>

  <label>Ваше повідомлення:</label><br>
  <textarea></textarea>
  <br>

  <input type="submit" value="Увійти">
</form>
```

Приклад форми для входу

Імя:

Email:

Ваш сайт:

Ваше повідомлення:

Увійти

Різниця між HTML 4.01 та HTML5

У HTML5 додано два нових атрибута: `autocomplete` та `novalidate`, і видалено атрибут `accept`.

Різниця між XHTML та HTML У XHTML, атрибут `name` не рекомендується до використання. Замість цього використовуйте загальний атрибут `id`.

Валідація

Валідація (англ. Validation) — проєкт підтвердження відповідності (обґрунтованості) або надання законної сили; набір процедур або дій, спрямованих на підтвердження за допомогою об'єктивних свідчень, математичних методів і вимірювань, наведення доказів, що вимоги до процесу, проєкту або системи виконано і цей процес, проєкт або система відповідають заданим параметрам.

Валідація в програмуванні — процес, що дозволяє визначити, наскільки точно з позицій потенційного користувача певна модель представляє задані сутності реального світу.

Коли ви збираєте інформацію від людей, через форму, ви застосовуєте якісь перевірки. Недотримання цих вимог може призвести до втрати клієнтів, небажаних даних в вашій базі даних або навіть слабких місць в системі безпеки вашого сайту. Історично, створення перевірок було болісним процесом. На стороні сервера, процес полегшується завдяки фреймворкам, які обробляють все за вас, але на стороні клієнта, ви, як правило, користуєтесь JavaScript-бібліотеками, які доволі складно інтегрувати.

На щастя, HTML5 дає нам ряд можливостей, завдяки яким можна виконувати більшість перевірок. Форми в HTML5 тепер мають вбудовану підтримку для перевірки за допомогою

використання спеціальних атрибутів і нових типів введення, і надають вам широкий контроль над CSS-стилями.

Спеціалізовані типи вводу

HTML5 вводить кілька нових типів введення. Вони можуть бути використані для створення полів вводу, які будуть приймати тільки певний тип даних.

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Щоб скористуватися новими типами, включіть їх в якості значення атрибута `type`:

```
<input type="email">
```

Якщо браузер не підтримує цей тип вводу, це поле буде вести себе, як звичайне поле для введення тексту. Також корисно знати, що деякі поля (наприклад, `email` і `tel`) викликають відкриття спеціалізованої клавіатури у мобільних пристроїв.

Обов'язкові Поля

Просто додавши "необхідний атрибут до `<input>`, `<select>`, `</select>` `<textarea>`, ви повідомляєте браузеру, які значення повинні бути в цій області. Це як червона зірочка (*), яку ми бачимо в більшості реєстраційних форм.

```
<input type="checkbox" name="terms" required >
```

Проблема тут в тому, що майже всі дані будуть виконувати цю вимогу - наприклад, ви можете обійти перевірку залишивши порожнє місце (ми покажемо вам, як запобігти цьому).

Коли ви встановлюєте потрібний атрибут для електронної пошти або поля `url`, браузер очікує певний шаблон, якому потрібно слідувати, але все це є доволі відносним, адреса пошти "a@aa" буде вважатися дійсною (читайте далі, щоб побачити, як обійти це).

Обмеження

Ми можемо встановити деякі основні обмеження, такі як максимальна довжина, мінімальні і максимальні значення для числових полів. Щоб обмежити довжину поля введення і textareas, використовуйте атрибут `maxlength`. Це означає, що забороняється введення значення, яке є більшим зазначеного в `maxlength`. Якщо користувач намагатиметься вставити з буферу рядок більший зазначеного, то форма просто обріже його.

```
<input type="text" name="name" required maxlength="15">
```

Поле `<input type="number">` використовує `max` та `min` атрибути, щоб створити діапазон можливих значень (в нашому прикладі ми зробили мінімальний допустимий вік 18).

```
<input type="number" name="age" min="18" required>
```

Джерела

- [Ніжинський державний університет імені Миколи Гоголя](#)
- [<form> HTML тег](#)
- [🗉 Валідація](#)

From:
<https://library.vpuhluhiv.com.ua/> - **Wiki Глухівського ВПУ**

Permanent link:
https://library.vpuhluhiv.com.ua/subjects:basic:informatika:web:vykorystannia_form_osnovni_elementy_formy?rev=1672355524

Last update: **30.12.2022 01:12**

