

Лабораторная работа №7.

Тема: Разработка формы обратной связи с валидацией на JavaScript.

Цель работы: изучить клиентскую валидацию форм, используя HTML и JavaScript.

Основные термины и понятия:

Форма — это компонент веб-страницы с элементами управления, такими как текстовые поля, кнопки, флажки, диапазон или поле выбора цвета.

Формы являются неотъемлемой частью веб-разработки, позволяя пользователям взаимодействовать с веб-сайтами. Они используются для сбора данных, отправки информации на сервер и выполнения других интерактивных задач.

Форма в HTML начинается с тега `<form>`, который определяет область, в которой будут находиться все элементы формы. Основные элементы формы включают:

- **Тег `<form>`:** контейнер для всех элементов формы.
- **Тег `<input>`:** используется для создания различных типов полей ввода.
- **Тег `<label>`:** связывает текстовую метку с элементом формы.
- **Тег `<textarea>`:** создает многострочное текстовое поле.
- **Тег `<button>`:** создает кнопку для отправки формы или выполнения других действий.
- **Тег `<select>` и `<option>`:** создают выпадающий список.

Тег `<form>` является контейнером для всех элементов формы и определяет область, в которой будут находиться все элементы формы. Он также используется для задания метода отправки данных и URL-адреса, на который будут отправляться данные формы. Тег `<input>` является одним из самых универсальных элементов формы и может использоваться для создания различных типов полей ввода, таких как текстовые поля, поля для ввода пароля, электронных адресов, числовых значений и других. Тег `<label>` связывает текстовую метку с элементом формы, что улучшает доступность и удобство использования формы. Тег `<textarea>` используется для создания многострочных текстовых полей, а тег `<button>` создает кнопку для отправки формы или выполнения других действий. Теги `<select>` и `<option>` используются для создания выпадающих списков, которые позволяют пользователям выбирать один или несколько вариантов из предложенного списка.

Поле ввода `<input>` может иметь различные типы, которые определяются атрибутом `type`. Вот некоторые из наиболее распространенных типов:

- **text:** однострочное текстовое поле.
- **password:** поле для ввода пароля, символы в котором скрыты.
- **email:** поле для ввода адреса электронной почты.
- **number:** поле для ввода числовых значений.
- **checkbox:** флажок для выбора одного или нескольких вариантов.
- **radio:** переключатель для выбора одного из нескольких вариантов.
- **submit:** кнопка для отправки формы.

```

<form>
  <label for="username">Имя пользователя:</label>
  <input type="text" id="username" name="username">

  <label for="password">Пароль:</label>
  <input type="password" id="password" name="password">

  <label for="email">Электронная почта:</label>
  <input type="email" id="email" name="email">

  <label for="age">Возраст:</label>
  <input type="number" id="age" name="age">

  <label for="newsletter">Подписаться на рассылку:</label>
  <input type="checkbox" id="newsletter" name="newsletter">

  <label for="gender">Пол:</label>
  <input type="radio" id="male" name="gender" value="male">
  <label for="male">Мужской</label>
  <input type="radio" id="female" name="gender" value="female">
  <label for="female">Женский</label>

  <button type="submit">Отправить</button>
</form>

```

Имя пользователя: Пароль:

Электронная почта: Возраст:

Подписаться на рассылку: ☐ Пол: ☐ Мужской ☐ Женский

Каждый тип поля ввода имеет свои особенности и используется в зависимости от потребностей формы. Поле типа `text` используется для ввода однострочного текста, такого как имя пользователя или адрес. Поле типа `password` предназначено для ввода паролей и скрывает введенные символы. Поле типа `email` проверяет, что введенное значение соответствует формату адреса электронной почты. Поле типа `number` позволяет вводить только числовые значения и может включать дополнительные атрибуты, такие как `min`, `max` и `step` для ограничения диапазона и шага значений. Поля типа `checkbox` и `radio` используются для выбора одного или нескольких вариантов из предложенного списка. Кнопка типа `submit` отправляет данные формы на сервер.

Для предупреждения и проверки некорректного ввода в HTML существует **механизм валидации**.

Преимущество использования валидации в HTML заключается в том, что пользователь после некорректного ввода может сразу получить сообщение об ошибке и внести соответствующие изменения в введенные данные.

Для создания валидации у элементов форм HTML используется ряд атрибутов:

- **required**: требует обязательного ввода значения. Для элементов `textarea`, `select`, `input` (с типом `text`, `password`, `checkbox`, `radio`, `file`, `datetime-local`, `date`, `month`, `time`, `week`, `number`, `email`, `url`, `search`, `tel`)

- **min** и **max**: минимально и максимально допустимые значения. Для элемента `input` с типом `datetime-local`, `date`, `month`, `time`, `week`, `number`, `range`

– **pattern**: задает шаблон, которому должны соответствовать вводимые данные. Для элемента input с типом text, password, email, url, search, tel

Атрибут **required** требует обязательного наличия значения:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Валидация в HTML</title>
  </head>
  <body>
    <form method="get">
      <p>
        <label for="login">Логин:</label>
        <input type="text" required id="login" name="login" />
      </p>
      <p>
        <label for="password">Пароль:</label>
        <input type="password" required id="password" name="password" />
      </p>
      <p>
        <input type="submit" value="Отправить" />
      </p>
    </form>
  </body>
</html>
```

Логин:

Пароль:

Если не ввести в эти поля никаких данных, оставив их пустыми, и нажать на кнопку отправки, то браузер высветит сообщения об ошибке, а данные не будут отправлены на сервер:

Логин:

Пароль:  Заполните это поле.

В зависимости от браузера визуализация сообщения может несколько отличаться. Также границы некорректного поля ввода могут окрашиваться в красный цвет.

Атрибуты max и min

Для ограничения диапазона вводимых значений применяются атрибуты max и min:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Валидация в HTML5</title>
  </head>
  <body>
    <form method="get">
      <p>
        <label for="age">Возраст:</label>
        <input type="number" min="1" max="100" value="18" id="age" name="age"/>
      </p>
      <p>
        <input type="submit" value="Отправить" />
      </p>
    </form>
  </body>
</html>

```

Возраст:

Атрибут pattern

Атрибут pattern задает шаблон, которому должны соответствовать данные. Для определения шаблона используется язык так называемых регулярных выражений. Рассмотрим самый простейший пример:

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Валидация в HTML5</title>
  </head>
  <body>
    <form method="get">
      <p>
        <label for="phone">Телефон:</label>
        <input type="text" placeholder="+1-234-567-8901"
pattern="\+\d-\d{3}-\d{3}-\d{4}" id="phone" name="phone" />
      </p>
      <p>
        <input type="submit" value="Отправить" />
      </p>
    </form>
  </body>
</html>

```

Телефон:

Не всегда валидация является желаемой, иногда требуется ее отключить. И в этом случае можно использовать либо у элемента формы атрибут **novalidate**, либо у кнопки отправки атрибут **formnovalidate**.

Чтобы запросить подтверждение перед отправкой формы, добавьте обработчик событию **onsubmit** и воспользуйтесь диалоговым окном **confirm**.

Задание:

- Создайте форму обратной связи с полями ввода: имя, email, сообщение.
- Реализуйте проверку на минимальную длину текста и корректность email.
- Выводите сообщения об ошибках под соответствующими полями.
- Добавьте возможность сброса формы с подтверждением действия через `confirm()`.

Дополнительно: реализуйте валидацию формы на нескольких этапах (пошаговая валидация).

Ход выполнения лабораторной работы должен быть отражен в *отчете*. Отчет должен содержать титульный лист, цель работы, задание, листинг исходного кода, описание проделанной работы, скриншоты результата, вывод, ответы на контрольные вопросы.

Контрольные вопросы:

1. Что такое формы? Для чего они нужны?
2. Перечислите основные элементы формы?
3. Перечислите типы полей, которые могут быть указаны в теге `<input>`.
4. Что такое валидация?
5. Какое преимущество использования валидации в HTML?
6. Опишите атрибут `required` для создания валидации.
7. Опишите атрибуты `max` и `min` для создания валидации.
8. Опишите атрибут `pattern` для создания валидации.
9. Что такое регулярное выражение?
10. Как отключить валидацию?
11. Что необходимо сделать, чтобы запросить подтверждение перед отправкой формы?