

Разработка валидации для форм ввода данных на стороне клиента

Круглик Роман Игоревич

Приамурский государственный университет им. Шолом-Алейхема

Студент

Брыкова Анастасия Леонидовна

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

В статье разобрана система проверка ввода данных на стороне клиента. Были приведены наиболее популярные проверки.

Ключевые слова: JavaScript, регистрация, формы ввода.

Development of validation for data entry forms on the client side

Kruglik Roman Igorevich

Sholom-Aleichem Priamursky State University

Student

Brykova Anastasia Leonidovna

Sholom-Aleichem Priamursky State University

Student

Abstract

In article, the system checks the input of data on the client side. Was given the famous check.

Keywords: JavaScript, registration, entry forms.

Валидация - процесс приведения доказательств того, что требования конкретного пользователя продукта, услуги или системы удовлетворены.

Первой пришла серверная валидация. Затем она развилась в валидацию на стороне клиента для проверки результатов в браузере. Базовый UI модуль CSS3 предоставляет несколько псевдо-классов, которые помогают стилизовать состояние валидности и менять внешний вид поля в зависимости от действий пользователя. Чем больше предоставлено подсказок пользователю, как правильно заполнять форму в процессе заполнения, тем меньше шансов, что он допустит ошибку.

Данная тема актуальна, так как используется многими разработчиками. В статье Сомова Е.В., Киселёв Е.В. [1] рассказывают о преимуществе клиентского языка при разработке сайта. На ресурсе [2] приводится статья о технике валидации форм. Metanit [3] рассматривают важной проверки

вводимых данных. В система Хабрахабр [4] опубликована статья о валидации форм на JavaScript.

Для начала необходимо создать формы для ввода данных для регистрации. Структура html документа будет выглядеть следующим образом (см.р.1).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Регистрация</title>
</head>
<body>
  <section id="registration">
    <div class="row">
      <div class="container">
        <center>
          <h2>Форма регистрации</h2>
          <form action="connect_db.php" method="POST">
            <input type="text" placeholder="Имя" name="name" class="name"><br>
            <input type="email" placeholder="Электронная почта" name="email" class="email"><br>
            <input type="text" placeholder="Логин" name="login" class="login"><br>
            <input type="password" placeholder="Пароль" name="password" class="password"><br>
            <input type="password" placeholder="Повторите пароль" name="passwordRep" class="passwordRep"><br>
            <input type="submit" value="Зарегистрироваться" class="btn_reg"><br>
          </form>
        </center>
      </div>
    </div>
  </section>
  <script src="js/validation_script.js?n=ffaf"></script>
</body>
</html>
```

Рисунок 1. Структура HTML документа

Можно перейти во вкладку и увидеть форму. (см.р.2).

Форма регистрации

Имя
E-mail
Логин
Пароль
Повторите пароль
Зарегистрироваться

Рисунок 2. Форма регистрации

Нужно реализовать несколько правил:

- 1) Все поля должны быть заполнены.
- 2) Имя должно включать не более 5 латинских символов.
- 3) Пароль состоит более чем из 5 символов.

Для начала все поля должны быть заполнены. Для этого нужен обработчик полей ввода (см.р.3).

```
var btn_reg = document.querySelector('.btn_reg');
var login = document.querySelector('.login');
var password = document.querySelector('.password');
var names = document.querySelector('.name');
var email = document.querySelector('.email');
var passwordRep = document.querySelector('.passwordRep');

btn_reg.addEventListener('click',function(e)
{
    var errors = document.querySelectorAll('.error')
    for (var i = 0; i < errors.length; i++) {
        errors[i].remove();
    }
    if (!names.value) {
        e.preventDefault();
        var error = document.createElement('div');
        error.className = 'error';
        error.innerHTML = 'Введите Имя';
        error.style.color = 'red';
        email.parentElement.insertBefore(error,email);
    }
})
```

Рисунок 3. Обработчик полей

При нажатии на кнопку регистрации срабатывает обработчик, который проверяет ввод данных. Если поле не заполнено, то выводится ошибка и браузер не обновляется. Все манипуляции происходят на стороне клиента и до того момента, пока условие не будет выполнено, переходы не произойдут. После повторного нажатия на кнопку удаляется ошибка и если поле опять не заполнено, то она возвращается. Далее нужно поставить условия для каждого поля. (см.р.4).

Форма регистрации

<input type="text"/>
Введите Имя
<input type="text"/>
Введите E-mail
<input type="text"/>
Введите Логин
<input type="text"/>
Введите пароль
<input type="text"/>
<input type="text"/>
<input type="button" value="Зарегистрироваться"/>

Рисунок 4. Ошибки при отправке формы

Пока не будут исправлены все ошибки форма не отправится. После следует правило для поля name, в котором должно быть не более 5 символов латинской раскладки. (см.р.5).

```
$('.name').on('keydown',function(e)
{
if ((names.value).match(/[а-я]/i) != null) {
var errors = document.querySelectorAll('.errorKir')
for (var i = 0; i < errors.length; i++) {
errors[i].remove();
}
}
var error = document.createElement('div');
error.className = 'errorKir';
error.innerHTML = 'Имя содержит недопустимые символы';
error.style.color = 'red';
email.parentElement.insertBefore(error,email);
}
if ((names.value).length>=5) {
var errors = document.querySelectorAll('.errorName')
for (var i = 0; i < errors.length; i++) {
errors[i].remove();
}
}
var error = document.createElement('div');
error.className = 'errorName';
error.innerHTML = 'Нельзя использовать больше 5 символов';
error.style.color = 'red';
email.parentElement.insertBefore(error,email);
}
})
```

Рисунок 5. Реализация второго правила

Сначала накладывается обработчик, который при вводе данных выполняет функцию. После идёт обработка ввода русских букв и проверка количества (не больше 5). При каждом следующем вводе символа ошибки устраняются. (см.р.6).

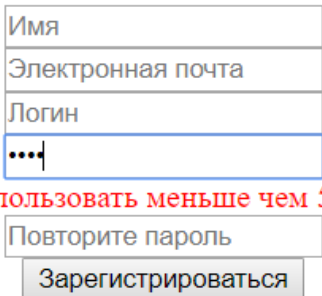
Форма регистрации

<input type="text" value="Валерия"/>
Имя содержит недопустимые символы
Нельзя использовать больше 5 символов
<input type="text" value="Электронная почта"/>
<input type="text" value="Логин"/>
<input type="text" value="Пароль"/>
<input type="text" value="Повторите пароль"/>
<input type="button" value="Зарегистрироваться"/>

Рисунок 6. Реализация второго правила

Далее пароль должен состоять более чем из 5 символов. Всё создаётся по аналогии (см.р.6).

Форма регистрации



The image shows a registration form with the following fields and elements:

- Имя
- Электронная почта
- Логин
- Пароль (masked with dots, containing 4 characters)
- Повторите пароль
- Зарегистрироваться

A red error message is displayed below the password field: **Нельзя использовать меньше чем 5 символов**.

Рисунок 7. Поле ввода пароля

Все ошибки появляются сразу при их возникновении, а исчезают последовательно при исправлении. Данную работу можно использовать для любых форм ввода информации в систему.

Библиографический список

1. Сомова Е.В., Киселёв Е.В. Javascript как незаменимая составляющая функционирования любого сайта//сборник статей III Международной научно-практической конференции: в 2 частях. 2017. С. 31-33.
2. HTML academy // URL:<https://htmlacademy.ru/blog/95-form-validation-techniques> (дата обращения: 23.01.2018).
3. Metanit // URL: <https://metanit.com/sharp/wpf/14.php> (дата обращения: 23.01.2018).
4. Валидация форм в JavaScript – Хабрахабр URL: <https://habrahabr.ru/post/112518/> (дата обращения: 23.01.2018).