

Лабораторная работа № 34

Тема работы: «Проверка данных форм при помощи библиотеки jQuery»

1. Цель работы

Формирование умений обрабатывать данные форм при помощи библиотеки jQuery

2. Задание

Создать HTML-документ, содержащий сценарии JavaScript.

3. Оснащение работы

ПК, текстовый редактор, браузер.

4. Основные теоретические сведения

Валидация форм и проверка их полей на соответствие определенному формату и наличие определенных данных — это фактически стандартная задача при работе с формами на сайте. Решить ее можно различными способами и средствами.

Можно воспользоваться новыми возможностями HTML5, которые позволяют задавать тип полям и даже использовать регулярные выражения для проверки введенных данных. Можно использовать один из многочисленных плагинов для валидации форм. Можно написать собственный скрипт для валидации.

В этой статье мы с вами воспользуемся вторым вариантом и используем возможности плагина jQuery Validation и поработаем со следующей формой, представленной на рисунке 34.1.



The image shows a simple HTML form. It consists of three text input fields stacked vertically. Each field has a label to its left: 'Имя *', 'Email *', and 'Сайт *'. The input fields contain the placeholder text 'Имя', 'Email', and 'Сайт' respectively. Below the input fields is a blue button with the text 'Отправить' in white.

Рисунок 34.1 – Форма HTML

Для работы с плагином его необходимо скачать и подключить. Подключается плагин после библиотеки jQuery:

```
<script src="jv/dist/jquery.validate.js"></script>
```

После подключения плагина его необходимо инициализировать, для этого достаточно вызвать метод `validate()` для формы, которую необходимо проверить:

```
$(function(){  
    $('#form').validate();  
});
```

Для плагина достаточно атрибута `required`, чтобы он не пропускал пустое поле. При этом плагин также отлично понимает типы полей. Например, если в

поле email мы введем строку, не соответствующую формату email, плагин уведомит нас, причем в режиме онлайн.

Для создания сообщения об ошибках передадим в метод `validate()` объект `messages` с текстом ошибки. Добавим необходимые сообщения:

```
$(function(){
$('#form').validate({
rules: {
name: {
required: true,
minlength: 2
},
},
messages: {
name: {
required: "Поле 'Имя' обязательно к заполнению",
minlength: "Введите не менее 2-х символов в поле 'Имя'"
},
email: {
required: "Поле 'Email' обязательно к заполнению",
email: "Необходим формат адреса email"
},
url: "Поле 'Сайт' обязательно к заполнению"
}
});
});
```

Обратите внимание, для каждого варианта валидации можно добавить свое сообщение. Также мы использовали объект `rules`, позволяющий добавить правила валидации для того или иного поля.

5. Порядок выполнения работы

В соответствии с вариантом 25 лабораторной работы создайте валидацию формы с помощью JQuery.

6. Форма отчета о работе

Лабораторная работа №

Номер учебной группы

Фамилия, инициалы учащегося

Дата выполнения работы _____

Тема работы: _____

Цель работы:

Оснащение работы:

Результат выполнения работы:

7. Контрольные вопросы

1. Опишите процесс подключения и работы плагина validate.
2. Каким образом в плагин validate можно добавить свои сообщения об ошибках?

8. Рекомендуемая литература

1. **Макфарланд, Д.** Новая большая книга CSS / Дэвид Макфарланд. – СПб.: Питер, 2016. – 720с.
 2. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 /Р. Никсон. – 4-е изд. – СПб.: ООО «ПИТЕР М», 2017. – 768 с.
 3. **Прохоренок, Н.А.** HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А.Прохоренок. – СПб.: БХВ-Петербург, 2010. – 912с.
 4. **Робсон, Э.** Изучаем HTML, XHTML и CSS / Э. Робсон. – 2-е изд. – СПб.: ООО «ПИТЕР М», 2017. – 720 с.
- Фрейн, Б.** HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Бен Фрейн. – СПб.: Питер Пресс, 2017. — 272с.