

Проверка данных формы

№ урока: 16 **Курс:** JavaScript Базовый

Средства обучения: Visual Studio Code
Web Browser

Обзор, цель и назначение урока

Научится использовать HTML5 атрибуты и JavaScript код для проверки правильности данных, введенных в форму.

Изучив материал данного занятия, учащийся сможет:

- Использовать события submit, input и change.
- Использовать HTML5 атрибуты для проверки введенных данных в форму.
- Локализовать сообщения, который выдает встроенный механизм проверки данных.
- Научиться базовым принципам работы с регулярными выражениями.
- Использовать пользовательский код для проверки введенных данных в форму.

Содержание урока

1. Событие submit
2. События blur и focus
3. События copy, paste, cut
4. Использование встроенной проверки данных HTML5
5. Добавление своих правил проверки данных при использовании встроенной проверки данных HTML5
6. Использование регулярных выражений
7. Использование пользовательского кода для проверки правильности введенных данных

Резюме

- При валидации формы используется несколько событий:
submit – событие может быть обработано на форме и срабатывает в момент отправки формы (нажатии на кнопку submit в форме)
focus – элемент управления получил фокус
blur – элемент управления потерял фокус
input – значение элемента управления изменилось
change – значение было изменено, и элемент управления потерял форму
copy – была выполнена операция копирования
paste – была выполнена операция вставки
cut – значение было вырезано из элемента управления
- Проверка данных формы или валидация формы – проверка значений, введенных в элементы управления формы на соответствие бизнес-правилам и вывод сообщений об ошибке, если правила были нарушены. Примеры правил – поле, обязательное для ввода, в поле должно находиться значение email адреса, поле должно содержать значение не менее 10 символов и т.д.

- Валидация формы должна обязательно сопровождаться ошибками, связанными с неправильно введенными данными.
- Валидация формы может проводиться:
 - На сервере
 - На клиенте
 - С помощью HTML5 атрибутов
 - С помощью JavaScript кода
- Проверка формы - обязательное действие, которое производится на стороне клиента, даже если проверка данных есть на стороне сервера. Проверка введенных данных обеспечивает правильность данных, защиту пользовательских данных, защиту серверной стороны. На стороне клиента проверка формы играет в основном роль, связанную с удобством использования, так как не нужно дожидаться ответа сервера, для того чтобы указать, где была допущена ошибка. Валидация на стороне сервера необходима для того, чтобы обеспечить целостность и безопасность данных.
- При использовании встроенной проверки через атрибуты HTML5, правила проверки указываются с помощью специальных атрибутов, а сама проверка производится в момент смены значения поля ввода и в момент отправки данных на сервер событием submit.
- При использовании встроенной проверки, элемент управления получает класс valid, если введенное значение соответствует ограничениям, заданным через атрибуты и invalid, если значение нарушает заданные правила.
- Типы input, которые поддерживают проверку введенных значений:
 - type="url" – проверяет значение на соответствие URL адресу
 - type="email" – проверяет на соответствие адресу электронной почты
- Атрибуты валидации:
 - pattern – значение должно соответствовать регулярному выражению
 - min – значение должно быть больше, чем указанное
 - max – значение должно быть меньше, чем указанное
 - required – значение обязательное
 - step – значение должно быть кратным шагу
 - maxlength – количество символов значения не должно превышать указанное
 - minlength – количество символов должно быть больше, чем указанное или равным ему
- setCustomValidity – метод, который используется для добавления пользовательского сообщения, связанного с проверкой правильности введенных данных. При вызове данного метода на элементе управления, значение, введенное в элемент управления, считается не валидным, а параметр, который передается в метод setCustomValidity, определяет текст сообщения об ошибке, которое будет показано пользователю. Для того, чтобы убрать ошибку с элемента управления, надо вызвать метод setCustomValidity с передачей пустой строки в качестве параметра. Данный метод может использоваться для добавления своих правил в валидацию формы через HTML5 атрибуты и для локализации сообщений.
- pattern – атрибут, который используется для проверки введенного значения с помощью регулярного выражения.

a - соответствует символу a
abc - соответствует символу a, далее b, далее c

\d - любая цифра
\D - любой символ, не являющийся цифрой
\w - любой символ латинского алфавита, цифры и символ нижнего подчеркивания
\W - все символы, кроме символов латинского алфавита, цифр и символ нижнего подчеркивания

[] - группа символов, указанных в скобках, например **[abc]** соответствует символу a или b или c
[a-z] - любой символ от a до z (в нижнем регистре)
[0-9] - любой символ от 0 до 9
[0-9a-zA-Z] - любой символ от 0 до 9 или любой символ латинского алфавита в нижнем или верхнем регистре

Квантификаторы – указывают количество символов, которое должно встретиться в проверяемом значении

***** - 0 или более раз
+ - 1 или более раз
? - 1 или 0 раз
{n} - n раз
{n,m} - от n до m раз

Примеры:

\d+ - соответствует строке, состоящей из одного и более цифровых символов, например, 1, 22, 123, 7777 и т.д.
\d{2,4} - соответствует строке, состоящей из двух, трех или четырех цифровых символов, например, 10, 88, 123, 5671 и т.д.
[a-z]{3}\d{2} - три любых символа латинского алфавита и две любые цифры, например, abc12, rgb43
abc\d\d\d - символы abc и три цифры, например, abc111, abc123 и т.д.
https? - символы http и 1 или 0 символов s - http или https

Закрепление материала

- Какие события могут использоваться для проверки изменения данных в поле ввода?
- Какие есть способы проверки данных формы?
- Какой атрибут нужно использовать для указания ограничения на ввод только email адресов в поле ввода?
- Что делает метод `setCustomValidity`?
- Что такое регулярное выражение?
- Какие HTML5 атрибуты поддерживают проверку через регулярные выражения?

Самостоятельная деятельность учащегося

Выполните задания в директории `Exercises\Tasks\016 Form Validation`. Текст задач находится в комментариях в тегах `script`.

Рекомендуемые ресурсы

Constraint Validation. HTML5 атрибуты для валидации форм

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

Регулярные выражения

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

Проверка данных формы

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