# Лабораторная работа №2.Формы HTML5. Разметка

В HTML4 типы полей ввода были ограничены следующим списком:

* input type="text"
* input type="checkbox"
* input type="radio"
* input type="password"
* input type="hidden" - для данных, невидимых пользователю
* input type="file" - для загрузки файлов
* textarea - для ввода больших объемов текста
* select - для выпадающих списков
* button — обычно используется для отправки данных формы, но также можно использовать input type="submit" и input type="image"

К тому же:

* возможности стилизации с помощью CSS ограничены
* расширенные элементы, вроде элементов для выбора даты и цвета нужно реализовывать самостоятельно, с помощью кода
* валидация на стороне клиента требует JavaScript

**Дополнительные типы полей**

В HTML5 было введено огромное количество новых типов полей. Они предоставляют интуитивные помощники ввода и валидации без JavaScript-кода

|  |  |
| --- | --- |
| Тип | Описание |
| email | ввод email-адреса |
| tel | ввод телефонного номера - нет строгого синтаксиса, но разрывы строк будут удалены |
| url | ввод URL |
| search | поле поиска с разрывами строк автоматически удаляется |
| number | число с плавающей точкой |
| range | элемент управления для ввода примерного значения, обычно представляемого с помощью слайдеров |
| date | ввод дня, месяца и года |
| datetime | ввод дня, месяца, года, часа, минуты, секунды и микросекунды относительно текущей временной зоны |
| datetime-local | ввод даты и времени без временной зоны |
| month | ввод месяца и года без временной зоны |
| week | ввод номера недели без временной зоны |
| time | ввод времени без временной зоны |
| color | выбор цвета |

**Атрибуты полей ввода**

Кроме перечисленных выше, поля ввода могут иметь любой из специфичных для формы атрибутов. Некоторые из них булевого типа, что означает, что они не требуют значений, например:

|  |  |
| --- | --- |
|  | <input type="email" name="email" required /> |

Или в стиле XHTML

<input type="email" name="email" required="required" />

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| name | имя поля ввода |
| value | изначальное значение |
| checked | делает флажок или переключатель выбранным |
| maxlength | Максимальная длина вводимой текстовой строки. В HTML5 также может быть применена и для полей textarea |
| minlength | Минимальная длина строки. Данная возможность документирована, но на момент написания поддержка браузерами слабая, и HTML валидаторы ругаются. Альтернативный вариант использования - pattern=".{3,}", что проверит на наличие минимум трех символов во вводимой строке |
| placeholder | ненавязчивая подсказка внутри поля |
| autofocus | устанавливает фокус на указанный элемент (видимый) при загрузке страницы |
| required | указывает, что в поле должно быть введено значение (не пустое поле) |
| pattern | проверяет значение на соответствие регулярному выражению |
| min | минимальное разрешенное значение (числовое и дата) |
| max | максимальное разрешенное значение (числовое и дата) |
| step | шаг изменения значения. Например, input type="number" min="10" max="19" step="2" разрешит только значения 10, 12, 14, 16 или 18 |
| autocomplete | указывает браузеру подсказку к автодополнению, например “расчетный счет”, или может быть выставлен в значение “on” или “off”, для включения/отключения овтозаполнения |
| inputmode | указывает способ ввода. Наиболее полезные возможности:   * verbatim - содержимое с дословной передачей смысла, например имена пользователей * latin - латиница, например, поля поиска * latin-name - имена, то есть с заглавной первой буквой * latin-prose - содержимое с не дословной передачей смысла, например сообщения, твиты, прочее. * numeric - числовые значения, где не подходят типы number и range , например, номера кредитных карт |
| size | размер в количестве символов для полей типа text или password, или количество пикселей для полей типа email, tel, url или search. Наверное, стоит избегать, и использовать стилизацию посредством CSS |
| rows | Количество текстовых строк (только для textarea) |
| cols | количество рядов символов (только для textarea) |
| list | ссылается на список опций |
| spellcheck | установить в true или false, чтобы включить/отключить проверку привописания для поля |
| form | идентификатор форму, которой принадлежит данное поле. Вообще-то поля должны размещаться внутри формы, но этот атрибут позволяет разместить поле за пределами формы, в любом месте страницы |
| formaction | указывает URI, переопределяющее действие формы при отправке данных (только для кнопок отправки/изображений) |
| formmethod | задает GET или POST, переопределяет атрибут формы method (только для кнопок отправки/изображений) |
| formenctype | указывает тип содержимого при отправке (text/plain, multipart/form-data или application/x-www-form-urlencoded, толкьо для кнопок отправки и изображений) |
| formtarget | указывает целевое окно/фрейм для переопределения атрибута формы target при отправке (только для кнопок отправки / изображений) |
| readonly | значение поля не может быть изменено, хотя оно будет отвалидировано и отправлено |
| disabled | отключает поле ввода - нет валидации, и данные не будут отправлены |

Поля date должны использовать формат YYYY-MM-DD для атрибутов value, min и max.

Следующий пример показывает поле, требующее обязательного ввода email, заканчивающегося на @mysite.com, на которое смещается фокус при загрузке страницы.

|  |
| --- |
| <input |
| type="email" | |

|  |
| --- |
| name="login" |
| pattern="@mysite\.com$" | |

|  |  |
| --- | --- |
| autocomplete="email" | |
| autofocus |

|  |
| --- |
| required /> |

**Списки данных**

Список данных содержит набор подходящих значений для любого типа поля, например:

|  |  |
| --- | --- |
| <input type="text" name="browser" list="browsers" /> | |
|  |

|  |
| --- |
| <datalist id="browsers"> |
| <option value="Chrome" /> | |

|  |
| --- |
| <option value="Firefox" /> |
| <option value="Internet Explorer" /> | |

|  |  |
| --- | --- |
| <option value="Safari" /> | |
| <option value="Opera" /> |

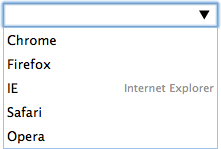
|  |
| --- |
| </datalist> |

Если поддерживается datalist, браузер предоставляет список значений для автозаполнения, как только вы начинаете вводить данные в поле. Полный список обычно отображается при клике на стрелку, указывающую вниз (если таковая имеется). В отличие от стандартного выпадающего списка, пользователь может вводить произвольные значения в поле.

Возможно указывать значения и текстовые пояснения к значениям, например:

<option value="IE">Internet Explorer</option>

но стоить помнить, что реализация отличается в различных браузерах. Например, Firefox производит автодополнение по тексту (Internet Explorer), тогда как Chrome - по значениям (IE), а текст указывает уменьшенным серым шрифтом:



Списки данных можно заполнять с помощью JavaScript, если вы хотите получать их по AJAX.

**Отключение валидации**

Отключит валидацию всей формы можно выставив атрибут novalidate элементу form. В дополнение, можно выставить атрибут formnovalidate кнопке/изображению отправки формы.

Запомните, что выставление полю атрибута disabled отключит валидацию этого поля.

**Поля вывода**

HTML5 также предоставляет поля для вывода информации

* output - результат вычисления пользовательских действий
* progress - полоса прогресса (атрибуты value и max задают состояние)
* meter - шкала, которая может менять свой цвет между зеленым, желтым и красным в зависимости от установленных значений атрибутов value, min, max, low, high и optimum

**Разделение и подпись полей**

Согласно спецификации, каждый элемент формы считается параграфом, и отделяется от остальных частей элементом <p>

Метки нужно использовать, либо окружая элемент формы, либо размещая их рядом, и связывая их атрибутом for с соответствующими полями, например:

<p>

  <p>

  <label for="firstname">Имя</label>

  <input type="text" id="firstname" name="firstname" placeholder="first name" required maxlength="20" />

</p>

<p>

  <label for="lastname">Фамилия</label>

  <input type="text" id="lastname" name="lastname" placeholder="last name" required maxlength="20" />

</p>

<p>

  <label for="email">Email адрес</label>

</p>

<p>

  <label>

    <input type="checkbox" name="newsletter" />

    Подпишитесь на рассылку

  </label>

</p>

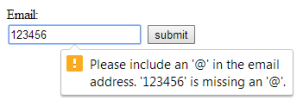
## Валидация форм с помощью HTML5

HTML5 предоставляет довольно надежный механизм, основанный на следующих атрибутах тега <input />: type, pattern и require. Благодаря этим новым атрибутам вы можете переложить некоторые функции проверки данных на плечи браузера.

## Атрибут type

Этот атрибут говорит, какое поле ввода отобразить для обработки данных, например, уже знакомое поле типа <input type="text" />

Некоторые поля ввода уже предоставляют стандартные способы валидации, без необходимости писать дополнительный код. Например, <input type="email" /> проверяет поле на то, что введенное значение соответствует шаблону правильного email адреса. Если в поле введен неверный символ, форму нельзя будет отправить, пока значение не будет исправлено.



формат телефонного номера различается для разных стран из-за несоответствия количества цифр в телефонных номерах и разности форматов. Как результат, спецификация не определяет алгоритм проверки телефонных номеров, так что данная возможность слабо поддерживается браузерами.

К счастью для нас, валидацию телефонных номеров можно провести с использованием атрибута pattern, который принимает как аргумент регулярное выражение.

## Атрибут pattern

Атрибут pattern, скорее всего, заставит многих фронтенд-разработчиков прыгать от радости. Этот атрибут принимает регулярное выражение (аналогичное формату регулярных выражений JavaScript), по которому будет проверяться корректность введенных в поле данных.

Регулярные выражения это язык, использующийся для разбора и манипуляции текстом. Они часто используются для сложных операций поиска и замены, а также для проверки корректности введенных данных.

На сегодняшний день регулярные выражения включены в большинство популярных языков программирования, а также во многие скриптовые языки, редакторы, приложения, базы данных, и утилиты командной строки.

Регулярные выражения (RegEX) являются мощным, кратким и гибким инструментом для сопоставления строки текста, вроде отдельных символов, слов или шаблонов символов.

Передав регулярное выражение в качестве значения атрибута pattern можно указать, какие значения приемлемы для данного поля ввода, а также проинформировав пользователя об ошибках.

### Телефонные номера

Как упоминалось ранее, тип поля tel не полностью поддерживается браузерами из-за несоответствия форматов номеров телефонов в разных странах.

Например, в некоторых странах формат телефонных номеров представляется в виде xxxx-xxx-xxxx, и сам телефонный номер будет что-то вроде этого: **0803-555-8205**.

Регулярное выражение, которому соответствует данный шаблон, такое: ^\d{4}-\d{3}-\d{4}$. В коде это можно записать так:

<label for="phonenum">Phone Number:</label>

<input type="tel" pattern="^\d{4}-\d{3}-\d{4}$" >

### Буквенно-цифровые значения

Следующий шаблон соответствует набору буквенно-цифровых значений (комбинации букв английского алфавита и цифр):

<input type="text" pattern="[a-zA-Z0-9]+" >

### Имя пользователя в Twitter

Данное регулярное выражение соответствует имени пользователя в твиттере, с предваряющим символом @, например@tech4sky:

<input type="text" pattern="^@[A-Za-z0-9\_]{1,15}$" >

### Цвет в шестнадцатеричном виде

Эта регулярка соответствует цвету в шестнадцатеричном виде, например **#3b5998** или **#000**:

<input type="text" pattern="^#+([a-fA-F0-9]{6}|[a-fA-F0-9]{3})$" >

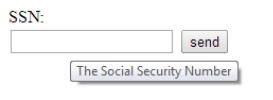
## Подсказки

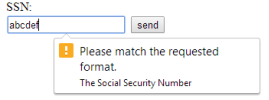
Для того, чтобы дать пользователю подсказку о шаблоне, или сообщить об ошибке в случае неверно введенного в поле значения, можно использовать атрибут title:

<input type="text" name="ssn"

pattern="^\d{3}-\d{2}-\d{4}$"

title="Номер Социального Страхования" />





## Атрибут required

Это атрибут булевого типа, использующийся для указания того, что значение данного поле обязательно заполнить для того, чтобы отправить форму. При добавлении этого атрибута полю браузер потребует от пользователя заполнить данное поле перед отправкой формы.

Это избавляет нас от реализации проверки полей с помощью JavaScript, что может сохранить немного времени разработчикам.

Например: <input type="text" name="my\_name" required /> или <input type="text" name="my\_name" required="required" />(для совместимости с XHTML)



Некоторые типовые регулярные выражения перечислены в таблице

|  |  |
| --- | --- |
| **Выражение** | **Описание** |
| \d [0-9] | Одна цифра от 0 до 9. |
| \D [^0-9] | Любой символ кроме цифры. |
| \s | Пробел. |
| [A-Z] | Только заглавная латинская буква. |
| [A-Za-z] | Только латинская буква в любом регистре. |
| [А-Яа-яЁё] | Только русская буква в любом регистре. |
| [A-Za-zА-Яа-яЁё] | Любая буква русского и латинского алфавита. |
| [0-9]{3} | Три цифры. |
| [A-Za-z]{6,} | Не менее шести латинских букв. |
| [0-9]{,3} | Не более трёх цифр. |
| [0-9]{5,10} | От пяти до десяти цифр. |
| ^[a-zA-Z]+$ | Любое слово на латинице. |
| ^[А-Яа-яЁё\s]+$ | Любое слово на русском включая пробелы. |
| ^[ 0-9]+$ | Любое число. |
| [0-9]{6} | Почтовый индекс. |
| \d+(,\d{2})? | Число в формате 1,34 (разделитель запятая). |
| \d+(\.\d{2})? | Число в формате 2.10 (разделитель точка). |
| \d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3} | IP-адрес |

# Пример формы

# <HTML>

# <HEAD>

# <META http-equiv="Content-Type" content="text/html; charset=Windows-1251">

# <TITLE>Форма</TITLE>

# </HEAD><BODY >

# <FORM name="contact\_form" method="post" action="check2.html" >

# <H1>Пожалуйста введите ваши данные.</H1>

# <p>

# <label for="firstname">Имя</label>

# <input type="text" id="firstname" name="firstname" placeholder="first name" required maxlength="20" />

# </p>

# <p>

# <label for="lastname">Фамилия</label>

# <input type="text" id="lastname" name="lastname" placeholder="last name" required maxlength="20" />

# </p>

# <P>Ваш пол: <INPUT type="radio" name="gender" value="Male"> Мужской

# &nbsp; <INPUT type="radio" name="gender" value="Female"> Женский</P>

# <P>Ваш возраст:

# <SELECT name="age">

# <OPTION value="">Выберите одну из опций:</OPTION>

# <OPTION value="0-18 years">0-18 лет</OPTION>

# <OPTION value="18-30 years">18-30 лет</OPTION>

# <OPTION value="30-45 years">30-45 лет</OPTION>

# <OPTION value="45-60 years">45-60 лет</OPTION>

# <OPTION value="60+ years">60+ лет</OPTION>

# </SELECT>

# </p>

# <p>

# <label for="email">Email адрес</label>

# <input type="email" id="email" name="email" placeholder=your@email.address required maxlength="50" />

# </p>

# <p>

# <label>

# <input type="checkbox" name="newsletter" />

# Подпишитесь на рассылку

# </label>

# </p>

# </P>

# <P><INPUT type="submit" name="send" value="Отправить данные"></P>

# </FORM>

# </BODY></HTML>

# Задание

# Создать произвольную форму с различными типами полей.

# Предусмотреть:

# подсказки пользователю,

# обязательность заполнения полей,

# выпадающие списки,

# валидацию на стороне клиента с использованием паттернов e-mail, числовых и текстовых полей.

# Создать стиль CSS для оформления формы используя псевдоклассы:

* Required
* Optional
* Valid
* Invalid
* Checked
* Lang

# Включить форму отдельной страницей в созданный в первой лабораторной работе сайт.