

Лекция 2.2.

Работа с формами

Элементы для ввода чисел

Простое числовое поле

Для ввода чисел используется элемент `input` с атрибутом `type="number"`. Он создает числовое поле, которое можно настроить с помощью следующих атрибутов:

- `min`: минимально допустимое значение
- `max`: максимально допустимое значение
- `readonly`: доступно только для чтения
- `required`: указывает, что данное поле обязательно должно иметь значение
- `step`: значение, на которое будет увеличиваться число в поле
- `value`: значение по умолчанию

например

```
<form>
```

```
  <p>
```

```
    <label for="age">Возраст: </label>
```

```
      <input type="number" step="1" min="1" max="100"
value="10" id="age" name="age"/>
```

```
  </p>
```

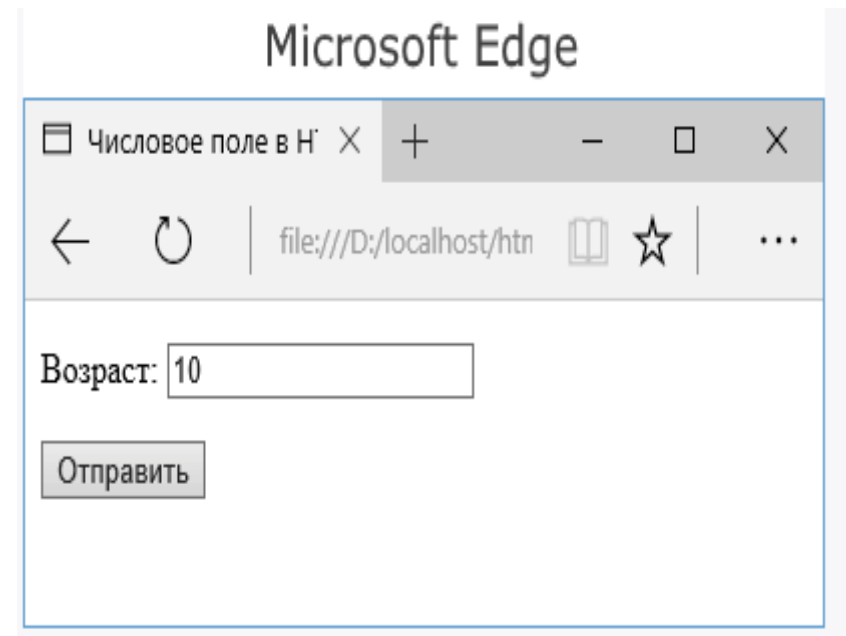
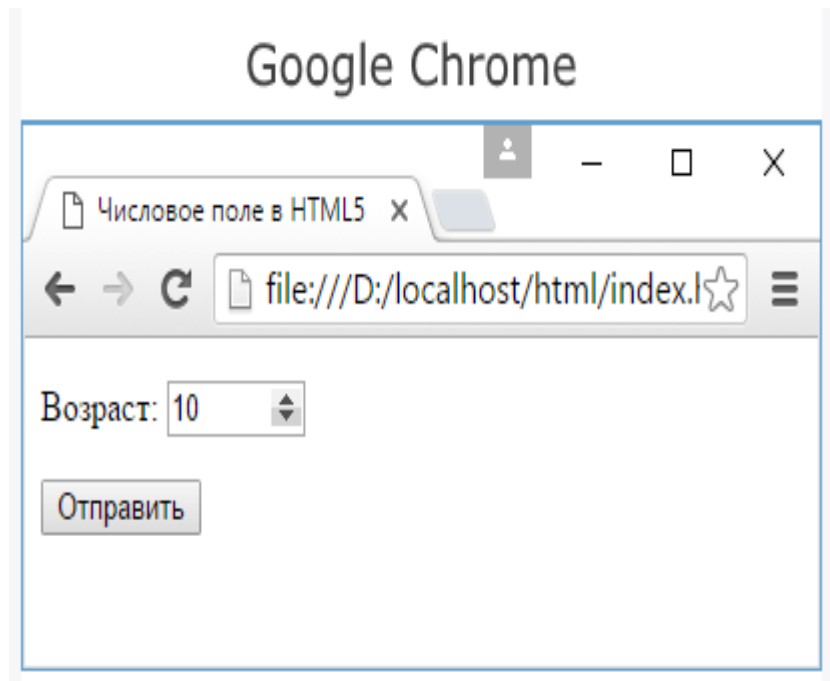
```
  <p>
```

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

```
  </p>
```

```
</form>
```

В зависимости от браузера визуализация этого поля может отличаться:



Если используется браузер Microsoft Edge, то можно здесь прикрепить список datalist с диапазоном возможных значений, как и в текстовом поле:

```
<form>
```

```
  <p>          <label for="price">Цена: </label>
```

```
    <input type="number" list="priceList"
```

```
      step="1000" min="3000" max="100000" value="10000" id="price" name="price"/>  </p>
```

```
  <p>          <button type="submit">Отправить</button>          </p>
```

```
</form>
```

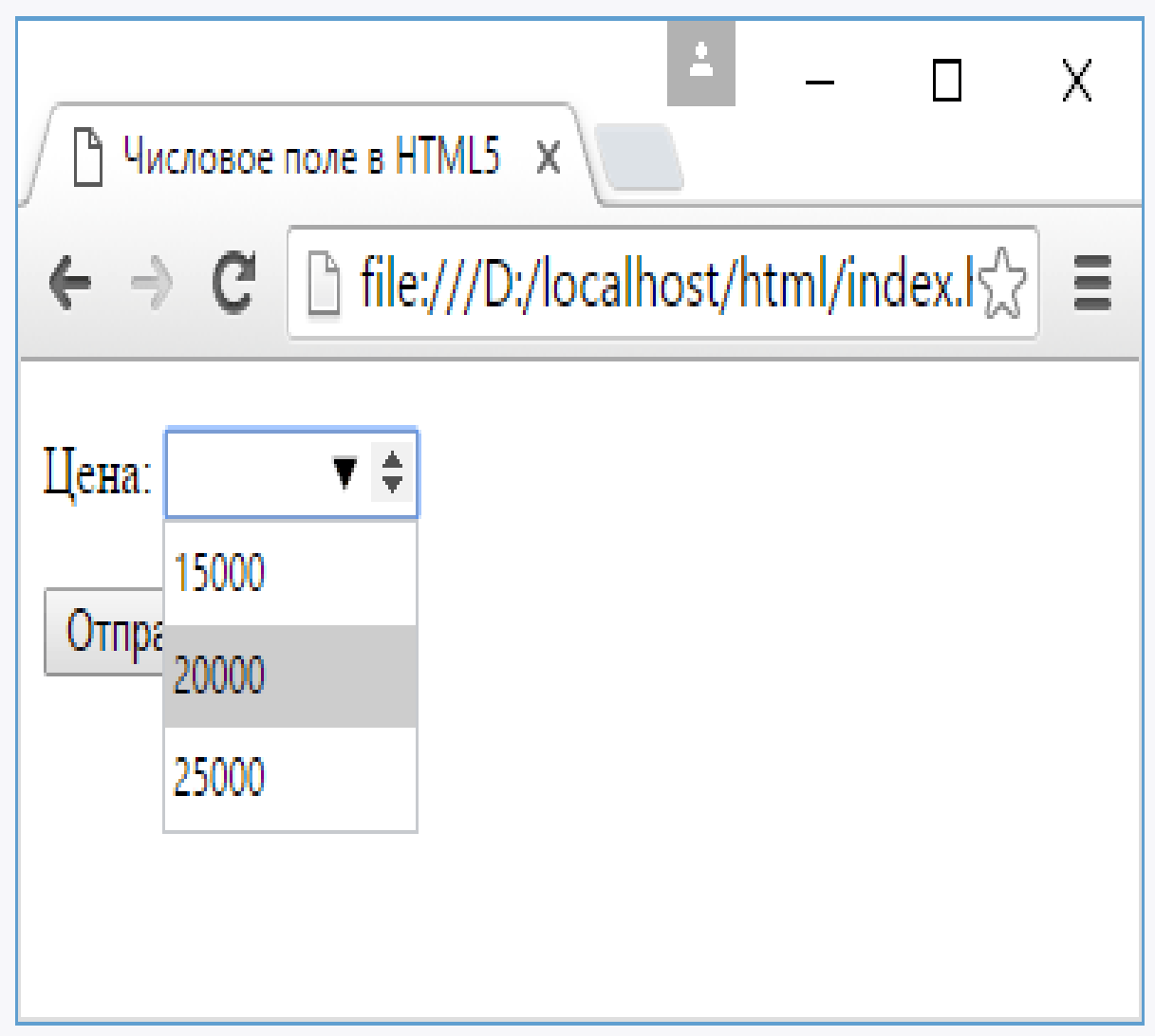
```
<datalist id="priceList">
```

```
  <option value="15000" />
```

```
  <option value="20000" />
```

```
  <option value="25000" />
```

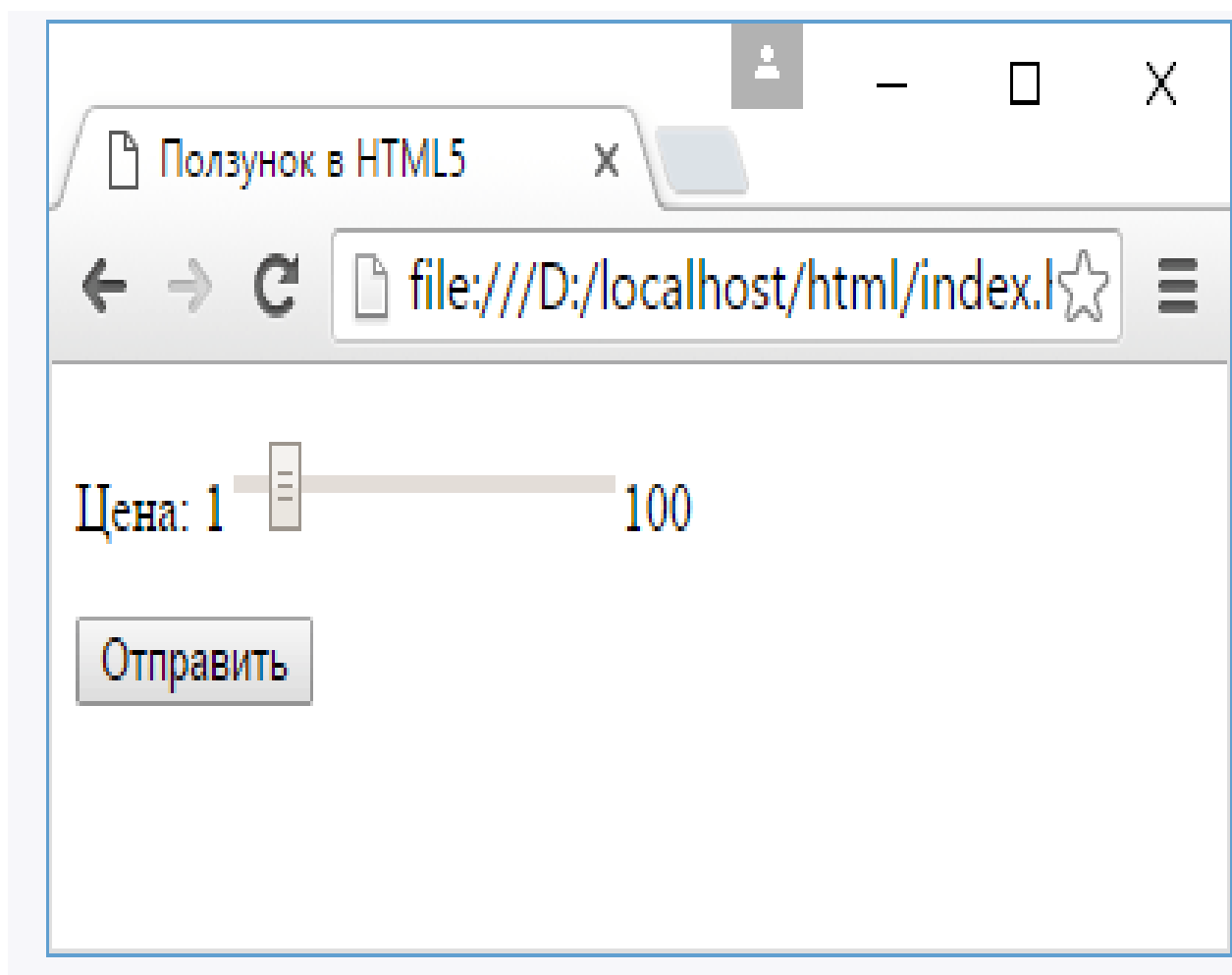
```
</datalist>
```



Ползунок

Ползунок представляет шкалу, на которой мы можем выбрать одно из значений. Для создания ползунка применяется элемент `input` с атрибутом `type="range"`. Во многом ползунок похож на простое поле для ввода чисел. Он также имеет атрибуты `min`, `max`, `step` и `value`:

```
<form>
    <p>                <label for="price">Цена:</label>
<input type="range" step="1" min="0" max="100" value="10" id="price"
name="price"/>100    </p>
    <p>                <button type="submit">Отправить</button>                </p>
</form>
```



Флажки и переключатели

Флажок

Флажок представляет элемент, который может находиться в двух состояниях: отмеченном и неотмеченном. Флажок создается с помощью элемента input с атрибутом type="checkbox":

```
<h2>Изучаемые технологии</h2>
```

```
<form>
```

```
    <p>        <input type="checkbox" checked name="html5"/>HTML5
</p>
```

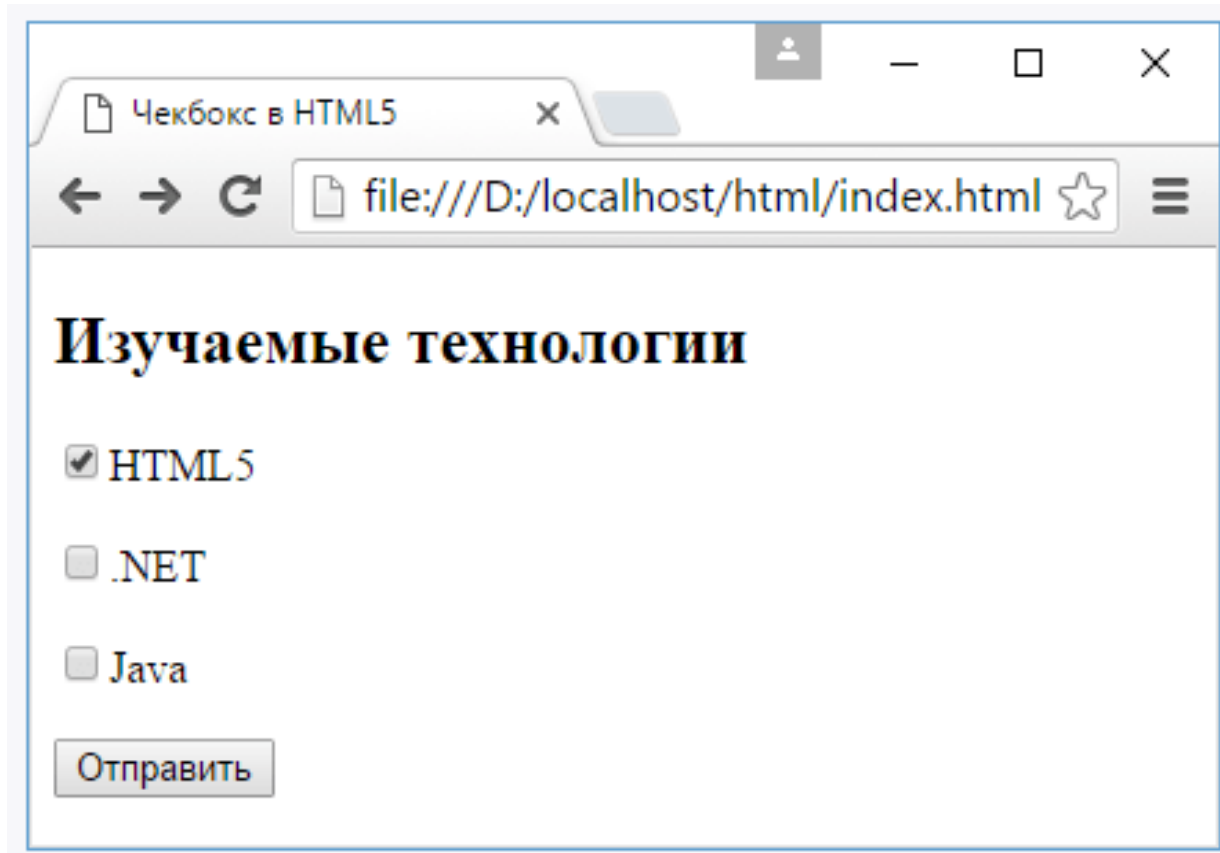
```
    <p>        <input type="checkbox" name="dotnet"/>.NET        </p>
```

```
    <p>        <input type="checkbox" name="java"/>Java        </p>
```

```
    <p>        <button type="submit">Отправить</button>        </p>
```

```
</form>
```

Атрибут checked позволяет установить флажок в отмеченное состояние.



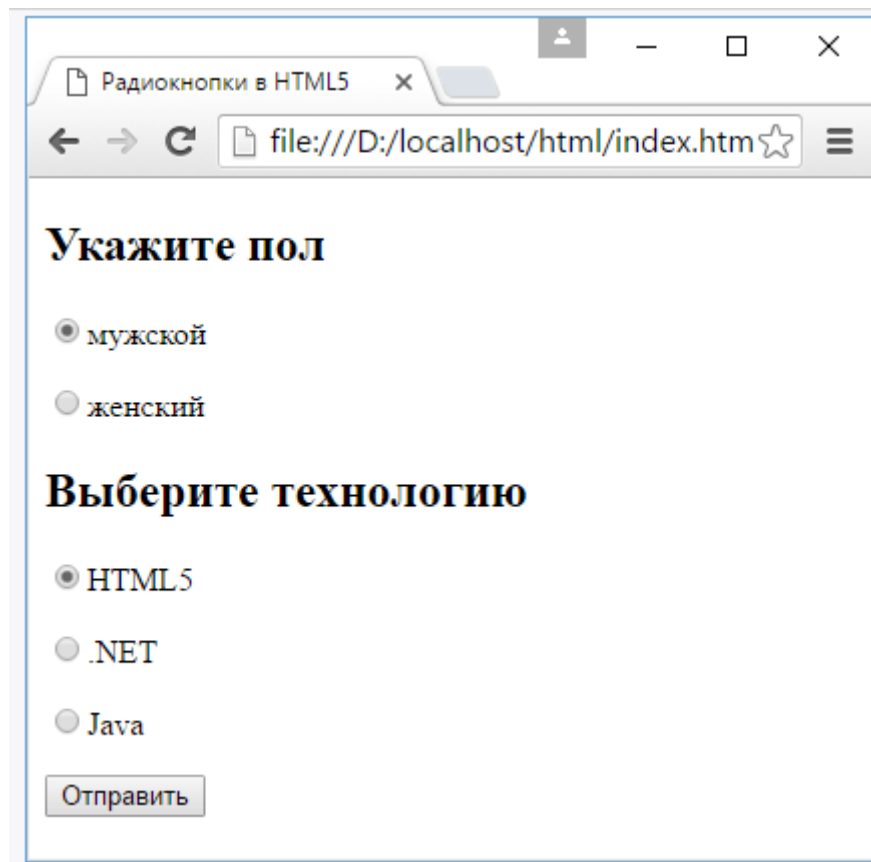
Переключатели

- Переключатели или радиокнопки похожи на флажки, они также могут находиться в отмеченном или неотмеченном состоянии.
- Только для переключателей можно создать одну группу, в которой одновременно можно выбрать только один переключатель.

```
...    <form>        <h2>Укажите пол</h2>
        <p>            <input type="radio" value="man" checked
name="gender"/>мужской    </p>
        <p>            <input type="radio" value="woman"
name="gender"/>женский      </p>
        <h2>Выберите технологию</h2>
        <p>            <input type="radio" value="html5" checked
name="tech"/>HTML5        </p>
        <p>            <input type="radio" value="net" name="tech"/>.NET
</p>
        <p>            <input type="radio" value="java" name="tech"/>Java </p>
        <p>            <button type="submit">Отправить</button>      </p>
        </form>
```

- Для создания радиокнопки надо указать атрибут `type="radio"`.
- И теперь другой атрибут `name` указывает не на имя элемента, а на имя группы, к которой принадлежит элемент-радиокнопка.
- В данном случае у нас две группы радиокнопок: `gender` и `tech`. Из каждой группы можно выбрать только один переключатель

чтобы отметить радиокнопку, у нее устанавливается атрибут checked:



Радиокнопки в HTML5

file:///D:/localhost/html/index.htm

Укажите пол

☒ мужской

☐ женский

Выберите технологию

☒ HTML5

☐ .NET

☐ Java

Отправить

Важное значение играет атрибут value, который при отправке формы позволяет серверу определить, какой именно переключатель был отмечен:

Радиокнопки в HTML5

file:///D:/localhost/html/index.html?gender=man&tech=java

Укажите пол

☒ мужской

☐ женский

Выберите технологию

☐ HTML5

☐ .NET

☒ Java

Отправить

The image shows a web browser window with a form titled "Радиокнопки в HTML5". The address bar shows the file path "file:///D:/localhost/html/index.html?gender=man&tech=java". The form contains two sections: "Укажите пол" (Specify gender) with radio buttons for "мужской" (male) and "женский" (female), and "Выберите технологию" (Choose technology) with radio buttons for "HTML5", ".NET", and "Java". The "мужской" and "Java" options are selected. Red arrows point from the "мужской" and "Java" radio buttons to the "gender=man" and "tech=java" parts of the query string in the address bar, respectively.

Элементы для ввода цвета, url, email, телефона

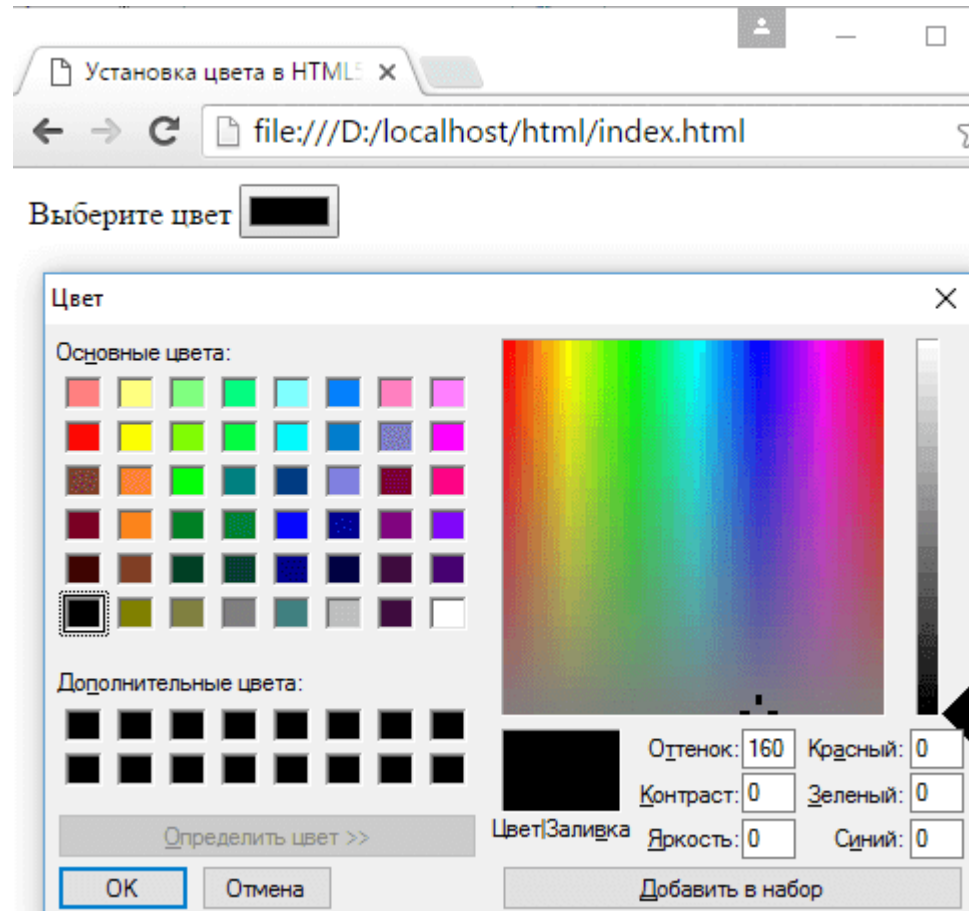
Установка цвета

За установку цвета в HTML5 отвечает специальный элемент input с типом color:

```
<label for="favcolor">Выберите цвет</label>
```

```
<input type="color" id="favcolor" name="favcolor" />
```

Элемент отображает выбранный цвет. А при нажатии на него появляется специальное диалоговое окно для установки цвета:

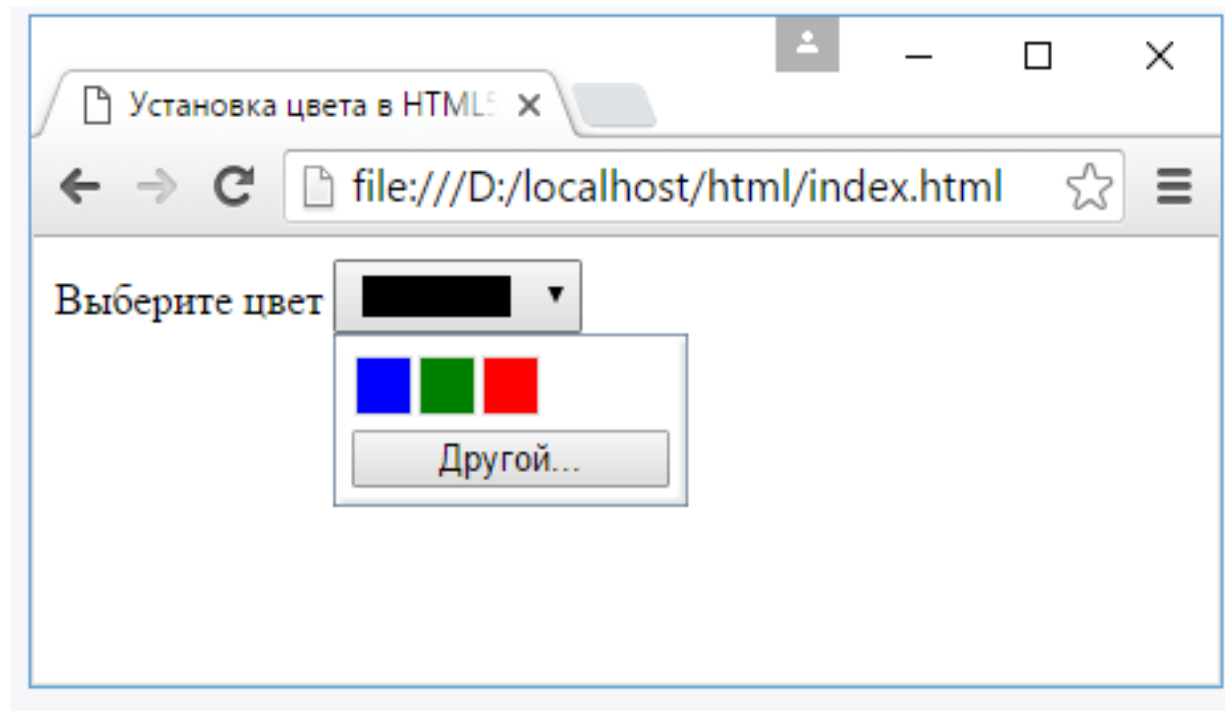


Значением этого элемента будет числовой шестнадцатеричный код выбранного цвета.

С помощью элемента `datalist` можно задать набор цветов, из который пользователь может выбрать нужный:

- `<label for="favcolor">Выберите цвет</label>`
- `<input type="color" list="colors" id="favcolor" name="favcolor" />`
- `<datalist id="colors">`
 - `<option value="#0000FF" label="blue">`
 - `<option value="#008000" label="green">`
 - `<option value="#ff0000" label="red">`
- `</datalist>`

Каждый элемент option в datalist должен в качестве значения принимать шестнадцатеричный код цвета, например, "#0000FF". После выбора цвета данный числовой код устанавливается в качестве значения в элементе input



Поля для ввода url, email, телефона

- Ряд полей input предназначены для ввода таких данных, как url, адреса электронной почты и телефонного номера.
- Они однотипны и во многом отличаются только тем, что для атрибута type принимают соответственно значения email, tel и url.

Для их настройки можно использовать те же атрибуты, что и для обычного текстового поля:

- `maxlength`: максимально допустимое количество символов в поле
- `pattern`: определяет шаблон, которому должен соответствовать вводимый текст
- `placeholder`: устанавливает текст, который по умолчанию отображается в поле
- `readonly`: делает текстовое поле доступным только для чтения
- `required`: указывает, что текстовое поле обязательно должно иметь значение
- `size`: устанавливает ширину поля в видимых символах
- `value`: устанавливает значение по умолчанию для поля
- `list`: устанавливает привязку к элементу `datalist` со списком возможных значений

... <form>

<p> <label for="email">Email: </label>

<input type="email" placeholder="user@gmail.com" id="email"
name="email"/> </p>

<p> <label for="url">URL: </label>

<input type="url" id="url" name="url"/> </p>

<p> <label for="phone">Телефон: </label>

<input type="tel" placeholder="(XXX)-XXX-XXXX" id="phone"
name="phone"/> </p>

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

</form>...

Форма ввода в HTML5

←

→

↺

file:///D:/localhost/html/index.html

☆

☰

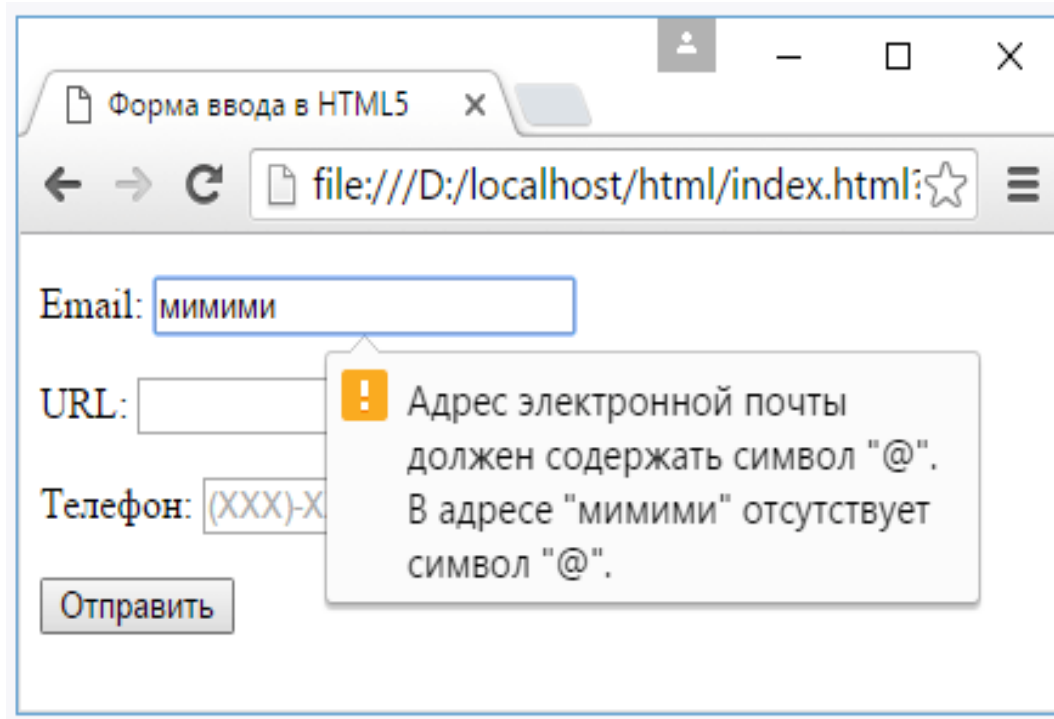
Email:

URL:

Телефон:

Отправить

Основное преимущество подобных полей ввода перед обычными текстовыми полями состоит в том, что поля ввода для email, url, телефона для проверки ввода используют соответствующий шаблон. Например, если введем в какое-либо поле некорректное значение и попробуем отправить форму, то браузер может отобразить нам сообщение о некорректном вводе, а форма не будет отправлена:



Элементы для ввода даты и времени

Для работы с датами и временем в HTML5 предназначено несколько типов элементов `input`:

- `datetime-local`: устанавливает дату и время
- `date`: устанавливает дату
- `month`: устанавливает текущий месяц и год
- `time`: устанавливает время
- `week`: устанавливает текущую неделю

```
<form>
```

```
    <p>        <label for="firstname">Имя: </label>
```

```
        <input type="text" id="firstname" name="firstname"/>    </p>
```

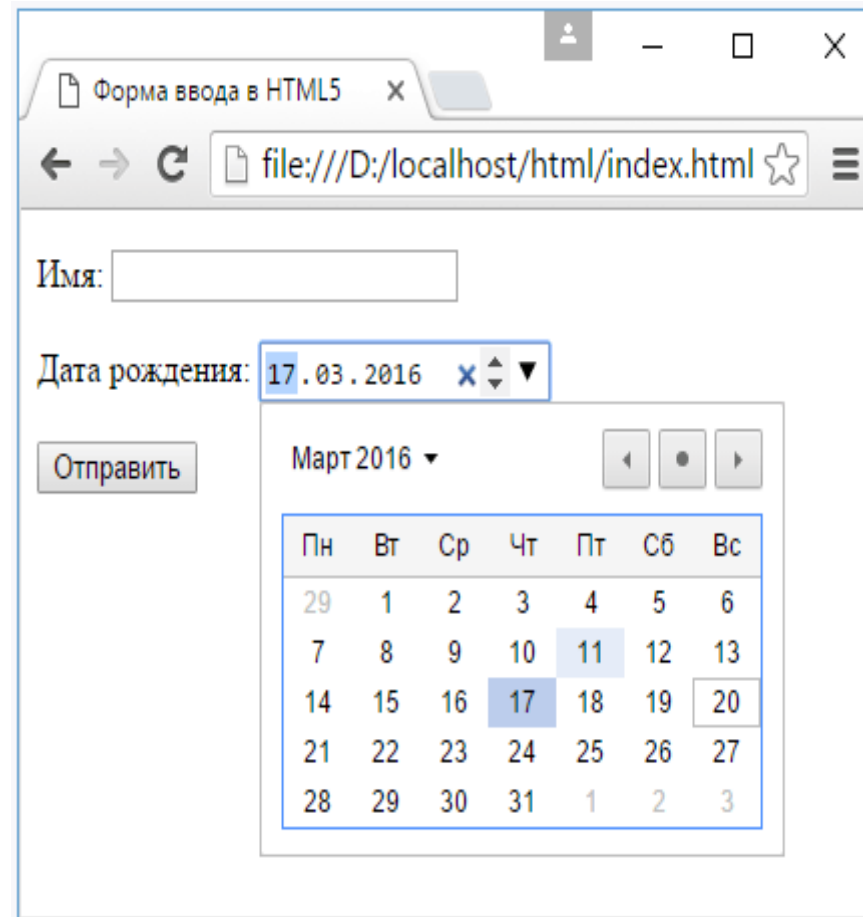
```
    <p>        <label for="date">Дата рождения: </label>
```

```
        <input type="date" id="date" name="date"/>    </p>
```

```
    <p>        <button type="submit">Отправить</button>    </p>
```

```
</form>
```

И при вводе в поле для даты будет открываться календарик:



The screenshot shows a web browser window with the title "Форма ввода в HTML5". The address bar displays the file path "file:///D:/localhost/html/index.html". The form contains an input field for "Имя:" and a date input field for "Дата рождения:". The date input field shows "17.03.2016" and has a small calendar icon. A calendar popup is open, showing the month of "Март 2016". The calendar grid has days of the week as headers (Пн, Вт, Ср, Чт, Пт, Сб, Вс) and dates from 1 to 31. The date "17" is highlighted in blue. There is a button labeled "Отправить" below the date input field.

Пн	Вт	Ср	Чт	Пт	Сб	Вс
29	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

Действие этого элемента зависит от браузера. В данном случае используется Google Chrome. В последних версиях Opera элемент не будет сильно отличаться. А вот в Microsoft Edge элемент будет выглядеть так:

Форма ввода в HTML5

Имя:

Дата рождения:

17	^	2013
18	декабря	2014
19	января	2015
20	февраля	2016
21	марта	2017
22	апреля	2018
23	▼	2019

Отправить

✓ ✕

```
<form>
```

```
    <p>        <label for="week">Неделя: </label>
```

```
        <input type="week" name="week" id="week" />        </p>
```

```
    <p>        <label for="localdate">Дата и время: </label>
```

```
        <input type="datetime-local" id="localdate" name="date"/>        </p>
```

```
    <p>        <label for="month">Месяц: </label>
```

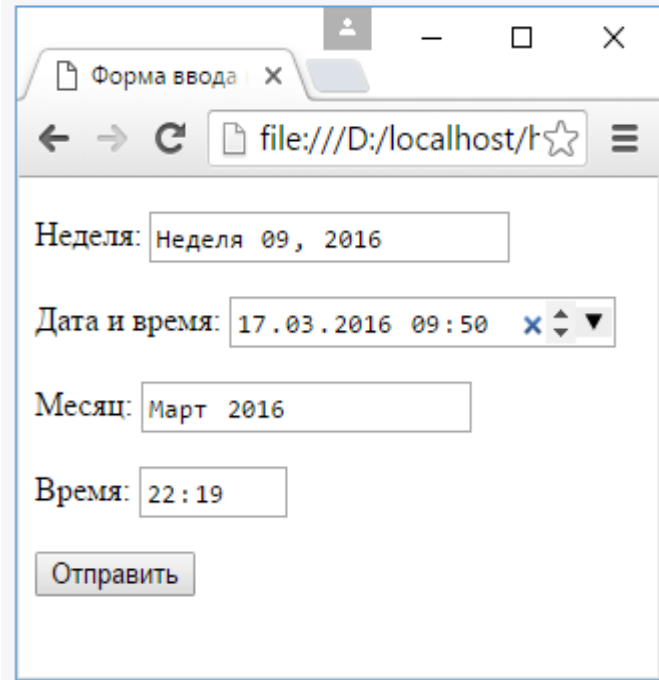
```
        <input type="month" id="month" name="month"/>        </p>
```

```
    <p>        <label for="time">Время: </label>
```

```
        <input type="time" id="time" name="time"/>        </p>
```

```
    <p>        <button type="submit">Отправить</button>        </p>
```

```
</form>
```



The screenshot shows a web browser window with a single tab titled "Форма ввода". The address bar displays a file path: "file:///D:/localhost/t". The form contains the following elements:

- A text input field labeled "Неделя:" containing the text "Неделя 09, 2016".
- A date and time picker control labeled "Дата и время:" showing "17.03.2016 09:50" with a clear button (X) and a dropdown arrow.
- A text input field labeled "Месяц:" containing the text "Март 2016".
- A text input field labeled "Время:" containing the text "22:19".
- A button labeled "Отправить" (Send).

//При использовании этих элементов также надо учитывать, что Firefox поддерживает только элементы date и time, для остальных создаются обычные текстовые поля. А IE11 и вовсе не поддерживают эти элементы.

Отправка файлов

За выбор файлов на форме отвечает элемент input с атрибутом type="file":

```
<form enctype="multipart/form-data" method="post"
action="http://localhost:8080/postfile.php">
```

```
<p>
```

```
<input type="file" name="file" />
```

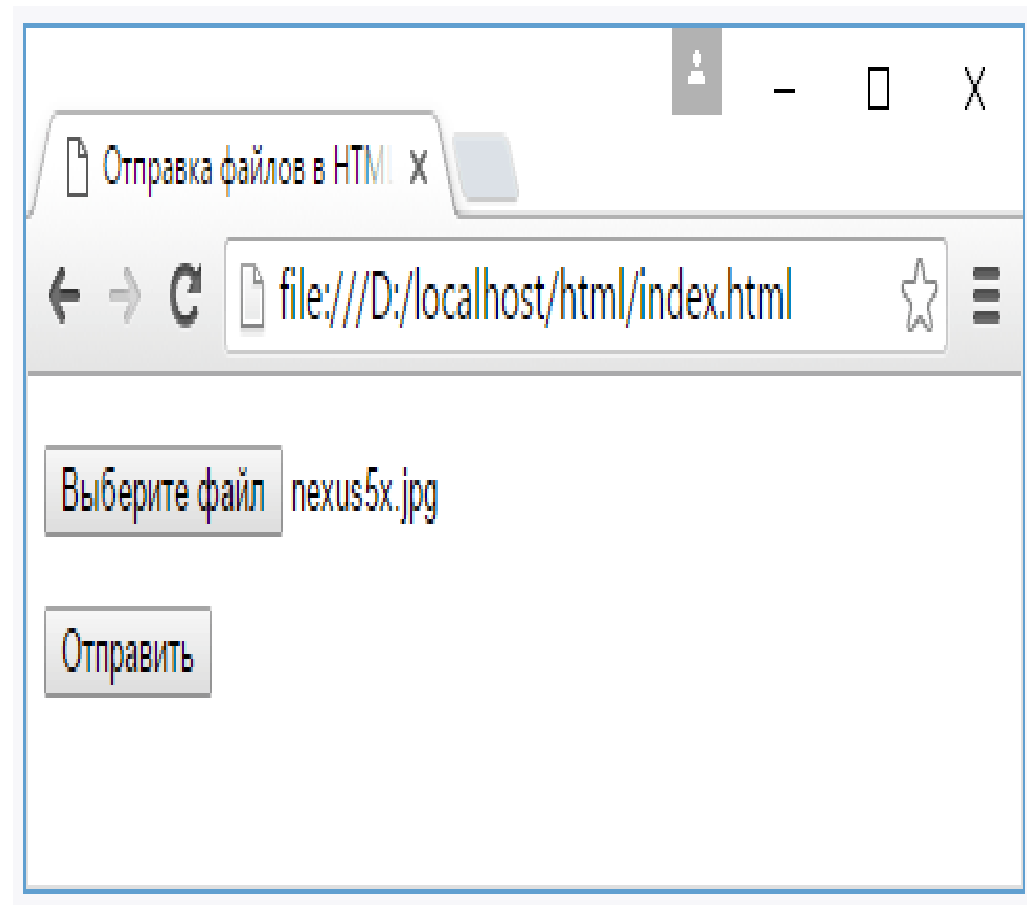
```
</p>
```

```
<p>
```

```
<input type="submit" value="Отправить" />
```

```
</p>
```

```
</form>
```



При нажатии на кнопку "Выберите файл" открывается диалоговое окно для выбора файла. А после выбора рядом с кнопкой отображается имя выбранного файла.

Важно отметить, что для отправки файла на сервер форма должна иметь атрибут `enctype="multipart/form-data"`.

С помощью ряда атрибутов мы можем дополнительно настроить элементы выбора файла:

- `accept`: устанавливает тип файл, которые допустимы для выбора
- `multiple`: позволяет выбирать множество файлов
- `required`: требует обязательной установки файла

Например, множественный выбор файлов:

```
<form enctype="multipart/form-data" method="post"
action="http://localhost:8080/postfile.php">
```

```
  <p>
```

```
    <input type="file" name="file" multiple />
```

```
  </p>
```

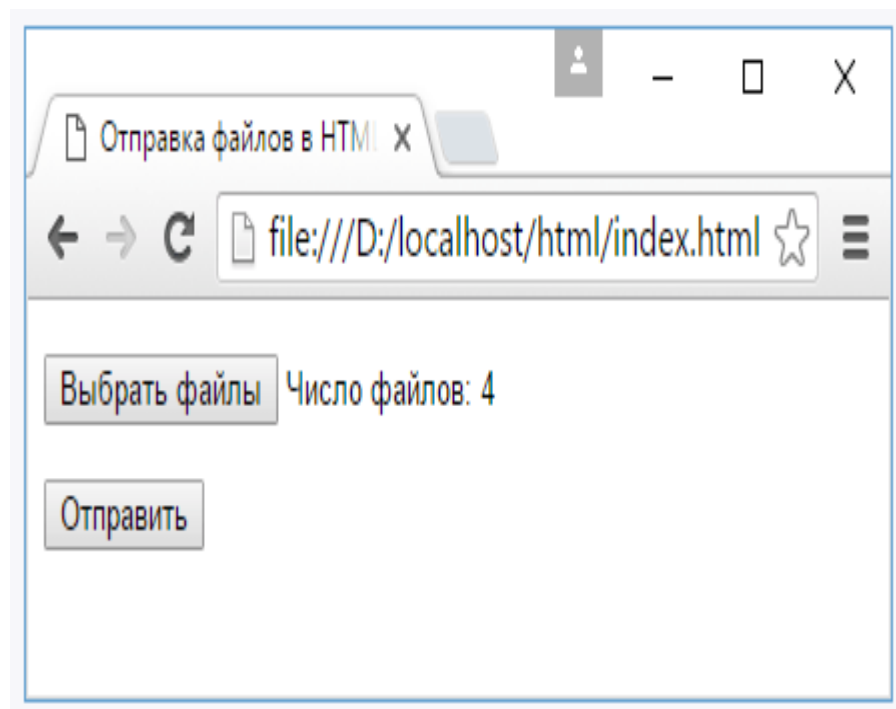
```
  <p>
```

```
    <input type="submit" value="Отправить" />
```

```
  </p>
```

```
</form>
```

При нажатии на кнопку также открывается диалоговое окно для выбора файлов и зажав клавишу CTRL или Shift, можно выбрать несколько файлов, а после выбора рядом с кнопкой отобразится количество выбранных файлов:



Список select

- Элемент select создает список. В зависимости от настроек это может быть выпадающий список для выбора одного элемента, либо раскрытый список, в котором можно выбрать сразу несколько элементов.

пример выпадающего списка:

```
<form method="get">
```

```
    <p>        <label for="phone">Выберите модель:</label>
```

```
    <select id="phone" name="phone">
```

```
        <option value="iphone 6s">iPhone 6S</option>
```

```
        <option value="lumia 950">Lumia 950</option>
```

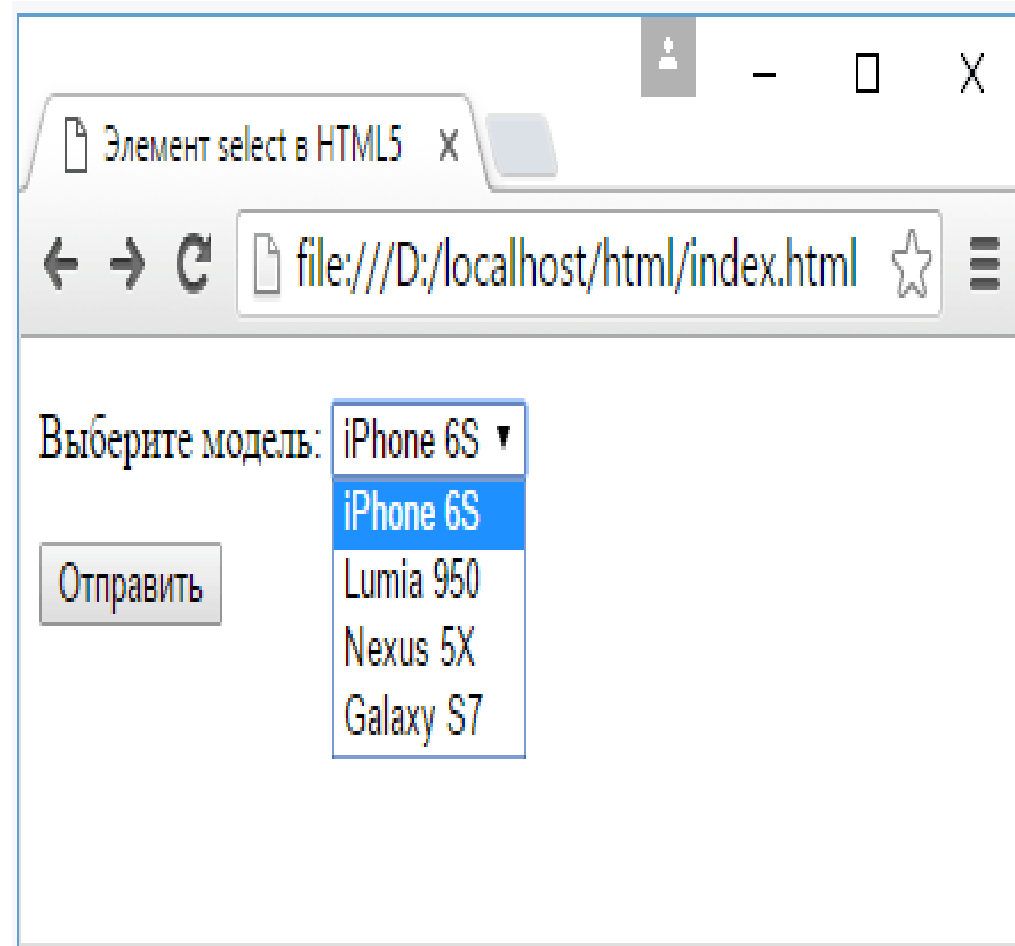
```
        <option value="nexus 5x">Nexus 5X</option>
```

```
        <option value="galaxy s7">Galaxy S7</option>
```

```
    </select>        </p>
```

```
    <p>        <input type="submit" value="Отправить" />        </p>
```

```
</form>
```



Внутри элемента `select` помещаются элементы `option` - элементы списка.

Каждый элемент `option` содержит атрибут `value`, который хранит значение элемента.

При этом значение элемента `option` не обязательно должно совпадать с отображаемым им текстом

```
<option value="apple">iPhone 6S</option>
```

С помощью атрибута `selected` можно установить выбранный по умолчанию элемент - это обязательно должен быть первый элемент в списке:

```
<select id="phone" name="phone">  
  <option value="iphone 6s">iPhone 6S</option>  
  <option value="lumia 950">Lumia 950</option>  
  <option value="nexus 5x" selected>Nexus 5X</option>  
</select>
```

С помощью другого атрибута disabled можно запретить выбор определенного элемента. Как правило, элементы с этим атрибутом служат для создания заголовков:

```
<select id="phone" name="phone">  
  <option disabled selected>Выберите модель</option>  
  <option value="iphone 6s">iPhone 6S</option>  
  <option value="lumia 950">Lumia 950</option>  
  <option value="nexus 5x" selected>Nexus 5X</option>  
</select>
```

Для создания списка с множественным выбором к элементу select надо добавить атрибут multiple:

```
<form method="get">
```

```
    <p>        <label for="phone">Выберите модель:</label> <br/>
```

```
        <select multiple id="phone" name="phone">
```

```
        <option value="iphone 6s">iPhone 6S</option>
```

```
        <option value="lumia 950">Lumia 950</option>
```

```
        <option value="nexus 5x">Nexus 5X</option>
```

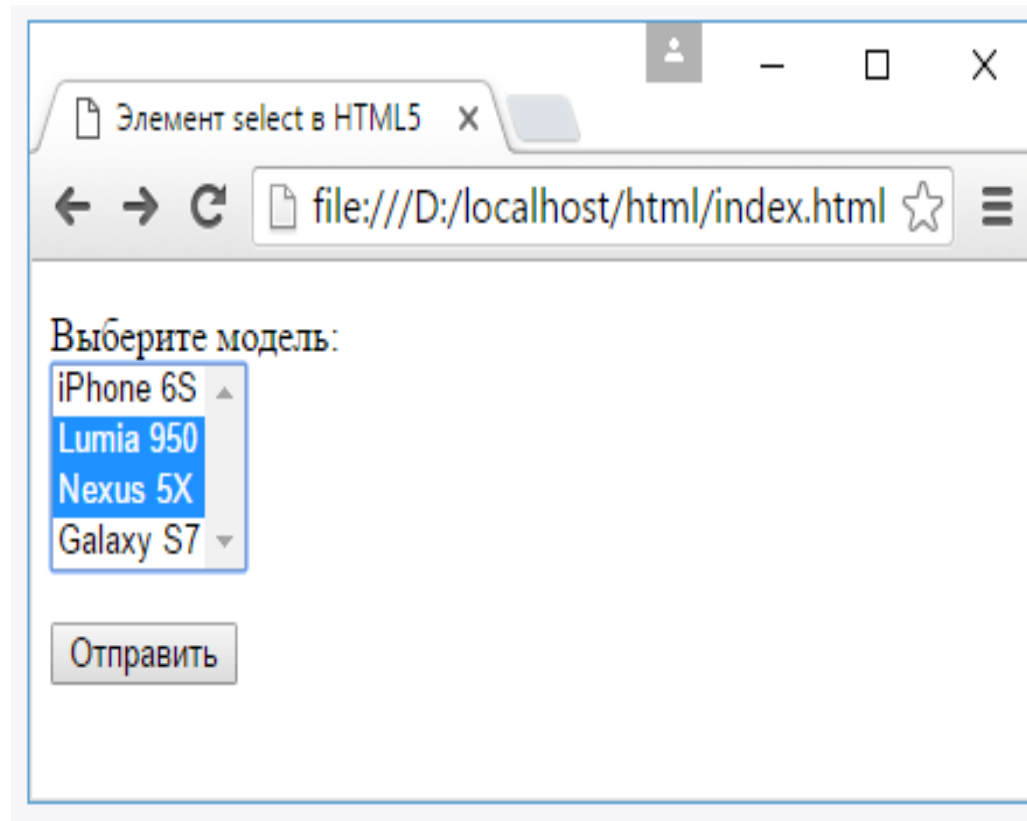
```
        <option value="galaxy s7">Galaxy S7</option>
```

```
    </select>        </p>
```

```
    <p>        <input type="submit" value="Отправить" />        </p>
```

```
</form>
```

Зажав клавишу Ctrl, мы можем выбрать в таком списке несколько элементов:



Select также позволяет группировать элементы с помощью тега <optgroup>:

```
<form method="get">
```

```
  <p>      <label for="phone">Выберите модель:</label>
```

```
    <select id="phone" name="phone">
```

```
      <optgroup label="Apple">
```

```
        <option value="iphone 6s">iPhone 6S</option>
```

```
        <option value="iphone 6s plus">iPhone 6S Plus</option>
```

```
        <option value="iphone 5se">iPhone 5SE</option>      </optgroup>
```

```
      <optgroup label="Microsoft">
```

```
        <option value="lumia 950">Lumia 950</option>
```

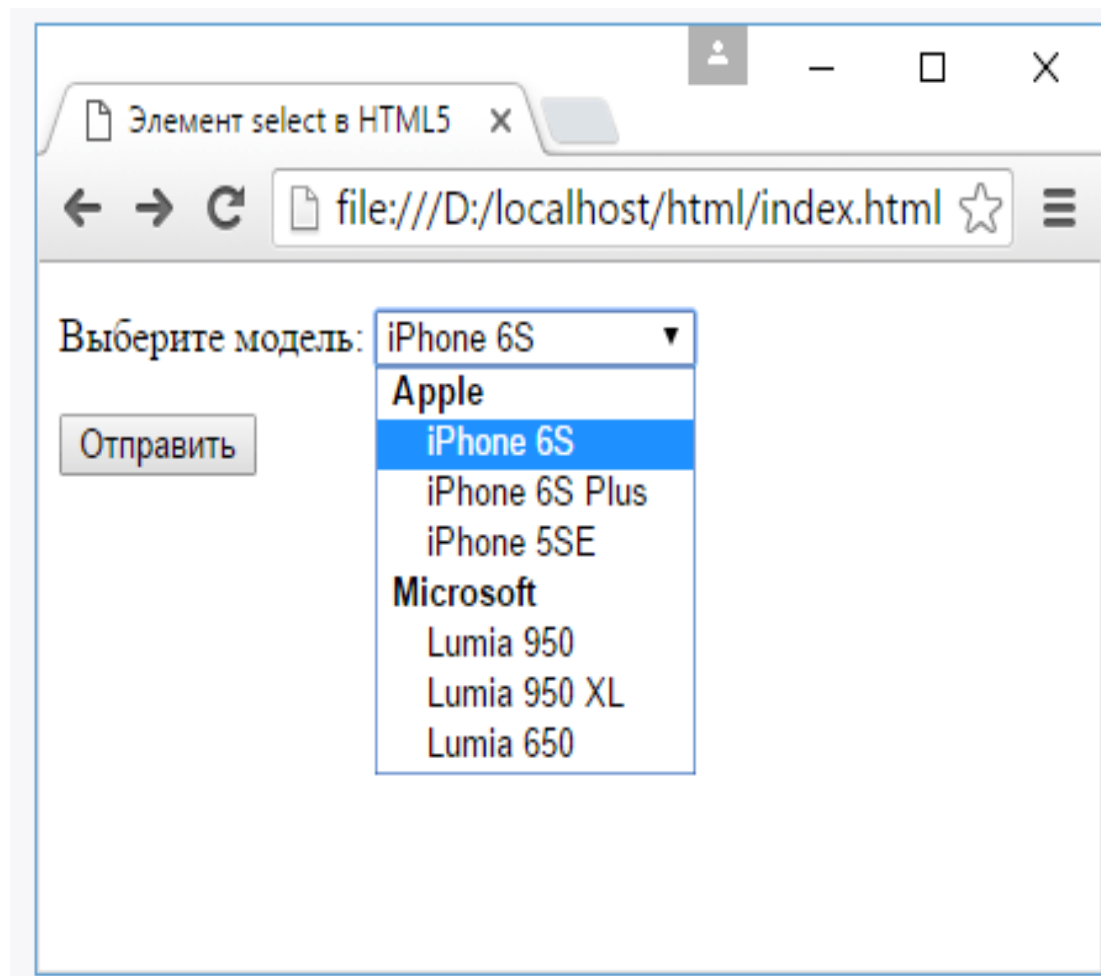
```
        <option value="lumia 950 xl">Lumia 950 XL</option>
```

```
        <option value="lumia 650">Lumia 650</option>      </optgroup>
```

```
    </select>      </p>
```

```
    <p>      <input type="submit" value="Отправить" />      </p>
```

```
</form>
```

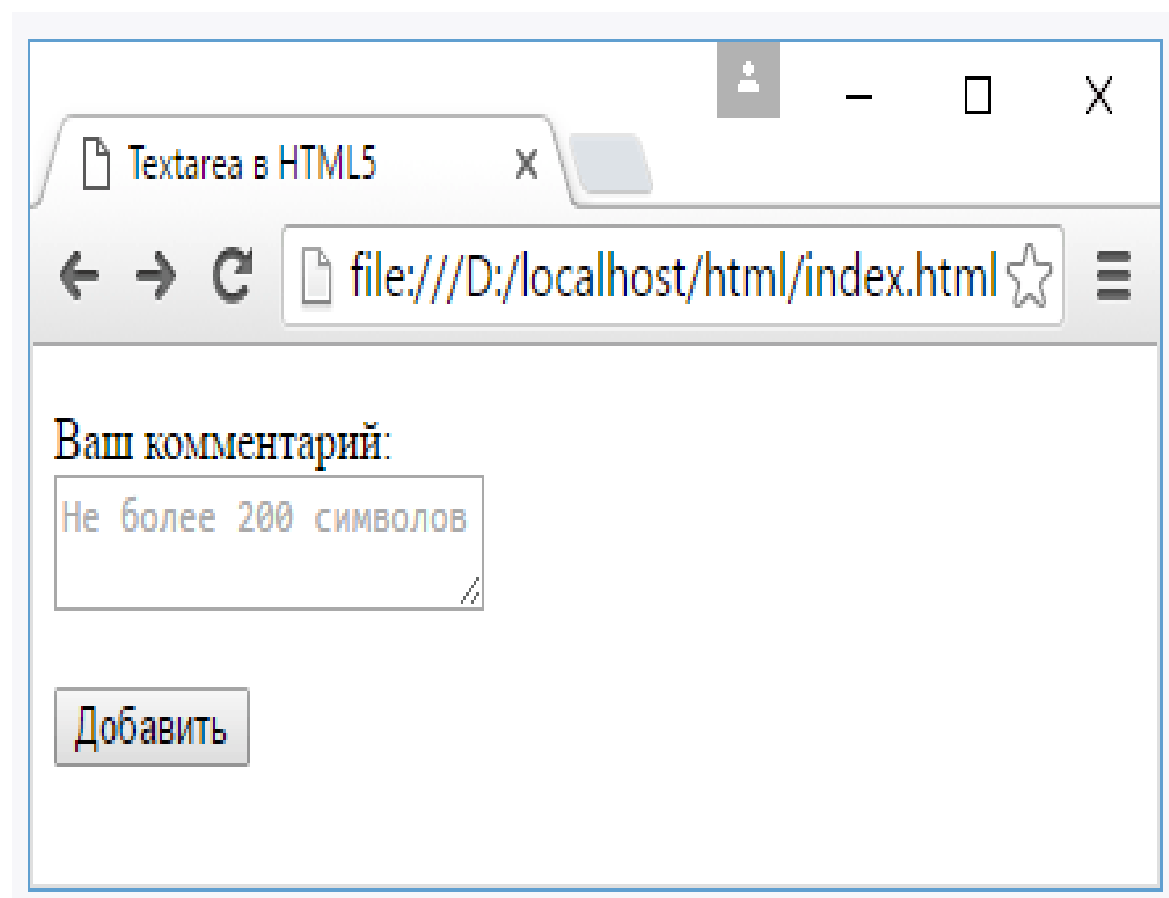


Textarea

- Элемент `<input type="text" />` позволяет создавать простое однострочное текстовое поле

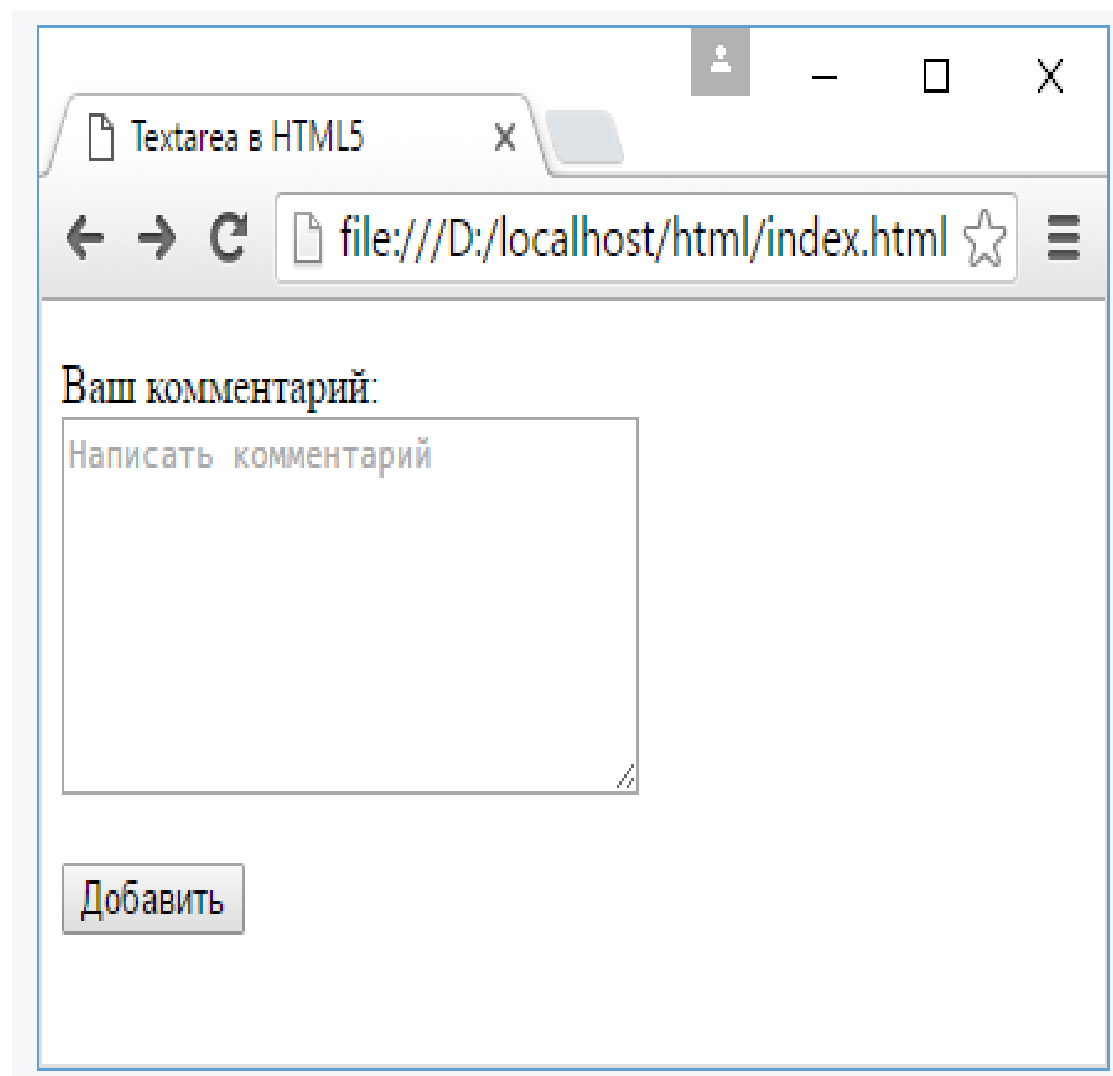
Хотя возможностей этого элемента по вводу текста бывает недостаточно, и в этой ситуации можно использовать многострочное текстовое поле, представленное элементом `textarea`:

```
<form method="get">
    <p>        <label for="comment">Ваш
комментарий:</label><br/>
        <textarea name="comment" id="comment" placeholder="Не
более 200 символов" maxlength="200"></textarea>        </p>
    <p>        <input type="submit" value="Добавить" />    </p>
</form>
```



С помощью дополнительных атрибутов cols и rows можно задать соответственно количество столбцов и строк:

```
<form method="get">
  <p>
    <label for="comment">Ваш комментарий:</label><br/>
    <textarea id="comment" name="comment"
placeholder="Написать комментарий"
    cols="30" rows="7"></textarea>
  </p>
  <p>
    <input type="submit" value="Добавить" />
  </p>
</form>
```



Валидация форм

- для предупреждения и проверки некорректного ввода в HTML5 существует механизм валидации.
- Преимущество использования валидации в HTML5 заключается в том, что пользователь после некорректного ввода может сразу получить сообщение об ошибке и внести соответствующие изменения в введенные данные.

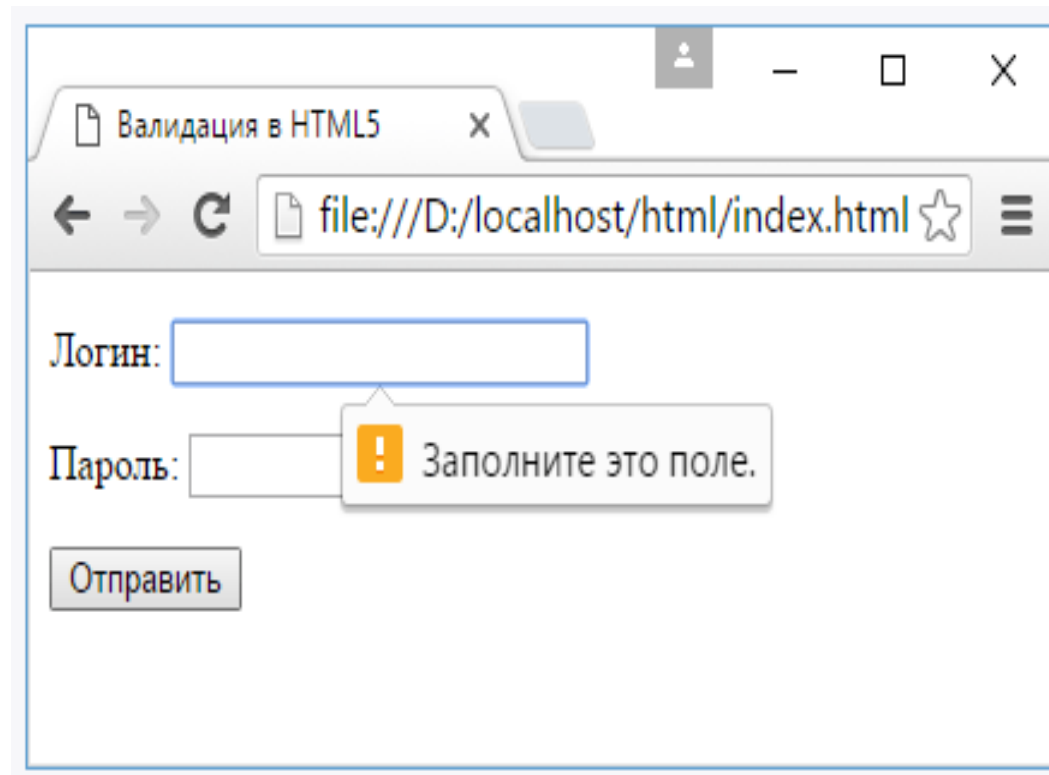
Для создания валидации у элементов форм HTML5 используется ряд атрибутов:

- **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`

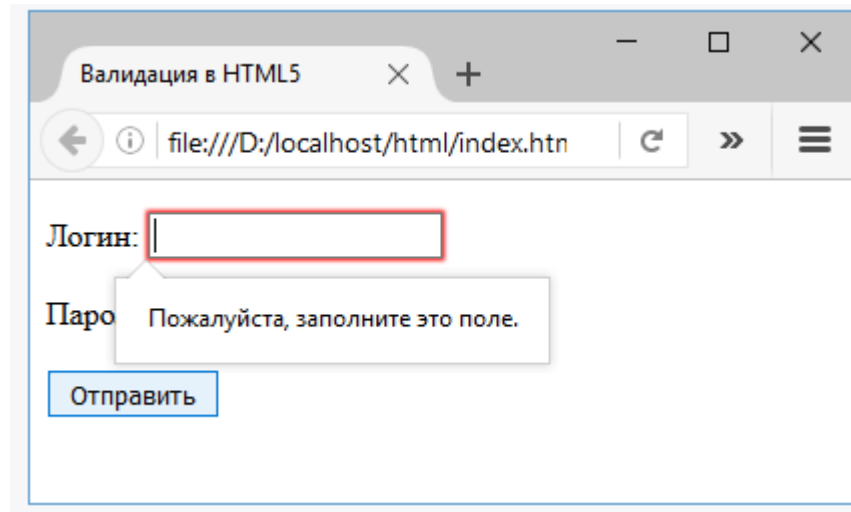
Атрибут required

```
<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>
```

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

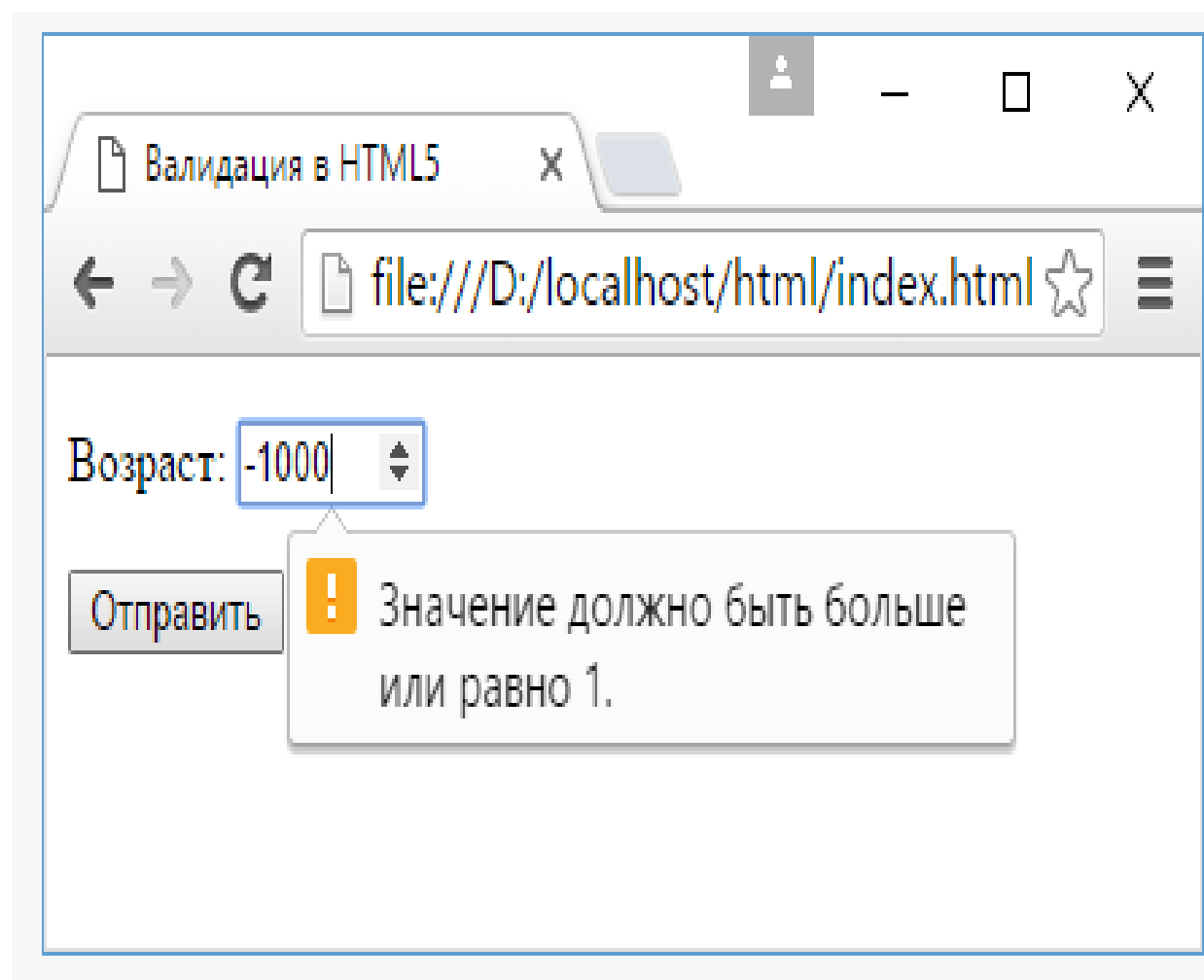


В зависимости от браузера визуализация сообщения может несколько отличаться. Также границы некорректного поля ввода могут окрашиваться в красный цвет. Например, поведение при отправке пустых сообщений в Firefox:



Атрибуты max и min

```
<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>
```



Атрибут pattern

- Атрибут pattern задает шаблон, которому должны соответствовать данные. Для определения шаблона используется язык так называемых **регулярных выражений**.

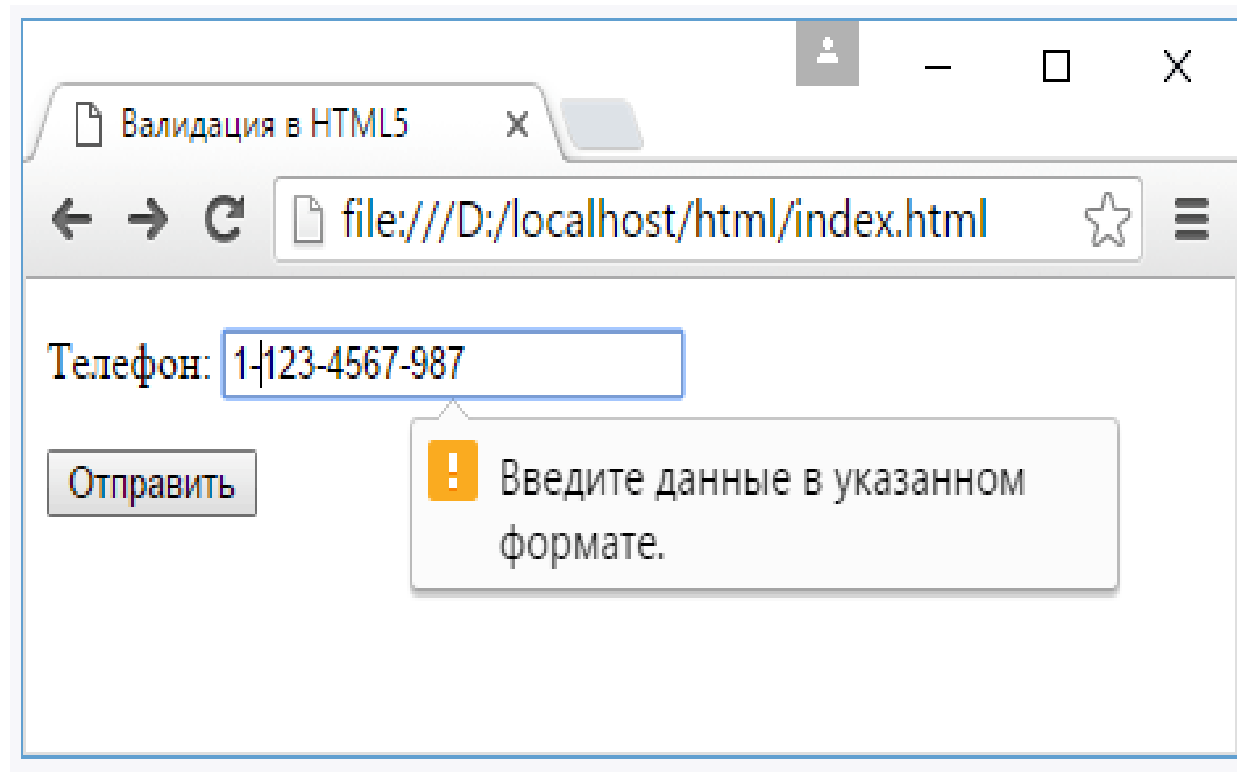
пример

```
<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>
```

Здесь для ввода номера телефона используется регулярное выражение `\+\d-\d{3}-\d{3}-\d{4}`.

- Оно означает, что первым элементом в номере должен идти знак плюс +.
- Выражение `\d` представляет любую цифру от 0 до 9.
- `\d{3}` означает три подряд идущих цифры, а `\d{4}` - четыре цифры подряд.
- То есть это выражение будет соответствовать номеру телефона в формате "+1-234-567-8901".

Если мы введем данные, которые не соответствуют этому шаблону, и нажмем на отправку, то браузер отобразит ошибку:



Отключение валидации

используется либо у элемента формы атрибут novalidate, либо у кнопки отправки атрибут formnovalidate:

```
<form novalidate 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" formnovalidate />    </p>
```

```
</form>
```

Элементы fieldset и legend

- Для группировки элементов формы нередко применяется элемент fieldset. Он создает границу вокруг вложенных элементов, как бы создавая из них группу. Вместе с ним используется элемент legend, который устанавливает заголовок для группы элементов

пример

```
<form>
```

```
  <fieldset>
```

```
    <legend>Введите данные:</legend>
```

```
    <label for="login">Логин:</label><br>
```

```
    <input type="text" name="login" id="login" /><br>
```

```
    <label for="password">Пароль:</label><br>
```

```
    <input type="password" name="password" id="password" /><br>
```

```
    <input type="submit" value="Авторизация">
```

```
  </fieldset>
```

```
</form>
```

Элементы форм

file:///D:/localhost/htn

Вход на сайт

Введите данные:

Логин:

Пароль:

Авторизация