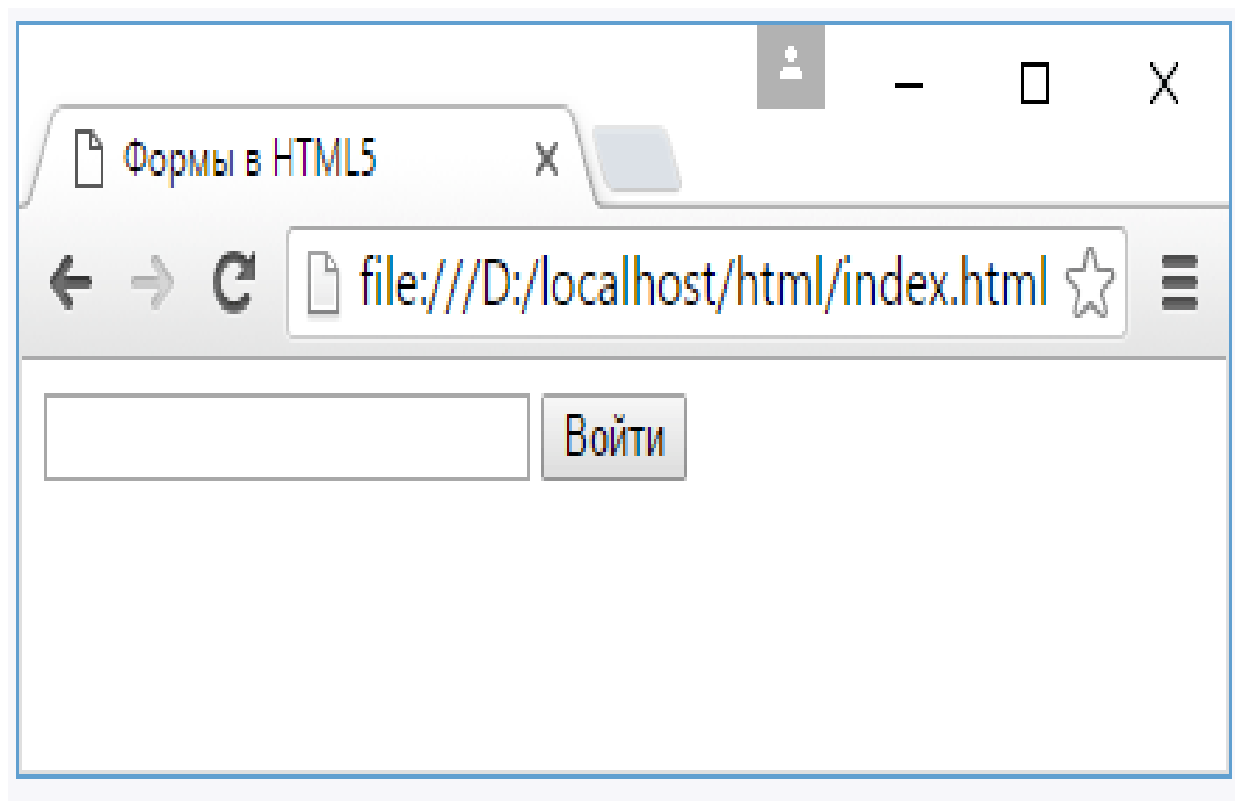


Работа с формами

Формы

- Формы в html представляют один из способов для ввода и отправки данных. Все поля формы помещаются между тегами `<form>` и `</form>`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Формы в HTML5</title>
  </head>
  <body>
    <form method="post" action="http://localhost:8080/login.php">
      <input name="login"/>
      <input type="submit" value="Войти" />
    </form>
  </body>
</html>
```



Для настройки форм у элемента form определены следующие атрибуты:

- `method`: устанавливает метод отправки данных на сервер. Допустимы два значения: `post` и `get`.
- Значение `post` позволяет передать данные на веб-сервер через специальные заголовки. А значение `get` позволяет передать данные через строку запроса.
- `action`: устанавливает адрес, на который передаются данные формы
- `enctype`: устанавливает тип передаваемых данных. Он свою очередь может принимать следующие значения:
- `application/x-www-form-urlencoded`: кодировка отправляемых данных по умолчанию
- `multipart/form-data`: эта кодировка применяется при отправке файлов
- `text/plain`: эта кодировка применяется при отправке простой текстовой информации

метод "post"

- В примере у формы установлен метод "post", то есть все значения формы отправляются в теле запроса, а адресом служит строка `http://localhost:8080/login.php`.
- Адрес здесь указан случайным образом и по указанному адресу работает веб-сервер, который, используя одну из технологий серверной стороны (PHP, NodeJS, ASP.NET и т.д.), может получать запросы и возвращать ответ.

Автодополнение

- Часто веб-браузеры запоминают вводимые данные, и при вводе браузеры могут выдавать список подсказок из ранее введенных слов, при необходимости его можно отключить с помощью атрибута `autocomplete` :
- `<form method="post" autocomplete="off" action="http://localhost:8080/login.php">`
- `<input name="login" />`
- `<input name="password" />`
- `<input type="submit" value="Войти" />`
- `</form>`

- Если нам надо включить автодополнение только для каких-то определенных полей, то мы можем применить к ним атрибут `autocomplete="on"`:
- `<form method="post" autocomplete="off" action="http://localhost:8080/login.php">`
- `<input name="login" />`
- `<input name="password" autocomplete="on" />`
- `<input type="submit" value="Войти" />`
- `</form>`

Элементы форм

- Формы состоят из определенного количества элементов ввода.
- Наиболее распространенным элементом ввода является элемент `input`.

атрибут type элемента формы input.

- text: обычное текстовое поле
- password: тоже текстовое поле, только вместо вводимых символов отображаются звездочки, поэтому в основном используется для ввода пароля
- radio: радиокнопка или переключатель. Из группы радиокнопок можно выбрать только одну
- checkbox: элемент флажок, который может находиться в отмеченном или неотмеченном состоянии
- hidden: скрытое поле

- `submit`: кнопка отправки формы
- `color`: поле для ввода цвета
- `date`: поле для ввода даты
- `datetime`: поле для ввода даты и времени с учетом часового пояса
- `datetime-local`: поле для ввода даты и времени без учета часового пояса

- email: поле для ввода адреса электронной почты
- month: поле для ввода года и месяца
- number: поле для ввода чисел
- range: ползунок для выбора числа из некоторого диапазона
- tel: поле для ввода телефона

- `time`: поле для ввода времени
- `week`: поле для ввода года и недели
- `url`: поле для ввода адреса url
- `file`: поле для выбора отправляемого файла
- `image`: создает кнопку в виде картинки

Кроме элемента `input` в различных модификациях есть еще небольшой набор элементов, которые также можно использовать на форме:

- `button`: создает кнопку
- `select`: выпадающий список
- `label`: создает метку, которая отображается рядом с полем ввода
- `textarea`: многострочное текстовое поле

Атрибуты name и value

- У всех элементов ввода можно установить атрибуты name и value. Эти атрибуты имеют важное значение. По атрибуту name можно идентифицировать поле ввода, а атрибут value позволяет установить значение поля ввода

например

```
...<form method="get" action="index.html">
```

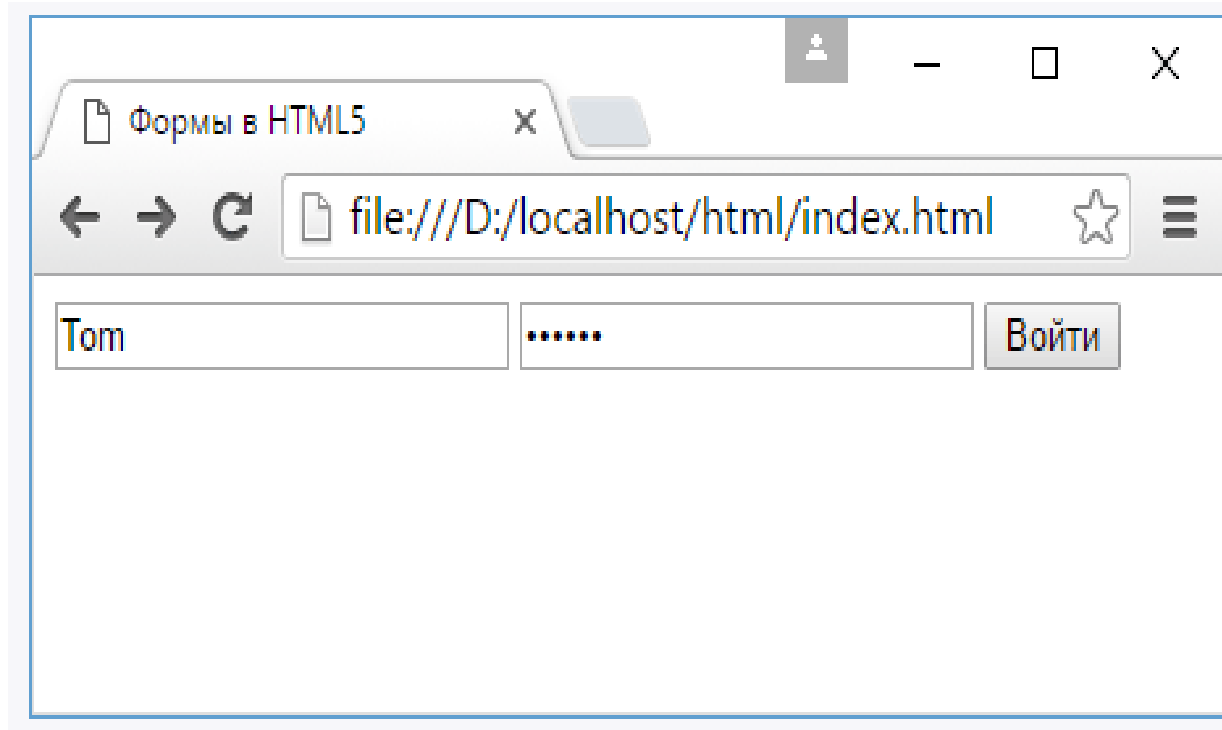
```
  <input type="text" name="login" value="Tom"/>
```

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

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

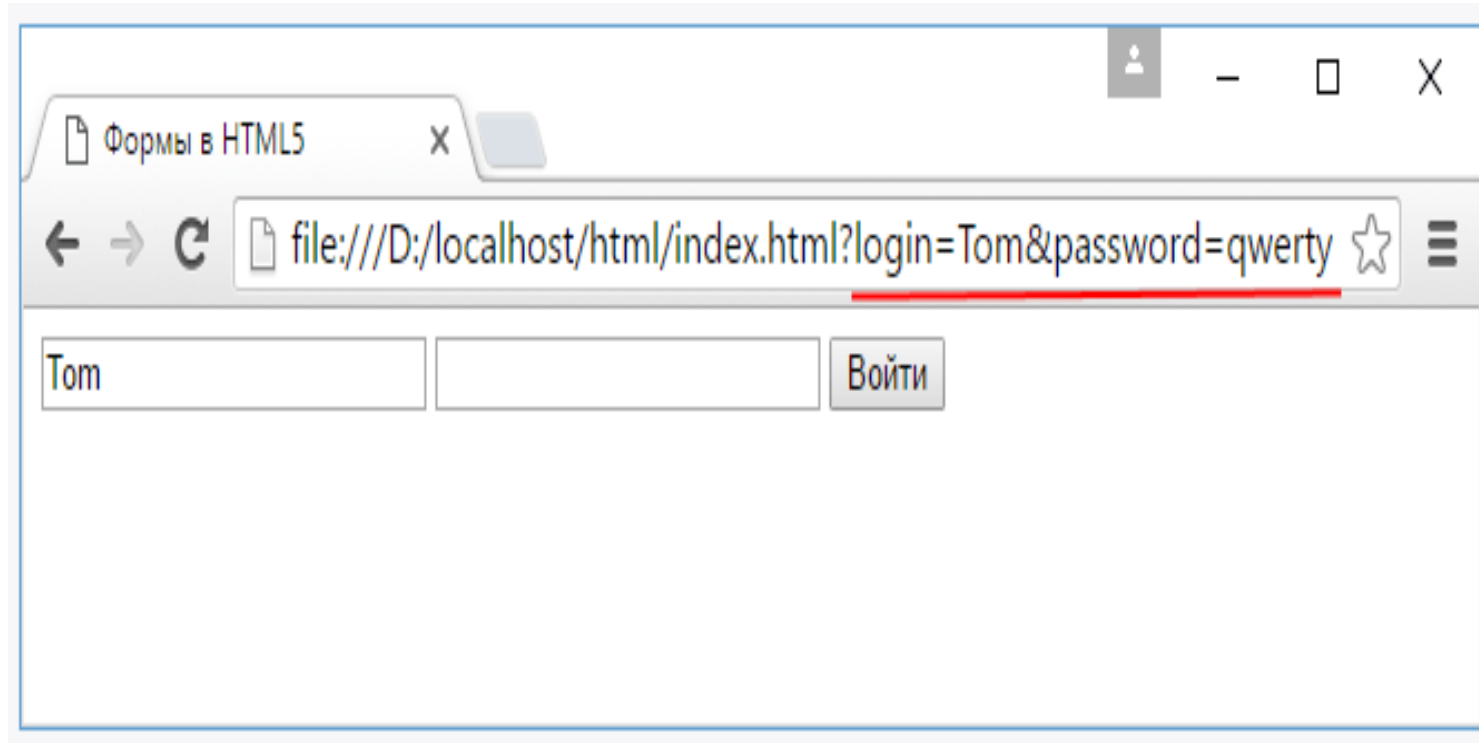
```
</form>...
```


Т к текстовое поле имеет значение "Tom" (как указано в атрибуте value), поэтому при загрузке веб-страницы в этом поле увидим данный текст.



- Т к методом отправки данных формы является метод "get", то данные будут отправляться через строку запроса.
- Поэтому нам в данном случае не важно, как данные будут приниматься, не важен сервер, который получает данные, следовательно в качестве адреса установим ту же самую страницу - то есть файл index.html.

При отправке можно увидеть введенные данные в строке запроса:



`login=Tom&password=qwerty` - данная запись при отправке формы браузер соединяет все данные в набор пар "ключ-значение".

В примере две таких пары:

`login=Tom`

`password=qwerty.`

Ключом в этих парах выступает название поля ввода, которое определяется атрибутом `name`, а значением - собственно то значение, которое введено в поле ввода (или значение атрибута `value`).

Получив эти данные, сервер легко может узнать, какие значения в какие поля ввода были введены пользователем.

Кнопки

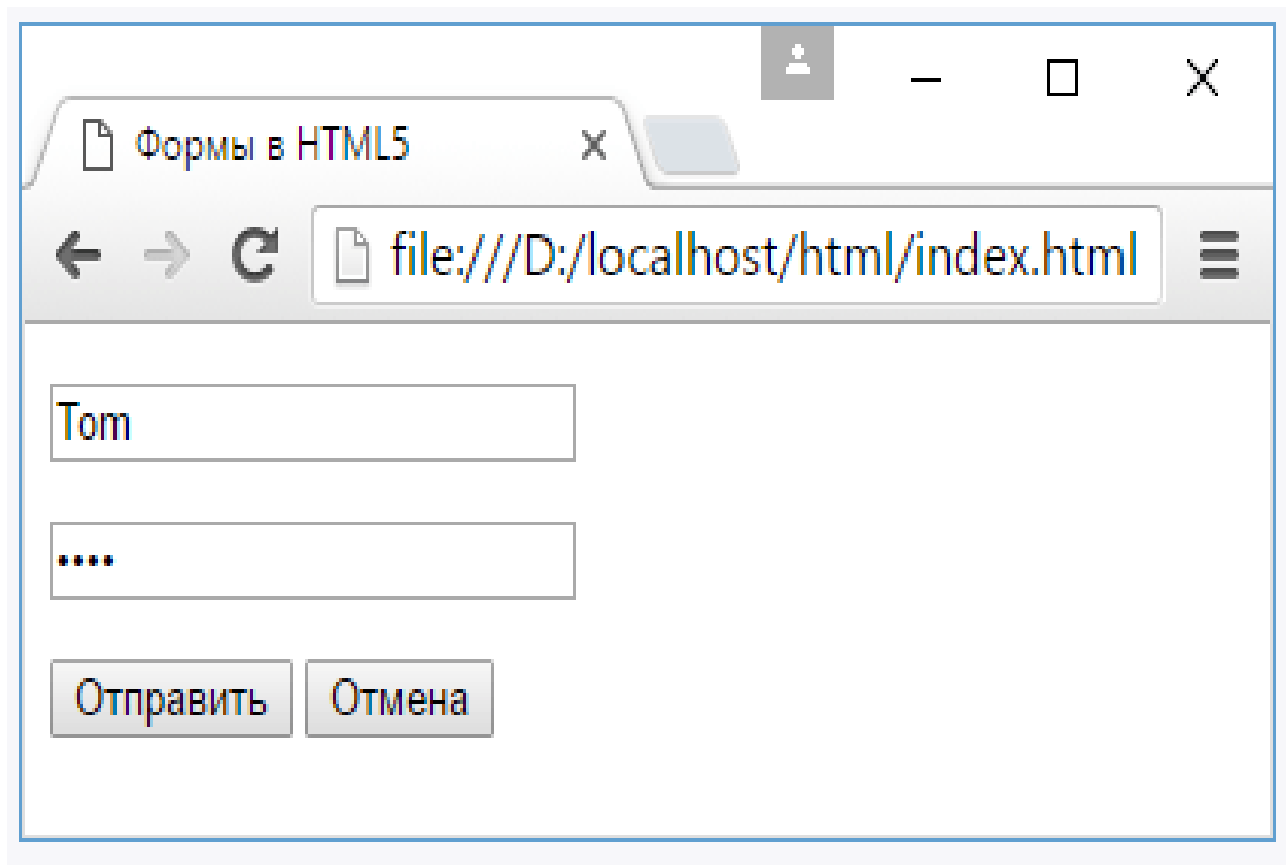
- Кнопки представлены элементом `button`. Они обладают широкими возможностями по конфигурации. Так, в зависимости от значения атрибута `type` мы можем создать различные типы кнопок:
- `submit`: кнопка, используемая для отправки формы
- `reset`: кнопка сброса значений формы
- `button`: кнопка без какого-либо специального назначения

Если кнопка используется для отправки формы, то есть у нее установлен атрибут `type="submit"`, то можно задать у нее ряд дополнительных атрибутов:

- `form`: определяет форму, за которой закреплена кнопка отправки
- `formaction`: устанавливает адрес, на который отправляется форма. Если у элемента `form` задан атрибут `action`, то он переопределяется
- `formenctype`: устанавливает формат отправки данных. Если у элемента `form` установлен атрибут `enctype`, то он переопределяется
- `formmethod`: устанавливает метод отправки формы (`post` или `get`). Если у элемента `form` установлен атрибут `method`, то он переопределяется

Например, определим на форме кнопку отправки и кнопку сброса:

```
... <form>
    <p><input type="text" name="login"/></p>
    <p><input type="password" name="password"/></p>
    <p>
    <button type="submit" formmethod="get" formaction="index.html">
Отправить</button>
        <button type="reset">Отмена</button>
    </p>
</form>...
```



Кроме элемента `button` для создания кнопок можно использовать элемент `input`, у которого атрибут равен `submit` или `reset`. Например:

```
...<form>
```

```
  <p><input type="text" name="login"/></p>
```

```
  <p><input type="password" name="password"/></p>
```

```
  <p>
```

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

```
    <input type="reset" value="Отмена" />
```

```
  </p>
```

```
</form>...
```

И еще один элемент `input` с атрибутом `type="image"` позволяет использовать в качестве кнопки изображение:

```
...<form>
```

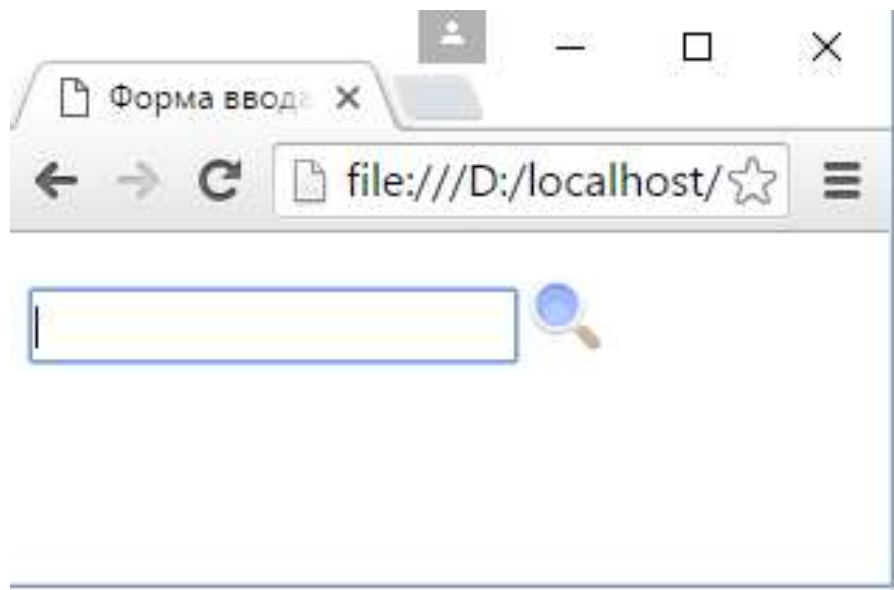
```
  <p>
```

```
    <input type="text" name="search" />
```

```
    <input type="image" src="search.png" name="submit" />
```

```
  </p>
```

```
</form>...
```



Кроме наличия изображения в остальном эта кнопка будет аналогична стандартной кнопке отправки `input type="submit"` или `button type="submit"`.

Текстовые поля

- Однострочное текстовое поле создается с помощью элемента `input`, когда его атрибут `type` имеет значение `text`:

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

С помощью ряда дополнительных атрибутов можно настроить текстовое поле:

- `dirname`: устанавливает направление текста
- `maxlength`: максимально допустимое количество символов в текстовом поле
- `pattern`: определяет шаблон, которому должен соответствовать вводимый текст
- `placeholder`: устанавливает текст, который по умолчанию отображается в текстовом поле
- `readonly`: делает текстовое поле доступным только для чтения
- `required`: указывает, что текстовое поле обязательно должно иметь значение
- `size`: устанавливает ширину текстового поля в видимых символах
- `value`: устанавливает значение по умолчанию в текстовом поле

например

```
<form>
```

```
    <p><input type="text" name="userName" placeholder="Введите имя"
size="18" /></p>
```

```
    <p><input type="text" name="userPhone" placeholder="Введите номер
телефона" size="18" maxlength="11" /></p>
```

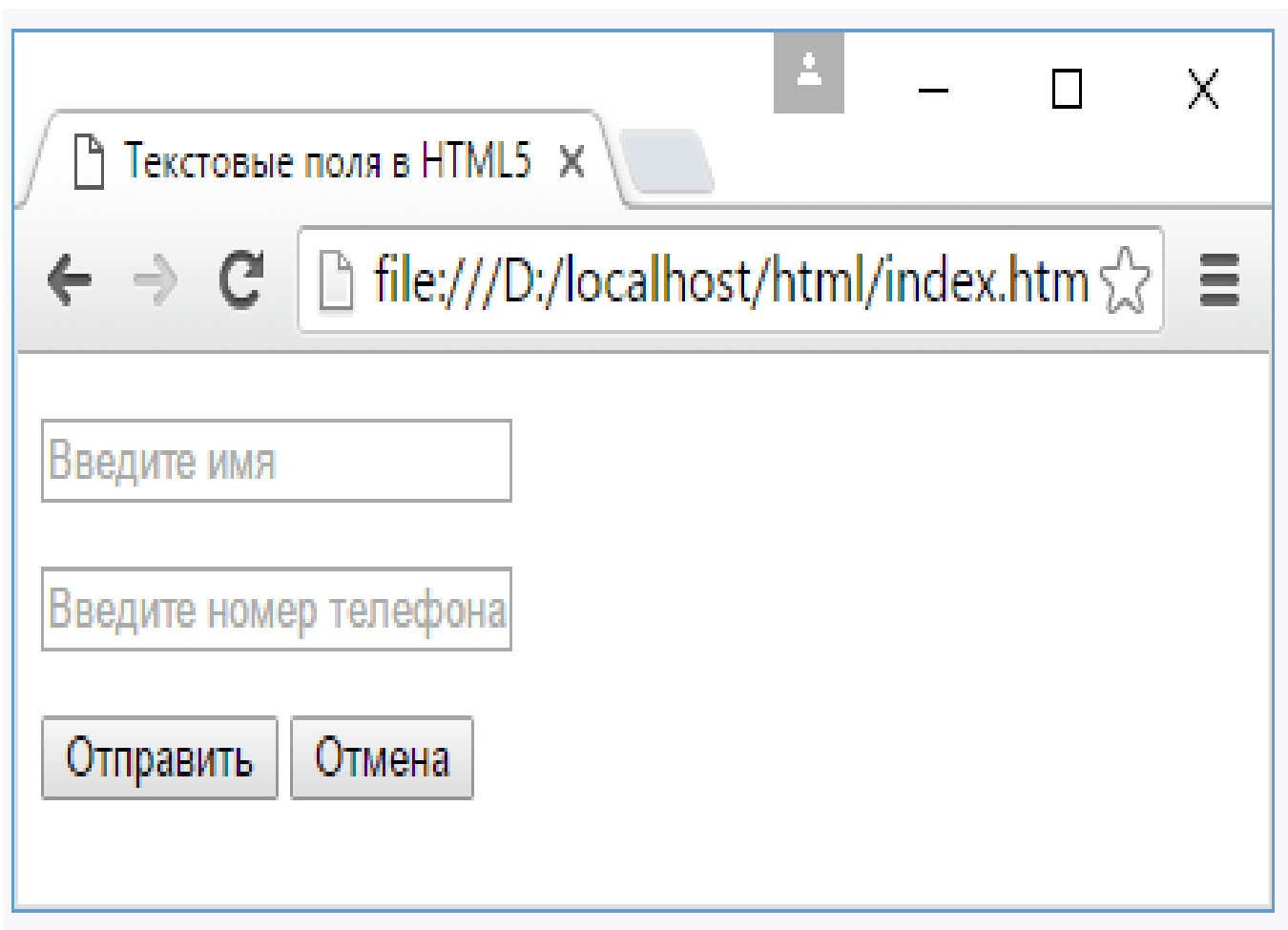
```
    <p>
```

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

```
        <button type="reset">Отмена</button>
```

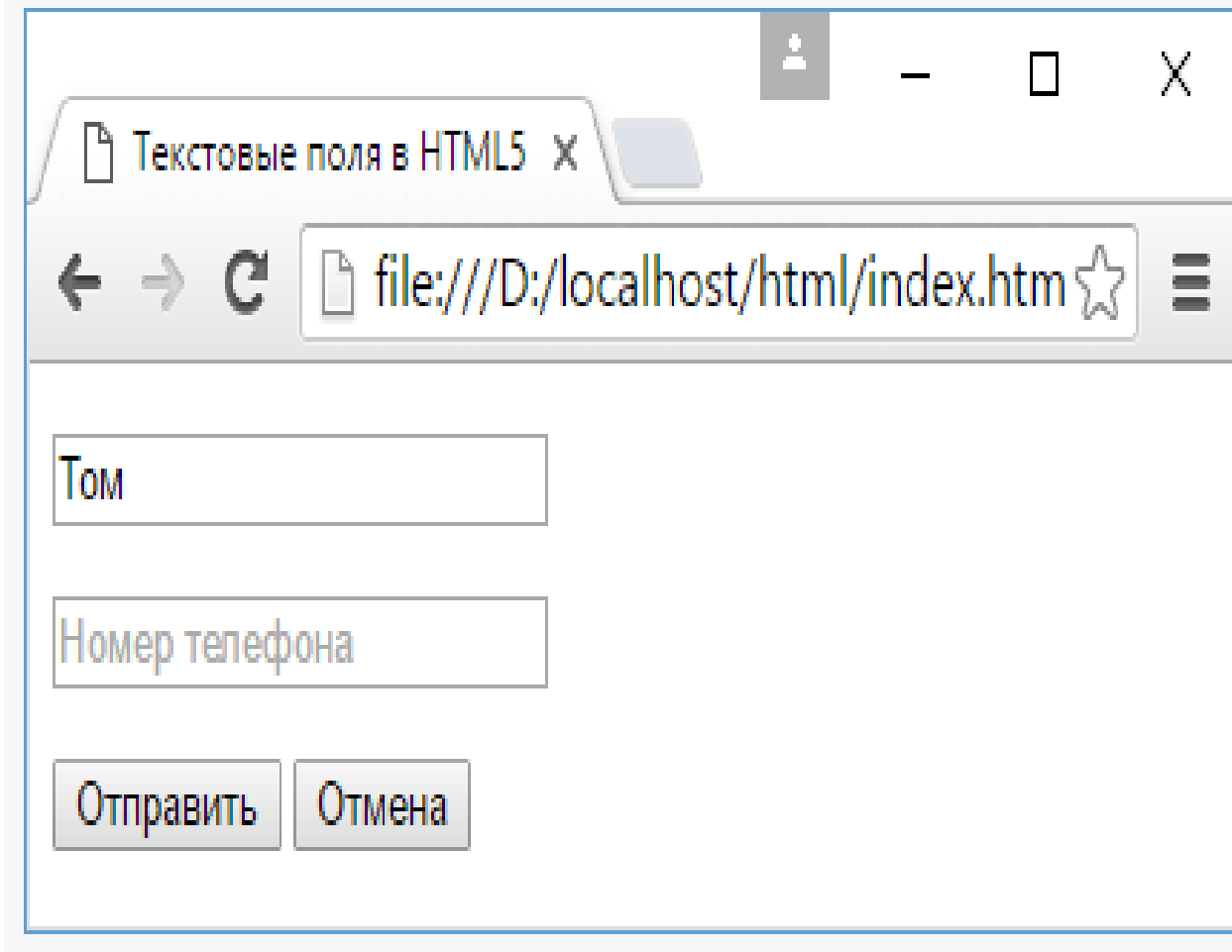
```
    </p>
```

```
</form>
```



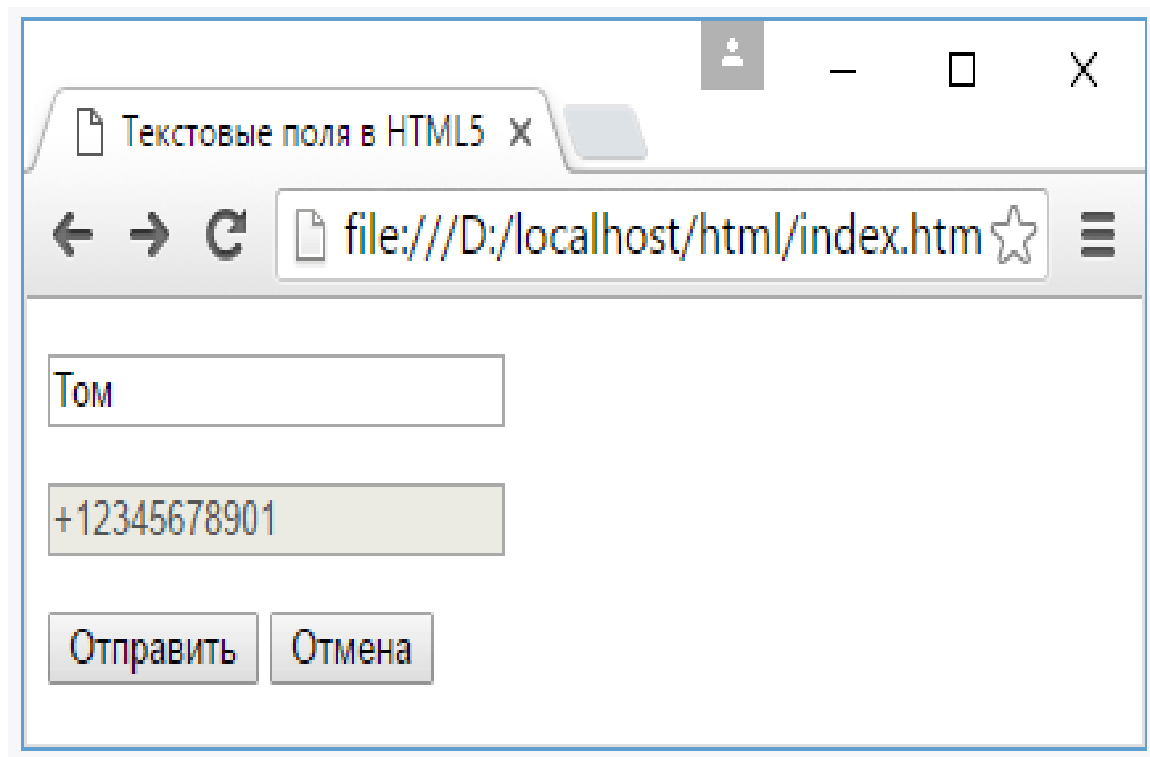
- Здесь во втором текстовом поле сразу устанавливаются атрибуты `maxlength` и `size`. При этом `size` - то есть количество символов, которые помещаются в видимое пространство поля больше, чем допустимое количество символов. Однако все равно ввести символов больше, чем `maxlength` не сможем.

- Важно различать атрибуты `value` и `placeholder`, хотя оба устанавливают видимый текст в поле. Однако `placeholder` устанавливает своего рода подсказку или приглашение к вводу, поэтому он обычно отмечается серым цветом. В то время как значение `value` представляет введенный в поле текст по умолчанию:
- `<p><input type="text" name="userName" value="Том" /></p>`
- `<p><input type="text" name="userPhone" placeholder="Номер телефона" /></p>`



Атрибуты `readonly` и `disabled` делают текстовое поле недоступным, однако сопровождаются разным визуальным эффектом. В случае с `disabled` текстовое поле затемняется:

```
<p><input type="text" name="userName" value="Том" readonly /></p>  
<p><input type="text" name="userPhone" value="+12345678901"  
disabled /></p>
```



Среди атрибутов текстового поля также следует отметить такой атрибут как **list**. Он содержит ссылку на элемент **datalist**, который определяет набор значений, появляющихся в виде подсказки при вводе в текстовое поле. Например:

```
<form>
```

```
  <p><input list="phonesList" type="text" name="model"
placeholder="Введите модель" /></p>
```

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

```
</form>
```

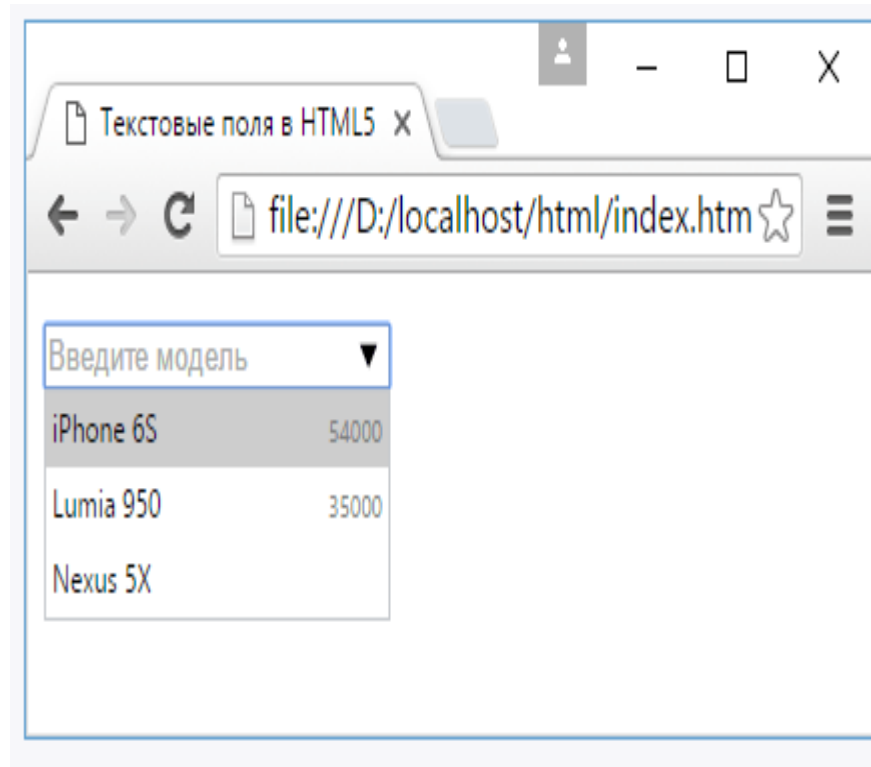
```
<datalist id="phonesList">
```

```
  <option value="iPhone 6S" label="54000"/>
```

```
  <option value="Lumia 950">35000</option>
```

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

```
</datalist>
```

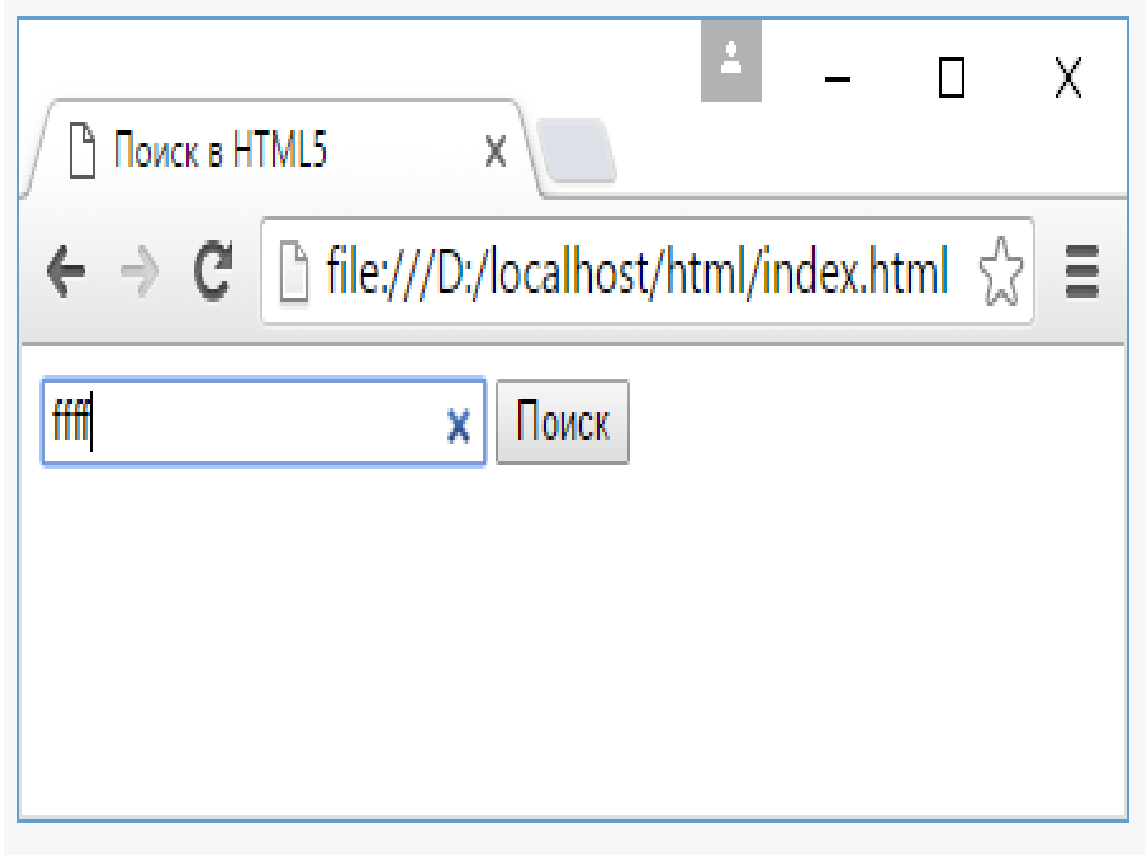


Атрибут `list` текстового поля указывает на `id` элемента `datalist`. Сам элемент `datalist` с помощью вложенных элементов `option` определяет элементы списка. И при вводе в текстовое поле этот список отображается в виде подсказки.

Поле поиска

Для создания полей поиска предназначен элемент `input` с атрибутом `type="search"`. Формально он представляет собой простое текстовое поле:

```
... <title>Поиск в HTML5</title>
</head>
<body>
<form>
  <input type="search" name="term" />
  <input type="submit" value="Поиск" />
</form>...
```



Поле ввода пароля

Для ввода пароля используется элемент `input` с атрибутом `type="password"`. Его отличительной чертой является то, что вводимые символы маскируются точками:

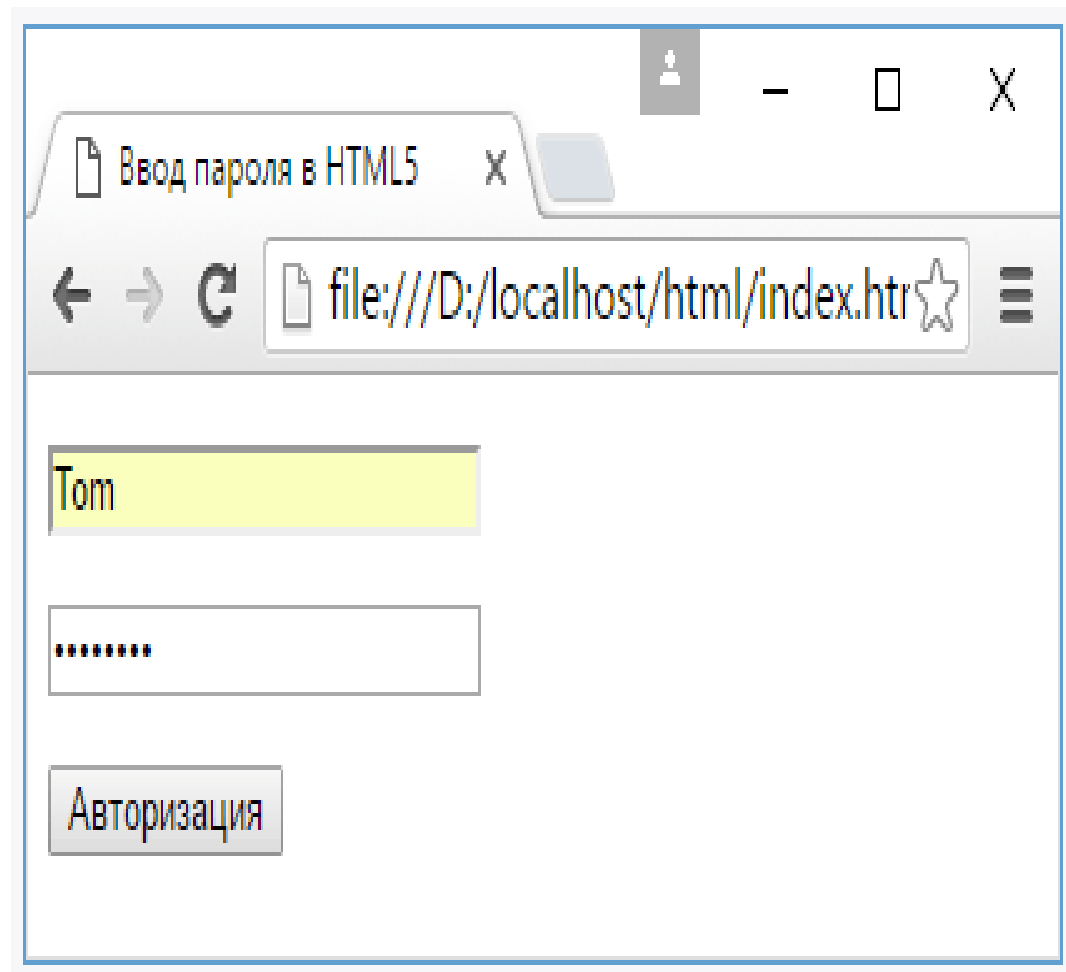
```
<form>
```

```
  <p><input type="text" name="login" /></p>
```

```
  <p><input type="password" name="password" /></p>
```

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

```
</form>
```

Метки и автофокус

- Вместе с полями ввода нередко используются метки, которые представлены элементом label. Метки создают аннотацию или заголовков к полю ввода, указывают, для чего это поле предназначено.

Для связи с полем ввода метка имеет атрибут for, который указывает на id поля ввода:

```
<form>
```

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

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

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

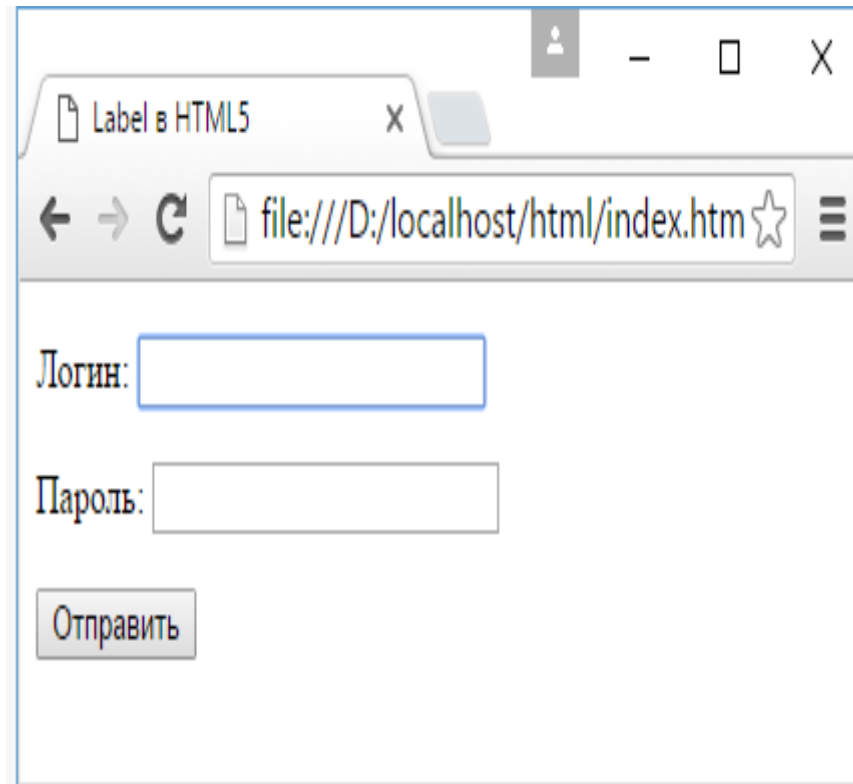
```
        <input type="password" id="password" name="password" />
```

```
</p>
```

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

```
</form>
```

Текстовое поле здесь имеет атрибут `id="login"`. Поэтому у связанной с ним метки устанавливается атрибут `for="login"`. Нажатие на эту метку позволяет перевести фокус на текстовое поле для ввода логина:



Также можно установить автофокус по умолчанию на какое-либо поле ввода. Для этого применяется атрибут `autofocus`, где при запуске страницы фокус сразу же переходит на текстовое поле.:

```
<form>
  <p>    <label for="login">Логин: </label>
    <input type="text" autofocus id="login" name="login" />    </p>
  <p>    <label for="password">Пароль: </label>
    <input type="password" id="password" name="password" />    </p>
  <p>    <button type="submit">Отправить</button>    </p>
</form>
```