

Глава 9. Формы

Формы являются наиболее популярным способом "обратной связи" с пользователем. С помощью HTML можно создавать как простые формы, предполагающие выбор одного из нескольких ответов, так и сложные формы для заказов или для того, чтобы получить от пользователей страницы какие-либо комментарии и пожелания.

Форма представляет собой несколько полей, где пользователь может ввести некоторую информацию, либо выбрать какой-то пункт. После того, как пользователь отправит информацию, она обрабатывается программой (скриптом), размещенной на сервере. Существует также возможность обрабатывать формы "на стороне клиента", встраивая в свои страницы скрипты, написанные на языках JavaScript и Visual Basic Script.

Формы передают информацию программам-обработчикам в виде пар [имя переменной]=[значение переменной]. Имена переменных следует задавать латинскими буквами. Значения переменных воспринимаются обработчиками как строки, даже если они содержат только цифры.

Принцип работы форм следующий: посетитель, зашедший на страницу, заполняет форму, а после нажатия определенной кнопки форма берет данные из заполненных полей и отправляет их в назначенное место.

HTML-документ может содержать в себе несколько форм, но они не должны находиться одна внутри другой.

Для внесения информации пользователем в форму используются поля ввода. Поля ввода бывают:

- однострочные текстовые поля,
- многострочные текстовые поля,
- поля пароля,
- переключатели,
- радиокнопки,
- кнопки отправки данных,
- скрытые поля,
- загрузки файла,
- кнопки изображения и т.д.

9.1. Создание формы

Форма создается элементом **form**, имеющим несколько атрибутов.

Синтаксис
<pre><form action="адрес" method=метод enctype=кодирование> содержание формы </form></pre>

Атрибут **action** является обязательным и содержит адрес обработчика формы (сервера).

Атрибут **method** определяет, каким образом данные из формы будут переданы обработчику. Допустимые значения: **method=post** и **method=get** (установлено по умолчанию).

Атрибут **enctype** определяет, каким образом данные из формы будут закодированы для передачи обработчику (по умолчанию **enctype=application/x-www-form-urlencoded**).

9.1.1. Метод отправки формы

Атрибут **method** имеет два значения:

- **get**: HTTP набор данных формы присоединяется к URI, определённому атрибутом **action** (со знаком "?" в качестве разделителя), и этот новый URI высылаётся программе-обработчику.
- **post**: HTTP набор данных формы включается в тело формы и высылаётся программе-обработчику.

Метод "**get**" нужно использовать, если форма цельна (напр., не осуществляет побочных действий). Многие средства поиска в базах данных не имеют видимых побочных действий и являются идеальными приложениями для использования данного метода.

Если сервис, связанный с обработчиком формы, имеет побочный эффект (например, если форма модифицирует базу данных или подписку на сервис), должен использоваться метод "**post**".

Метод "**get**" ограничивает набор данных формы символами ASCII. Только метод "**post**" (с **enctype="multipart/form-data"**) определен для использования с полным набором символов [ISO10646].

9.1.2. Типы содержимого формы

Атрибут **enctype** определяет тип содержимого, используемый при кодировке набора данных формы для отправки на сервер.

Возможные значения:

- **application/x-www-form-urlencoded**. Это тип содержимого по умолчанию. Формы, отправленные с содержимым этого типа, должны кодироваться так: имена элементов управления и их значения переводятся в **escape**-последовательности. Символы пробелов заменяются "+", и затем зарезервированные символы переводятся в **escape**-последовательности. Неалфавитные символы заменяются кодом типа "%HH" (знаком процента и двумя 16-ричными цифрами, представляющими ASCII-код символа). Символы новой строки представляются парами "CR LF" (например, "%0D%0A"). Пары имя/значение перечисляются в том порядке, в котором они появляются в документе. Имя отделено от значения знаком "=" и пары имя/значение отделены друг от друга знаком "&".
- **multipart/form-data**. Используется для отправки форм, содержащих файлы, не-ASCII данные и бинарные данные. Сообщение "**multipart/form-data**" содержит несколько частей, каждая из которых представляет "действующий" элемент управления. Эти части пересылаются обработчику в том же порядке, в каком соответствующие элементы управления появляются в потоке документа.

9.2. Однострочное поле ввода

Использование элемента **input** для ввода текста ограничено одной строкой.

Синтаксис
<pre><input type=тип name=имя value="значение" checked size="размер" maxlength="длина" src="адрес" align=выравнивание></pre>

Атрибут **name** определяет наименование поля.

Атрибут **value** инициализирует текстовую строку в поле, когда документ впервые загружен.

Атрибут **size** устанавливает видимый размер поля, например, **size="40"** для 40-символьного поля.

Атрибут **maxlength** задает максимальное количество отображаемых символов в поле.

Атрибут **type** определяет тип поля ввода. Может иметь значения:

тип	Пояснение
text	<i>однострочное текстовое поле.</i> Пользователи могут вводить и больше символов, чем предел установленный атрибутами size и maxlength , но с текстовым скроллингом (пролистыванием) поля, чтобы курсор ввода оставался видимым
password	<i>однострочное поле ввода пароля.</i> Подобно типу text, однако все вводимые символы представляются в виде *, чтобы скрыть текст, когда вводится пароль. Можно использовать атрибуты size и maxlength , чтобы управлять видимой длиной и максимальной длиной поля точно так же, как для обычного текстового поля
checkbox	<i>переключатели.</i> Используется для простых булевых атрибутов (т.е. атрибутов, принимающих значение истина или ложь) или для атрибутов, которые одновременно могут принимать множество значений. Каждое заполненное переключательное поле генерирует отдельную пару имя/значение в формируемых данных, даже если это приводит к дублированию имен. Следует использовать атрибут checked для инициализации поля установки по умолчанию
radio	<i>радиокнопка.</i> Используется, когда нужно выбрать единственное значение из множества. Каждое поле радиокнопки в группе должно быть задано только одним значением атрибута name . Радиокнопки требуют явного атрибута value . Единственная нажатая радиокнопка в группе генерирует пару имя/значение в формируемых данных. Одна радиокнопка в группе с атрибутом checked должна быть предварительно установлена по умолчанию
submit	<i>определяет кнопку, при нажатии на которую запускается процесс передачи данных из формы обработчику</i> (например, кнопку [Отправить]). Метка на кнопке устанавливается Атрибутом value . Если Атрибут name задан, то пара наименование/значение для исполняемой кнопки будет включена в передаваемые данные. Можно включить несколько исполняемых кнопок в форму
reset	<i>определяет кнопку, которую пользователь может нажать, чтобы вернуть поля формы к исходному состоянию, когда документ был впервые загружен.</i> Метка на кнопке устанавливается Атрибутом value . Кнопки перезагрузки никогда не посылаются как часть содержимого формы
file	<i>этот тип дает возможность пользователям прикрепить файл к содержимому формы.</i> Элемент обычно отображается, как текстовое поле с кнопкой, при нажатии

	<p>на которую появляется файловый менеджер для отбора имени файла. Имя файла также может быть введено прямо в текстовое поле.</p> <p>Можно использовать Атрибут size, чтобы установить видимую ширину поля. Можно установить верхний предел для длины имени файла, используя Атрибут maxlength. Некоторые браузеры ограничивают виды файлов, которые могут быть прикреплены к форме, перечислением разделяемого запятыми списка файлов с содержимым типа MIME, задаваемого Атрибутом accept. Например, accept="image/*" ограничит файлы изображениями</p>
hidden	<p><i>этот тип поля не отображается пользователю.</i> Скрытое поле дает возможность для серверов хранить информацию о состоянии вместе с формой. Когда форма "исполняется" при нажатии соответствующей кнопки, серверу будет передана пара имя/значение, определенная с использованием соответствующих Атрибутов. Этот тип создает рабочее окружение для полноты возможностей http</p>
button	<p><i>создает кнопку.</i> Для метки кнопки необходимо использовать атрибут value.</p>
image	<p><i>используется для графических кнопок отправки данных формы серверу</i> (графическая кнопка submit). Адрес изображений специфицируется Атрибутом src. Выравнивание изображения может быть специфицировано Атрибутом align (по умолчанию - bottom). Значения Атрибута указаны в главе Графика (таблица 20). Если для щелчка на кнопке image используется указательное устройство (мышь и т.п.), форма отправляется и координаты щелчка передаются серверу. Значение X измеряется в пикселах от левого края изображения, а значение Y - в пикселах от верхнего края изображения. Отправленные данные включают значения <code>name.x=x-value</code> и <code>name.y=y-value</code>, где "name" это значение Атрибута name, а <code>x-value</code> и <code>y-value</code> это значения координат <code>x</code> и <code>y</code> соответственно.</p>

Таблица 23. Типы полей ввода

Пример:

```

Содержимое файла sample9.2-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Формы.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <form method="post" action="mailto:mail@mail.ru?subject=student">
      <pre>
        Имя: <input type="text" name="username" value="" size=20 align=left>
        Пароль: <input type="password" name="pass" size=15 maxlength=20 align=left>
        Пол:
          <input type="radio" name="question" value="male">Мужской
      </pre>
    </form>
  </body>
</html>

```

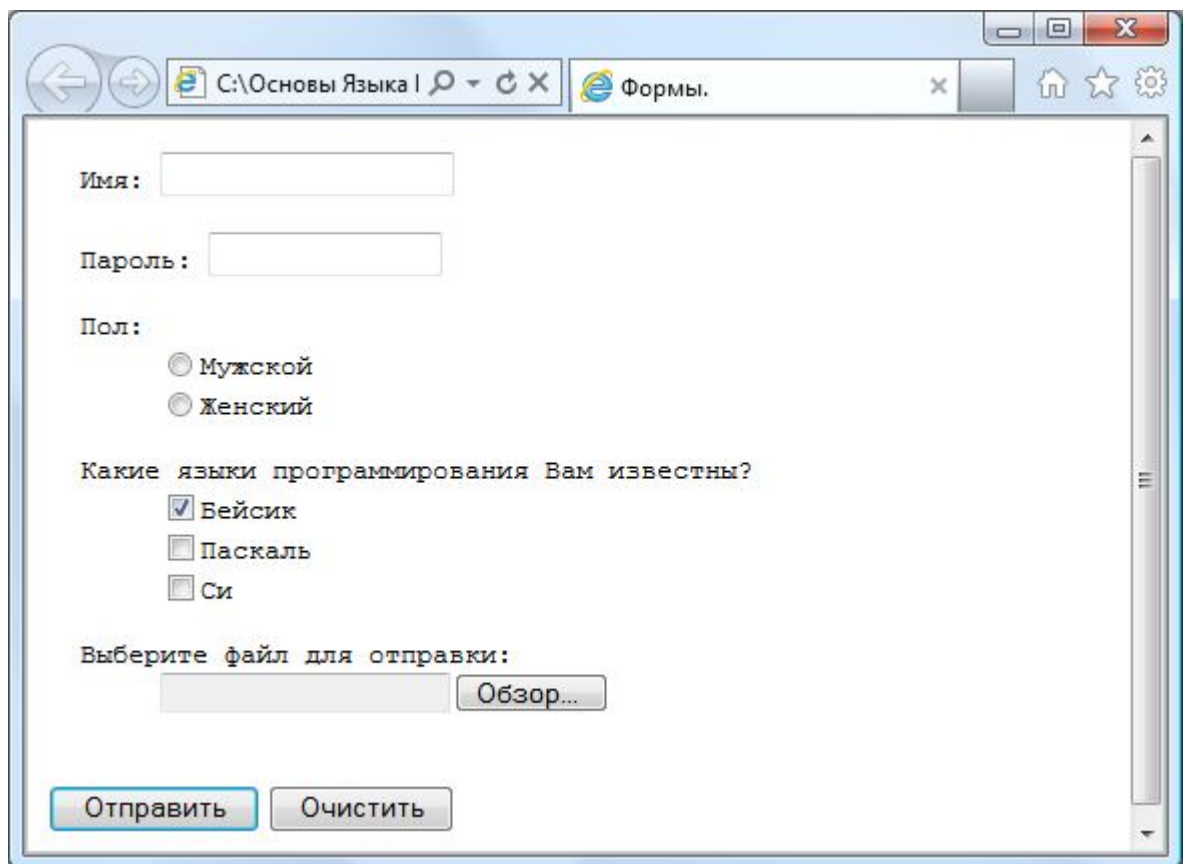
```

        <input type=radio name=question value="female">Женский
Какие языки программирования Вам известны?
        <input type=checkbox name=lang value="Бейсик" checked>Бейсик
        <input type=checkbox name=lang value="Паскаль">Паскаль
        <input type=checkbox name=lang value="Си">Си
    Выберите файл для отправки:
        <input type=file>
</pre>

<p><input type=submit value="Отправить">
    <input type=reset value="Очистить"></p>
</form>
</body>
</html>

```

Результат на экране:



9.3. Кнопка. Элемент BUTTON

Синтаксис
<code><button name="имя" value="значение" type=тип></code>
Содержимое
<code></button></code>

Атрибут **name** определяет имя кнопки.

Атрибут **value** устанавливает начальное значение кнопки. Значение кнопки отсылается на сервер при отправке формы

Атрибут **type** объявляет тип кнопки.

Возможные значения:

назначение	Пояснение
<code>submit</code>	создаёт кнопку <code>submit</code> . Это значение по умолчанию.
<code>reset</code>	создаёт кнопку <code>reset</code> .
<code>button</code>	создаёт кнопку.

Таблица 24. Значения атрибута `type`

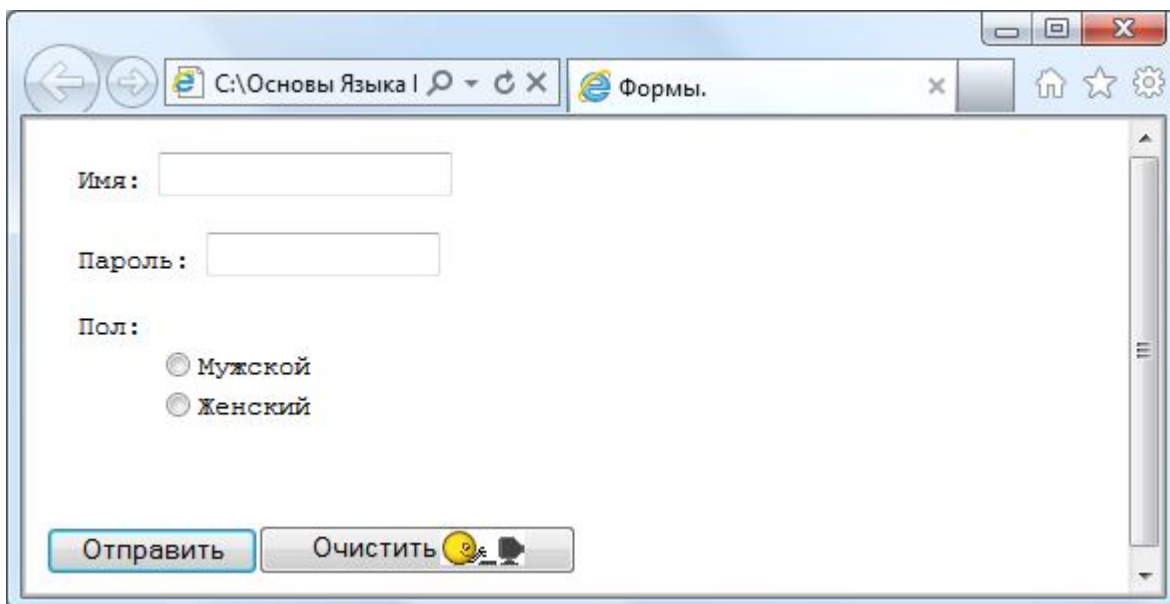
Кнопки, созданные элементом `button` работают почти так же, как и кнопки, созданные элементом `input`, но дают больше возможностей представления - элемент `button` может иметь содержимое. Например, элемент `button`, содержащий изображение, работает и может выглядеть как элемент `input`, чей атрибут `type` установлен `"image"`, но элемент `button` может иметь содержимое.

Визуальные браузеры могут представлять кнопку `button` с рельефом и смещением вверх/вниз при нажатии, в то время как они могут рассматривать кнопки `input` только как "плоские" изображения.

Пример:

```
Содержимое файла sample9.3-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Формы.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <form method="post" action="mailto:mail@mail.ru?subject=student">
      <pre>
        Имя: <input type="text" name="username" value="" size=20 align=left>
        Пароль: <input type="password" name="pass" size=15 maxlength=20 align=left>
        Пол:
          <input type="radio" name="question" value="male">Мужской
          <input type="radio" name="question" value="female">Женский
      </pre>
      <p>
        <button name="submit" type="submit" value="Отправить"> Отправить
      </button>
        <button name="reset" type="reset" value="Очистить">Очистить
        </button>
      </p>
    </form>
  </body>
</html>
```

Результат на экране:



9.4. Многострочное поле ввода

Область ввода, появляющаяся в виде отдельного прямоугольника и обычно со скроллингом как в вертикальном, так и в горизонтальном направлении. Для ввода большого текста используется элемент многострочного поля ввода:

Синтаксис

```
<textarea name=имя rows="целое" cols="целое">
первоначальный текст
</textarea>
```

Первоначальный текст используется для инициализации и выводится в поле при первой загрузке документа.

Атрибут **name** определяет наименование поля, которое используется для идентификации поля с текстовым содержимым, когда форма передается серверу.

Атрибут **rows** устанавливает количество строк видимого текста.

Атрибут **cols** устанавливает видимую ширину поля ввода.

Пример:

Содержимое файла sample9.4-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Формы.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <form method="post" action="mailto:mail@mail.ru?subject=student">
      <pre>
        Имя: <input type="text" name="username" value="" size=20 align=left>

        Пароль: <input type="password" name="pass" size=15 maxlength=20 align=left>

        Пол:
```

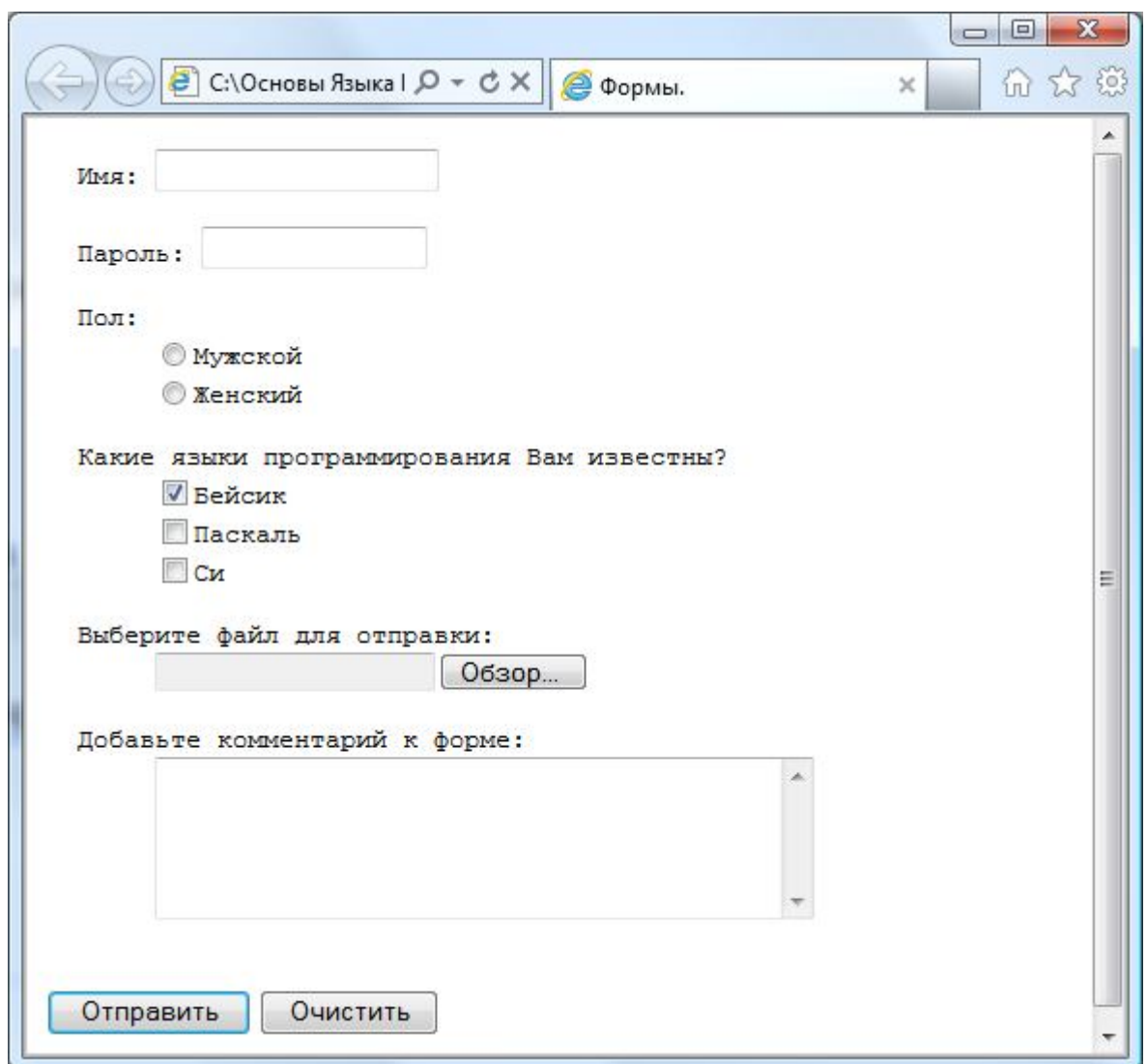
```
<input type=radio name=question value="male">Мужской
<input type=radio name=question value="female">Женский

Какие языки программирования Вам известны?
<input type=checkbox name=lang value="Бейсик" checked>Бейсик
<input type=checkbox name=lang value="Паскаль">Паскаль
<input type=checkbox name=lang value="Си">Си

Выберите файл для отправки:
<input type=file>

Добавьте комментарий к форме:
<textarea name=comment rows="5" cols="40">
</textarea>
</pre>
<p><input type=submit value="Отправить">
<input type=reset value="Очистить"></p>
</form>
</body>
</html>
```

Результат на экране:



9.5. Меню выбора

Меню выбора, обычно в графических браузерах это означает ниспадающее меню. В зависимости от браузера все варианты выбора могут быть видимы одновременно или пользователь может просматривать элементы меню.

Синтаксис
<code><select name=имя size=размер multiple></code> пункты меню <code></select></code>

Атрибут **name** определяет наименование поля, которое используется для идентификации, когда форма передается серверу. Каждый отобранный пункт образует пару имя/значение, включаемую в содержимое формы.

Атрибут **size** устанавливает число одновременно видимых пунктов меню и используется, когда задан атрибут **multiple**.

Атрибут **multiple** означает, что пользователь может сделать множественный выбор из меню.

9.5.1. Пункты меню выбора

Для создания одного пункта меню выбора служит специальный элемент:

Синтаксис
<code><option value="строка" selected>текст</option></code>

Когда активизировано меню выбора, включающее элементы выбора, пользователь видит текст элементов выбора (пунктов) меню, либо часть списка такого текста, либо имеет возможность просматривать текст пунктов.

Атрибут **value** устанавливает значение, которое будет передано обработчику формы, если выбран этот пункт меню.

Атрибут **selected** задает пункт меню по умолчанию.

Пример:

Содержимое файла sample9.5.1-1.html:
<pre><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html> <head> <title>Формы.</title> <meta http-equiv="Content-Type" content="text/html; charset=windows-1251"> </head> <body> <form method="post" action="mailto:mail@mail.ru?subject=student"> <pre> Имя: <input type="text" name="username" value="" size=20 align=left> Пароль: <input type="password" name="pass" size=15 maxlength=20 align=left> Пол: <input type="radio" name="question" value="male">Мужской <input type="radio" name="question" value="female">Женский Какие языки программирования Вам известны?</pre>

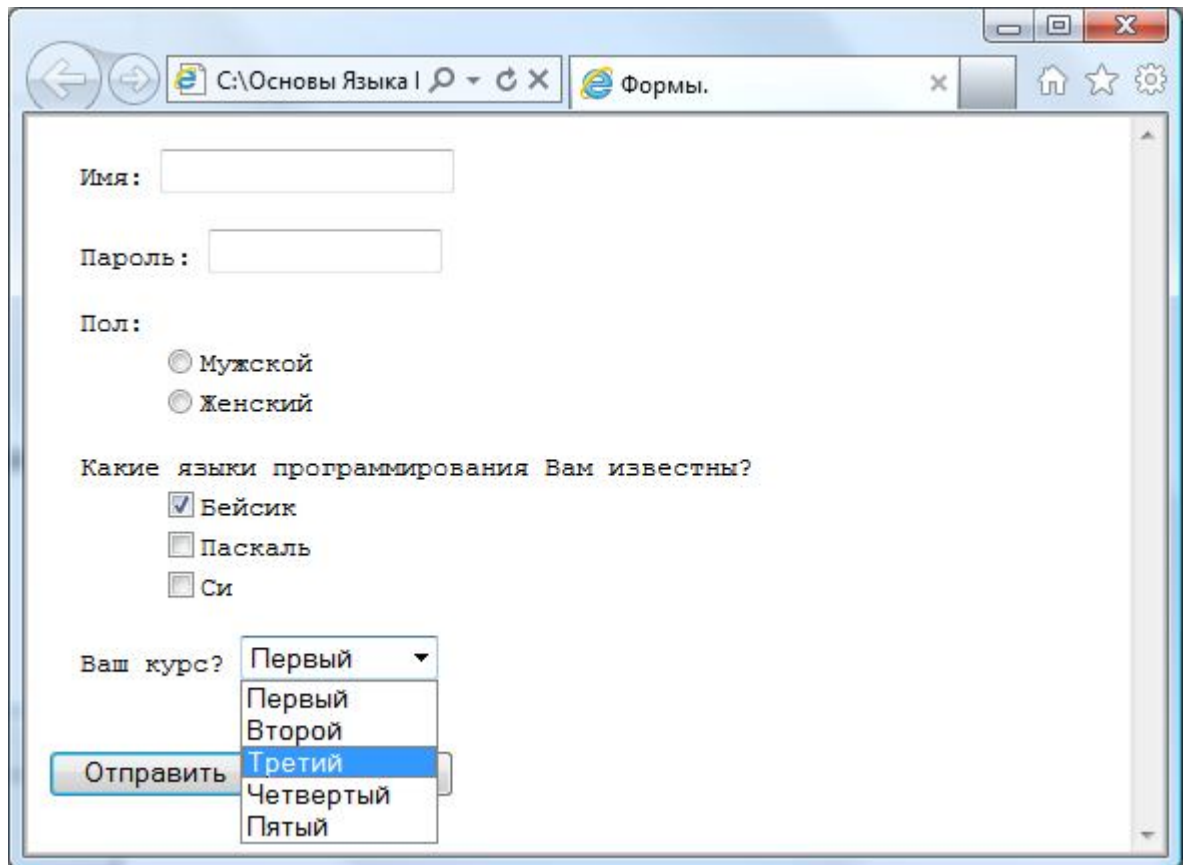
```
<input type=checkbox name=lang value="Бейсик" checked>Бейсик
<input type=checkbox name=lang value="Паскаль">Паскаль
<input type=checkbox name=lang value="Си">Си

Ваш курс? <select name=curs size="1">
  <option value="1" selected>Первый</option>
  <option value="2">Второй</option>
  <option value="3">Третий</option>
  <option value="4">Четвертый</option>
  <option value="5">Пятый</option>
</select>

</pre>

<p><input type=submit value="Отправить">
  <input type=reset value="Очистить"></p>
</form>
</body>
</html>
```

Результат на экране:



9.6. Группирование

Элемент `optgroup` позволяет группировать опции логически. Это особенно удобно, когда пользователь должен выбрать из большого списка опций: группы связанных опций легче осмыслить и запомнить, чем один большой список опций. В HTML 4 все элементы `optgroup` должны быть определены непосредственно в элементе `select` (т.е. группы не могут быть вложенными).

```
<optgroup label="заголовок">
  Элементы option
</optgroup>
```

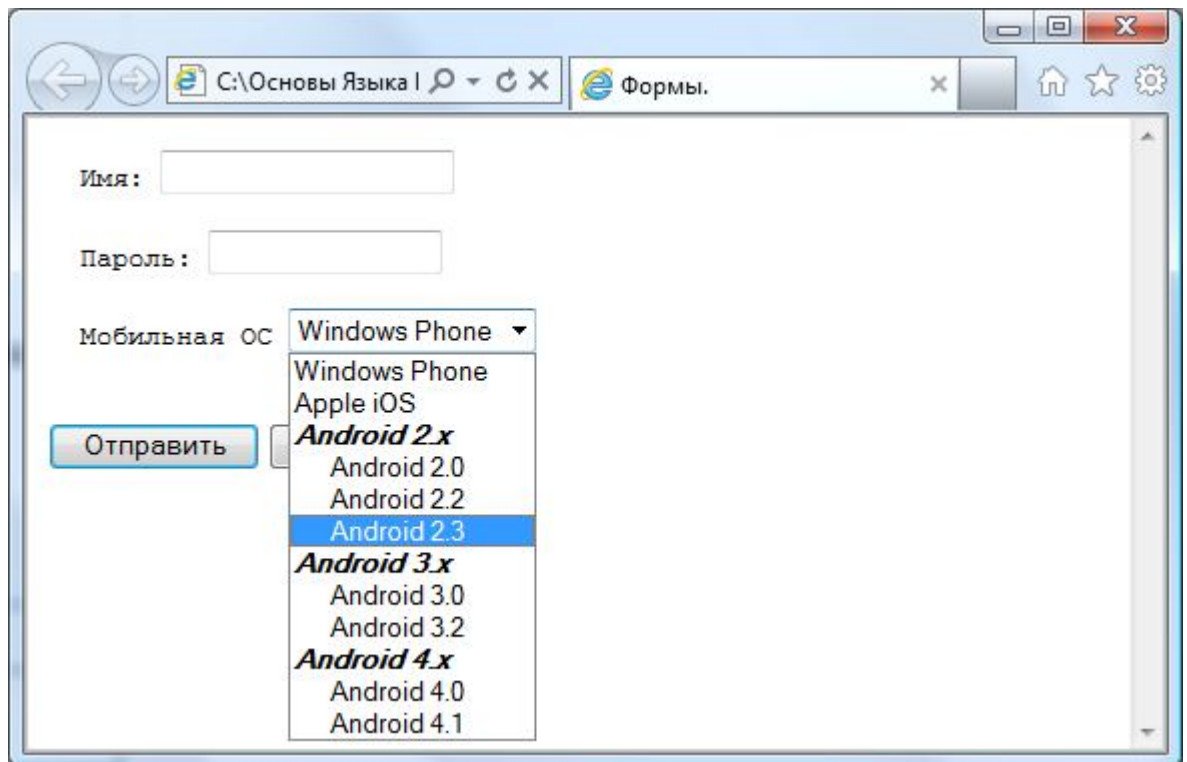
Атрибут **label** определяет метку (заголовок) для группы опций.

Пример:

Содержимое файла sample9.6-1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Формы.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <form method="post" action="mailto:mail@mail.ru?subject=student">
      <pre>
        Имя: <input type="text" name="username" value="" size=20 align=left>
        Пароль: <input type="password" name="pass" size=15 maxlength=20 align=left>
        Мобильная ОС <select name="mobOS" size="1">
          <option value="WPh" selected>Windows Phone</option>
          <option value="iOS">Apple iOS</option>
          <optgroup label="Android 2.x">
            <option value="A20">Android 2.0</option>
            <option value="A22">Android 2.2</option>
            <option value="A23">Android 2.3</option>
          </optgroup>
          <optgroup label="Android 3.x">
            <option value="A30">Android 3.0</option>
            <option value="A32">Android 3.2</option>
          </optgroup>
          <optgroup label="Android 4.x">
            <option value="A40">Android 4.0</option>
            <option value="A41">Android 4.1</option>
          </optgroup>
        </select>
      </pre>
      <p><input type="submit" value="Отправить">
        <input type="reset" value="Очистить"></p>
    </form>
  </body>
</html>
```

Результат на экране:



9.7. Структурирование формы

Элемент **fieldset** позволяет тематически группировать элементы управления и метки. Группирование элементов управления облегчает пользователю понимание их назначения, одновременно облегчая навигацию табуляцией для визуальных браузеров и речевую навигацию для браузеров, ориентированных на голосовое управление. Правильное использование этого элемента делает документы более доступными.

Синтаксис
<code><fieldset></code>
Элементы формы
<code></fieldset></code>

Элемент **legend** даёт возможность установить заглавие к **fieldset**. "Легенда" улучшает доступность, если **fieldset** просматривается незрительно.

Синтаксис
<code><fieldset></code>
<code><legend align="выравнивание">Легенда</legend></code>
Элементы формы
<code></fieldset></code>

Атрибут **align** определяет позицию "легенды" (подсказки) в том, что касается набора полей. Возможные значения:

выравнивание	Пояснение
<code>top</code>	"Легенда" находится сверху над набором полей. Это значение по умолчанию.

bottom	"Легенда" находится под набором полей.
left	"Легенда" находится слева от набора полей.
right	"Легенда" находится справа от набора полей.

Таблица 25. Значения атрибута align для элемента legend

Пример:

```

Содержимое файла sample9.7-1.html:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Формы.</title>
    <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  </head>
  <body>
    <form method="post" action="mailto:mail@mail.ru?subject=student">
      <pre>
        <fieldset>
          <legend>Персональная информация</legend>
          Имя: <input type="text" name="username" value="" size=20 align="left">

          Пароль: <input type="password" name="pass" size=15 maxlength=20 align="left">

          Пол:
            <input type="radio" name="question" value="male">Мужской
            <input type="radio" name="question" value="female">Женский
          </fieldset>

          <fieldset>
            <legend>Учебная информация</legend>
            Какие языки программирования Вам известны?
            <input type="checkbox" name="lang" value="Бейсик" checked="">Бейсик
            <input type="checkbox" name="lang" value="Паскаль">Паскаль
            <input type="checkbox" name="lang" value="Си">Си

            Ваш курс? <select name="curs" size="1">
              <option value="1" selected="">Первый</option>
              <option value="2">Второй</option>
              <option value="3">Третий</option>
              <option value="4">Четвертый</option>
              <option value="5">Пятый</option>
            </select>
          </fieldset>
        </pre>

        <p><input type="submit" value="Отправить">
          <input type="reset" value="Очистить"></p>
      </form>
    </body>
  </html>

```

Результат на экране:

The image shows a web browser window with the following elements:

- Address Bar:** Contains the text "С:\Основы Языка |" and "Формы." with navigation icons (back, forward, search, refresh, close) and browser controls (home, star, settings).
- Form Structure:**
 - Персональная информация (Personal Information):** A section containing:
 - Имя: (Name):
 - Пароль: (Password):
 - Пол: (Gender):
 - Мужской (Male)
 - Женский (Female)
 - Учебная информация (Academic Information):** A section containing:
 - Какие языки программирования Вам известны? (Which programming languages are you familiar with?):
 - Бейсик (Basic)
 - Паскаль (Pascal)
 - Си (C)
 - Ваш курс? (Your course?): (Dropdown menu)
- Buttons:** "Отправить" (Send) and "Очистить" (Clear) buttons are located at the bottom of the form.