

6. Лекция: Формы HTML и ввод данных:

Формы HTML предназначены для организации взаимодействия с пользователем. Они позволяют вводить текст, осуществлять выбор из предложенных значений при помощи списков или кнопок. С помощью форм можно организовать интерактивный обмен информацией между Web-страницей и сервером. Можно определить формы как электронные бланки для заполнения различных данных таких как, например, имя, возраст, выбор страны проживания и других. Как правило, форма работает совместно с установленным на сервере сценарным приложением, обрабатывающим введенную информацию.

Примеры форм

Форма поиска

Одна из наиболее распространенных форм. Пользователь получает возможность ввести искомый запрос, определить область поиска и отправить запрос на сервер для обработки.

```
<html>
<body>

<form name="input" action="html_form_action.asp" method="get">
  <table border=1 bgcolor="#ddffdd">
    <tr>
      <td align="center">
        <input type="text" name="search" size="50" value="Строка для поиска">
        <input type="submit" value="Поиск">
        <br>
        <input type="checkbox" name="news">Искать в новостях
        <input type="checkbox" name="archive">Искать в архивах
      </td>
    </tr>
  </table>
</form>

</body>
</html>
```

Отправка e-mail из формы

Вторая форма представляет собой форму заказа электронной рассылки МГГТК АГУ, заполненная форма, отправляется по электронной почте для последующей обработки.

```
<html>
<body>

<form action="MAILTO:info@mgttk.ru" method="post" enctype="text/plain">
  <h3>Подписка на новостную рассылку МГГТК АГУ</h3>
  Имя:<br>
  <input type="text" name="name" value="ваше имя" size="20">
  <br>
```

```
Mail:<br>
<input type="text" name="mail" value="ваш e-mail" size="20">
<br>
Я хочу получать:
<br>
<input type="checkbox" name="news"> Новости о жизни МГГТК АГУ
<br>
<input type="checkbox" name="courses"> Информацию о новых курсах
<br>
<input type="checkbox" name="books"> Информацию о событиях
<br>
Комментарий:<br>
<textarea rows="5" cols="30">
</textarea>
<hr>
<input type="submit" value="Послать">
<input type="reset" value="Сброс">
</form>

</body>
</html>
```

Формы

Форма является областью, которая может содержать элементы, позволяющие пользователю вводить информацию (такие как текстовые поля, поля многострочного текста, раскрывающиеся меню, переключатели, флажки, и т.д.).

Форма определяется с помощью тегов `<form>`/`</form>`, между которыми располагаются поля ввода, кнопки, а также все необходимые элементы оформления формы.

Тег `<form>` имеет ряд атрибутов, из которых необходимо выделить атрибуты `action` и `method`. Без этих атрибутов форма не сможет передать информацию от пользователя на сервер.

```
<form action="html_form_action.asp" method=get>
```

Атрибут `Action` указывает URL-адрес объекта, который должен получить данные формы.

Атрибут `method` может иметь два значения: `get` и `post`.

Значение атрибута `method=get` заставляет Web-браузер передать все данные формуляра по URL-адресу, заданному в `action`. При этом введенные при заполнении формы данные просто добавляются в адресную строку с использованием разделителя – знака вопроса. Этот метод удобен для небольших форм.

Значение атрибута `method=post` заставляет Web-браузер, прежде всего, связаться с сервером, обрабатывающим форму, и только после установки связи приступить к передаче данных, для обработки которых будут использоваться специальные сценарии.

Поля ввода

Большинство элементов ввода и управления в форме можно описать при помощи тега `<input>`, обязательными для которого являются атрибуты `name` (приписывает данному элементу ввода уникальное имя, использующееся для дальнейшей обработки формы) и `type` (определяет тип элемента управления или ввода).

Текстовые поля

Текстовое поле (`type=text`) определяет однострочное поле ввода и используется, когда необходимо, чтобы пользователь ввел в форму данные в произвольной форме, но ограниченные по объему (слова, словосочетания, числа и т.д.). Следующий пример демонстрирует простейшую форму для ввода имени и фамилии.

```
<html>
<body>

<form>
  Имя:
  <input type="text" name="firstname">
  <br>
  Фамилия:
  <input type="text" name="lastname">
</form>

</body>
</html>
```

В большинстве браузеров по умолчанию ширина текстового поля равна 20 символам. Для изменения этого значения используется атрибут `size`. В следующем примере поле для ввода имени ограничено 10 символами.

```
<html>
<body>

<form>
  Имя:
  <input type="text" name="firstname" size="10">
  <br>
  Фамилия:
  <input type="text" name="lastname">
</form>

</body>
</html>
```

Здесь необходимо отметить, что атрибут `size` ограничивает только видимую область ввода данных, а не длину вводимой строки. Для этой цели используется атрибут `maxlength`. В следующем примере максимальное количество символов для ввода имени ограничено тремя.

```
<html>
<body>
```

```
<form>
Имя:
<input type="text" name="firstname" maxlength="3">
<br>
Фамилия:
<input type="text" name="lastname">
</form>

</body>
</html>
```

При вводе в поле имени слова "Елена" форма воспримет только первые три символа имени:

Кроме того, при использовании текстовых полей возможно задать значения по умолчанию. Для этого используется атрибут value. Применение этого атрибута показано на следующем примере:

```
<html>
<body>

<form>
Имя:
<input type="text" name="firstname" value="Билл">
<br>
Фамилия:
<input type="text" name="lastname" value="Гейтс">
</form>

</body>
</html>
```

Необходимо отметить, что применение атрибутов для различных типов полей вывода и управления может отличаться. Так, например, атрибут size для текстовых полей ввода (type="text" или type="password") указывает максимальное количество символов, отображаемых в поле, а для прочих элементов – занимаемый по горизонтали размер в пикселях.

Поле пароля

Поле пароля (type=password) создает защищенное поле ввода, которое дает возможность пользователю, заполняющему форму ввести текст, но в отличие от обычного текстового поля, вводимые данные при отображении на мониторе заменяются звездочками или точками.

Следующий пример демонстрирует простейшую форму для ввода имени пользователя и пароля.

```
<html>
<body>

<form>
```

```
Имя пользователя:
<input type="text" name="user" value="Елена">
<br>
Пароль:
<input type="password" name="password" value="tktyf">
</form>

</body>
</html>
```

Необходимо обратить внимание, что хотя значение пароля и задано по умолчанию, при отображении браузер выводит вместо символов звездочки.

Переключатели

Переключатели или радиокнопки (type=radio) определяют поля выбора одного значения из нескольких доступных. Поля этого типа часто используются в диалоговых окнах. Для каждой позиции переключателя создается свой тег <input type=radio>. Группируются переключатели при помощи одинакового имени, задаваемого атрибутом name.

```
<html>
<body>

<form>
  Укажите Ваш пол:
  <br>
  <input type="radio" name="sex" value="male"> мужчина
  <br>
  <input type="radio" name="sex" value="female"> женщина
</form>

</body>
</html>
```

Необходимо отметить, что в отличие от текстового поля и поля пароля атрибут value задает значение, которое будет передано серверу для дальнейшей обработки в случае выбора данного переключателя. Для выбора по умолчанию одного из возможных значений группы переключателей используется атрибут checked.

```
<html>
<body>

<form>
  Укажите Ваш пол:
  <br>
  <input type="radio" name="sex" value="male" checked="checked"> мужчина
  <br>
  <input type="radio" name="sex" value="female"> женщина
</form>

</body>
</html>
```

В данном примере по умолчанию выбран мужской пол:

Флажки

Флажки (type=checkbox) используются, когда необходимо, чтобы пользователь выбрал один или несколько вариантов из ограниченного числа вариантов выбора. Флажки в форме не зависят друг от друга, их можно установить или сбросить в любой комбинации. Для каждого флажка необходимо задать свое уникальное имя при помощи атрибута name. Создание двух флажков с одним именем не вызовет ошибки при отображении формы, но не позволит сценарию обработки на сервере корректно обработать передаваемые с формы данные.

```
<html>
<body>

<form>
  В этом году я собираюсь приобрести:
  <br>
  <input type="checkbox" name="computer">
  Компьютер
  <br>
  <input type="checkbox" name="notebook">
  Ноутбук
  <br>
  <input type="checkbox" name="printer">
  Принтер
  <br>
  <input type="checkbox" name="scanner">
  Сканер
</form>

</body>
</html>
```

При помощи атрибута checked можно установить, какие из флажков будут выбраны по умолчанию при загрузке страницы. Отличие от переключателей заключается только в том, что для флажков можно отметить сразу несколько вариантов.

```
<html>
<body>

<form>
  В этом году я собираюсь приобрести:
  <br>
  <input type="checkbox" name="computer" checked="checked">
  Компьютер
  <br>
  <input type="checkbox" name="notebook">
  Ноутбук
  <br>
  <input type="checkbox" name="printer">
```

```
Принтер
<br>
<input type="checkbox" name="scanner" checked="checked">
Сканер
</form>

</body>
</html>
```

В данном примере по умолчанию выбраны флажки "Компьютер" и "Сканер":

При отправке данных формы с флажками на сервер выбранным флажкам присваивается значение по умолчанию "on". Как правило, этого достаточно для корректной обработки данных, но в некоторых случаях удобнее задать каждому флажку свое значение при помощи атрибута value.

```
<input type="checkbox" name="printer" value="Принтер">
```

Командные кнопки

Командная кнопка отправки (type=submit) используется для выполнения пересылки данных формы на сервер. Метод отправки и адрес файла, обрабатывающего полученную информацию, задаются в теге <form> при помощи атрибутов method и action. Командная кнопка сброса (type=reset) возвращает форму к исходному состоянию (очищает форму). При этом данные не передаются.

В следующем примере показана форма поиска с двумя кнопками отправки и сброса.

```
<html>
<body>

  <form name="input" action="html_form_action.asp" method="get">
    Найти:
    <input type="text" name="search" size=25>
    <br>
    <input type="submit">
    <input type="reset">

  </form>

</body>
</html>
```

Если ввести в текстовое поле какие-то символы и нажать кнопку "Подача запроса", то введенная информация будет послана на страницу с именем "html_form_action.asp". При нажатии на кнопку "Сброс" текстовое поле очистится.

Надписи на кнопках "Подача запроса" и "Сброс" установлены по умолчанию. Для их изменения необходимо использовать атрибут value.

Кроме кнопок отправки и сброса существует также возможность добавлять пользовательские кнопки (type=button), которые могут использоваться для выполнения процедур (скриптов) непосредственно на Web-странице.

Поле выбора файла

Поле выбора файла (type=file) создает поле для выбора файла, который будет загружен на сервер вместе с информацией формы. Рядом с полем ввода отображается командная кнопка "Обзор...", открывающая стандартное диалоговое окно выбора файла.

```
<html>
<body>

<form>
  Прикрепить файл:
  <br>
  <input type="file" size="50">
</form>

</body>
</html>
```

Для поля выбора файла по аналогии с текстовым полем можно использовать атрибуты size, maxlength, value.

Списки выбора

Списки выбора бывают двух типов: раскрывающиеся списки (выпадающие меню) и списки с множественным выбором. Независимо от типов списков описываются они одинаково с помощью пары тегов <select> </select>. Отдельные элементы списка задаются с использованием тега <option>. Тип списка определяется при помощи атрибутов тега <select>.

Атрибут name задает имя поля для отправки выбранных пунктов списка на сервер. Атрибут multiple разрешает множественный выбор. Атрибут size определяет, какое количество пунктов списка будет одновременно отображено на экране. При этом, если атрибут multiple не задан и size=1, то на экране отображается раскрывающийся список, если же задан атрибут Multiple или значение size больше 1, то список отображается развернутым.

Раскрывающийся список выбора

Раскрывающиеся списки выбора по своему назначению соответствуют переключателям, в то же время их использование предпочтительно в тех случаях, когда количество вариантов выбора достаточно велико. Как правило, при выборе более чем из трех вариантов желательно вместо переключателей использовать раскрывающиеся списки.

В следующем примере создан простой раскрывающийся список выбора ноутбука по производителю.

```
<html>
<body>
```



```

<form>
  Выбор ноутбука по производителю:
  <select name="notebook">
    <option value="acer">Acer
    <option value="asus">Asus
    <option value="compaq">Compaq
    <option value="hp">HP
    <option value="sony">Sony
    <option value="toshiba">Toshiba
  </select>
</form>

</body>
</html>

```

Необходимо заметить, что по умолчанию выбирается первое значение из списка. При помощи атрибута `selected` тега `<option>` это значение можно изменить. Следующий пример показывает раскрывающийся список выбора размера экрана ноутбука с предварительно установленным значением "15.4".

```

<html>
<body>

<form>
  Выбор размера экрана ноутбука
  <select name="tft">
    <option value="tft-12">12"
    <option value="tft-13">13"
    <option value="tft-14">14"
    <option value="tft-15">15"
    <option value="tft-15-4" selected="selected">15.4"
    <option value="tft-17">17"
  </select>
</form>

</body>
</html>

```

Развернутый список выбора

Развернутые списки используются, как правило, при необходимости множественного выбора значений. В следующем примере раскрывающийся список выбора ноутбука по производителю преобразован в развернутый список с возможностью множественного выбора.

```

<html>
<body>

<form>
  Выбор ноутбука по производителю:
  <select name="notebook" multiple>

```

```

    <option value="aser">Aser
    <option value="asus">Asus
    <option value="compaq">Compaq
    <option value="hp">HP
    <option value="sony">Sony
    <option value="toshiba">Toshiba
  </select>
</form>

</body>
</html>

```

Необходимо отметить, что для списков с множественным выбором значение атрибута size по умолчанию устанавливается равным 4, кроме того, ни одно из значений не является выбранным по умолчанию. В следующем примере показана форма с развернутым списком высотой в 6 пунктов с предварительно заданными значениями "Compaq" и "Sony".

```

<html>
<body>

<form>
  Выбор ноутбука по производителю:
  <select name="notebook" multiple size=6>
    <option value="aser">Aser
    <option value="asus">Asus
    <option value="compaq" selected="selected">Compaq
    <option value="hp">HP
    <option value="sony" selected="selected">Sony
    <option value="toshiba">Toshiba
  </select>
</form>

</body>
</html>

```

Текстовая область

В отличие от текстового поля `<input type="text">` текстовая область позволяет вводить многострочный текст большого объема. Такие области часто используются при вводе сообщений, комментариев.

Текстовое поле описывается при помощи тегов `<textarea>` `</textarea>`, между которыми можно разместить предварительно отформатированный стандартный текст. Атрибуты `cols` и `rows` задают размер видимой области текстового поля.

В следующем примере создано текстовое поле с предварительно введенным текстом.

```

<html>
<body>

  <textarea rows="7" cols="30">

```

В данном примере мы создали текстовую область с шириной в 30 символов и высотой в 7 строк.

Заданное значение высоты не ограничивает общий объем вводимого текста, а влияет только на размер отображаемой на экране текстовой области.

Для просмотра всего текста необходимо воспользоваться полосой прокрутки.

```
</textarea>
```

```
</body>
```

```
</html>
```

Внешний вид форм

Существует различные способы оформления внешнего вида форм. Один из них основан на использовании пары тегов `<fieldset>` `</fieldset>`, которые позволяют объединить тематически близкие и рядом расположенные поля формы в группу и выделить ее визуально. Другой способ основан на применении стандартных средств форматирования HTML, включая цветное и шрифтовое оформление, графику, таблицы и т.п.

Группировка полей формы

Теги `<fieldset>` `</fieldset>` объединяют поля формы в группы, выделяют их визуально, упрощают навигацию в форме с помощью клавиши [Tab] (в первую очередь обходятся поля в пределах группы). Для того чтобы объединить несколько элементов ввода или управления в группу достаточно поместить их внутри тегов `<fieldset>` `</fieldset>`. Закрывающий тег `</fieldset>` обязателен.

В следующем примере создается форма, содержащая поля для ввода имени и фамилии, объединенные в группу, а также кнопку отправки

```
<html>
```

```
<body>
```

```
<form name="input" action="html_form_action.asp" method="get">
```

```
<fieldset>
```

```
Имя:
```

```
<input type="text" name="firstname" value="Билл">
```

```
<br>
```

```
Фамилия:
```

```
<input type="text" name="lastname" value="Гейтс">
```

```
</fieldset>
```

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

```
</form>
```

```
</body>
```

```
</html>
```

При помощи пары тегов `<legend>` `</legend>` образованной группе полей можно присвоить имя. В следующем примере форма ввода дополнена полями для ввода информации о месте работы и занимаемой должности, объединенными во вторую группу. Каждой группе полей присвоен свой заголовок.

```
<html>
```

```

<body>

<form name="input" action="html_form_action.asp" method="get">
  <fieldset>
    <legend>Личные данные</legend>
    Имя:
    <input type="text" name="firstname" value="Билл">
    <br>
    Фамилия:
    <input type="text" name="lastname" value="Гейтс">
    <br>
    Пол:
    <input type="radio" name="Sex" value="Male" checked="checked"> Мужской
    <input type="radio" name="Sex" value="Female"> Женский
  </fieldset>
  <fieldset>
    <legend>Данные о работе</legend>
    Место работы:
    <input type="text" name="work" value="Microsoft">
    <br>
    Должность:
    <input type="text" name="status" value="Президент">
  </fieldset>

  <input type="submit" value="отправить">
</form>

</body>
</html>

```

Оформление форм стандартными средствами HTML

Несмотря на простоту использования тегов группировки <fieldset>, их возможности по оформлению форм весьма ограничены. Чаще всего для оформления внешнего вида форм используются стандартные средства HTML. В следующем примере показано, как с помощью таблицы поля формы выровнены друг под другом, а заголовок формы и кнопка отправки размещены по центру формы. Сама форма визуально выделена желтым цветом.

```

<html>
<body>

<form name="input" action="html_form_action.asp" method="get">
  <table bgcolor="#FFFF99">
    <tr>
      <th colspan="2" align="center">Личные данные:</th>
    </tr>
    <tr>
      <td align="right">Имя:</td>
      <td align="left"><input type="text" name="firstname" value="Билл"></td>
    </tr>
    <tr>
      <td align="right">Фамилия:</td>

```

```

        <td align="left"><input type="text" name="lastname" value="Гейтс"></td>
    </tr>
    <tr>
        <td colspan="2" align="center"><input type="submit" value="отправить"></td>
    </tr>
</table>
</form>

</body>
</html>

```

Примеры форм

Регистрация в системе

```

<html>
<body>

<form name="input" action="html_form_action.asp" method="post">
    <table bgcolor="#9999FF">
        <tr>
            <th colspan="2" align="center">Регистрация в системе</th>
        </tr>
        <tr>
            <td align="right">Имя:</td>
            <td align="left"><input type="text" name="name"></td>
        </tr>
        <tr>
            <td align="right">Пароль:</td>
            <td align="left"><input type="password" name="password"></td>
        </tr>
        <tr>
            <td colspan="2" align="center"><input type="checkbox">Запомнить пароль</td>
        </tr>
        <tr>
            <td colspan="2" align="center"><input type="submit" value="Вход"></td>
        </tr>
    </table>
</form>

</body>
</html>

```

В этом примере представлена типовая форма входа в систему авторизованного пользователя. Форма состоит из поля ввода имени, поля ввода пароля, флажка сохранения пароля и кнопки входа.

Необходимо отметить, что при работе с паролями следует использовать только метод post. При использовании метода get пароль будет передаваться для обработки в открытом виде. Для проверки этого достаточно в данном примере заменить метод post на get, запустить пример, заполнить поля (например, имя пользователя: Bill, пароль

superstar) и нажать кнопку "Вход". В этом случае в адресной строке браузера появится адрес: {ваш рабочий каталог}/html_form_action.asp?name=Bill&password=superstar.

Примечание: необходимо, чтобы в Вашем рабочем каталоге находился файл с именем "html_form_action.asp" (это может быть даже пустой файл).

Запрос документа

```
<html>
<body>
```

```
<form action="MAILTO:info@mgtk.ru" method="post" enctype="text/plain">
```

```
<form name="input" action="html_form_action.asp" method="get">
```

```
<table bgcolor="#dddddd">
  <tr>
    <th colspan="2" align="center">Запрос документа</th>
  </tr>
  <tr>
    <td align="right">Тип документа:</td>
    <td align="left">
      <select name="do_type">
        <option value="mail">Письмо
        <option value="order">Приказ
        <option value="message">Служебная записка
      </select>
    </td>
  </tr>
  <tr>
    <td align="right">Дата регистрации с</td>
    <td align="left"><input type="text" name="date_begin" size="21"
value="дд.мм.гггг"></td>
  </tr>
  <tr>
    <td align="right">по</td>
    <td align="left"><input type="text" name="date_end" size="21" value="дд.мм.гггг"></td>
  </tr>
  <tr>
    <td align="right">Регистрационный номер:</td>
    <td align="left"><input type="text" name="number" size="21"></td>
  </tr>
  <tr>
    <td colspan="2"><hr></td>
  </tr>
  <tr>
    <td align="right" valign="top">Ключевые слова:</td>
    <td align="left">
      <textarea cols="17" rows="5">
      </textarea>
    </td>
  </tr>
  <tr>
    <td colspan="2"><hr></td>
```

```

</tr>
<tr>
  <td colspan="2" align="center">
    <input type="submit" value="Отправить">
    <input type="reset">
  </td>
</tr>
</table>
</form>

</body>
</html>

```

Особенностью данной формы является то, что введенные данные отправляются на указанный адрес электронной почты. В остальном, это достаточно типовая форма, содержащая поля ввода текста, текстовую область и выпадающий список. Для пояснения формата ввода в полях даты регистрации по умолчанию показан образец заполнения. Но при этом необходимо отметить, что форма не проверяет корректность введенной информации.

Регистрация нового пользователя

```

</body>
</html>

<html>
<body>

<form name="input" action="html_form_action.asp" method="post">
  <table bgcolor="#dddddd">
    <tr>
      <th colspan="2" align="center">Регистрация нового пользователя</th>
    </tr>
    <tr>
      <td colspan="2" align="center">Обязательные поля выделены красным
        цветом<br><br></td>
    </tr>
    <tr>
      <td align="right"><font color="red">Ник (псевдоним):</td>
      <td align="left"><input type="text" name="nik" size="27"></td>
    </tr>
    <tr>
      <td align="right"><font color="red">E-mail</td>
      <td align="left"><input type="text" name="e_mail" size="27"></td>
    </tr>
    <tr>
      <td align="right"><font color="red">Пароль (не менее 6 сим.):</td>
      <td align="left"><input type="password" name="password" size="27"></td>
    </tr>
    <tr>
      <td align="right"><font color="red">Подтвердите пароль:</td>
      <td align="left"><input type="password" name="f_password" size="27"></td>
    </tr>
  </table>
</form>

```

```
<tr>
  <td colspan="2"><hr></td>
</tr>
<tr>
  <td align="right">Фамилия</td>
  <td align="left"><input type="text" name="name_f" size="27"></td>
</tr>
<tr>
  <td align="right">Имя</td>
  <td align="left"><input type="text" name="name_i" size="27"></td>
</tr>
<tr>
  <td align="right">Отчество</td>
  <td align="left"><input type="text" name="name_o" size="27"></td>
</tr>
<tr>
  <td align="right">Дата рождения:</td>
  <td align="left">
    <select name="day">
      <option>1
      <option>2
      <option>3
      <option>4
      <option>5
      <option>6
      <option>7
      <option>8
      <option>9
      <option>10
      <option>11
      <option>12
      <option>13
      <option>14
      <option>15
      <option>16
      <option>17
      <option>18
      <option>19
      <option>20
      <option>21
      <option>22
      <option>23
      <option>24
      <option>25
      <option>26
      <option>27
      <option>28
      <option>29
      <option>30
      <option>31
    </select>
    <select name="month">
```



```

        <option>января
        <option>февраля
        <option>марта
        <option>апреля
        <option>мая
        <option>июня
        <option>июля
        <option>августа
        <option>сентября
        <option>октября
        <option>ноября
        <option>декабря
    </select>
    <select name="year">
        <option>1980
        <option>1981
        <option>1982
        <option>1983
        <option>1984
        <option>1985
        <option>1986
        <option>1987
        <option>1988
        <option>1989
        <option>1990
    </select>
</td>
</tr>
<tr>
    <td align="right">Пол:</td>
    <td align="left">
        <input type="radio" name="sex" value="male">Мужской
        <input type="radio" name="sex" value="female">Женский
    </td>
</tr>
<tr>
    <td align="right">Город проживания:</td>
    <td align="left"><input type="text" name="state" size="27"></td>
</tr>
<tr>
    <td align="right" valign="top">Образование:</td>
    <td align="left">
        <input type="radio" name="edu" value="h">Высшее<br>
        <input type="radio" name="edu" value="hh">Незаконченное высшее<br>
        <input type="radio" name="edu" value="ss">Среднее специальное<br>
        <input type="radio" name="edu" value="s">Среднее
    </td>
</tr>
<tr>
    <td colspan="2"><hr></td>
</tr>
<tr>

```

```

        <td colspan="2" align="center">
            <input type="submit" value="Зарегистрироваться">
            <input type="reset" value="Очистить форму">
        </td>
    </tr>
</table>
</form>

</body>
</html>

```

В данном примере представлена форма регистрации нового пользователя. Отличительной особенностью данной формы является организация ввода даты рождения, построенная с использованием трех выпадающих списков. Хотя реализация такого ввода дат на первый взгляд более трудоемка в процессе создания Web-страницы (для сравнения, в предыдущем примере для организации поля ввода даты потребовалась всего одна строка), но она оправдывает себя за счет сведения к минимуму ошибок при вводе информации. Такая схема организации ввода широко используется при вводе и других данных, таких как, например, страна и город проживания (в данном примере не реализовано).

Следует обратить внимание и на присутствие в форме двух групп переключателей, которые отвечают за ввод пола и образования пользователя.

Теги форм	
Тег	Описание
<form>	Определяет форму для ввода пользователя
<input>	Определяет поле ввода
<textarea>	Определяет текстовую область (элемент управления для ввода многострочного текста)
<label>	Определяет метку для элемента управления
<fieldset>	Определяет набор полей
<legend>	Определяет заглавие для набора полей
<select>	Определяет список выбора (раскрывающееся поле)
<optgroup>	Определяет группу вариантов выбора
<option>	Определяет вариант в раскрывающемся поле
<button>	Определяет кнопку
<isindex>	Не рекомендуется. Используйте вместо этого <input>