**WEB-формы**

**Веб формы** – это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку, где информация будет некоторым образом обработана. С помощью веб форм на сервер отправляются тексты, изображения и другие файлы.

В ранее выполненных уроках Вы освоили технологии, позволяющие предоставлять информацию посетителю веб-сайта. Но при разработке любой программы одна из главных задач, стоящих перед программистом, – это создание механизма взаимодействия с пользователем. Вам нужно научиться получать информацию от посетителя веб-сайта, а затем обрабатывать эту информацию. В HTML для получения данных посетителя предусмотрены веб формы и их элементы. В этом уроке Вы освоите технологию создания веб форм для сбора информации от посетителей.

После того, как посетитель заполняет предложенную ему форму (например, комментируя Ваш сайт или оформляя заказ в Вашем Internet-магазине), данные, введенные им в поля формы, отправляются серверной программе для последующей обработки.

**1. Создание веб формы**

Веб форма организуется с помощью тега **<form>** и **</form>**. HTML-документ может содержать в себе несколько форм, однако **формы не следует вкладывать друг в друга.**

Атрибуты тега **<form>:**

* **action** – обязательный атрибут, указывается URL, по которому следует передать введенную в форму информацию;
* **method** – атрибут, указывающий каким образом (иначе говоря, с помощью какого метода протокола передачи гипертекстов) данные из формы будут переданы обработчику.

**Задание 1**

Откройте справочник **Sprav\_HTML.doc** и ознакомьтесь с атрибутами тега **<form>** и правилами их назначения.

**2. Описание элементов веб формы**

Существует три типа элементов формы: **<textarea>, <select>, <input>.**

**2.1. Элемент формы <textarea>...</textarea>**

Этот элемент служит для создания многострочного поля ввода (рис. 1).

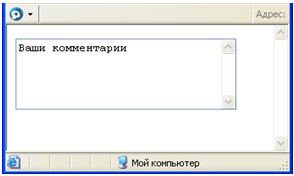


Рис. 1. Пример использования элемента <textarea>

У элемента **<textarea>** имеются несколько атрибутов:

* **name** – обязательный атрибут, определяющий имя, под которым содержимое поля будет передано обработчику;
* **rows** – атрибут, устанавливающий высоту окна в строках;
* **cols** – атрибут, устанавливающий ширину окна в символах.

Например:

<textarea name="address" rows="5" cols="30">.

Внешний вид формы, содержащей элемент <textarea>, приведен на рис. 1.

Многострочные поля ввода можно использовать, например, для ввода комментариев посетителя.

**Задание 2**

1. Реализуйте форму, изображенную на рис. 1.
2. Сохраните файл в папке **Lab\_5** под именем **forms.htm**

**2.2. Элемент формы <select>   </select>**

Этот элемент служит для организации разворачивающихся меню и полей списка (рис. 2).

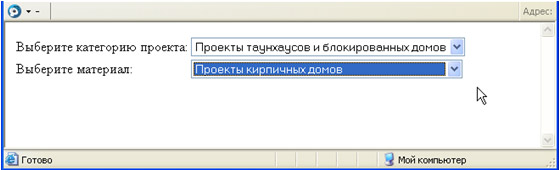


Рис. 2. Пример раскрывающегося списка

 У элемента **<select>** имеются следующие атрибуты:

* **name** – обязательный, определяющий имя переменной, которую генерирует меню;
* **multiple** –  устанавливает режим выбора нескольких значений из списка;
* **size** – устанавливает число строк, отображаемых одновременно.

Между тегами **<select>** и **</select>** находятся значения, которые может выбирать пользователь. Они перечислены в теге **<option>.** Тег **<option>** – одиночный элемент, не требующий закрывающего тега.

Тег **<option>**определяет элемент меню. У тега **<option>** имеются следующие атрибуты:

* **value** – обязательный, устанавливающий значение, которое будет передано обработчику, если выбран этот элемент меню;
* **selected** – показывает, что данный элемент отображается выбранным.

HTML-код списка выглядит так (рисунок 3):

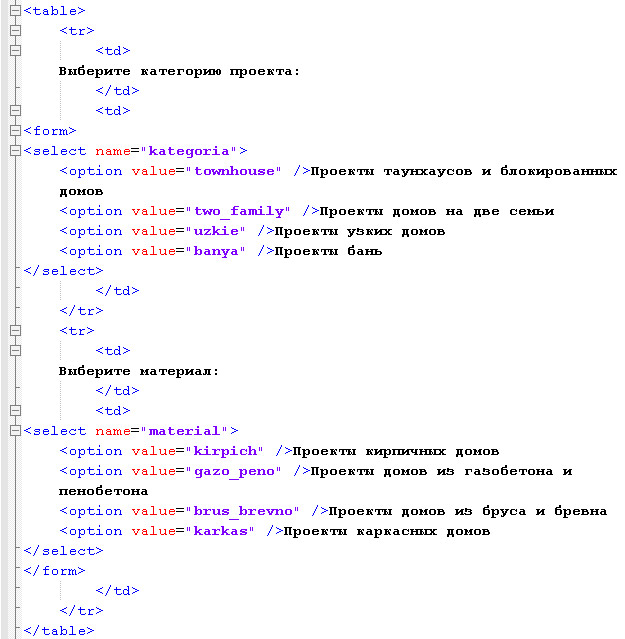


Рис. 3

 Внешний вид этой формы показан на рис. 2.

**Задание 3**

В файле **forms.html** реализуйте код, предложенный на рисунке 3. Выясните, как влияют на вид веб формы атрибуты **size** и **multiple** тега **<select>** и атрибут **selected** тега **<option>.**

Сохраните файл.

**2.3. Элемент формы <input>**

Этот элемент предоставляет возможность использовать другие типы ввода информации: отдельные строки текста, переключатели, флажки и кнопки передачи запроса и отказа от ввода. Это одиночный элемент, не требующий закрывающего тега.

У тега **<input>** имеются следующие атрибуты: **name, size, maxlength, value, checked, type.**

**Задание 4**

Откройте справочник **Srav\_html.doc** и ознакомьтесь с атрибутами тега **<input>** и правилами их назначения.

**Типы элементов <input>**

Основные типы элементов **<input>,** определяемые атрибутом type:

* **type="radio"** – определяет переключатель.

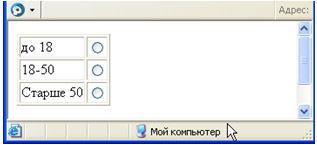


Рис. 4. Пример использования элементов radio

Используется в тех случаях, когда необходимо выбрать только один из родственных элементов. Это значение атрибута **type** используется с атрибутами **name** (обязателен), **value** (определяет значение, возвращаемое серверу в случае выбора данной кнопки), **checked** (показывает, что кнопка выбрана по умолчанию). Присвоив одинаковые значения атрибуту **name** разных переключателей,  их можно объединить в группу. В группе радиокнопок с одинаковыми именами может быть только одна помеченная радиокнопка. Пример:

<input type="radio" name="modem" value="9600" checked /> 9600 бит/с  
<input type="radio" name="modem" value="14400" /> 14400 бит/с  
<input type="radio" name="modem" value="28800" /> 28800 >бит/с

В данном примере определена группа из трех радиокнопок, подписанных, соответственно: 9600 бит/с, 14400 бит/с и 28800 бит/с. Первоначально помечена первая из кнопок. Если пользователь не отметит другую кнопку, обработчику будет передана переменная modem со значением 9600. Если пользователь отметит другую кнопку, обработчику будет передана переменная modem со значением 14400 или 28800.

**Задание 5**

В файл **forms.html** добавьте веб форму, запрашивающую сведения о возрасте посетителя. Форма должна выглядеть как на рис. 4.

Примечание: при создании формы наиболее распространенная ошибка, доставляющая неудобство пользователю – неровно расположенные поля. Поэтому для размещения полей формы можно использовать таблицу.

* **type="checkbox"** – кнопка с независимой фиксацией. Используется в тех случаях, когда одновременно можно выбрать несколько опций. Атрибуты: **name (обязателен), value, checked**(рис. 5)**.**

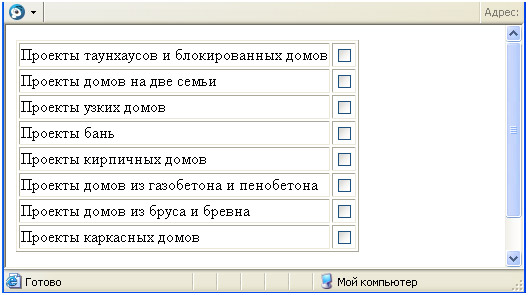


Рис. 5. Пример использования элементов checkbox

Если на Ваш сайт зайдет пользователь, желающий купить проект, то, прежде чем оформить его заказ, следует уточнить, какие проекты его интересуют. Причем, возможно, он еще не решил, какой проект ему нужен, и готов рассмотреть несколько вариантов.

Чтобы решить эту задачу, нужно создать форму, использующую элементы **checkbox**, так как в этом случае, в отличие от использования элементов radio, можно выбрать несколько вариантов.

**Задание 6**

В файле **forms.html** создайте веб форму как на рис. 5.

* **type="submit"** – командная кнопка, которая отправляет на сервер все введенные данные. Данные отправляются по адресу, указанному в атрибуте action тега <form>. Надпись на кнопке можно задать путем введения атрибута value="Надпись", например:

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

* **type="reset"** – определяет кнопку, при нажатии на которую форма возвращается в исходное состояние. В этом случае данные обработчику не передаются, поэтому кнопка может и не иметь атрибута name. Надпись на кнопке задается при помощи атрибута value="Надпись".

**Задание 7**

1. В файле **forms.html** добавьте в конец каждой формы две кнопки: submit и reset с соответствующими надписями. Проверьте действие кнопки reset.
2. По нажатию кнопки **submit** данные, введенные пользователем, отправляются соответствующей программе для обработки. Имя этой программы и путь к ней указываются в атрибуте **action** тега **<form>.** Вы пока не умеете писать такие программы но, чтобы проверить работу кнопки submit, Вы можете в качестве значения атрибута **action** указать адрес любой Web-страницы или картинки.
3. Сохраните созданную страницу. Результат для одной из веб форм на рис. 6

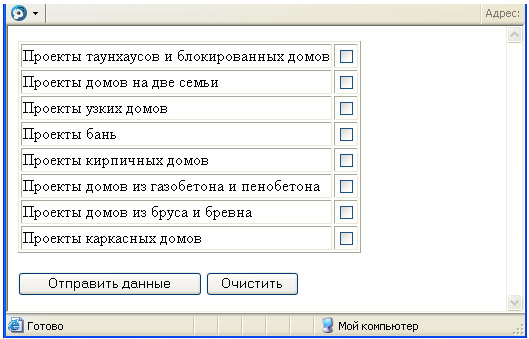


Рис. 6. Пример использования элементов submit и reset

* **type="text"** – определяет окно для ввода строки текста. Это значение атрибута type можно использовать с атрибутами **name** (обязателен), **value**, **size** и **maxlength**.

Если на Ваш сайт зайдет пользователь, желающий купить проект, то, прежде чем оформить его заказ, следует уточнить, каким образом Вы с ним будете связываться.

Чтобы решить эту задачу, нужно создать форму, позволяющую клиенту вводить контактную информацию о себе. При ее создании используются элементы **text**.

**Задание 8**

В файле **forms.html**создайте веб форму, которая выглядит как на рис. 7. Поля, обязательные для заполнения, отметьте звездочками.

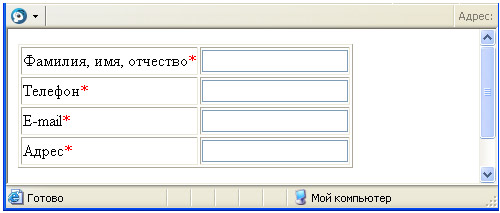


Рис. 7. Пример использования элементов text

* **type="password"** – определяет окно для ввода пароля. Абсолютно аналогичен типу text, только вместо символов вводимого текста показывает на экране звездочки (**\***).
* **type="button"** – не выполняет никаких действий при нажатии на кнопку, если к ней не прикреплена программа. Это значение атрибута type используется с атрибутами **name** (имя кнопки), **value** (надпись на кнопке), **onclick** (для указания имени программы, которая будет выполняться при нажатии на эту кнопку).

**Создание ссылки для отправки письма на электронный адрес**

Вы можете создать ссылку на сайте таким образом, чтобы пользователь мог отправить письмо на Ваш электронный адрес. Например, при нажатии по ссылке «**Написать письмо**» (рис. 8), загрузится почтовая программа, и в строке «**Кому**» уже будет прописан адрес Вашей электронной почты, а в строке «**Oт**» будет электронный адрес пользователя, к которому подключена почтовая программа (рис. 9).

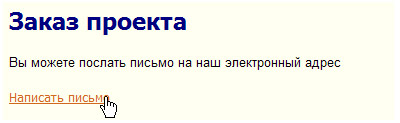


Рис. 8. Ссылка для отправки письма на электронный адрес

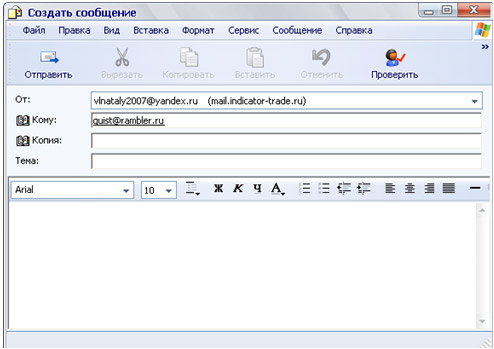


Рис. 9. Загрузка почтовой программы

Структура ссылки для отправки сообщений на электронный адрес выглядит следующим образом

<a href="mailto:Ваш электронный адрес">Написать письмо (или другой текст)</a>

**Пример:**

<a href="mailto:quist@rambler.ru">Написать создателям сайта</a>

**Контрольное задание**

1. Разработать для сайта проектов страницу **zakaz.htm** с формой, через которую будет осуществляться связь с посетителями сайта через веб форму (рис. 10). В раскрывающемся списке «Выберите категорию проекта» должны быть следующие пункты:

* Проекты таунхаусов и блокированных домов,
* Проекты домов на две семьи,
* Проекты узких домов,
* Проекты бань.

В раскрывающемся списке «Выберите материал» должны быть следующие пункты:

* Проекты кирпичных домов,
* Проекты домов из газобетона и пенобетона,
* Проекты домов из бруса и бревна,
* Проекты каркасных домов.

2. Создайте ссылку **«Написать письмо»** для отправки сообщений с сайта на Ваш электронный адрес (рис. 10).

3. При нажатии на кнопку **Очистить** данные, введенные в форму, должны исчезать.

4. Включите созданную страницу с веб формой в структуру сайта. Для этого в левом меню создайте пункт **«Заказ проекта»** и сделайте его гиперссылкой на страницу с формой заказа (рис. 11).

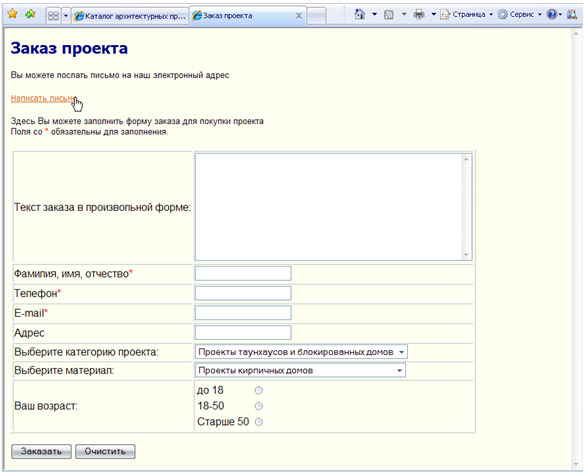


Рис. 10. Web-страница с формой заказа проекта

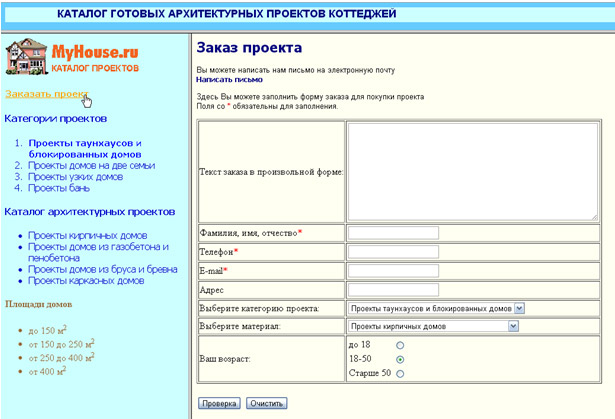


Рис. 11. Сайт с веб формой заказа проекта

**Исследовательское задание**

1. Реализуйте html-код и css-код, как на рисунках 12 и 13.

2. Результат показан на рисунке 14.

3. Используя справочник **http://htmlbook.ru/** ответьте на вопросы:

* для чего предназначены теги  **fieldset**, **legend**, **label**?
* как отобразить на веб-странице календарь, месяц, время, выбор цвета, загрузку файла?
* что такое **placeholder**?
* какую функцию выполняет параметр **required**?



Рис. 12. Html-код

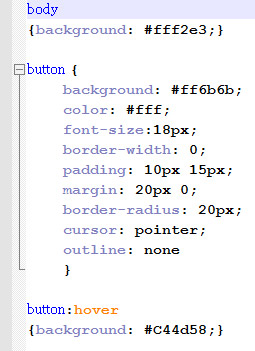


Рис. 13. CSS-код



Рис. 14. Результат