**Работаем с html-формами.**

**HTML-формы** являются элементами управления, которые применяются для сбора информации от посетителей веб-сайта.

Веб-формы состоят из набора текстовых полей, кнопок, списков и других элементов управления, которые активизируются щелчком мыши. Технически формы передают данные от пользователя удаленному серверу.

Для получения и обработки данных форм используются языки веб-программирования, такие как **PHP**.

**Давайте рассмотрим все элементы формы:**

Любые элементы формы находятся как бы в **теле формы**. Т.е. у любой формы есть каркас, а уже внутри него вставляются различные элементы формы, и пишется html код.

**Пример 1:**

**<form>**  
**<!-- сюда вставляют различные элементы -->**  
**</form>**

Как видите здесь два базовых тега: открывающий **<form>** и обязательный закрывающий тег **</form>**.

Но в этом выше, как бы голый каркас, т.е. без атрибутов. А теперь рассмотрим более реальный каркас *html формы*:

**Пример 2:**

**<form name="forma zakaza" method="post" action="obrabotchik.php">**  
**<!-- сюда вставляют различные элементы -->**  
**</form>**

**Атрибуты:**

**NAME** – определяет имя формы, уникальное для данного документа.

**ACTION** – **обязательный атрибут**, который указывает url обработчика формы на сервере, которому передаются данные. Представляет из себя файл (например, obrabotchik.php), в котором описано, что нужно делать с данными формы.  Если атрибут action отсутствует, текущая страница перезагружается, возвращая все элементы формы к их значениям по умолчанию.

**METHOD** – определяет способ отправки содержимого html формы. Возможные значения GET (по умолчанию) и POST.

Пару слов о методах передачи:

Метод GET передает данные на сервер через адресную строку браузера. Метод GET используется для передачи различных переменных, или очень коротких сообщений. Информация передается в явном виде через строку браузера, т.е. ее можно перехватить. Например, если вы видите в строке набора браузера нечто вроде **http://adress.com/lessons.php?rub=28** это значит, что передается значение переменной rub равное 28. (Имена и значения переменных присоединяются к адресу сервера после знака ? и разделяются между собой знаком &.)   
Метод POST применяется для пересылки данных больших объемов, а также конфиденциальной информации и паролей. Данные, отправляемые с помощью этого метода, не видны в заголовке URL, так как они содержатся в теле сообщения.

Элемент **<fieldset>...</fieldset>**предназначен для группировки элементов, связанных друг с другом, разделяя таким образом форму на логические фрагменты.

Парный тег **<legend>,** находящийся внутри тега <fieldset>, позволяет задать заголовок для этого блока.

Элемент**input** создает большинство полей формы. Атрибуты элемента отличаются в зависимости от типа поля, для создания которого используется этот элемент.

**Основные атрибуты:**

**TYPE** - определяет тип поля для ввода данных. По умолчанию – это "text".

**NAME -** определяет имя, используемое при передаче содержания данной html формы на сервер. Если Вы собираетесь куда-нибудь передавать информацию из формы, то обязательно вводите имя.

**SIZE -** определяет размер поля в символах. По умолчанию имеет значение равное 20.

Есть еще и такой полезный атрибут как **MAXLENGTH** , который определяет максимальное количество символов, которые можно ввести в текстовом поле. Оно может быть больше, чем количество символов, указанных в атрибуте SIZE. По умолчанию количество символов не ограничено.

## **Текстовые поля**

<input type="text"> - создает текстовые поля в форме, выводя однострочное текстовое поле для ввода текста.

<input type="email"> - браузеры, поддерживающие данный атрибут, будут ожидать, что пользователь введет данные, соответствующие синтаксису адресов электронной почты.

<input type="password">- cимволы показываются как точки

<input type="number"> - предназначено для ввода целочисленных значений. Его атрибуты min, max и step задают верхний, нижний пределы и шаг между значениями соответственно.

**Пример:**

<form name="primer2" method="post" action="obrabotchik.php">Введите пароль(максимум шесть символов):<br>  
**<input type="password" name ="pass" maxlength="6">**  
**<**/form>

**Результат:**

Введите пароль (максимум шесть символов):  
  


Еще есть атрибут **VALUE** который определяет, что будет по умолчанию написано в поле для ввода.

**Пример:**

<form name="primer3" method="post" action="obrabotchik.php">Введите свой e-mail: <br> **<input type="text" name="e-mail" size="35" value="пример: admin@zvirec.com">**  
**<**/form>

Результат:

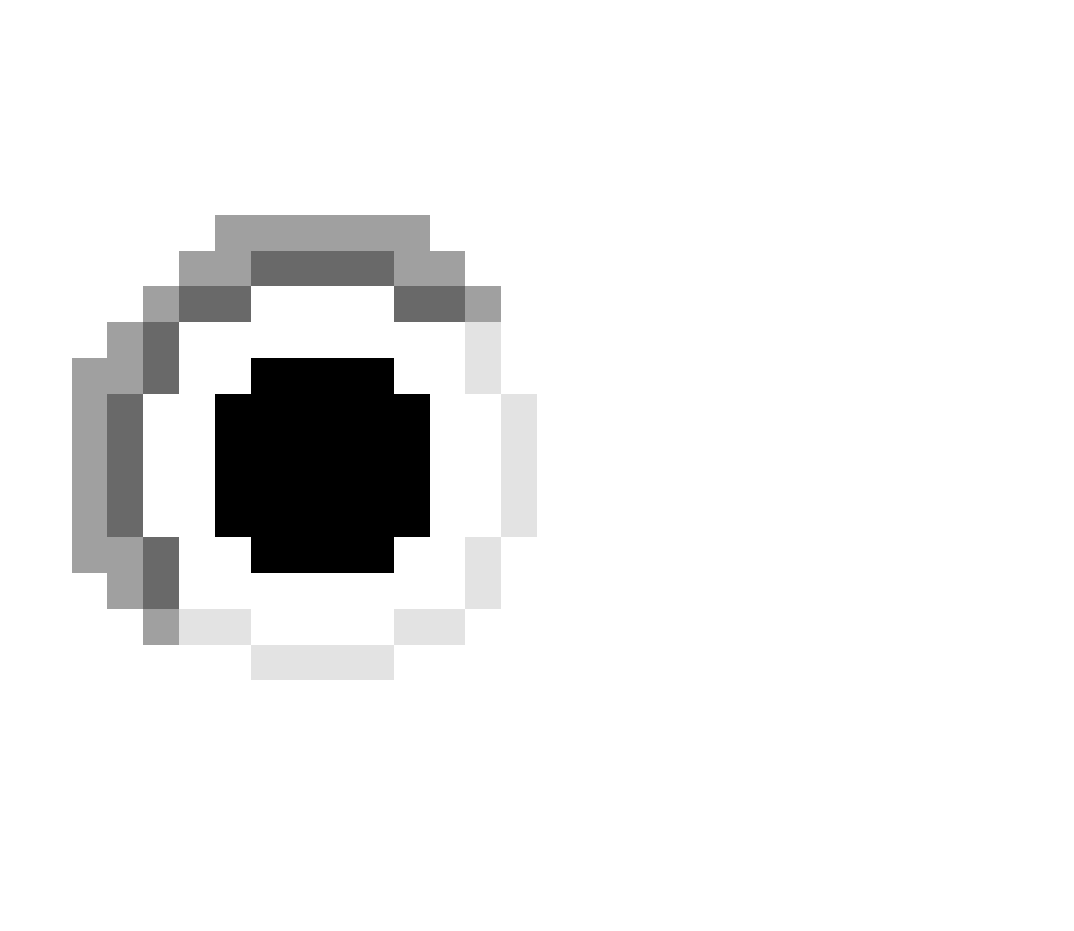
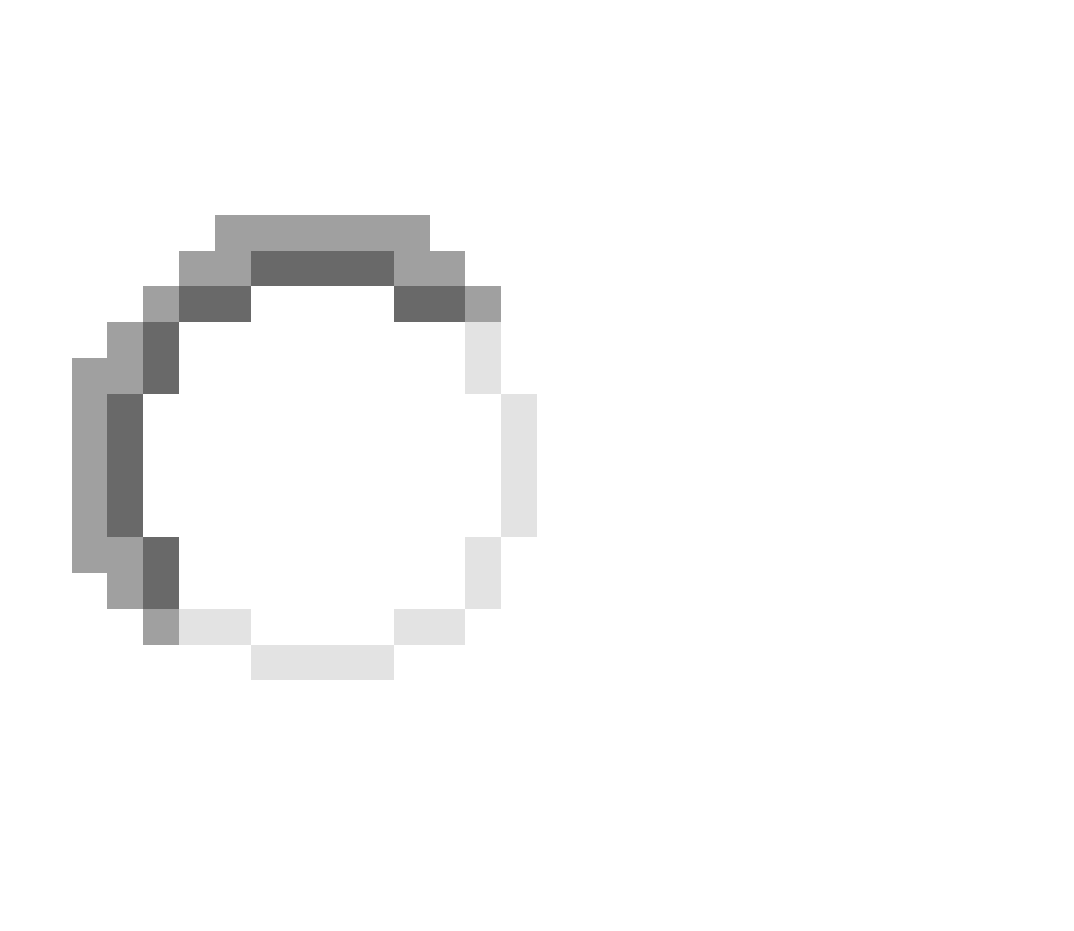
Введите свой е-mail:  
  
 

**Элемент формы radio:**

<form name="primer4" method="post" action="obrabotchik.php">  
<p> Какой диск вы хотите получить?</p>  
<p>  
**<input name="disc" type="radio" value="cd" checked>**  
CD<br>  
**<input name="disc" type="radio" value="dvd">**  
DVD </p></form>

**Результат:**

Какой диск вы хотите получить?

 CD  
 DVD

Данный тип элементов html формы называется радиопереключатель(переключает: либо одно значение, либо другое - два одновременно быть не может). Поэтому атрибут type имеет значение **type="radio"**. Раз мы выбираем между CD и DVD, то радиопереключателя у нас два, поэтому два раза пользуемся элементом INPUT. Как видите у них одинаковое имя - **name="disc"** и разные значения **value**. Почему это так? давайте подумаем логически:

Нас интересует какой диск хочет получить клиент, CD или DVD. Поэтому у нас одинаковое значение имени **name="disc"** и разные значение (**value="cd"** и **value="dvd"**). Т.е. если мы выбираем первый вариант, то переменная **disc** примет значение **cd** а если второй- то **dvd**.

Если вы хотите сделать, чтобы по умолчанию у Вас радиопереключатель был установлен на каком-либо варианте, то просто допишите атрибут-флаг **checked**(включен).

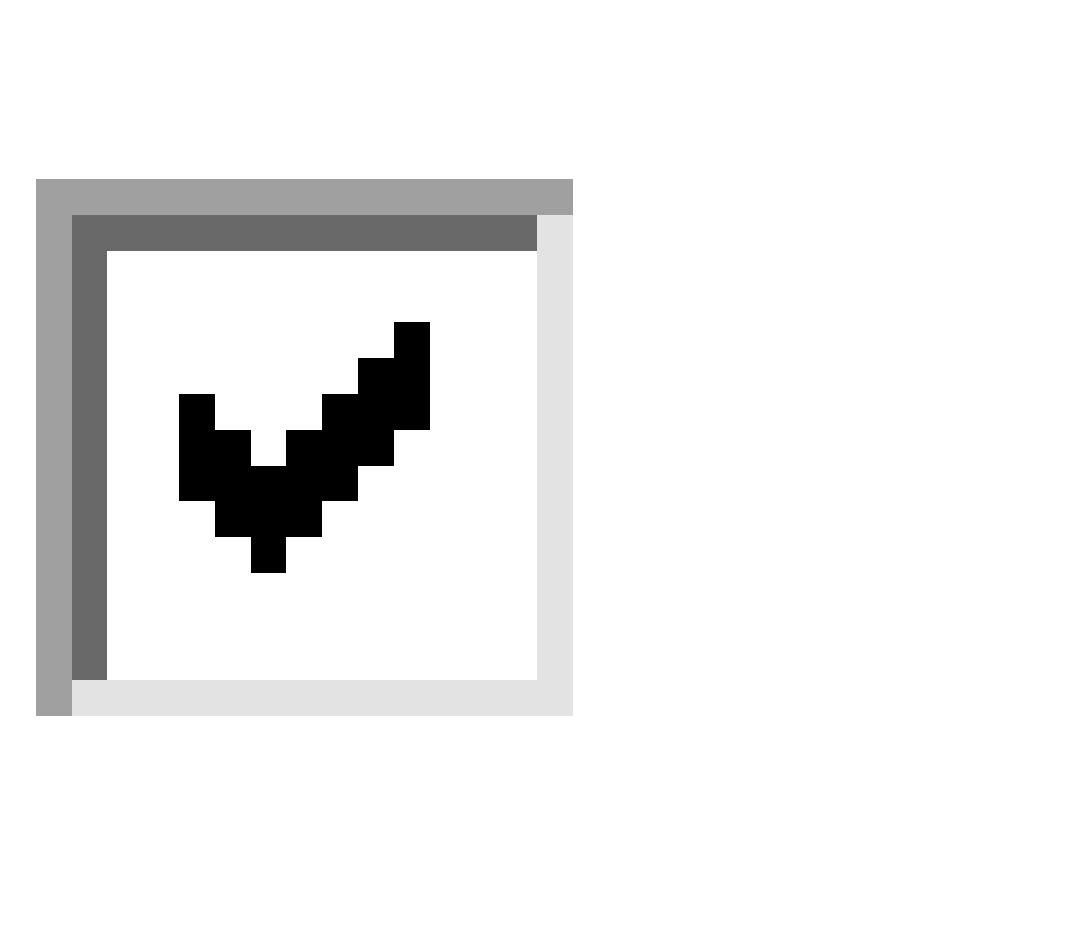
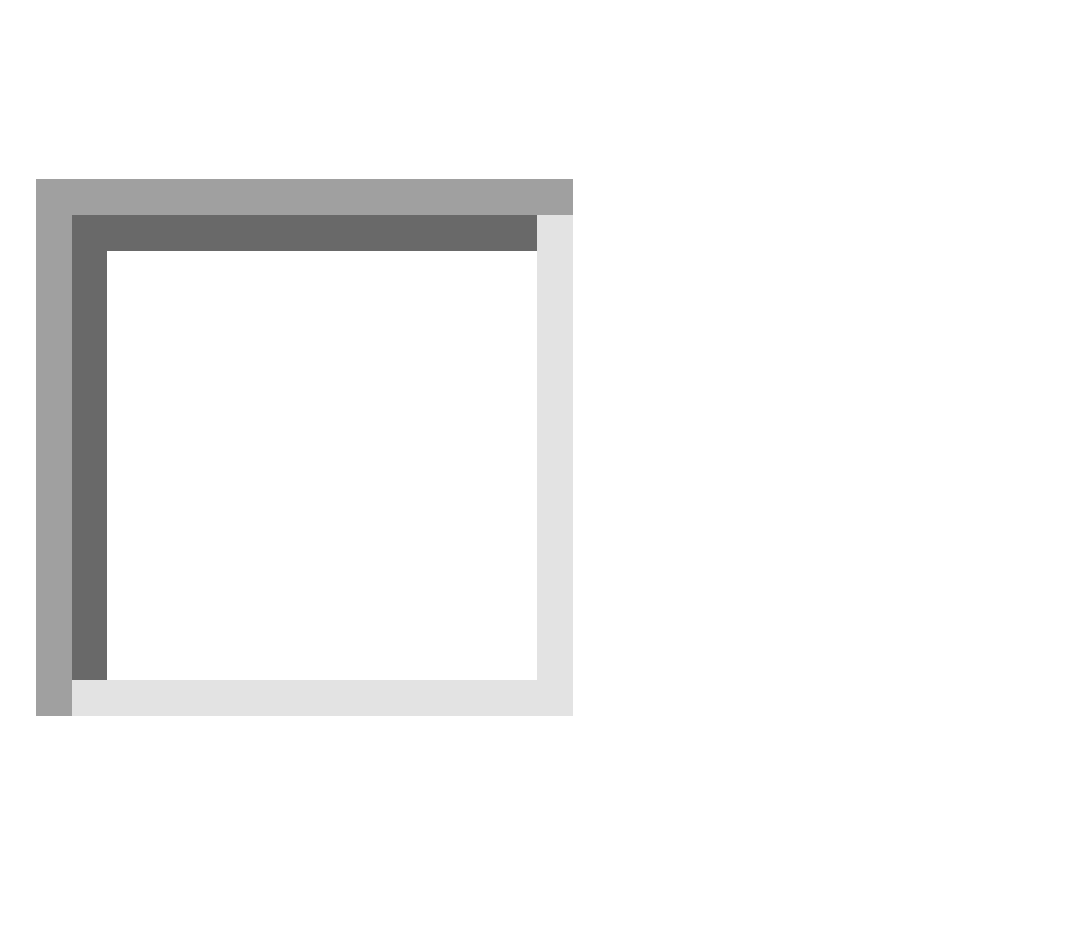
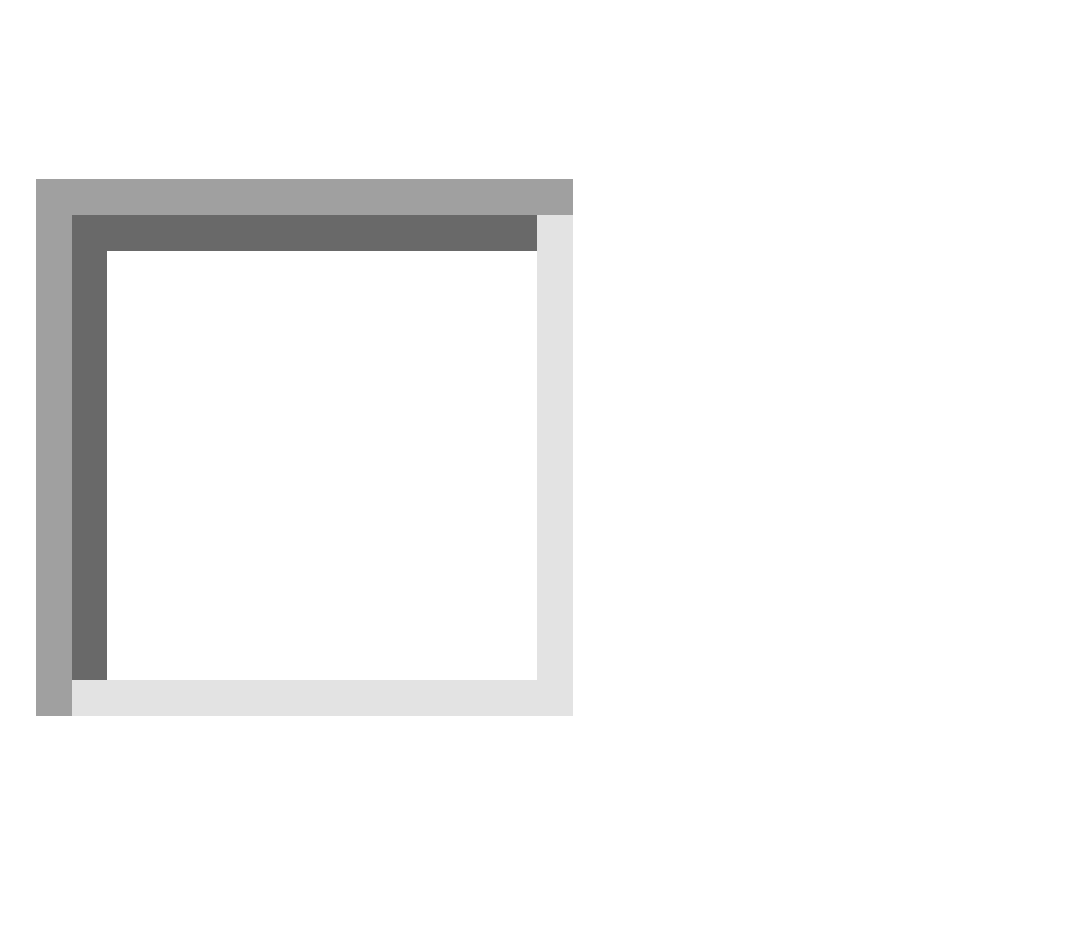
Примечание: в радиопереключателе обязательно должен присутствовать атрибут **value** иначе ничего работать не будет.

**Элемент формы checkbox:**

<form name="primer5" method="post" action="obrabotchik.php">  
<p>Какие обучающие курсы вы хотите видеть на диске?</p>  
**<input type="checkbox" name="fotoshop" value="yes" checked>**  
**Курсы по Фотошопу** <br>  
**<input type="checkbox" name="dreamweaver" value="yes">**  
**Курсы по Adobe Dreamweaver** <br>  
**<input type="checkbox" name="php" value="yes">Курсы по PHP**  
</form>

Результат:

Какие обучающие курсы вы хотите видеть на диске?

 Курсы по Фотошопу   
 Курсы по Adobe Dreamweaver   
 Курсы по PHP

Данный тип элементов html формы называется **checkbox** и отличается от радиопереключателя тем, что здесь можно выбрать несколько вариантов. Как видите **type="checkbox"** означает что тип элемента - Чекбокс, атрибут **name** нужен для того, чтобы обработчик мог идентифицировать данное поле и наконец **value** - определяет тот параметр, который будет отправлен при поставленной галочке.

В данном элементе, атрибут **value** не является обязательным, в отличие от радиопереключателя.

**Элемент формы select:**

<form name="primer6" method="post" action="obrabotchik.php">  
<p>Выберите способ доставки:</p>  
<p>  
**<select name="dostavka" size="1">**  
**<option value="srochnaya" >** Срочная **</option>**  
**<option value="ne srochnaya"** **selected>** Не срочная **</option>**  
**<option value="kurierom">** Курьером **</option>**  
**</select>**  
</p>   
</form>

**Результат:**

Выберите способ доставки:



Базовым элементом здесь является **SELECT**(выбор). У него обязательно должен быть закрывающий тег! Внутри него содержатся элементы **OPTION** (опции выбора). Как вы поняли, без элементов option элемент select потеряет смысл, т.к. выбирать будет не из чего.

Для систематизации списков применяется элемент **<optgroup>...</optgroup>,** который создает заголовки в списках.

Как видите, у элемента SELECT есть уже знакомый нам атрибут **name**, а у элементов OPTION тоже знакомый нам атрибут **value**. Они нужны для того, чтобы обработчик мог идентифицировать выбранный вариант.

Атрибут **SIZE** задает количество одновременно видимых пунктов меню. В нашем случае это 1. Если количество элементов списка превышает установленное количество, появляется полоса прокрутки. Значение атрибута задается целым положительным числом.

Атрибут **MULTIPLE** дает возможность выбора одного или нескольких пунктов, для этого при выборе нужно нажать и удерживать нажатой клавишу Ctrl. Превращает список из выпадающего в прокручивающийся. Если не задан еще атрибут size, список будет иметь достаточную высоту, чтобы вместить все элементы.

Атрибуты элементы **OPTION**

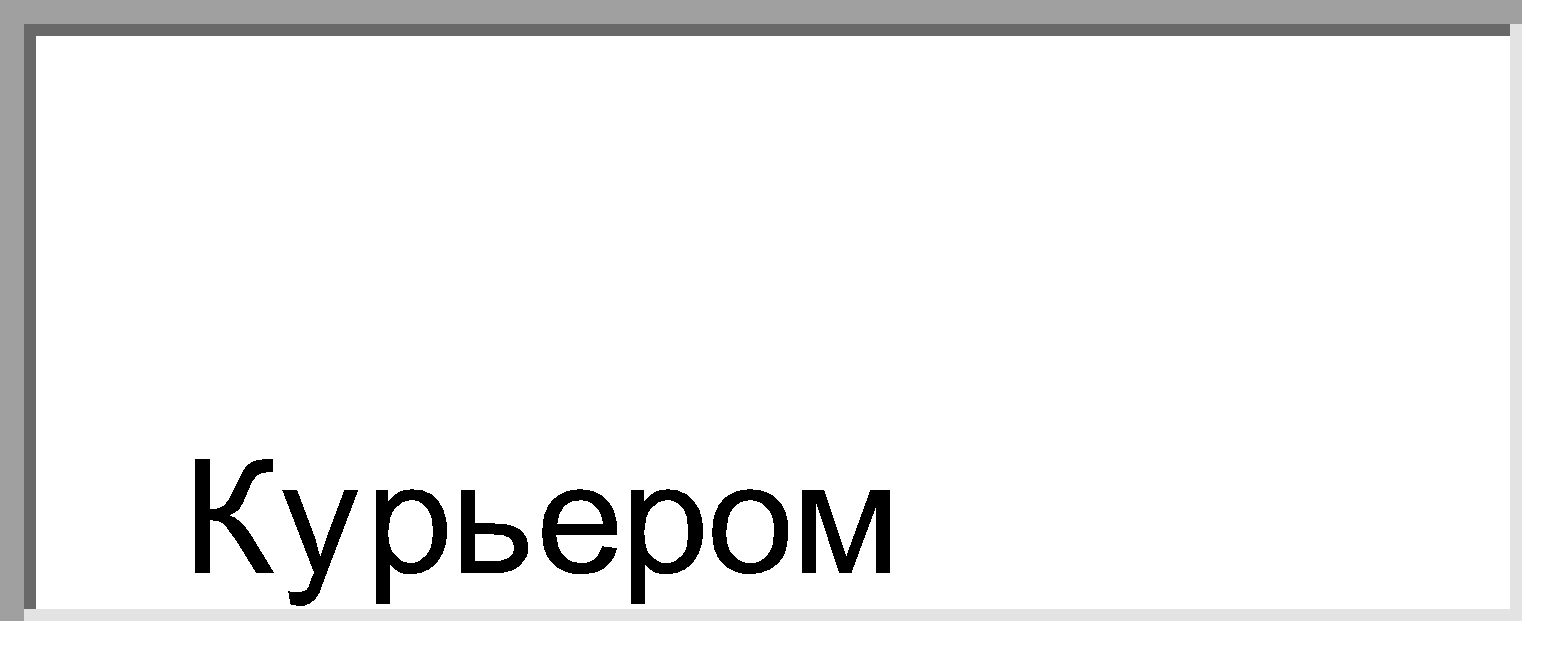
Атрибут **SELECTED** означает что данное значение, будет выбрано по умолчанию.

Атрибут **DISABLED** делает недоступным для выбора элемент списка.

<form name="primer7" method="post" action="obrabotchik.php">  
<p>Выберите способ доставки:</p>  
<p>  
**<select name="dostavka" size="3"** **multiple** **>**  
**<option value="srochnaya" >** Срочная **</option>**  
**<option value="ne srochnaya">** Не срочная **</option>**  
**<option value="kurierom">** Курьером **</option>**  
**</select>**  
</p>   
</form>

Результат:

Выберите способ доставки:



**Элемент формы textarea:**

Элемент **TEXTAREA** – элемент формы, позволяющий вводить многострочный текст. У этого элемента обязательно наличие закрывающего тега.

<form name="primer8" method="post" action="obrabotchik.php">  
Введите адрес для доставки:<br>  
**<textarea name="adress"** **cols="45" rows="5"></textarea>**  
</form>

**Результат:**

Введите адрес для доставки:  

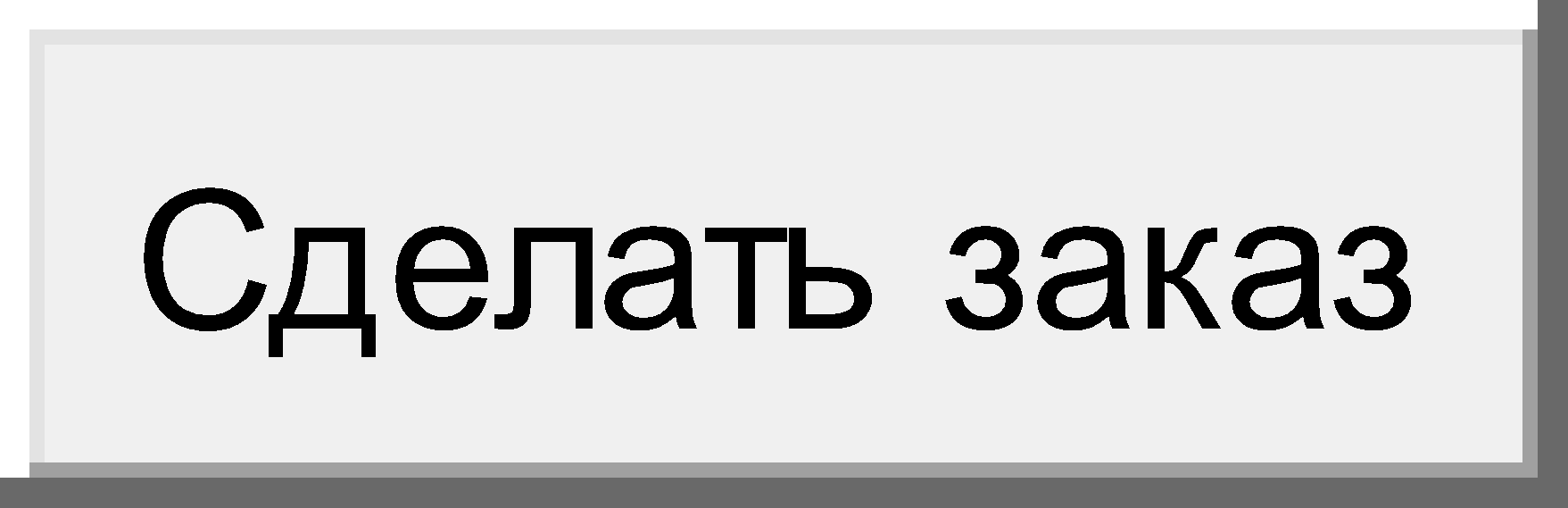
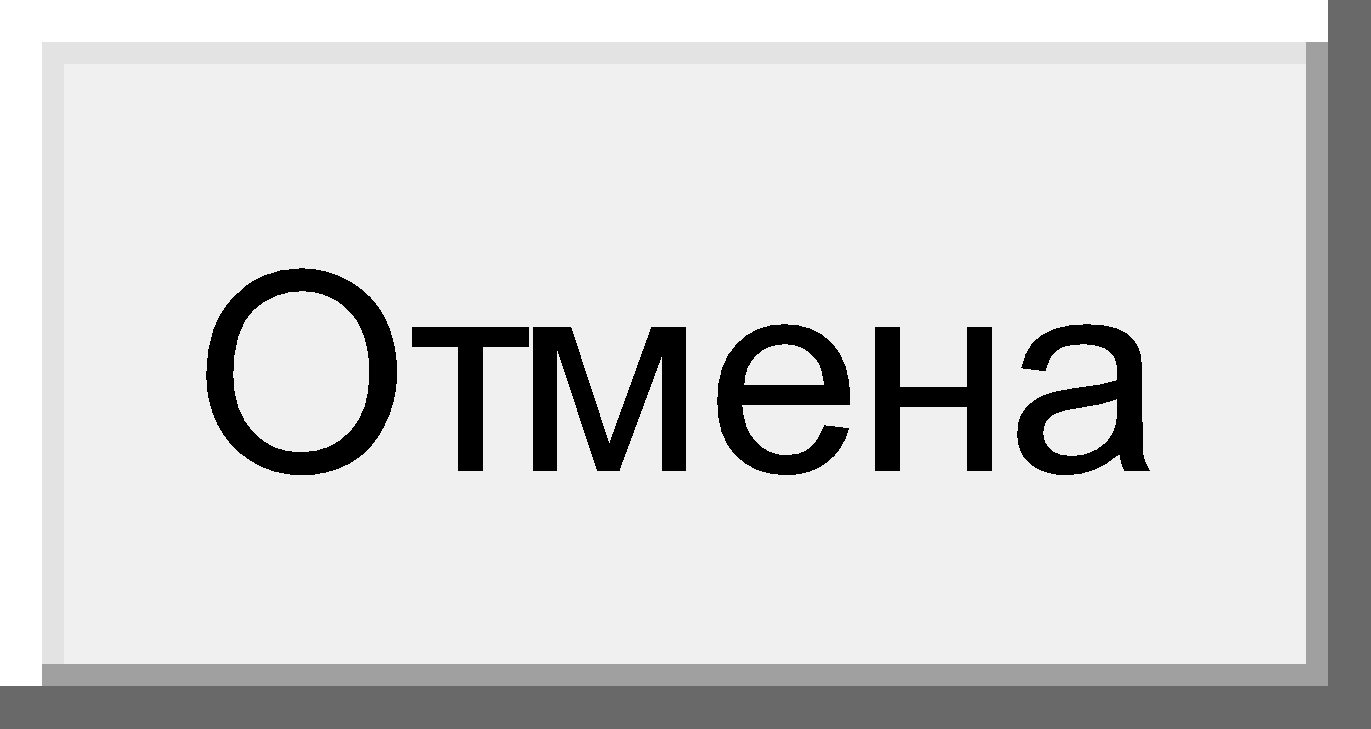

У него есть атрибут **name,** который поможет идентифицировать заполненное поле на сервере. Как видите здесь нигде нет параметра value, т.к. в качестве значения обработчику посылается текст, который мы введем в текстовое поле.

Атрибуты **COLS** и **ROWS** определяют ширину и высоту поля. К примеру, cols="45" rows ="5" означает что ширина будет 45 колонок, а высота 5 рядов. Колонки и ряды естественно формируются из символов. Т.е. ширина 25 колонок, означает ширину в 25 символов.

**Кнопки:**

<form name="primer8" method="post" action="obrabotchik.php">  
**<input type="submit" value="Сделать заказ">**  
**<input type="reset" value="Отмена">**  
</form>

Результат:

Для создания кнопок используется уже знакомый нам элемент **INPUT**. Если необходимо создать кнопку, которая будет отправлять данные обработчику, то пишем **type="submit"**(тип-отправить), а если хотите, чтоб кнопка выполняла функцию сброса, то **type="reset"**.

**Элемент button**

Элемент **<button>...</button>** создает кликабельные кнопки. В отличие от кнопок, созданных <input> (<input type="submit"></input>,  <input type="reset">, <input type="button">), внутрь элемента <button> можно поместить контент — текст или изображение.

Для корректного отображения элемента <button> разными браузерами нужно указывать атрибут **TYPE**, например, <button type="submit"></button>.

Атрибут **TYPE** задает поведение кнопки**:**

* **Submit-** отправка формы (т.е. всех элементов и их значений) серверному скрипту.
* **Reset –** сброс формы (т.е. возвращает первоначальное значение)
* **Button –** кликабельная кнопка

Кнопки позволяют пользователям передавать данные в форму, очищать содержимое формы или предпринимать какие-либо другие действия. Можно создавать границы, изменять фон и выравнивать текст на кнопке.