

Тема: Формы в HTML

Формы

Назначение форм

HTML-формы предназначены для пересылки данных от удаленного пользователя к Web-серверу. С их помощью можно организовать простейший диалог между пользователем и сервером, например, регистрацию пользователя на сервере или выбор нужного документа из представленного списка. Формы поддерживаются всеми популярными браузерами.

Синтаксис формы

В HTML-документе для задания формы используются тэги `<form>...</form>`. Документ может содержать несколько форм, но они не могут быть вложены одна в другую.

Тэг `<form>` имеет параметры `action`, `method` и `enctype`. Отдельные браузеры (Netscape, Internet Explorer) поддерживают дополнительные параметры, например, `class`, `name`, `style` и др.

В общем виде форма задается следующим образом:

```
<form action="url" method="метод_передачи" enctype="МIME-тип">
содержание_формы
</form >
```

Параметр `action` является единственным обязательным. Его значением является URL-адрес CGI-программы, которая будет обрабатывать информацию, извлеченную из данной формы.

Взаимодействие между браузером и web-сервером

Взаимодействие между клиентом-браузером и Web-сервером осуществляется по правилам, заданным протоколом HTTP, и состоит из запросов клиента и ответов сервера. Запрос разбивается на три части. В первой строке запроса содержится HTTP-команда, называемая методом, URL-адрес запрашиваемого файла и номер версии протокола HTTP. Вторая часть — заголовок запроса. Третья часть — тело запроса, собственно данные, посылаемые серверу.

MIME-типы

Одним из первых применений Интернета была электронная почта, ориентированная на пересылку текстовых сообщений. Часто возникает необходимость вместе с текстом переслать данные в нетекстовом формате, например, упакованный zip-файл, рисунок в формате GIF, JPEG и т. д. Для того, чтобы переслать средствами электронной почты такие файлы без искажения, они кодируются в соответствии с некоторым стандартом. Стандарт MIME (Multipurpose Internet Mail Extensions, многоцелевые расширения электронной почты для Интернета) определяет набор MIME-типов, соответствующих различным типам данных, и правила их пересылки по электронной почте. Для обозначения MIME-типа используется запись вида тип/подтип, где тип определяет общий тип данных, например, `text`, `image`, `application` (тип `application` обозначает специфический внутренний формат данных, используемый некоторой программой), а подтип — конкретный формат внутри типа данных, например, `application/zip`, `image/gif`, `text/html`.

MIME-типы нашли применение в Web, где они называются также медиа-типами, для идентификации формата документов, передаваемых по протоколу HTTP. В HTML-форме параметр `enctype` определяет медиа-тип, который используется для кодирования и пересылки специального типа данных — содержимого формы.

Значением параметра `enctype` является медиа-тип, определяющий формат кодирования данных при передаче их от браузера к серверу. Браузер кодирует данные,

чтобы исключить их искажение в процессе передачи. Возможны два значения этого параметра: application/x-www-form-urlencoded (по умолчанию) и multipart/form-data.

Второй метод нужен только в том случае, если к содержимому формы присоединяется локальный файл, выбранный при помощи элемента формы `<input type="files">`. В остальных случаях следует использовать метод кодирования по умолчанию.

URL-кодирование

Схема кодирования application/x-www-form-urlencodedодинакова для обоих методов пересылки (GETи POST) и заключается в следующем. Для каждого элемента формы, имеющего имя, заданное параметром name, формируется пара "name=value", где value— значение элемента, введенное пользователем или назначенное по умолчанию. Если значение отсутствует, соответствующая пара имеет вид "name=". Для радиокнопок и переключателей используются значения только выбранных элементов. Если элемент выбран, а значение параметра valueне определено, по умолчанию используется значение "on".

Все пары объединяются в строку, в качестве разделителя служит символ "&". Так как имена и значения представляют собой обычный текст, то они могут содержать символы, недопустимые в составе URL (метод GETпересылает данные как часть URL). Такие символы заменяются последовательностью, состоящей из символа % и их шестнадцатеричного ASCII-кода. Символ пробела может заменяться не только кодом %20, но и знаком + (плюс). Признак конца строки, встречающийся в поле textarea, заменяется кодом %0D%0A. Такое кодирование называется URL-кодированием.

Методы передачи данных

Закодированная информация пересылается серверу одним из методов GETили POST. Основное отличие заключается в том, как метод передает информацию CGI-программе.

При использовании метода GETданные формы пересылаются в составе URL-запроса, к которому присоединяются после символа "?" в виде совокупности пар переменная = значение, разделенных символом "&". В этом случае первая строка запроса может иметь следующий вид:

GET /cgi-bin/cgi-program.pl?name=Ivan&surname=Ivanov HTTP/1.1

Часть URL после символа "?" называется строкой запроса. Web-сервер, получив запрос, присвоит переменной среды QUERY_STRINGзначение строки запроса и вызовет CGI-программу, обозначенную в первой части URL до символа "?".

При использовании метода POSTданные формы пересылаются серверу в теле запроса, после чего передаются сервером в CGI-программу через стандартный ввод.

Методы GETи POSTимеют свои достоинства и недостатки. Метод GETобеспечивает лучшую производительность при пересылке форм, состоящих из небольшого набора коротких полей. При пересылке большого объема данных следует использовать метод POST, так как браузер или сервер могут накладывать ограничения на размер данных, передаваемых в составе URL, и отбрасывать часть данных, выходящую за границу. Метод POST, к тому же, является более надежным при пересылке конфиденциальной информации.

Поля ввода формы

Форма отображается в окне браузера в виде набора стандартных элементов управления, используемых для заполнения полей формы значениями, которые затем передаются Web-серверу. Значение вводится в поле ввода пользователем или назначается по умолчанию. Для создания полей средствами языка HTML существуют специальные тэги: `<input>`, `<select>`, `<textarea>`, которые употребляются только внутри тэга `<form>`.

Тэг `<input>`

`<input type="тип_поля_ввода" name="имя_поля_ввода" другие_параметры>`

Это наиболее употребительный тэг, с помощью которого можно генерировать внутри формы поля для ввода строки текста, пароля, имени файла, различные кнопки. Он имеет два обязательных параметра: `type` и `name`. Параметр `type` определяет тип поля: селекторная кнопка, кнопка передачи и др. Параметр `name` определяет имя, присваиваемое полю. Оно не отображается браузером, а используется в качестве идентификатора значения, передаваемого Web-серверу. Остальные параметры меняются в зависимости от типа поля. Ниже приведено описание типов полей, создаваемых при помощи тэга `<input>`, и порождаемых ими элементов ввода.

`type="text"`

Создает элемент для ввода строки текста.

Дополнительные параметры:

- `maxlength="n"` - задает максимальное количество символов, разрешенных в текстовом поле. По умолчанию — не ограничено.
- `size="n"` - максимальное количество отображаемых символов.
- `value = "начальное_значение"`. Первоначальное значение текстового поля.

`type="password"`

Создает элемент ввода строки текста, отличающийся от предыдущего только тем, что все вводимые символы представляются в виде символа *. Поле `password` обеспечивает безопасности введенного текста, так как на сервер он передается в незашифрованном виде.

`type="files"`

Создает поле для ввода имени локального файла, сопровождаемое кнопкой Browse. Выбранный файл присоединяется к содержимому формы при пересылке на сервер. Имя файла можно ввести непосредственно или выбрать его из диалогового окна. Для корректной передачи присоединенного файла следует установить значения параметров формы равными `enctype="multipart/form-data"` и `method="post"`. В противном случае будет передана введенная строка, то есть маршрутное имя файла, а не его содержимое. Дополнительные параметры `maxlength` и `size` имеют тот же смысл, что и для элементов типа `text` и `password`.

`type="checkbox"`

Создает поле для установки флажка. Элементы `checkbox` можно объединить в группу, установив одинаковое значение параметра `name`.

Дополнительные параметры:

- `value="строка"`. Значение, которое будет передано серверу, если данная кнопка выбрана. Если кнопка не выбрана, значение не передается. Обязательный параметр. Если флажки образуют группу, то передаваемым значением является строка разделенных запятыми значений параметра `value` всех установленных флажков.
- `checked`. Если указан параметр `checked`, элемент является выбранным по умолчанию.

`type="radio"`

Создает элемент-переключатель, существующий только в составе группы подобных элементов, из которых может быть выбран только один. Все элементы группы должны иметь одинаковое значение параметра `name`.

Отображается в виде круглой кнопки. Дополнительные параметры:

- `value="строка"`. Обязательный параметр, значение которого передается серверу при выборе данной кнопки. Должен иметь уникальное значение для каждого члена группы.
- `checked`. Устанавливает элемент выбранным по умолчанию. Один и только один элемент в группе должен иметь этот параметр.

`type="submit"`

Создает кнопку передачи, нажатие которой вызывает пересылку на сервер всего содержимого формы. По умолчанию отображается в виде прямоугольной кнопки с надписью Submit.

Дополнительный параметр позволяет изменить надпись на кнопке. Параметр name для данного элемента может быть опущен. В этом случае значение кнопки не включается в список параметров формы и не передается на сервер.

Если параметры name и value присутствуют, например,
`<input type="submit" name="submit_button" value="ok">`,
то в список параметров формы, передаваемых на сервер, включается параметр `submit_button="ok"`. Внутри формы могут существовать несколько кнопок передачи.

type="reset"

Создает кнопку сброса, нажатие которой отменяет все сделанные изменения, восстанавливая значения полей формы на тот момент, когда она была загружена. По умолчанию отображается в виде прямоугольной кнопки с надписью Reset. Надпись можно изменить при помощи дополнительного параметра

`value="название_кнопки"`

Значение кнопки Reset никогда не пересылается на сервер, поэтому у нее отсутствует параметр name.

type="image"

Создает элемент в виде графического изображения, действующий аналогично кнопке Submit. Дополнительные параметры:

- `src="url_изображения"`. Задает URL-адрес файла с графическим изображением элемента.
- `align="тип_выравнивания"`. Задает тип выравнивания изображения относительно текущей строки текста.

Если на изображении элемента щелкнуть мышью, то координаты указателя мыши в виде `name.x=n&name.y=m` включаются браузером в список параметров формы, посылаемых на сервер.

type="hidden"

Создает скрытый элемент, не отображаемый пользователю. Информация, хранящаяся в скрытом поле, всегда пересылается на сервер и не может быть изменена ни пользователем, ни браузером. Скрытое поле можно использовать, например, в следующем случае. Пользователь заполняет форму и отправляет ее серверу. Сервер посылает пользователю для заполнения вторую форму, которая частично использует информацию, содержащуюся в первой форме. Сервер не хранит историю диалога с пользователем, он обрабатывает каждый запрос независимо и при получении второй формы не будет знать, как она связана с первой. Чтобы повторно не вводить уже введенную информацию, можно заставить CGI-программу, обрабатывающую первую форму, переносить необходимые данные в скрытые поля второй формы. Они не будут видимы пользователем и, в то же время, доступны серверу. Значение скрытого поля определяется параметром value.

Тэг <select>

Тэг <select> предназначен для того, чтобы организовать внутри формы выбор из нескольких вариантов без применения элементов типа checkbox и radio. С помощью тэга <select> варианты выбора более компактно представляются в окне браузера в виде элементов ниспадающего меню или списка прокрутки. Пример:

```
<select name="sel_name" size="2">
  <option value="Red">Red</option>
  <option value="Yellow">Yellow</option>
  <option value="Green">Green</option>
</select>
```

Тэг имеет следующие параметры:

name="имя_поля". Обязательный параметр. При выборе одного или нескольких элементов формируется список выбранных значений, который передается на сервер под именем name.

size="n". Устанавливает число одновременно видимых элементов выбора. Если n="1", то отображается ниспадающее меню, если n>1, то — список прокрутки с одновременно видимыми элементами.

Элементы меню задаются внутри тэга <select>при помощи тэга <option>:

<option selected value="строка"/> содержимое_тэга

Параметр value содержит значение, которое пересылается серверу, если данный элемент выбран из меню или списка. Если значение этого параметра не задано, то по умолчанию оно устанавливается равным содержимому тэга <option>.

Тэг <textarea>

Тэг <textarea>создает внутри формы поле для ввода многострочного текста, отображаемое в окне браузера в виде прямоугольной области с горизонтальной и вертикальной полосами прокрутки. Для пересылки на сервер каждая введенная строка дополняется символами %0D%0A (ASCII-символы "Возврат каретки" и "Перевод строки" с предшествующим символом %), полученные строки объединяются в одну строку, которая и отправляется на сервер под именем, задаваемым параметром name.

<textarea name="ta_name" rows="m" cols="n">

текст

</textarea>

Параметры:

- name. Необходимый параметр, используемый для идентификации данных при пересылке на сервер.
- cols="n". Задаёт число столбцов видимого текста.
- rows="m". Задаёт число строк видимого текста.

Между тэгами <textarea>и </textarea>можно поместить текст, который будет отображаться по умолчанию.

Форма

Формы нужны, когда от пользователя требуется ввод какой-либо информации (интересно, а где еще он может что-то ввести?). Пользователь вводит нужную информацию в специально отведенных для этого полях формы, затем нажимает нужную кнопку и посылает заполненную форму на сервер. Сервер принимает данные формы, обрабатывает их и генерирует ответ в виде HTML-файла (например, "введен неправильный пароль"). Этот файл и получает пользователь в ответ на отправку формы. Узнали? Наверняка Вам встречалось это, или что-то очень похожее. Итак, формы.

Все формы начинаются тегом `<FORM>` и завершаются тегом `</FORM>`.
Формы нельзя вкладывать одну в другую (это Вам не таблицы)!!!

Структура формы такова:

```
<FORM METHOD="get|post" ACTION="URL" ENCTYPE="MIME">  
Элементы_формы_и_другие_элементы_HTML  
</FORM>
```

METHOD

Метод отправки сообщения с данными из формы. В зависимости от используемого метода вы можете посылать результаты ввода данных в форму двумя путями:

GET: Информация из формы добавляется в конец URL, который был указан в описании заголовка формы: `http://www.webboard.ru/wb.php?board=10767`

Метод GET **рекомендуется** для обмена небольшими (до 256 байт) порциями данных, а также для передачи данных в другой HTML-документ, который может с помощью JavaScript их обработать на стороне клиента.

Не рекомендуется для передачи секретной информации (идентификатор, пароль, ...).

POST: Данный метод передает всю информацию о форме немедленно после обращения к указанному URL. CGI-программа на сервере получает данные формы из стандартного потока ввода. Данный метод рекомендуется к использованию на стороне сервера.

ENCTYPE

Указание типа передаваемой информации. В зависимости от MIME-типа информация будет преобразована соответствующим образом.

ACTION

ACTION описывает URL, который будет вызываться для обработки формы. Данный URL почти всегда указывает на CGI-программу, обрабатывающую данную форму.

Пример:

```
<FORM METHOD="post" ACTION="http://www.mysite.com/cgi-bin/process.pl">
```

Если же Вы хотите принять данные формы по электронной почте, то используйте один из вариантов:

1. Стандартный.
Action="mailto:xxx@xxx.xx" enctype="application/x-www-form-urlencoded"
2. Упрощенный.
Action="mailto:xxx@xxx.xx" enctype="text/plain"
(сравните принимаемые данные далее в примере с формой)

Элементы формы

Все элементы формы должны иметь имя (Name).

В HTML 4 вместо Name рекомендуется использовать ID (идентификатор).

INPUT

Пустой тег

В зависимости от параметра TYPE создает различные типы полей ввода.

Типы полей ввода:

<INPUT TYPE=TEXT> : однострочное поле ввода.

Атрибуты:

SIZE=20 _____ размер отображаемого поля ввода на экране (по умолчанию=20)

MAXLENGTH=15 _____ максимальная длина вводимого значения в символах

Пример:

введите имя: <input name="имя" type="text" size=40>

введите серийный код: <input name="код" type="text" size=20 maxlength=10>

введите имя:	Аглабердыев Акакий Марсэнович
введите серийный код:	2223333222

<INPUT TYPE=RESET> : кнопка сброса

Данный тип обозначает кнопку, при нажатии которой все поля формы примут значения, описанные для них по умолчанию.

<INPUT TYPE=SUBMIT> : кнопка для отправки формы

Данный тип обозначает кнопку, при нажатии которой будет вызвана CGI-программа (или URL), описанная в параметре ACTION. Атрибут VALUE может содержать надпись на кнопке.

Пример:

<input type="submit" value="послать">

<input type="reset" value="очистить">

послать	очистить
---------	----------

<INPUT TYPE=RADIO> : одиночный выбор значения из нескольких

Для создания набора альтернатив вам необходимо создать несколько полей ввода с атрибутом TYPE="RADIO" с **разными** значениями атрибута VALUE, но с **одинаковыми** значениями атрибута NAME. В CGI-программу будет передано значение типа NAME=VALUE, причем VALUE примет значение атрибута VALUE того поля ввода, которое в данный момент будет выбрано (будет активным). Выбор одного из полей ввода типа RADIO автоматически отменит выбор всех остальных полей данного типа с тем же именем (атрибут NAME).

Атрибуты:

NAME="name" _____ имя возвращаемой переменной

VALUE="yes" _____ значение возвращаемой переменной (т.е. поля)

CHECKED _____ выбрано изначально

Пример:

<INPUT TYPE=RADIO NAME="Мои друзья" VALUE="Ed" CHECKED>Ed Holleran

<INPUT TYPE=RADIO NAME="Мои друзья" VALUE="Rick"> Rick Weinberg

<INPUT TYPE=RADIO NAME="Мои друзья" VALUE="Tom"> Tom Studd<P>

<input checked="" type="radio"/>	Ed Holleran
<input type="radio"/>	Rick Weinberg
<input type="radio"/>	Tom Studd

<INPUT TYPE=CHECKBOX> : множественный выбор

Пример:

```
<INPUT TYPE=CHECKBOX NAME="ED" VALUE="YES" CHECKED> Ed Holleran<BR>
<INPUT TYPE=CHECKBOX NAME="Rick" VALUE="YES"> Rick Weinberg<BR>
<INPUT TYPE=CHECKBOX NAME="Tom" VALUE="YES" CHECKED> Tom Studd<BR>
```

<input checked="" type="checkbox"/>	Ed Holleran
<input type="checkbox"/>	Rick Weinberg
<input checked="" type="checkbox"/>	Tom Studd

<INPUT TYPE=HIDDEN> : "скрытое" поле

Поля данного типа не отображаются браузером и не дают пользователю изменять присвоенные данному полю по умолчанию значение. Это поле используется для передачи в CGI-программу вспомогательной информации: ID пользователя, пароля или другой информации.

<INPUT TYPE=PASSWORD> : для ввода пароля

То же самое, что и атрибут TEXT, но вводимое пользователем значение не отображается браузером на экране.

Атрибуты:

SIZE=20 _____ размер отображаемого поля ввода на экране (по умолчанию=20)
MAXLENGTH=15 _____ максимальная длина вводимого значения в символах

<INPUT TYPE=IMAGE> : графическая кнопка

При нажатии мышью на графической кнопке будет вызвана ассоциированная форме CGI-программа. Сама иллюстрация задается атрибутом SRC.

Пример: `<input type="image" name="ED" src="buttons/win98ready.gif" width="60" height="60">`

<INPUT TYPE=BUTTON> : кнопка

Просто кнопка. Параметр "value=" задает надпись на кнопке.

Используется для вызова программы на JavaScript, например, для проверки данных формы.

<INPUT TYPE=FILE> : Передача файлов при помощи форм

В параметрах <FORM> необходимо указать ENCTYPE="multipart/form-data"

Внимание! Данная возможность требует поддержки получения файлов WEB-сервером!

Пример:

```
<FORM ENCTYPE="multipart/form-data" ACTION="url" METHOD=POST>
```

Отправить данный файл:

```
<INPUT NAME="userfile" TYPE="file">
```

```
<P>
```

```
<INPUT TYPE="submit" VALUE="Отправить файл">
```

```
</FORM>
```

Отправить данный файл:	<input type="text" value="D:\xenofex.zip"/>	<input type="button" value="Browse"/>
<input type="button" value="Отправить файл"/>		

LABEL

Связывает поле метки LABEL с полем формы.

Закрывающий тег: обязателен

Поле формы определяется атрибутом FOR, значением которого является идентификатор ID элемента формы.

По щелчку на метке активизируется связанный с ней элемент формы. То есть пользователь сможет использовать как поле ввода, так и связанный с ним текст.

Пример:

```
<input type="checkbox" id="e1" value="1"> <label for=e1>знание HTML и CSS </label><br>
<input type="checkbox" id="e2" value="1"> <label for=e2>знание Perl</label><br>
<input type="checkbox" id="e3" value="1"> <label for=e3>знание ASP</label><br>
```


SELECT

Закрывающий тег: обязателен

Выбор значений из разворачивающегося списка значений, заданных при помощи OPTION.
Возможен одиночный или множественный выбор.

Атрибуты:

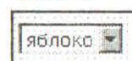
SIZE=5 _____ начальная высота списка

MULTIPLE _____ разрешить множественный выбор нескольких значений

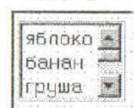
Пример:

```
<select name="fruit" size="1">
  <option value="яблоко"> яблоко </option>
  <option value="банан"> банан </option>
  <option value="груша"> груша </option>
  <option value="лимон"> лимон </option>
  <option value="киви"> киви </option>
  <option value="папайя"> папайя </option>
</select>
```

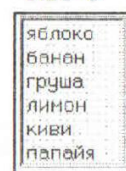
size=1



size=3



size=6



Для множественного выбора используйте **SELECT MULTIPLE**.

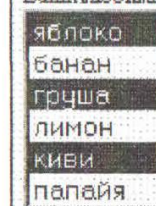
Атрибут **SIZE** определяет количество одновременно видимых на экране элементов, атрибут **MULTIPLE** - максимальное количество одновременно выбранных элементов.

Если выбрано одновременно несколько значений, то серверу передается соответствующее количество параметров **NAME=VALUE** с одинаковыми значениями **NAME**, но разными **VALUE**.

Пример:

```
Ваши любимые фрукты:<br>
<select name="fruit" size="6" multiple>
  <option value="яблоко"> яблоко </option>
  <option value="банан"> банан </option>
  <option value="груша"> груша </option>
  <option value="лимон"> лимон </option>
  <option value="киви"> киви </option>
  <option value="папайя"> папайя </option>
</select>
```

Ваши любимые фрукты:



TEXTAREA

Закрывающий тег: обязателен

Многострочное поле ввода.

Пример:

```
<TEXTAREA NAME="address" ROWS=5 COLS=25>
```

Москва,
ул. 2 Бауманская, д. 5.
офис 703. Тел. 263-6443

```
</TEXTAREA>
```

Атрибуты:

NAME="address" _____ имя поля ввода

ROWS=5 _____ высота поля ввода в символах

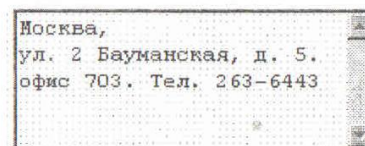
COLS=10 _____ ширина поля ввода в символах

WRAP=off | _____ не разбивать строки автоматически

virtual | _____ разбивать при вводе, но передавать как одну строку

physical _____ разбивать на строки и передавать в таком виде

Если вы хотите, чтобы в поле ввода по умолчанию выдавался какой-либо текст, то необходимо вставить его внутри тегов **<TEXTAREA>** и **</TEXTAREA>**.



анкета WEB-разработчика : - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Назад Вперед Остановить Обновить Домой Поиск Избранное Журнал

Адрес E:\@ MY work\Методичка\big form JS 2.html

Анкета WEB-разработчика

Регистрационное имя	user18
Пароль	<input type="password"/> <input type="password"/> подтвердите пароль
Ваша специализация	WEB-дизайнер
Пол	М Ж
Ваши навыки	<input checked="" type="checkbox"/> знание HTML и CSS <input type="checkbox"/> знание Perl <input type="checkbox"/> знание ASP <input checked="" type="checkbox"/> знание Adobe PhotoShop <input type="checkbox"/> знание JAVA <input checked="" type="checkbox"/> знание JavaScript <input checked="" type="checkbox"/> знание Flash
Дополнительные сведения о себе	Свободно владею английским языком

зарегистрировать очистить форму

Готово Мой компьютер

Результаты:

enctype="text/plain"

```
a=user18
b1=qwerty
b2=qwerty
c=c2
d=d1
f=Свободно владею
английским языком
```

enctype="application/x-www-form-urlencoded"

```
a=user18&b1=qwerty&b2=qwerty&c=c2&d=d1&f=
%D1%E2%EE%E1%EE%E4%ED%EE+%E2%EB%
E0%E4%E5%FE+%E0%ED%E3%EB%E8%E9%F1
%EA%E8%EC+%FF%E7%FB%EA%EE%EC
```