

Формы. Элементы форм

Формы располагают всеми необходимыми элементами управления для создания любого пользовательского интерфейса - это кнопки, текстовые поля, флажки, списки и пр.

Создание формы

Тег `<form>` устанавливает форму на веб-странице. Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

Формы создаются при помощи HTML элемента `<form></form>` следующим образом:

```
<body>
  <form name="имя_формы" action="url" method="GET|POST">
    <!-- элементы управления -->
  </form>
</body>
```

где

- `name` - имя формы (необязательно); служит для доступа к форме и ее элементам из сценариев
- `action` - адрес программы или сценария, которому передаются данные формы на обработку
- `method` - способ передачи данных на сервер

Примечание: при использовании форм исключительно на клиенте, атрибуты `action` и `method` не используются. Они необходимы для реализации взаимодействия "клиент-сервер".

Элементы управления

Элементами управления называются компоненты форм, необходимые пользователю для ввода или управления данными. Условно их можно разделить на 4 группы:

- Текстовые поля
- Флажки и переключатели
- Списки
- Кнопки

Текстовые поля

Начнем с текстовых полей, которые, в свою очередь, делятся на несколько типов:

- `text` - текстовое поле, в которое пользователь может ввести одну строку каких-либо данных
- `textarea` - многострочное текстовое поле
- `password` - поле для ввода пароля. Строка текста, введенная в такое поле, отображается в виде звездочек.
- `hidden` - скрытое поле. Используется для передачи на сервер вычисленных на клиенте данных, в формировании которых пользователь не принимал непосредственного участия (например, локальное время и подобные данные).

Все эти текстовые поля, за исключением textarea, создаются при помощи элемента `<input>`:

```
<input type="тип_элемента" name="имя_элемента" value="значение_по_умолчанию"
      size="количество_символов" />
```

где

- атрибут type - тип элемента может принимать значения text, password и hidden,
- name указывает имя элемента
- value задает значение элемента по умолчанию (текст, выводимый в элемент)
- size задает длину элемента в символах

Кроме перечисленных атрибутов, элемент поддерживает атрибуты style, class, id и другие общие для всех элементов HTML атрибуты.

Элемент управления textarea создается при помощи одноименного элемента :

```
<textarea name="имя_элемента" cols="количество_символов" rows="количество_строк">
  //<!-- текст элемента -->
</textarea>
```

Флажки и переключатели

Очень часто в интерактивных сценариях приходится давать пользователям возможность выбора одного или нескольких вариантов из множества. Формы предоставляют нам для этого флажки и переключатели.

Флажки - checkbox - позволяют сделать независимый выбор нескольких альтернатив из множества доступных.

Переключатели – radio – позволяют сделать зависимый выбор (одна альтернатива из множества).

Как и текстовые поля, эти элементы ввода данных реализуются при помощи элемента `<input />`. При этом, для создания флажка используется тип checkbox, а для создания переключателя - тип radio:

```
<!-- флажок -->
<input type="checkbox" name="имя_элемента" checked />
<input type="checkbox" name="имя_элемента" checked />
<input type="checkbox" name="имя_элемента" checked />
<!-- переключатель -->
<input type="radio" name="имя_переключателя" />
<input type="radio" name="имя_переключателя" checked />
<input type="radio" name="имя_переключателя" />
```

При создании флажка атрибут name задает имя флажка. Необязательный атрибут checked указывает - будет ли флажок по умолчанию выбран или нет.

С переключателями дело обстоит несколько иначе. Дело в том, что для создания переключателя необходимо несколько элементов `<input type="radio" />` с одинаковым значением атрибута name. Именно в этом случае выбор будет зависимым, т.е. вы не сможете выбрать одновременно более одного варианта. Таким образом, атрибут name дает имя не самим элементам, а целой группе зависимых флажков.

Для того, чтобы задать в группе выбранный по умолчанию вариант используется необязательный атрибут checked. Если данный атрибут не указан ни для одного из членов группы, не будет выбранных по умолчанию вариантов.

Списки

Списки так же можно использовать для реализации выбора одного или нескольких вариантов из множества. Список может быть раскрывающимся или многострочным. Для реализации этого элемента управления используются два типа элементов:

- `<select>` - служит для создания самого списка
- `<option>` - служит для создания элемента списка

Ниже приводится формат реализации списка:

```
<select name="имя_элемента" size="кол_строк" multiple>
  <option value="значение">текст элемента</option>
  <option value="значение" selected>текст элемента</option>
  <option value="значение">текст элемента</option>
</select>
```

где

- `name` - задает имя списка
- `size` - указывает количество одновременно отображаемых на экране строк списка. Если данный атрибут не указан или равен "1", список будет раскрывающимся, в противном случае - многострочным
- `multiple` - если данный атрибут указан для многострочного списка (`size > 1`), то пользователи будут иметь возможность выбирать сразу несколько элементов списка, удерживая `Ctrl` или `Shift`
- `value` - задает значение элемента списка (не текст, отображаемый на экране!)
- `selected` - задает элемент списка по умолчанию.

Кнопки

Последним "штрихом" к созданию форм будем считать добавление кнопок. Именно кнопки чаще всего используются программистами для добавления функциональности в форму. К кнопкам "привязываются" обработчики событий, выполняющие действия над данными форм.

Кнопки создаются при помощи того же элемента `<input>` и делятся на три типа:

- `button` - обыкновенная кнопка общего назначения
- `submit` - специальная кнопка при, щелчке на которой выполняется отправка данных формы на сервер сценарию, указанному атрибутом `action` текущей формы
- `button` - специальная кнопка, щелчке на которой выполняется "сброс" элементов формы в исходное состояние.

Рассмотрим пример создания кнопок (неважно какого типа):

```
<input type="тип" value="надпись" />
```

Для кнопок, в отличие от текстовых полей, атрибут `value` задает текст надписи на кнопке.