**Практическое занятие № 3**

**ТЕМА: Формы**

|  |  |  |
| --- | --- | --- |
| **Теоретический материал** | | |
| Электронная форма — эффективное средство, благодаря которому HTML-страница превращается из "пассивной", лишь предоставляющей информацию пользователю, в "активную", позволяющую принять информацию от пользователя и передать ее для обработки. Способ обработки и передачи данных определяется тегом <FORM>, внутри которого и заключается код формы.  Внутреннее содержание формы можно разделить на две части: активное и пассивное. Пассивными элементами формы являются все комментирующие и декоративные элементы, которые могут там содержаться. Это обычные составляющие HTML-страницы.  Активные элементы формы предназначены для ввода данных. Это строки и поля ввода, списки и кнопки. У каждого активного элемента формы — как и у всей формы — есть два основных параметра — name и value. Первый определяет имя элемента, по которому его можно отличить от других элементов формы, второй — значение, которое передается через этот элемент. Большинство активных элементов формы описывается тегом <INPUT>, а их вид определяется значением параметра type, который может принимать одно из значений:   * text соответствует строке ввода, * file — строке выбора файла, * password — строке ввода пароля; * значения submit, reset и button определяют кнопки различных видов * значения checkbox и radio — два типа списков: список вариантов и список-переключатель, соответственно. * другие значения для решения частных задач   Еще два вида элементов ввода, используемых в формах, создаются с помощью  следующих тегов. Тег <TEXTAREA> позволяет создавать поля ввода — прямоугольные окна с собственными средствами прокрутки, в которые можно вводить произвольный текст (без форматирования). Тег <TEXTAREA> — парный.  Внутри него помещается текст, который содержится в поле ввода по умолчанию.  Наконец, еще один распространенный элемент электронных форм — раскрывающийся список — создается с помощью конструкции HTML, образуемой тегами <SELECT> и <OPTION>. Первый является парным и заключает в себе весь  список, вторые предназначены для создания отдельных пунктов. Списки, созданные таким образом, могут состоять из любого количества строк (если список состоит из одной строки, то он является "раскрывающимся"), а также, в зависимости от параметра multiple, позволяют выбрать один или несколько элементов.  Данные, вводимые посредством формы, обрабатываются не средствами HTML. Они могут передаваться по электронной почте или непосредственно программе-  обработчику. Язык, на котором может быть написана такая программа, значения  не имеет. В частности, для обработки таких данных могут использоваться сценарии на языке JavaScript.  **Текстовые строки**  Для ввода однострочных полей используется дескриптор <INPUT>. В зависимости от значения параметра *type* этот дескриптор может "принимать вид" самых разных элементов формы. В частности, когда этот параметр имеет значение *text*, тег <INPUT> "превращается" в текстовую строку.  Для того чтобы ограничить также вводимую строку, используется параметр *maxlength*. Его значением служит количество символов, принимаемых формой и передаваемых для обработки. Если присвоить этому параметру значение 2, то пользователь просто не сможет ввести третий символ.  Довольно часто встречаются ситуации, когда вводимая строка не должна быть видна на экране. Типичный случай — пароль. Для того чтобы обеспечить конфиденциальность такого рода, используются текстовые строки специального вида - параметру *type* присваивается значение *password*  *Работа с текстовыми строками форм продемонстрирована в примере 3.1*  **Кнопки**  Кнопки — главный элемент любой электронной формы. Кнопки создаются с помощью того же тега <INPUT>, что и текстовые строки. Однако значение параметра type в этом случае другое — в зависимости от назначения кнопки. Чаще всего — практически всегда — в формах встречается кнопка для передачи данных программе-обработчику. Надписи на ней бывают разные — "Принять", "Отправить", "ОК", "Поехали!".  Для того чтобы создать такую кнопку, параметру *type* присваивается значение *submit*. Для задания надписи на кнопке используется параметр *value*.  Стандартная кнопка может быть изменена на любую картинку. Для этого параметру type присваивается значение *image*. Далее в теге <input> настраиваются те же параметры, что и в теге <img>  Для создания кнопки возврата параметру *type* присваивается значение *reset*. В результате получаем точно такую же кнопку, что и в случае *submit*, только результат ее действия другой, вместо того чтобы отправить данные на обработку, данные просто удаляются, и форма приводится к исходному состоянию.  Элемент <input> типа button создает произвольную кнопку. Для таких кнопок действие по умолчанию не определено, а браузеры пользователей должны использовать в качестве надписи на кнопке значение атрибута value. С атрибутами событий каждой такой кнопки (щелчок мышью на кнопке или дрyrое событие) могyт быть связаны прогpаммы обработки этих событий, которые могyт выполнять определенные действия. Эти программы пишутся, например, на языке JavaScript.  *Работа с кнопками продемонстрирована в примере 3.2*  **Метки**  Связь между элементом формы и поясняющим текстом может быть установлена с помощью парного тега <label>, внутри которого помещается элемент <input>.  *Работа с метками продемонстрирована в примере 3.3*  **Текстовое поле**  Не всегда текст, который нужно ввести, помещается в одной строке. Поэтому для крупных блоков текста предусмотрен элемент формы <textarea>, в который можно поместить любой текст. При этом данный элемент сохраняет переносы между строками. Основные параметры данного элемента – это ширина поля в символах *cols* и высота поля в строках *rows*.  *Работа с текстовым полем продемонстрирована в примере 3.3*  **Списки вариантов (флажки, «галочки»)**  Флажки используют, когда необходимо выбрать любое количество вариантов из предложенного списка. Флажок создается следующим образом:  <input type=”checkbox”>  Если при первоначальном отображении формы флажок должен быть выделен, то задается атрибут *checked*. Принадлежность флажка к определенному списку задается с помощью атрибута *name*  *Работа с флажками продемонстрирована в примере 3.3*  **Радиокнопки**  Радиокнопки используют, когда необходимо выбрать один вариантов из предложенного списка. Радиокнопка создается следующим образом:  <input type=”radio”>  Если при первоначальном отображении формы радиокнопка должна быть выделена, то задается атрибут checked. Принадлежность к определенному списку задается с помощью атрибута *name*  *Работа с радиокнопками продемонстрирована в примере 3.3*  **Группирование элементов формы**  Элемент <fieldset> предназначен для группировки элементов формы. Такая группировка облегчает работу с формами, содержащими большое число данных. Для создания заголовка группы элементов формы применяется парный тег <legend>  *Работа с группированием элементов формы продемонстрирована в примере 3.3*  **Значение атрибута type: date**  Для запроса ввода даты вы можете использовать элемент <input> со значением date атрибута type:  <input type="date">    **Значение атрибута type: week**  При использовании типа поля ввода week соответствующий инструмент-указатель позволит пользователю выбрать одну неделю в году, после чего  обеспечит ввод данных в формате, например, Неделя 25, 2017.  <input type="week" name="week">    **Значение атрибута type: month**  Для запроса ввода определенного месяца года вы можете использовать элемент <input> со значением month атрибута type. Интерфейс позволит пользователю выбрать один месяц, после чего обеспечит ввод данных в виде года и месяца.  <input type="month" name="month">  **Значение атрибута type: time**  Поле ввода, относящееся к типу time, допускает ввод значений в 24-часовом формате, например 17:30.  <input type="time" name="time">    **Значение атрибута type: number**  Элемент <input> типа number создает поле, в которое пользователь может вводить только числовое значение. Для типа ввода number браузер предоставляет виджет счетчика, который представляет собой поле, справа от которого находятся две кнопки со стрелками — для увеличения и уменьшения числового значения. В поле счетчика по умолчанию разрешен прямой ввод с клавиатуры. Для указания минимальных и максимальных допустимых значений ввода предназначены атрибуты min и max, а также можно установить шаг приращения с помощью атрибута step.  <input type="number" name="num" min="1" max="12" value="1" step="1">  **Значение атрибута type: range**  Поле ввода типа *range* элемента ***<input>*** позволит создать такой элемент интерфейса, как ползунковый регулятор. Ползунок предназначен только для выбора числовых значений в некоем диапазоне, при этом для пользователя не все браузеры отображают текущее числовое значение. Основной синтаксис создания ползунка:  <input type="range" min="0" max="100" step="1" value="50">  Нижняя и верхняя границы диапазона min и max ограничивают значения, которые могут храниться в поле формы. Диапазон по умолчанию — от 0 до 100. Атрибут **step** позволяет разработчикам указывать шаг изменения чисел (по умолчанию 1). Текущее значение задается в атрибуте **value**. По умолчанию **value** = (max + min)/2. Вышеперечисленные атрибуты не являются обязательными и, если их опустить, то в таком случае они принимают значения по умолчанию.    **Значение атрибута type: color**  Поле ввода **type="color"** генерирует палитры цветов обеспечивая пользователям возможность выбирать значения цветов в шестнадцатеричном формате RGB (#RRGGBB). Синтаксис создания поля для задания цвета:  <input type="color" value="цвет" name="имя">  Атрибут **value** предназначен для задания исходного цвета (#RRGGBB) и не является обязательным. Атрибут **name** применяется для идентификации получаемого значения.    **Значение атрибута type: email**  Поле типа *email* представляет из себя однострочное текстовое поле и предназначено для ввода либо отдельного адреса, либо списка адресов электронной почты. Браузеры, поддерживающие язык HTML5, проверят, соответствует ли введенный посетителем адрес электронной почты принятому стандарту для данного типа адресов. Синтаксис создания поля следующий:  <input type="email">  Элемент **<input type="email">** может включать атрибуты свойственные типу *text*, а также добавлен атрибут **multiple**, который позволяет вводить сразу список из допустимых электронных адресов, разделенных запятыми.  **Значение атрибута type: url**  Элемент ***<input>*** типа *url* адаптирован для ввода URL-адресов, например адреса какой-либо страницы во всемирной паутине. Строка **<input type="url">** заставляет браузер проверять, правильно ли пользователь ввел URL-адрес. При использовании поля ввода type="url" на устройствах с сенсорными экранами, внешний вид встроенной виртуальной клавиатуры будет оптимизирован для отображения символов, наиболее часто встречающихся в URL-адpecax. Атрибуты для элемента ***<input>*** типа *url* совпадают с текстовым полем (<input type="text">). Некоторые браузеры добавляют специфическую информацию в предупреждающие сообщения, выводимые на экран, при попытке отправить форму с некорректными значениями URL-адреса. Далее приведен пример кода, включающего атрибут **placeholder** (с англ. — заполнитель), значение которого в виде подсказки будет по умолчанию отображаться, пока поле ввода URL-адреса не получит фокус:  <input id="web" name="web" type="url" placeholder="www.yoursite.com">    **Значение атрибута type: tel**  Элемент <input> типа tel применяется для того, чтобы сообщить браузеру, что в соответствующем поле формы пользователь должен ввести телефонный номер. Несмотря на то, что телефонный номер представляет из себя числовой формат вводимых данных, в браузерах поле типа tel ведет себя как обычное текстовое поле ввода. Однако, применение типа поля ввода tel приводит к появлению на экранах мобильных устройств специальной клавиатуры, предназначенной для облегчения ввода информации.  *Использование различных значений атрибута type тега input продемонстрировано в примере 3.4*  **Раскрывающийся список (select)**  Элемент ***<select>*** создает на веб-странице раскрывающийся список (также называемый раскрывающимся или выпадающим меню), позволяющий выбрать одно значение из множества возможных. Элемент ***<select>*** работает в паре с элементом ***<option>***, создающим меню. Элемент ***<option>*** задается для каждого пункта меню. Текст, расположенный между тегами ***<option>*** и ***</option>***, будет выведен в окне браузера как пункт раскрывающегося списка. Атрибут **value** используется в каждом элементе ***<option>*** для установки значения, которое будет отправлено на сервер вместе с именем выбранного элемента. Указав для элемента ***<option>*** атрибут **selected**, вы сделаете его значением по умолчанию для данного списка. При загрузке страницы такие элементы будут выделены. Если атрибут **selected** не установлен, то при загрузке страницы будет отображен первый вариант ответа из созданного списка. Если пользователь не выберет другой пункт списка, то на сервер будет отправлен пункт, отмеченный атрибутом **selected** по умолчанию:  **Прокручиваемые списки**  По умолчанию с помощью тега ***<select>*** создается раскрывающийся список, в котором изначально видно только одно значение. Для создания прокручиваемого списка укажите число видимых строк, введя соответствующее значение атрибута **size**. Если количество элементов списка превышает значение, указанное в атрибуте **size**, справа появляется полоса прокрутки. Несмотря на то, что атрибут **size** позволяет отобразить сразу несколько пунктов списка, посетитель может выбрать из списка по-прежнему только один пункт. Чтобы посетитель мог выбрать из списка одновременно несколько элементов (с помощью клавиши *Shift* или *Ctrl*), нужно использовать атрибут **multiple**. Как и в случае с раскрывающимся списком, указав в элементе ***<option>*** атрибут **selected**, вы сделаете его значением по умолчанию. Если для тега ***<select>*** или для ***<option>*** установить атрибут **disabled**, то в первом случае становится отключенным весь список, а во втором — только отдельный пункт списка.    **Группировка пунктов списка (optgroup)**  Если в списке находится большое количество пунктов, то посетителю довольно сложно найти нужный пункт. В этом случае можно воспользоваться элементом ***<optgroup>***, предназначенным для создания смысловых групп пунктов списка. Элементы ***<option>*** помещаются внутрь контейнера ***<optgroup>***, а заголовок группы указывается в атрибуте **label**.    *Работа со списком продемонстрирована в примере 3.5*  **Общие атрибуты элементов форм**  Атрибут **name** задает уникальное имя поля. Введенная пользователем информация передается обработчику в качестве значения переменной с именем поля. Атрибут **name** имеет значение для программистов, поэтому имя должно быть логичным  <input type="text" **name**="name">  Атрибут **value** позволяет присвоить элементу управления значение по умолчанию. Введя в поле значение по умолчанию, можно пояснить пользователю, какие данные и в каком формате вы хотите здесь видеть. Для флажков и радиокнопок определяет значение, которое будет передано на сервер, если кнопка выбрана. Оно будет состоять из имени кнопки и ее значения.  Ваше имя:<br>  <input type="text" name="firstname" **value="Иван"**><br>    Атрибут **disabled** весьма полезен для ограничения доступа к некоторым полям формы на основании ранее введенных данных. Чтобы запретить изменение данных, в любое поле можно добавить атрибут **disabled**. При этом поле становится неактивным, т.е. щелчки на нем не приводят ни к каким действиям.  Атрибут **readonly** не позволяет пользователю изменять значения элементов формы, но, в отличие от атрибута **disabled**, такие поля их можно выделять. Это позволяет разработчикам устанавливать значения элементов формы в зависимости от ранее введенных данных с помощью сценариев.  Атрибут **placeholder** позволяет ввести в поле формы замещающий текст. Замещающий текст отображается внутри поля ввода до тех пор, пока поле не имеет фокуса ввода. Как только пользователь сфокусируется на поле и начнет ввод текста, замещающий текст исчезает.  Атрибут **required** указывает, что поле элемента формы обязательно к заполнению пользователем. Когда пользователь отправляет форму, браузер проверяет, не оставлены ли пустыми все обязательные поля. | | |
| **Пример 3.1 Демонстрация текстовых строк** | | |
|  | ***Задача:*** | |
|  | Создать форму, содержащую текстовую строку неограниченного размера, текстовую строку длиной не более 5 символов, а также поле для ввода пароля |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Пример 3.2. Демонстрация кнопок форм** | | |
|  | ***Задача:*** | |
|  | Дополнить пример 3.1 кнопками «Отправить» и «Очистить форму» |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Пример 3.3.** | | |
|  | ***Задача:*** | |
|  | Продемонстрировать работу с различными элементами форм (checkbox, radio, textarea, label и др.). |
| ***Решение:*** | |
|  |  |
| ***Ответ:*** | |
|  |  |
| **Пример 3.4. Демонстрация атрибутов тега input** | | |
|  | ***Задача:*** | |
|  | Создать форму, в которой отобразить элементы: выбор дня недели, недели, времени, ползунок, выбор цвета, поле для числового значения |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Пример 3.5. Списки** | | |
|  | ***Задача:*** | |
|  | Создать список с прокруткой и группировкой пунктов |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Задание 3.1** | | |
|  | ***Задача:*** | |
|  | Повторите форму по данному образцу: |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Задание 3.2** | | |
|  | ***Задача:*** | |
|  | Повторите форму по данному образцу: |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |
| **Задание 3.3** | | |
|  | ***Задача:*** | |
|  | Создайте страницу по образцу: |
| ***Решение:*** | |
|  | Файл index.html |
| ***Ответ:*** | |
|  |  |