

Лабораторная работа № 3

Тема работы: «Создание форм на веб-странице»

1. Цель работы

Формирование умений создания форм на веб-странице, добавления различных элементов формы, использования паттернов ввода.

2. Задание

Создать форму на веб-странице в соответствии с порядком выполнения работы.

3. Оснащение работы

ПК, редактор исходного кода, текстовый редактор, браузер.

4. Основные теоретические сведения

Форма создаётся с помощью тега `<form>`, внутри которой могут быть любые необходимые теги. Элемент `<form>...</form>` не предусматривает ввод данных, так как является контейнером, удерживая вместе все элементы управления формы – поля. Атрибуты этого элемента содержат информацию, общую для всех полей формы, поэтому в одну форму нужно включать поля, объединенные логически.

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.

Для отправки формы на сервер используется кнопка Submit, того же можно добиться, если нажать клавишу Enter в пределах формы. Если кнопка Submit отсутствует в форме, клавиша Enter имитирует ее использование. Атрибуты тега `<form>` приведены в таблице 3.1.

Таблица 3.1 – Атрибуты тега `<form>`

Атрибут	Значение / описание
accept-charset	Значение атрибута представляет собой разделенный пробелами список кодировок символов , которые будут использоваться для отправки формы, например, <code><form accept-charset="ISO-8859-1"></code> .
action	Обязательный атрибут , который указывает url обработчика формы на сервере, которому передаются данные. Представляет из себя файл (например, action.php), в котором описано, что нужно делать с данными формы. Если значение атрибута не будет указано, то после перезагрузки страницы элементы формы примут значения по умолчанию. В случае, если вся работа будет выполняться на стороне клиента сценариями JavaScript, то для атрибута action можно указать значение # .

Атрибут	Значение / описание
	<p>Также можно сделать так, чтобы заполненная посетителем форма приходила вам на почту. Для этого нужно внести следующую запись:</p> <pre data-bbox="529 344 1417 427"><form action="mailto:адрес вашей электронной почты" enctype="text/plain"></form></pre>
autocomplete	<p>Отвечает за запоминание введенных в текстовое поле значений и автоподстановку их при последующем вводе:</p> <p>on — означает, что поле не защищено, и его значение можно сохранять и извлекать,</p> <p>off — отключает автозаполнение для полей форм.</p>
enctype	<p>Используется для указания MIME-типа данных, отправляемых вместе с формой, например, <code>enctype="multipart/form-data"</code>. Указывается только в случае <code>method="post"</code>.</p> <p>application/x-www-form-urlencoded — тип содержимого по умолчанию, указывает на то, что передаваемые данные представляют список URL-кодированных переменных формы. Символы пробела (ASCII 32) будут закодированы как <code>+</code>, а специальный символ, например, такой как <code>!</code> будет закодирован шестнадцатиричной форме как <code>%21</code>.</p> <p>multipart/form-data — используется для отправки форм, содержащих файлы, не-ASCII данные и бинарные данные, состоит из нескольких частей, каждая из которых представляет содержимое отдельного элемента формы.</p> <p>text/plain — указывает на то, что передается обычный (не html) текст.</p>
method	<p>Задаёт способ передачи данных формы.</p> <p>Метод get передает данные на сервер через адресную строку браузера. При формировании запроса к серверу все переменные и их значения формируют последовательность вида <code>www.anysite.ru/form.php?var1=1&var2=2</code>. Имена и значения переменных присоединяются к адресу сервера после знака <code>?</code> и разделяются между собой знаком <code>&</code>. Все специальные символы и буквы, отличные от латинских, кодируются в формате <code>%nn</code>, пробел заменяется на <code>+</code>. Этот метод нужно использовать, если вы не передаете больших объемов информации. Если вместе с формой предполагается отправка какого-либо файла, этот метод не подойдет.</p> <p>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.</p> <pre data-bbox="529 1980 1369 2063"><form action="action.php" enctype="multipart/form-data" method="post"></form></pre>

Атрибут	Значение / описание
<code>name</code>	Задаёт имя формы , которое будет использоваться для доступа к элементам формы через сценарии, например, <code>name="opros"</code> .
<code>novalidate</code>	Отключает проверку в кнопке для отправки формы. Атрибут используется без указания значения
<code>target</code>	Указывает окно, в которое будет направлена информация: <code>_blank</code> — новое окно; <code>_self</code> — тот же фрейм; <code>_parent</code> — родительский фрейм (если он существует, если нет — то в текущий); <code>_top</code> — окно верхнего уровня по отношению к данному фрейму. Если вызов происходит не из дочернего фрейма, то в тот же фрейм.

В HTML5 для работы с формами предусмотрены следующие элементы, таблица 3.2.

Таблица 3.2 – Элементы HTML5 для работы с формами

Формы		
<code><form></form></code>	html-форма	block
<code><input></code>	многофункциональные поля формы	inline-block
<code><textarea></code>	многострочное поле формы	inline-block
<code><label></label></code>	текстовая метка для элемента <code><input></code>	inline
<code><datalist></datalist></code>	контейнер для выпадающего списка элемента <code><input></code> с <code><option></code> -значениями	none
<code><select></select></code>	элемент управления с выбором значений из предложенных вариантов <code><option></code>	inline-block
<code><optgroup></optgroup></code>	контейнер с заголовком для группы элементов <code><option></code>	block
<code><option></option></code>	вариант (опция) в раскрывающемся списке	block
<code><fieldset></fieldset></code>	группирует связанные элементы в форме	block
<code><legend></legend></code>	заголовок элементов формы, сгруппированных с помощью <code><fieldset></code>	block
<code><button></button></code>	интерактивная кнопка	inline-block
<code><keygen></code>	генерирует закрытый и открытый ключи	inline-block
<code><progress></progress></code>	индикатор выполнения задачи	inline-block
<code><meter></meter></code>	индикатор измерения в заданном диапазоне	inline-block
<code><output></output></code>	поле для вывода результата вычисления	inline

На рисунке 3.1 показано назначение различных элементов при создании форм в веб-документе.

Анкета посетителя ресторана

Контактная информация

Имя

Телефон

Email

Дата посещения

Персональная информация

Возраст

Любимая кухня Русская

Какие блюда Вы хотели бы увидеть в меню?

Оценка нашего заведения

Почему Вы выбрали наше заведение?

☐ Недалеко от дома/работы

☐ Увидел рекламу

☐ Посоветовали

☐ Оптимальное соотношение цены и качества

Вы будете рекомендовать наше заведение своим знакомым?

☐ Да

☐ Нет

Рисунок 3.1 – Назначение элементы HTML5 для работы с формами

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

Каждой группе элементов можно присвоить название с помощью элемента `<legend>`, который идет сразу за тегом `<fieldset>`. Название группы проявляется слева в верхней границе `<fieldset>`. Например, если в элементе `<fieldset>` хранится контактная информация:

```
<form id="auth" action="handler.php" method="post">
  <fieldset>
    <legend>Контактная информация</legend>
    <p><label for="name">Имя <em>*</em></label><input type="text"
id="name"></p>
    <p><label for="email">E-mail</label><input type="email" id="email"></p>
  </fieldset>
  <p><input type="submit" value="Отправить"></p>
</form>
```

Атрибуты тега `<fieldset>` приведены в таблице 3.3.

Таблица 3.3 – Атрибуты тега `<fieldset>`

Атрибут	Значение / описание
---------	---------------------

disabled	Если атрибут присутствует, то группа связанных элементов формы, находящихся внутри контейнера <code><fieldset></code> , отключены для заполнения и редактирования. Используется для ограничения доступа к некоторым полям формы, содержащих ранее введенные данные. Атрибут используется без указания значения — <code><fieldset disabled></code> .
form	Значение атрибута должно быть равно атрибуту <code>id</code> элемента <code><form></code> в этом же документе. Указывает на одну или несколько форм, к которым принадлежит данная группа элементов. На данный момент атрибут не поддерживается ни одним браузером.
name	Определяет имя, которое будет использоваться для ссылки на элементы в JavaScript, или для ссылки на данные формы после заполнения и отправки формы. Является аналогом атрибута <code>id</code> .

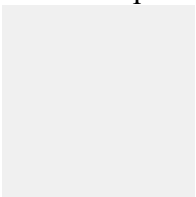
Элемент `<input>` создает большинство полей формы. Атрибуты элемента отличаются в зависимости от типа поля, для создания которого используется этот элемент. Атрибуты тега `<input>` представлены в таблице 3.4.

Таблица 3.4. – Атрибуты тега `<input>`

Атрибут	Значение / описание
accept	Определяет тип файла, разрешенных для отправки на сервер. Указывается только для <code><input type="file"></code> . Возможные значения: <code>file_extension</code> — разрешает загрузку файлов с указанным расширением, например, <code>accept=".gif"</code> , <code>accept=".pdf"</code> , <code>accept=".doc"</code> <code>audio/*</code> — разрешает загрузку аудиофайлов <code>video/*</code> — разрешает загрузку видеофайлов <code>image/*</code> — разрешает загрузку изображений <code>media_type</code> — указывает на медиа-тип загружаемых файлов.
Alt	Определяет альтернативный текст для изображений, указывается только для <code><input type="image"></code> .
autocomplete	Отвечает за запоминание введенных в текстовое поле значений и автоподстановку их при последующем вводе: <code>on</code> — означает, что поле не защищено, и его значение можно сохранять и извлекать, <code>off</code> — отключает автозаполнение для полей форм.
autofocus	Позволяет сделать так, чтобы в загружаемой форме то или иное поле ввода уже имело фокус (было выбрано), являясь готовым к вводу значения.
checked	Атрибут проверяет, установлен ли флажок по умолчанию при загрузке страницы для полей типа <code>type="checkbox"</code> и <code>type="radio"</code> .
disabled	Отключает возможность редактирования и копирования содержимого поля.

Атрибут	Значение / описание
<code>form</code>	Значение атрибута должно быть равно атрибуту <code>id</code> элемента <code><form></code> в этом же документе. Определяет одну или несколько форм, которым принадлежит данное поле формы.
<code>formaction</code>	Задаёт <code>url</code> файла, который будет обрабатывать введенные в поля данные при отправке формы. Задается только для полей типа <code>type="submit"</code> и <code>type="image"</code> . Атрибут переопределяет значение атрибута <code>action</code> самой формы.
<code>formenctype</code>	Определяет, как будут кодироваться данные полей формы при отправке на сервер. Переопределяет значение атрибута <code>enctype</code> формы. Задается только для полей типа <code>type="submit"</code> и <code>type="image"</code> . Варианты: <code>application/x-www-form-urlencoded</code> — значение по умолчанию. Все символы кодируются перед отправкой (пробелы заменяются на символ <code>+</code> , специальные символы преобразуются в значения ASCII HEX) <code>multipart/form-data</code> — символы не кодируются <code>text/plain</code> — пробелы заменяются на символ <code>+</code> , а специальные символы не кодируются.
<code>formmethod</code>	Атрибут определяет метод, который браузер будет использовать для отправки данных формы на сервер. Задается только для полей типа <code>type="submit"</code> и <code>type="image"</code> . Переопределяет значение атрибута <code>method</code> формы. Варианты: <code>get</code> — значение по умолчанию. Данные из формы (пара имя/значение) добавляются в <code>url</code> -адрес и отправляются на сервер: <code>URL?имя=значение&имя=значение</code> <code>post</code> — данные формы отправляются в виде <code>http</code> -запроса.
<code>formnovalidate</code>	Определяет, что данные полей формы не должны проверяться при отправке формы. Переопределяет значение атрибута <code>novalidate</code> формы. Можно использовать без указания значения атрибута.
<code>formtarget</code>	Определяет, где выводить ответ, полученный после отправки формы. Задается только для полей типа <code>type="submit"</code> и <code>type="image"</code> . Переопределяет значение атрибута <code>target</code> формы. <code>_blank</code> — загружает ответ в новое окно/вкладку <code>_self</code> — загружает ответ в то же окно (значение по умолчанию) <code>_parent</code> — загружает ответ в родительский фрейм <code>_top</code> — загружает ответ во весь экран <code>framename</code> — загружает ответ во фрейм с указанным именем.
<code>height</code>	Значение атрибута содержит количество пикселей без указания единицы измерения. Устанавливает высоту поля формы типа <code>type="image"</code> , например, <code><input type="image"</code>

Атрибут	Значение / описание
	<code>src="img_submit.gif" height="50"></code> . Рекомендуется одновременно устанавливать как высоту, так и ширину поля.
<code>list</code>	Является ссылкой на элемент <code><datalist></code> , содержит его <code>id</code> . Позволяет предоставить пользователю несколько вариантов на выбор, когда он начинает вводить значение в соответствующем поле.
<code>max</code>	Позволяет ограничить допустимый ввод числовых данных максимальным значением, значение атрибута может содержать целое или дробное число. Рекомендуется использовать этот атрибут вместе с атрибутом <code>min</code> . Работает со следующими типами полей: <code>number</code> , <code>range</code> , <code>date</code> , <code>datetime</code> , <code>datetime-local</code> , <code>month</code> , <code>time</code> и <code>week</code> .
<code>maxlength</code>	Атрибут задает максимальное количество символов, вводимых в поле. Значение по умолчанию 524288 символов.
<code>min</code>	Позволяет ограничить допустимый ввод числовых данных минимальным значением.
<code>multiple</code>	Позволяет пользователю ввести несколько значений атрибутов, разделяя их запятой. Применяется в отношении файлов и адресов электронной почты. Указывается без значения атрибута.
<code>name</code>	Определяет имя, которое будет использоваться для доступа к элементу <code><form></code> , к примеру, в таблицах стилей <code>css</code> . Является аналогом атрибута <code>id</code> .
<code>pattern</code>	Позволяет определять с помощью регулярного выражения синтаксис данных, ввод которых должен быть разрешен в определенном поле. Например, <code>pattern="[a-z]{3}-[0-9]{3}"</code> — квадратные скобки устанавливают диапазон допустимых символов, в данном случае — любые строчные буквы, число в фигурных скобках указывает, что нужны три строчные буквы, после которых следует тире, далее — три цифры в диапазоне от 0 до 9.
<code>placeholder</code>	Содержит текст, который отображается в поле ввода до заполнения (чаще всего это подсказка).
<code>readonly</code>	Не позволяет пользователю изменять значения элементов формы, выделение и копирование текста при этом доступно. Указывается без значения атрибута.
<code>required</code>	Выводит сообщение о том, что данное поле является обязательным для заполнения. Если пользователь попытается отправить форму, не введя в это поле требуемое значение, то на экране отобразится предупреждающее сообщение. Указывается без значения атрибута.
<code>size</code>	Задаёт видимую ширину поля в символах. Значение по умолчанию — 20. Работает со следующими типами по-

Атрибут	Значение / описание
	лей: <code>text</code> , <code>search</code> , <code>tel</code> , <code>url</code> , <code>email</code> и <code>password</code> .
<code>src</code>	Задаёт <code>url</code> изображения, используемого в качестве кнопки отправки данных формы. Указывается только для поля <code><input type="image"></code> .
<code>step</code>	Используется для элементов, предполагающих ввод числовых значений, указывает величину увеличения или уменьшения значений в процессе регулировки диапазона (шаг).
<code>type</code>	<code>button</code> — создаёт кнопку.
	<code>checkbox</code> — превращает поле ввода во флажок, который можно установить или очистить, например, <input type="checkbox"/> У меня есть автомобиль
	<code>color</code> — генерирует палитры цветов в поддерживающих браузерах, давая пользователям возможность выбирать значения цветов в шестнадцатеричном формате.
	<code>date</code> — позволяет вводить дату в формате дд.мм.гггг. День рождения:
	<code>datetime-local</code> — позволяет вводить дату и время, разделённые прописной английской буквой <code>T</code> по шаблону дд.мм.гггг чч:мм. День рождения — день и время:
	<code>email</code> — браузеры, поддерживающие данный атрибут, будут ожидать, что пользователь введёт данные, соответствующие синтаксису адресов электронной почты. E-mail:
	<code>file</code> — позволяет загружать файлы с компьютера пользователя. Выберите файл:
	<code>hidden</code> — скрывает элемент управления, который не отображается браузером и не даёт пользователю изменять значения по умолчанию.
	<code>image</code> — создаёт кнопку, позволяя вместо текста на кнопке вставить изображение. 
	<code>month</code> — позволяет пользователю вводить год и номер месяца по шаблону гггг-мм.

Атрибут	Значение / описание
	<p>number — предназначено для ввода целочисленных значений. Его атрибуты min, max и step задают верхний, нижний пределы и шаг между значениями соответственно. Эти атрибуты предполагаются у всех элементов, имеющих численные показатели. Их значения по умолчанию зависят от типа элемента. Укажите количество (от 1 до 5):</p>
	<p>password — создает текстовые поля в форме, при этом вводимые пользователем символы заменяются на звездочки, маркеры, либо другие, установленные браузером значки. Введите пароль: <input type="password"/></p>
	<p>radio — создает переключатель — элемент управления в виде небольшого кружка, который можно включить или выключить. Вегетарианец: <input type="radio"/></p>
	<p>range — позволит создать такой элемент интерфейса, как ползунок, min / max — позволят установить диапазон выбора</p>
	<p>reset — создает кнопку, которая очищает поля формы от введенных пользователем данных. <input type="reset"/></p>
	<p>search — обозначает поле поиска, по умолчанию поле ввода имеет прямоугольную форму. Поиск: <input type="search"/></p>
	<p>submit — создает стандартную кнопку, активируемую щелчком мыши. Кнопка собирает информацию с формы и отправляет ее для обработки. <input type="submit"/></p>
	<p>text — создает текстовые поля в форме, выводя однострочное текстовое поле для ввода текста. <input type="text"/></p>
	<p>time — позволяет вводить время в 24-часовом формате по шаблону чч:мм. В поддерживающих браузерах оно отображается как элемент управления в виде числового поля ввода со значением, изменяемым с помощью мыши, и допускает ввод только значений времени. Укажите время: <input type="time"/></p>
	<p>url — поле предназначено для указания URL-адресов. Главная страница: <input type="url"/></p>
	<p>week — соответствующий инструмент-указатель позволяет пользователю выбрать одну неделю в году, после чего обеспечит ввод</p>

Атрибут	Значение / описание
	данных в формате нн-гггг. В зависимости от года число недель может быть 52 или 53. Укажите неделю:
value	Определяет текст, отображаемый на кнопке, в поле или связанный текст. Не указывается для полей типа file.
width	Значение атрибута содержит количество пикселей. Позволяет задать ширину полей формы.

```

<input maxlength="25" size="40" value="Вася">
<input type="password" maxlength="25" size="40" name="password">
<input type="hidden" name="name" value="Vasya">

```

Элемент `<textarea>...</textarea>` используется вместо элемента `<input type="text">`, когда нужно создать большие текстовые поля. Текст, отображаемый как исходное значение, помещается внутрь тега. Размеры поля устанавливаются при помощи атрибутов `cols` – размеры по горизонтали, `rows` – размеры по вертикали. Высоту поля можно задать свойством `height`. Все размеры считаются исходя из размера одного символа моноширинного шрифта. Атрибуты тега `</textarea>` приведены в таблице 3.5.

Таблица 3.5 – Атрибуты тега `</textarea>`

Атрибут	Значение / описание
autofocus	Устанавливает фокус на нужном начальном текстовом поле автоматически.
cols	Устанавливает ширину через количество символов. Если пользователь вводит больше текста, появляется полоса прокрутки.
disabled	Отключает возможность редактирования и копирования содержимого поля.
form	Значение атрибута должно быть равно значению атрибута <code>id</code> элемента <code><form></code> в этом же документе. Определяет одну или несколько форм, которым принадлежит данное текстовое поле.
maxlength	Значение атрибута задает максимальное число символов для ввода в поле.
name	Задаёт имя текстового поля.
placeholder	Определяет короткую текстовую подсказку, которая описывает ожидаемое вводимое значение.
readonly	Отключает возможность редактирования содержимого поля.
required	Выводит сообщение о том, что данное поле является обязательным для заполнения.

<code>rows</code>	Указывает число, которое означает, сколько строк должно отображаться в текстовой области.
<code>wrap</code>	Определяет, должен ли текст сохранять переносы строк при отправке формы. Значение <code>hard</code> сохраняет перенос, а значение <code>soft</code> не сохраняет. Если используется значение <code>hard</code> , то должно указываться значение атрибута <code>cols</code> .

```

<form action="handler.php">
  <p><b>Введите ваш отзыв:</b></p>
  <p><textarea name="comment"></textarea></p>
  <p><input type="submit"></p>
</form>

```

Раскрывающиеся списки создаются при помощи элемента `<select>...</select>`. Они позволяют выбрать одно или несколько значений из предложенного множества. По умолчанию в поле списка отображается его первый элемент.

Для добавления в список пунктов используются элементы `<option>...</option>`, которые располагаются внутри `<select>`.

Для систематизации списков применяется элемент `<optgroup>...</optgroup>`, который создает заголовки в списках. Атрибуты тега `<select>` приведены в таблице 3.6.

Таблица 3.6 – Атрибуты тега `<select>`

Атрибут	Значение / описание
<code>autofocus</code>	Устанавливает автоматический фокус на элементе при загрузке страницы.
<code>disabled</code>	Отключает раскрывающийся список.
<code>form</code>	Определяет форму, которой принадлежит данный список. В качестве значения атрибута указывается идентификатор формы.
<code>multiple</code>	Дает возможность выбора одного или нескольких пунктов, для этого при выборе нужно нажать и удерживать нажатой клавишу <code>Ctrl</code> .
<code>name</code>	Определяет имя для выпадающего списка. Значение атрибута содержит название, отражающее тематику списка.
<code>required</code>	Выводит сообщение о том, что пользователь должен выбрать значение из раскрывающегося списка перед отправкой формы.
<code>size</code>	Задаёт количество одновременно видимых на экране элементов списка. Если количество элементов списка превышает установленное количество, появляется полоса прокрутки. Значение атрибута задается

Атрибуты тега `<option>` приведены в таблице 3.7.

Таблица 3.7 – Атрибуты тега <option>

Атрибут	Значение / описание
disabled	Делает недоступным для выбора элемент списка.
Label	Задаёт укороченную версию для элемента, которая будет отражаться в выпадающем списке. Значение атрибута содержит текст, описывающий соответствующий пункт выпадающего списка.
selected	Отображает выбранный элемент списка по умолчанию при загрузке страницы браузером.
Value	Указывает значение, которое будет отправлено на сервер при отправке формы.

Атрибуты тега <optgroup> приведены в таблице 3.8.

Таблица 3.8 – Атрибуты тега <optgroup>

Атрибут	Значение / описание
disabled	Отключает данную группу элементов списка для выбора.
Label	Задаёт заголовок для группы элементов выпадающего списка. Значение атрибута содержит текст, недоступный для выбора, который будет располагаться над соответствующими пунктами списка. Текст выделяется в браузере жирным начертанием.

```
<form>
  <p><select name="food">
    <optgroup label="Русская кухня">
      <option value="r1">Закуска Барская</option>
      <option value="r2">Раки, фаршированные по-царски</option>
      <option value="r3">Биточки в горшочке</option>
    </optgroup>
    <optgroup label="Украинская кухня">
      <option value="u1">Галушки славянские</option>
      <option value="u2">Пампушки украинские</option>
      <option value="u3">Жаркое по-харьковски</option>
    </optgroup>
  </select></p>
  <p><input type="submit" value="Отправить"></p>
</form>
```

Надписи к элементам формы создаются с помощью элемента <label>...</label>. Существует два способа группировки надписи и поля. Если поле находится внутри элемента <label>, то атрибут for указывать не нужно.

```
<!-- с указанием атрибута for -->
<label for="comments">Когда вы последний раз летали на самолете?</label>
<textarea id="comments"></textarea>
```

<!-- без атрибута for -->

<p><label>Кошка<input id="cat" type="checkbox"></label></p>

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

Для корректного отображения элемента <button> разными браузерами нужно указывать атрибут type, например, <button type="submit"></button>. Атрибуты тега <button> приведены в таблице 3.9.

Таблица 3.9 – Атрибуты тега <button>

Атрибут	Значение / описание
autofocus	Устанавливает фокус на кнопке при загрузке страницы.
disabled	Отключает кнопку, делая ее некликабельной.
form	Указывает на одну или несколько форм, которым принадлежит данная кнопка. Значение атрибута — идентификатор соответствующей формы.
formaction	Значение атрибута содержит url-адрес обработчика данных формы, отправляемых при нажатии на кнопку. Только для кнопки типа type="submit". Переопределяет значение атрибута action, указанного для элемента <form>.
formenctype	Задаёт тип кодировки данных формы перед отправкой на сервер при нажатии на кнопки типа type="submit". Переопределяет значение атрибута enctype, указанного для элемента <form>. Возможные значения: application/x-www-form-urlencoded — значение по умолчанию. Все символы перед отправкой будут закодированы. multipart/form-data — символы не кодируются. Используется в случае, когда с помощью формы загружаются файлы. text/plain — символы не кодируются, а пробелы заменяются на символ +.
formmethod	Атрибут определяет метод, который браузер будет использовать для отправки формы. Переопределяет значение атрибута method, указанного для элемента <form>. Указывается только для кнопок типа type="submit". Возможные значения: get — данные из формы (пара имя/значение) добавляются в url-адрес и отправляются на сервер. Данный способ имеет ограничения по размеру отправляемых данных и не подходит для отправки паролей и конфиденциальной информации. post — данные из формы добавляются в виде http-запроса. Метод

Атрибут	Значение / описание
	является более надежным и безопасным, чем <code>get</code> и не имеет ограничений по размеру.
<code>formnovalidate</code>	Атрибут задает, что данные формы не должны проверяться при отправке. Указывается только для кнопок типа <code>type="submit"</code> .
<code>formtarget</code>	Атрибут задает, в каком окне выводить результат после отправки формы. Указывается только для кнопок типа <code>type="submit"</code> . Переопределяет значение атрибута <code>target</code> , указанного для элемента <code><form></code> . <code>_blank</code> — загружает ответ в новое окно/вкладку <code>_self</code> — загружает ответ в то же окно (значение по умолчанию) <code>_parent</code> — загружает ответ в родительский фрейм <code>_top</code> — загружает ответ во весь экран <code>framename</code> — загружает ответ во фрейм с указанным именем.
<code>name</code>	Задаёт имя кнопки, значение атрибута — текст. Используется для ссылки на данные формы, после того как форма была отправлена, или для ссылки на данную кнопку (кнопки) в JavaScript.
<code>type</code>	Определяет тип кнопки. Возможные значения: <code>button</code> — кликабельная кнопка <code>reset</code> — кнопка сброса, возвращает первоначальное значение <code>submit</code> — кнопка для отправки данных формы.
<code>value</code>	Задаёт значение по умолчанию, отправляемое при нажатии на кнопку.

```

<form>
<p><button>Кнопка с текстом</button>
<button>

Кнопка с рисунком
</button></p>
</form>

```

Флажки в формах задаются с помощью конструкции `<input type="checkbox">`, а переключатель – при помощи `<input type="radio">`.
Атрибуты переключателей приведены в таблице 3.10.

Таблица 3.10 – Атрибуты переключателей

Атрибут	Описание
Checked	Предварительное выделение переключателя. По определению, набор переключателей может иметь только один выделенный пункт, поэтому добавле-

	ние checked сразу к нескольким полям не даст особого результата. В любом случае, будет отмечен элемент, находящийся в коде HTML последним.
Name	Имя группы переключателей для идентификации поля. Поскольку переключатели являются групповыми элементами, то имя у всех элементов группы должно быть одинаковым.
Value	Задаёт, какое значение будет отправлено на сервер. Здесь уже каждый элемент должен иметь свое уникальное значение, чтобы можно было идентифицировать, какой пункт был выбран пользователем.

```
<input type=radio name=answer value=yes checked>Да
<input type=radio name=answer value=no>Нет
```

Флажков, в отличие от переключателей, в одной форме может быть установлено несколько. Если для флажков указан атрибут checked, то при загрузке страницы на соответствующих полях формы флажки уже будут установлены.

Атрибуты флажков приведены в таблице 3.11.

Таблица 3.11 – Атрибуты флажков

Атрибут	Описание
Checked	Предварительное выделение флажка.
Name	Имя флажка для его идентификации обработчиком формы.
Value	Задаёт, какое значение будет отправлено на сервер.

```
<form action="handler.php">
<p>В каких годах произошли самые известные извержения вулкана Крака-
тау?</p>
<p><input type="checkbox" name="a" value="1417"> 1417</p>
<p><input type="checkbox" name="a" value="1680" checked> 1680</p>
<p><input type="checkbox" name="a" value="1883" checked> 1883</p>
<p><input type="checkbox" name="a" value="1934"> 1934</p>
<p><input type="checkbox" name="a" value="2010"> 2010</p>
<p><input type="submit" value="Отправить"></p>
</form>
```

Элемент <label> применяется при реализации выбора с помощью переключателей и флажков. Можно выбрать нужный пункт, просто щелкая кнопкой мыши на тексте, связанном с ним. Для этого нужно поместить <input> внутрь элемента <label>.

Элементы формы традиционно располагаются внутри тега <form>, тем самым определяя те данные, которые будут передаваться на сервер. В то же время в HTML5 есть возможность отделить форму от её элементов.

```
<body>
<form id="auth" action="handler.php" method="post"></form>
```



```
<p>...</p>
<p><input name="login" form="auth">
<input type="password" name="pass" form="auth"></p>
<p><input type="submit" form="auth"></p>
</body>
```

5. Порядок выполнения работы

1. Создайте HTML-страницу. Добавьте на страницу форму в соответствии с вариантом. Форма должна содержать различные типы элементов. Не менее одного поля должно содержать шаблон ввода данных и подсказку.

Варианты:

1. Регистрация на сайте программистов
 2. Отправки заказа в магазине
 3. Отзыв о культурном мероприятии
 4. Регистрация на сайте поиска вакансий
 5. Поиск и фильтрация в магазине техники
 6. Поиск и фильтрация на сайте вакансий
 7. Добавление нового товара
 8. Заказ билета на самолет
 9. Покупка билета на концерт
 10. Добавление блюда меню
 12. Поиск и фильтрация людей на сайте знакомств
 13. Запрос на выдачу кредита
 14. Добавление информации о рейсе
 15. Добавление информации о мероприятии
 16. Поиска и фильтрации книг
 17. Поиск и фильтрация фильмов
 18. Поиск и фильтрация игр
 19. Оформление денежного перевода
 20. Вопрос на форуме программистов
 21. Поиск и фильтрация вопросов на форуме программистов
 22. Поиск и фильтрация платежей
 23. Поиск и фильтрация блюд в ресторане
 24. Поиска и фильтрации врачей в поликлинике
 25. Добавление пациента
 26. Заказ ремонта автомобиля
 27. Поиск и фильтрация автомобилей
 28. Поиск и фильтрация ресторанов
 29. Регистрация фирмы
 30. Заявка на замену паспорта
2. Обеспечить валидацию формы средствами HTML.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы: _____

7. Контрольные вопросы и задания

1. С помощью какого тега создаются формы в веб-документе?
2. Какие элементы форм можно создать с помощью тега `<input>`?
3. Охарактеризуйте элементы форм, используемые при выполнении лабораторной работы.
4. Сколько форм можно добавлять на одну веб-страницу?

8. Рекомендуемая литература

1. **Гарднер, Л.** Разработка веб-сайтов для мобильных устройств / Л. Гарднер, Д. Григсби. – СПб.: Питер, 2013.
2. **Дронов, В.** HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. – СПб.: БХВ-Петербург, 2014.
3. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. 4-е изд. – СПб.: Питер, 2018.