**Тег <input>**

Тег **<input>** является одним из разносторонних элементов формы и позволяет создавать разные элементы интерфейса и обеспечить взаимодействие с пользователем. Главным образом **<input>** предназначен для создания текстовых полей, различных кнопок, переключателей и флажков. .

Основной атрибут тега **<input>**, определяющий вид элемента — type. Он позволяет задавать следующие элементы формы:

**<input type="text" size="30" placeholder="Ваше Имя">**

**<input type="password" size="8">**

**<input type="checkbox" checked="checked">**

**<input type="radio" name="radio">**

**<input type="hidden" value="скрытое поле">**

**<input type="file">**

**<input type="submit" value="Сохранить">**

**<input type="reset" value="Очистить">**

**<input type="button" value="просто кнопка">**

Для каждого элемента существует свой список атрибутов, которые определяют его вид и характеристики.

### **Атрибуты**

**align ="bottom | left | middle | right | top"**

Определяет выравнивание изображения.

**alt ="Текст"**

Альтернативный текст для кнопки с изображением.

**autocomplete ="on | off"**

Включает или отключает автозаполнение.

**autofocus**

Устанавливает фокус в поле формы. В таком поле можно сразу набирать текст без явного щелчка по нему курсором мыши. <input autofocus>

**border ="толщина рамки"**

Толщина рамки вокруг изображения.

**checked**

Определяет, помечен ли заранее такой элемент формы, как флажок или переключатель. В случае использования переключателей (radiobutton), может быть отмечен только один элемент группы, для флажков (checkbox) допустимо пометить хоть все элементы.

**disabled или disabled="disabled"**

Блокирует доступ и изменение элемента.

**form="<идентификатор>"**

Связывает поле с формой по её идентификатору.

**formaction ="URL"**

Определяет адрес обработчика формы. <input type="submit" formaction="URL">

**list ="<идентификатор>"**

Указывает на список вариантов, созданный с помощью тега <datalist>, которые можно выбирать при наборе текста. Изначально этот список скрыт и становится доступным при получении полем фокуса.

<input list="идентификатор">

<datalist id=идентификатор">

<option value="Текст1">

<option value="Текст2">

</datalist>.

**maxlength="число"**

Максимальное количество символов разрешенных в тексте.

**multiple** Позволяет загрузить несколько файлов одновременно. <input type="file" multiple>

**name="имя**" Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать.

**pattern="выражение"** Указывает регулярное выражение, согласно которому требуется вводить и проверять данные в поле формы. Если присутствует атрибут pattern, то форма не будет отправляться, пока поле не будет заполнено правильно.

**placeholder= "строка"** Выводит подсказывающий текст внутри поля формы, который исчезает при получении фокуса.

**readonly** Устанавливает, что поле не может изменяться пользователем. <input type="text" readonly>

**required** Обязательное для заполнения поле. <input type="text" required>

**size ="ширина"** Ширина текстового поля, которое определяется числом символов моноширинного шрифтаЗначение по умолчанию 20<input type="text" size="ширина">

**src ="URL"** Адрес графического файла, который будет отображаться на веб-странице в поле с изображением.

**tabindex="число"** Определяет порядок перехода между элементами с помощью клавиши Tab.

**type ="button | checkbox | file | hidden | image | password | radio | reset | submit | text"**

Сообщает браузеру, к какому типу относится элемент формы.

**value ="значение"** Определяет значение элемента формы, которое будет отправлено на сервер или получено с помощью клиентских скриптов.

Значение атрибута type для тега **<input>** по умолчанию определено как **text**, поэтому его можно не указывать явно.У первого тега <input /> значение атрибута type="text", это означает, что это обычное однострочное поле ввода. В значении атрибута size указывается размер этого поля в символах. Атрибут placeholder выводит текст внутри текстового поля, который исчезает при начале ввода символов.

Атрибуты текстового поля перечислены в табл. 1.

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| size | Ширина текстового поля, которая определяется числом символов моноширинного шрифта. Иными словами, ширина задается количеством близстоящих букв одинаковой ширины по горизонтали. |
| maxlength | Устанавливает максимальное число символов, которое может быть введено пользователем в текстовом поле. Когда это количество достигается при наборе, дальнейший ввод становится невозможным. Если этот атрибут опустить, то можно вводить строку длинее самого поля. |
| name | Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать. |
| value | Начальный текст отображаемый в поле. |

Создание текстового поля показано в примере 1.

**Пример 1. Текстовое поле**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Текстовое поле**</title>**

**</head>**

**<body>**

**<form** action="handler.php"**>**

**<p><strong>**Как вас зовут?**</strong></p>**

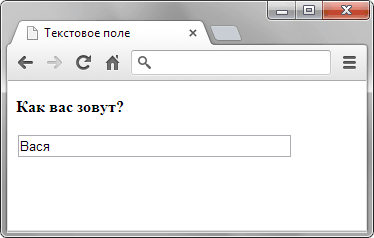
**<p><input** maxlength="25" size="40" value="Вася"**></p>**

**</form>**

**</body>**

**</html>**

В результате получим следующее (рис. 1).



**Поле для пароля** — обычное текстовое поле, но отличается от него тем, что все вводимые символы отображаются звездочками, точками или другими знаками (это зависит от браузера). Поле предназначено для того, чтобы никто не подглядел вводимый пароль.

Синтаксис создания следующий.

**<input** type="password" **атрибуты>**

Атрибуты совпадают с атрибутами текстового поля.

В формах часто требуется ввод адреса **электронной почты**, для чего обычно применяется однострочное текстовое поле. Однако в отличие от него специальное поле для ввода адреса почты позволяет проверять корректность записи введённого адреса.

Синтаксис создания поля следующий.

**<input** type="email" **атрибуты>**

Атрибуты по большей части совпадают с текстовым полем и приведены в табл. 1.

Табл. 1. Атрибуты поля для почтового адреса

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| maxlength | Устанавливает максимальное число символов, которое может быть введено пользователем в поле. Когда это количество достигается при наборе, дальнейший ввод становится невозможным. Если этот атрибут опустить, то можно вводить строку длинее самого поля. |
| multiple | Позволяет указывать несколько адресов через запятую. |
| name | Имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать. |
| size | Ширина поля, которая определяется числом символов моноширинного шрифта. Иными словами, ширина задается количеством близстоящих букв одинаковой ширины по горизонтали. |
| value | Начальный почтовый адрес отображаемый в поле. |

По сравнению со значением text, добавлен атрибут multiple, который позволяет вводить сразу несколько почтовых адресов.

В примере 1 показано применение значения email для создания формы авторизации на сайте, где в качестве логина указывается адрес электронной почты.

**Пример 1. Адрес электронной почты**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Авторизация на сайте**</title>**

**</head>**

**<body>**

**<form** action="handler.php"**>**

**<p>**Вход на сайт**</p>**

**<p>**Email: **<input** type="email" name="login"**></p>**

**<p>**Пароль: **<input** type="password" name="pass"**></p>**

**<p><input** type="submit" value="Вход"**></p>**

**</form>**

**</body>**

**</html>**

По своему виду поле для ввода адреса ничем не отличается от текстового поля. Различия проявляются, если указать некорректный адрес, в этом случае браузер выведет замечание об ошибке.

**Для ввода адресов сайтов** или, как их ещё называют веб-адресов, предназначено значение url тега **<input>**, которое делает проверку на правильность ввода данных. Каждый веб-адрес должен начинаться с протокола (http://, https://, ftp://), больше ограничений нет — адрес может быть набран латинскими символами, кириллицей, содержать точку или наоборот, писаться без неё. Браузер Opera не требует даже наличие протокола, подставляя «http://» перед текстом автоматически в случае его отсутствия.

Синтаксис написания поля для веб-адреса следующий.

**<input** type="url" **атрибуты>**

Атрибуты совпадают с текстовым полем (<input type="text">).

В примере 1 показано использование поля для веб-адреса, которое должно быть заполнено перед отправкой формы.

**Пример 1. Ввод адреса сайта**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Веб-адрес**</title>**

**</head>**

**<body>**

**<form>**

**<p><input** placeholder="Ваше имя" name="user"**></p>**

**<p><input** type="url" placeholder="Сайт" name="site" required**></p>**

**<p><input** type="submit" value="Отправить"**></p>**

**</form>**

**</body>**

**</html>**

В данном примере при нажатии на кнопку «Отправить» браузер будет выводить сообщение, что необходимо правильно ввести URL.

Чтобы указать **телефонный номер** используйте для этого специальное поле. По своему виду и работе оно совпадает с текстовым полем. Синтаксис создания этого поля следующий.

**<input** type="tel" **атрибуты>**

Атрибуты полностью совпадают с текстовым полем.

В отличие от веб-адреса или адреса электронной почты, поле для телефона не проверяется на правильность синтаксиса. Это связано с тем, что телефонные номера в разных местах мира могут иметь самые разнообразные формы написания, включать цифры, буквы и другие символы. Смысл применения этого поля в настройке шаблона ввода телефона, который используется в определённом городе или стране, а также в некоторых расширенных возможностях по управлению стилями.

Создание поля для телефона показано в примере 1. Требуется ввести телефон в указанном формате, иначе браузер выведет сообщение об ошибке и не станет отправлять форму, пока поле не будет заполнено корректно.

**Пример 1. Поле для телефона**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Телефон**</title>**

**</head>**

**<body>**

**<form>**

**<p>**Ваше имя: **<input** name="login"**></p>**

**<p>**Телефон в формате 2xxx-xxx: **<input** type="tel" name="tel"

pattern="2[0-9]{3}-[0-9]{3}"**></p>**

**<p><input** type="submit" value="Отправить"**></p>**

**</form>**

**</body>**

**</html>**

**Переключатели** (жарг. радиокнопки) используют, когда необходимо выбрать один единственный вариант из нескольких предложенных. Создаются следующим образом.

**<input** type="radio" name="имя" **атрибуты>**

Атрибуты переключателей перечислены в табл. 1.

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| checked | Предварительное выделение переключателя. По определению, набор переключателей может иметь только один выделенный пункт, поэтому добавление 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. |

Создание группы переключателей показано в примере 1.

**Пример 1. Создание переключателей**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Переключатели**</title>**

**</head>**

**<body>**

**<form** action="handler.php"**>**

**<p><b>**Какое у вас состояние разума?**</b></p>**

**<p><input** name="dzen" type="radio" value="nedzen"**>** Не дзен**</p>**

**<p><input** name="dzen" type="radio" value="dzen"**>** Дзен**</p>**

**<p><input** name="dzen" type="radio" value="pdzen" checked**>** Полный дзен**</p>**

**<p><input** type="submit" value="Выбрать"**></p>**

**</form>**

**</body>**

**</html>**

Заметьте, что в данном примере значение атрибута name для всех переключателей одинаково, именно в таком случае браузер понимает, что переключатели связаны между собой и помечает только один пункт из предложенных. Значение атрибута value же должно различаться, чтобы обработчик формы мог понять, какой вариант выбран пользователем.

**Флажки** (жарг. чекбоксы) используют, когда необходимо выбрать любое количество вариантов из предложенного списка. Если требуется выбор лишь одного варианта, то для этого следует предпочесть переключатели (radiobutton). Флажок создается следующим образом.

**<input** type="checkbox" **атрибуты>**

Атрибуты флажков такие же как и у переключателей.

Использование флажков показано в примере 1.

**Пример 1. Создание флажков**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Флажки**</title>**

**</head>**

**<body>**

**<form** action="handler.php"**>**

**<p>**В каких годах произошли самые известные извержения вулкана Кракатау?**</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>**

**</body>**

**</html>**

Эти два варианта <input type="checkbox" /> и <input type="radio" /> предоставляют

пользователю возможность право выбора некоторых пунктов. В первом случае есть возможность множественного выбора нескольких пунктов в виде флажков, а во втором — единственного выбора в виде переключателей. В обоих случаях есть возможность отметить любой пункт по умолчанию, если задать атрибут checked="checked".

**Кнопки** являются одним из самых понятных и интуитивных элементов интерфейса. По их виду сразу становится понятно, что единственное действие, которое с ними можно производить — это нажимать на них. За счёт этой особенности кнопки часто применяются в формах, особенно при их отправке и очистке.

Кнопку на веб-странице можно создать двумя способами — с помощью тега **<input>** и тега **<button>**.

Рассмотрим вначале добавление кнопки через **<input>** и его синтаксис.

**<input** type="button" **атрибуты>**

Атрибуты кнопки перечислены в табл. 1.

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| name | Имя кнопки, предназначено для того, чтобы обработчик формы мог его идентифицировать. |
| value | Значение кнопки и одновременно надпись на ней. |

Создание кнопки показано в примере 1.

**Пример 1. Добавление кнопки**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Кнопка**</title>**

**</head>**

**<body>**

**<form>**

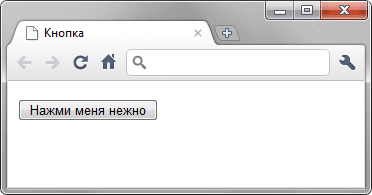
**<p><input** type="button" value=" Нажми меня нежно "**></p>**

**</form>**

**</body>**

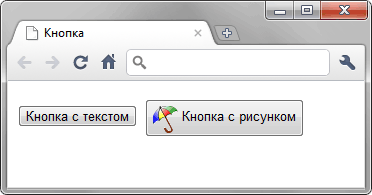
**</html>**

Пробелы в надписи на кнопке, в отличие от текста HTML, учитываются, поэтому можно ставить любое количество пробелов, которые в итоге влияют на ширину кнопки. Результат примера показан на рис. 1.



*Рис. 1. Вид кнопки*

Второй способ создания кнопки основан на использовании тега **<button>**. Он по своему действию напоминает результат, получаемый с помощью тега **<input>**. Но в отличие от него предлагает расширенные возможности по созданию кнопок. Например, на подобной кнопке можно размещать любые элементы HTML включая изображения и таблицы. На рис. 2 показаны разные виды кнопок, полученные с помощью **<button>**.



*Рис. 2. Кнопки, созданные с помощью <button>*

Синтаксис создания такой кнопки следующий.

**<button атрибуты>**Надпись на кнопке**</button>**

Атрибуты перечислены в табл. 1, но в отличие от кнопки **<input>** атрибут value определяет только отправляемое на сервер значение, а не надпись на кнопке. Если требуется вывести на кнопке изображение, то тег **<img>** добавляется внутрь **<button>**, как показано в примере 2.

**Пример 2. Рисунок на кнопке**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Кнопка**</title>**

**</head>**

**<body>**

**<form>**

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

**<button>**

**<img** src="images/umbrella.gif" alt="" style="vertical-align:middle"**>**

Кнопка с рисунком

**</button></p>**

**</form>**

**</body>**

**</html>**

В данном примере показано создание обычной кнопки с текстом, при этом , а также кнопки с одновременным использованием текста и рисунка. Размер кнопки зависит от содержимого контейнера **<button>**, но пробелы игнорируются, поэтому простым увеличением их количества, как в случае использования **<input>**, ширину кнопки изменить не удастся.

## Кнопка Submit

Для отправки данных на сервер предназначена специальная кнопка Submit. Её вид ничем не отличается от обычных кнопок, но при нажатии на нее происходит выполнение серверной программы, указанной атрибутом action тега **<form>**. Эта программа, называемая еще обработчиком формы, получает данные, введенные пользователем в полях формы, производит с ними необходимые манипуляции, после чего возвращает результат в виде HTML-документа. Что именно делает обработчик, зависит от автора сайта, например, подобная технология применяется при создании опросов, форумов, тестов и многих других вещей.

Синтаксис создания кнопки Submit зависит от используемого тега **<input>** или **<button>**.

**<input** type="submit" **атрибуты>**

**<button** type="submit"**>**Надпись на кнопке**</button>**

Атрибуты те же, что и у рядовых кнопок (пример 3).

**Пример 3. Отправка данных на сервер**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Кнопка**</title>**

**</head>**

**<body>**

**<form>**

**<p><input** name="login"**></p>**

**<p><input** type="submit"**></p>**

**</form>**

**</body>**

**</html>**

Атрибут name для этого типа кнопки можно не писать. Если не указать значение value, то браузер автоматически добавит текст, он различается в зависимости от браузера. Так, Firefox пишет «Отправить запрос», IE — «Подача запроса», Opera и Chrome — «Отправить». Сам текст надписи никак на функционал кнопки не влияет.

## Кнопка Reset

При нажатии на кнопку Reset данные формы возвращаются в первоначальное значение. Как правило, эту кнопку применяют для очистки введенной в полях формы информации. Для больших форм от использования кнопки Reset лучше вообще отказаться, чтобы по ошибке на нее не нажать, ведь тогда придётся заполнять форму заново.

Синтаксис создания указанной кнопки прост и похож на другие кнопки.

**<input** type="reset" **атрибуты>**

**<button** type="reset"**>**Надпись на кнопке**</button>**

В примере 4 показана форма с одним текстовым полем, которое уже содержит предварительно введённый текст с помощью атрибута value тега **<input>**. После изменения текста и нажатия на кнопку «Очистить», значение поля будет восстановлено и в нём снова появится надпись «Введите текст».

**Пример 4. Кнопка для очистки формы**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Кнопка**</title>**

**</head>**

**<body>**

**<form>**

**<p><input** value="Введите текст"**></p>**

**<p><input** type="submit" value="Отправить"**>**

**<input** type="reset" value="Очистить"**></p>**

**</form>**

**</body>**

**</html>**

Значение кнопки Reset никогда не пересылается на сервер. Если надпись на кнопке опустить, иными словами, не задавать атрибут value, на кнопке по умолчанию будет добавлен текст «Очистить».

Для отправки формы на сервер применяется кнопка Submit, которая имеет слабые средства по настройке своего вида. Чтобы не ограничивать дизайн и расширить возможности по оформлению форм, используйте **поле с изображением**, аналогичное по своему действию кнопке Submit. При нажатии на рисунок данные формы отправляются на сервер и обрабатываются программой, заданной атрибутом action тега **<form>**.

Поле с изображением создаётся следующим образом.

**<input** type="image" src="URL" alt="альтернативный текст"**>**

Здесь URL это адрес изображения в формате JPEG, PNG или GIF, alt указывает альтернативный текст, который виден при отключении картинок в браузере. Вообще, это поле похоже на добавление в код изображения и работает подобно элементу **<img>**.

В примере 1 показано использование поля с изображением.

**Пример 1. Кнопка с изображением**

<!DOCTYPE html>

**<html>**

**<head>**

**<meta** charset="utf-8"**>**

**<title>**Загрузка файла**</title>**

**<style>**

**body** { background: #053f38; color: #98baba; }

**p**.question { color: #ffd595; }

**</style>**

**</head>**

**<body>**

**<form** action="handler.php"**>**

**<p** class="question"**>**В какую игру вы предпочитаете играть?**</p>**

**<p><input** type="radio" name="game" value="1"**>** Руммикуб**</p>**

**<p><input** type="radio" name="game" value="2"**>** Колонизаторы**</p>**

**<p><input** type="radio" name="game" value="3"**>** Каркассон**</p>**

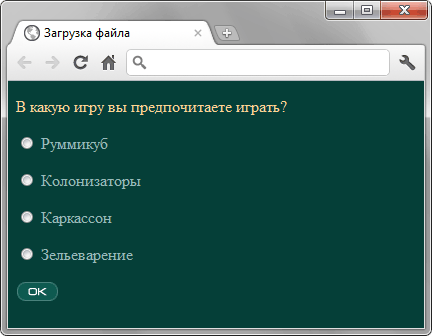
**<p><input** type="radio" name="game" value="4"**>** Зельеварение**</p>**

**<p><input** type="image" src="images/okbutton.png" alt="OK"**>**

**</form>**

**</body>**

**</html>**



В отличие от кнопки Submit на сервер также отправляются координаты точки щелчка по изображению в виде x и y. Отсчёт координат ведётся от левого верхнего угла картинки и указывается в пикселах. Если добавить к полю с изображением уникальное имя через атрибут name, например, ok, тогда координаты передаются в виде ok.x и ok.y, где впереди через точку стоит имя поля.