

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 использовать по своему усмотрению.

Тег `<form>`, как и тег ``, сам по себе ничего не добавляет. Внутри него располагаются теги для взаимодействия с пользователем, например, `<button>`.

Тег `<button>`

Тег `<button>` создаёт на веб-странице кнопки, на которых можно размещать любые элементы HTML, включая изображения. С помощью стилей можно изменять вид кнопки, меняя шрифт, цвет фона, размеры и другие параметры.

Теоретически тег `<button>` должен располагаться внутри формы, устанавливаемой тегом `<form>`. Однако браузеры не выводят сообщение об ошибке, если тег `<button>` встречается отдельно. Если нажатие на кнопку должно отправлять данные на сервер, размещать `<button>` внутри контейнера

```
<form action="">
```

```
  <button>Кнопка</button>
```

Тег `<input>`

Тег `<input>` является универсальным элементом формы и позволяет создавать различные элементы интерфейса, обеспечивая взаимодействие с пользователем. Он может быть использован для создания текстовых полей, кнопок, переключателей и флажков.

Хотя элемент `<input>` не обязательно помещать внутрь контейнера `<form>`, если данные пользователя должны быть отправлены на сервер, форма обязательно должна включать `<form>`. Это также важно при обработке данных с помощью клиентских скриптов, например, на языке

```
<form action="">
```

```
  <input type="text">
```

```
  <button>Кнопка</button>
```

```
</form>
```

Основной атрибут тега `<input>`, определяющий его вид — `type`. Он позволяет задавать следующие элементы формы: текстовое поле (`text`), поле с паролем (`password`), переключатель (`radio`), флажок (`checkbox`), скрытое поле (`hidden`), кнопка (`button`), кнопка для отправки формы (`submit`), кнопка для очистки формы (`reset`), поле для отправки файла (`file`), кнопка с изображением (`image`). В HTML5 добавлено еще более десятка новых элементов.

А так же рассмотрим важный атрибут `placeholder` который подсказывает пользователю, что именно нужно ввести в поле ввода (обычно это пример слова или фразы). Выглядит это как будто

поле уже заполнено, только гораздо бледнее. А ещё значение атрибута исчезает, как только пользователь начинает вводить текст.

```
<form action="">

  <input type="text" placeholder="Введите ваше имя">

  <textarea name="" id="">введите Ваш комментарий</textarea>

</form>
```

Тег <textarea>

Тег <textarea> создаёт область для ввода многострочного текста. В отличие от тега <input>, в этом поле допустимы переносы строк, которые сохраняются при отправке данных на сервер.

Между тегами <textarea> и </textarea> можно поместить текст, который будет отображаться внутри поля.

```
Введите ваш комментарий</textarea>

</form>
```

Тег <select>

Тег <select> создаёт раскрывающийся список или список с одним или множественным выбором. Конечный вид списка зависит от атрибута size, который устанавливает высоту списка. Каждый пункт списка создаётся с помощью тега <option>, вложенного в контейнер <select>. Если данные списка должны быть отправлены на сервер, элемент <select> необходимо поместить внутрь формы.

```
<form action="">

  <select name="" id="">

    <option value="">Вариант 1</option>

    <option value="">Вариант 2</option>

    <option value="">Вариант 3</option>

  </select>
```

Универсальные атрибуты тегов

Существует ряд универсальных атрибутов, которые можно задать любому тегу. Рассмотрим важнейшие из них.

- Атрибут title описывает содержимое элемента в виде всплывающей подсказки.

```
"Подсказка">Текст с подсказкой</p>
```

- Атрибуты class и id нужны для связывания тега со стилями CSS.
 - задаёт имя класса для оформления стилей.
 - указывает уникальный идентификатор элемента. Названия классов могут дублироваться, а id должен быть уникальным для каждого элемента.

Текст с классом и идентификатором</p>

Спецсимволы

Спецсимволы HTML (символы-мнемоники) используются для отображения символов, которых нет на стандартной клавиатуре или не поддерживаются кодировкой HTML-страницы (например,

Чтобы разместить спецсимвол на веб-странице, используется его HTML-код или мнемоника.

Пример использования спецсимволов:

Пример спецсимвола письма 🖂</p>

Спецсимвол телефонной трубки ✆</p>

Самостоятельно ознакомиться со списком всех спецсимволов и html-кодом для их использования можно [по ссылке](#)