

Цели занятия

- разобрать что такое формы и для чего они используются;
- изучить основные теги, которые используются для создания форм на сайтах;
- познакомиться с некоторыми тонкостями группировки элементов и стилизации форм

Полезные материалы

- [Справочник HTML](#)
- [HTML5-формы](#)
- [Как стилизовать элементы форм](#)

Формы

HTML-формы являются элементами управления, которые применяются для сбора информации от посетителей веб-сайта.

Веб-формы состоят из набора текстовых полей, кнопок, списков и других элементов управления, с которыми можно взаимодействовать и заполнять необходимыми данными.

Основу любой формы составляет элемент `<form>...</form>`. Он 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.

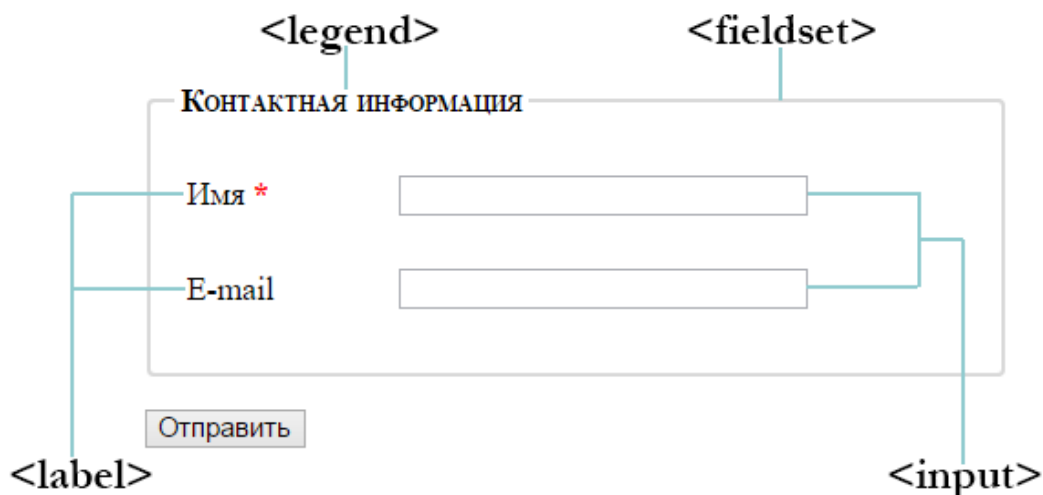
Более детально атрибуты тега `<form>` мы будем изучать в модуле JavaScript, когда будем добавлять функционал форм, на этапе вёрстки страницы эти атрибуты не требуются, но можно [ознакомиться с ними в справочнике](#).

Группировка элементов формы

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 открывающим тегом элемента `<fieldset>`. Название группы проявляется слева в верхней границе `<fieldset>`. Например, если в элементе `<fieldset>` хранится контактная информация:

```
<form>
  <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>
</form>
```



Поля ввода

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

С помощью CSS-стилей можно изменить размер шрифта, тип шрифта, цвет и другие свойства текста, а также добавить границы, цвет фона и фоновое изображение. Ширина поля задается свойством `width`.

`<input type="text">` — указывает, что `input` должен быть текстовым полем, в которое, как правило, необходимо ввести одно или несколько слов. Если вам необходимо поле, в которое пользователи могут писать много текста, то крайне желательно использовать тег [`<textarea>`](#)

`<input type="password">` — указывает, что `input` является полем ввода скрытых данных. Как правило используется для ввода паролей, символы которого автоматически скрываются звездочками;

`<input type="radio">`: данный вид `input` используется когда необходимо осуществить единственный выбор из нескольких вариантов, например, для создания опросов. Но когда мы указываем `<input type="radio">`, необходимо так же указать еще один атрибут `<input type="radio" name="название">`, чтобы привязать несколько вариантов к названию:

```
<input type="radio" name="answer">  
<input type="radio" name="answer">  
<input type="radio" name="answer">
```

`<input type="checkbox">`: работает примерно как `type="radio"`, но предоставляет возможность выбрать сразу несколько вариантов, также необходимо указывать одинаковый `name` для каждого варианта;

`<input type="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. Например, `<input type="number" min="1" max="5" step="1">`

`<input type="reset">`: создает кнопку, которая очищает поля формы от введенных пользователем данных.

Текстовые поля ввода

Элемент `<textarea>` используется вместо элемента `<input type="text">`, когда нужно создать большие текстовые поля. Текст, отображаемый как исходное значение, помещается внутрь.

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.

По умолчанию браузеры разрешают пользователям изменять размеры такого поля ввода, но это поведение можно изменить или отключить с помощью CSS-свойства [resize](#)

Подсказки

Текстовые поля очень часто имеют подсказки, которые будут автоматически скрываться как только пользователь начнёт заполнять форму. Значением подсказки можно управлять с помощью атрибута `placeholder`, например, `<input type="text" name="phone" placeholder="Ваш номер телефона">` или `<textarea name="text" placeholder="Текст отзыва">`

Зачастую дизайн требует изменения стандартного вида подсказок и в CSS это делается с помощью вот такой записи ([подробнее](#)):

```
.input:placeholder {  
  color: red;  
}
```

Раскрывающийся список

Списки дают возможность расположить большое количество пунктов компактно. Раскрывающиеся списки создаются при помощи элемента `<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>`, которые располагаются внутри `<select>`.

Для систематизации списков применяется элемент `<optgroup>`, который создает заголовки в списках.

```
<select>
```

```
<option>Пункт 1</option>
```

```
<option>Пункт 2</option>
```

```
</select>
```