

# HTML CSS basics





## Построение форм

Форма предназначена для обмена данными между пользователем и сервером.

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.



## Инициализация формы

Для добавления формы на страницу, используется элемент `form`

Данный элемент определяет, где на странице появятся элементы управления. Кроме того, элемент `form` оборачивает все элементы включенные в форму, подобно элементу `div`

```
<form action="" method=""> ... </form>
```

К элементу `form` можно применить как универсальные атрибуты, так и специфические, наиболее используемыми из которых являются `action` и `method`

Атрибут `action` содержит URL, на который информация в форме будет отправлена для обработки сервером.

Атрибут `method` является методом HTTP, который должны использовать браузеры для отправки данных формы.

элементы формы

“



# input

Элемент **input** является одним из разносторонних элементов формы. Он позволяет создавать различные элементы интерфейса, дабы обеспечить взаимодействие с пользователем.

Главным образом **input** предназначен для создания однострочных текстовых полей, различных кнопок, переключателей и флажков.

**input** не требует контейнера **form**, определяющего форму. Но если введенные пользователем данные должны быть отправлены на сервер, использование родительского элемента **form** обязательно.



## Атрибуты input

**autofocus** – автоматическое получение фокуса

**checked** – предварительно активированный переключатель или флажок

**form** – связывает текстовое поле с формой по её идентификатору

**maxlength** – максимальное число введенных символов

**name** – имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать

**placeholder** – указывает замещающий текст

**readonly** – устанавливает, что поле не может изменяться пользователем

**required** – обязательное для заполнения поле

**type** – сообщает браузеру, к какому типу относится элемент формы

**value** – значение элемента



## Атрибут type

Основным атрибутом тега `<input>`, определяющим вид элемента, является `type`

Он позволяет задавать следующие (наиболее часто используемые) элементы формы:

`text` — текстовое поле

`password` — поле с паролем

`radio` — переключатель

`checkbox` — флажок

`submit` — кнопка для отправки формы

Для каждого элемента существует список специальных атрибутов, определяющих его вид и характеристики.

В HTML5 добавлено более десятка новых `input`'ов.

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



## textarea

textarea представляет собой элемент формы, создающий поле для ввода несколько строк текста.

В текстовом поле допустимы переносы строк. При этом такие переносы сохраняются при отправке данных на сервер.

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

<textarea атрибуты> текст </textarea>





## Атрибуты textarea

**autofocus** – автоматическое получение фокуса

**cols** – ширина поля в символах

**form** – связывает текстовое поле с формой по её идентификатору

**maxlength** – максимальное число введенных символов

**name** – имя поля, предназначено для того, чтобы обработчик формы мог его идентифицировать

**placeholder** – указывает замещающийся текст

**readonly** – устанавливает, что поле не может изменяться пользователем

**required** – обязательное для заполнения поле

**rows** – высота поля в строках текста



## Переключатели

Элемент формы, позволяющий пользователю выбрать **один** вариант из предложенного списка.

Чтобы создать переключатель, используется элемент **input** со значением **radio** у атрибута `type`

Каждый переключатель должен иметь одинаковое значение атрибута **name** в рамках группы выбора.

Конкретное же значение каждого варианта задается с помощью уникального значения атрибута **value**

Для выбора переключателя «по умолчанию» используется логический атрибут **checked**

```
<input type="radio" name="day" value="Friday" checked> Пятница  
<input type="radio" name="day" value="Saturday"> Суббота  
<input type="radio" name="day" value="Sunday"> Воскресенье
```



## Флажки

Флажки очень похожи на переключатели.

Они используют те же атрибуты и шаблоны, за исключением значения атрибута type

Разница между ними состоит в том, что флажки позволяют пользователю выбрать несколько вариантов, в то время как переключатели этот выбор ограничивают.

```
<input type="checkbox" name="day" value="Friday" checked> Пятница
```

```
<input type="checkbox" name="day" value="Saturday"> Суббота
```

```
<input type="checkbox" name="day" value="Sunday"> Воскресенье
```



## Выпадающие списки

Выпадающий список является элементом формы, предоставляющим пользователю доступ к списку вариантов выбора.

Для создания выпадающего списка применяются элементы **select** и **option**

Элемент **select** оборачивает все пункты меню, размеченные с помощью элементов **option**

Атрибут **name** располагается в теге **<select>**, а атрибут **value** в тегах **<option>**, вложенных в элемент **<select>**.

Каждый элемент **<option>** оборачивает текст (который виден пользователям) отдельного пункта в списке.

С помощью специального атрибута **selected** можно выбрать пункт «по умолчанию».

```
<select name="day">  
    <option value="Friday" selected>Пятница</option>  
    <option value="Saturday">Суббота</option>  
    <option value="Sunday">Воскресенье</option>  
</select>
```



## Кнопка для отправки в виде элемента `input`

Когда пользователь щёлкает по кнопке, данные заполненной формы предварительно проверяются и отправляются на сервер для обработки.

Кнопка для отправки создаётся с помощью элемента `input` со значением `submit` у атрибута `type`.

Атрибут `value` применяется для указания текста, который отображается внутри кнопки.

Кнопка для отправки в виде элемента `input` не может оборачивать любой другой контент! Также стоит отметить, что такой вариант кнопки крайне сложно стилизовать.

```
<input type="submit" name="submit" value="Отправить">
```



## Кнопка для отправки в виде элемента button

Для большего контроля над структурой и дизайном кнопки используется элемент `button`

Он выполняет те же функции, однако включает в себя открывающий и закрывающий теги, которые могут обернуть другие элементы.

По умолчанию элемент `button` действует, как если бы у атрибута `type` было задано значение `submit`, поэтому атрибут `type` и его значение можно опустить.

А вместо атрибута `value` используется текст, который пишется между открывающим и закрывающим тегами элемента `button`.

```
<button>
```

```
<strong>Отправьте нам</strong> сообщение
```

```
</button>
```



## Скрытое поле

Скрытые поля предоставляют способ передачи данных на сервер без доступа пользователя к этой информации.

Скрытые поля обычно используются для отслеживания кодов, ключей или другой информации, которая не имеет отношения к пользователю, но может быть полезна при обработке формы.

Эта информация не отображается на странице, однако может быть найдена путём просмотра исходного кода страницы.

Для создания скрытого поля используется элемент `input` со значение `hidden` атрибута `type`. Дополнительно включает в себя соответствующие значения атрибутов `name` и `value`.

```
<input type="hidden" name="tracking-code" value="abc-123">
```

# организация элементов формы

Знать, как получить данные с полей формы — лишь половина дела.  
Важный момент — это организация элементов формы и полей  
в удобном и логичном формате.

При взаимодействии с формами пользователи должны понимать,  
что от них требуется и как предоставить запрашиваемую информацию.

С помощью **label**, **fieldset** и **legend** можно структурировать формы  
и направлять пользователей по мере заполнения ими форм.



“





# label

Элемент `label` устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы.

Такая связь необходима, чтобы изменять значения элементов формы при нажатии курсором мыши на текст.

Существует два способа связывания объекта и метки:

- первый заключается в использовании идентификатора `id` внутри элемента формы и указании его имени в качестве атрибута `for` тега `<label>`
- второй вариант – элемент формы помещается внутрь контейнера `<label>`

```
<label for="идентификатор">Текст</label>
```

```
<input id="идентификатор" type="...">
```

```
<label>
```

```
<input type="...">
```

```
</label>
```



## fieldset

Элемент `fieldset` группирует поля формы в организованные разделы.

Подобно `section` или иным структурным элементам, `fieldset` является блочным элементом и оборачивает семантически связанные элементы.

```
<fieldset>
  <label>
    Имя пользователя
    <input type="text" name="username">
  </label>
  <label>
    Пароль
    <input type="password" name="password">
  </label>
</fieldset>
```



## legend

Элемент `legend` предоставляет заголовок для элемента `fieldset` – оборачивает текст, описывающий элементы управления формы, которые находятся внутри `fieldset`

Разметка должна включать в себя элемент `legend` следующий сразу за открывающим тегом `<fieldset>`

На странице подпись появится в левом верхнем углу рамки `fieldset`.

```
<fieldset>
  <legend>Вход в систему</legend>
  <label>
    Имя пользователя
    <input type="text" name="username">
  </label>
  <label>
    Пароль
    <input type="password" name="password">
  </label>
</fieldset>
```

*основные атрибуты управления*

“



## disabled

Логический атрибут disabled «выключает» элемент управления.

Другими словами – делает его недоступным для взаимодействия.

Заблокированные элементы не будут отправлять никакого значения на сервер для обработки формы.

Применение атрибута disabled к элементу fieldset отключит все элементы управления формы внутри группы.

```
<label>
```

```
Имя пользователя
```

```
<input type="text" name="username" disabled>
```

```
</label>
```



## placeholder

Атрибут `placeholder` предлагает подсказку или совет внутри элемента `input` или `textarea`, которая исчезает с началом введения текста в поле.

```
<label>
```

```
    Email
```

```
    <input type="email" name="email-address" placeholder="name@domain.com">
```

```
</label>
```



## required

Логический атрибут **required** утверждает, что элемент формы обязан содержать значение при отправке на сервер.

Если у элемента формы нет значения, будет отображаться сообщение об ошибке с просьбой заполнить обязательное поле.

Проверка специфична для конкретного типа элемента управления.

Например, элемент `<input>` со значением `email` у атрибута `type` потребует не только чтобы значение в поле существовало, но также чтобы это был корректный адрес электронной почты.

```
<label>
```

```
    Email
```

```
    <input type="email" name="email-address" required>
```

```
</label>
```