

# Занятие 5

HTML

Атрибуты и формы

# Атрибут style и приоритеты CSS стилей

*// Для использования CSS стилей можно использовать также атрибут style.*

```
<div class="team" style="text-align: center;"></div>
```

*// Такая запись является наиболее приоритетной, нежелателен к повсеместному использованию в редких случаях допустимо.*

*// Сделать стиль более приоритетным можно только используя !important в конце описания стиля. Крайне не желателен к использованию.*

```
background: red !important;
```

# Форма и ее элементы

`<form>...</form>` Тег для размещения на странице формы, которая нужна для передачи данных на сервер.

```
// // Для формы используется атрибут  
action="url";
```

```
// // На фронте вместо url при  
разработке можно использовать "#";
```

```
// Любые элементы формы могут  
использоваться как внутри формы  
<form>, так и вне её.
```

You have registered successfully!

User Name

First Name

Last Name

Password

Confirm Password

Email

Gender

Male  Female

I accept Terms and Conditions

Register

# <input> и его атрибуты

**<input />** Один из самых распространенных элементов формы. Представляет собой поле ввода, функции и интерфейс которых определяется атрибутом `type`.

`// type` - определяет интерфейс и функционал поля ввода

`// //` Возможные значения атрибута `type` [https://jsfiddle.net/r\\_agevchik/8ekcxm1a/](https://jsfiddle.net/r_agevchik/8ekcxm1a/);

`// //` Обновленные атрибуты `type` из HTML5 [https://jsfiddle.net/r\\_agevchik/db32sgav/1/](https://jsfiddle.net/r_agevchik/db32sgav/1/);

`// placeholder` - Описание поля, выводится только если поле не заполнено;

`// value` - Значение поля ввода;

`// name` - Имя для поля ввода, в верстке используются для автозаполнения;

`// maxlength` - Максимальное количество символов, разрешенных в поле ввода;

`// size` - Ширина поля;

`// checked` - Предварительно установленный переключатель или флажок;

`// required` - Устанавливает режим обязательного для заполнения;

`// pattern` - Регулярное выражение, которое указывает по каким правилам надо заполнять данное поле;

`// tabindex` - Устанавливает порядок переключения полей при нажатии на клавишу TAB;

`// disabled` - Для неактивных полей;

`// readonly` - Устанавливает режим "только для чтения";

`//` Также для `input type="checkbox || radio"` есть CSS псевдоселектор `:checked`, который позволяет задавать стили, при выбранном чекбоксе или радио-кнопке.

```
form input [type="checkbox"]:checked {...}
```

```
form input [type="checkbox"]:checked .content {...}
```

# <textarea> и <button>

<textarea>...</textarea> - Элемент формы, представляющий собой многострочное поле для ввода текста более одной строки.

// Для `textarea` действуют некоторые атрибуты, которые встречаются у `input`: `disabled`, `maxlength`, `name`, `placeholder`, `readonly`, `required`.

// Также есть и уникальные атрибуты:

// // `cols` - Ширина поля в символах;

// // `rows` - Высота поля в строках;

// И уникальное CSS свойство `resize: none | both | horizontal | vertical | inherit` для установки или отмены ресайза текстового поля.

```
<textarea name="comment" cols="30" rows="7"></textarea>
```

```
/*-----*/
```

<button>...</button> - Создает кнопку, внешне и функционально очень похожую на элемент `<input>` с соответствующим типом.

// // Вспоминаем, что должен содержать `type` для `<input>`, чтобы поле приняло вид кнопки;

// // Попробуем догадаться зачем было создавать ещё один лишний тег для кнопки;

```
<button>Click me =></button>
```

# Атрибуты <button> и раскрывающиеся списки

```
// Кнопка также заимствует многие атрибуты у input такие как:
disabled, name, type (button | reset | submit).

// Однако, в отличии от <input type="button" value="Submit form">, value - это не подпись кнопки, а
то, что она отправляет на сервер.
/*-----*/
<select>...</select> Элемент, позволяющий создать раскрывающийся список.
// Атрибуты disabled, name, required, tabindex мы уже изучили выше.
// multiple изменяет внешний вид списка, и позволяет выбрать сразу несколько элементов.
// size - отличается поведением от size для input. У списка size задает количество пунктов
<option>, ширина же по-умолчанию будет равна ширине самого большого <option>

<select required name="user-age">...</select>

// Каждый пункт меню создается с помощью тега <option>...</option>
// Для элемента списка используются следующие атрибуты:
// // selected - этот пункт будет выбран по-умолчанию;
// // disabled - этот пункт меню выбрать невозможно, за исключением совместного использования с
атрибутом selected;
// // value - данные выбранного пункта, которые отправятся на сервер, после отправки формы;

// С различными вариантами формы можно ознакомиться тут https://jsfiddle.net/r\_ageychik/0bmo597e/.
```

# Управление элементами формы

`<label>...</label>` позволяет привязать любое поле ввода с его меткой.

*// Делается это двумя способами:*

*// // привязать по id нужного элемента, с помощью атрибута for (и это самый распространенный вариант, где может понадобится id, использовать id для стилей не желательно!!!);*

```
<label for="username">
```

```
  <span>Name:</span>
```

```
</label>
```

```
<input type="text" id="username" name="user-name">
```

*// // либо поместить поле ввода и метку внутрь тега <label>;*

```
<label>
```

```
  <span>Name:</span>
```

```
  <input type="text" name="user-name">
```

```
</label>
```

*// В обоих случаях при клике на <span>Name:</span> текстовое поле получает фокус. При этом они могут находиться в разных частях макета. Особенно это удобно при использовании <input type="checkbox | radio"> совместно с псевдоселектором :checked*

# Группировка элементов формы

`<fieldset>...</fieldset>` - Предназначен для группировки элементов формы. Иногда это необходимо для упрощения дальнейшей работы с данными формы или для визуального разделения формы на логические части.

`<legend>...</legend>` - Используется для создания заголовка группы элементов формы, обозначенной тегом `<fieldset>`. Выводится на верхней границе блока `<fieldset>`, тем самым перекрывая его верхнюю рамку (`border`).

// Пример использования тегов можно увидеть [https://jsfiddle.net/r\\_ageychik/36sfwmpa/](https://jsfiddle.net/r_ageychik/36sfwmpa/).



# Псевдоклассы для работы с формами

```
// :checked - применяется для полей с type="checkbox | radio" задает стили при выбранной радио
кнопки или чекбокса;
form input[type="checkbox"]:checked .content {...}

// :focus - задает стили для элементов формы при получении фокуса (активный инпут);
textarea:not([type="button"]):focus,
input:not([type="button"]):focus {...}

// :required - задает стили для обязательных элементов формы;
// // Давайте попробуем придумать альтернативу этому псевдоселектору!!!
input:required {...}

// :optional - противоположность :required, задает стили для необязательных полей формы;
input:optional {...}

// :disabled - стили применяются для всех элементов формы, у которых есть атрибут disable;
input[type="button"]:disabled {...}

// :enabled - противоположность :disable, стили для доступных (не заблокированных) элементов формы;
input:enabled {...}
```