**Форма** – это компонент страницы для получения информации от пользователя. Для её создания используются теги ***form***.

Для группировки нескольких полей формы, связанных по смыслу, используются филдсеты с тегами ***fieldset***. Например, первый блок отведен под сбор контактной информации, а второй – под информацию о доставке. Рекомендуется в каждом филдсете указывать его описания для скринридера с помощью тегов ***legend***.

Для получения информации из полей ввода формы используются элементы с одиночным тегом ***input***.

Атрибут ***type*** позволяют задать логику для ввода данных с помощью элементов ***input*** и ***button***. Он может принимать значения *text*, *number*, *email*, *password*, *tel*, *url*, *color*, *range*, *date*, *datetime-local*, *month*, *week*, *time*, *radio*, *checkbox*, *button*, *submit*, *file*, *reset* и другие.

Атрибуты ***min***, ***max*** и ***step*** позволяют задать минимальное и максимальное значения для числового типа полей ввода данных, а также размер шага. По умолчанию они имеют значения от 0 до 100 с шагом 1.

Атрибуты ***minlength*** и ***maxlength*** позволяют задать минимальное и максимальное количество символов для текстового типа полей ввода данных.

Атрибут ***placeholder*** позволяет задавать подсказки в полях ввода.

Для создания кнопки в форме используются теги ***button***. Они работают аналогично ***<input type=”button” />***, но позволяет разместить контент, текст или изображение.

*<form>*

*<fieldset>*

*<legend>Ваши контактные данные</legend>*

*<label for="email" class="visually-hidden">Почта</label>*

*<input type="email" placeholder="me@example.com" id="email" />*

*<label for="password" class="visually-hidden">Пароль</label>*

*<input type="text" placeholder="Пароль" id="password" />*

*</fieldset>*

*<fieldset>*

*<legend>Время и адрес доставки</legend>*

*</fieldset>*

*<button class="button" type="submit">Войти</button>*

*</form>*

Атрибут ***name*** задает имя поля ввода, которые передаются на сервер.

Атрибут ***value*** хранит данные поля ввода, которые передаются на сервер.

*<form>*

*<label for="email">Почта</label>*

*<input type="email" name="email" id="email" />*

*<label for="password">Пароль</label>*

*<input type="password" name="password" id="password" />*

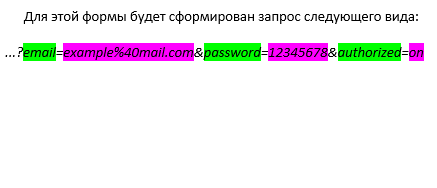
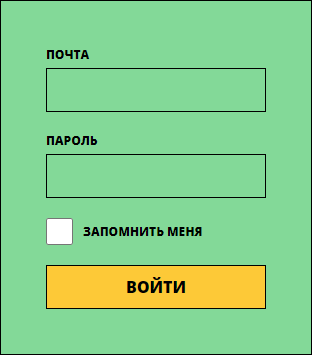
*<label for="authorized" class="agree">*

*<input type="checkbox" name="authorized" id="authorized" /> Запомнить меня*

*</label>*

*<button>Войти</button>*

*</form>*

**

Атрибут ***action*** для ***form*** отвечает за отправку данных формы на сервер или внешний сервис. Его значением могут быть:

* Пустое значение, заданное по умолчанию.
* Путь до скрипта.
* URL до сервиса, предоставляющего готовый скрипт (форм-процессорами). Например, [таким](http://formspark.io/).

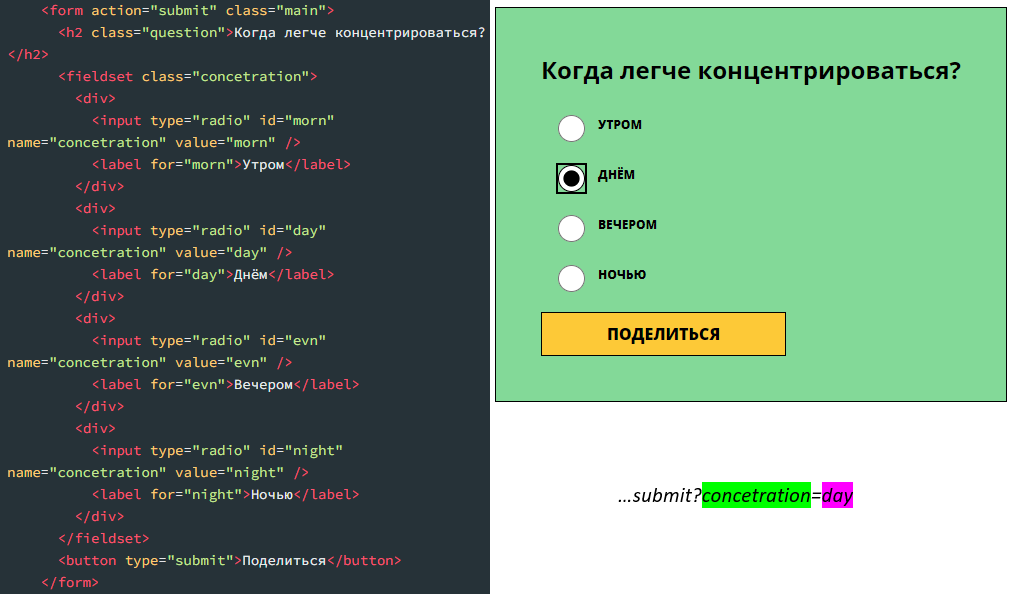
*<form action=""></form>*

Главным отличием значения ***radio*** от ***checkbox*** для атрибута ***type*** является отсутствие возможности множественного выбора. В первом случае, значение переменной ***name*** просто перезаписывается. Во втором же – дублируется при отправке запроса.

Атрибут **checked** внутри тега ***input*** позволяет отмечать дефолтные варианты при загрузке страницы.

Атрибут **required** внутри тега ***input*** позволяет отмечать обязательные к заполнению поля.

Атрибут **disabled** позволяет сделать одно или несколько полей недоступными для выбора.





Для ввода многострочного текста в форме лучше использовать теги ***textarea***. Размер шрифта влияет на размер поля.

Атрибут ***rows*** позволяет задать количество строк, которые помещаются в текстовом поле. По умолчанию равно 2.

Атрибут ***cols*** позволяет задать сколько символов поместится в каждой строке.

Свойство ***resize*** управляет возможностью изменения размера формы ввода текста. По умолчанию можно изменить размер, потянув за правый нижний угол. Свойства ***width*** и ***height*** при применении ***resize*** игнорируют ***rows*** и ***cols***.

*resize: none; // запретить любое изменение*

*resize: vertical; // запретить изменение по вертикали*

*resize: horizontal; // запретить изменение по горизонтали*

Для создания поля с выпадающим списком используются теги ***select***. Варианты же заключены между тегами ***option***. Для объединения опций в группы можно использовать тег ***optgroup***, но не стоит вкладывать один ***optgroup*** в другой.

*<select>*

*<optgroup label="Выберите тотем дня" id="totem">*

*<option value="Тигр из мадагаскара">Тигр из мадагаскара</option>*

*<option value="Бешеная каракатица">Бешеная каракатица</option>*

*</optgroup>*

*</select>*

С помощью тегов ***label*** (ярлык) мы сообщаем пользователю, что именно вводить в каждое поле. Для более тесной связки ярлыка и поля ввода используются атрибуты ***for*** и ***id***. Сами по себе поля ввода не имеют семантики, а ярлыки позволяют скринридерам озвучивать их. Ярлыки также можно скрывать с помощью *visually-hidden*. Например:

*<label class="name"> // первый вариант*

*<span>Имя</span>*

*<input type="text" />*

*</label>*

*<label for="surname">Фамилия</label> // второй вариант*

*<input type="text" id="surname" /> // позволяет расположить ярлык и поле в разных местах*

Поля ввода – это интерактивный элемент. Нельзя портить его фокусное состояние и ухудшать доступность. Не убирайте *outline*, а если меняете его, меняйте на контрастный.

Атрибут ***autocomplete*** позволяет управлять автозаполнением полей формы, основываясь на ***type*** полей ввода. По умолчанию включено настройками *html*, а также политиками браузеров. Он также позволяет задать значение автозаполнения для конкретного поля с тегами ***input***, ***select*** и ***textarea***.

*<form autocomplete="off">*

*<label for="email" class="visually-hidden">Почта</label>*

*<input*

*type="text"*

*placeholder="me@example.com"*

*id="email"*

*name="email"*

*autocomplete="email"*

*/>*

*</form>*

Атрибут ***inputmode*** отвечает за адаптацию виртуальной клавиатуры под тип данных. Он во многом пересекается с атрибутом ***type***. Например, для ***<input type="number">*** у ***inputmode*** есть значения ***numeric*** (только цифры) и ***decimal*** (цифры и разделитель дробных чисел, адаптированный под язык раскладки). Расширить набор значений можно с атрибутом ***pattern***. Подробнее можно почитать [тут](https://doka.guide/html/pattern/).

По умолчанию форма проверяет соответствие введенного в поле текста и его ожидаемым значением. Обычно вместо браузерной валидации используют проверку на *JavaScript*.

Например, тип *email* ожидает почту с собакой *@*. Выделить корректное и некорректное значение можно с помощью псевдоклассов ***:valid*** и ***:invalid***.

*input:invalid {*

*background-color: #fd24af;*

*}*

*form:valid {*

*outline: 2px solid #5f0dee;*

*}*

Например, для проверки числовых значений в диапазонах, указанных с помощью атрибутов min и max , есть псевдоклассы ***:in-range*** и ***:out-of-range***.

*input[type='number']:out-of-range {*

*background-color: #fd24af;*

*}*

*input[type='number']:in-range {*

*background-color: #80d994;*

*}*

Рассмотрим стили форм ввода, используемые по умолчанию:

* Поля ввода и ярлыки являются строчно-блочными элементами. Изменить это можно с помощью свойства ***display***.
* Текстовые стили и цвет вводимого в поле ввода текста не наследуется от родителя, поэтому всем полям ввода нужно задавать наследование явно с помощью специального значения ***inherit***.
* Полю ввода по умолчанию задан белый фон. Изменить это можно с помощью свойства ***background-color***.
* Полю ввода по умолчанию задана рамка. Изменить это можно с помощью свойства ***border***.
* Поле ввода всегда небольшое по ширине. Изменить это можно задав ***width: 100%*** от родителя.
* У поля ввода по умолчанию отсутствуют отступы и текст может прилипать к краям.
* По умолчанию поле ввода окружается рамкой, когда находится в фокусе. Изменить поведение можно, использовав псевдоклассы ***:focus*** и ***:focus-visible***.

*input {*

*font-family: inherit;*

*font-size: inherit;*

*color: inherit;*

*background-color: transparent;*

*padding: 10px 15px;*

*border: 1px solid #000;*

*}*

*input:focus {*

*outline: none;*

*}*

*input:focus-visible {*

*outline: 1px solid -webkit-focus-ring-color; // стандартный цвет браузера пользователя*

*}*

По умолчанию поля ввода, в которые нельзя ввести текст (***checkbox***, ***radio***, ***range***, ***select***), стилизируются каждым браузером по-своему. Для изменения стилизации чекбоксов и радиокнопки используется следующий порядок действий:

1. Пишем правильную разметку
2. Скрываем стандартный элемент
3. Рисуем на его месте новый элемент
4. Пишем сложносочиненный селектор, чтобы все работало

*<form>*

*<fieldset>*

*<label>*

*<input class=”visually-hidden” type="radio" name="answer" />*

*<span></span>*

*Да*

*</label>*

*<label>*

*<input class=”visually-hidden” type="radio" name="answer" />*

*<span></span>*

*Нет*

*</label>*

*</fieldset>*

*<fieldset>*

*<label>*

*<input class=”visually-hidden” type="checkbox" name="tv-show" />*

*<span></span>*

*Новости*

*</label>*

*<label>*

*<input class=”visually-hidden” type=" checkbox " name="tv-show" />*

*<span></span>*

*Фантастика*

*</label>*

*</fieldset>*

*</form>*

*.visually-hidden { // css-свойство для скрытия стандартного элемента*

*position: absolute;*

*clip: rect(0 0 0 0); // свойство clip только для старых браузеров*

*clip-path: inset(50%); // свойство clip-path превращает элемент в невидимую глазу точку*

*width: 1px;*

*height: 1px;*

*overflow: hidden;*

*white-space: nowrap;*

*}*

*label span { // рисуем новый элемент checkbox*

*width: 20px;*

*height: 20px;*

*background-color: #feafe8;*

*}*

*input[type='radio'] + span { // рисуем новый элемент radio*

*border-radius: 50%;*

*}*

*input[type='radio']:checked + span { // стилизуем новый элемент radio в статусе checked*

*border: 3px solid #fff;*

*}*

*input[type='checkbox']:checked + span { // стилизуем новый элемент checkbox в статусе checked*

*background-image: url(./images/check.svg);*

*background-repeat: no-repeat;*

*background-size: contain;*

*background-position: center;*

*}*

*input[type='radio']:focus-visible + span, // стилизуем оба новых элемента в статусе focus-visible*

*input[type='checkbox']:focus-visible + span {*

*box-shadow: 0 0 0 5px rgb(254 175 232 / 0.5);*

*}*



Рассмотрим реализацию стилизации выпадающего списка с помощью *html* и *css*:

*select {*

*// заставляем браузер забыть стандартные стили этого элемента*

*-webkit-appearance: none;*

*-moz-appearance: none;*

*appearance: none;*

*// задаем свои стили*

*padding: 10px 15px;*

*border: 1px solid #000;*

*border-radius: 0;*

*background-color: transparent;*

*font-family: inherit;*

*font-size: 15px;*

*font-weight: 800;*

*text-transform: uppercase;*

*// добавим кнопку со стрелочкой*

*background-image: url(../images/up-down.svg);*

*background-size: 25px;*

*background-repeat: no-repeat;*

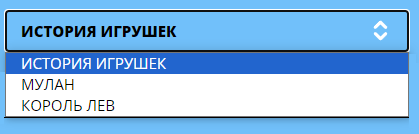
*background-position: center right 15px;*

*}*

*select:focus-visible { // стилизуем элемент в статусе focus-visible*

*box-shadow: 0 0 0 5px rgb(254 175 232 / 0.5);*

*}*



Рассмотрим реализацию стилизации элемента *range*:

*input[type="range"] {*

*// заставляем браузер забыть стандартные стили этого элемента*

*-webkit-appearance: none;*

*-moz-appearance: none;*

*appearance: none;*

*// задаем свои стили*

*width: 100%;*

*background-color: transparent;*

*}*

*input[type="range"]:focus { // стилизуем элемент в фокусе*

*outline: 2px solid #000;*

*outline-offset: 2px;*

*}*

*// задаем стилизацию ползунка для браузеров на Chromium*

*input[type="range"]::-webkit-slider-thumb {*

*// повторно сбрасываем стили*

*-webkit-appearance: none;*

*-moz-appearance: none;*

*appearance: none;*

*// задаем свои стили*

*border: 1px solid #000;*

*height: 16px;*

*width: 26px;*

*border-radius: 0;*

*background-color: #fff;*

*cursor: pointer; // курсор в виде руки*

*margin-top: -3px;*

*}*

*// задаем стилизацию полоски для браузеров на Chromium*

*input[type="range"]::-webkit-slider-runnable-track {*

*width: 100%;*

*height: 10px;*

*cursor: pointer;*

*background-color: #000;*

*border-radius: 0;*

*border: none;*

*}*

*// задаем стилизацию ползунка для Firefox*

*input[type="range"]::-moz-range-thumb {*

*box-sizing: border-box;*

*border: 1px solid #000;*

*height: 16px;*

*width: 26px;*

*border-radius: 0;*

*background-color: #fff;*

*cursor: pointer; // курсор в виде руки*

*}*

*// задаем стилизацию полоски для Firefox*

*input[type="range"]::-moz-range-track {*

*width: 100%;*

*height: 10px;*

*cursor: pointer;*

*background-color: #000;*

*border-radius: 0;*

*border: none;*

*}*

