



ФОРМЫ



СЕМЕН БОЙКО



СЕМЕН БОЙКО

Front-end разработчик



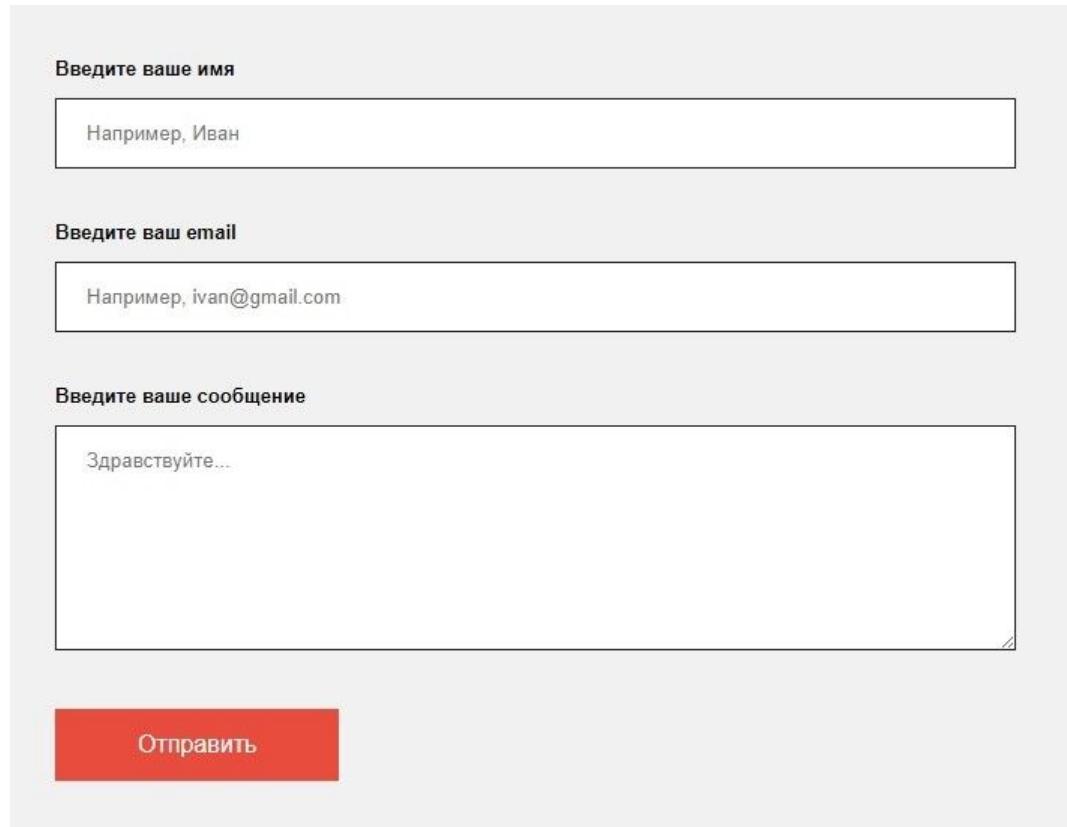
simonderus@gmail.com



fb.me/simonboycko

НАША ЗАДАЧА

Сегодня нам выпала довольно интересная задача. Мы с вами сверстаем форму отправки сообщения.



A light gray rectangular form with three input fields and a red button at the bottom. The first field is for name, the second for email, and the third for message content. Each field has a placeholder text example.

Ведите ваше имя

Например, Иван

Ведите ваш email

Например, ivan@gmail.com

Ведите ваше сообщение

Здравствуйте...

Отправить

ЭЛЕМЕНТ `input`

Элемент `input` позволяет создавать различные элементы формы. Для этого у элемента существует атрибут `type`.

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

- `type="text"` – элемент `input` становится полем для ввода текста;
- `type="password"` – элемент `input` становится текстовым полем ввода для пароля.

Привет Нетология!

СОЗДАЕМ РАЗМЕТКУ

```
1 <div class="form-row">
2   <span class="form-hint">Введите ваше имя</span>
3   <input type="text" class="field">
4 </div>
5 <div class="form-row">
6   <span class="form-hint">Введите ваш email</span>
7   <input type="text" class="field">
8 </div>
```

Введите ваше имя

Введите ваш email

СТИЛИЗУЕМ ПОДСКАЗКИ

```
1 .form-hint {  
2   display: block;  
3   font-size: 13px;  
4   font-weight: 700;  
5   margin-bottom: 12px;  
6 }
```

Ведите ваше имя

Ведите ваш email

СТИЛИЗУЕМ ПОЛЯ

```
1 .field {  
2     box-sizing: border-box;  
3     width: 640px;  
4     border: 1px solid #222222;  
5     padding: 15px 20px;  
6 }
```

Ведите ваше имя

Ведите ваш email

[Демо](#)

Атрибут `placeholder`

Атрибут `placeholder` предназначен для вывода текста внутри поля формы, чтобы помочь пользователю ввести верные данные.

```
1 <div class="form-row">
2   <span class="form-hint">Введите ваше имя</span>
3   <input type="text" class="field" placeholder="Например, Иван">
4 </div>
5 <div class="form-row">
6   <span class="form-hint">Введите ваш email</span>
7   <input type="text" class="field" placeholder="Например, ivan@gmail.com">
8 </div>
```

Введите ваше имя

Например, Иван

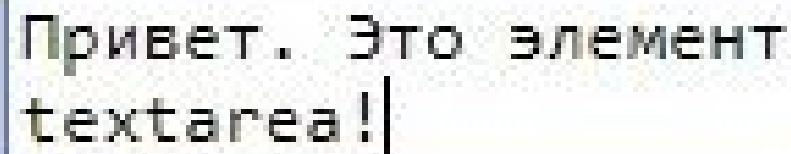
Введите ваш email

Например, ivan@gmail.com

ЭЛЕМЕНТ `textarea`

Элемент `textarea` представляет собой элемент формы для создания области, в которую можно вводить несколько строк текста.

```
<textarea></textarea>
```



Привет. Это элемент
textarea!

ДОБАВЛЯЕМ textarea

```
1 <div class="form-row">
2   <span class="form-hint">Введите ваше сообщение</span>
3   <textarea class="field" placeholder="Здравствуйте..."></textarea>
4 </div>
```

Ведите ваше имя

Например, Иван

Ведите ваш email

Например, ivan@gmail.com

Ведите ваше сообщение

Здравствуйте...

[Демо](#)

СВОЙСТВА `min-width`, `max-width`, `min-height` И `max-height`

В CSS есть свойства, которые контролируют диапазон изменения ширины и высоты контентной области, т.е свойств `width` и `height`.

- `min-width` – задает минимальное значение, которое может принимать свойство `width`;
- `max-width` – задает максимальное значение, которое может принимать свойство `width`;
- `min-height` – задает минимальное значение, которое может принимать свойство `height`;
- `max-height` – задает максимальное значение, которое может принимать свойство `height`.

width (height) МЕНЬШЕ ЧЕМ min-width (min-height)

```
1 .box {  
2     width: 700px;  
3     min-width: 1000px;  
4 }
```

Ширина контентной области элемента будет взята из свойства `min-width`, т.е `1000px`. Тот же самый алгоритм сработает в случае высоты контентной области.

```
1 .box {  
2     height: 700px;  
3     min-height: 1000px;  
4 }
```

width (height) РАВНЯЕТСЯ min-width (min-height)

```
1 .box {  
2     width: 800px;  
3     min-width: 800px;  
4 }
```

Браузер возьмет значение из свойства width , т.е 800px . Тот же самый алгоритм сработает в случае высоты контентной области.

```
1 .box {  
2     height: 800px;  
3     min-height: 800px;  
4 }
```

width (height) БОЛЬШЕ ЧЕМ min-width (min-height)

```
1 .box {  
2     width: 700px;  
3     min-width: 500px;  
4 }
```

Ширина контентной области будет равняться значению из свойства `width`, т.е `700px`. По аналогии с шириной контентной области, высота будет рассчитываться также.

```
1 .box {  
2     height: 700px;  
3     min-height: 500px;  
4 }
```

width (height) МЕНЬШЕ ЧЕМ max-width (max-height)

```
1 .box {  
2     width: 700px;  
3     max-width: 1000px;  
4 }
```

Ширина контентной области элемента будет взята из свойства `width`, т.е `700px`. Тот же самый алгоритм сработает в случае высоты контентной области.

```
1 .box {  
2     height: 700px;  
3     max-height: 1000px;  
4 }
```

width (height) РАВНЯЕТСЯ max-width (max-height)

```
1 .box {  
2     width: 800px;  
3     max-width: 800px;  
4 }
```

Браузер возьмет значение из свойства `width`, т.е `800px`. Тот же самый алгоритм сработает в случае высоты контентной области.

```
1 .box {  
2     height: 800px;  
3     max-height: 800px;  
4 }
```

width (height) БОЛЬШЕ ЧЕМ max-width (max-height)

```
1 .box {  
2     width: 700px;  
3     max-width: 500px;  
4 }
```

Ширина контентной области будет равняться значению из свойства `max-width`, т.е `500px`. По аналогии с шириной контентной области, высота будет рассчитываться также

```
1 .box {  
2     height: 700px;  
3     max-height: 500px;  
4 }
```

РАЗМЕРЫ `textarea`

Сначала добавим отдельный класс `textarea` для элемента.

```
1 | <textarea class="field textarea" placeholder="Здравствуйте..."></textarea>
```

Далее стилизуем элемент

```
1 | .textarea {  
2 |   min-width: 640px;  
3 |   max-width: 640px;  
4 |   min-height: 150px;  
5 |   max-height: 200px;  
6 | }
```

РЕЗУЛЬТАТ

Введите ваше имя

Например, Иван

Введите ваш email

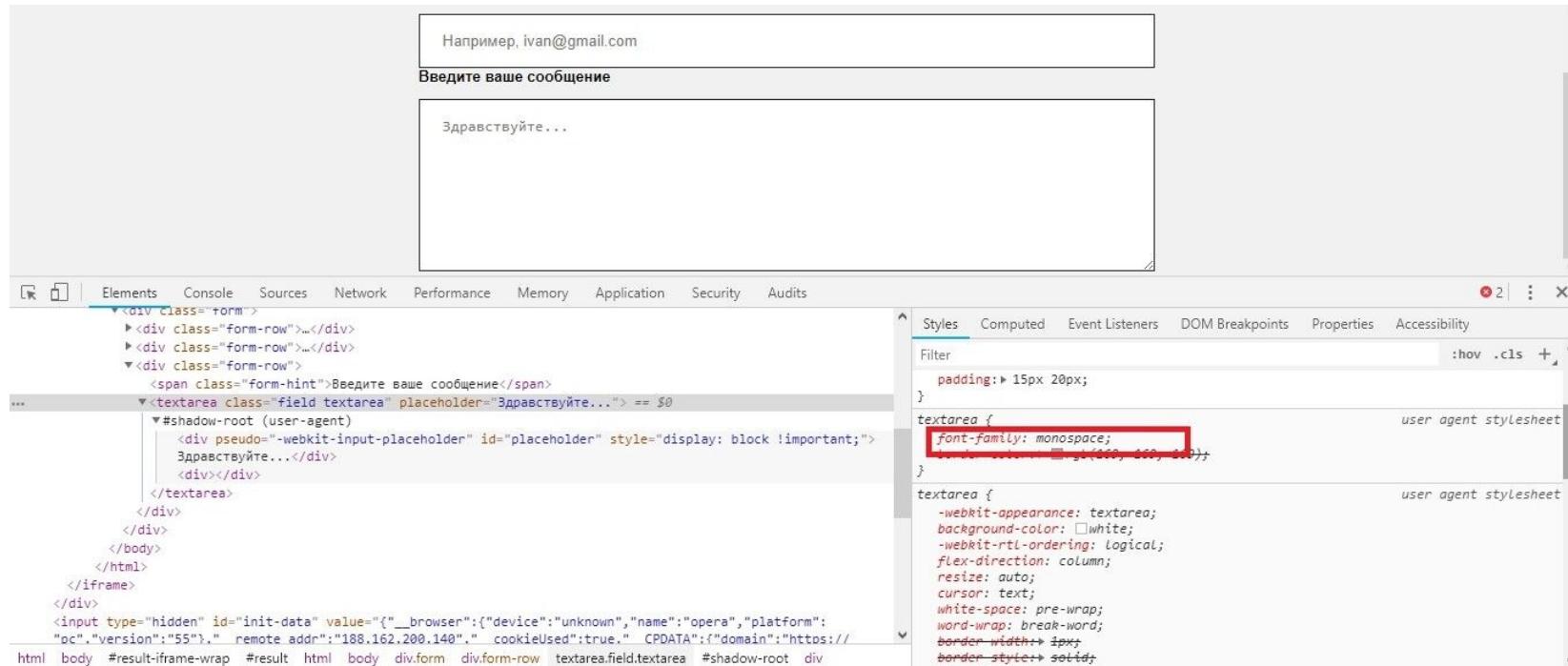
Например, ivan@gmail.com

Введите ваше сообщение

Здравствуйте...

[Демо](#)

textarea ИМЕЕТ СВОЙСТВО font-family ПО-УМОЛЧАНИЮ

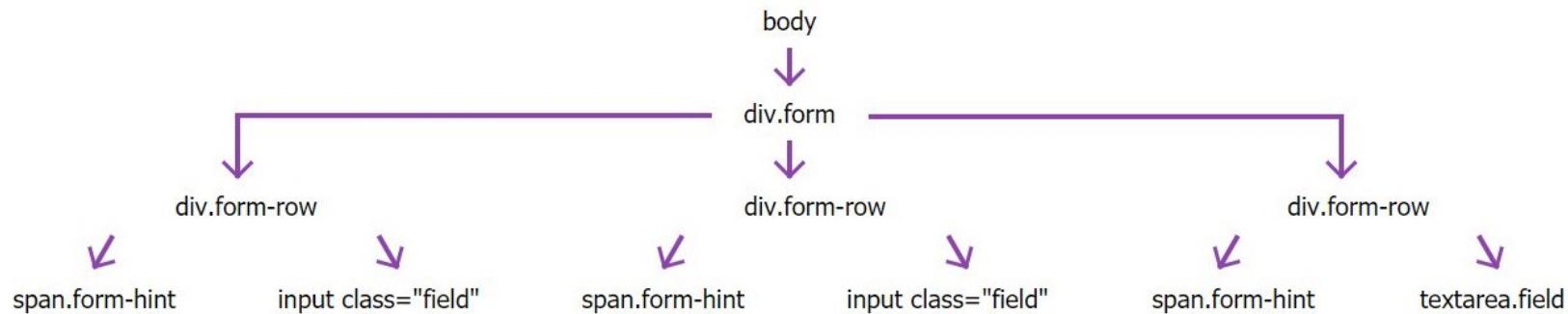


ЗНАЧЕНИЕ `inherit`

Сейчас у нас есть следующий HTML

```
1 <body>
2   <div class="form">
3     <div class="form-row">
4       <!-- поле для ввода имени --&gt;
5     &lt;/div&gt;
6     &lt;div class="form-row"&gt;
7       <!-- поле для ввода email --&gt;
8     &lt;/div&gt;
9     &lt;div class="form-row"&gt;
10      <!-- поле для ввода сообщения --&gt;
11    &lt;/div&gt;
12  &lt;/div&gt;
13 &lt;/body&gt;</pre>
```

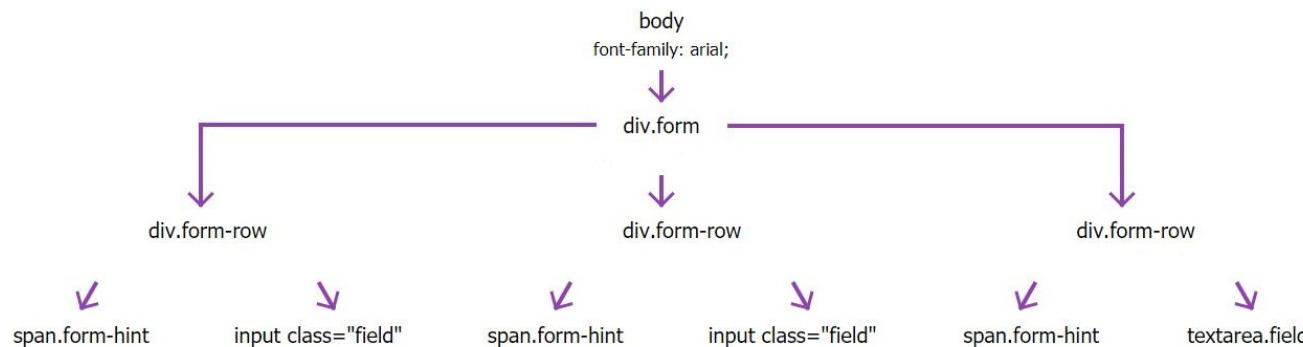
ДЕРЕВО ЭЛЕМЕНТОВ



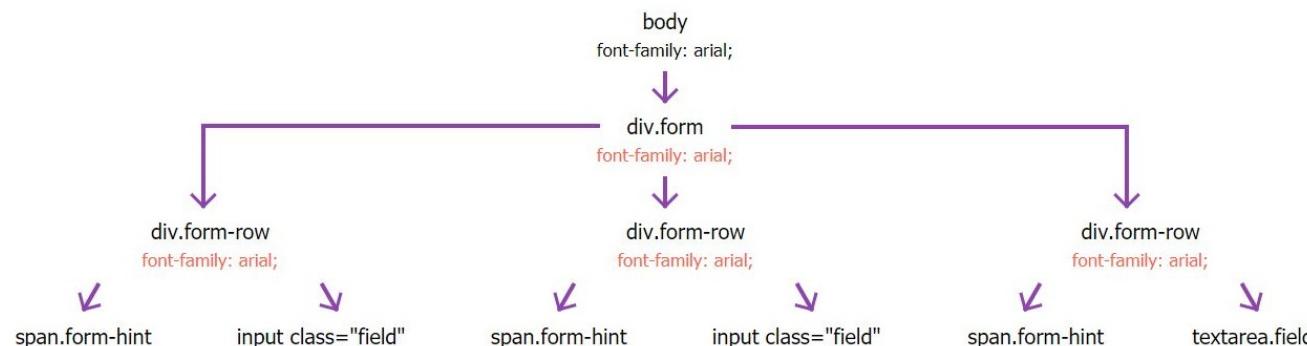
У элемента `span` с классом `form-hint` есть один родитель – это элемент `div` с классом `form-row`.

ДЕРЕВО ЭЛЕМЕНТОВ

Мы с вами задали свойство `font-family` для `body`.

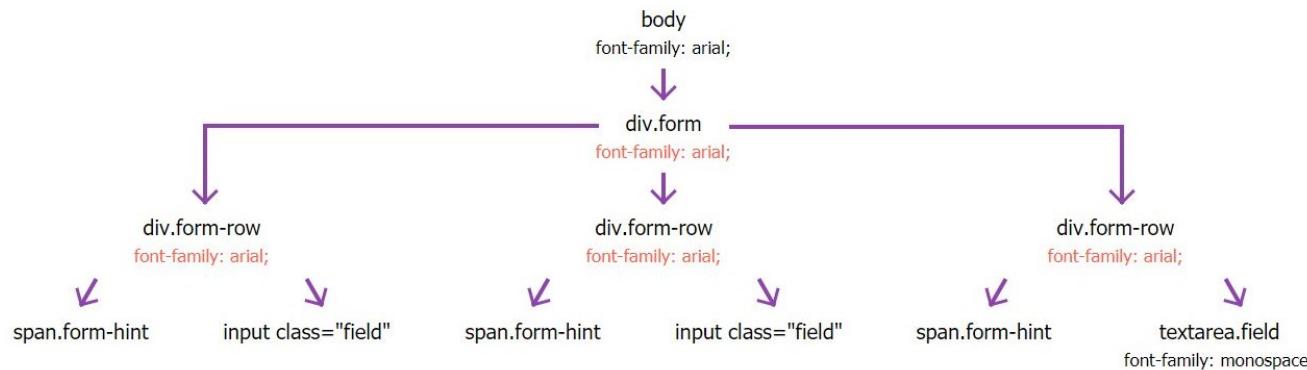


И это значение применяется не только для самого элемента, но и для его детей и потомков.

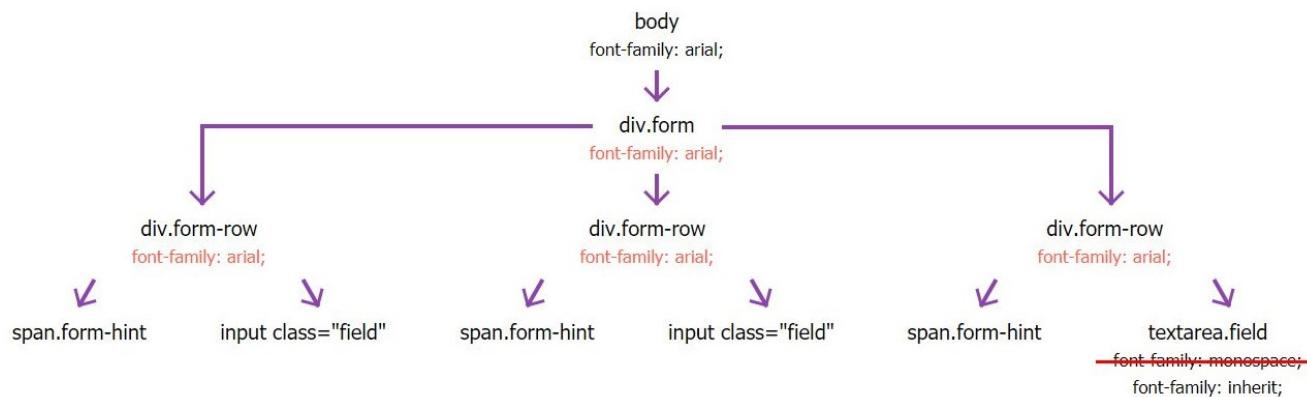


НАСЛЕДОВАНИЕ СТИЛЯ

У браузера есть собственные стили, и они также вносят корректиды.



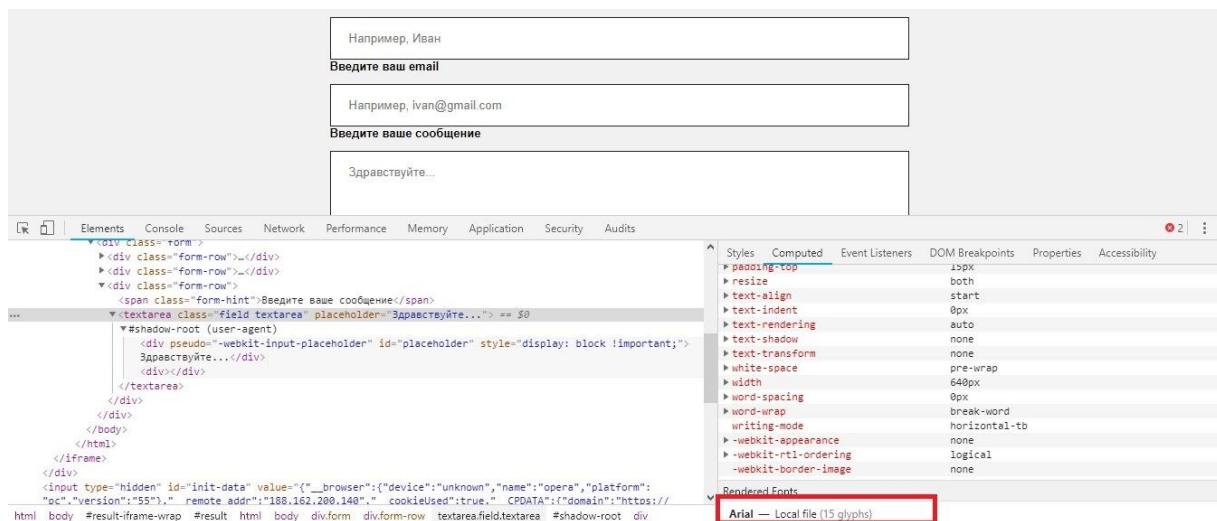
В CSS есть возможность сказать браузеру, что ему нужно наследовать значение свойства от родителей.



ДОБАВИМ inherit для font-family

```

1 .field {
2   box-sizing: border-box;
3   width: 640px;
4   border: 1px solid #222222;
5   padding: 15px 20px;
6   font-family: inherit;
7 }
```



ЭЛЕМЕНТ `button`

Элемент `button` представляет собой элемент управления, позволяющий пользователю запускать определенные действия.

```
<button type="submit">Отправить</button>
```

Типы кнопок:

- `submit` - кнопка отправки формы. Также это значение по умолчанию;
- `reset` - кнопка, которая очищает введенные значения в форме;
- `button` - кнопка, которая не запускает никакого действия.

ДОБАВЛЯЕМ КНОПКУ

```
1 <div class="form-row">  
2   <button class="button">Отправить</button>  
3 </div>
```

И стилизуем ее:

```
1 .button {  
2   border: none;  
3   padding: 15px 55px;  
4   color: #ffffff;  
5   background-color: #e74c3c;  
6   font-family: inherit;  
7   font-size: 16px;  
8 }
```

РЕЗУЛЬТАТ

Введите ваше имя

Например, Иван

Введите ваш email

Например, ivan@gmail.com

Введите ваше сообщение

Здравствуйте...

Отправить

СВОЙСТВО cursor

CSS Custom Cursors

I auto	↗ move	↙ no-drop	↔ col-resize
↔ all-scroll	☞ pointer	🚫 not-allowed	↕ row-resize
+ crosshair	➡ progress	↔ e-resize	↗ ne-resize
↖ default	I text	↑ n-resize	↖ nw-resize
↖? help	▬ vertical-text	↓ s-resize	↖ se-resize
I inherit	⌚ wait	↔ w-resize	↗ sw-resize

Существуют следующие значения:

- default
- none
- text
- pointer

ДОБАВИМ cursor СО ЗНАЧЕНИЕМ pointer

```
1 .button {  
2     border: none;  
3     padding: 15px 55px;  
4     color: #ffffff;  
5     background-color: #e74c3c;  
6     font-family: inherit;  
7     font-size: 16px;  
8     cursor: pointer;  
9 }
```

[Демо](#)

ПОСМОТРИМ НА СТРУКТУРУ

```
1 <div class="form">
2   <div class="form-row">
3     <!-- поле для ввода имени -->
4   </div>
5   <div class="form-row">
6     <!-- поле для ввода email -->
7   </div>
8   <div class="form-row">
9     <!-- поле для ввода сообщения -->
10  </div>
11  <div class="form-row">
12    <!-- кнопка -->
13  </div>
14 </div>
```

ОДИНАКОВЫЙ ОТСТУП МЕЖДУ ЭЛЕМЕНТАМИ

The diagram shows a vertical stack of four form elements, each enclosed in a red-bordered box. A red double-headed arrow labeled '35px' indicates the uniform vertical distance between the top of the first element and the top of the second, between the second and the third, and between the third and the fourth. The elements are:

- Ведите ваше имя**
Например, Иван
- Ведите ваш email**
Например, ivan@gmail.com
- Ведите ваше сообщение**
Здравствуйте...
- Отправить**

СТИЛИЗУЕМ .form-row

```
1 | .form-row:nth-child(n+2) {  
2 |   margin-top: 35px;  
3 | }
```

Ведите ваше имя

Например, Иван

Ведите ваш email

Например, ivan@gmail.com

Ведите ваше сообщение

Здравствуйте...

Отправить

ПОПРОБУЕМ ВВЕСТИ ДАННЫЕ

Введите ваше имя

Иван

Введите ваш email

ivan@gmail.com

Введите ваше сообщение

Привет!

Отправить

Данные не отправляются.

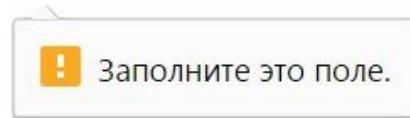
ЭЛЕМЕНТ `form`

```
1 <form class="form">
2   <div class="form-row">
3     <!-- поле для ввода имени --&gt;
4   &lt;/div&gt;
5   &lt;div class="form-row"&gt;
6     <!-- поле для ввода email --&gt;
7   &lt;/div&gt;
8   &lt;div class="form-row"&gt;
9     <!-- поле для ввода сообщения --&gt;
10  &lt;/div&gt;
11  &lt;div class="form-row"&gt;
12    <!-- кнопка --&gt;
13  &lt;/div&gt;
14 &lt;/form&gt;</pre>
```

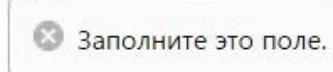
Атрибут `required`

Атрибут `required` устанавливает поле формы обязательным для заполнения перед отправкой данных.

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



Google Chrome



Opera



Firefox



Internet Explorer

ДОБАВИМ required ДЛЯ ПОЛЕЙ ФОРМЫ

```
1 <div class="form-row">
2   <span class="form-hint">Введите ваше имя</span>
3   <input type="text" class="field" placeholder="Например, Иван" required>
4 </div>
5 <div class="form-row">
6   <span class="form-hint">Введите ваш email</span>
7   <input type="text" class="field" placeholder="Например, ivan@gmail.com" required>
8 </div>
9 <div class="form-row">
10  <span class="form-hint">Введите ваше сообщение</span>
11   <textarea class="field textarea" placeholder="Здравствуйте..." required></textarea>
12 </div>
```

[Демо](#)

РЕЗУЛЬТАТ

Ведите ваше имя

Ведите ваш email

Заполните это поле.

Ведите ваше сообщение

Отправить

ЗНАЧЕНИЯ ДЛЯ АТРИБУТА `type`

- `type="email"` - браузер будет ждать, что пользователь введет email;
- `type="search"` - элемент становится полем ввода поискового запроса.

ИЗМЕНИМ АТРИБУТ `type`

```
<input type="email" class="field" placeholder="Например, ivan@gmail.com" required>
```

Ведите ваше имя

Ведите ваш email

! Адрес электронной почты должен содержать символ "@". В адресе "Это не электронный адрес" отсутствует символ "@".

Отправить

ЭЛЕМЕНТ `label`

Элемент `label` позволяет разметить текстовое описание элемента формы, которое связано с элементом `input`.

Связать `label` и `input` можно двумя способами:

- Вложить `input` в элемент `label`:

```
1 <label>  
2   Введите текст  
3   <input type="text">  
4 </label>
```

- Использовать атрибут `for` для элемента `label` и `id` у элемента `input`:

```
1 <label for="field-one">Введите текст</label>  
2 <input id="field-one" type="text">
```

ДОБАВИМ LABEL

```
1 <div class="form-row">
2   <label class="form-group">
3     <span class="form-hint">Введите ваше имя</span>
4     <input type="text" class="field" placeholder="Например, Иван" required>
5   </label>
6 </div>
7 <div class="form-row">
8   <label class="form-group">
9     <span class="form-hint">Введите ваш email</span>
10    <input type="email" class="field" placeholder="Например, ivan@gmail.com" required>
11  </label>
12 </div>
13 <div class="form-row">
14   <label class="form-group">
15     <span class="form-hint">Введите ваше сообщение</span>
16     <textarea class="field textarea" placeholder="Здравствуйте..." required></textarea>
17   </label>
18 </div>
```

[Демо](#)

СТИЛИЗУЕМ label

```
1 .form-group {  
2   display: block;  
3 }
```

The screenshot shows a browser's developer tools with the "Elements" tab selected. A form is displayed with several input fields and labels. The top input field has the placeholder "Например, Иван". Below it is a blue input field with the placeholder "Ведите ваш email". Another blue input field below that has the placeholder "Например, ivan@gmail.com". At the bottom, there is a text area with the placeholder "Ведите ваше сообщение" containing the text "Здравствуйте...". The developer tools sidebar on the right shows the DOM tree and the CSS styles applied to the elements. The class ".form-group" is defined with the rule "display: block;". The class "label" is defined with the rule "cursor: default;". The "Properties" tab in the developer tools shows the style "BOEYro:34" for the font size.

```
<html lang="en">
  <head>...</head>
  <body translate="no">
    <form class="form">
      <div class="form-row">...</div>
      <div class="form-row">
        <label class="form-group"> == $0
          <span class="form-hint">Ведите ваш email</span>
          <input type="email" class="field" placeholder="Например, ivan@gmail.com" required>...
        </label>
      </div>
    </div class="form-row">
  </form>
</body>
</html>
```

Styles

```
element.style {  
}  
.form-group {  
  display: block;  
}  
label {  
  cursor: default;  
}  
Inherited from body  
body {  
  BOEYro:16  
}
```

Демо

СИНЯЯ ОБВОДКА

Введите ваше имя

Например, Иван

СОСТОЯНИЕ :focus

Когда мы нажимаем клавишу Tab, браузер начинает переключаться между активными (интерактивными) элементами, которые пользователь может использовать. Когда браузер перешел к такому элементу и готов с ним взаимодействовать, то это называют состоянием фокуса.

Username:

A text input field containing the value "Kevin".

Password:

[Forgot your password?](#)

A password input field, currently empty, used for entering a password.

СВОЙСТВО `outline`

Свойство `outline` задает обводку вокруг элемента. Оно состоит из свойств:

- `outline-width`;
- `outline-style`;
- `outline-color`.

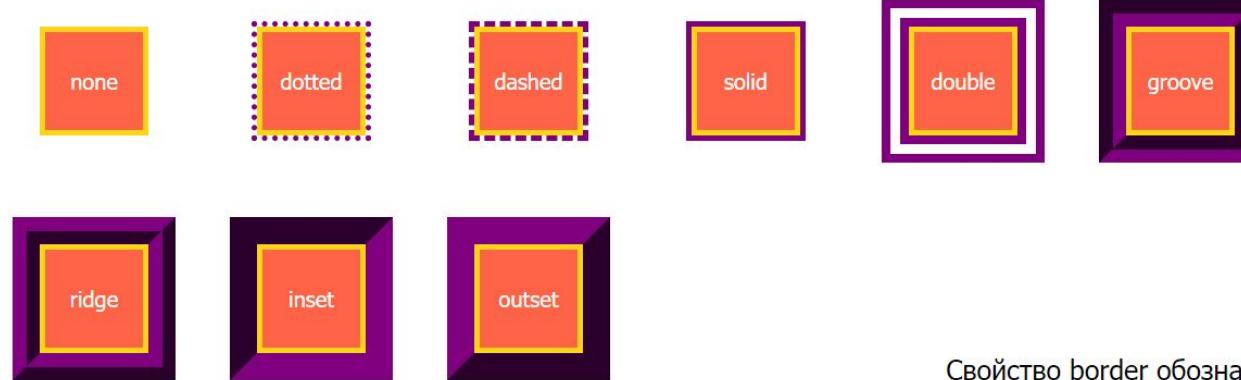
outline-width



Свойство `border` обозначено желтым

Свойство `outline-width` задает ширину обводки, которая задается в `px`.

outline-style



Свойство border обозначено желтым

Свойство `outline-style` задает вид обводки.

Значения свойства:

- `none`
- `dotted`
- `dashed`
- `solid`
- `double`
- `groove`
- `ridge`
- `inset`
- `outset`

outline-color



```
outline-color:  
#edaded
```

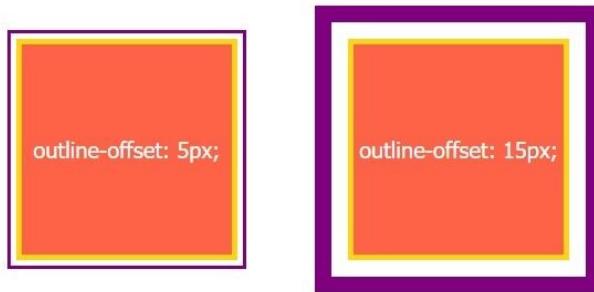


```
outline-color:  
#7e19ce
```

Свойство `border` обозначено желтым

Свойство `outline-color` задает цвет обводки. В качестве значения можно использовать шестнадцатеричный формат цвета.

СВОЙСТВО outline-offset



Свойство border обозначено желтым

Можно задать расстояние между обводкой `outline` и элементом при помощи свойства `outline-offset`. Значение для свойства указываются в `px`.

```
1 .box {  
2   outline: 15px solid #800080;  
3   outline-offset: 15px;  
4 }
```

СТИЛИЗУЕМ :focus

```
1 .field:focus {  
2     outline: 2px solid #c53b2d;  
3     outline-offset: 5px;  
4 }  
5  
6 .button:focus {  
7     outline: 2px solid #c53b2d;  
8     outline-offset: 5px;  
9 }
```

Введите ваш email

Например, ivan@gmail.com

Отправить

[Демо](#)

СПИСОК СЕЛЕКТОРОВ

В CSS есть возможность объединять повторяющиеся правила. Например:

```
1 | h1 { font-family: sans-serif; }  
2 | h2 { font-family: sans-serif; }  
3 | h3 { font-family: sans-serif; }
```

Этот код можно переписать, создав список селекторов:

```
h1, h2, h3 { font-family: sans-serif; }
```

ВЕРНЕМСЯ К НАШЕМУ КОДУ

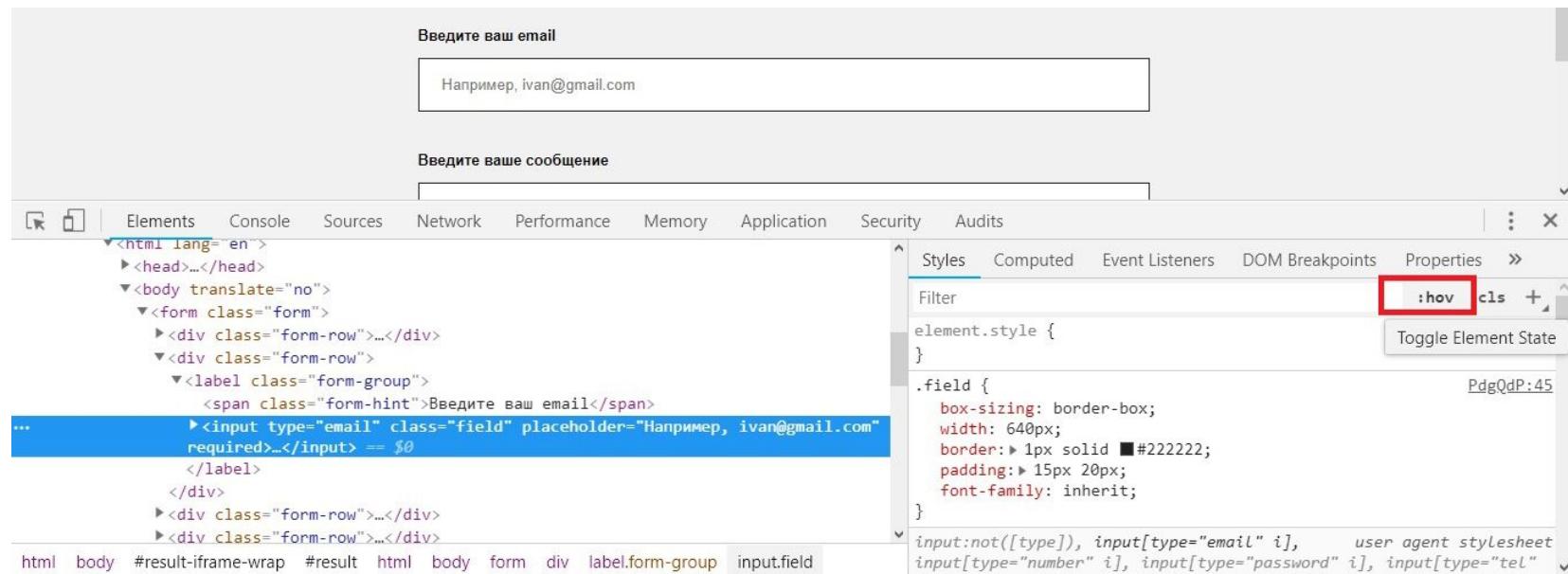
```
1 .field:focus {  
2     outline: 2px solid #c53b2d;  
3     outline-offset: 5px;  
4 }  
5  
6 .button:focus {  
7     outline: 2px solid #c53b2d;  
8     outline-offset: 5px;  
9 }
```

Объединим в список:

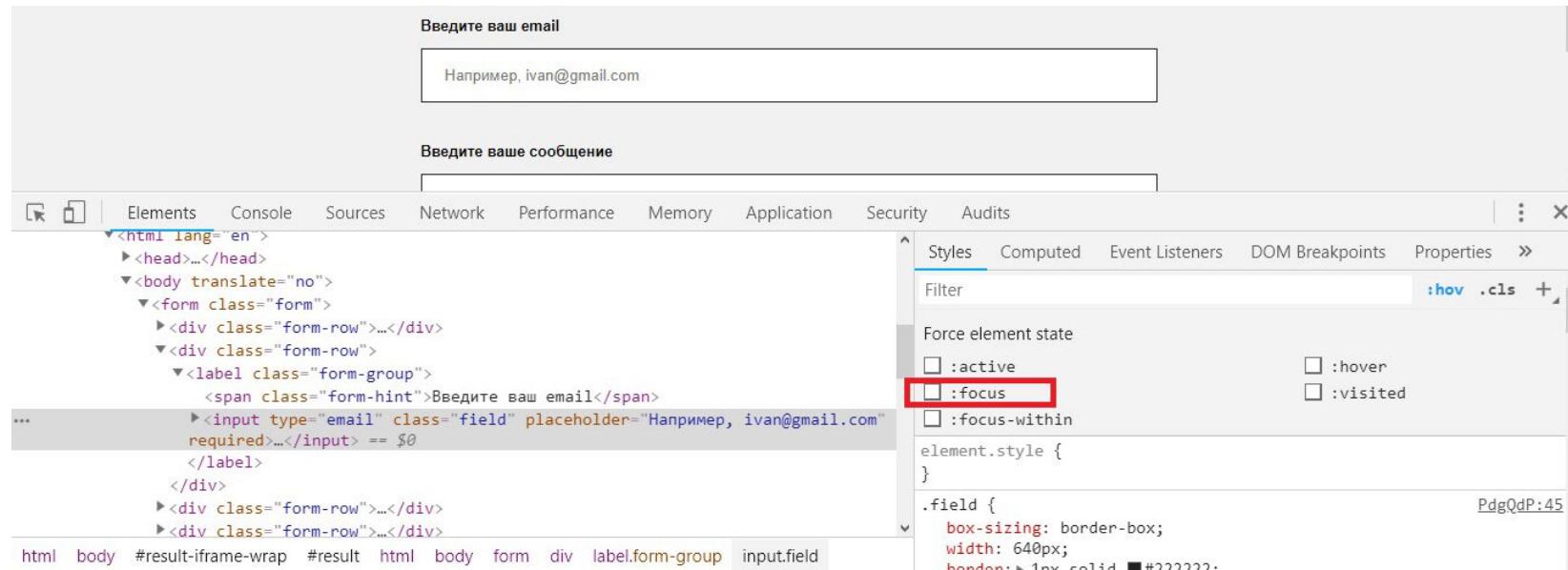
```
1 .field:focus, .button:focus {  
2     outline: 2px solid #c53b2d;  
3     outline-offset: 5px;  
4 }
```

ПРОВЕРЯЕМ В ИНСПЕКТОРЕ

Откроем инспектор и нажмем на надпись «:hover»



ВЫБИРАЕМ ПСЕВДОКЛАСС :focus



ПРОВЕРЯЕМ СТИЛИ

The screenshot shows the Chrome DevTools interface. The left pane displays the HTML structure of a form. The right pane shows the Styles tab with the following CSS rules:

```
.field:focus, .button:focus {  
    outline: 2px solid #c53b2d;  
    outline-offset: 5px;  
}  
  
.field {  
    box-sizing: border-box;  
    width: 640px;  
}
```

The input field in the browser's developer mode has a red border, indicating it is focused. A red box highlights the CSS rule for the focused state of the input field.

СОСТОЯНИЕ :hover

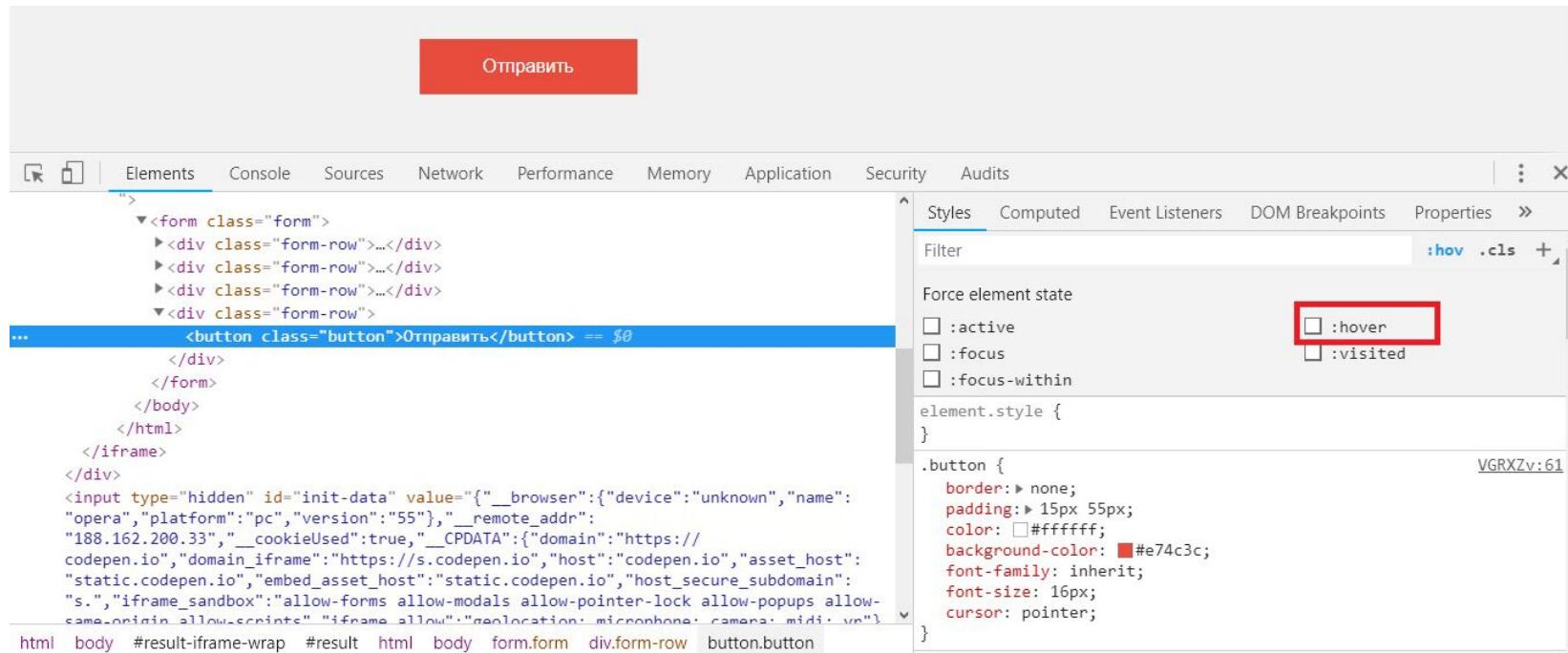
Браузер дает возможность задать свойства, которые будут применяться только при наведении на элемент. Эту возможность называют псевдоклассом `:hover`.

```
1 .button:hover {  
2     background-color: #dddddd;  
3 }  
4 a:hover {  
5     color: #dddddd;  
6 }
```

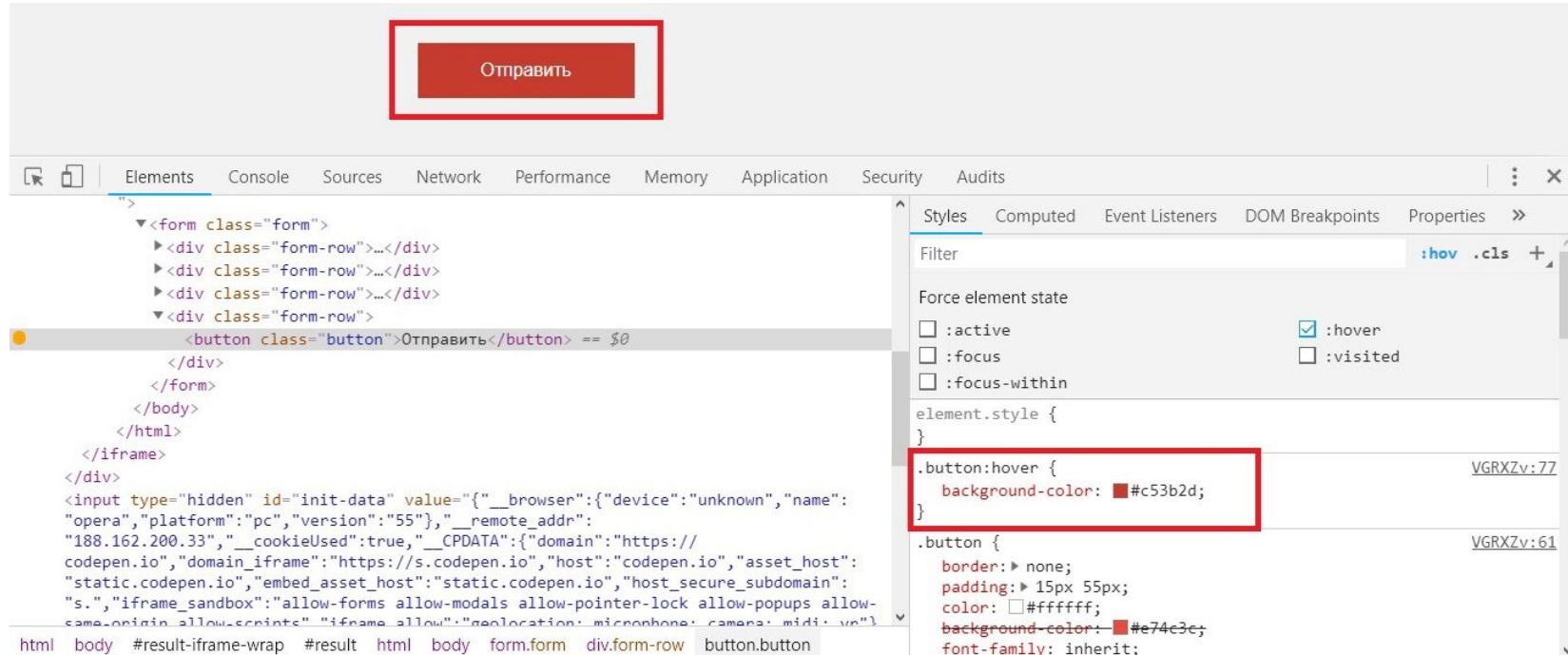
Добавим смену цвета кнопки при наведении мыши:

```
1 .button:hover {  
2     background-color: #c53b2d;  
3 }
```

ВЫБИРАЕМ ПСЕВДОКЛАСС :hover



ПРОВЕРЯЕМ СТИЛИ



ИТОГИ

- Элемент `input` позволяет создавать различные элементы формы, благодаря различным значениям атрибута `type`.
- Атрибут `placeholder` предназначен для вывода текста внутри поля формы.
- Элемент `textarea` предназначен для создания области, в которую можно вводить несколько строк текста.
- Свойства `min-width`, `max-width`, `min-height` и `max-height` контролируют диапазон изменения ширины и высоты контентной области.
- Элемент `button` предназначен для создания элемента управления, позволяющего пользователю запускать определенные действия.

ИТОГИ

- Элемент `form` дает возможность браузеру отправить введенные данные пользователя на сервер.
- Элемент `label` позволяет разметить текстовое описание элемента формы.
- Свойство `outline` задает обводку вокруг элемента.
- С помощью псевдокласса `:hover` можно задать свойства, которые будут применяться только при наведении мышкой на элемент.
- С помощью псевдокласса `:focus` можно задать свойства, которые будут применяться только когда браузер установил фокус на элементе.



Ваши вопросы?

СЕМЕН БОЙКО



simonderus@gmail.com



fb.me/simonboycko