

Fullstack-разработка

JavaScript

События

Поля ввода

Цели урока:

1. Научимся обрабатывать ввод пользователя на лету
2. Создать генератор аватарки

Разбор ДЗ

Что получилось?

Что было трудно?

Теория

Событие input

Событие input

Это событие срабатывает, когда пользователь вводит
НОВЫЙ СИМВОЛ В ТЕКСТОВОЕ ПОЛЕ ВВОДА

Пост

Напишите сообщение

0/140

Событие input

Это событие срабатывает, когда пользователь вводит
НОВЫЙ СИМВОЛ В ТЕКСТОВОЕ ПОЛЕ ВВОДА

```
textNode.addEventListener(`input`,  
function () {  
    console.log(textNode.value);  
});
```

Событие input

Чтобы сделать обработчик события более **универсальным**, сегодня мы будем использовать **evt.target**

```
textNode.addEventListener(`input`,  
function (evt) {  
    console.log(evt.target.value);  
});
```

Полезный input

`type="text"`

Текст

`type="number"`

Число

`type="password"`

Пароль

`type="email"`

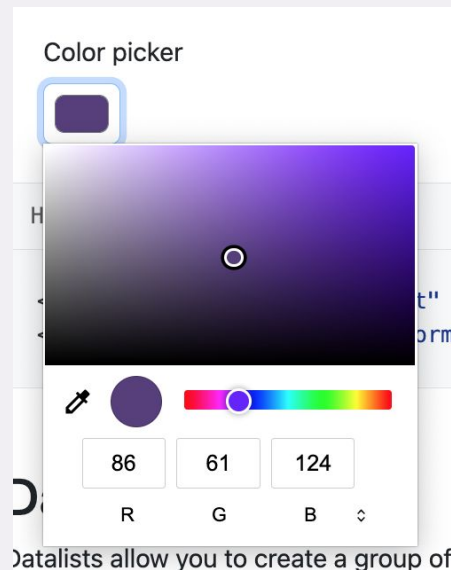
name@example.com

Отправить

Полезный input

Это поле ввода хранит значение в виде строки цвета, например, #000000

type="color"



Практика

Поля со значением

Теория

Поля с выбором

Иногда пользователя нужно ограничить в выборе.

Для этого на сайтах вы могли встречать такие поля ввода:

1. Чекбокс



2. Выпадающий список

Open this select menu



Выпадающий список – это поле, которое встречается в любом проекте.

Сейчас и мы научимся его использовать через JS!

select

```
<select class="form-select">  
  <option value="1">One</option>  
  <option value="2">Two</option>  
  <option value="3">Three</option>  
</select>
```

Популярные



Событие input

Событие **input** для тега **select** срабатывает в тот момент, когда пользователь выбрал одну из опций внутри него.

Значение этой опции можно посмотреть в объекте события, как у обычных текстовых полей.

```
select.addEventListener('input', function  
(evt) {  
    console.log(evt.target.value);  
});
```

Полезный input

Это поле ввода позволяет ввести число в ограниченном диапазоне.

Оно не совсем с выбором и не совсем текстовое... Но зато очень полезное)

Example range



`type="range"`

А теперь перейдём к самому сложному и полезному – чекбоксам!

Бренды

- ☐ Xiomi
- ☒ Apple
- ☐ Sumsung



checkbox

```
<div class="form-check">  
  <input class="form-check-input"  
    id="check"  
    type="checkbox"  
  />  
  
  <label class="form-check-label"  
    for="check">  
    Чекбокс  
  </label>  
</div>
```



Чекбокс

checked

```
<input class="form-check-input"  
      id="accepted"  
      type="checkbox"  
      checked  
>
```



Чекбокс

Атрибут **checked** переводит поле
в состояние "выбрано"

Свойство checked

В JS можно посмотреть **значение атрибута checked** для чекбокса. У него возможны два значения:

1. **true** – чекбокс выбран
2. **false** – чекбокс не выбран

```
checkboxbox.addEventListener(`input`,  
function (evt) {  
    console.log(evt.target.checked);  
});
```

Свойство checked

Это позволяет делать интересные проверки!

```
checkboxbox.addEventListener(`input`, function (evt) {  
    if (evt.target.checked == true) {  
        // если чекбокс выбран  
    } else {  
        // если чекбокс не выбран  
    }  
});
```

Свойство checked

Или даже ещё короче!

```
checkboxbox.addEventListener(`input`, function (evt) {  
    if (evt.target.checked) {  
        // если чекбокс выбран  
    } else {  
        // если чекбокс не выбран  
    }  
});
```

Свойство disabled

Этот атрибут **блокирует** работу любого поля ввода. И его тоже можно задавать через **true** и **false**!

```
// заблокировать  
input.disabled = true;  
  
// разблокировать  
input.disabled = false;
```

Итоги

- Свойство **value** хранит значение для всех полей ввода, кроме checkbox
- Checkbox использует свойство **checked** со значением **true** или **false**.

Практика

Поля с выбором

Цели урока:

- ✓ Научились обрабатывать ввод пользователя на лету
- ✓ Создали генератор аватарки

Домашнее задание