



JS

JavaScript

JavaScript Dev





Document Object Model



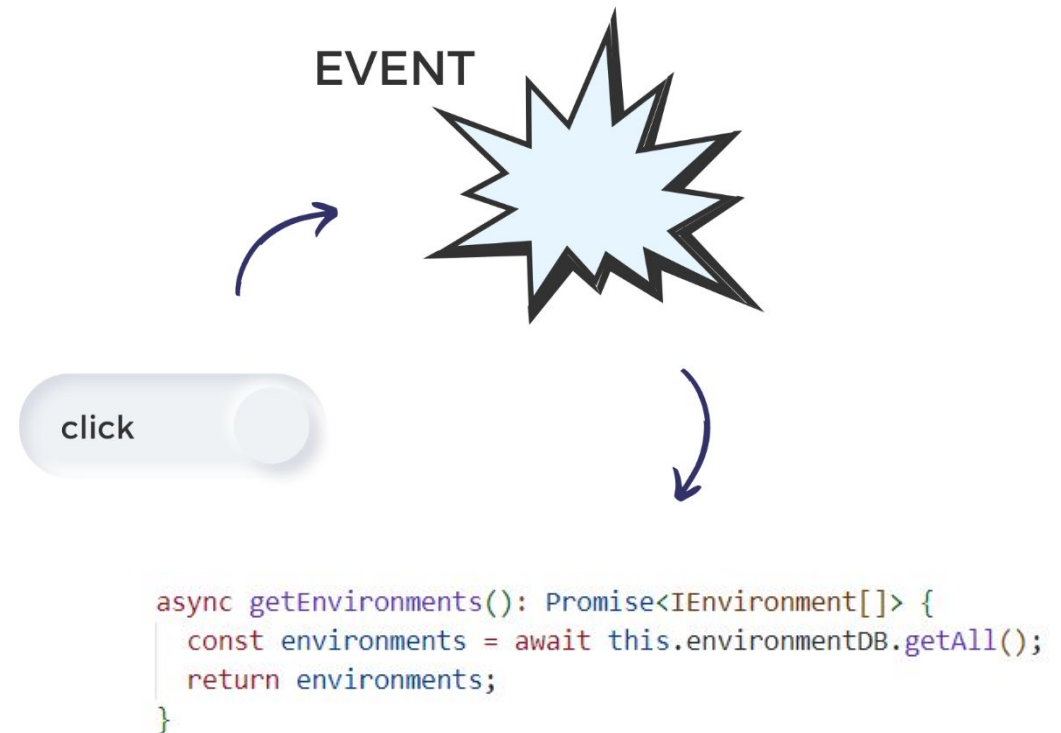
event



target

События DOM

События DOM присылаются чтобы уведомить код о том, что интересующие его действия произошли. События могут возникать в результате действий пользователя, таких как клик по кнопке авторизации или изменение значения поля для ввода (input), использование мыши или изменение размера окна, изменения состояния базовой среды (например, низкий заряд батареи) или мультимедийные события из операционной системы) и других причин.



addEventListener

Для того чтобы отследить событие необходимо использовать метод `addEventListener`, который содержит 2 параметра: событие и функция обратного вызова. Это мы уже знаем с предыдущего занятия. Но 😊...

На самом деле callback

этого метода содержит

скрытый, необязательный

параметр `event`.

```
<> i.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible"
6     content="IE=edge">
7   <meta name="viewport"
8     content="width=device-width,
9     initial-scale=1.0">
10  <title>Document</title>
11 </head>
12 <body>
13   <button class="btn">Кнопка</button>
14
15   <script src="a.js"></script>
16 </body>
17 </html>

JS a.js > ...
1 const btn = document.querySelector('.btn')
2
3 btn.addEventListener('click', () => {
4   alert('Событие по клику')
5 })
6
7
8
9
10
11
12
13
14
15
16
17
```

Параметр event, свойство таргет

Параметр callback принимает событие исполняемое в текущий момент. Это событие включает свойство target, содержащее элемент, на котором сработало событие. Это не тот элемент, к которому был привязан обработчик этого события, а именно самый глубокий тег, на который непосредственно был, совершен клик, как в примере слайда.

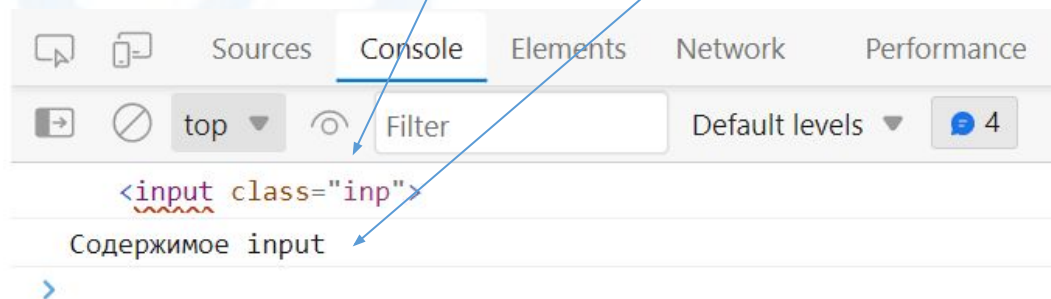
```
<> i.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible"
6     content="IE=edge">
7   <meta name="viewport"
8     content="width=device-width,
9     initial-scale=1.0">
10  <title>Document</title>
11 </head>
12 <body>
13   <button class="btn">Кнопка</button>
14
15   <script src="a.js"></script>
16 </body>
17 </html>
```

```
JS a.js > ...
1 const btn = document.querySelector('.btn')
2
3 btn.addEventListener('click', (event) => {
4   alert(event.target)
5 })
6
7
8
9
10
11
12
13
14
15
16
17
```

```
<> i.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <input class="inp"/>
11
12  <script src="a.js"></script>
13 </body>
14 </html>
```

Содержимое input

```
JS a.js > ...
1 const inp = document.querySelector('.inp')
2
3 inp.addEventListener('change', (event) => {
4   console.log(event.target);
5   console.log(event.target.value);
6 })
```



1. На странице расположен маркированный список. Пользователь кликает на каждый item списка. Необходимо отловить на какой из элементов нажал пользователь и отобразить

- Первый
- Второй
- Третий
- Четвертый
- Пятый

2. По наведению на кнопку вывести сообщение «Hi». Когда мышка выходит за границы кнопки сновка скрывать

button

- По двойному клику на кнопку изменить цвет кнопки

button

- Вывести в строчку все вводимые значения через инпут в виде массива. Добавить проверку, что поле не пустое. После каждого клика очищать значение input. Если вводимое значение не проходит валидацию, то на месте массива вывести сообщение об ошибке красным цветом

Результат:

- Добавьте картинку на страницу. При наведении на картинку мышкой необходимо ее изменять на другую картинку. Но как только мышка снова отходит в сторону, то возвращается первая картинка



- Вы вводите строку. Проверить является ли это слово палиндром. Добавить проверки

Результат:

Это палиндром

7. *Дана таблица с 1 колонкой. Под таблицей сделайте форму (инпут и кнопка), с помощью которой можно будет добавить нового значение в таблицу.

Ячейка 1

Ячейка 3

<input type="text"/>	Кнопка
----------------------	--------

```
<> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  > <head> ...
8  </head>
9  <body>
10     <table>
11         <tbody class="table">
12             <tr>
13                 <td>Ячейка 1</td>
14             </tr>
15             <tr>
16                 <td>Ячейка 3</td>
17             </tr>
18         </tbody>
19     </table>
20
21     <input class="inp" type="text">
22     <button class="btn">Кнопка</button>
23
24     <script src="s.js"></script>
25 </body>
26 </html>
```

ДЗ:

1. Задачи
2. Теория

