

JS: DOM, Events

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

Повторим;)



Что такое DOM?



Какие методы поиска элементов в DOM вы знаете?



Назовите функцию для создания элемента с помощью JS



Как удалить html элемент с помощью JS

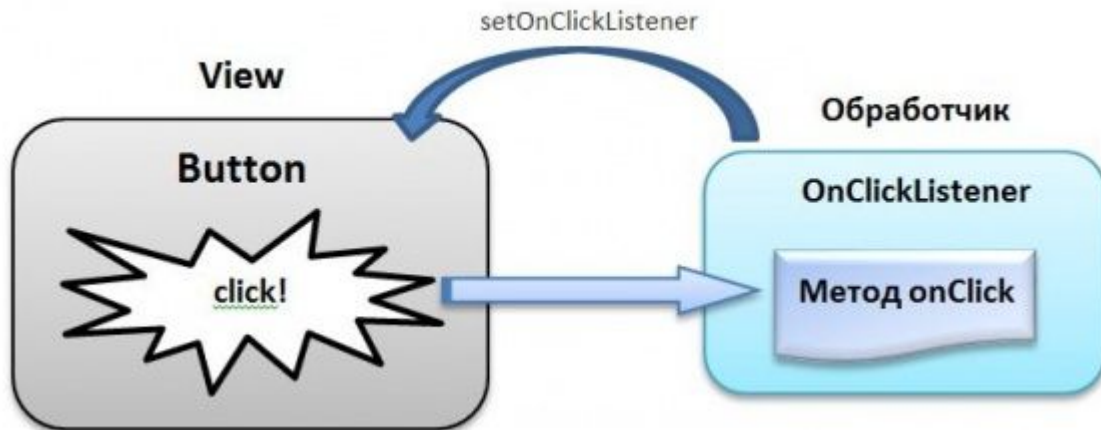
ЦЕЛЬ

Изучить работу с событиями в DOM. Получить данные из формы с помощью JS

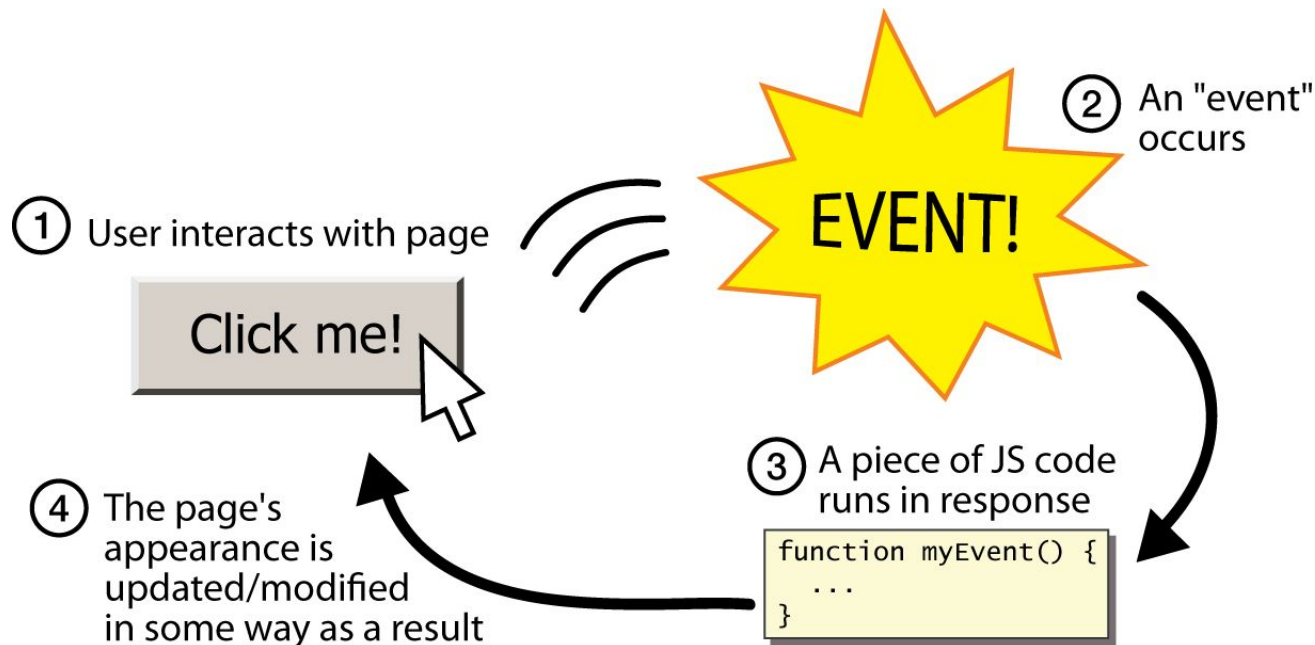
ПЛАН ЗАНЯТИЯ

- Методы работы с DOM. Продолжение
- Получение данных из формы

Events, event listeners

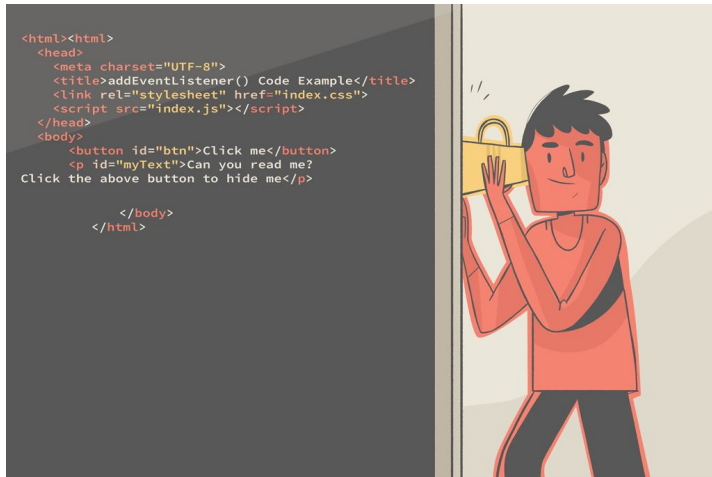


Любой DOM элемент запускает событие, когда мы с ним как-то взаимодействуем (кликаем, наводим мышь и др.). Обработчики событий в JS используются для того, чтобы реагировать на эти события.



Чтобы "повесить" обработчик событий на элемент, нужно использовать специальный метод - **addEventListener**. Этот метод принимает 2 аргумента:

1. **Тип события** (например "click").
2. Так называемую колбэк (**callback**) функцию, которая запускается после срабатывания нужного события.



```
element.addEventListener('click', handleClickFunction)
```


Пример

Найдём кнопку на странице и будем выводить сообщение в консоль, когда произошёл клик по этой кнопке.

```
1  const element = document.querySelector('button')
2
3  element.addEventListener('click', function (event) {
4      console.log('Произошло событие', event.type)
5  })
-
```

Типы событий

eventType (первый аргумент `addEventListener`) - строка, содержащая название события.

Наиболее популярные события:

- 'click'
- 'change'
- 'submit'
- 'keydown'
- 'keyup'
- 'mousemove'
- 'mouseenter'
- 'mouseleave'

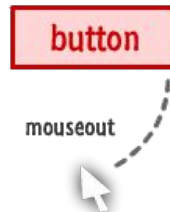
1. Normal



2. Hover Over



3. Hover Out



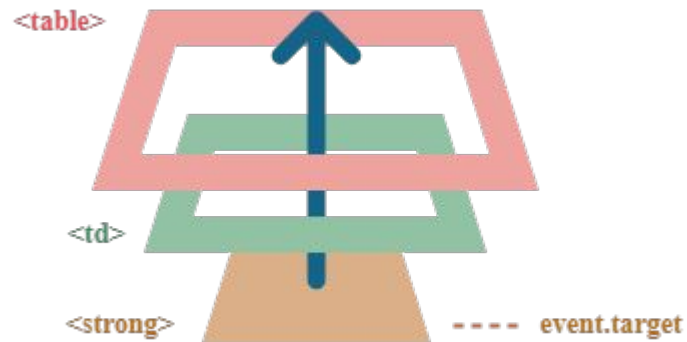
Объект Event

Когда происходит событие, браузер создаёт объект события - event, записывает в него детали и передаёт его в качестве аргумента функции-обработчику (второму аргументу `addEventListener`).



Объект Event - основные свойства

- **target** — ссылка на объект, которым было инициировано событие. Например, если событие произошло на поле ввода, мы получим ссылку на этот DOM элемент.
- **type** — тип события.
- **preventDefault()** - метод отменяет действие браузера по умолчанию для данного события



Объект Event - метод preventDefault

`event.preventDefault()` — предотвращает дефолтное поведение события.

Например, при нажатии на ссылку, отменить переход по адресу ссылки



Работа с формой

```
<form id="myForm">
  <label for="username">Username:</label>
  <input type="text" id="username"
name="username">


  <label for="password">Password:</label>
  <input type="password" id="password"
name="password">

  <button type="button"
onclick="getData()">Submit</button>
</form>
```

```
function getData() {
  const form = document.getElementById('myForm');

  // Получение данных по имени
  // два варианта
  let username = form.elements["username"].value;
  let password = form.password.value;

  // Действия с данными
  console.log('Username:', username);
  console.log('Password:', password);
}
```



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH