Домашнее задание по теме «Объект Event. Делегирование событий»

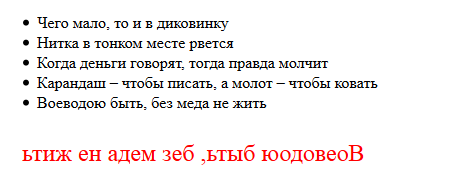
## Задание 1: Использование свойства event.target

Создайте список из 5-6 любых пословиц или цитат.

При клике на любом элементе этого списка пословица должна быть выведена в div-е, расположенном ниже списка, но буквы должны идти в обратном порядке.

Важно: решение должно быть с использованием свойства event.target. Обработчик события назначается для элемента ul, а не для каждого li.

Подсказка: используйте методы для работы со строками.



**Задание 2. Выбор цвета и цены футболки**

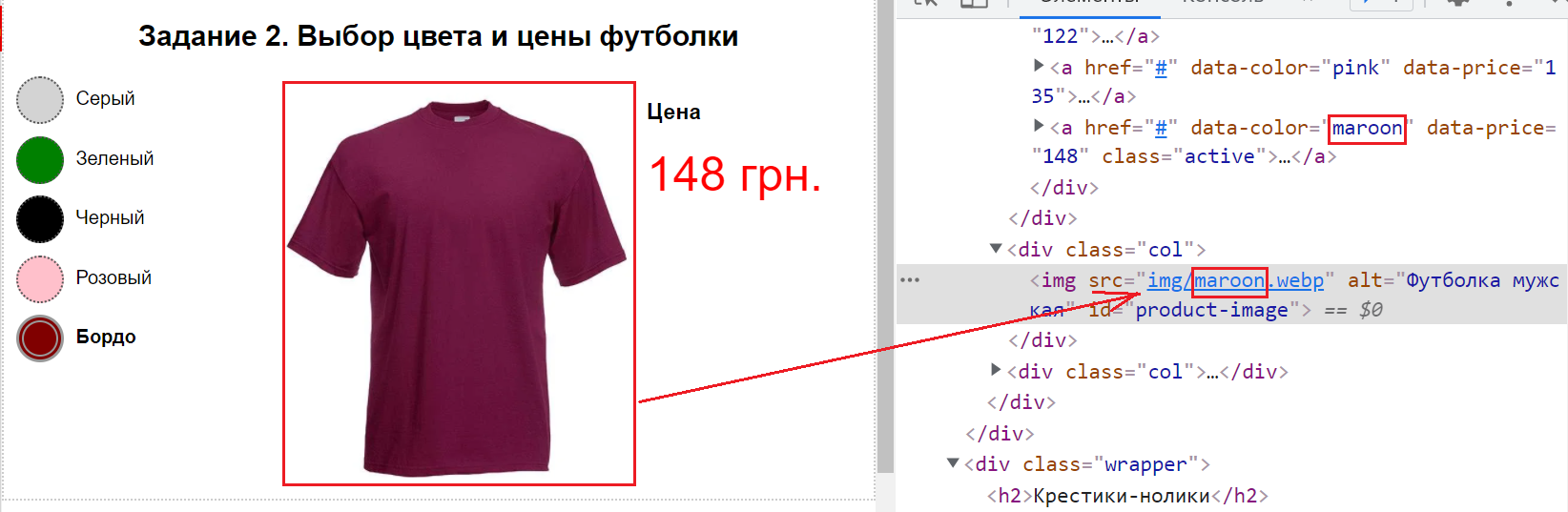
**Необходимо использовать делегирование событий при работе со ссылками (свойство event.target) на основе файла index.html**

В разметке файла index.html, который вы найдете в папке с ДЗ, для ссылок с цветами футболок заданы data-атрибуты с названиями цветов и ценой каждой футболки. Вам нужно с помощью JavaScript назначить цвет из **data-color** для вложенного в ссылку элемента span в виде его стилевого свойства background-color.



Также нужно использовать data-color для изменения изображения, т.к. все фото футболок в папке img имеют то же название, что и цвет.

Получить доступ к атрибуту data-color элемента можно с помощью свойства *elem.dataset.color*.



При клике на ссылку ей должен назначаться класс *active*, который указывает на выбранный в данный момент цвет.



Кроме того, цена изделия справа от изображения с футболкой также должна меняться в соответствии с указанным в аттрибуте *data-price* значением.

## Задание 3

## Необходимо для таблицы, которую вы найдете в html-разметке файла index.html, обработать события click. При этом нужно *делегировать обработку этого события для ячеек*.

## При клике на любой из ячеек в нее нужно помещать “крестик” или “нолик”, представленные svg-файлами в папке img, в зависимости от того, какой клик (четный или нечетный) совершил пользователь. То есть добавление крестика или нолика нужно чередовать.

## 

## Процесс добавления изображения можно реализовать с помощью методов [document.createElement(tagName)](https://html-plus.in.ua/sozdanie-html-yelementov-v-javascript/) и elem.append(tag):

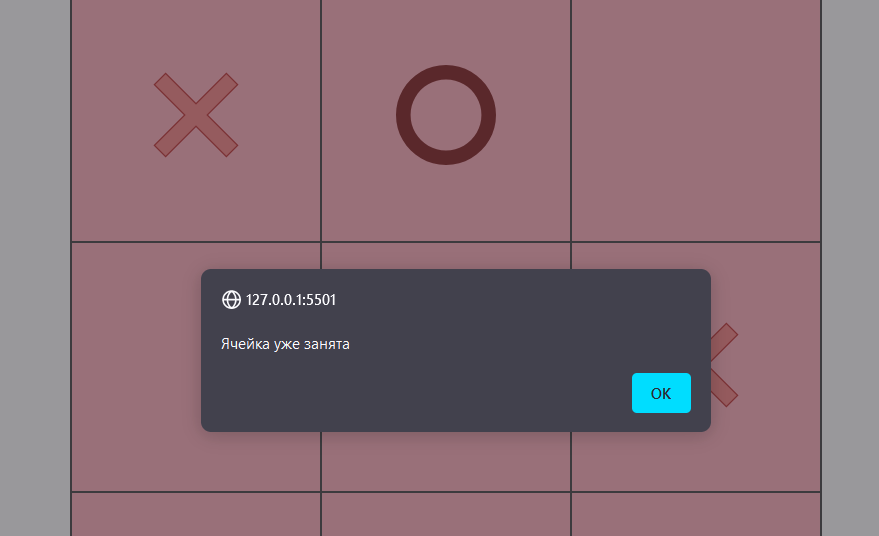
let img = document.createElement('img');

td.append(img);

## **td** - это та ячейка, на которой кликает пользователь.

## При этом нужно отслеживать, чтобы в ячейке с уже добавленным туда изображением сделать клик с добавлением нового изображения-значка нолика или крестика, было нельзя.

## Проверить это можно с помощью свойства **element.children.length** для ячейки таблицы. Если это свойство > 0, то элемент уже содержит вложенный элемент, и второй двобаить туда нельзя.



Обратите внимание, что скрипт должен содержать **обработчик клика только для таблицы**, но с делегированием его к ячейкам и изображениям.

**Завершение игры**

Когда пользователь сделает 9 кликов, вы должны вывести ему предупреждение, что игра закончена, и через 5 секунд очистить все ячейки таблицы от изображений с помощью методов [elem.remove()](https://html-plus.in.ua/udalenie-html-elementov-v-javascript/) и [setTimeout(func, delay)](https://learn.javascript.ru/settimeout-setinterval).

Клики нужно записывать в отдельную переменную и считать только тогда, когда добавляется изображение в пустую ячейку.

**Дополнительное задание**

Будет очень здорово, если вы реализуете игру до конца, т.е. при выстраивании крестиков или ноликов в одну линию или по одной из диагоналей, это направление перечеркивается, и игра должна закончиться, о чем можно вывести предупреждение в виде диалогового окна alert().

Для реализации перечеркивания в коде есть классы *.line, .line-vertical* для горизонтальных и вертикальных линий и *.line-diagonal-left* и *.line-diagonal-right* для диагоналей с css-свойствами для [псевдоэлемента ::before](https://html-plus.in.ua/psevdo-elements-what-is-before-and-after/).

