**[Скрыть элемент по нажатию кнопки](https://learn.javascript.ru/introduction-browser-events" \l "skryt-element-po-nazhatiyu-knopki)**

важность: 5

Добавьте JavaScript к кнопке button, чтобы при нажатии элемент <div id="text"> исчезал.

 document.getElementById('hider').onclick = function() {

      document.getElementById('text').hidden = true;

    }

**[Спрятать себя](https://learn.javascript.ru/introduction-browser-events" \l "spryatat-sebya)**

важность: 5

Создайте кнопку, которая будет скрывать себя по нажатию.

Ответ: Добавить кнопку через input и установить onclick="this.hidden=true"

**[Какой обработчик запустится?](https://learn.javascript.ru/introduction-browser-events" \l "kakoy-obrabotchik-zapustitsya)**

важность: 5

В переменной button находится кнопка. Изначально на ней нет обработчиков.

Который из обработчиков запустится? Что будет выведено при клике после выполнения кода?

button.addEventListener("click", () => alert("1"));

button.removeEventListener("click", () => alert("1"));

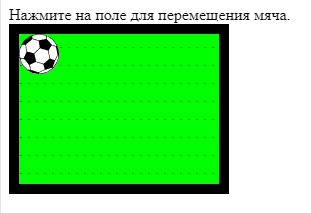
button.onclick = () => alert(2);

Ответ: будет выведено 1 и 2

### [Передвиньте мяч по полю](https://learn.javascript.ru/introduction-browser-events" \l "peredvinte-myach-po-polyu)

важность: 5

Пусть мяч перемещается при клике на поле, туда, куда был клик, вот так:



Требования:

* Центр мяча должен совпадать с местом нажатия мыши (если это возможно без пересечения краёв поля);
* CSS-анимация желательна, но не обязательна;
* Мяч ни в коем случае не должен пересекать границы поля;
* При прокрутке страницы ничего не должно ломаться;

Заметки:

* Код должен уметь работать с различными размерами мяча и поля, не привязываться к каким-либо фиксированным значениям.
* Используйте свойства event.clientX/event.clientY для определения координат мыши при клике.

field.addEventListener('click', (event) => {

            let moveY = event.clientY - field.clientTop - field.offsetTop - ball.offsetHeight / 2;

            let moveX = event.clientX - field.clientLeft - field.offsetLeft - ball.offsetWidth / 2;

            let rightLimit = field.clientWidth - ball.offsetWidth;

            let bottomLimit = field.clientHeight - ball.offsetHeight;

            if (moveY < 0) {

                moveY = 0;

            }

            if (moveX < 0) {

                moveX = 0;

            }

            if (moveX > rightLimit) {

                moveX = rightLimit;

            }

            if (moveY > bottomLimit) {

                moveY = bottomLimit;

            }

            ball.style.top = moveY + 'px';

            ball.style.left = moveX + 'px';

        })

### [Создать раскрывающееся меню](https://learn.javascript.ru/introduction-browser-events" \l "sozdat-raskryvayuscheesya-menyu)

важность: 5

Создать меню, которое по нажатию открывается либо закрывается:

P.S. HTML/CSS исходного документа можно и нужно менять.

Ответ в <Index_1.html>

### [Добавить кнопку закрытия](https://learn.javascript.ru/introduction-browser-events" \l "dobavit-knopku-zakrytiya)

важность: 5

Есть список сообщений.

При помощи JavaScript для каждого сообщения добавьте в верхний правый угол кнопку закрытия.

 let panes = document.querySelectorAll('.pane');

        for (let pane of panes) {

            pane.insertAdjacentHTML("afterbegin", '<button class="remove-button">[x]</button>');

            pane.firstChild.onclick = () => pane.remove();

        };

### [Карусель](https://learn.javascript.ru/introduction-browser-events" \l "karusel)

важность: 4

Создайте «Карусель» –- ленту изображений, которую можно листать влево-вправо нажатием на стрелочки.

В дальнейшем к ней можно будет добавить анимацию, динамическую подгрузку и другие возможности.

P.S. В этой задаче разработка структуры HTML/CSS составляет 90% решения.

let i = 1;

        for (let li of carousel.querySelectorAll('li')) {

            li.style.position = 'relative';

            li.insertAdjacentHTML('beforeend', `<span style="position:absolute;left:0;top:0">${i}</span>`);

            i++;

        }

        let width = 130;

        let count = 3;

        let list = carousel.querySelector('ul');

        let listElems = carousel.querySelectorAll('li');

        let position = 0;

        carousel.querySelector('.prev').onclick = function () {

            position += width \* count;

            position = Math.min(position, 0)

            list.style.marginLeft = position + 'px';

        };

        carousel.querySelector('.next').onclick = function () {

            position -= width \* count;

            position = Math.max(position, -width \* (listElems.length - count));

            list.style.marginLeft = position + 'px';

        };