**[Спрячьте сообщения с помощью делегирования](https://learn.javascript.ru/event-delegation" \l "spryachte-soobscheniya-s-pomoschyu-delegirovaniya)**

важность: 5

Дан список сообщений с кнопками для удаления [x]. Заставьте кнопки работать.

P.S. Используйте делегирование событий. Должен быть лишь один обработчик на элементе-контейнере для всего.

 container.onclick = function (event) {

            if (event.target.className != 'remove-button') return;

            let pane = event.target.closest('.pane');

            pane.remove();

        };

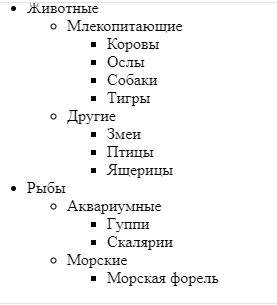
### [Раскрывающееся дерево](https://learn.javascript.ru/event-delegation" \l "raskryvayuscheesya-derevo)

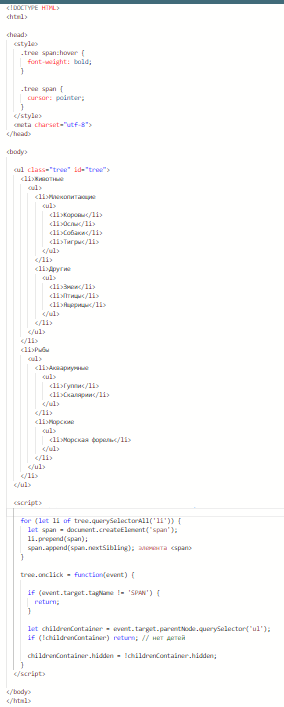
важность: 5

Создайте дерево, которое по клику на заголовок скрывает-показывает потомков:

Требования:

* Использовать только один обработчик событий (применить делегирование)
* Клик вне текста заголовка (на пустом месте) ничего делать не должен.





### [Сортируемая таблица](https://learn.javascript.ru/event-delegation" \l "sortiruemaya-tablitsa)

важность: 4

Сделать таблицу сортируемой: при клике на элемент <th> строки таблицы должны сортироваться по соответствующему столбцу.

Каждый элемент <th> имеет атрибут data-type:

<table id="grid">

<thead>

<tr>

<th data-type="number">Возраст</th>

<th data-type="string">Имя</th>

</tr>

</thead>

<tbody>

<tr>

<td>5</td>

<td>Вася</td>

</tr>

<tr>

<td>10</td>

<td>Петя</td>

</tr>

...

</tbody>

</table>

В примере выше первый столбец содержит числа, а второй – строки. Функция сортировки должна это учитывать, ведь числа сортируются иначе, чем строки.

Сортировка должна поддерживать только типы "string" и "number".

Работающий пример:



P.S. Таблица может быть большой, с любым числом строк и столбцов.

grid.onclick = function (e) {

            if (e.target.tagName != 'TH') return;

            let th = e.target;

            // если ячейка TH, тогда сортировать

            // cellIndex - это номер ячейки th:

            //   0 для первого столбца

            //   1 для второго и т.д.

            sortGrid(th.cellIndex, th.dataset.type);

        };

        function sortGrid(colNum, type) {

            let tbody = grid.querySelector('tbody');

            let rowsArray = Array.from(tbody.rows);

            // compare(a, b) сравнивает две строки, нужен для сортировки

            let compare;

            switch (type) {

                case 'number':

                    compare = function (rowA, rowB) {

                        return rowA.cells[colNum].innerHTML - rowB.cells[colNum].innerHTML;

                    };

                    break;

                case 'string':

                    compare = function (rowA, rowB) {

                        return rowA.cells[colNum].innerHTML > rowB.cells[colNum].innerHTML ? 1 : -1;

                    };

                    break;

            }

            // сортировка

            rowsArray.sort(compare);

            tbody.append(...rowsArray);

        }

**[Поведение "подсказка"](https://learn.javascript.ru/event-delegation" \l "povedenie-podskazka)**

важность: 5

Напишите JS-код, реализующий поведение «подсказка».

При наведении мыши на элемент с атрибутом data-tooltip, над ним должна показываться подсказка и скрываться при переходе на другой элемент.

Пример HTML с подсказками:

<button data-tooltip="эта подсказка длиннее, чем элемент">Короткая кнопка</button>

<button data-tooltip="HTML<br>подсказка">Ещё кнопка</button>

Результат в ифрейме с документом:

В этой задаче мы полагаем, что во всех элементах с атрибутом data-tooltip – только текст. То есть, в них нет вложенных тегов (пока).

Детали оформления:

* Отступ от подсказки до элемента с data-tooltip должен быть 5px по высоте.
* Подсказка должна быть, по возможности, посередине элемента.
* Подсказка не должна вылезать за границы экрана, в том числе если страница частично прокручена, если нельзя показать сверху – показывать снизу элемента.
* Текст подсказки брать из значения атрибута data-tooltip. Это может быть произвольный HTML.

Для решения вам понадобятся два события:

* mouseover срабатывает, когда указатель мыши заходит на элемент.
* mouseout срабатывает, когда указатель мыши уходит с элемента.

Примените делегирование событий: установите оба обработчика на элемент document, чтобы отслеживать «заход» и «уход» курсора на элементы с атрибутом data-tooltip и управлять подсказками с их же помощью.

После реализации поведения – люди, даже не знакомые с JavaScript смогут добавлять подсказки к элементам.

P.S. В один момент может быть показана только одна подсказка.

    document.addEventListener('mouseover', addTooltip);

        document.addEventListener('mouseout', removeTooltip);

        let tooltip = document.createElement('div');

        function addTooltip(event) {

            if (event.target.tagName === 'BUTTON') {

                let button = event.target;

                let tooltipContent = event.target.dataset.tooltip;

                tooltip.className = 'tooltip';

                tooltip.innerHTML = tooltipContent;

                document.body.append(tooltip);

                let coords = button.getBoundingClientRect();

                let top = coords.top - tooltip.offsetHeight - 5;

                if (top < 0) {

                    top = coords.top + button.offsetHeight + 5;

                }

                let left = coords.left + (button.offsetWidth - tooltip.offsetWidth) / 2;

                if (left < 0) { left = 0 }

                tooltip.style.top = top + 'px';

                tooltip.style.left = left + 'px';

            }

        }

        function removeTooltip() {

            tooltip.remove();

        }