**Лабораторная работа № 17**

**Обработка событий в JS**

На указанных в ссылках материалах и источниках проработайте примеры на каждый вид интерфейсного события. Минимум по 1 скрипту на каждый вид интерфейсного события. Разработанные скрипты внедрить в ваш индивидуальный сайт (макет по практике).

1. События мыши

<div id="mouseArea" style="width: 300px; height: 300px; background-color: lightblue;">

    Наведите курсор на меня!

</div>

<script>

    const mouseArea = document.getElementById('mouseArea');

    mouseArea.addEventListener('mouseover', () => {

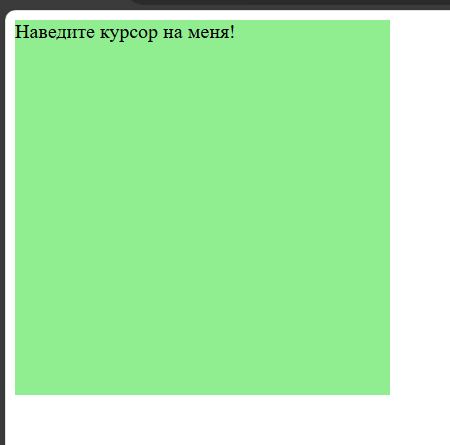
        mouseArea.style.backgroundColor = 'lightgreen';

    });

    mouseArea.addEventListener('mouseout', () => {

        mouseArea.style.backgroundColor = 'lightblue';

    });

</script>

1. События клавиатуры

<input type="text" id="textInput" placeholder="Начните вводить текст..." />

<p id="output"></p>

<script>

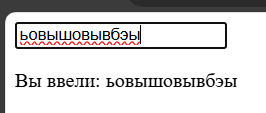
    const textInput = document.getElementById('textInput');

    const output = document.getElementById('output');

    textInput.addEventListener('input', () => {

        output.textContent = `Вы ввели: ${textInput.value}`;

    });

</script>

1. Drag&DROP события

<div id="dragItem" draggable="true" style="width: 100px; height: 100px; background-color: tomato;">

    Перетащите меня!

</div>

<div id="dropArea" style="width: 300px; height: 300px; background-color: lightgray; margin-top: 20px;">

    Сюда уроните элемент

</div>

<script>

    const dragItem = document.getElementById('dragItem');

    const dropArea = document.getElementById('dropArea');

    dragItem.addEventListener('dragstart', (event) => {

        event.dataTransfer.setData('text/plain', null);

    });

    dropArea.addEventListener('dragover', (event) => {

        event.preventDefault();

    });

    dropArea.addEventListener('drop', () => {

        dropArea.appendChild(dragItem);

    });

</script>

1. События указателя

<div id="pointerArea" style="width: 300px; height: 300px; background-color: lavender;">

    Нажмите и удерживайте здесь!

</div>

<script>

    const pointerArea = document.getElementById('pointerArea');

    pointerArea.addEventListener('pointerdown', () => {

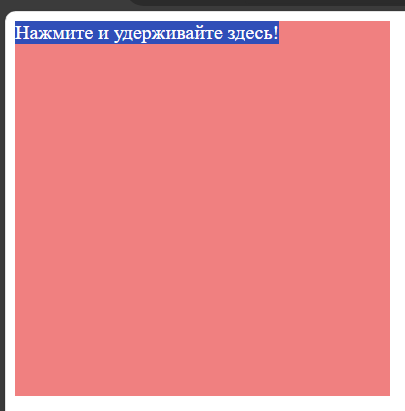
        pointerArea.style.backgroundColor = 'lightcoral';

    });

    pointerArea.addEventListener('pointerup', () => {

        pointerArea.style.backgroundColor = 'lavender';

    });

</script>

1. События полосы прокрутки

<div id="scrollArea" style="height: 200px; overflow-y: scroll; border: 1px solid black;">

    <div style="height: 600px;">

        Прокручиваемый контент!

    </div>

</div>

<p id="scrollOutput"></p>

<script>

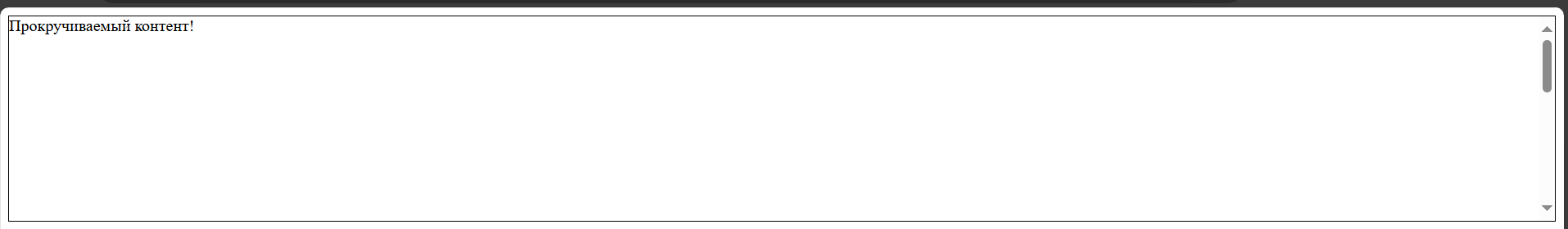
    const scrollArea = document.getElementById('scrollArea');

    const scrollOutput = document.getElementById('scrollOutput');

    scrollArea.addEventListener('scroll', () => {

        scrollOutput.textContent = `Позиция прокрутки: ${scrollArea.scrollTop}px`;

    });

</script>

1. События сенсорных экранов

<div id="touchArea" style="width: 300px; height: 300px; background-color: lightyellow;">

    Коснитесь меня!

</div>

<script>

    const touchArea = document.getElementById('touchArea');

    touchArea.addEventListener('touchstart', () => {

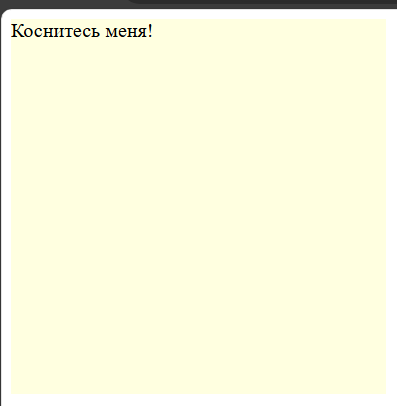
        touchArea.style.backgroundColor = 'lightblue';

    });

    touchArea.addEventListener('touchend', () => {

        touchArea.style.backgroundColor = 'lightyellow';

    });

</script>

1. События, связанные с таймером

<button id="startTimer">Начать таймер</button>

<p id="timerOutput">0 секунд</p>

<script>

    let seconds = 0;

    let timer;

    document.getElementById('startTimer').addEventListener('click', () => {

        clearInterval(timer);

        seconds = 0;

        timer = setInterval(() => {

            seconds++;

            document.getElementById('timerOutput').textContent = `${seconds} секунд`;

        }, 1000);

    });

</script>

