|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | | | |
| **по дисциплине** | | | |
| **«Разработка клиентских частей интернет-ресурсов»** | | | |
| Выполнил студент группы ИКБО-01-19 | | Львов Т.Р. | |
|  | |  | |
| Принял  *Ассистент* | | Коваленко М.А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись руководителя) | |
|  |  | |  | |

Москва 2020

**Оглавление**

[Практическая работа №5: «JavaScript. Работа с переменными, строками, массивами, циклами и условиями» 3](#_Toc57432337)

[**Задача 1.** 3](#_Toc57432338)

[**Задача 2.** 6](#_Toc57432339)

[**Задача 3.** 8](#_Toc57432340)

[**Задача 4.** 10](#_Toc57432341)

[**Задача 5.** 11](#_Toc57432342)

[**Задача 6.** 13](#_Toc57432343)

[**Задача 7.** 14](#_Toc57432344)

[**Задача 8.** 17](#_Toc57432345)

[**Задача 9.** 20](#_Toc57432346)

[**Задача 10.** 23](#_Toc57432347)

[**Задача 11.** 29](#_Toc57432348)

[**Задача 12.** 31](#_Toc57432349)

[**Задача 13.** 35](#_Toc57432350)

[**Задача 14.** 37](#_Toc57432351)

[**Задача 15.** 41](#_Toc57432352)

[**Задача 16.** 42](#_Toc57432353)

[**Задача 17.** 44](#_Toc57432354)

# Практическая работа №7: «JavaScript. Работа с DOM»

**Цель работы:** Изучить принцип работы с DOM.

**Задача 1.**

События через атрибуты:

* 1. Создать страницу с кнопкой, при наведении на которую появляется всплывающее окно в браузере с вашим сообщением;
  2. Создать страницу с кнопкой, при двойном нажатии на которую появляется всплывающее окно в браузере с вашим сообщением.

**Описание выполнения задания:**

Листинг 1.1 – создать на странице кнопку, при наведении на которую выводится сообщение.

    <script>

        let a = document.getElementsByTagName("button")[0];

        a.addEventListener("mouseover", function() {

         alert("Вы навели на кнопку");

         });

    </script>

Результат выполнения кода на странице (см. Рисунок 1.1.1):



Рисунок 1.1.1 - результат выполнения кода

Листинг 1.2 - создать на странице кнопку, при двойном нажатии на которую выводится сообщение.

    <script>

        let a = document.getElementsByTagName("button")[0];

        a.addEventListener("dblclick", function() {

            alert("Вы дважды нажали кнопку");

        });

    </script>

Результат выполнения кода на странице (см. Рисунок 1.2.1):

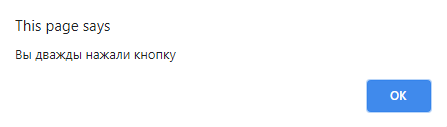


Рисунок 1.2.1

**Задача 2.**

Метод getElementById и работа с атрибутами:

1. Создать страницу с заполненным вашим сообщением текстовым полем и кнопкой рядом, при нажатии на которую данное сообщение меняется на другое;
2. Создать страницу с изображением и кнопкой рядом, при нажатии на которую данное изначальное изображение меняется на другое.

**Описание выполнения задания:**

Листинг 2.1 – создать текстовое поле, при нажатии на которое меняется его содержимое.

<textarea name="pole" id="tar" cols="30" rows="10">

        some text

    </textarea>

    <button id="btn1">click me</button>

<script>

        let btn1 = document.getElementById("btn1"),

            content = document.getElementById("tar");

            btn2 = document.getElementById("btn2")

        let status1 = false;

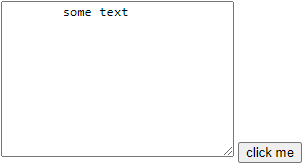
        btn1.addEventListener("click", function() {

            content.textContent = Math.random().toString(36).substr(2, 5);

        });

</script>

Результат выполнения кода на странице (см. Рисунок 2.1.1-2.1.2):

|  |  |
| --- | --- |
| Рисунок 2.1.1 | Рисунок 2.1.2 |

Листинг 2.2 – создать поле с картинкой, при нажатии на которое меняется изображение.

<img src="1.jpg" alt="" height=300px; width=400px;>

    <button id="btn2">click me</button>

<script>

        let status2 = false;

        btn2.addEventListener("click", function() {

            if (!status2) {

                document.getElementsByTagName("img")[0].setAttribute("src", "2.jpeg");

            status2 = true;

            }

            else {

                document.getElementsByTagName("img")[0].setAttribute("src", "1.jpg");

            status2 = false;

            }

        })

 </script>

Результат выполнения кода на странице (см. Рисунок 2.2.1-2.2.2):

|  |  |
| --- | --- |
| Рисунок 2.2.1 | Рисунок 2.2.2 |

**Задача 3.**

Работа с this:

* 1. Создать страницу с заполненным изначальным вашим сообщением текстовым полем, при нажатии на которое данное сообщение меняется на другое, а после нажатия на любое другое место страницы (вне текстового поля) изначальное сообщение возвращается обратно;
  2. Создать страницу с двумя кнопками, на одну из них можно нажать только один раз (после одного нажатия эта кнопка становится неактивной), при нажатии на вторую – первая кнопка снова становится активной;
  3. Создать страницу с кнопкой, которая считает количество нажатий на нее (кол-во нажатий должно отображаться на самой кнопке).

**Описание выполнения задания:**

Листинг 3.1 – создать заполненное текстовое поле, при нажатии на которое меняется его содержимое, а при нажатии на любой другую часть страницы меняется обратно.

<textarea name="t" id="txt" cols="30" rows="10" data-swap="Another text">Some text</textarea>

    <script>

        let txt = document.querySelector("#txt");

        txt.setAttribute("data-original", txt.innerHTML);

        txt.addEventListener("focus", () => {

            txt.innerHTML = txt.getAttribute("data-swap");

        });

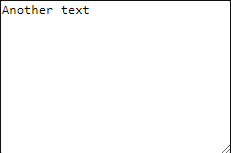
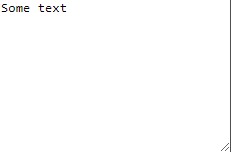
        txt.addEventListener("blur", () => {

            txt.innerHTML = txt.getAttribute("data-original");

        });

    </script>

Результат выполнения кода на странице (см. Рисунок 3.1.1-3.1.2):

|  |  |
| --- | --- |
| Рисунок 3.1.1 | Рисунок 3.1.2 |

Листинг 3.2 –создать две кнопки, при нажатии на одну из них первая становится неактивной, при нажатии на вторую первая снова становится активной.

<button id="btn1" onclick="this.setAttribute('disabled',true)">click1</button>

<button id="btn2" onclick="document.getElementById('btn1').disabled = false;">click2</button>

Результат выполнения кода на странице (см. Рисунок 3.2.1-3.2.2):



|  |  |
| --- | --- |
| Рисунок 3.2.1 | Рисунок 3.2.2 |

Листинг 3.3 –создать две кнопки, при нажатии на одну из них первая становится неактивной, при нажатии на вторую первая снова становится активной.

<button id="btnC" onclick="this.innerHTML++">0</button>

Результат выполнения кода на странице (см. Рисунок 3.3.1-3.3.2):

** **

|  |  |
| --- | --- |
| Рисунок 3.3.1 | Рисунок 3.3.2 |

**Задача 4.**

Работа с CSS:

1. Создать страницу с заполненным вашим сообщением текстовым полем и кнопкой рядом, при нажатии на которую данное сообщение изменяется на другое и меняет свой цвет, а текстовое поле становится шире и его края становятся закругленными;
2. Создать страницу с заполненным вашим сообщением текстовым полем и двумя кнопками рядом, при нажатии на первую из них текстовое поле должно полностью исчезнуть, при нажатии на вторую – снова появиться в изначальном виде.

**Описание выполнения задания:**

Листинг 4.1 – создать текстовое поле, при нажатии на которое сообщение внутри изменяется, меняется цвет, меняется форма поля.

<script>

        let str = "abcde";

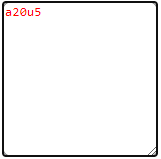
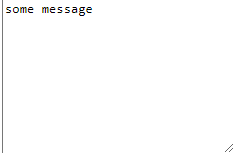
        alert(str[0]);

        alert(str[2]);

        alert(str[4]);

</script>

Результат выполнения кода на странице (см. Рисунок 4.1.1-4.1.2):

****

|  |  |
| --- | --- |
| Рисунок 4.1.1 | Рисунок 4.1.2 |

Листинг 4.2 – реализовать скрытие и показывание контента текстового поля.

<textarea name="" id="ta" cols="30" rows="10"

    onclick="this.innerHTML= Math.random().toString(36).substr(2, 5);

    this.style.width = '150px';

    this.style.color = 'red';

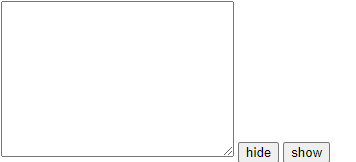
    this.style.borderRadius = '10px';"

    >some message</textarea>

    <button id="hide" onclick="document.getElementById('ta').style.color = 'transparent';">hide</button>

    <button id="show" onclick="document.getElementById('ta').style.color = 'black'">show</button>

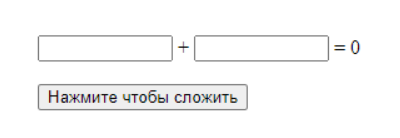
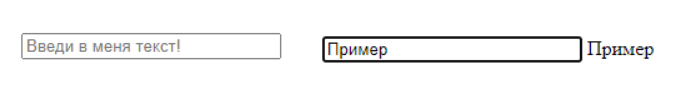
Результат выполнения кода на странице (Рисунок 4.2.1-4.2.2):

****

|  |  |
| --- | --- |
| Рисунок 4.2.1 | Рисунок 4.2.2 |

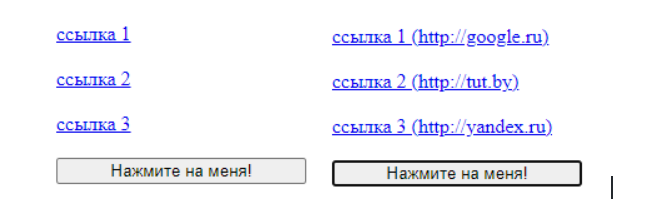
**Задача 5.**

Свойства innerHTML, outerHTML:

1. Создать страницу с абзацем текста и кнопкой рядом. При нажатии на кнопку текст в абзаце и его размер и выделение (станет либо жирным, либо курсивом) поменяются;
2. Создать страницу по данному образцу (при нажатии на кнопку введенные в текстовые поля числовые значения сложатся, и результат выведится после знака равно):  
   
3. Создать страницу по данному образцу   
   

После ввода текста

До ввода текста

1. Создать страницу по данному образцу:  
   

После нажатия на кнопку

До нажатия на кнопку

**Описание выполнения задания:**

Листинг 5.1 – создать абзац и кнопку, при нажатии на которую меняются стили и текст.

    <p id="par"><span>Lorem ipsum dolor sit amet consectetur adipisicing elit. Rerum, voluptates.</span></p>

    <input type="button" id="tb" value="Поменять текст">

    <script>

        let p = document.querySelector("#par");

        let pb = document.querySelector("#tb");

        pb.onclick = function() {

            p.classList.toggle("change");

            p.classList.toggle("changespan");

            p.classList.toggle("change::after");

        };

    </script>

Результат выполнения кода на странице (см. Рисунки 5.1.1-5.1.2):

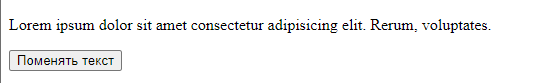


Рисунок 5.1.1

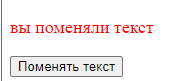


Рисунок 5.1.2

Листинг 5.2 –создать страницу по заданному шаблоны для ввода и последующего сложения двух чисел.

<input type="text" id="num1"> + <input type="text" id="num2"> = <span id="result">0</span>

    <br> <br>

    <input type="button" id="btn" value="Нажмите, чтобы сложить">

    <script>

        let num1 = document.querySelector("#num1"),

        num2 = document.querySelector("#num2"),

        res = document.querySelector("#result"),

        btn = document.querySelector("#btn");

        btn.onclick = function() {

            let result = parseInt(num1.value,0) + parseInt(num2.value,0);

            res.innerHTML = result

        };

    </script>

Результат выполнения кода на странице (см. Рисунки 5.2.1-5.2.2):

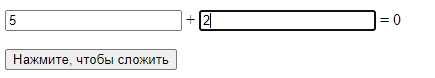


Рисунок 5.2.1

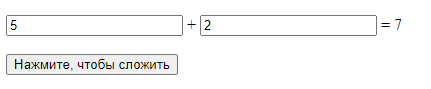


Рисунок 5.2.2

Листинг 5.3 – создать страницу по заданному шаблону для заполнения текстовых полей.

    <input type="text" id="input" placeholder="Введи в меня текст!">

    <p id="values"></p>

    <script>

        let input = document.querySelector("#input");

        let content = document.querySelector("#values");

        input.addEventListener("input", function(e) {

            content.innerHTML = e.target.value;

            content.style.display = "inline";

        });

    </script>

Результат выполнения кода на странице (см. Рисунки 5.3.1-5.3.2):



Рисунок 5.3.1



Рисунок 5.3.2

Листинг 5.4 – создать страницу по заданному в условию образцу.

<a href="http://google.com" id="google">ссылка1</a>

     <a href="http://tut.by" id="tut">ссылка2</a>

     <a href="https://yandex.ru" id="yandex">ссылка3</a>

    <input type="button" id="btn1" value="Нажмите меня!">

    <script>

        let g = document.querySelector("#google");

        let t = document.querySelector("#tut");

        let y = document.querySelector("#yandex");

        let btn1 = document.querySelector("#btn1");

        btn1.onclick = function() {

            g.innerHTML = g.innerHTML + " (http://google.com)";

            t.innerHTML = t.innerHTML + " (http://tut.by)";

            y.innerHTML = y.innerHTML + " (https://yandex.ru)";

            this.style.border = "2px solid black";

        }

    </script>

Результат выполнения кода на странице (см. Рисунки 5.4.1-5.4.2):

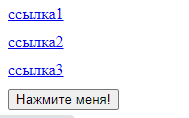


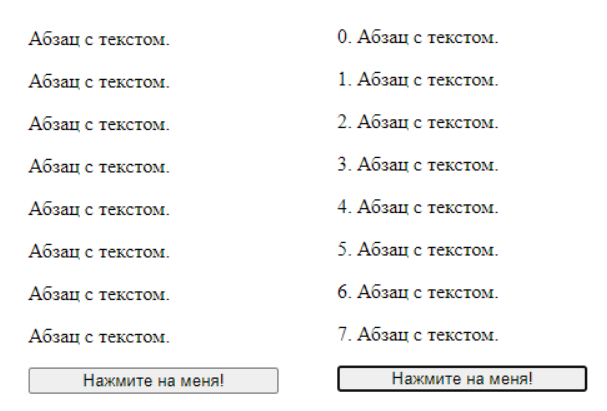
Рисунок 5.4.1



Рисунок 5.4.2

**Задача 6.**

Метод getElementsByTagName:

1. Создать страницу с тремя абзацами текста и кнопкой рядом. При нажатии на кнопку текст в двух абзацах должен поменяться, в третьем – остаться неизменным;
2. Создать страницу по данному образцу (при нажатии на кнопку текст в каждом абзаце поменяется - спереди добавится порядковый номер абзаца):  
   

После нажатия на кнопку

До нажатия на кнопку

**Описание выполнения задания:**

Листинг 6.1 – создать 3 абзаца и кнопку, при нажатии на которую меняется текст первых двух абзацев, а третий остается неизменным.

<p>Lorem ipsum dolor sit amet.</p>

    <p>Lorem ipsum dolor sit amet.</p>

    <p>Lorem ipsum dolor sit amet.</p>

    <input type="button" id="btn" value="Нажми меня">

    <script>

        let p = document.getElementsByTagName("p");

        let btn = document.querySelector("#btn");

        btn.onclick = function() {

            for(let i = 0; i < p.length-1; i++){

                p[i].innerHTML = Math.random().toString(36).substr(2, 5);

            }

        }

    </script>

Результат выполнения кода на странице (см. Рисунок 6.1.1-6.1.2):

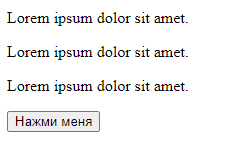


Рисунок 6.1.1

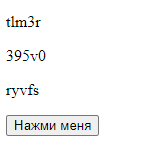


Рисунок 6.1.2

Листинг 6.2 – создать страницу по заданному в условии образцу, при нажатии на кнопку абзацы должны пронумероваться.

<p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <p>Абзац с текстом</p>

    <input type="button" id="btn1" value="Нажмите на меня!">

    <script>

        let p1 = document.getElementsByTagName("p");

        let btn1 = document.querySelector("#btn1");

        btn1.onclick = () => {

            for (let i = 3; p1.length; i++) {

                p1[i].innerHTML = i-3 + ". " + p1[i].innerHTML;

            }

        }

    </script>

Результат выполнения кода на странице (см. Рисунок 6.2.1-6.2.2):

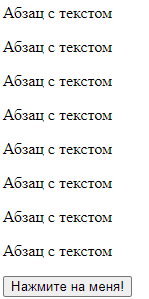


Рисунок 6.2.1

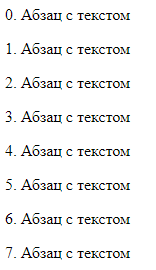


Рисунок 6.2.2

**Задача 7.**

Работа с таймерами:

1. Создать страницу с отсчетом от 0 до бесконечности и кнопкой, при нажатии на которую и начнется этот отсчет;
2. Создать страницу с отсчетом от 0 до бесконечности и двумя кнопками, при нажатии на одну из которых отсчет начнется, при нажатии на вторую - остановится;
3. Создать страницу с тикающими цифровыми часами;
4. Создать страницу с простым слайдером из 4 изображений (на экране изначально одно изображение, и оно поочередно сменяется другими);
5. Создать страницу с каруселью из 5 изображений и двумя кнопками: первая – запускает карусель, вторая – останавливает;
6. Создать страницу с тикающими цифровыми часами, показывающими отсчет оставшегося времени до полуночи.

**Описание выполнения задания:**

Листинг 7.1 - создать таймер с отсчетом от 0 до бесконечности и кнопкой для его запуска.

<div class="timer1">

        0

    </div>

    <input type="button" id="btn" value="Начать отсчет">

Результат выполнения кода на странице (см. Рисунок 7.1.1-7.1.2):



Рисунок 7.1.1



Рисунок 7.1.2

Листинг 7.2 - создать таймер с отсчетом от 0 до бесконечности и кнопками для его запуска и остановки.

<div class="timer2">

        0

    </div>

    <input type="button" id="btn1" value="Начать отсчет"> <input type="button" id="btn2" value="Остановить отсчет">

    <script>

        let d2 = document.querySelector(".timer2");

        let btn1 = document.querySelector("#btn1");

        let btn2 = document.querySelector("#btn2");

        let timerID;

        btn1.addEventListener("click", function() {

            timerID = setInterval(timer2,1000);

        })

        btn2.addEventListener("click", function() {

            clearInterval(timerID);

        })

        function timer2() {

            d2.innerHTML++;

        }

    </script>

Результат выполнения кода на странице (см. Рисунок 7.2.1-7.2.3):



Рисунок 7.2.1



Рисунок 7.2.2.



Рисунок 7.2.3

Листинг 7.3 - создать страницу с тикающими цифровыми часами.

    <div class="clock">

        <span id="hours">00</span>:<span id="minutes">00</span>:<span id="second">00</span>

        <input type="button" id="btnS"value="start"> <input type="button" id="btnC" value="stop">

    </div>

    <script>

        let start = document.querySelector("#btnS");

        let stop = document.querySelector("#btnC");

        let clockID;

        start.addEventListener("click", function() {

            clockID = setInterval(clock,1000);

        });

        stop.addEventListener("click", function() {

            clearInterval(clockID);

        });

        function clock() {

            let date = new Date();

            document.querySelector("#hours").innerHTML = date.getHours();

            document.querySelector("#minutes").innerHTML = date.getMinutes();

            document.querySelector("#second").innerHTML = date.getSeconds();

        }

    </script>

Результат выполнения кода на странице (см. Рисунок 7.3.1-7.3.3):



Рисунок 7.3.1



Рисунок 7.3.2



Рисунок 7.3.3

Листинг 7.4 -создать страницу с простым слайдером из 4 изображений.

    <script>

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

        img.setAttribute("src","pic1.jpg");

        img.style.height = "300px";

        img.style.width = "400px";

        document.body.append(img);

        changeImg();

        setInterval(changeImg,10000);

        function changeImg() {

            setTimeout(function(){

                img.src = "pic1.jpg";

            },2500);

            setTimeout(function(){

                img.src = "pic2.jpg"

            },5000);

            setTimeout(function(){

                img.src = "pic3.jpg"

            },7500);

            setTimeout(function(){

                img.src = "pic4.jpg"

            },10000);

        }

    </script>

Результат выполнения кода на странице (см. Рисунок 7.4.1-7.4.4):

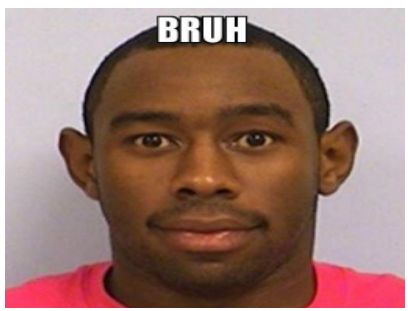


Рисунок 7.4.1



Рисунок 7.4.2

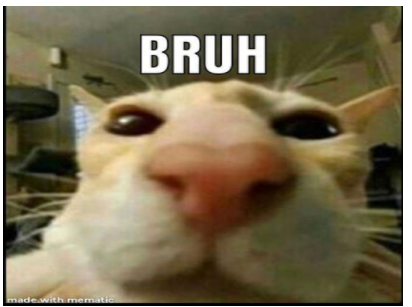


Рисунок 7.4.3



Рисунок 7.4.4

Листинг 7.5 -создать страницу с простой каруселью из 5 изображений с кнопками запуска и остановки.

<script>

    let img1 = document.createElement("img");

    img1.setAttribute("src","pic1.jpg");

    img1.style.height = "300px";

    img1.style.width = "400px";

    let btnStart = document.createElement("input");

    btnStart.value = "Начать";

    btnStart.type = "button";

    let btnStop= document.createElement("input");

    btnStop.value = "Остановить";

    btnStop.type = "button";

    document.body.append(img1);

    document.body.append(btnStart);

    document.body.append(btnStop);

    let imgStop; let i = 0;

    btnStart.addEventListener("click", function() {

        changeImg3();

        imgStop = setInterval(changeImg3,2500);

    })

    btnStop.addEventListener("click", function() {

        clearInterval(imgStop);

    })

        let imgs = {

        0 : "pic1.jpg",

        1 : "pic2.jpg",

        2 : "pic3.jpg",

        3 : "pic4.jpg"

    }

    function changeImg3() {

        img1.src = imgs[(i%4)];

        i++;

    }

</script>

Результат выполнения кода на странице (см. Рисунок 7.5.1-7.5.5):

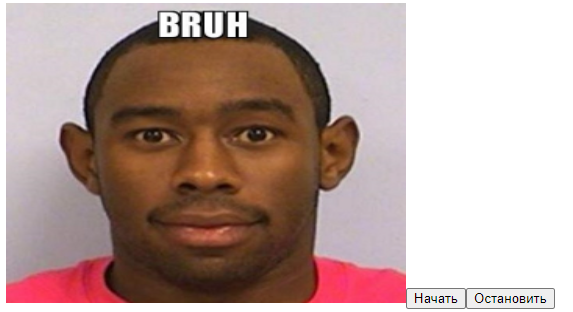


Рисунок 7.5.1



Рисунок 7.5.2

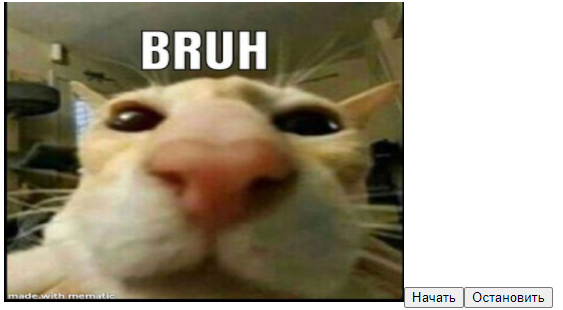


Рисунок 7.5.3



Рисунок 7.5.4



Рисунок 7.5.5

Листинг 7.6 -создать счетчик оставшегося времени до полуночи.

<div class="clock1">

    <span id="hours1">00</span>:<span id="minutes1">00</span>:<span id="second1">00</span>

    <input type="button" id="btnS1"value="start"> <input type="button" id="btnC1" value="stop">

</div>

<script>

    let start1 = document.querySelector("#btnS1");

    let stop1 = document.querySelector("#btnC1");

    let clockID1;

    start1.addEventListener("click", function() {

        clockID1 = setInterval(clock1,1000);

    });

    stop1.addEventListener("click", function() {

        clearInterval(clockID1);

    });

    function clock1() {

        date = new Date();

        document.querySelector("#hours1").innerHTML = 24 - date.getHours();

        document.querySelector("#minutes1").innerHTML = 60 - date.getMinutes();

        document.querySelector("#second1").innerHTML = 60 -date.getSeconds();

    }

</script>

Результат выполнения кода на странице (см. Рисунок 7.6.1-7.6.2):



Рисунок 7.6.1

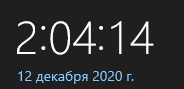


Рисунок 7.6.2

**Задача 8.**

Прокрутка элемента:

1. Дан элемент #elem с вертикальной полосой прокрутки. По нажатию на кнопку вывести на сколько элемент прокручен сверху;
2. Дан элемент #elem с вертикальной полосой прокрутки. По нажатию на кнопку прокрутить его до позиции 200px сверху;
3. Дан элемент #elem с вертикальной полосой прокрутки. По нажатию на кнопку прокрутить его до позиции 200px от нижнего края элемента;

**Описание выполнения задания:**

Листинг 8.1 – вывести на экран насколько прокручен элемент.

    <style>

        #elem {

            overflow: scroll;

            height: 250px;

            width: 250px;

        }

        #content {

            height: 800px;

            width: 2000px;

        }

    </style>

    <div id="elem">

        <div id ="content">

            Lorem ipsum dolor sit amet.

        </div>

    </div>

    <input type="button" value="Нажми меня" id="btn">

    <script>

        let div = document.querySelector("#elem");

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", function() {

            let counterT = div.scrollTop;

            alert("Вертикально: " +counterT + "px")

        })

    </script>

Результат выполнения кода на странице (Рисунок 8.1.1):

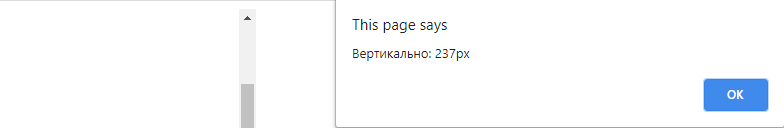


Рисунок 8.1.1

Листинг 8.2 – прокрутить до позиции 200px сверху от элемента.

    <style>

        #elem {

            overflow: scroll;

            height: 250px;

            width: 250px;

        }

        #content {

            height: 800px;

            width: 2000px;

        }

    </style>

    <div id="elem">

        <div id ="content">

            Lorem ipsum dolor sit amet.

        </div>

    </div>

    <input type="button" value="Нажми меня" id="btn">

    <script>

        let div = document.querySelector("#elem");

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", function() {

            div.scrollTop = 200;

        })

    </script>

Результат выполнения кода на странице (см. Рисунок 8.2.1-8.2.2):

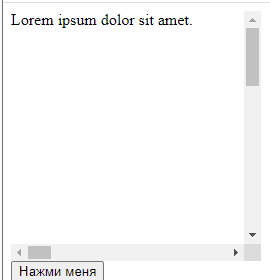


Рисунок 8.2.1

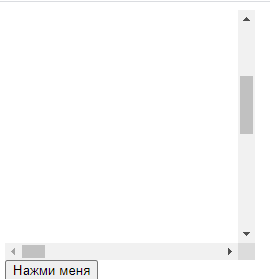


Рисунок 8.2.2

Листинг 8.3 – прокрутить до позиции 200px снизу от нижней границы.

    <style>

        #elem {

            overflow: scroll;

            height: 250px;

            width: 250px;

        }

        #content {

            height: 800px;

            width: 2000px;

        }

    </style>

    <div id="elem">

        <div id ="content">

            Lorem ipsum dolor sit amet.

        </div>

    </div>

    <input type="button" value="Нажми меня" id="btn">

    <script>

        let div = document.querySelector("#elem");

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", function() {

            div.scrollTop = div.scrollHeight - 200 - div.clientHeight;

            let counterT = div.scrollTop;

            alert("Вертикально: " +counterT + "px");

        })

    </script>

Результат выполнения кода на странице (см. Рисунок 8.3.1-8.3.2):

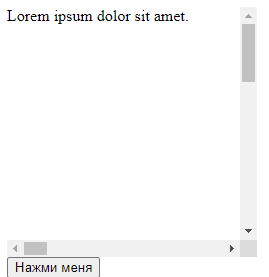


Рисунок 8.3.1

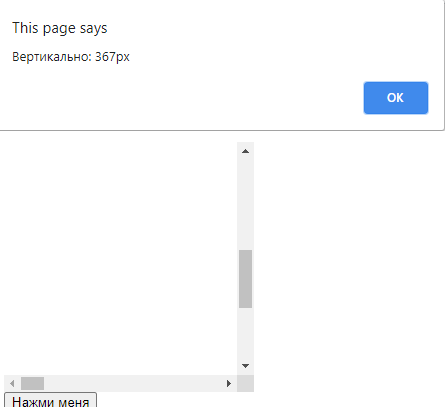


Рисунок 8.3.2

**Задача 9.**

Прокрутка страницы:

* 1. Дана страница с вертикальной полосой прокрутки. По нажатию на кнопку узнать на сколько страница прокручена по вертикали;
  2. Дана страница с горизонтальной полосой прокрутки. По нажатию на кнопку узнать на сколько страница прокручена по горизонтали;
  3. Дана страница с горизонтальной и вертикальной полосами прокрутки. По нажатию на кнопку прокрутить ее в точку 300px слева, 500px сверху;
  4. Дана страница с вертикальной полосой прокрутки. По нажатию на кнопку прокрутить на 300px вниз от текущего положения;
  5. По нажатию на кнопку прокрутить страницу до самого низа.

**Описание выполнения задания:**

Листинг 9.1 - создать кнопку, по нажатию которой выведется на экран текущее количество пикселей прокрученных по вертикали.

    <style>

        body {

            height: 2500px;

            overflow: scroll;

        }

        input {

            position: fixed;

        }

    </style>

    <div id="t">

        <input type="button" id="btn" value="Нажми меня">

    </div>

    <script>

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", () => {

            alert("pageYOffset: " + window.pageYOffset)

        });

    </script>

Результат выполнения кода на странице (см. Рисунок 9.1.1):

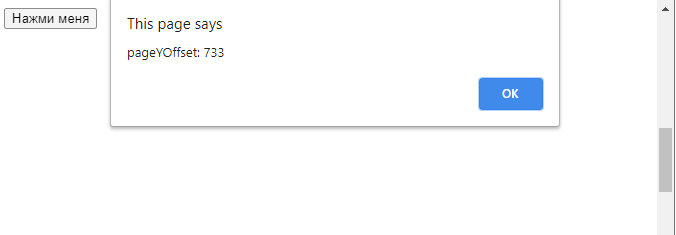


Рисунок 9.1.1

Листинг 9.2 - создать кнопку, по нажатию которой выведется на экран текущее количество пикселей прокрученных по горизонтали.

    <style>

        body {

            height: 2500px;

            width: 2500px;

            overflow: scroll;

        }

        input {

            position: fixed;

        }

    </style>

    <div id="t">

        <input type="button" id="btn" value="Нажми меня">

    </div>

    <script>

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", () => {

            alert("pageXOffset: " + window.pageXOffset)

        });

    </script>

Результат выполнения кода на странице (см. Рисунок 9.2):

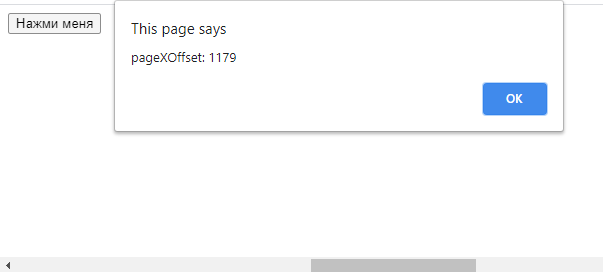


Рисунок 9.2

Листинг 9.3 - прокрутить страницу в заданные координаты.

    <style>

        body {

            height: 2500px;

            width: 2500px;

            overflow: scroll;

        }

        input {

            position: fixed;

        }

    </style>

    <div id="t">

        <input type="button" id="btn" value="Нажми меня">

    </div>

    <script>

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", () => {

            window.scrollTo(300,500);

            alert("pageXOffset: " + window.pageXOffset + ", pageYOffset: " + window.pageYOffset)

        });

    </script>

Результат выполнения кода на странице (Рисунок 9.3):

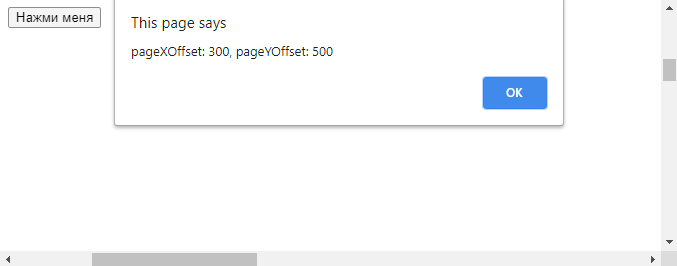


Рисунок 9.3

Листинг 9.4 - прокрутить страницу от заданного положения на 300px по вертикали.

    <style>

        body {

            height: 2500px;

            width: 2500px;

            overflow: scroll;

        }

        input {

            position: fixed;

        }

    </style>

    <div id="t">

        <input type="button" id="btn" value="Нажми меня">

    </div>

    <script>

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", () => {

            alert("pageXOffset: " + window.pageXOffset + ", pageYOffset: " + window.pageYOffset)

            window.scrollBy(0,300);

            alert("pageXOffset: " + window.pageXOffset + ", pageYOffset: " + window.pageYOffset)

        });

    </script>

Результат выполнения кода на странице (см. Рисунок 9.4):

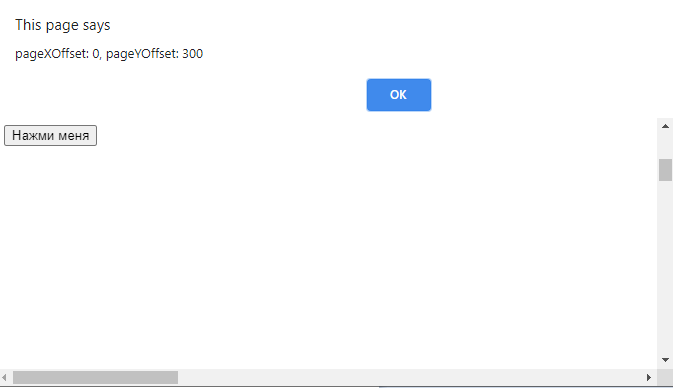


Рисунок 9.4

Листинг 9.5 - по нажатию на кнопку прокрутить страницу до самого низа

    <style>

        body {

            height: 2500px;

            width: 2500px;

            overflow: scroll;

        }

        input {

            position: fixed;

        }

    </style>

    <div id="t">

        <input type="button" id="btn" value="Нажми меня">

    </div>

    <script>

        let btn = document.querySelector("#btn");

        btn.addEventListener("click", () => {

            window.scrollBy(0,document.body.scrollHeight);

        });

    </script>

Результат выполнения кода на странице (Рисунок 9.5):

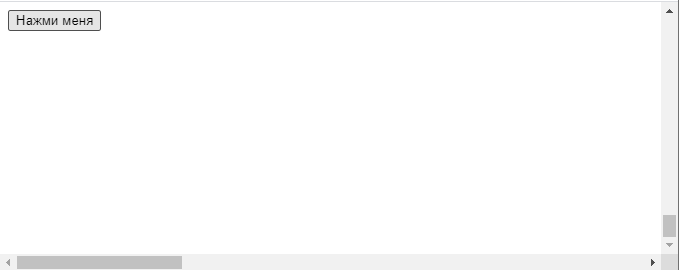


Рисунок 9.5

**Задача 10.**

Добавление кнопки закрытия: Создать список из 3 сообщений друг под другом (1 сообщение – один отдельный абзац текста). При помощи JavaScript для каждого сообщения добавить в верхний правый угол кнопку закрытия.

**Описание выполнения задания:**

Листинг 10 –создать список и кнопку, с помощью которой можно удалить абзац.

<div>

    <div class="elem">

       <p>1</p>

    </div>

    <div class="elem">

       <p>2</p>

    </div>

    <div class="elem">

       <p>3</p>

    </div>

 </div>

 <script type="text/javascript">

     let a = document.getElementsByClassName("elem");

     for(let i of a) {

         i.insertAdjacentHTML('afterbegin', '<button>Close</button>');

         i.firstChild.addEventListener("click", function() {

             i.remove();

         });

     }

 </script>

Результат выполнения кода на странице (см. Рисунок 10.1-10.2):

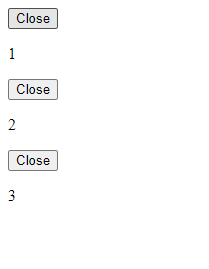


Рисунок 10.1

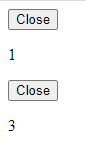


Рисунок 10.2

**Задача 11.**

Карусель: Создать «Карусель» –- ленту изображений, которую можно листать влево-вправо нажатием на стрелки, отображаемые по левой и правой сторонам карусели.

**Описание выполнения задания:**

Листинг 11 – создать карусель изображений с возможностью листать в обе стороны.

    <div id="gallery">

        <div class="photos">

            <img src="pic1.jpg">

            <img src="pic2.jpg">

            <img src="pic3.jpg">

            <img src="pic4.jpg">

            <img src="1.jpg">

        </div>

        <div class="buttons">

            <button class="prev">&#8592</button>

            <button class="next">&#8594</button>

        </div>

    </div>

    <script>

        let btn\_prev = document.querySelector('#gallery .buttons .prev');

        let btn\_next = document.querySelector('#gallery .buttons .next');

        let images = document.querySelectorAll('#gallery .photos img');

        let i = 0; // номер текущей картинки, на экране

        btn\_next.onclick = function() {

            images[i].style.display = "none";

            i++;

            if(i >= images.length){

            i = 0; // переменная i равна 0

            }

            images[i].style.display = 'block';  // прячем текущую картину

        }

        btn\_prev.onclick = function(){

            images[i].style.display = 'none';

            i = i - 1;

            if(i < 0){

                i = images.length - 1;

            }

            images[i].style.display = 'block';

        }

    </script>

<style>

#gallery{

    position: relative;

    width: 500px;

    margin: 20px auto;

    text-align: center;

}

#gallery .photos img{

    width: 100%;

    height:450px;

    display: none;

}

#gallery .photos img:first-child{

    display: block;

}

.buttons {

        margin-top: 20px;

}

button {

    position: absolute;

    transform: translate(0,-50%);

    background-color: transparent;

    border: none;

    color: white;

    font-size: 24pt;

    outline:none;

   top:50%;

   cursor: pointer;

}

.prev {

    left:0%;

}

.next {

    right:0%;

}

</style>

Результат выполнения кода на странице (см. Рисунок 11.1-11.2):



Рисунок 11.1

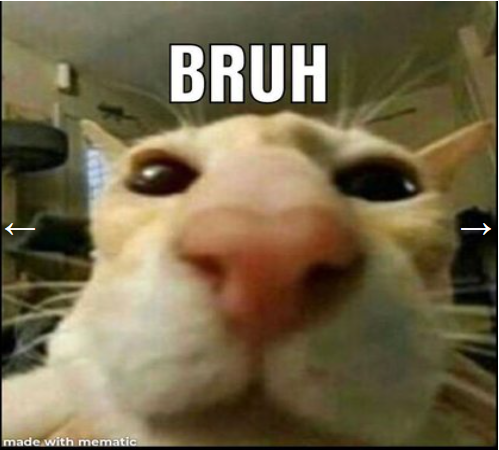


Рисунок 11.2

**Задача 12.**

Раскрывающееся дерево: Создать «дерево» (в виде многоуровневых списков), которое по клику на заголовок скрывает и показывает потомков. Использовать только один обработчик событий (применить делегирование).

**Описание выполнения задания:**

Листинг 12.1 – создать скрывающиеся/раскрывающиеся дерево в виде многоуровневых списков.

    <style>

        .tree span:hover {

          font-weight: bold;

        }

        .tree span {

          cursor: pointer;

        }

      </style>

    <ul class="tree" id="tree">

        <li>Лист 1

          <ul>

            <li>Лист 1.1

              <ul>

                <li>элемент</li>

                <li>элемент</li>

                <li>элемент</li>

                <li>элемент</li>

              </ul>

            </li>

            <li>Лист 1.2

              <ul>

                <li>элемент</li>

                <li>элемент</li>

                <li>элемент</li>

              </ul>

            </li>

          </ul>

        </li>

        <li>Лист 2

          <ul>

            <li>Лист 2.1

              <ul>

                <li>элемент</li>

                <li>элемент</li>

              </ul>

            </li>

            <li>Лист 2.2

              <ul>

                <li>элемент</li>

              </ul>

            </li>

          </ul>

        </li>

      </ul>

      <script>

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

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

          li.prepend(span);

          span.append(span.nextSibling);

        }

        tree.onclick = function(event) {

          if (event.target.tagName != 'SPAN') {

            return;

          }

          let childrenContainer = event.target.parentNode.querySelector('ul');

          if (!childrenContainer) return;

          childrenContainer.hidden = !childrenContainer.hidden;

        }

      </script>

Результат выполнения кода на странице (см. Рисунок 12.1-12.2):

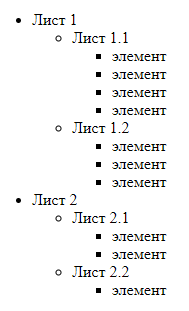


Рисунок 12.1

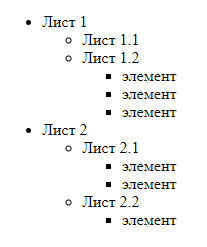


Рисунок 12.2

**Задача 13.**

Поведение "подсказка": Написать 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. В этой задаче предполагается, что во всех элементах с атрибутом data-tooltip – только текст. То есть, в них нет вложенных тегов.

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.

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

В один момент может быть показана только одна подсказка.

**Описание выполнения задания:**

Листинг 13 – создать всплывающие мобильные подсказки для элементов.

body {

      height: 2000px;

    }

    .tooltip {

      position: fixed;

      padding: 10px 20px;

      border: 1px solid #b3c9ce;

      border-radius: 4px;

      text-align: center;

      font: italic 14px/1.3 sans-serif;

      color: #333;

      background: #fff;

      box-shadow: 3px 3px 3px rgba(0, 0, 0, .3);

    }

  </style>

<body>

    <p>Lorem ipsum dolor sit amet consectetur.</p>

    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit.</p>

    <button data-tooltip="Очень длинная подсказка 1 сказка">Короткая кнопка</button>

    <button data-tooltip="Подсказка 2">Ещё кнопка</button>

     <script>

        let tooltipElem;

        document.addEventListener("mouseover", () => {

            let target = event.target;

            let tooltipHtml = target.dataset.tooltip;

            if (!tooltipHtml) return;

            tooltipElem = document.createElement("div");

            tooltipElem.className = "tooltip";

            tooltipElem.innerHTML = tooltipHtml;

            document.body.append(tooltipElem);

            let coords = target.getBoundingClientRect();

            let left = coords.left + (target.offsetWidth - tooltipElem.offsetWidth)/2;

            if (left < 0) left = 0;

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

            if (top < 0) {

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

            }

            tooltipElem.style.left = left + "px";

            tooltipElem.style.top = top + "px";

        });

        document.addEventListener("mouseout", () => {

            if (tooltipElem) {

                tooltipElem.remove();

                tooltipElem = null;

            }

        });

    </script>

Результат выполнения кода на странице (см. Рисунок 13.1-13.2):

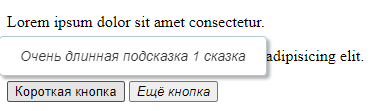


Рисунок 13.1

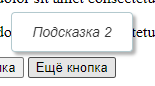


Рисунок 13.2

**Задача 14.**

Переход по ссылке: Сделать так, чтобы при клике на ссылки внутри элемента id="contents" пользователю выводился вопрос о том, действительно ли он хочет покинуть страницу, и если он не хочет, то прерывать переход по ссылке.

* Содержимое #contents может быть загружено динамически и присвоено при помощи innerHTML. Так что найти все ссылки и поставить на них обработчики нельзя. Использовать делегирование.
* Содержимое может иметь вложенные теги, в том числе внутри ссылок, например, <a href=".."><i>...</i></a>.

**Описание выполнения задания:**

Листинг 14 – перехватить переход по ссылке и спросить хочет ли пользователь действительно перейти по ссылке.

<style>

    .linkWindow {

        position: fixed;

        border: 1px solid black;

        left: calc(50% - 125px);

        top: 20px;

        width: 250px;

        text-align: center;

        padding: 10px;

    }

    .linkWindow input[type="button"]:nth-of-type(1){

        margin-right: 20px;

    }

</style>

<body>

    <div id="contents">

    </div>

    <script>

        let div = document.querySelector("#contents");

        let linkElem;

        for(let i = 1; i < 4; i++) {

            let a = document.createElement("a");

            a.href = "https://google.com";

            a.style.display = "block";

            a.innerHTML = "Ссылка " + i;

            div.append(a);

        }

        document.onclick = function() {

            let target = event.target;

            if (target.dataset.func == "go") {

                let bruh = document.getElementsByClassName("linkWindow");

                for (i of bruh) {

                    i.remove();

                }

                document.location.href = "http://www.google.com";

                return true;

            }

            else if (target.dataset.func == "stop") {

                let bruh = document.getElementsByClassName("linkWindow");

                for (i of bruh) {

                    i.remove();

                }

                //perm.remove();

            }

            else if (target.nodeName == "A"){

               var perm = document.createElement("div");

                perm.className = "linkWindow";

            let p = document.createElement("p");

            p.append("Вы хотите перейти на данную страницу?");

            p.insertAdjacentHTML("beforeend", "<br>" + target.getAttribute('href'))

            let btnY = document.createElement("input");

            let btnN = document.createElement("input");

            btnY.type = "button";

            btnY.dataset.func = "go";

            btnN.type = "button";

            btnN.dataset.func = "stop";

            btnY.value = "Перейти";

            btnN.value = "Остаться";

            perm.append(p,btnY,btnN);

            document.body.prepend(perm);

            }

            else return false;

            return false;

        }

    </script>

Результат выполнения кода на странице (см. Рисунок 14.1):

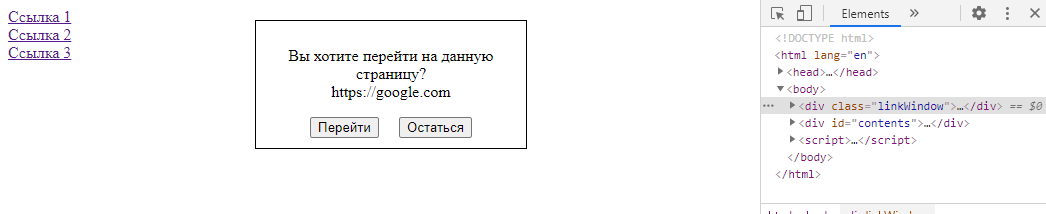


Рисунок 14.1

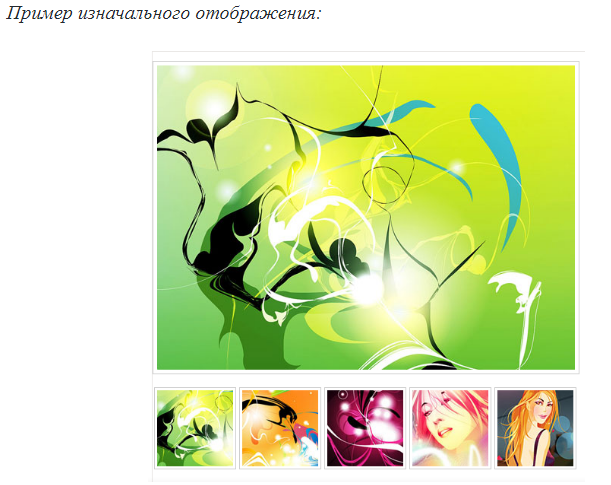
При нажатии кнопки "Остаться" окошко удаляется и остается изначальное полотно(см. Рисунок 14.2):



Рисунок 14.2

**Задача 15.**

Галерея изображений: Создать галерею изображений, в которой основное изображение изменяется при клике на уменьшенный вариант. Использовать делегирование.



**Описание выполнения задания:**

Листинг 15 – создать галерею изображений, где выбранное изображение имеет увеличенный вариант, а остальные уменьшенный.

<style>

    body {

  margin: 0;

  padding: 0;

  font: 75%/120% sans-serif;

}

h2 {

  font: bold 190%/100% sans-serif;

  margin: 0 0 .2em;

}

h2 em {

  font: normal 80%/100% sans-serif;

  color: #999999;

}

#largeImg {

  border: solid 1px #ccc;

  width: 550px;

  height: 400px;

  padding: 5px;

}

#thumbs a img {

  width: 100%;

  height: 100%;

}

#thumbs a {

  border: solid 1px #ccc;

  width: 100px;

  height: 100px;

  padding: 3px;

  margin: 2px;

  float: left;

}

#thumbs a:hover {

  border-color: #FF9900;

}

#thumbs li {

  list-style: none;

}

#thumbs {

  margin: 0;

  padding: 0;

}

</style>

<body>

    <p><img id="largeImg" src="pic1.jpg" alt="Large image"></p>

  <ul id="thumbs">

    <li>

      <a href="pic2.jpg" title="Image 2"><img src="pic2.jpg"></a>

    </li>

    <li>

      <a href="pic3.jpg" title="Image 3"><img src="pic3.jpg"></a>

    </li>

    <li>

      <a href="pic4.jpg" title="Image 4"><img src="pic4.jpg"></a>

    </li>

    <li>

      <a href="1.jpg" title="Image 5"><img src="1.jpg"></a>

    </li>

    <li>

      <a href="2.jpeg" title="Image 6"><img src="2.jpeg"></a>

    </li>

  </ul>

  <script>

    thumbs.onclick = function(event) {

      let thumbnail = event.target.closest('a');

      if (!thumbnail) return;

      showThumbnail(thumbnail.href, thumbnail.title);

      event.preventDefault();

    }

    function showThumbnail(href, title) {

      largeImg.src = href;

      largeImg.alt = title;

    }

  </script>

Результат выполнения кода на странице (см. Рисунок 15.1-15.2):

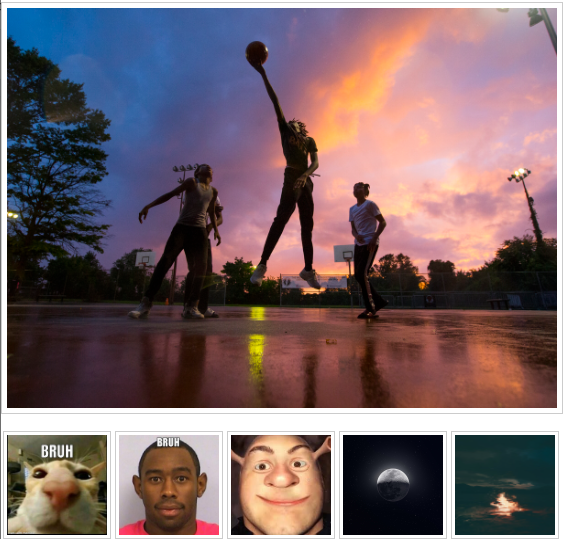


Рисунок 15.1

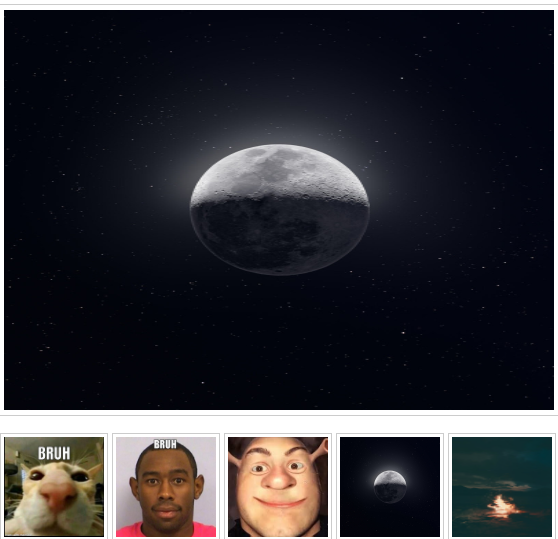
\

Рисунок 15.2