|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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 | | Иноземцев Н.С. | |
|  | |  | |
| 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.

let a = document.getElementById("btn");

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

window.open('dialog.html');

});

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.

let a = document.getElementById("btn2");

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

window.open('dialog.html');

});

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.

<p id="p1">Какой-то текст</p>

<button id="btn">Кнопка</button>

<script>

let a = document.getElementById("btn");

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

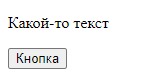
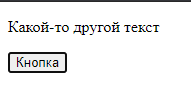
let b = document.getElementById("p1");

b.innerHTML = "Какой-то другой текст";

});

</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 id="img1" src="https://www.proplay.ru/images/users/gallery/127866/187721\_l.jpg">

<button id="btn2">Кнопка2</button>

<script>

let a = document.getElementById("btn2");

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

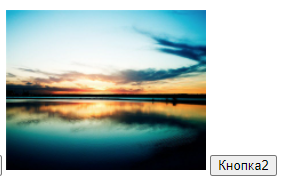
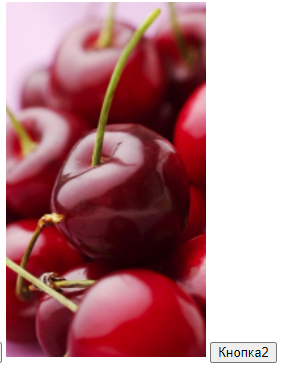
let b = document.getElementById("img1");

b.src = "https://proprikol.ru/wp-content/uploads/2020/02/kachat-krasivye-kartinki-na-telefon-besplatno-1.jpg";

});

</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.

<p id="p1">Какой-то текст</p>

<script>

function c() {

a.innerHTML = "Какой-то текст";

}

document.addEventListener("click", c);

let a = document.getElementById("p1");

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

this.innerHTML = "Какой-то другой текст";

document.removeEventListener("click", c);

});

</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">Кнопка1</button><br>

<button id="btn2">Кнопка2</button>

<script>

let a = document.getElementById("btn1");

let a1 = document.getElementById("btn2");

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

this.disabled = true;

});

a1.addEventListener("click", function() {

a.disabled = false;

});

</script>

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="btn">0</button>

<script>

let a = document.getElementById("btn");

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

let i = (Number)(a.innerHTML);

i++;

a.innerHTML = i;

});

</script>

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.

<p id="p1" style="background: blue">Какой-то текст</p>

<button id="btn1">Кнопка</button>

<script>

let a = document.getElementById("p1");

let b = document.getElementById("btn1");

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

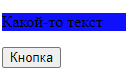
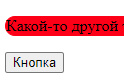
a.style.cssText = "width: 200px; background: red; border: solid red 1px; border-radius: 10px ";

a.innerHTML = "Какой-то другой текст";

});

</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.

<p id="p2">Какой-то текст</p>

<button id="btn2">Кнопка1</button>

<button id="btn21">Кнопка2</button>

<script>

let a = document.getElementById("p2");

let b = document.getElementById("btn2");

let c = document.getElementById("btn21");

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

a.style.opacity = "0";

});

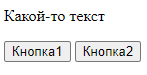
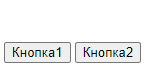
c.addEventListener("click", function() {

a.style.opacity = "1";

});

</script>

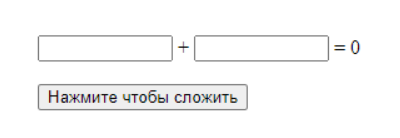
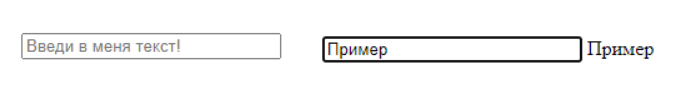
Результат выполнения кода на странице (Рисунок 4.2.1-4.2.2):

**** ****

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

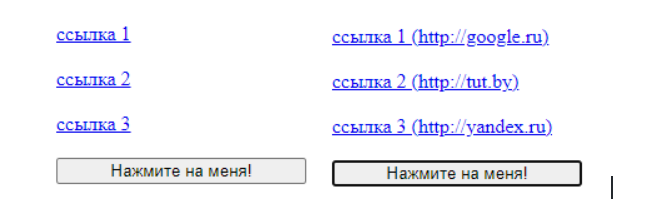
**Задача 5.**

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="p1">Какой-то текст</p>

<button id="btn1">Кнопка</button>

<script>

let a = document.getElementById("p1");

let b = document.getElementById("btn1");

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

a.outerHTML = "<p style=\"font-size: 24pt;\"><b>Какой-то другой текст</b></p>";

})

</script>

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



Рисунок 5.1.1

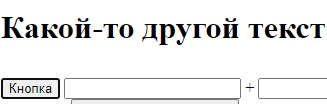


Рисунок 5.1.2

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

<input type="text" id="in1" value>

+

<input type="text" id="in2" value>

=

<span id="p2">0</span>

<br>

<button id="btn2" style="margin-left: 70px">Нажмите чтобы сложить</button><br>

<script>

let a = document.getElementById("btn2");

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

let in1 = document.getElementById("in1");

let in2 = document.getElementById("in2");

let num1 = (Number)(in1.value);

let num2 = (Number)(in2.value);

let b = document.getElementById("p2");

let res = num1 + num2;

b.innerHTML = res;

});

</script>

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

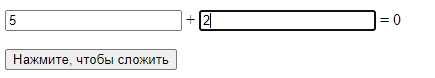


Рисунок 5.2.1

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

<input type="text" id="in3" value="Введи в меня текст!"> <span id="p3"></span><br>

<script>

let a = document.getElementById("in3");

let b = document.getElementById("p3");

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

b.innerHTML = e.target.value;

});

</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 id="a1" href="https://www.google.ru/">ссылка 1</a><br>

<a id="a2" href="https://www.tut.by/">ссылка 2</a><br>

<a id="a3" href="https://yandex.ru/">ссылка 3</a><br>

<button id="btn3">Нажми на меня!</button>

<script>

let a1 = document.getElementById("a1");

let a2 = document.getElementById("a2");

let a3 = document.getElementById("a3");

let b = document.getElementById("btn3");

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

a1.innerHTML += a1.href;

a2.innerHTML += a2.href;

a3.innerHTML += a3.href;

});

</script>

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

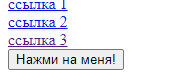


Рисунок 5.4.1

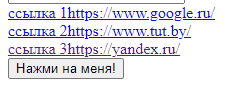
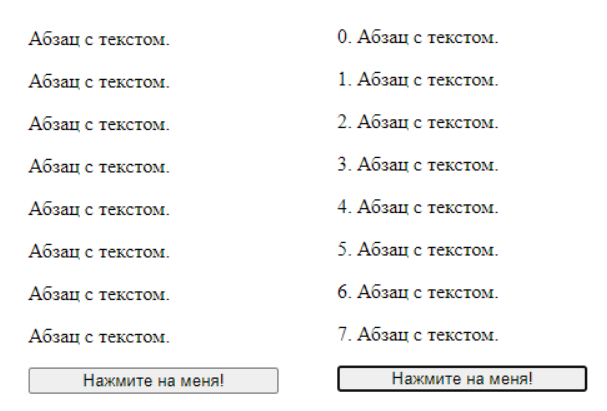


Рисунок 5.4.2

**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>Какой-то текст</p>

<p>Какой-то текст</p>

<p>Какой-то текст</p>

<button id="btn1">Кнопка</button>

<script type="text/javascript">

//task 6.1

let a = document.getElementById("btn1");

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

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

b[0].innerHTML = "Какой-то другой текст";

b[1].innerHTML = "Какой-то другой текст";

});

</script>

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

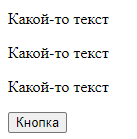


Рисунок 6.1.1

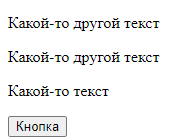


Рисунок 6.1.2

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>

<button id="btn1">Нажми на меня!</button>

<script type="text/javascript">

//task 6.2

let a = document.getElementById("btn1");

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

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

for(let i = 0; i < b.length; i++) {

b[i].innerHTML = i + "." + b[i].innerHTML;

}

});

</script>

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

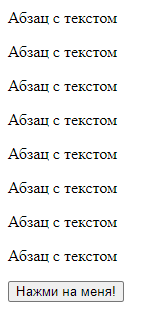


Рисунок 6.2.1

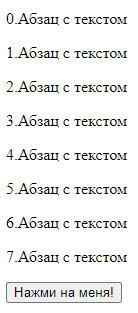


Рисунок 6.2.2

**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.

<button id="btn">Кнопка</button>

<p id="p1"></p>

<script type="text/javascript">

//task 7.1

let i = 0;

let a = document.getElementById("btn");

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

let time = setInterval(fun, 1000);

});

function fun() {

let b = document.getElementById("p1");

b.innerHTML += "<br>" + i;

i++;

}

</script>

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



Рисунок 7.1.1

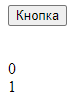


Рисунок 7.1.2

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

<button id="btn">Кнопка</button>

<button id="btn2">Кнопка2</button>

<p id="p1"></p>

<script type="text/javascript">

//task 7.2

let i = 0;

let a = document.getElementById("btn");

let a2 = document.getElementById("btn2");

let time;

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

time = setInterval(fun, 1000);

});

a2.addEventListener("click", function() {

clearInterval(time);

});

function fun() {

let b = document.getElementById("p1");

b.innerHTML += "<br>" + i;

i++;

}

</script>

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



Рисунок 7.2.1

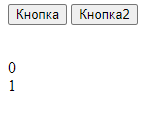


Рисунок 7.2.2.

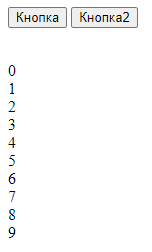


Рисунок 7.2.3

Листинг 7.3 - создать страницу с тикающими цифровыми часами.

<p id="p1"></p>

<script type="text/javascript">

//task 7.3

let time = setInterval(fun, 1000);

function fun() {

let date = new Date();

let a = document.getElementById("p1");

a.innerHTML = date.getHours() + ":" + date.getMinutes() + ":" + date.getSeconds();

}

</script>

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



Рисунок 7.3.1

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

<div>

<img src="https://www.ixbt.com/img/n1/news/2019/5/3/chrome-73-mode-sombre-android\_large.jpg" height="400px" style="position: absolute">

<img src="https://avatarko.ru/img/kartinka/1/avatarko\_anonim.jpg" height="400px" style="position: absolute">

<img src="https://i.pinimg.com/originals/f7/0f/82/f70f827e33a61a247663e7aeb282f895.jpg" height="400px" style="position: absolute">

<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQI-SaYWlXmVicHWYEEpRgrmFir507tWQk3pA&usqp=CAU" height="400px" style="position: absolute">

</div>

<script type="text/javascript">

//task 7.4

let imgs = document.getElementsByTagName("img");

let n = 1;

setInterval(showSlide, 3000);

setInterval(next, 3000);

function next() {

n++;

}

function showSlide() {

if(n > imgs.length){

n = 1;

}

for(let i = 0; i < imgs.length; i++) {

imgs[i].style.opacity = "0";

}

imgs[n-1].style.opacity = "1";

}

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



Рисунок 7.4.1



Рисунок 7.4.2

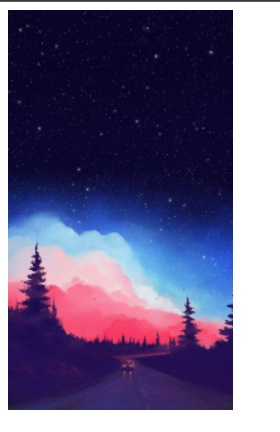


Рисунок 7.4.3



Рисунок 7.4.4

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

<div>

<img src="https://www.ixbt.com/img/n1/news/2019/5/3/chrome-73-mode-sombre-android\_large.jpg" height="400px" style="position: absolute; opacity: 1;" >

<img src="https://avatarko.ru/img/kartinka/1/avatarko\_anonim.jpg" height="400px" style="position: absolute">

<img src="https://i.pinimg.com/originals/f7/0f/82/f70f827e33a61a247663e7aeb282f895.jpg" height="400px" style="position: absolute; opacity: 0;">

<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQI-SaYWlXmVicHWYEEpRgrmFir507tWQk3pA&usqp=CAU" height="400px" style="position: absolute ; opacity: 0;">

<img src="https://img13.postila.ru/resize?w=600&src=%2Fdata%2F21%2F33%2Fcf%2Fa1%2F2133cfa14b903cd9d0fd64046b2c459bae1458d0bfd4b10c0b18a5941eb25779.jpg" height="400px" style="position: absolute; opacity: 0;">

</div>

<button id="start">start</button>

<button id="stop">stop</button>

<script type="text/javascript">

//task 7.5

let imgs = document.getElementsByTagName("img");

let n = 1;

let start = document.getElementById("start");

let st, ts;

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

st = setInterval(showSlide, 1000);

ts = setInterval(next, 1000);

});

let stop = document.getElementById("stop");

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

clearInterval(st);

clearInterval(ts);

});

function next() {

n++;

}

function showSlide() {

if(n > imgs.length){

n = 1;

}

for(let i = 0; i < imgs.length; i++) {

imgs[i].style.opacity = "0";

}

imgs[n-1].style.opacity = "1";

}

</script>

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



Рисунок 7.5.1

Листинг 7.6 -создать счетчик оставшегося времени до полуночи.

<p id="p1"></p>

<script type="text/javascript">

let fin = {

'year': '2020',

'month': '12',

'day': '13',

'hour': '00',

'min': '00',

'sec': '00'

}

let a = document.getElementById('p1')

function timeLeft(to, from) {

return to - from

}

int1 = setInterval(function () {

let finDate = new Date(`2020-12-06T00:00:00`);

let left = timeLeft(finDate, new Date());

if (left > 0) {

let leftTime = new Date(left)

a.innerText = `${leftTime.getUTCDate() - 1}/${leftTime.getUTCMonth()}/${leftTime.getUTCFullYear() - 1970} ${leftTime.getUTCHours()}:${leftTime.getUTCMinutes()}:${leftTime.getUTCSeconds()}`

} else {

clearInterval(int1)

alert('Время закончилось')

}

}, 1000)

</script>

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



Рисунок 7.6.1

**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):

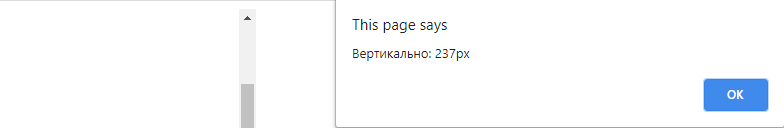


Рисунок 8.1.1

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):

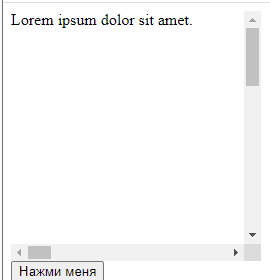


Рисунок 8.2.1

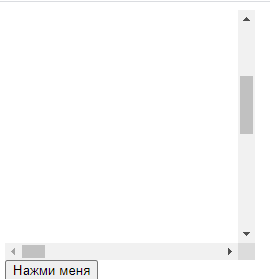


Рисунок 8.2.2

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):

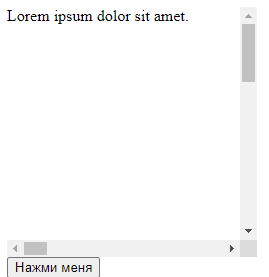


Рисунок 8.3.1

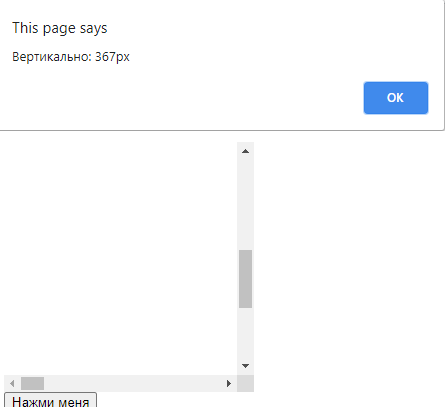


Рисунок 8.3.2

**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):

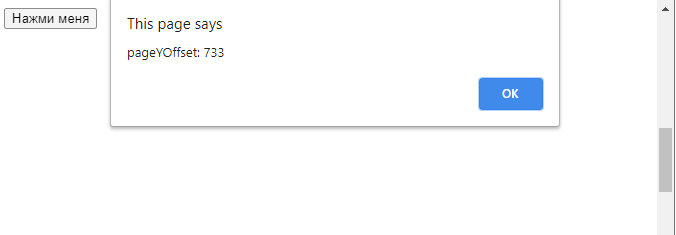


Рисунок 9.1.1

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):

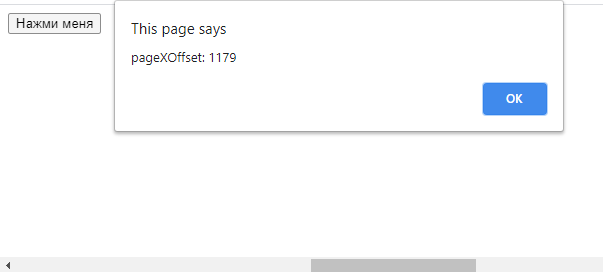


Рисунок 9.2

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):

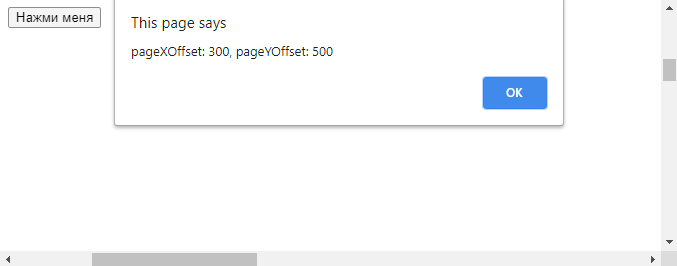


Рисунок 9.3

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):

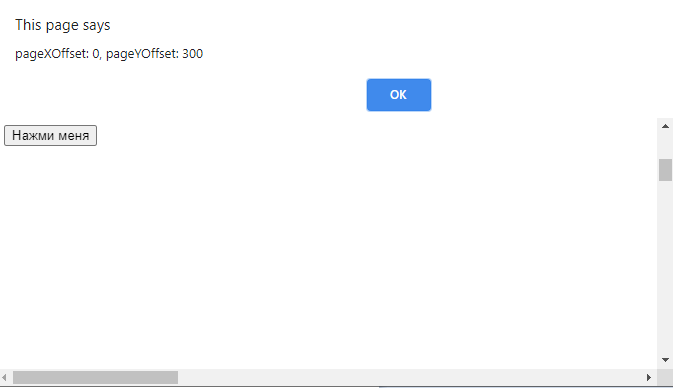


Рисунок 9.4

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):

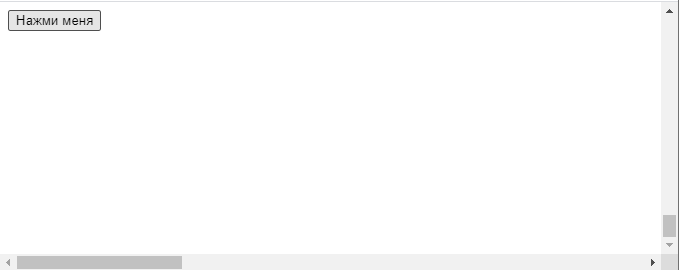


Рисунок 9.5

**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):

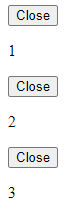


Рисунок 10.1

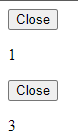


Рисунок 10.2

**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>

<img src="https://www.ixbt.com/img/n1/news/2019/5/3/chrome-73-mode-sombre-android\_large.jpg" height="400px" style="position: absolute; opacity: 1;" >

<img src="https://avatarko.ru/img/kartinka/1/avatarko\_anonim.jpg" height="400px" style="position: absolute">

<img src="https://i.pinimg.com/originals/f7/0f/82/f70f827e33a61a247663e7aeb282f895.jpg" height="400px" style="position: absolute; opacity: 0;">

<img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQI-SaYWlXmVicHWYEEpRgrmFir507tWQk3pA&usqp=CAU" height="400px" style="position: absolute ; opacity: 0;">

<img src="https://img13.postila.ru/resize?w=600&src=%2Fdata%2F21%2F33%2Fcf%2Fa1%2F2133cfa14b903cd9d0fd64046b2c459bae1458d0bfd4b10c0b18a5941eb25779.jpg" height="400px" style="position: absolute; opacity: 0;">

</div>

<button id="left"><-</button>

<button id="right">-></button>

<script type="text/javascript">

//task 7.5

let imgs = document.getElementsByTagName("img");

let n = 1;

let left = document.getElementById("left");

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

previous();

showSlide();

});

let right = document.getElementById("right");

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

next();

showSlide();

});

function previous() {

n--;

}

function next() {

n++;

}

function showSlide() {

if(n > imgs.length){

n = 1;

}

if(n < 1) {

n = imgs.length;

}

for(let i = 0; i < imgs.length; i++) {

imgs[i].style.opacity = "0";

}

imgs[n-1].style.opacity = "1";

}

</script>

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



Рисунок 11.1

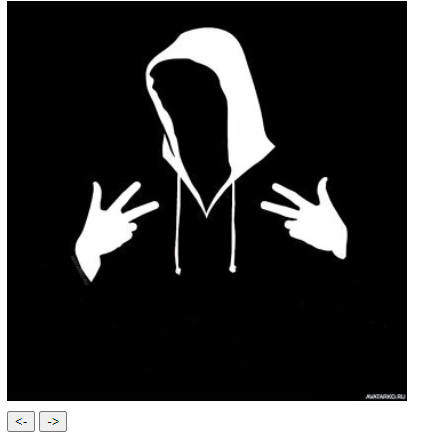


Рисунок 11.2

**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.

<ul class="tree" id="tree">

<li>1

<ul>

<li>1.1

<ul>

<li>1.1.1</li>

</ul>

</li>

<li>1.2

<ul>

<li>1.2.1</li>

</ul>

</li>

</ul>

</li>

<li>2

<ul>

<li>2.1

<ul>

<li>2.1.1</li>

<li>2.1.2</li>

<li>2.1.3</li>

</ul>

</li>

<li>2.2

<ul>

<li>2.2.1</li>

<li>2.2.2</li>

</ul>

</li>

</ul>

</li>

</ul>

<script type="text/javascript">

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

let span = document.createElement('span')

i.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):

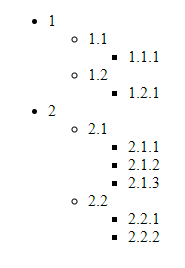


Рисунок 12.1

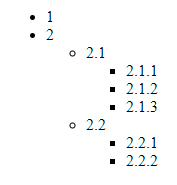


Рисунок 12.2

**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.

<span data-tooltip="Подсказка 1">Текст 1</span>

<span data-tooltip="Подсказка 2">Текст 2</span>

<script>

let tooltipElem;

document.onmouseover = function(event) {

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.onmouseout = function(e) {

if (tooltipElem) {

tooltipElem.remove();

tooltipElem = null;

}

};

</script>

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

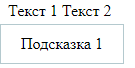


Рисунок 13.1

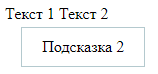


Рисунок 13.2

**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.

<p> <a id="contents" href="https://ru.wikipedia.org/wiki/">ссылка</a></p>

<script>

contents.onclick = function(event) {

function handleLink(href) {

let isLeaving = confirm(`Перейти на ${href}?`);

if (!isLeaving) return false;

}

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

if (target && contents.contains(target)) {

return handleLink(target.getAttribute('href'));

}

};

</script>

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

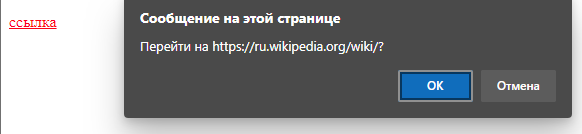


Рисунок 14.1

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

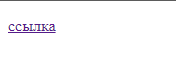
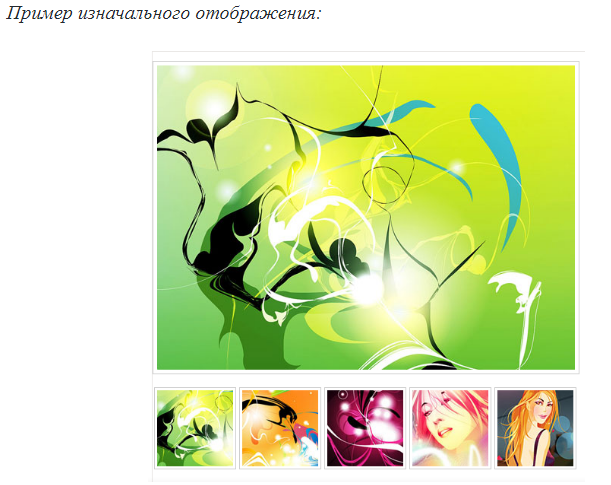


Рисунок 14.2

**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>

ul > li {

display: inline;

}

ul > li img {

max-height: 150px;

}

p img {

height: 600px;

}

</style>

<p><img id="largeImg" src="https://www.ixbt.com/img/n1/news/2019/5/3/chrome-73-mode-sombre-android\_large.jpg" ></p>

<ul id="items">

<li>

<a href="https://www.ixbt.com/img/n1/news/2019/5/3/chrome-73-mode-sombre-android\_large.jpg"><img src="https://www.ixbt.com/img/n1/news/2019/5/3/chrome-73-mode-sombre-android\_large.jpg"></a>

</li>

<li>

<a href="https://avatarko.ru/img/kartinka/1/avatarko\_anonim.jpg"><img src="https://avatarko.ru/img/kartinka/1/avatarko\_anonim.jpg"></a>

</li>

<li>

<a href="https://i.pinimg.com/originals/f7/0f/82/f70f827e33a61a247663e7aeb282f895.jpg" ><img src="https://i.pinimg.com/originals/f7/0f/82/f70f827e33a61a247663e7aeb282f895.jpg"></a>

</li>

<li>

<a href="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQI-SaYWlXmVicHWYEEpRgrmFir507tWQk3pA&usqp=CAU"><img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQI-SaYWlXmVicHWYEEpRgrmFir507tWQk3pA&usqp=CAU"></a>

</li>

<li>

<a href="https://img13.postila.ru/resize?w=600&src=%2Fdata%2F21%2F33%2Fcf%2Fa1%2F2133cfa14b903cd9d0fd64046b2c459bae1458d0bfd4b10c0b18a5941eb25779.jpg"><img src="https://img13.postila.ru/resize?w=600&src=%2Fdata%2F21%2F33%2Fcf%2Fa1%2F2133cfa14b903cd9d0fd64046b2c459bae1458d0bfd4b10c0b18a5941eb25779.jpg"></a>

</li>

</ul>

<script>

items.onclick = function(event) {

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

if (!elem) return;

show(elem.href);

event.preventDefault();

}

function show(href) {

largeImg.src = href;

}

</script>

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



Рисунок 15.1

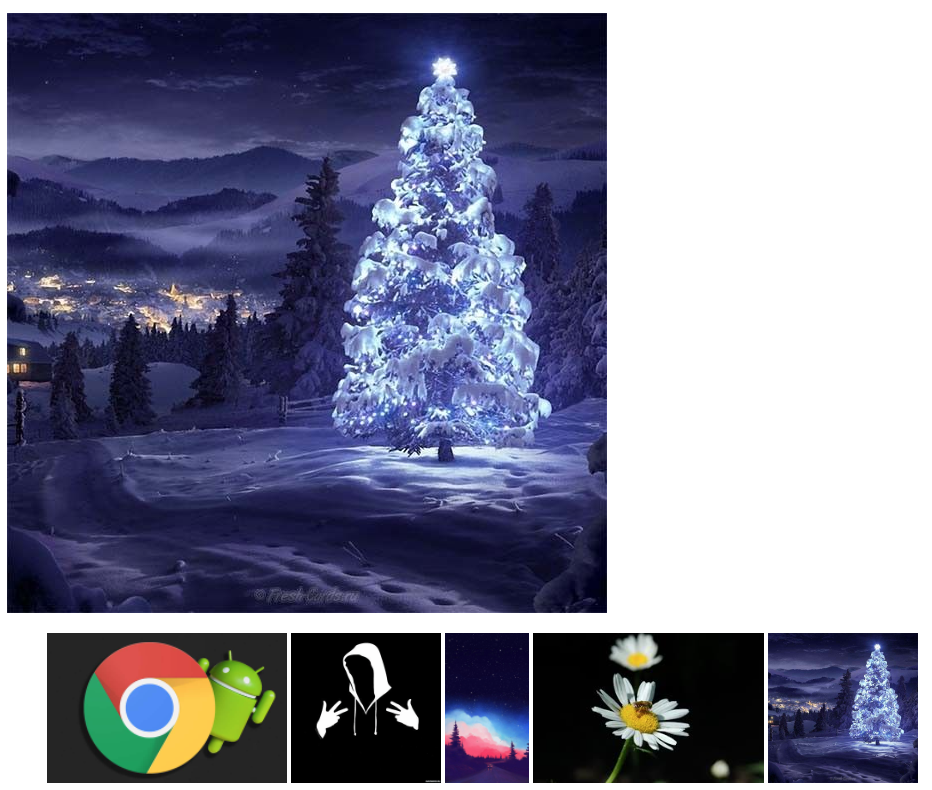
\

Рисунок 15.2