|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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 ПРАКТИЧЕСКОЙ РАБОТЕ №7** | | | |
| **по дисциплине** | | | |
| **«Разработка клиентских частей интернет-ресурсов»** | | | |
| Выполнил студент группы ИКБО-01-19 | | Соколовский И.М. | |
|  | |  | |
| Принял  *Ассистент* | | Коваленко М.А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | (подпись руководителя) | |
|  |  | |  | |
|  |  | |  | |

Москва 2020

СОДЕРЖАНИЕ

[Практическая работа №7 «Основы работы с DOM в JavaScript» 2](#_Toc59252303)

[Цель работы 2](#_Toc59252304)

[Практическое задание 1 «События через атрибуты» 2](#_Toc59252305)

[Практическое задание 2 «Метод getElementById работа с атрибутами» 3](#_Toc59252306)

[Практическое задание 3: «Работа с this» 5](#_Toc59252307)

[Практическое задание 4: «Работа с CSS» 9](#_Toc59252308)

[Практическое задание 5: «Свойства innerHTML, outerHTML» 11](#_Toc59252309)

[Практическое задание 6: «Метод getElementsByTagName» 15](#_Toc59252310)

[Практическое задание 7: «Работа с таймером» 18](#_Toc59252311)

[Практическое задание 8: «Прокрутка элемента» 29](#_Toc59252312)

[Практическое задание 9: «Прокрутка страницы» 33](#_Toc59252313)

[Практическое задание 10: «Добавление кнопки закрытия» 38](#_Toc59252314)

[Практическое задание 11: «Карусель» 41](#_Toc59252315)

[Практическое задание 12: «Раскрывающееся дерево» 46](#_Toc59252316)

[Практическое задание 13: «Поведение "подсказка"» 49](#_Toc59252317)

[Практическое задание 14: «Переход по ссылке» 52](#_Toc59252318)

[Практическое задание 15: «Галерея изображений» 54](#_Toc59252319)

[Вывод 59](#_Toc59252320)

**Практическая работа №7 «Основы работы с DOM в JavaScript»**

**Цель работы**

Ознакомиться с DOM. Овладеть навыками работы с обработчиком событий, изучить this, делегирование событий, овладеть навыком отлавливания события мыши на странице

**Практическое задание 1 «События через атрибуты»**

**Задача 1.1**

Создать страницу с кнопкой, при наведении на которую появляется всплывающее окно в браузере с вашим сообщением.

**Задача 1.2**

Создать страницу с кнопкой, при двойном нажатии на которую появляется всплывающее окно в браузере с вашим сообщением.

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<button onmouseover="alert('сначала так')"></button>

<button ondblclick="alert('а теперь так')"></button>

</body></html>

Результат работы кода:



Рисунок 1



Рисунок 2

**Практическое задание 2 «Метод getElementById работа с атрибутами»**

**Задача 2.1**

Создать страницу с заполненным вашим сообщением текстовым полем и кнопкой рядом, при нажатии на которую данное сообщение меняется на другое.

**Задача 2.2**

Создать страницу с изображением и кнопкой рядом, при нажатии на которую данное изначальное изображение меняется на другое

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<p>Введите ваш ник</p>

<input maxlength="25" size="40" type="text" value="woof" id = "plhd">

<button onclick="func()">нажми меня</button>

<br><br><br>

<img src="https://64.media.tumblr.com/34f611448fb743f9992b9ee174da4748/tumblr\_ol282sGDvy1tht5bro1\_1280.jpg" onclick="imgscr()" id="imgid" width="350" height="300">

<script>

function func(){

elem = document.getElementById('plhd');

elem.value = 'meow';

}

function imgscr(){

elemim = document.getElementById('imgid');

elemim.src = "https://64.media.tumblr.com/5220aa760121d3419d8ffc08ce1c5fda/tumblr\_pipy80YmPj1vmvaaq\_540.png";

}

</script>

</body>

</html>

Результат работы кода

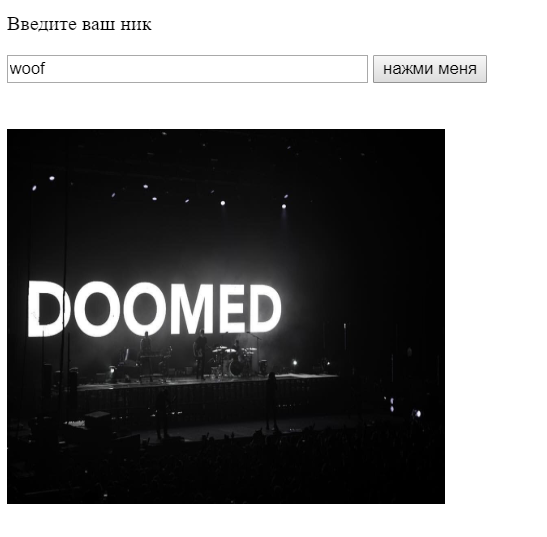


Рисунок 3



Рисунок 4

**Практическое задание 3: «Работа с this»**

**Задача 3.1**

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.

**Задача 3.2**

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.

**Задача 3.3**

Создать страницу с кнопкой, которая считает количество нажатий на нее (кол-во нажатий должно отображаться на самой копке).

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<input type="text" value="well hello there" onfocus="func1(this)" onblur="func2(this)">

<br><br>

<button id = "b1" onclick="bye(this)">Нажми на кнопку</button>

<button onclick="hi()">Теперь чини кнопку</button>

<br><br>

<input type="submit" onclick="count(this)" value="0">

<script>

function func1(elem){

elem.value = ('hi again');

}

function func2(elem){

elem.value = ('bye');

}

function bye(elem){

elem.disabled = true;

}

function hi(){

elem = document.getElementById('b1');

elem.disabled = false;

}

function count(elem){

elem.value++;

elem.disabled = false;

}

</script>

</body>

</html>Результат работы кода:

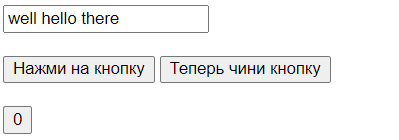


Рисунок 5

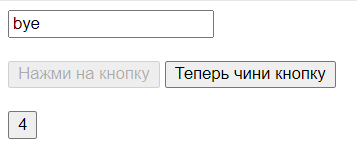


Рисунок 6

**Практическое задание 4: «Работа с CSS»**

**Задача 4.1**

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.

**Задача 4.2**

Создать страницу с заполненным вашим сообщением текстовым полем и двумя кнопками рядом, при нажатии на первую из них текстовое поле должно полостью исчезнуть, при нажатии на вторую – снова появиться в изначальном виде.

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<input type="text" value="сообщение раз" id="shiandt" style="width: 400px;">

<button onclick="change()">раз-два</button>

<br><br>

<input type="text" value="Налоги выросли" id="corpse" style="width: 400px;">

<button onclick="hide()">Это враньё</button>

<button onclick="hi()">Повтори, я не услышал</button>

<script>

function change(){

elem = document.getElementById('shiandt');

elem.style.color = 'rgb(65, 95, 167)';

elem.style.width = '200px';

elem.value = "сообщение два";

elem.style.borderRadius = '15px';

}

function hide(){

elem1 = document.getElementById('corpse');

elem1.style.display = 'none';

}

function hi(){

elem2 = document.getElementById('corpse')

elem2.style.display = 'block';

}

</script>

</body></html>Результат работы кода:

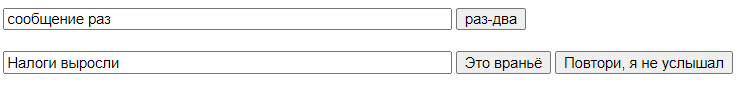


Рисунок 7

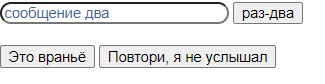


Рисунок 8

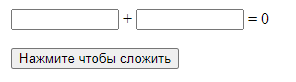
**Практическое задание 5: «Свойства innerHTML, outerHTML»**

**Задача 5.1**

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.

**Задача 5.2**

Создать страницу по данному образцу:



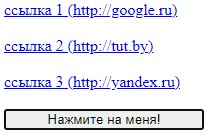
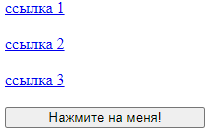
**Задача 5.3**

Создать страницу по данному образцу (до/после):



**Задача 5.4**

Создать страницу по данному образцу:



**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<p id="text">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Tempore, repudiandae eius nihil incidunt non et consectetur perspiciatis rem libero quisquam.</p>

<input onclick="change()" type="button">

<br><br>

<input type="text" value="" id="input1">

+

<input type="text" value="" id="input2">

=

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

<br><br>

<button onclick="addi()">Сложение</button>

<br><br>

<input type="text" value="" placeholder="Пример" id="test" onkeyup="go(this)">

<span id="textt"></span>

<br><br>

<a href="https://github.com/">github</a><br><br>

<a href="https://vk.com/">вк</a><br><br>

<input type="submit" value="Клик клик" onclick="shoow()">

<script>

function change(){

elem = document.getElementById('text')

elem.innerHTML = '<h1><b>meow meow meow<b><h1>';

}

function addi(){

let input1 = +document.getElementById('input1').value;

let input2 = +document.getElementById('input2').value;

let result = document.getElementById('result');

result.innerHTML = input1 + input2;

}

function go(elem){

let text = document.getElementById('textt');

text.innerHTML = elem.value;

}

function shoow(){

let elements = document.getElementsByTagName('a');

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

elements[i].innerHTML = elements[i].innerHTML+' ('+elements[i].getAttribute('href')+')';

}

}

</script>

</body>

</html>

Результат работы кода:

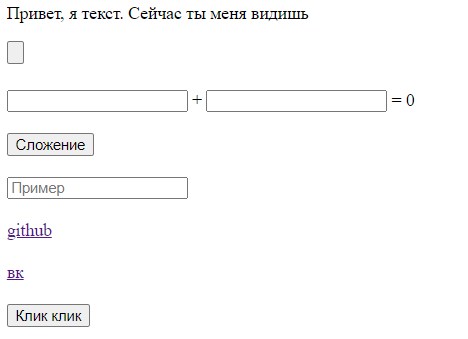


Рисунок 9

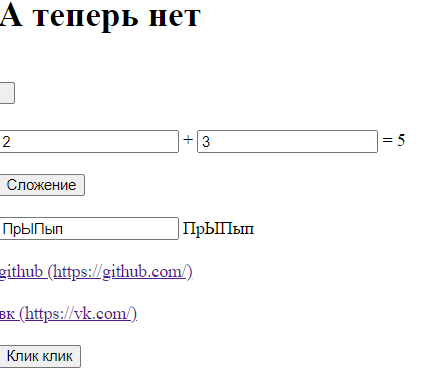


Рисунок 10

**Практическое задание 6: «Метод getElementsByTagName»**

**Задача 6.1**

Создать страницу с тремя абзацами текста и кнопкой рядом. При нажатии на кнопку текст в двух абзацах должен поменяться, в третьем – остаться неизменным.

**Задача 6.2**

Создать страницу по данному образцу (при нажатии на кнопку текст в каждом абзаце поменяется - спереди добавится порядковый номер абзаца

**Описание выполнения задания:**

Листинг 85.1 – Изменение внутреннего HTML по клику с помощью вызова через название тега

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<!--6.1-->

<p>My secrets are burning a hole through my heart

And my bones catch a fever

When it cuts you up this deep

It's hard to find a way to breathe</p>

<p>Your eyes are swallowing me

Mirrors start to whisper, shadows start to sing

My skin's smothering me

Help me find a way to breathe</p>

<p>Time stood still the way it did before

It's like I'm sleepwalking

Fell into another hole again

It's like I'm sleepwalking</p>

<button onclick="func()">click</button>

<script>

<!--6.1-->

function func(){

const elem = document.getElementsByTagName('p');

for (var i = 0; i < 2; i++) {

elem[i].innerHTML = 'boom';

}

}

</script>

</body>

</html>

Результат работы кода:

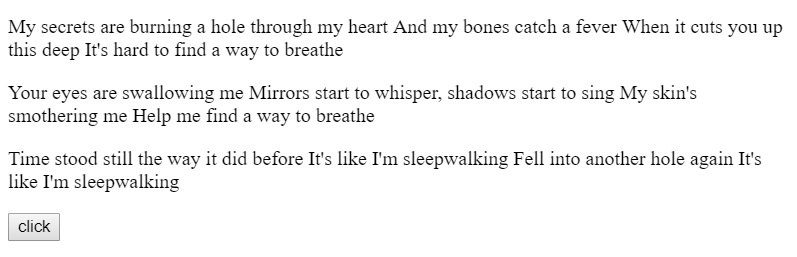


Рисунок 11

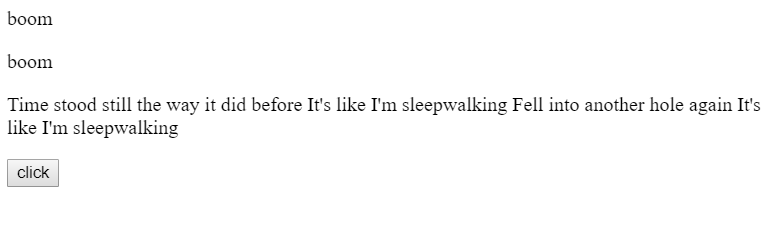


Рисунок 12

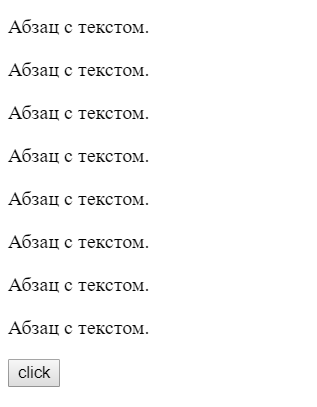


Рисунок 13

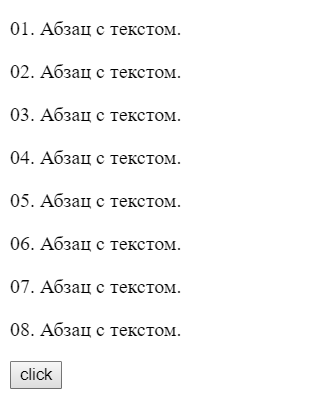


Рисунок 14

**Практическое задание 7: «Работа с таймером»**

**Задача 7.1**

Создать страницу с отсчетом от 0 до бесконечности и кнопкой, при нажатии на которую и начнется этот отсчет.

**Задача 7.2**

Создать страницу с отсчетом от 0 до бесконечности и двумя кнопками, при нажатии на одну из которых отсчет начнется, при нажатии на вторую - остановится.

**Задача 7.3**

Создать страницу с тикающими цифровыми часами.

**Задача 7.4**

Создать страницу с простым слайдером из 4 изображений (на экране изначально одно изображение, и оно поочередно сменяется другими).

**Задача 7.5**

Создать страницу с каруселью из 5 изображений и двумя кнопками: первая – запускает карусель, вторая – останавливает.

**Задача 7.6**

Создать страницу с тикающими цифровыми часами, показывающими отсчет оставшегося времени до полуночи.

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<p id="test"> 0 </p>

<button onclick="set()">we will push the button</button>

<br><br>

<script>

function set(){

window.setInterval(timer, 1000);

}

function timer(){

let test = document.getElementById('test');

test.innerHTML = parseInt(test.innerHTML)+1;

}

</script>

</body>

</html>

Результат работы кода:

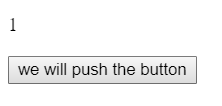


Рисунок 15

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

</head>

<body>

<p id="dhtvz"> 0 </p>

<input type="submit" value="Кронк жми на рычаг" onclick="go()" id="go">

<input type="submit" value="НЕ ТОТ РЫЧАГ" onclick="stop()" id="stop" disabled>

<script>

function go(){

window.timerId = window.setInterval(time, 500);

document.getElementById('go').disabled = true;

document.getElementById('stop').disabled = false;

}

function stop(){

window.clearInterval(window.timerId);

document.getElementById('go').disabled = false;

document.getElementById('stop').disabled = true;

}

function time(){

let test = document.getElementById('dhtvz');

test.innerHTML = parseInt(test.innerHTML)+1;

}

</script>

</body>

</html>

Результат работы кода:

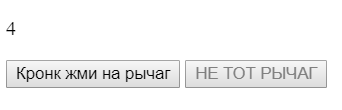


Рисунок 16

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title></title>

<style>

#clock {

font-size: 40px;

margin: 0;

}

</style>

<script>

function go() {

window.timerId = window.setInterval(timer, 500);

}

function timer() {

var clock = document.getElementById("clock");

var date = new Date();

clock.innerHTML =

addZero(date.getHours()) +

":" +

addZero(date.getMinutes()) +

":" +

addZero(date.getSeconds());

}

function addZero(num) {

if (num <= 9) return "0" + num;

else return num;

}

</script>

</head>

<body onload="go()">

<div id="wrapper">

<p id="clock">00:00:00</p>

</div>

</body>

</html>

Результат работы кода

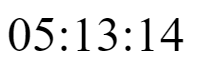


Рисунок 17

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<style>

#wrapper,{

text-align: center;

}

</style>

<script>

function go() {

window.timerId = window.setInterval(timer, 1000);

}

function timer() {

if (window.number == undefined || window.number == 4) {

window.number = 1;

} else window.number = window.number + 1;

var img = document.getElementById("img");

img.src = window.number + ".gif";

}

</script>

</head>

<body onload="go()">

<div id="wrapper">

<img src="1.gif" id="img" width="300" height="300">

</div>

</body>

</html>

Результат работы кода



Рисунок 18

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<style>

#wrapper

{

text-align: center;

}

</style>

<script>

function go() {

window.timerId = window.setInterval(timer, 1000);

}

function timer() {

if (window.number == undefined || window.number == 4) {

window.number = 1;

} else window.number = window.number + 1;

var img = document.getElementById("img");

img.src = window.number + ".gif";

}

function stop(){

window.clearInterval(window.timerId);

}

</script>

</head>

<body>

<div id="wrapper">

<img src="1.gif" id="img" width="300" height="300">

</div>

<button onclick="go()">thnk u next</button>

<button onclick="stop()">stooop</button>

</body>

</html>

Результат работы кода



Рисунок 19

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<style>

#wrapper {

font-size: 240%;

justify-content: center;

display: flex;

flex-wrap: wrap;

}

</style>

<script>

function go(){

window.timerId = window.setInterval(timer, 1000);

}

function timer(){

var seconds = document.getElementById('seconds');

var minutes = document.getElementById('minutes');

var hours = document.getElementById('hours');

var now = new Date();

var midnight = new Date(now.getFullYear(), now.getMonth(), now.getDate()+1, 0, 0, 0);

var diff = Math.floor((midnight - now)/1000);

var hoursRemain = Math.floor(diff/(60\*60));

var minutesRemain = Math.floor((diff-hoursRemain\*60\*60)/60);

var secondsRemain = Math.floor(diff%60);

hours.innerHTML = hoursRemain;

minutes.innerHTML = addZero(minutesRemain);

seconds.innerHTML = addZero(secondsRemain);

}

function addZero(num){

if(num <=9) return '0'+num;

else return num;

}

</script>

</head>

<body onload="go()">

<div id="wrapper">

<div id="hours"></div>

<div class="dv">:</div>

<div id="minutes"></div>

<div class="dv">:</div>

<div id="seconds"></div>

</div>

</body>

</html>

Результат работы кода

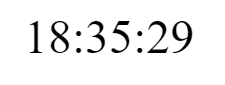


Рисунок 20

**Практическое задание 8: «Прокрутка элемента»**

**Задача 8.1**

Дан элемент #elem с вертикальной полосой прокрутки. По нажатию на кнопку вывести на сколько элемент прокручен сверху.

**Задача 8.2**

Дан элемент #elem с вертикальной полосой прокрутки. По нажатию на кнопку прокрутить его до позиции 200px сверху.

**Задача 8.3**

Дан элемент #elem с вертикальной полосой прокрутки. По нажатию на кнопку прокрутить его до позиции 200px от нижнего края элемента.

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<style>

#elem {

width: 150px;

height: 150px;

border: 2px solid black;

overflow: auto;

padding: 10px;

}

</style>

</head>

<body>

<div id="elem">

ПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриветПриве

тПриветПриветПриветПриветПривет

ПриветПриветПриветПриветПриветПривет

ПриветПриветПрив

етПриветПриветПривет

ПриветПриветПривет

ПриветПриветПриветПриветП

риветПриветПриветПриветПриветППриветПриветПриветПриветПриветПривет </div>

<br>

<button onclick="func()">и где</button>

<button onclick="scrolll()">на 200 пкс</button>

<button onclick="bott()">на 200 от низа</button>

<script>

function func() {

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

alert(elem.scrollTop);

}

function scrolll(){

let elem = document.querySelector('#elem');

elem.scrollTop = 200;

}

function bott(){

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

elem.scrollTop = elem.scrollHeight - 389; <!--паддинги сверху и снизу по 10-->

}

</script>

</body>

</html>

Результат работы кода

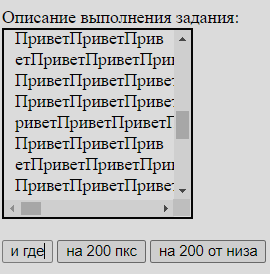


Рисунок 21



Рисунок 22

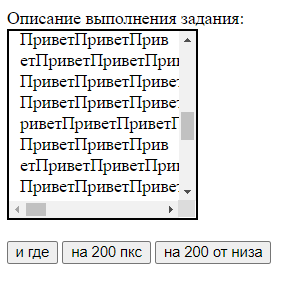


Рисунок 23

**Практическое задание 9: «Прокрутка страницы»**

**Задача 9.1** Дана страница с вертикальной полосой прокрутки. По нажатию на кнопку узнать на сколько страница прокручена по вертикали.

**Задача 9.2** Дана страница с горизонтальной полосой прокрутки. По нажатию на кнопку узнать на сколько страница прокручена по горизонтали.

**Задача 9.3** Дана страница с горизонтальной и вертикальной полосами прокрутки. По нажатию на кнопку прокрутить ее в точку 300px слева, 500px сверху.

**Задача 9.4** Дана страница с вертикальной полосой прокрутки. По нажатию на кнопку прокрутить на 300px вниз от текущего положения.

**Задача 9.5** По нажатию на кнопку прокрутить страницу до самого низа.

**Описание выполнения задания:**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<style>

#elem {

width: 200px;

height: 100px;

border: 2px solid black;

overflow-y: scroll;

padding: 10px;

overflow-x: scroll;

white-space: nowrap;

}

</style>

</head>

<body>

<div id="elem">

Попытки писать фантастическую прозу А. Н. Стругацкий предпринимал ещё до войны<br> (по свидетельству Бори<br>са Стругацкого, это была повесть «Находка м<br>айора Ковалёва», утраченная во время ленин<br>градской блок<br>ады). Первое сохранившееся п<br>роизведение Аркадия Стругацкого — рассказ «Как погиб Канг» — было закончено в 1946году и опубликовано в 2001 году. Он п<br>родолжал попытк<br>и писать фантастику и в<br> 1950-е годы во время военной

службы, прибл<br>изительно 1952<br> годом датирована повесть «Четвё<br>ртое царство». Борис Натанович начал писать с начала 1950-х годов; братья обмени<br>вались творческими замыслами попереписке и во время отпусков Аркад<br>ия Натановича. Первая художественная<br> публикация Аркадия Стругацкого — повесть «Пепел Бикини»</div>

<br>

<button onclick="func()">offset по Y</button>

<button onclick="scrolll()">offset по X</button>

<button onclick="bott()">скролл на 300,500</button>

<button onclick="scrby()">скролл на 0,300</button>

<button onclick="scrcl()">Вниз</button>

<script>

let elem = document.getElementById('elem');

function func(){

alert(window.pageYOffset);

}

function scrolll(){

alert(window.pageXOffset);

}

function bott(){

elem.scrollTo(300, 500);

}

function scrby(){

elem.scrollBy(0, 300);

}

function scrcl(){

elem.scrollTo(0, document.body.clientHeight);

}

</script>

</body>

</html>

Результат работы кода

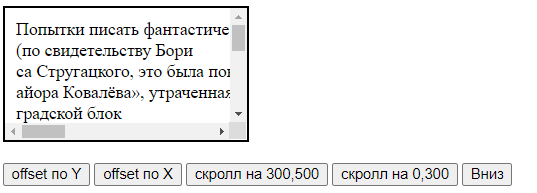


Рисунок 24

Рисунок 25

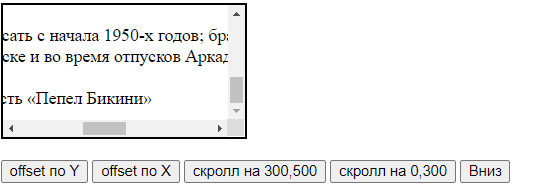


Рисунок 26

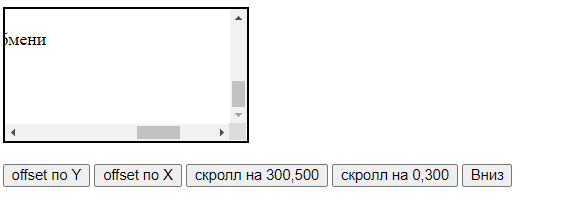


Рисунок 27



Рисунок 28

**Практическое задание 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**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<style>

body {

background: rgb(186, 161, 227);

margin: 10px auto;

width: 470px;

}

h3 {

margin: 0;

padding-bottom: 0.3em;

padding-right: 20px;

font-size: 1.1em;

}

p {

margin: 0;

padding: 0 0 0.5em;

}

.game {

background: linear-gradient(to right, rgb(129, 18, 235), rgb(154, 80, 168));

padding: 10px 20px 10px;

border: solid 2px rgb(197, 33, 221);

position: relative;

}

.remove-button {

position: absolute;

font-size: 110%;

top: 0;

color: rgb(246, 246, 246);

right: 10px;

display: block;

width: 24px;

height: 24px;

border: none;

background: transparent;

cursor: pointer;

}

</style>

</head>

<body>

<div>

<div class="game">

<h3>Apex Legends</h3>

<p>

Apex Legends — компьютерная игра в жанре многопользовательского шутера от первого лица и королевской битвы,

разработанная американской студией Respawn Entertainment и выпущенная компанией Electronic Arts для

платформ Windows, PlayStation 4 и Xbox One в 2019 году

</p>

</div>

<div class="game">

<h3>Witcher</h3>

<p>

«Ведьмак 3: Дикая Охота» — мультиплатформенная компьютерная игра в жанре action/RPG,

разработанная польской студией CD Projekt RED по мотивам серии романов «Ведьмак» польского

писателя Анджея Сапковского, выпущенная в 2015 году для Windows, PlayStation 4 и Xbox One.

</p>

</div>

<div class="game">

<h3>Cyberpunk 2077</h3>

<p>

Cyberpunk 2077 — мультиплатформенная компьютерная игра в жанре Action/RPG, разработанная польской студией CD Projekt RED.

Игра является продолжением настольной игры «Киберпанк 2020», действия происходят спустя 57 лет в большом городе Найт-Сити, расположенном в штате Калифорния.

</p>

</div>

</div>

<script>

let games = document.querySelectorAll(".game");

for (let game of games) {

game.insertAdjacentHTML(

"afterbegin",

'<button class="remove-button">[x]</button>'

);

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

}

</script>

</body>

</html>

Результат работы кода:

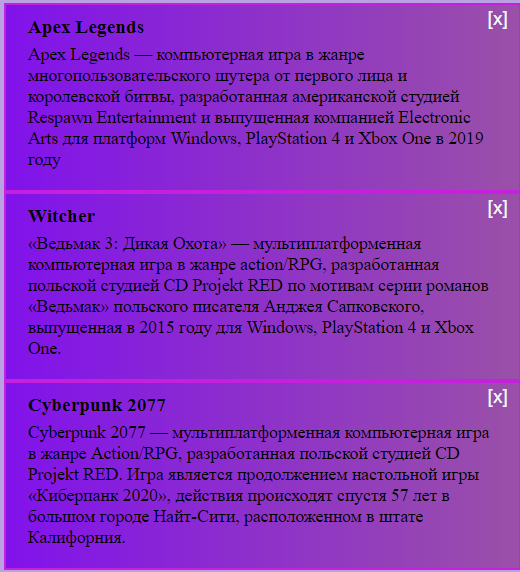


Рисунок 29

**Практическое задание 11: «Карусель»**

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

**Описание выполнения задания**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<style media="screen">

.formation\_of\_solar\_system\_slideshow{

width: auto;

display: grid;

justify-content: center;

grid-template-areas: "prev slide next";

position: relative;

z-index: 1;

}

.slide{

grid-area: slide;

}

.slide\_img{

border-radius: 4px;

}

.prev, .next {

cursor: pointer;

position: absolute;

top: 45%;

width: auto;

padding: 16px;

color: white;

font-weight: bold;

font-size: 18px;

transition: 0.6s ease;

border-radius: 0 3px 3px 0;

user-select: none;

}

.prev:hover, .next:hover {

background-color: rgba(0,0,0,0.8);

}

.prev{

left: 0;

grid-area: prev;

}

.next{

right: 0;

grid-area: next;

}

h3{

text-align: center;

}

.fade {

-webkit-animation-name: fade;

-webkit-animation-duration: 1.5s;

animation-name: fade;

animation-duration: 1.5s;

}

@-webkit-keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

</style>

</head>

<body>

<div class="formation\_of\_solar\_system\_slideshow">

<div class="slide fade">

<img src="https://64.media.tumblr.com/3cb013ca470d69b79502f233d546bdd6/tumblr\_p0tj8qwq7r1vjhboso1\_400.gif" class="slide\_img" height="370" width="620">

</div>

<div class="slide fade">

<img src="https://64.media.tumblr.com/b37c0e343320f19967696608e9e96c35/tumblr\_p0tj8qwq7r1vjhboso2\_400.gif" class="slide\_img" height="370" width="620">

</div>

<div class="slide fade">

<img src="https://64.media.tumblr.com/bfa8a701aec4aee6aa88b49fe19c11ff/tumblr\_p0tj8qwq7r1vjhboso3\_400.gif" class="slide\_img" height="370" width="620">

</div>

<a class="prev" onclick="plusSlides(-1)">&#10094;</a>

<a class="next" onclick="plusSlides(1)">&#10095;</a>

</div>

<h2>Скажу честно, это часть моего курсача, которую я просто вырезала для этого задания(прикольный фейд-эффект при перелистывании кстати...)</h2>

<script>

var slideIndex = 1;

showSlides(slideIndex);

// Next/previous controls

function plusSlides(n) {

showSlides(slideIndex += n);

}

function showSlides(n) {

var i;

var slides = document.getElementsByClassName('slide');

if (n > slides.length){

slideIndex = 1;

}

if (n < 1) {

slideIndex = slides.length;

}

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

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

}

slides[slideIndex-1].style.display = "block";

}

</script>

</body>

</html>

Результат работы кода

**Практическое задание 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**

<!DOCTYPE HTML>

<html>

<head>

<style>

.tree span:hover {

font-weight: bold;

}

.tree span {

cursor: pointer;

}

</style>

<meta charset="utf-8">

</head>

<body>

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

<li>Игры

<ul>

<li>RPG

<ul>

<li>VTMB</li>

<li>Disco Elysium</li>

<li>Planescape Torment</li>

</ul>

</li>

<li>Immersive Sim

<ul>

<li>Thief</li>

<li>Deus Ex</li>

<li>Prey</li>

</ul>

</li>

</ul>

</li>

<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>

</body>

</html>

Результат работы кода:

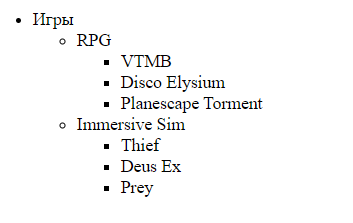


Рисунок 30

**Практическое задание 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**

<!DOCTYPE HTML>

<html>

<head>

<meta charset="utf-8">

<style>

body {

height: 2000px;

}

.tooltip {

position: fixed;

padding: 10px 20px;

border: 1px solid black;

border-radius: 4px;

text-align: center;

color: #333;

background: yellow;

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

}

</style>

</head>

<body>

<button data-tooltip="подсказка раз">элемент 1</button>

<button data-tooltip="подсказка номер<br>два">элемент 2</button>

<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>

</body></html>

Результат работы кода

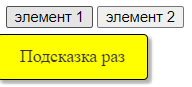


Рисунок 31

**Практическое задание 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**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<style>

#contents {

padding: 5px;

border: 1px black solid;

}

</style>

</head>

<body>

<fieldset id="contents">

<legend>Легенда</legend>

<p>Ссылки

<a href="https://vk.com/xeno\_xenia">ВК</a>

<a href="https://github.com/xenoxenia/mirea-html">Github</a>

</p>

</fieldset>

<script>

contents.onclick = function (event) {

function handleLink(href) {

let isLeaving = confirm(`Leave for ${href}?`);

if (!isLeaving) return false;

}

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

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

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

}

};

</script>

</body>

</html>

Результат работы кода:

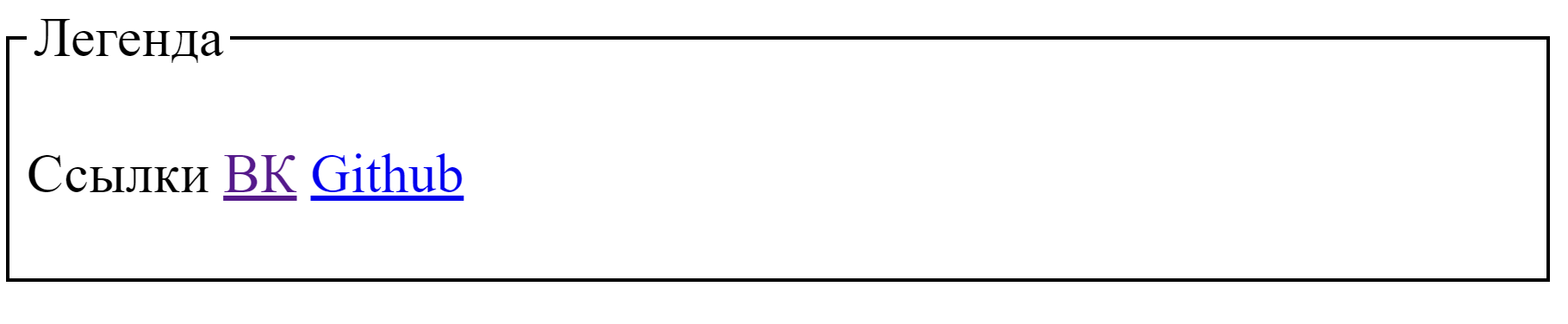
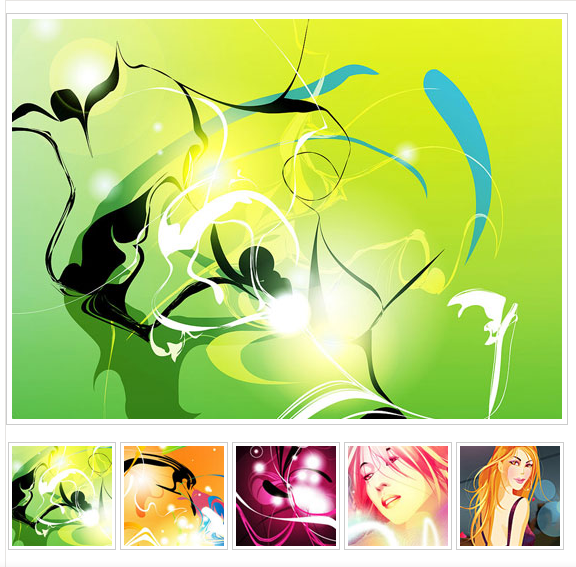


Рисунок 32

**Практическое задание 15: «Галерея изображений»**

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

*Пример изначального отображения:*



**Описание выполнения задания**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<style>

body {

margin: 0;

padding: 0;

font: 75%/120% sans-serif;

}

h2 {

font: bold 190%/100% sans-serif;

margin: 0 0 0.2em;

}

h2 em {

font: normal 80%/100% sans-serif;

color: black;

}

#largeImg {

border: solid 1px #ccc;

width: 550px;

height: 400px;

padding: 5px;

}

#thumbs a {

border: solid 1px #ccc;

width: 100px;

height: 100px;

padding: 3px;

margin: 2px;

float: left;

}

#thumbs a:hover {

border-color: yellow;

}

#thumbs li {

list-style: none;

}

#thumbs {

margin: 0;

padding: 0;

}

</style>

</head>

<body>

<p>

<img id="largeImg" src="https://64.media.tumblr.com/3f6a9d6aaa60ed733e3630edca0648dc/tumblr\_or1cwiPQv51uiyjnoo1\_1280.jpg">

</p>

<ul id="thumbs">

<li>

<a href="https://64.media.tumblr.com/f4f457459c0fe183f17bee59b90b3f6b/tumblr\_p7lofrMtrA1ui670jo1\_500.jpg" title="манчкин"

> <img src="https://64.media.tumblr.com/f4f457459c0fe183f17bee59b90b3f6b/tumblr\_p7lofrMtrA1ui670jo1\_500.jpg" height="100" width="100"

/></a>

</li>

<li>

<a href="https://64.media.tumblr.com/54453da012cf412cd31d7501d3305d98/5255bb07367ca21e-99/s500x750/c6913b530dacdeeef840db9928e78b4c878e5590.gif" title="howl's moving castle"

><img src="https://64.media.tumblr.com/54453da012cf412cd31d7501d3305d98/5255bb07367ca21e-99/s500x750/c6913b530dacdeeef840db9928e78b4c878e5590.gif" height="100" width="100"

/></a>

</li>

<li>

<a href="https://64.media.tumblr.com/16eece791bce948a9c21a4b80899bbee/56745b3eea5e3d07-80/s500x750/473a62a9871ecdac4cf1e1014e4b16dce92cc8ac.gif" title="howl's moving castle"

><img src="https://64.media.tumblr.com/16eece791bce948a9c21a4b80899bbee/56745b3eea5e3d07-80/s500x750/473a62a9871ecdac4cf1e1014e4b16dce92cc8ac.gif" height="100" width="100"

/></a>

</li>

<li>

<a href="https://64.media.tumblr.com/1c4662f210e5989c16e0d2802adc1d43/d3b9847f95a6f0d7-db/s500x750/d681014cd1cd54b08a8d9c966c085450a6d6f6a5.gif" title="howl's moving castle"

><img src="https://64.media.tumblr.com/1c4662f210e5989c16e0d2802adc1d43/d3b9847f95a6f0d7-db/s500x750/d681014cd1cd54b08a8d9c966c085450a6d6f6a5.gif" height="100" width="100"

/></a>

</li>

<li>

<a href="https://64.media.tumblr.com/5a67c6b21911106bd1571dccdf047ca6/6016c4c51ddd6896-6c/s500x750/375bb7e36c60c6342d3f438f9cf842beb497bf87.gif" title="howl's moving castle"

><img src="https://64.media.tumblr.com/5a67c6b21911106bd1571dccdf047ca6/6016c4c51ddd6896-6c/s500x750/375bb7e36c60c6342d3f438f9cf842beb497bf87.gif" height="100" width="100"

/></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>

</body>

</html>

Результат работы кода:



Рисунок 33

**Вывод**

В результате выполнения данной практической работы были получены навыки работы с DOM, обработчиками событий и this, приобретены навыки в работе с делегированием.