

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

старший преподаватель		Н.А. Соловьева
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

**Клиентские языки сценариев. Javascript**

по дисциплине: Web-технологии

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №	Z1431		М.Д.Быстров
	номер группы	подпись, дата	инициалы, фамилия

Студенческий билет №	2021/3572
----------------------	-----------

Санкт-Петербург 2024

## Оглавление

Оглавление .....	2
Цель работы .....	3
Задание .....	4
Средства, использованные при выполнении работы .....	6
Таблица созданных функций .....	7
Тексты функций .....	8
Скриншоты.....	11
Исходный код программы .....	14
Вывод.....	41

## **Цель работы**

**Цель работы:** получение опыта написания и применения функций на языке javascript.

## Задание

Таблица 1 Вариант индивидуального задания

№	Гр z1431	Тема сайта	Вариант таблицы и списка
1	Быстров		1

Таблица 2 Тема в соответствии с вариантом

№ варианта	Тема
1	Язык Котлин

Таблица 3 Вариант выполнения работы

№	Событие	Способ включени я сценария	Поиск элемента	Оформле ние функции	Сценари й
1	click	тег <script>	getElementById	с именем	1

Сценарий 1: Часы, которые показывают время в трех разных часовых поясах.

### Базовое задание

Подготовить несколько сценариев по заданиям ниже. Все сценарии встроить в сайт из лабораторной работы № 2. Запрещается создавать отдельную страницу для выполнения данной работы. Варианты в таблице 1 определяют способы реализации сценариев (тип события, способ включения сценария, метод выбора элемента на странице, оформление функции). Вид метода (getElementById, getElementsByName, getElementsByTagName), заданный вариантом, ОБЯЗАТЕЛЬНО должен применяться для большей части всех случаев обращения к элементам на странице.

- 1) Вызов всплывающего окна с ФИО и группой автора сайта
- 2) Добавить сценарий по варианту из Таблицы 1 (столбец «сценарий»)
- 3) Добавить пустую строку в таблицу, подготовленную в работе 1
- 4) Изменить цвет элемента
- 5) Изменить размер элемента

### Расширенное задание

- 1.. При добавлении строки в таблицу для заполнения ячеек брать данные, введенные пользователем в элементы интерфейса (input)
- 2.. Удалять только добавленные через сценарий строки таблицы
- 3.. Применить лямбда (стрелочную) функцию

- 4.. Использовать события окна браузера (загрузка, перемещение, скроллинг)
- 5.. Сделать функцию с несколькими параметрами и показать ее применение
- 6.. Использовать `addEventListener`
- 7.. Добавить и удалить элемент списка

## **Средства, использованные при выполнении работы**

1. Редактор Visual Studio Code
2. Браузер Mozilla Firefox

### Таблица созданных функций

Имя функции	Обработчик	Описание
addRowToTable()	<code>onclick="addRowToTable();"</code>	Добавление строки в таблицу
removeRowsFromTable()	<code>onclick="removeRowsFromTable();"</code>	Удаление добавленных строк из таблицы
removeRow(table, index)	-	Удаление указанной строки из таблицы
init()	<code>onload="init();"</code>	Инициализация при загрузке страницы
alertAuthor()	<code>onclick="alertAuthor();"</code>	Вывод информации об авторе
renderClocks()	-	Отрисовка часов
changeColor()	<code>onclick="changeColor();"</code>	Изменение цвета элемента
changeSize()	<code>onclick="changeSize();"</code>	Изменение размера элемента
getRandomColor()	-	Получение случайного цвета

## Тексты функций

```
var addedRows = [];  
  
// 3. добавление строки в таблицу  
function addRowToTable() {  
    var table = document.getElementById("table");  
  
    var row = table.insertRow();  
  
    var cell1 = row.insertCell(0);  
    var cell2 = row.insertCell(1);  
    var cell3 = row.insertCell(2);  
    var cell4 = row.insertCell(3);  
    var cell5 = row.insertCell(4);  
  
    var input1 = document.getElementById("propInput");  
    var input2 = document.getElementById("input17");  
    var input3 = document.getElementById("input18");  
    var input4 = document.getElementById("input19");  
    var input5 = document.getElementById("input20");  
  
    cell1.outerHTML = "<th>" + input1.value + "</th>";  
    cell2.innerHTML = input2.value;  
    cell3.innerHTML = input3.value;  
    cell4.innerHTML = input4.value;  
    cell5.innerHTML = input5.value;  
  
    input1.value = "";  
    input2.value = "";  
    input3.value = "";  
    input4.value = "";  
    input5.value = "";  
  
    addedRows.push(row.rowIndex);  
}  
  
// доп. задание - удаление добавленных строк  
function removeRowsFromTable() {  
    var table = document.getElementById("table");  
  
    // доп. задание: использование стрелочной функции  
    addedRows.reverse().forEach((row) => removeRow(table, row));  
    addedRows = [];  
}  
  
// доп. задание - использование функции нескольких аргументов  
function removeRow(table, index) {  
    table.deleteRow(index);  
}
```



```

function init() {
    /// настройка часов
    renderClocks();
    setInterval(renderClocks, 1000);
}

// 1. Вывод информации об авторе
function alertAuthor() {
    alert("Быстров Максим Z1431");
}

// 2. Отрисовка часов
function renderClocks() {
    var clock1 = document.getElementById("clock1");
    var clock2 = document.getElementById("clock2");
    var clock3 = document.getElementById("clock3");

    var date = new Date();

    var timeZone1 = 'America/New_York';
    var timeZone2 = 'Etc/UCT';
    var timeZone3 = 'Europe/Moscow';

    clock1.innerHTML = date.toLocaleString('ru-RU', { timeZone: timeZone1
}) + " " + timeZone1;
    clock2.innerHTML = date.toLocaleString('ru-RU', { timeZone: timeZone2
}) + " " + timeZone2;
    clock3.innerHTML = date.toLocaleString('ru-RU', { timeZone: timeZone3
}) + " " + timeZone3;
}

// 4-5. изменение цвета и размера элемента
function changeColor() {
    document.getElementById("modifyableDiv").style.backgroundColor =
getRandomColor();
}

function getRandomColor() {
    var letters = '0123456789ABCDEF';
    var color = '#';
    for (var i = 0; i < 6; i++) {
        color += letters[Math.floor(Math.random() * 16)];
    }
    return color;
}

function changeSize() {
    document.getElementById("modifyableDiv").style.width = 20 +
Math.random() * 80 + "%";
}

```

}

## Скриншоты

Описание поддержки версий языка Kotlin				
Версия	1.7	1.8	1.9	2.0
Обновления безопасности	Нет			Да
Дата выпуска	09 Jun 2022	28 Dec 2022	06 Jul 2023	21 May 2024

Свойство:

1.7:

1.8:

1.9:

2.0:

Рисунок 1 Ввод данных для добавления в таблицу

Описание поддержки версий языка Kotlin				
Версия	1.7	1.8	1.9	2.0
Обновления безопасности	Нет			Да
Дата выпуска	09 Jun 2022	28 Dec 2022	06 Jul 2023	21 May 2024
Номер	1	2	3	4

Свойство:

1.7:

1.8:

1.9:

2.0:

Рисунок 2 Добавленные данные в таблицу

Описание поддержки версий языка Kotlin				
Версия	1.7	1.8	1.9	2.0
Обновления безопасности	Нет 			Да
Дата выпуска	09 Jun 2022	28 Dec 2022	06 Jul 2023	21 May 2024

Свойство:

1.7:

1.8:

1.9:

2.0:

Рисунок 3 Удаление данных из таблицы

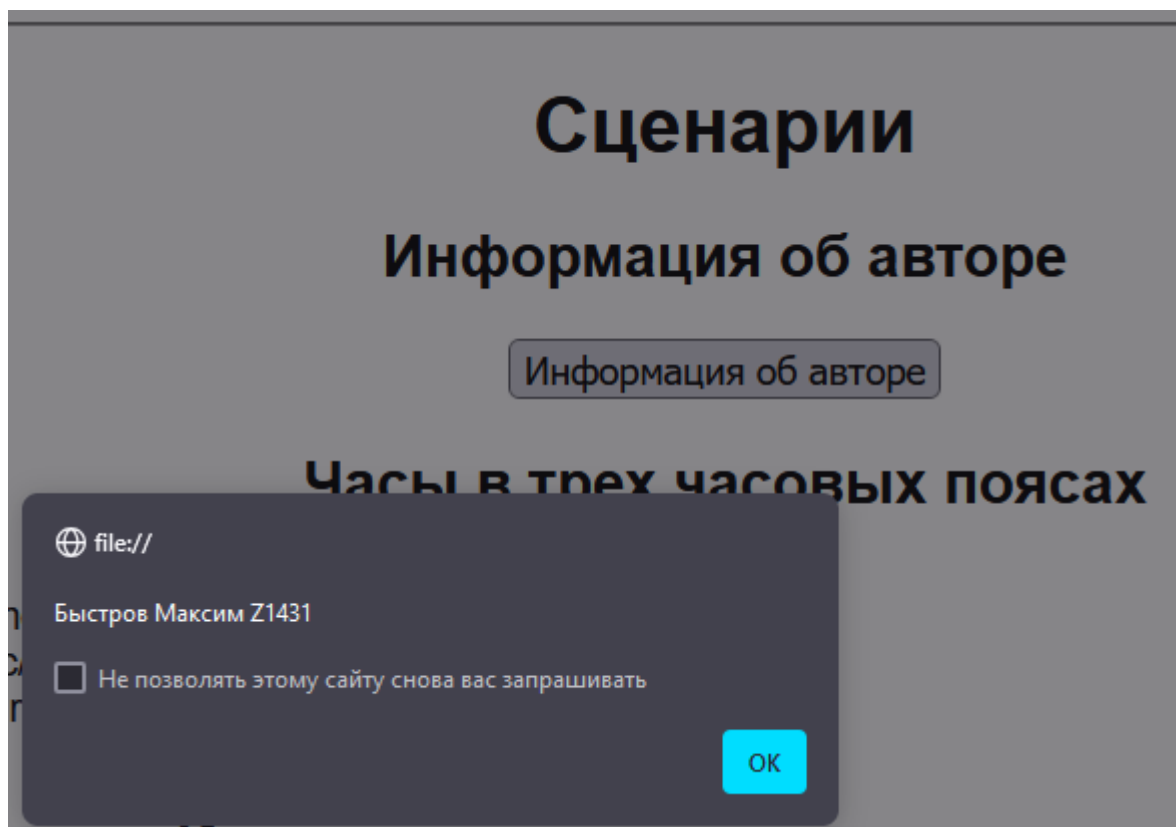


Рисунок 4 Вывод информации об авторе

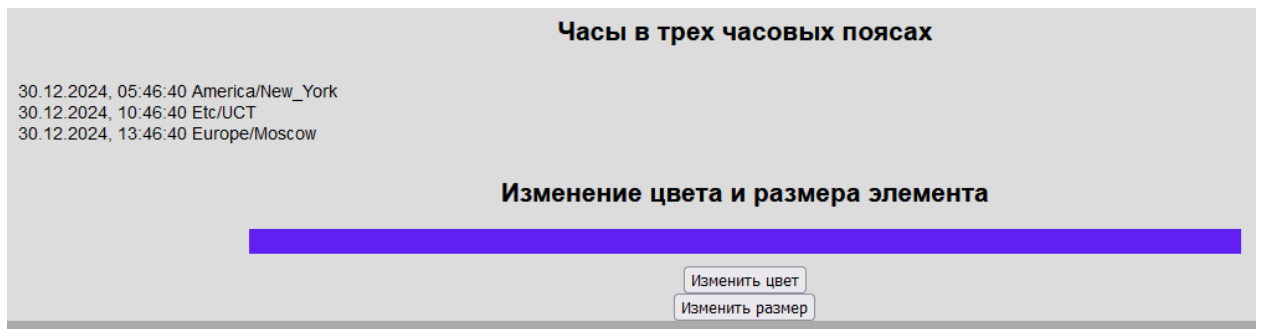


Рисунок 5 Часы в трех часовых поясах, элемент для изменения

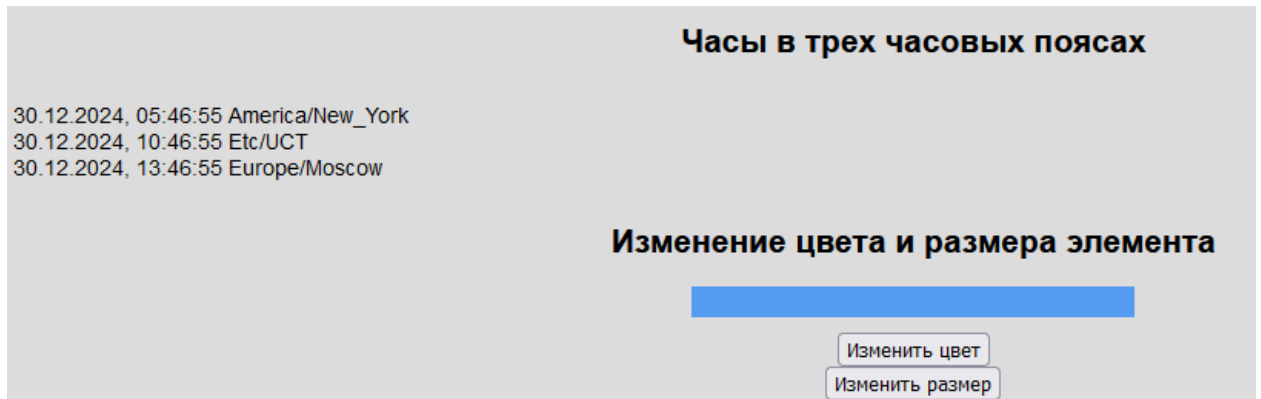


Рисунок 6 Результат изменения цвета и размера элемента

# Исходный код программы

## 1. Файл «1.html»

```
<html>

<head>
  <title>Язык Kotlin</title>
  <link rel="shortcut icon" type="image/x-icon" href="img/kotlin_logo.png" />
  <link rel="stylesheet" type="text/css" href="1.css">
  <link rel="stylesheet" type="text/css" href="list.css">
  <meta charset="UTF-8">
  <!-- meta - дополнительное задание -->
  <meta name="description" content="Kotlin programming language">
  <meta name="keywords" content="Kotlin, Java, JVM, JDK">
  <meta name="author" content="Maxim Bystrov">

  <style>
    ol [id="or"]{
      list-style-type: lower-roman;
    }
  </style>
</head>

<body>
  <div id="container" class="container">

    <!-- шапка - обязательный элемент сайта -->
    <div id="header">

      <div id="logo"></div>

      <h2 id="sitename">Язык Kotlin</h2>

      <h3 id="pagename">История</h3>

    </div>

    <!-- навигация между страницами - обязательный элемент сайта -->
    <div id="navigation" class="container">
      <h2><a href="1.html">История</a>
        <a href="2.html">Документация</a>
        <a href="3.html">Источники</a>
      </h2>
    </div>

    <br>
```

```

<!-- боковая панель -->
<div id="sidebar">
    <h2>Новости</h2>
    <h3><a href="https://kotlinlang.org/docs/releases.html">Release
Notes</a></h3>
    <h3><a href="https://blog.jetbrains.com/kotlin/2024/11/kotlin-
roundup-kodee-s-top-picks/">Kotlin Roundup</a>
    </h3>
    <h3><a href="https://blog.jetbrains.com/kotlin/2024/10/ktor-3-
0/">Вышел Ktor 3.0</a></h3>
</div>

<!-- основной контент -->
<div id="content">
    <h2>История языка Kotlin</h2>

    <h3>Введение</h3>
    <p>
        Kotlin (Кóтлин) – кроссплатформенный, статически типизированный,
        объектно-ориентированный язык программирования, работающий поверх
        Java Virtual Machine и разрабатываемый
        компанией JetBrains.
        Также компилируется в JavaScript и в исполняемый код ряда
        платформ через инфраструктуру LLVM.
    </p>
    <p>
        Авторы ставили целью создать язык более лаконичный и
        типобезопасный, чем Java, и более простой, чем
        Scala. Следствием упрощения по сравнению со Scala стали также
        более быстрая компиляция и лучшая
        поддержка языка в IDE. Язык полностью совместим с Java, что
        позволяет Java-разработчикам постепенно
        перейти к его использованию; в частности, язык также встраивается
        в Android, что позволяет для
        существующего Android-приложения внедрять новые функции на Kotlin
        без переписывания приложения целиком.
    </p>

    <h3>История</h3>
    <p>
        Язык назван в честь русского острова Котлин в Финском заливе,
        на котором расположен город Кронштадт. Андрей Бреслав, бывший
        ведущий дизайнер Kotlin, упомянул, что команда решила назвать его
        в честь острова, так же как язык программирования Java был назван
        в честь индонезийского острова Ява (есть мнение, что название
        языка было навеяно «java» – американским
        сленговым термином для кофе,
        который сам по себе происходит от названия острова).
    </p>
    <p>

```

```
<!-- дополнительное задание - cite -->
<cite>
    На этом фоне новый проект компании JetBrains под кодовым
    названием Kotlin (ударение на «о»), с
    одной стороны, выглядит почти данью моде, а с другой –
    находится в окружении заметного числа
    конкурентов. Однако мы чувствуем себя достаточно уверенно в
    этой ситуации, и тому есть несколько
    причин. Во-первых, JetBrains уже более десяти лет занимается
    интегрированными средами разработки для
    разных языков программирования (многие из которых работают на
    платформе Java), и за это время была
    собрана сильная команда специалистов и накоплен значительный
    опыт в области языков программирования
    и смежных технологий. Во-вторых, мы не можем сказать, что
    какой-либо из существующих языков на
    платформе Java полностью удовлетворяет нашим потребностям, и
    полагаем, основываясь на
    предварительных отзывах программистов всего мира, что наши
    коллеги в других компаниях испытывают
    похожие затруднения.
</cite>
```

```
</p>
```

```
<div id="citelegend">
    <a href="https://www.osp.ru/os/2011/09/13011550">
        Андрей Бреслав. Язык программирования Kotlin // Открытые
        системы. – 2011. – № 09
    </a>
</div>
```

```
<p>
    Kotlin 1.0 был выпущен 15 февраля 2016 года. Он считается первым
    официально стабильным релизом и
    начиная с этой версии,
    JetBrains взяла на себя обязательство по долгосрочной обратной
    совместимости.
</p>
```

```
<p>
    В мае 2017 на Google I/O 2017 года компания Google объявила, что
    инструменты языка Kotlin, основанные на
    JetBrains IDE, будут включены в Android Studio 3.0 – официальный
    инструмент разработки для ОС
    Android.
</p>
```

```
<p>
```



Kotlin 1.2 был выпущен 28 ноября 2017 года. В релиз добавлена функция совместного использования кода между платформами JVM и JavaScript (мультиплатформенное программирование).

</p>

<p>

Kotlin 1.3 был выпущен 29 октября 2018 года, добавив поддержку сопрограмм для использования с асинхронным программированием.

</p>

<p>

На Google I/O 2019 было объявлено, что язык программирования Kotlin стал приоритетным в разработке под Android.

</p>

<p>

Kotlin 1.4 был выпущен в августе 2020 года, в том числе с некоторыми небольшими изменениями в поддержке платформ Apple (во взаимодействии Objective-C / Swift).

</p>

<p>

В ноябре 2020 года Андрей Бреслав объявил об уходе из JetBrains, руководство разработкой языка было передано Роману Елизарову.

</p>

<p>

Kotlin 1.5 был выпущен в мае 2021 года.

</p>

<p>

Kotlin 1.6 был выпущен в ноябре 2021 года.

</p>

<p>

Kotlin 1.7 был выпущен в июне 2022 года, включая альфа-версию нового компилятора Kotlin K2.

</p>

<p>

Kotlin 1.8 был выпущен в декабре 2022 года.

</p>

<p>

Kotlin 1.9 был выпущен в июле 2023 года.

</p>

<p>

Kotlin 2.0 был выпущен в мае 2024 года.

</p>

### <h3>Синтаксис</h3>

<p>

Синтаксис языка преимущественно комбинирует наследство из двух языковых ветвей: Си/C++/Java и ML (по словам создателей, через Scala).

</p>

<p>

Из наиболее характерных элементов от первой ветви унаследованы блоки кода, обрамлённые фигурными скобками; а от второй – постфиксное указание типов переменных и параметров (сперва идентификатор, затем разделитель – двоеточие, и затем тип) и ключевые слова «fun» и «val». Точка с запятой как разделитель операторов необязательна (как в Scala, Groovy и JavaScript); в большинстве случаев перевода строки достаточно, чтобы компилятор понял, что выражение закончилось.

</p>

<p>

Кроме объектно-ориентированного подхода, Kotlin также поддерживает процедурный стиль с использованием функций. Как и в Си, С++ и D, точка входа в программу – функция main, принимающая массив параметров командной строки. Программы на Kotlin также поддерживают perl- и shell-стиль интерполяции строк (переменные, включённые в строку, заменяются на своё содержимое). Также поддерживается вывод типов.

</p>

<h3>Дальнейшее развитие</h3>

<p>

<!-- список в соответствии с заданием - основное задание -->

<ol id="list">

<li>Компилятор</li>

<!-- дополнительное задание - применение различных уровней css -->

>

<ol id="or" class="decimal" style="list-style-type: decimal">

<li>Улучшить качество рекомендаций компилятора</li>

<li>Генерация методов JVM по умолчанию</li>

</ol>

<li>Мультиплатформенность

<ol class="roman">

<li>Публичный релиз Swift Export</li>

<li>Включенная по умолчанию многопоточная работа GC (Mark

+ Sweep)</li>

</ol>

</li>

<li>Инструменты

<ol class="decimal">

<li>Полноценный релиз IntelliJ IDEA K2</li>

<li>Разработка Design Tools API</li>

</ol>

</li>

</ol>

</p>

```

    <h2>Лекция от создателя языка</h2>

    <!-- video - дополнительное задание -->
    <video width="100%" controls>
        <source src="img/breslav.mp4" type="video/mp4">
        Your browser does not support the video tag.
    </video>

</div>

<div id="clear"> </div>

<!-- подвал - обязательный элемент сайта -->
<div id="footer">
    <h2>Контакты</h2>
    <p>
        <a href="https://github.com/maxi7665">GitHub</a><br>
        <a href="https://github.com/maxi7665/web-technologies">Исходный
код сайта</a>
    </p>
</div>
</div>
</body>

</html>

```

## 2. Файл «l.css»

```

body {
    background: #f1f1f1;
    color: #000;
    font-family: Arial, sans-serif;
    font-size: 14px;
    padding: 10px;
}

#header {
    background: #DCDCDC;
    width: 100%;
    height: 100px;
    display: flex;
    align-items: center;
}

#header h2 {
    text-align: left;
}

#sitename {

```

```

        margin-left: auto;
    }

    #pagename {
        margin-left: auto;
        margin-right: auto;
        font-size: 30;
    }

    #logo {
        height: 100%;
        width: 16%;
        text-align: left;
    }

    #logo img{
        height: 100%;
        width: auto;
    }

    #navigation {
        background: rgb(51,51,51);
        width: 100%;
        height: 50px;
    }

    #sidebar {
        background: #DCDCDC;
        float: left;
        width: 20%;
    }

    #content {
        background: #DCDCDC;
        float: right;
        width: 79%;
    }

    #content p {
        text-align: left;
        padding-left: 10px;
        padding-right: 10px;
    }

    #clear {
        clear: both;
    }

    #footer {
        /* дополнительное задание: применение градиента */

```

```

        background-image: linear-gradient(#333, #000000);
        width: 100%;
    }

    #footer h2 {
        color: #f2f2f2;
    }

    #footer a {
        color: #f2f2f2;
        text-decoration: none;
    }

    #footer p {
        padding: 3px;
    }

    #container {
        background: #AAA;
        margin: auto auto;
        text-align: center;
        width: 80%;
        padding: 10px;
    }

    #navigation a {
        float: left;
        display: block;
        color: #f2f2f2;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }

    /* Change color on hover */
    #navigation a:hover {
        background-color: #ddd;
        color: black;
    }

    #list{
        text-align: left;
    }

    #sitelegend{
        text-align: right;
        padding-right: 10px;
    }

    /* дополнительное задание - комбинатор "+" */

```

```
h3 + p {  
    margin-top: 20px;  
}
```

### 3. Файл «list.css»

```
/* дополнительное задание - связанные таблицы стилей */  
ol.roman {  
    list-style-type: upper-roman;  
}  
  
ol.decimal {  
    list-style-type: decimal-leading-zero;  
}
```

### 4. Файл «2.html»

```
<html>  
  
<!-- шапка - обязательный элемент сайта -->  
  
<head>  
    <title>Язык Kotlin</title>  
    <link rel="shortcut icon" type="image/x-icon" href="img/kotlin_logo.png" />  
    <link rel="stylesheet" type="text/css" href="2.css">  
  
    <meta charset="UTF-8">  
    <!-- meta - дополнительное задание -->  
    <meta name="description" content="Kotlin programming language">  
    <meta name="keywords" content="Kotlin, Java, JVM, JDK">  
    <meta name="author" content="Maxim Bystrov">  
  
    <!-- основное задание - глобальные таблицы стилей -->  
    <style>  
        .table {  
            border: 2px solid black;  
            overflow-x: scroll;  
            width: 300px;  
            display: block;  
            border-radius: 3%;  
        }  
  
        tr,  
        th,  
        td {  
            border: 1px solid grey;  
        }  
    </style>
```

```

<script type="text/javascript">

    var addedRows = [];

    // 3. добавление строки в таблицу
    function addRowToTable() {
        var table = document.getElementById("table");

        var row = table.insertRow();

        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        var cell3 = row.insertCell(2);
        var cell4 = row.insertCell(3);
        var cell5 = row.insertCell(4);

        var input1 = document.getElementById("propInput");
        var input2 = document.getElementById("input17");
        var input3 = document.getElementById("input18");
        var input4 = document.getElementById("input19");
        var input5 = document.getElementById("input20");

        cell1.innerHTML = "<th>" + input1.value + "</th>";
        cell2.innerHTML = input2.value;
        cell3.innerHTML = input3.value;
        cell4.innerHTML = input4.value;
        cell5.innerHTML = input5.value;

        input1.value = "";
        input2.value = "";
        input3.value = "";
        input4.value = "";
        input5.value = "";

        addedRows.push(row.rowIndex);
    }

    // доп. задание - удаление добавленных строк
    function removeRowsFromTable() {
        var table = document.getElementById("table");

        // доп. задание: использование стрелочной функции
        addedRows.reverse().forEach((row) => removeRow(table, row));
        addedRows = [];
    }

    // доп. задание - использование функции нескольких аргументов
    function removeRow(table, index) {
        table.deleteRow(index);
    }

```

```

    }

</script>
</head>

<body>
    <div id="container" class="container">
        <header id="header">
            <div id="logo"></div>

            <h2 id="sitename">Язык Kotlin</h2>

            <h3 id="pagename">Документация</h3>

        </header>

        <!-- навигация между страницами - обязательный элемент сайта -->
        <nav id="navigation" class="container">
            <h2><a href="1.html">История</a>
                <a href="2.html">Документация</a>
                <a href="3.html">Источники</a>
            </h2>
        </nav>

        </p>

        <!-- боковая панель -->
        <aside id="sidebar">
            <h2>Новости</h2>
            <h3><a href="https://kotlinlang.org/docs/releases.html">Release
Notes</a></h3>
            <h3><a href="https://blog.jetbrains.com/kotlin/2024/11/kotlin-
roundup-kodee-s-top-picks/">Kotlin Roundup</a>
            </h3>
            <h3><a href="https://blog.jetbrains.com/kotlin/2024/10/ktor-3-
0/">Вышел Ktor 3.0</a></h3>
        </aside>

        <!-- основной контент -->
        <section id="content">

            <article>
                <h2>Статус поддержки последних версий</h2>

                <!-- таблица - основное и дополнительное задание -->
                <table class="table" id="table" cellpadding="4" cellspacing="0">
                    <colgroup>
                        <col>

```



```

        <col span="3" style="background-color:yellow">
        <col style="background-color:green">
    </colgroup>
    <caption>Описание поддержки версий языка Kotlin</caption>
    <thead>
        <tr>
            <th>Версия</th>
            <th>1.7</th>
            <th>1.8</th>
            <th>1.9</th>
            <th>2.0</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <th>Обновления безопасности</th>
            <!-- основное задание - локальная таблица стилей -->
            <td colspan="3">Нет </td>
            <td>Да</td>
        </tr>
        <tr align="center">
            <th>Дата выпуска</th>
            <td>09 Jun 2022</td>
            <td>28 Dec 2022</td>
            <td>06 Jul 2023</td>
            <td>21 May 2024</td>
        </tr>
    </tbody>
</table>

<!-- 3. добавление строки в таблицу -->
<div id="inputDiv">

    Свойство:<input id="propInput"><br>
    1.7:<input id="input17"><br>
    1.8:<input id="input18"><br>
    1.9:<input id="input19"><br>
    2.0:<input id="input20"><br>

    <button onclick="addRowToTable();">Добавить строку</button>
    <button onclick="removeRowsFromTable();">Удалить добавленные
строки</button>
</div>

</article>
<hr>

<article>

```

## <h2>Kotlin Multiplatform</h2>

<p>Поддержка мультиплатформенного программирования является одним из ключевых преимуществ Kotlin. Она сокращает время, затрачиваемое на написание и поддержку одного и того же кода для разных платформ, сохраняя при этом гибкость и преимущества нативного программирования.</p>

```
<!-- набор картинок - основное задание -->
<picture id="pic1">
  <source srcset="img/kotlin_multiplatform_scheme.png"
type="image/png" />
  
</picture>
```

### <h3>Примеры использования Kotlin Multiplatform</h3>

```
<!-- strong - дополнительное задание -->
<p><strong>Android и iOS приложения</strong></p>

<picture id="pic2">
  
</picture>

<!-- mark - дополнительное задание -->
<p>Использование <mark>одного кода</mark> на всех мобильных
платформах является одним из основных
вариантов
использования Kotlin Multiplatform. С помощью Kotlin
Multiplatform Mobile вы можете создавать
мультиплатформенные мобильные приложения, совместно
использующие один код и в Android, и в iOS,
например
бизнес-логику, возможности подключения и многое другое.</p>
```

```
<p><strong>Фулстек веб-приложения</strong></p>

<picture id="pic3">
  
</picture>
```

<p>Другой сценарий, когда совместное использование кода может принести пользу, — это приложение, в котором логика может быть повторно использована как на стороне сервера, так и на стороне клиента,

запущенного в  
браузере. С этим Kotlin Multiplatform тоже прекрасно  
справляется.</p>

<p><strong>Мультиплатформенные библиотеки</strong></p>

<picture id="pic3">  
   
</picture>

<!-- abbr - дополнительное задание -->

<p>Kotlin Multiplatform также отлично подойдет для авторов  
библиотек. Вы можете создать  
мультиплатформенную  
библиотеку с общим кодом и ее платформенными реализациями для  
<abbr>JVM</abbr>, <abbr>JS</abbr> и  
собственных платформ. После  
публикации мультиплатформенная библиотека может  
использоваться в других кроссплатформенных проектах  
в  
качестве зависимости.</p>

<p><strong>Общий код для мобильных и веб-приложений</strong></p>

<picture id="pic3">  
   
</picture>

<p>Еще одним популярным примером использования Kotlin  
Multiplatform является совместное использование  
одного и того же кода в Android, iOS и веб-приложениях. Это  
сокращает объем бизнес-логики,  
написанной  
фронтенд-разработчиками, и помогает более эффективно внедрять  
продукты, сокращая затраты на  
разработку и  
тестирование.

</p>

</article>

<hr>

<h2>Определение имени пакета и импорт</h2>

<p>Имя пакета указывается в начале исходного файла, так же как и в  
Java.</p>

<pre>

<!-- здесь и далее code - дополнительное задание -->

```
<code class="language-kotlin hljs" data-highlighted="yes">
```

```
package my.demo
```

```
import java.util.*
```

```
// ...</code>
```

```
</pre>
```

<p>Но в отличие от Java, нет необходимости, чтобы структура пакетов совпадала со структурой папок:

исходные файлы могут располагаться в произвольном месте на диске.</p>

```
<hr>
```

```
<h2>Точка входа в программу</h2>
```

<p>В Kotlin точкой входа в программу является функция main.</p>

```
<pre>
```

```
<code class="language-kotlin hljs" data-highlighted="yes">
```

```
fun main() {  
    println("Hello world!")  
}</code>
```

```
</pre>
```

<p>Другая форма main может принимать массив строк String.</p>

```
<pre>
```

```
<code class="language-kotlin hljs" data-highlighted="yes">
```

```
fun main(args: Array<String>) {  
    println(args.contentToString())  
}</code>
```

```
</pre>
```

```
<hr>
```

```
<h2>Вывод в стандартный поток</h2>
```

<p>print выводит свой аргумент в стандартный поток вывода.</p>

```
<pre>
```

```
<code class="language-kotlin hljs" data-highlighted="yes">
```

```
print("Hello")  
print("world!")</code>
```

```
</pre>
```

<p>println выводит свой аргумент и добавляет перевод строки, так что следующее, что вы выведете, появится на

следующей строке.</p>

```
<pre>
```

```
<code class="language-kotlin hljs" data-highlighted="yes">
```

```
println("Hello world!")  
println(42)</code>
```

```
</pre>
```

<hr>

## <h2>Функции</h2>

<p>Функция принимает два аргумента Int и возвращает Int.</p>

<pre>

```
<code class="language-kotlin hljs" data-highlighted="yes">
fun sum(a: Int, b: Int): Int {
    return a + b
}</code>
```

</pre>

<p>В качестве тела функции может выступать выражение. Тогда тип возвращаемого значения определяется автоматически.</p>

<pre>

```
<code class="language-kotlin hljs" data-highlighted="yes">
fun sum(a: Int, b: Int) = a + b</code>
```

</pre>

<p>Функция, не возвращающая никакого значения (void в Java).</p>

<pre>

```
<code class="language-kotlin hljs" data-highlighted="yes">
fun printSum(a: Int, b: Int): Unit {
    println("сумма $a и $b равна ${a + b}")
}</code>
```

</pre>

<p>Тип возвращаемого значения Unit может быть опущен.</p>

<pre>

```
<code class="language-kotlin hljs" data-highlighted="yes">
fun printSum(a: Int, b: Int) {
    println("сумма $a и $b равна ${a + b}")
}</code>
```

</pre>

<hr>

## <h2>Переменные</h2>

<p>Неизменяемые (только для чтения) локальные переменные определяются с помощью ключевого слова val.

Присвоить им значение можно только один раз.</p>

<pre>

```
<code class="language-kotlin hljs" data-highlighted="yes">
val a: Int = 1    // Инициализация при объявлении
val b = 1         // Тип `Int` определен автоматически
val c: Int        // Указывать тип обязательно,
// если переменная не инициализирована сразу
c = 1             // Последующее присвоение</code>
```

</pre>

```

        <p>Изменяемые переменные объявляются с помощью ключевого слова
var.</p>
        <pre>
            <code class="language-kotlin hljs" data-
highlighted="yes">
                var x = 5 // Тип `Int` определен автоматически
                x += 1</code>
            </pre>
        <p>Вы можете объявлять глобальные переменные.</p>
        <pre>
            <code class="language-kotlin hljs" data-
highlighted="yes">
                val PI = 3.14
                var x = 0

                fun incrementX() {
                    x += 1
                }</code>
            </pre>
        </section>

        <div id="clear"> </div>

        <!-- подвал - обязательный элемент сайта -->
        <footer id="footer">
            <h2>Контакты</h2>
            <p>
                <a href="https://github.com/maxi7665">GitHub</a>
                <a href="https://github.com/maxi7665/web-technologies">Исходный код
сайта</a>
            </p>
        </footer>
    </div>
</body>

</html>

```

## 5. Файл «2.css»

```

body {
    background: #f1f1f1;
    color: #000;
    font-family: Arial, sans-serif;
    font-size: 14px;
    padding: 10px;
}

/* body {
    font-family: Arial;

```

```

        padding: 10px;
        background: #f1f1f1;
    } */

#header {
    background: #DCDCDC;
    width: 100%;
    height: 100px;
    display: flex;
    align-items: center;
}

#header h2 {
    text-align: left;
    /* padding: 10px; */
}

#sitename {
    margin-left: auto;
}

#pagename{
    margin-left: auto;
    margin-right: auto;
    font-size: 30;
}

#logo {
    height: 100%;
    width: 16%;
    text-align: left;
}

#logo img{
    height: 100%;
    width: auto;
}

#navigation {
    background: #333;
    width: 100%;
    height: 50px;
}

#sidebar {
    background: #DCDCDC;
    float: left;
    width: 20%;
}

```

```

#content {
    background: #DCDCDC;
    float: right;
    width: 79%;
    /* height: 100%; */
}

#content p {
    text-align: left;
    padding-left: 10px;
    padding-right: 10px;
}

#clear {
    clear: both;
}

#footer {
    background: #333;
    width: 100%;
    /* height: 40px; */
}

#footer h2 {
    color: #f2f2f2;
}

#footer a {
    color: #f2f2f2;
    text-decoration: none;
}

#footer p {
    padding: 3px;
}

#container {
    background: #AAA;
    margin: auto auto;
    text-align: center;
    width: 80%;
    /* height: 800px; */
    padding: 10px;
}

#navigation a {
    float: left;
    display: block;
    color: #f2f2f2;
}

```



```

        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
    }

    /* Change color on hover */
    #navigation a:hover {
        background-color: #ddd;
        color: black;
    }

    #list{
        text-align: left;
    }

    #table {
        margin: 10px;
    }

    #content picture img {
        width: 80%;
    }

    #content pre{
        text-align: left;
    }

    abbr
    {
        font-style: italic;
        color: chocolate;
    }

    #inputDiv{
        text-align: left;
        margin: 10px;
    }

```

## 6. Файл «3.html»

```

<html>

<head>
    <title>Язык Kotlin</title>
    <link rel="shortcut icon" type="image/x-icon" href="img/kotlin_logo.png" />
    <link rel="stylesheet" type="text/css" href="3.css">

    <meta charset="UTF-8">
    <!-- meta - дополнительное задание -->
    <meta name="description" content="Kotlin programming language">

```

```

<meta name="keywords" content="Kotlin, Java, JVM, JDK">
<meta name="author" content="Maxim Bystrov">

<script type="text/javascript">

    function init() {
        /// настройка часов
        renderClocks();
        setInterval(renderClocks, 1000);
    }

    // 1. Вывод информации об авторе
    function alertAuthor() {
        alert("Быстров Максим Z1431");
    }

    // 2. Отрисовка часов
    function renderClocks() {
        var clock1 = document.getElementById("clock1");
        var clock2 = document.getElementById("clock2");
        var clock3 = document.getElementById("clock3");

        var date = new Date();

        var timeZone1 = 'America/New_York';
        var timeZone2 = 'Etc/UCT';
        var timeZone3 = 'Europe/Moscow';

        clock1.innerText = date.toLocaleString('ru-RU', { timeZone: timeZone1
    }) + " " + timeZone1;
        clock2.innerText = date.toLocaleString('ru-RU', { timeZone: timeZone2
    }) + " " + timeZone2;
        clock3.innerText = date.toLocaleString('ru-RU', { timeZone: timeZone3
    }) + " " + timeZone3;
    }

    // 4-5. изменение цвета и размера элемента
    function changeColor() {
        document.getElementById("modifyableDiv").style.backgroundColor =
getRandomColor();
    }

    function getRandomColor() {
        var letters = '0123456789ABCDEF';
        var color = '#';
        for (var i = 0; i < 6; i++) {
            color += letters[Math.floor(Math.random() * 16)];
        }
        return color;
    }
}

```

```

        function changeSize() {
            document.getElementById("modifyableDiv").style.width = 20 +
Math.random() * 80 + "%";
        }

</script>
</head>

<!-- дополнительное задание - использование события окна браузера -->
<body onload="init();">
    <div id="container" class="container">

        <!-- шапка - обязательный элемент сайта -->
        <div id="header">
            <div id="logo"></div>

            <h2 id="sitename">Язык Kotlin</h2>

            <h3 id="pagename">Источники</h3>

        </div>

        <!-- навигация между страницами - обязательный элемент сайта -->
        <div id="navigation" class="container">
            <h2><a href="1.html">История</a>
                <a href="2.html">Документация</a>
                <a href="3.html">Источники</a>
            </h2>
        </div>

    </p>

    <!-- боковая панель -->
    <div id="sidebar">
        <h2>Новости</h2>
        <h3><a href="https://kotlinlang.org/docs/releases.html">Release
Notes</a></h3>
        <h3><a href="https://blog.jetbrains.com/kotlin/2024/11/kotlin-
roundup-kodee-s-top-picks/">Kotlin Roundup</a>
        </h3>
        <h3><a href="https://blog.jetbrains.com/kotlin/2024/10/ktor-3-
0/">Вышел Ktor 3.0</a></h3>
    </div>

    <!-- основной контент -->
    <div id="content">

```

```

        <p><a href="https://kotlinlang.org/docs/home.html">Документация
Kotlin</a></p>
        <p><a href="https://blog.jetbrains.com/kotlin/">Kotlin Blog</a></p>
        <p><a href="https://developer.mozilla.org/en-US/">MDN Web
Docs</a></p>
        <p><a href="https://www.w3schools.com/html/default.asp">W3
Schools</a></p>
        <p><a href="https://www.osp.ru/os/2011/09/13011550">Открытые системы.
Язык программирования Kotlin</a></p>

<hr>

<h1>Сценарии</h1>

<h2>Информация об авторе</h2>

<button onclick="alertAuthor();">Информация об авторе</button>

<h2>Часы в трех часовых поясах</h2>

<div id="clockdiv">
    <div id="clock1"></div>
    <div id="clock2"></div>
    <div id="clock3"></div>
</div>

<h2>Изменение цвета и размера элемента</h2>

<!-- 4-5. изменение цвета и размера элемента -->
<div style="display: grid;">

    <div id="modifyableDiv" style="height: 20px; width: 100%;
background-color: aqua; margin-left: auto; margin-right: auto; margin-bottom:
10px;"></div><br>
    <button onclick="changeColor();" style="width: auto; margin-left:
auto; margin-right: auto;">Изменить цвет</button>
    <button onclick="changeSize();" style="width: auto; margin-left:
auto; margin-right: auto;">Изменить размер</button>
</div>

</div>

<div id="clear"> </div>

<!-- подвал - обязательный элемент сайта -->
<div id="footer">
    <h2>Контакты</h2>
    <p>
        <a href="https://github.com/maxi7665">GitHub</a>
    </p>

```

```

        <a href="https://github.com/maxi7665/web-technologies">Исходный код
сайта</a>
    </p>
</div>
</div>
</body>

</html>

```

## 7. Файл «3.css»

```

body {
    background: #f1f1f1;
    color: #000;
    font-family: Arial, sans-serif;
    font-size: 14px;
    padding: 10px;
}

/* body {
    font-family: Arial;
    padding: 10px;
    background: #f1f1f1;
} */

#header {
    background: #DCDCDC;
    width: 100%;
    height: 100px;
    display: flex;
    align-items: center;
}

#header h2 {
    text-align: left;
    /* padding: 10px; */
}

#sitename {
    margin-left: auto;
}

#pagename{
    margin-left: auto;
    margin-right: auto;
    font-size: 30;
}

```

```

#logo {
    height: 100%;
    width: 16%;
    text-align: left;
}

#logo img{
    height: 100%;
    width: auto;
}

#navigation {
    background: #333;
    width: 100%;
    height: 50px;
}

#sidebar {
    background: #DCDCDC;
    float: left;
    width: 20%;
    /* height: 100%; */
}

#content {
    background: #DCDCDC;
    float: right;
    width: 79%;
    /* height: 100%; */
}

#content p {
    text-align: left;
    padding-left: 10px;
    padding-right: 10px;
}

#clear {
    clear: both;
}

#footer {
    background: #333;
    width: 100%;
    /* height: 40px; */
}

#footer h2 {
    color: #f2f2f2;
}

```

```

#footer a {
    color: #f2f2f2;
    text-decoration: none;
}

#footer p {
    padding: 3px;
}

#container {
    background: #AAA;
    margin: auto auto;
    text-align: center;
    width: 80%;
    /* height: 800px; */
    padding: 10px;
}

#navigation a {
    float: left;
    display: block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

/* Change color on hover */
#navigation a:hover {
    background-color: #ddd;
    color: black;
}

#list{
    text-align: left;
}

#table {
    margin: 10px;
}

#content picture img {
    width: 80%;
}

#content pre{
    text-align: left;
}

```

```
abbr
{
    font-style: italic;
    color: chocolate;
}

#pagename::first-letter
{
    font-size: 35px;
}

#clockdiv {
    text-align: left;
    padding: 10px;
}
```



## **Вывод**

В ходе выполнения третьей лабораторной работы были написаны скрипты на языке клиентских сценариев JavaScript.

Выполнены пункты обязательного и дополнительного задания.

Приобретены навыки по программированию на языке JavaScript, работе с DOM, обработке пользовательского ввода, работе с таймерами, событиями.

Приведены скриншоты, демонстрирующие работу созданных функций, а также полный исходный код разработанного сайта.