

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОННИКИ» (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

**HTML-ЭЛЕМЕНТЫ ДЛЯ СОЗДАНИЯ ИНТЕРАКТИВНЫХ
ПРИЛОЖЕНИЙ. ЗНАКОМСТВО С JAVASCRIPT. ВНЕШНИЕ
СКРИПТЫ**

Отчет по лабораторной работе №3

Обучающийся гр. 571-2

_____ Д. В. Судаков

Руководитель:

к.т.н., доцент кафедры КСУП

_____ Е.С. Мурзин

Томск 2023

Оглавление

Введение	3
Задание 1	4
Задание 2	5
Задание 3	7
Задание 4	10
Задание 5	12
Задание 6	14
Задание 7	16
Ответы на контрольные вопросы	18
Заключение	20

Введение

Цель работы:

- знакомство с HTML-элементами для создания форм и меню;
- знакомство с DHTML;
- изучение языка JavaScript;
- изучение его возможностей создания динамических веб-страниц.

Задание:

1. На каждой страничке реализуйте часики (реализуйте разные варианты отображения часиков);
2. Найдите скрипт, реализующий таймер. Внедрите его на одну из ваших страничек, сопроводив надписью, например, «На моей страничке Вы уже находитесь ... секунд»;
3. Реализуйте вертикальное и горизонтальное меню на одной странице. Если в предыдущих лабораторных работах уже было реализовано меню, то реализуйте обработку нажатия пункта меню;
4. Определить количество нечётных отрицательных элементов в последовательности целых чисел. (Вариант 23)
5. Сколько дней осталось до летних каникул (уточните дату начала каникул по учебному плану)? (Вариант 23)
6. По нажатию на кнопку button_1 должно происходить закрытие окна документа. (Так как варианты в данном задании идут только до 21, а у меня 23, то был взят 2ой вариант)
7. Написать скрипт, который выполнял бы следующее: пользователь получает запрос: "Сколько раз пожелать вам доброго вечера?". В конце программа должна написать "Добрый вечер" столько раз, сколько ее просили. (По аналогии с предыдущим заданием был взят вариант 4, так как в этом задании всего 19 вариантов)

Задание 1

Для реализации часов была создана функция `fulltime()`, в которой инициализируется переменная `time` объектом `Date`. После, при помощи встроенных методов объекта `Date`, такими как `getHours()`, `getMinutes()`, `getSeconds()`, `getDate()`, `getMonth()`, `getFullYear()`, генерируются строки для вывода. После, при помощи метода `getElementById()` из `document`, вставляем текст в теги. И затем, при помощи метода `setTimeout()` запускаем функцию каждые 1000 единиц времени. Код представлен в листинге 1.

```
function fulltime() {  
    var time = new Date();  
    var clock =  
        (time.getHours() > 9 ? time.getHours() : "0" + time.getHours())  
        + ":" + (time.getMinutes() > 9 ? time.getMinutes() : "0" +  
time.getMinutes())  
        + ":" + (time.getSeconds() > 9 ? time.getSeconds() : "0" +  
time.getSeconds());  
    var date = time.getDate() + "." + time.getMonth() + 1 + "." +  
time.getFullYear();  
    document.getElementById("clock").textContent = "Текущее время: " + clock;  
    document.getElementById("date").textContent = "Дата: " + date;  
    setTimeout('fulltime()', 1000);  
}  
fulltime()
```

Листинг 1 – Реализация скрипта для часов.

Задание 2

Для реализации таймера была создана функция `tick()`, в которой прибавляется одна секунда и совершается проход по условиям, что если секунд более или равно 60, то секунды обнуляются, а минуты увеличиваются на 1. Тоже самое было проделано с минутами. После была создана функция `add`, в которой сначала вызывается функция `tick()`, затем отформатированный текст со временем вставляется в тег, после вызывается функция `timer()`. Сама функция `timer` вызывает метод `setTimeout()`, который вызывает метод `add` каждые 1000 единиц времени. Код представлен в листинге 2, результат работы часов и таймера представлен на рисунке 1.

```
var s = 0;
var m = 0;
var h = 0;

function tick() {
    s++;
    if (s >= 60) {
        s = 0;
        m++;
        if (m >= 60) {
            m = 0;
            h++;
        }
    }
}

function add() {
    tick();
    document.getElementById("timer").textContent =
        "Вы находитесь на этой страничке уже: "
        + (h > 9 ? h : "0" + h)
        + ":" + (m > 9 ? m : "0" + m)
        + ":" + (s > 9 ? s : "0" + s);
    timer();
}

function timer() {
    setTimeout(add, 1000);
}
add()
```

Листинг 2 – Реализация скрипта для таймера.

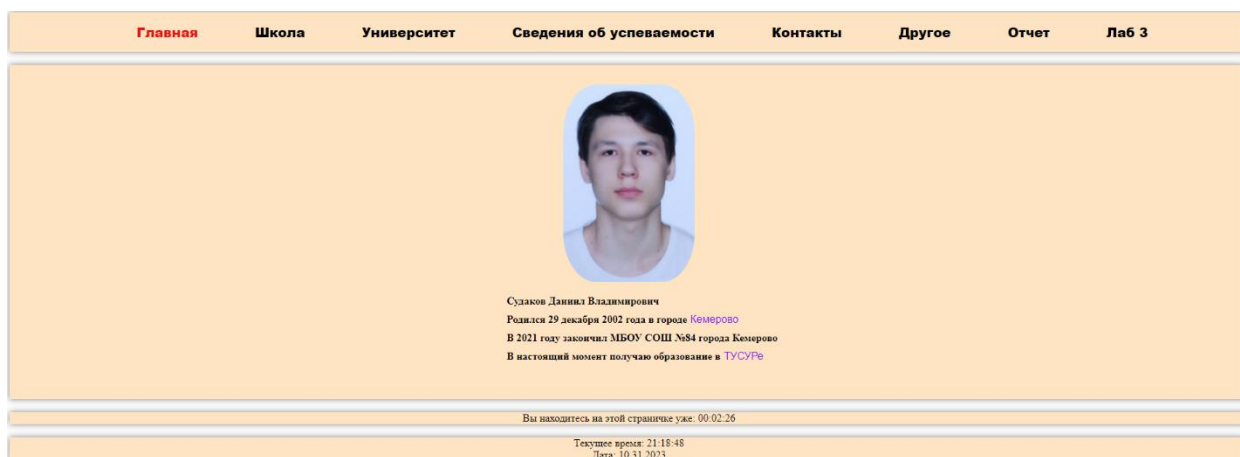


Рисунок 1 – Итоговый вид.

Задание 3

Ранее уже было реализовано перемещение по страницам с помощью горизонтального меню, но через ссылки. Для этой лабораторной было реализовано вертикальное меню через обработчик события onClick.

```
<body>
  <div class="content">
    <header class="content_header">
      <ul class="header_menu">
        <li>
          <a href="../index.html">
            Главная
          </a>
        </li>
        <li>
          <a href="school.html">
            Школа
          </a>
        </li>
        <li>
          <a href="university.html">
            Университет
          </a>
        </li>
        <li>
          <a href="academic-performance.html">
            Сведения об успеваемости
          </a>
        </li>
        <li>
          <a href="contact.html">
            Контакты
          </a>
        </li>
        <li>
          <a href="others.html">
            Другое
          </a>
        </li>
        <li>
          <a class="nav_already">
            Лаб 3
          </a>
        </li>
      </ul>
    </header>
    <div class="wrapper">
      <div class="side_menu">
        <ul class="menu">
```

```

        <li>
            <a onclick="window.location.href='../index.html'">
                Главная
            </a>
        </li>
        <li>
            <a onclick="window.location.href='school.html'">
                Школа
            </a>
        </li>
        <li>
            <a onclick="window.location.href='university.html'">
                Университет
            </a>
        </li>
        <li>
            <a onclick="window.location.href='academic-
performance.html'">
                Сведения об успеваемости
            </a>
        </li>
        <li>
            <a onclick="window.location.href='contact.html'">
                Контакты
            </a>
        </li>
        <li>
            <a onclick="window.location.href='others.html'">
                Разное
            </a>
        </li>
        <li>
            <a class="nav_already">
                Лаб 3
            </a>
        </li>
    </ul>
</div>

```

Листинг 3 – Html код реализация для горизонтального и вертикального меню.

```

.side_menu {
    background-color: bisque;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);
    margin-bottom: 20px;
    width: 20%;
    height: auto;
    display: flex;
    justify-content: center;
}

```



```

    align-items: center;
}

.side_menu li a {
    font-family: 'Montserrat', sans-serif;
    font-weight: 800;
    color: rgb(0, 0, 0);
    transition: .5s all;
}

.menu{
    list-style-type: none;
}

.side_menu li a:hover {
    color: rgb(202, 12, 12);
    transition: .5s all;
}

```

Листинг 4 – Реализация на CSS для вертикального меню.

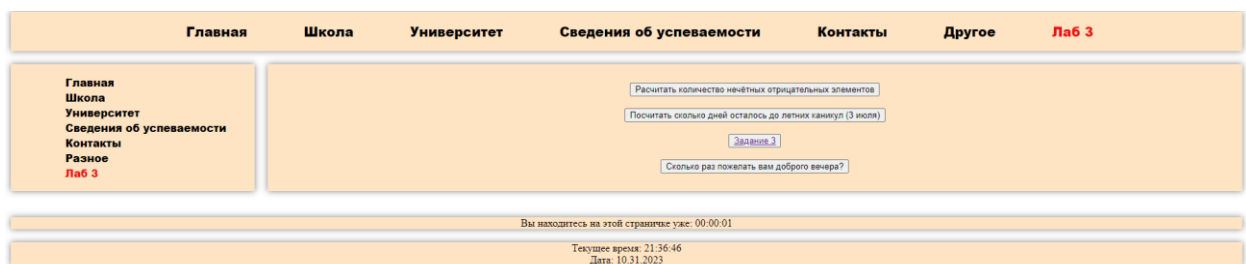


Рисунок 2 – Итоговый вид меню.

Задание 4

Задача:

Сгенерировать последовательность с помощью датчика случайных чисел и обработать согласно варианту задания. Определить количество нечётных отрицательных элементов в последовательности целых чисел. (Вариант 23)

Решение:

Для реализации данного задания был создан файл со скриптом. Для генерации случайных чисел была создана функция `generate()`, в которой задается диапазон возможных значений и минимальное возможное значение. В данном случае приведен пример с диапазоном от -100 до 100. Далее идет функция `generate_array()`, в ней создается пустой массив, который в дальнейшем через цикл заполнится 10 случайными значениями с помощью вызова функции `generate()`. Функция `CountOddNegatives(array)` находит количество нечётных отрицательных элементов в последовательности целых чисел.

```
function generate() {
    return (Math.floor(Math.random() * 201) - 100);
}

function generate_array() {
    let array = [];
    for (let i = 0; i < 10; i++) {
        array.push(generate());
    }
    return array;
}

function countOddNegatives(array)
{
    let count = 0;
    for (let i = 0; i < array.length; i++)
    {
        if (array[i] < 0 && array[i] % 2 !== 0)
        {
            count++;
        }
    }
}
```

```

    }
  }
  return count;
}

let array = document.getElementById("array");
let result = document.getElementById("result");

document.getElementById("btn_task1").addEventListener("click", () =>
{
  let x = generate_array();
  let y = countOddNegatives(x);
  let z = x.map(item => {return " " + item;})
  array.textContent = `Сгенерированный массив значений: ${z}`
  result.textContent = `Количество нечётных отрицательных элементов: ${y}`
})

```

Листинг 5 – Реализация скрипта.

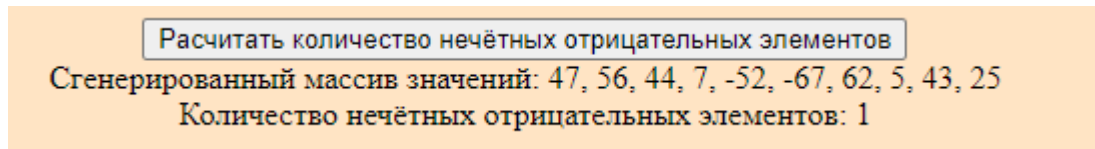


Рисунок 3 – Результат работы скрипта.

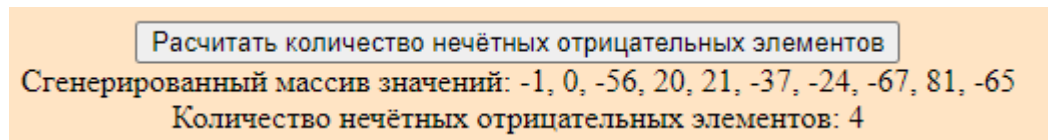


Рисунок 4 – Результат работы скрипта.

Задание 5

Задача:

Написать скрипт который определит сколько дней осталось до летних каникул (уточните дату начала каникул по учебному плану)? (дата начала 3 июля) (Вариант 23)

Решение:

Сначала мы создаем объект `Date`, который будет содержать текущую дату, затем получаем текущий год, используя метод `getFullYear()`, записываем его в переменную `year`. Устанавливаем дату начала летних каникул, используя конструктор `Date()` на 3 июля текущего года. Если текущая дата позже даты начала летних каникул, то мы устанавливаем дату начала следующих летних каникул, добавляя 1 год к текущему году. После мы вычисляем разницу между датой начала летних каникул и текущей датой, используя метод `getTime()`. Результат вычисления будет в миллисекундах.

В переменной `daysLeft` вычисляем количество дней, разделив разницу в миллисекундах на количество миллисекунд в дне и округляя результат до ближайшего целого числа, используя метод `Math.ceil()`.

```
document.getElementById("btn_task2").addEventListener("click", () => {
    let today = new Date();
    let year = today.getFullYear();
    let summerVacationStart = new Date(year, 6, 3);
    if (today > summerVacationStart)
    {
        summerVacationStart = new Date(year + 1, 6, 3);
    }
    let timeDiff = summerVacationStart.getTime() - today.getTime();
    let daysLeft = Math.ceil(timeDiff / (1000 * 3600 * 24));

    let result = document.getElementById("result days left");
    result.textContent = `До каникул осталось ${daysLeft} дней.`
})
```

Листинг 6 – Реализация скрипта.

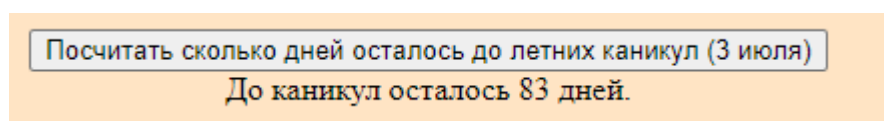


Рисунок 5 – Результат работы скрипта.

Задание 6

Задание:

По нажатию на кнопку `button_1` должно происходить закрытие окна документа. (Так как варианты в данном задании идут только до 21, а у меня 23, то был взят 2ой вариант)

Решение:

Из-за некоторых ограничений `windows.close()`, я создал пустой сайт который будет открываться по нажатию на кнопку с ссылкой "Задание 3"(Рисунок 6) в новой вкладке. На данном сайте присутствует кнопка "Закрыть окно"(Рисунок 7) по нажатию на которую вылезет форма подтверждения закрытия окна(Рисунок 8), после который в зависимости от ответа будет выполнен скрипт.

Создаем функцию ``close_Window()``, которая вызывает подтверждение и если ответ `true(ДА)` то вызывается метод ``window.close()``

```
function close_window() {  
    if (confirm("Вы хотите закрыть окно?"))  
    {  
        window.close("../pages/window.html");  
    }  
}
```

Листинг 7 – Реализация скрипта.

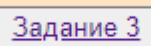
A rectangular button with a thin border and a light blue background, containing the text "Задание 3" in a blue, underlined font.

Рисунок 6 – Кнопка с ссылкой

A small, rectangular button with a thin border and a light gray background, containing the text "Закрыть окно" in a small, black font.

Рисунок 7 – Результат перехода по ссылке.

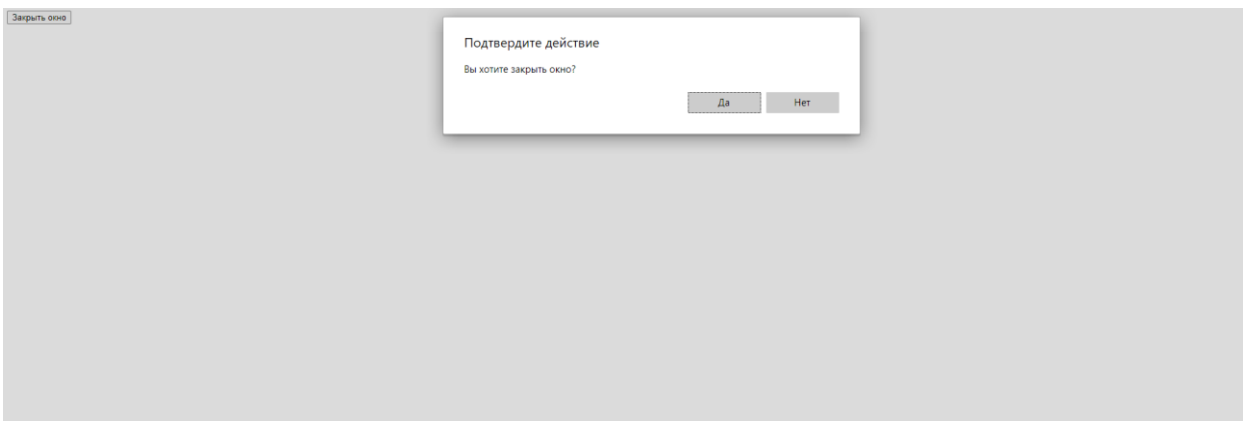


Рисунок 8 – Форма подтверждения закрытия окна.

Задание 7

Задание:

Написать скрипт, который выполнял бы следующее: пользователь получает запрос: "Сколько раз пожелать вам доброго вечера?". В конце программа должна написать "Добрый вечер" столько раз, сколько ее просили. (По аналогии с предыдущим заданием был взят вариант 4, так как в этом задании всего 19 вариантов)

Решение:

Этот код запрашивает у пользователя количество раз, которое нужно написать "Добрый вечер", и затем выводит это сообщение при помощи метода `alert` нужное количество раз.

```
function sayGoodEvening()
{
    let count = prompt("Введите сколько раз пожелать вам доброго вечера");
    for (let i = 0; i < count; i++)
    {
        alert("Добрый вечер!")
    }
}
```

Листинг 8 – Реализация скрипта.

Сколько раз пожелать вам доброго вечера?

Рисунок 9 – Кнопка задания.

The screenshot shows a web application interface with a top navigation bar containing links: "Школа", "Университет", "Объекты", "Другое", and "Личный кабинет". A modal dialog box titled "Ввод данных" (Data Entry) is centered on the screen. It contains the text "Введите сколько раз пожелать вам доброго вечера" (Enter how many times you want to wish you a good evening) and a text input field with the number "4" entered. Below the input field are two buttons: "Продолжить" (Continue) in yellow and "Отмена" (Cancel) in gray. At the bottom of the page, there is a status bar with the text "Вы находитесь на этой страничке уже: 00:06:05" (You have been on this page for: 00:06:05), "Текущее время: 04:09:27" (Current time: 04:09:27), and "Дата: 11.31.2023" (Date: 11.31.2023).

Рисунок 10 – Результат нажатия на кнопку.

The screenshot shows the same web application interface as Figure 10. A modal dialog box titled "Уведомление" (Notification) is centered on the screen. It contains the text "Добрый вечер!" (Good evening!) and a checkbox labeled "Больше не показывать сообщения от этого сайта" (Do not show messages from this site anymore). The checkbox is currently unchecked. A "Закреть" (Close) button is located at the bottom right of the dialog box. The background of the web application is visible behind the dialog box.

Рисунок 11 – n-ое количество уведомлений с сообщением “Добрый вечер”.

Ответы на контрольные вопросы

1) Какой тег используется для организации списка?

`<select>`

2) Какой тег используется для подписки элементов формы?

`<label>`

3) Какой тег используется для группировки элементов формы?

`<optgroup>`

4) Какой объект используется для работы с датой?

`Date`

5) Как создать объект `Date`?

`var date = new Date();`

6) Какие методы используются для получения текущей даты?

`getDate, getHours, getMinutes, getSeconds, getYear, getMonth, getDay, getFullYear`

7) Какие методы используются для установки текущей даты?

`setDate, setHours, setMinutes, setSeconds, setYear, setMonth, setTime`

8) Что такое DHTML?

Динамический HTML, в котором предусмотрена интеграция со специальными языками программирования.

9) Для чего используется язык программирования JavaScript?

-Создание динамических страниц, т.е. страниц, содержимое которых может меняться после загрузки

-Проверка правильности заполнения пользовательских форм

-Решение «локальных» задач спомощью сценариев

10) Опишите правила формирования имени переменной JavaScript?

Имя переменной должно начинаться с буквы латинского алфавита или с символа подчеркивания. Само имя может включать буквы латинского алфавита, цифры и знак подчеркивания.

11) Какой тег используется для размещения сценария в документе?

`<script>`

12) Опишите средства, используемые для отладки сценария.

Весьма удобным средством отладки является Консоль, входящая в состав браузеров.

13) Опишите варианты исполнения сценария.

При загрузке документа, при возникновении события, а также при вызове его интерпретатором.

14) Какой тег используется для организации формы?

`<form>`

15) Перечислите стандартные элементы управления.

Стандартными являются все элементы управления, которые можно поместить на HTML-форму (однострочное текстовое поле, поле для ввода пароля, флажки, переключатели, кнопки как пользовательские, так и выполняющие стандартные действия, поля имен файлов и т.д.)

16) Как организовать вертикальное и горизонтальное меню?

При помощи тега `<table>`

17) Как создать внутренний и внешний скрипт?

Внешний – создать в директории с проектом файл с расширением .js, после чего подключить его к странице внутри тега `<head>` при помощи тега `<script>` с аргументов `src`, который в свою очередь принимает путь до скрипта. Внутренний – описать скрипт внутри тега `<script>`

Заключение

В ходе выполнения лабораторной работы были изучены основы языка программирования JavaScript и возможности работы со страницей.