

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

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (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

В. К. Медянкин

(подпись)

(И.О. Фамилия)

\_\_\_\_\_  
(дата)

Руководитель:

к.т.н., доцент кафедры КСУП

(должность, ученая степень, звание)

Е. С. Мурзин

\_\_\_\_\_  
(оценка)

(подпись)

(И.О. Фамилия)

\_\_\_\_\_  
(дата)

Томск 2022

## Оглавление

Введение .....	3
Задания.....	4
1 РЕАЛИЗАЦИЯ СКРИПТОВ «ЧАСИКИ» И «ТАЙМЕР» .....	5
2 ПЕРЕМЕЩЕНИЕ ПО СТРАНИЦАМ ЧЕРЕЗ ОБРАБОТЧИК СОБЫТИЯ ..	8
3 ЗАДАНИЯ ПО ВАРИАНТУ .....	9
3.1 Задание 1 .....	9
3.2 Задание 2 .....	11
3.3 Задание 3 .....	13
3.4 Задание 4 .....	14
4 Ответы на контрольные вопросы.....	16
Заключение .....	18

## Введение

По-настоящему эффектными страницы получаются, если использовать при их создании следующие возможности: управление элементами HTML - документов или даже самим браузером, генерацию новых документов, организацию диалогового взаимодействия с пользователем, выполнение каких-то расчетов и обработку данных. Для реализации данных возможностей используется DHTML – динамический HTML, в котором предусмотрена интеграция со специальными языками программирования. Программы, написанные на этих языках, называются сценариями (scripts).

Из языков, разработанных для программирования сценариев, можно привести JavaScript, VBScript, JScript как наиболее простые. В настоящих лабораторных работах будут рассмотрены приемы программирования только на языке JavaScript.

Язык программирования сценариев JavaScript является значительно упрощенной версией популярного языка Java. В отличие от Java, JavaScript является интерпретируемым (как и другие языки программирования сценариев). Это значит, что при загрузке HTML-документа браузер находит в нем все программы JavaScript, проверяет их на наличие ошибок, после чего выполняет команду за командой.

Целью работы: знакомство с элементами для создания форм и меню, знакомство с DHTML, изучение языка JavaScript, изучение его возможностей для создания динамических веб-страниц.

## Задания

1. Для хранения внешних скриптов создайте специальную папку;
2. На каждой страничке реализуйте часики (реализуйте разные варианты отображения часиков);
3. Найдите скрипт, реализующий таймер. Внедрите его на одну из ваших страничек, сопроводив надписью, например, «На моей страничке Вы уже находитесь ... секунд»;
4. Реализуйте вертикальное и горизонтальное меню на одной странице. Если в предыдущих лабораторных работах уже было реализовано меню, то реализуйте обработку нажатия пункта меню;
5. Выполните задания, представленные ниже;
6. Оформите отчет, в котором перечислите, объясните назначение и опишите созданные скрипты;
7. Сгенерировать последовательность с помощью датчика случайных чисел и обработать согласно варианту задания; (15 вариант)
8. Выполнить задание на работу с датой и временем согласно варианту задания; (15 вариант)
9. Напишите функции-обработчики для событий согласно варианту задания; (15 вариант)
10. Напишите скрипт согласно варианту задания. (15 вариант)

## 1 РЕАЛИЗАЦИЯ СКРИПТОВ «ЧАСИКИ» И «ТАЙМЕР»

Для реализации часов была создана функция `fulltime()`, которая инициализирует переменную `time` объектом `Date`. После, при помощи встроенных методов объекта `Date`, генерируются строки для вывода. После, при помощи метода `getElementById()` из `document`, вставляется текст в теги. После при помощи метода `setTimeout()` запускается функция каждую секунду.

```
1 function fulltime() {
2   var time = new Date();
3   var clock =
4     (time.getHours() > 9 ? time.getHours() : "0" + time.getHours())
5     + ":" + (time.getMinutes() > 9 ? time.getMinutes() : "0" +
6       time.getMinutes())
7     + ":" + (time.getSeconds() > 9 ? time.getSeconds() : "0" +
8       time.getSeconds());
9
10  var date = time.getDate() + "." + time.getMonth() + "." +
11    time.getFullYear();
12
13  document.getElementById("clock").textContent =
14    "Текущее время: " + clock;
15  document.getElementById("date").textContent =
16    "Дата: " + date;
17  setTimeout('fulltime()', 1000);
18 }
19 fulltime()
```

Листинг 1.1 – реализация скрипта для часов

Для реализации таймера создана функция `tick()`, которая прибавляет одну секунду и через условие, если секунд более или равно 60, обнуляет секунды, а минуты увеличивает на 1. Такая же работа проходит между минутами и часами. После создается функция `add`, в которой сначала вызывается функция `tick()`, затем вставляется форматированный текст со временем в тег, а после вызывается функция `timer()`. Сама функция `timer` вызывает метод `setTimeout()`, который вызывает метод `add()` каждую секунду.

```

1 var s = 0;
2 var m = 0;
3 var h = 0;
4
5 function tick() {
6     s++;
7     if (s >= 60) {
8         s = 0;
9         m++;
10        if (m >= 60) {
11            m = 0;
12            h++;
13        }
14    }
15 }
16
17 function add() {
18     tick();
19     document.getElementById("timer").textContent =
20         "В ы  н а  х о  д и  т е  с  ь  н а  э  т о  й  с  т р а  н и  ч  к е  у  ж е  :  "
21         + (h > 9 ? h : "0" + h)
22         + ":" + (m > 9 ? m : "0" + m)
23         + ":" + (s > 9 ? s : "0" + s);
24     timer();
25 }
26
27 function timer() {
28     setTimeout(add, 1000);
29 }
30 add()

```

Листинг 1.2 – реализация скрипта для таймера

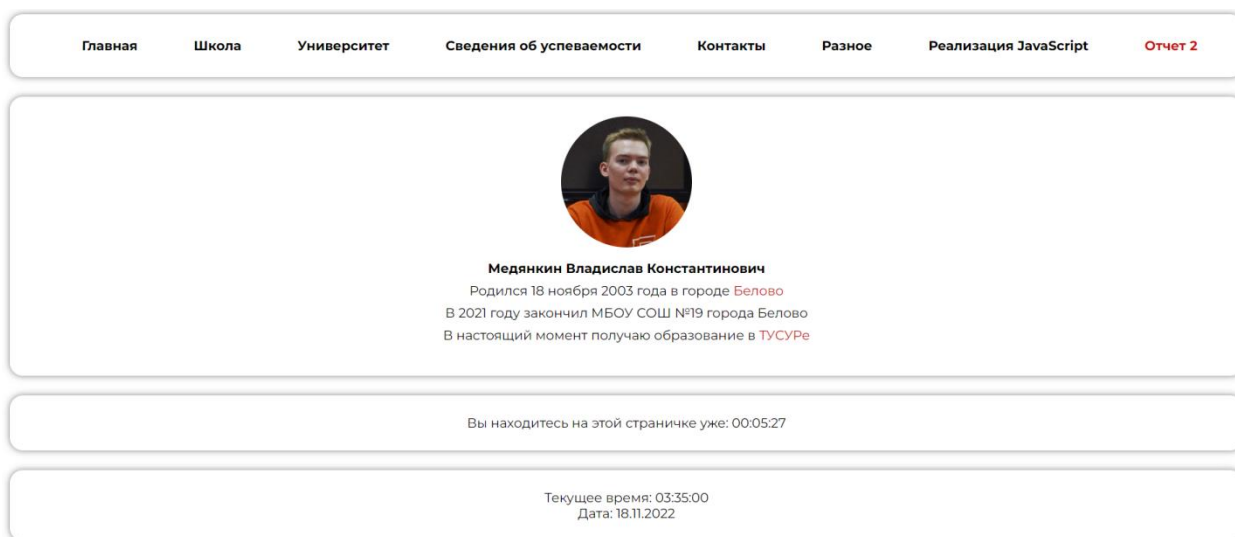


Рисунок 1.1 – результат работы скриптов

## 1 ПЕРЕМЕЩЕНИЕ ПО СТРАНИЦАМ ЧЕРЕЗ ОБРАБОТЧИК СОБЫТИЯ

Ранее я уже реализовывал перемещение по страницам с помощью горизонтального меню, но делал это просто через ссылки. Поэтому для этой лабораторной работы было реализовано вертикальное меню через обработчик события onClick.

```
1 <div class="menu_body">
2   <ul class="menu">
3     <li>
4       <a onclick="window.location.href='../index.html'">
5         Г л а в н а я </a>
6     </li>
7     <li>
8       <a onclick="window.location.href='school.html'">
9         Ш к о л а </a>
10    </li>
11    <li>
12      <a onclick="window.location.href='high-education.html'">
13        У н и в е р с и т е т </a>
14    </li>
15    <li>
16      <a onclick="window.location.href='academic-performance.html'">
17        С в е д е н и я   о б   у с п е в а е м о с т и </a>
18    </li>
19    <li>
20      <a onclick="window.location.href='contact.html'">
21        К о н т а к т ы </a>
22    </li>
23    <li>
24      <a onclick="window.location.href='other.html'">
25        Р а з н о е </a>
26    </li>
27    <li>
28      <a class="nav_already">
29        Р е а л и з а ц и я   JavaScript </a>
30    </li>
31  </ul>
32 </div>
```

Листинг 2.1 – реализация меню через обработчик события



## 2 ЗАДАНИЯ ПО ВАРИАНТУ

### 1.1 Задание 1

Задача:

Сгенерировать последовательность с помощью датчика случайных чисел и обработать согласно варианту задания. Выбрать максимальный из модулей элементов последовательности.

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 `get_max_modul(array)` находит максимальное значение в массиве по модулю.

```
1 function generate() {
2     return (Math.floor(Math.random() * 201) - 100);
3 }
4
5 function generate_array() {
6     var array = [];
7     for (let i = 0; i < 10; i++) {
8         array.push(generate());
9     }
10    return array;
11 }
12
13 function get_max_modul(array) {
14     var max = -101;
15     for (let i = 0; i < array.length; i++) {
16         if (Math.abs(array[i]) > Math.abs(max)) {
17             max = array[i];
18         }
19     }
```

```

20     return max;
21 }
22
23 let array = document.getElementById("array");
24 let result = document.getElementById("result");
25
26 document.getElementById("btn_task1").addEventListener("click", () => {
27     let x = generate_array();
28     let y = get_max_modul(x);
29     let z = x.map(item => {
30         return " " + item;
31     })
32
33     array.textContent =
34     `Сгенерированный массив значений:
35 ${z}`
36     result.textContent =
37     `Максимальное значение по модулю:
38     ${y}`
39 })

```

Листинг 3.1.1 – реализация скрипта

Найти максимальное значение по модулю.  
 Сгенерированный массив значений: 93, -87, 28, 100, -88, -74, 96, -47, -33, 34  
 Максимальное значение по модулю: 100

Рисунок 4.1.1 – результат работы программы

Найти максимальное значение по модулю.  
 Сгенерированный массив значений: 78, -87, -97, -87, -73, -81, -91, 76, -23, 76  
 Максимальное значение по модулю: -97

Рисунок 4.1.2 – результат работы программы

## 1.2 Задание 2

Задача:

Составить скрипт определяющий сколько суток осталось до дней весеннего и осеннего равноденствий (22 марта и 22 сентября)?

Решение:

Для реализации этого скрипта необходим объект Date, который инициализируется в переменную date\_now. Затем, в переменных spring\_equinoxes и autumn\_equinoxes также инициализируется объект Date, но с параметрами на вход, которые обозначают: следующий год, месяц март/сентябрь и 22 число. Также создается переменная one\_day, в которую записывается значение равное одному дню. Далее идет создается кнопка с обработчиком события, при нажатии на которую в переменные date1, date2 и date3, с помощью встроенного метода getTime(), записываются значения из переменных выше, соответственно. После высчитывается разница дат равноденствия и делится на значения одного дня. Получившиеся округленные значения записываются, как x и y.

```
1 var date_now = new Date();
2 var spring_equinoxes = new Date(date_now.getFullYear() + 1, 2, 23);
3 var autumn_equinoxes = new Date(date_now.getFullYear() + 1, 8, 23);
4 var one_day = 1000 * 60 * 60 * 24;
5
6 let springEquinoxes = document.getElementById("spring equinoxes");
7 let autumnEquinoxes = document.getElementById("autumn equinoxes");
8
9 document.getElementById("btn_task2").addEventListener("click", () => {
10   let date1 = spring_equinoxes.getTime();
11   let date2 = autumn_equinoxes.getTime();
12   let date3 = date_now.getTime();
13
14   let x = Math.round((date1 - date3) / one_day);
15   let y = Math.round((date2 - date3) / one_day);
16
17   springEquinoxes.textContent = `Дней осталось до весен
18 него равноденствия: ${x}`
```

```
19 autumnEquinoxes.textContent = `Дней осталось до осенн  
20 его равноденствия: ${y}`  
21 })
```

Листинг 3.2.1 – реализация скрипта

Сколько суток осталось до дней весеннего и осеннего равноденствий (22 марта и 22 сентября)?

Дней осталось до весеннего равноденствия: 95

Дней осталось до осеннего равноденствия: 279

Рисунок 4.2.1 – результат работы программы

### 1.3 Задание 3

Задача:

Напишите функции-обработчики для событий согласно варианту задания. При изменении значения в текстовом поле `text_1` должно выводиться: «Быстро сделай все, как было!».

Решение:

Для решения этой задачи объявим переменные `input` и `text_1`, также в коде самой странички стоит прописать тег `<input>` и блок с `id="text_1"`. Используя обработчик события `onChange` на `input`, создадим переменную `x`, в которой будет содержаться исходный текст. После сделаем условие, согласно которому, если значение внутри `x` не будет равно изначальному значению, то выведем строку `text_1`, иначе - ничего.

```
1 var input = document.getElementById("input_text");
2 var text_1 = document.getElementById("text_1");
3 input.addEventListener("change", (e) => {
4     let x = "Не трогай, это на новый год!";
5     if (x !== e.target.value) {
6         text_1.textContent =
7         "Быстро сделай все, как было!";
8     }
9     else {
10         text_1.textContent = "";
11     }
12 })
```

Листинг 3.3.1 – реализация скрипта

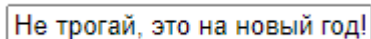


Рисунок 4.3.1 – стандартное состояние

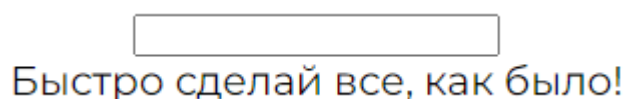


Рисунок 4.3.2 – измененная строка

## 1.4 Задание 4

Задача:

Напишите скрипт согласно варианту задания. Пользователь вводит в поле формы данные. Затем эти данные будут использованы для поиска в Yahoo (или в другой поисковой системе). При подаче запроса должно выскакивать окошко с надписью "Сейчас поищем..."

Решение:

Используя обработчик события `onClick` на кнопке, вызовем метод `alert` для появления всплывающего окна с подтверждением, после которого текст, введенный в тег `<input>` на страничке, преобразуется в ссылку через секунду.

```
1 var searcher = document.getElementById("searcher");
2 var button = document.getElementById("btn_search");
3
4 button.addEventListener("click", () => {
5     alert("У ж е и щ е м в а ш з а п р о с !")
6     setTimeout(() => {
7         window.location.href=
8         `https://www.yandex.ru/search/?lr=67&offline_search=1&text=${searcher.value}`,
9         1000);
10 })
```

Листинг 3.4.1 – реализация скрипта

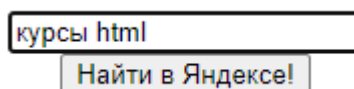


Рисунок 4.4.1 – введенный запрос

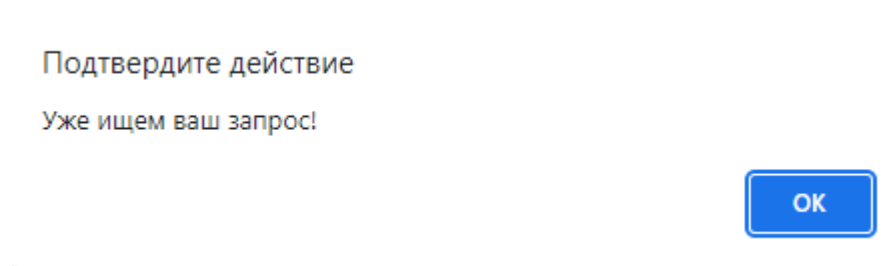


Рисунок 4.4.2 – окно с подтверждением

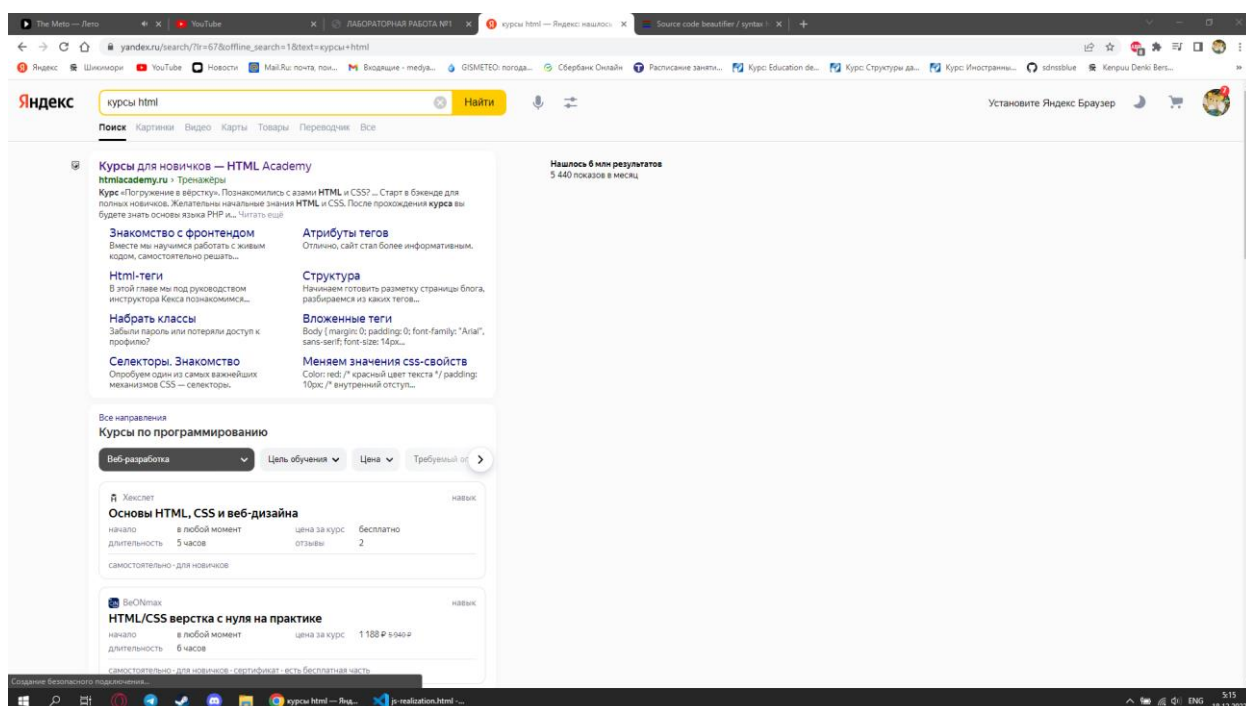


Рисунок 4.4.3 – результат работы скрипта

## 2 Ответы на контрольные вопросы

**1) Какой тег используется для организации списка?**

`<select>`

**2) Какой тег используется для подписи 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 time = new Date();`

**6) Какие методы используются для получения текущей даты?** getDate, getHours, getMinutes, getSeconds, getDay, getMonth, getYear, 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?**

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

2. Проверка правильности заполнения пользовательских форм.

3. 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?**

Как из вариантов, 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 аргументом `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.