

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)

**ДОМ-МОДЕЛЬ. МЕТОДЫ ОБРАБОТКИ
СТРОКОВЫХ ПЕРЕМЕННЫХ ПОСРЕДСТВОМ
ЯЗЫКА JAVASCRIPT. РЕГУЛЯРНЫЕ
ВЫРАЖЕНИЯ**

ОТЧЕТ ПО
ЛАБОРАТОРНОЙ РАБОТЕ № 4

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)

Оглавление

Введение	3
Задания.....	4
1 Дерево	6
2 Форма регистрации	9
Заключение	14

Введение

Объектная модель документа DOM является программным интерфейсом для доступа и манипулирования документом (документ в терминологии SGML/HTML/XML). Координирующая роль в DOM принадлежит консорциуму W3C - <http://www.w3.org/DOM/>.

Модель DOM позволяет получить доступ, как к элементам разметки, так и свойствам, позволяющими управлять браузером (получить доступ к текущему документу, открыть окно с новым документом, получить данные из другого документа), а также установить обработчики событий на определенные действия мыши, клавиатуры, таймера и пр. Отдельно обрабатываются формы.

Основной принцип программирования с использованием JavaScript заключается в управлении свойствами браузера и манипулировании элементами документа, согласно модели DOM.

Целью работы: изучение приемов работы с DOM и изучение языка JavaScript: обработка строковых переменных, регулярные выражения.

Задания

1. Написать JavaScript-код для вывода дерева элементов страницы, с которой этот код запущен. Отступы для отображения формировать как символ .

В процессе выполнения работы реализовать следующие пункты:

- сформировать страницу с произвольным кодом разметки, нообеспечить уровень вложенности внутри элемента <body> не менее 3;
 - добавить внутри элемента <body> секцию <div>, предназначенную для вывода результата обхода дерева элементов страницы;
 - выбрать способ активации рекурсивной программы обхода дерева элементов, реализовать и подключить эту программу;
 - реализовать вывод на странице.
2. В отчете привести код страницы с программой обхода и примеры обхода дерева элементов страницы.
 3. Реализуйте форму. На ней расположите не менее 15 элементов (полей формы). Следующие элементы являются обязательными: фамилия, имя, отчество, контактный телефон, электронный адрес. Другие поля формы разработайте самостоятельно. Используйте все представленные в данной лабораторной работе элементы управления формы. Расположите элементы для ввода текстовых данных так, чтобы они были выровнены по левому краю.
 4. Реализуйте проверку правильности введенных данных: пусть у Вас будут на форме обязательные и необязательные для заполнения поля; необходимо проверить на корректность их ввода, например, в поле телефон должны быть только цифры; в ФИО не может быть цифр; электронный адрес должен содержать символы до и после знака @, последняя метка не более 3-х

символов и т.п. Для проверки правильности введенных данных можно использовать регулярные выражения.

5. Оформите отчет, в котором перечислите, объясните назначение и приведите примеры использования всех созданных скриптов.

1 Дерево

Для реализации алгоритма была сформирована html страница со следующей разметкой:

Начало дерева
Нулевой уровень
Первый уровень
Второй уровень
Третий уровень
Конец дерева

Рисунок 1.1 – сформированная страница

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="../../../css/style_tree.css">
7     <link rel="stylesheet" href="../../../css/style.css">
8     <title>Д е р е в о</title>
9 </head>
10
11 <body onload=generateTree(document.documentElement);>
12     <div class="content">
13         <div id="tree_generation" class="content_body">
14             <h1>Н а ч а л о  д е р е в а </h1>
15             <div>Н у л е в о й  у р о в е н ь
16                 <div class="test">
17                     <b>П е р в ы й  у р о в е н ь </b>
18                     <div class="test">
19                         <i>В т о р о й  у р о в е н ь </i>
20                         <div class="test">
21                             <u>Т р е т и й  у р о в е н ь </u>
22                         </div>
23                     </div>
24                 </div>
25             <h1>К о н е ц  д е р е в а </h1>
26         </div>
27         <br>
```



```

22
23     return text;
24 }
25 }
26
27 function generateTree() {
28     var tree = findChild(document.getElementById("tree_generation"), "", "");
29     document.getElementById("tree").innerHTML += tree;
30 }

```

Листинг 1.2 – реализация алгоритма

Начало дерева

Нулевой уровень

Первый уровень

Второй уровень

Третий уровень

Конец дерева

```

DIV
  H1
    Начало дерева
  DIV
    Нулевой уровень
    DIV
      В
        Первый уровень
        DIV
          I
            Второй уровень
            DIV
              U
                Третий уровень
    H1
      Конец дерева
  BR
  HR
  DIV

```

Рисунок 1.2 – результат работы скрипта

2 Форма регистрации

Была сформирована страница, содержащая в себе 15 полей, некоторые из которых являются необязательными к заполнению.

The image shows a web form for registration. It is divided into two main sections: 'Контактная информация:' (Contact information) and 'Пароль' (Password). The 'Контактная информация:' section contains 13 fields: 'Фамилия:*' (Surname), 'Имя:*' (Name), 'Отчество:*' (Patronymic), 'Пол:*' (Gender), 'Адрес:*' (Address), 'Дата рождения:*' (Date of birth) with a date picker, 'Место рождения:*' (Place of birth), 'Гражданство:*' (Citizenship), 'Статус:*' (Status), 'Номер телефона:*' (Phone number), 'Дополнительный номер телефона:' (Additional phone number), 'Электронная почта:*' (Email), and 'Дополнительная электронная почта:' (Additional email). The 'Пароль' section contains two fields: 'Пароль:*' (Password) and 'Повторите пароль:*' (Repeat password). Below these sections is a button labeled 'Отправить' (Send).

Рисунок 2.1 – сформированная страница

Затем был создан скрипт, который при помощи регулярных выражений обрабатывает значения полей и после заносит все ошибки в один общий текст и выводит на экран.

Для восьми полей было реализовано пять регулярных выражений. Для полей «Фамилия», «Имя», «Отчество», «Пол», «Адрес», «Место рождения», «Гражданство», «Статус», реализовано выражение, что поля должны содержать буквы и их количество должно равняться 3 или более.

Для поля «Дата рождения» было реализовано выражение, что сначала в строке должна идти цифра и при этом один или более раз, затем

идет знак «.», после снова должна идти цифра и при этом один или более раз, затем снова «.» и в конце снова цифра и при этом один или более раз. То есть содержать формат даты DD.MM.YYYY.

Для поля «Номер телефона» было реализовано выражение, что строка должна содержать цифры. К этому выражению так же был применен модификатор поиска g, чтобы найти все совпадения.

Для поля «Электронная почта» было реализовано выражение, что сначала в строке должны идти буквы, цифры или нижнее подчеркивание один или более раз, затем идет знак «@», после идут символы от а до z и/или от А до Z, после идет знак «.», и после него так же идут символы от а до z и/или от А до Z встречающиеся от 2 до 3 раз.

Для поля «Пароль» было реализовано выражение, что строка должна содержать любые буквы, цифры или подчеркивание от 8 символов. Также дальше идет проверка на совпадение первого и второго пароля.

```
1 function reg() {
2     var errors = "";
3     var reg1 = /^D{3}/;
4     var reg2 = /(\d+).(\d+).(\d+)/;
5     var reg3 = /\d/g;
6     var reg4 = /^w+[a-zA-Z]+?\.[a-zA-Z]{2,3}$/;
7     var reg5 = /^w{8}/
8
9     var lastnameBool = reg1.test(document.regForm.lastname.value);
10    var firstnameBool = reg1.test(document.regForm.firstname.value);
11    var fathersnameBool = reg1.test(document.regForm.fathersname.value);
12    var genderBool = reg1.test(document.regForm.gender.value);
13    var addressBool = reg1.test(document.regForm.address.value);
14    var birthplaceBool = reg1.test(document.regForm.birthplace.value);
15    var citizenshipBool = reg1.test(document.regForm.citizenship.value);
16    var statusBool = reg1.test(document.regForm.status.value);
17    var birthdayBool = reg2.test(document.regForm.birthday.value);
18    var phonenumberBool = document.regForm.phonenumber.value.match(reg3);
19    var emailBool = reg4.test(document.regForm.email.value);
20    var passwordBool = reg5.test(document.regForm.password.value);
21
```

```

22     if (phonenumberBool == null) {
23         phonenumberBool = false;
24     }
25     else if (phonenumberBool.length == 11) {
26         phonenumberBool = true;
27     }
28     else {
29         phonenumberBool = false;
30     }
31
32     if (!lastnameBool) errors += "\nФамилия должна состо
33 ять от 3 букв или более.";
34
35     if (!firstnameBool) errors += "\nИмя должно состоять
36 от 3 букв или более.";
37
38     if (!fathersnameBool) errors += "\nОтчество должно сос
39 тоять от 3 букв или более.";
40
41     if (!genderBool) errors += "\nПол должен содержать о
42 т 3 букв или более.";
43
44     if (!addressBool) errors += "\nАдрес должен contain
45 ь от 3 букв или более.";
46
47     if (!birthplaceBool) errors += "\nМесто рождения долж
48 но содержать от 3 букв или более.";
49
50     if (!citizenshipBool) errors += "\nГражданство должно
51 содержать от 3 букв или более.";
52
53     if (!statusBool) errors += "\nСтатус должен contain
54 ь от 3 букв или более.";
55
56     if (!birthdayBool) errors += "\nНеправильный формат
57 даты рождения. Необходимо DD.MM.YYYY.";
58
59     if (!phonenumberBool) errors += "\nНомер телефона долж
60 ен состоять из 11 цифр.";
61
62     if (!emailBool) errors += "\nНеверный формат электр
63 онного адреса.";
64
65     if (!passwordBool) errors += "\nПароль должен состоя
66 ь минимум из 8 символов.";
67     else if (passwordBool == true &&

```

```

68     document.regForm.password.value !=
69     document.regForm.passwordtwo.value)
70     errors += "\nП а р о л и   н е   с о в п а д а ю т .";
71
72     if (errors == "") errors = "У с п е ш н о";
73     alert(errors);
74 }

```

Листинг 2.1 – реализация скрипта.

Подтвердите действие

- Отчество должно состоять от 3 букв или более.
- Пол должен содержать от 3 букв или более.
- Адрес должен содержать от 3 букв или более.
- Место рождения должно содержать от 3 букв или более.
- Гражданство должно содержать от 3 букв или более.
- Статус должен содержать от 3 букв или более.
- Неправильный формат даты рождения. Необходим DD.MM.YYYY.
- Номер телефона должен состоять из 11 цифр.
- Неверный формат электронного адреса.
- Пароль должен состоять минимум из 8 символов.

OK

Место рождения:*

Гражданство:*

Статус:*

Номер телефона:*

Дополнительный номер телефона:

Электронная почта:*

Дополнительная электронная почта:

Пароль

Пароль:*

Повторите пароль:*

Отправить

Рисунок 2.2 – результат работы скрипта при некорректном вводе

Подтвердите действие

Успешно

OK

Контактная информация:

Фамилия:*

Имя:*

Отчество:*

Пол:*

Адрес:*

Дата рождения:*

Место рождения:*

Гражданство:*

Статус:*

Номер телефона:*

Дополнительный номер телефона:*

Электронная почта:*

Дополнительная электронная почта:*

Медянкин

Владислав

Константинович

Мужской

Кирова, 56А

30.11.2022

Кемерово, Белово

Российская Федерация

Не женат

89130717846

88005553535

medyankin03@mail.ru

medyankin03@gmail.com

Пароль

Пароль:*

Повторите пароль:*

.....

.....

Отправить

Рисунок 2.3 – результат работы скрипта при корректном вводе

Заключение

В ходе выполнения лабораторной работы были изучены приемы работы с DOM, а также способы обработки строковых переменных и регулярные выражения.