

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)

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

Отчет по лабораторной работе №4

Обучающийся гр. 571-2

_____ Д. В. Судаков

Руководитель:

к.т.н., доцент кафедры КСУП

_____ Е.С. Мурзин

Томск 2023

Оглавление

Введение	3
Задания	4
Дерево	5
Форма регистрации	8
Заключение	12

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. Основным принципом 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, но обеспечить уровень вложенности внутри элемента не менее 3; - добавить внутри элемента секцию, 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 5 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

Дерево

Для реализации алгоритма была сформирована html страница со следующей разметкой:

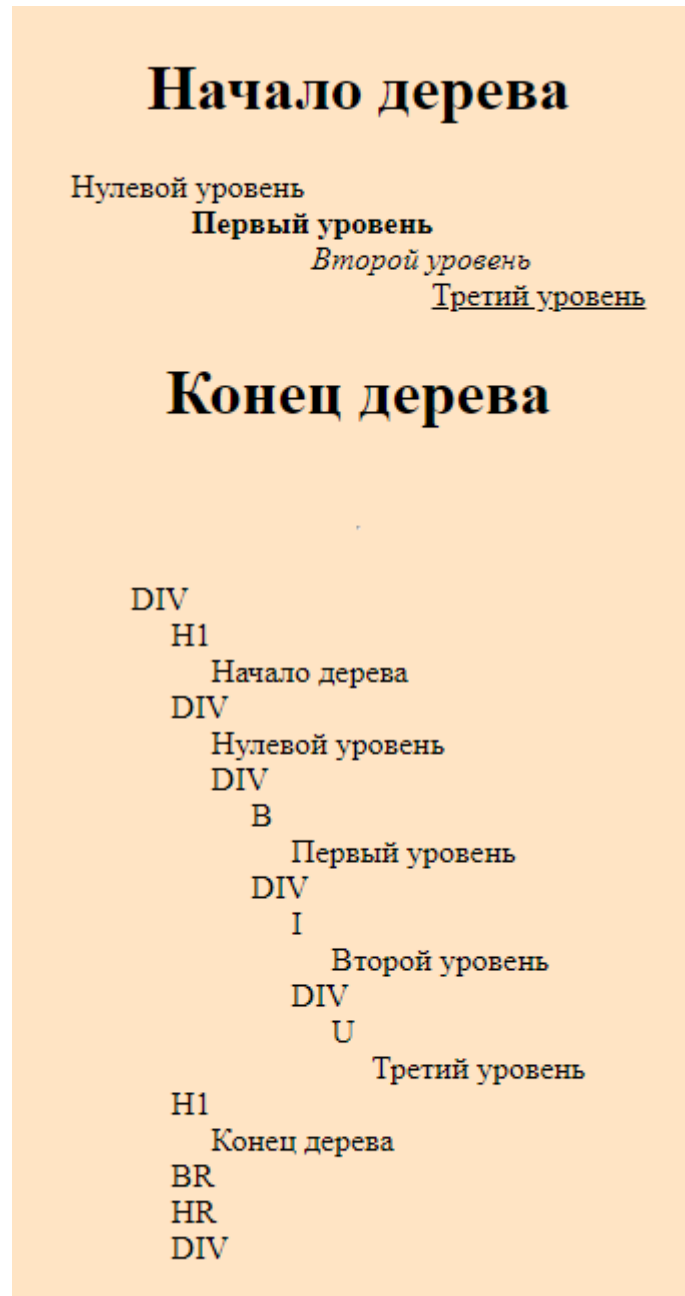


Рисунок 1 - Итоговая страница с деревом.

```
<!DOCTYPE html>  
<html lang="en">  
  
  <head>  
    <meta charset="utf-8">  
    <link rel="stylesheet" href="../../css/style_lab4.css">  
    <link rel="stylesheet" href="../../css/main_style.css">
```


возвращается конечное дерево. После создается функция `generateTree()`, в которой запускается рекурсивная функция `findChild()` и выводится результат.

```
var addSpace = "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&";

function findChild(element, space, text) {
    var count = element.childNodes.length;

    if (element.nodeName == "#text") {
        if (element.data.trim().replace("\n") != "") {
            text = text + "<br>" + space + element.data;
        }
    }
    else {
        text = text + "<br>" + space + element.nodeName;
    }

    if (count == 0) {
        return text;
    }
    else if (count > 0) {
        for (var i = 0; i < count; i++) {
            text = findChild(element.childNodes[i], space + addSpace, text);
        }

        return text;
    }
}

function generateTree() {
    var tree = findChild(document.getElementById("tree_generation"), "", "");
    document.getElementById("tree").innerHTML += tree;
}
```

Листинг 2 - Код реализации скрипта.

Форма регистрации

Реализованная форма регистрации состоящая из 15-ти полей, также в форме реализована валидация вводимых данных при помощи регулярных выражений. Поля обязательные к заполнению помечены “*”.

The image shows a web registration form with a light yellow background. It is divided into two main sections. The first section, titled 'Контактная информация:', contains 13 input fields. The first 12 fields are labeled with names followed by an asterisk, indicating they are required: 'Фамилия:', 'Имя:', 'Отчество:', 'Пол:', 'Адрес:', 'Дата рождения:', 'Место рождения:', 'Гражданство:', 'Статус:', 'Номер телефона:', 'Дополнительный номер телефона:', 'Электронная почта:', and 'Дополнительная электронная почта:'. The 'Дата рождения:' field is a date picker showing 'ДД.ММ.ГГГГ'. The second section, titled 'Пароль', contains two required fields: 'Пароль:' and 'Повторите пароль:'. Below these sections is a button labeled 'Отправить'.

Контактная информация:	
Фамилия:*	<input type="text"/>
Имя:*	<input type="text"/>
Отчество:*	<input type="text"/>
Пол:*	<input type="text"/>
Адрес:*	<input type="text"/>
Дата рождения:*	<input type="text" value="ДД.ММ.ГГГГ"/>
Место рождения:*	<input type="text"/>
Гражданство:*	<input type="text"/>
Статус:*	<input type="text"/>
Номер телефона:*	<input type="text"/>
Дополнительный номер телефона:	<input type="text"/>
Электронная почта:*	<input type="text"/>
Дополнительная электронная почта:	<input type="text"/>

Пароль	
Пароль:*	<input type="password"/>
Повторите пароль:*	<input type="password"/>

Рисунок 3 - Страница с формой регистрации.

Реализован скрипт, который при помощи регулярных выражений обрабатывает значения полей и после заносит все ошибки в переменную errors, а затем выводит ее на экран про помощи alert().

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 10 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 должно содержать формат даты ДД.ММ.ГГГГ.

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 может содержать любые буквы, цифры или подчеркивание от 8 символов, а затем происходит сравнение первого и второго пароля на совпадение.

```
function reg() {
    var errors = ""
    var reg1 = /^D{3}/;
    var reg2 = /(\d+).(\d+).(\d+)/;
    var reg3 = /\d/g;
    var reg4 = /^w+@[a-zA-Z]{2,3}$ /;
    var reg5 = /^w{8}/

    var lastnameBool = reg1.test(document.regForm.lastname.value);
    var firstnameBool = reg1.test(document.regForm.firstname.value);
    var fathersnameBool = reg1.test(document.regForm.fathersname.value);
    var genderBool = reg1.test(document.regForm.gender.value);
    var addressBool = reg1.test(document.regForm.address.value);
    var birthplaceBool = reg1.test(document.regForm.birthplace.value);
    var citizenshipBool = reg1.test(document.regForm.citizenship.value);
    var statusBool = reg1.test(document.regForm.status.value);
    var birthdayBool = reg2.test(document.regForm.birthday.value);
    var phonenumberBool = document.regForm.phonenumber.value.match(reg3)
    var emailBool = reg4.test(document.regForm.email.value);
    var passwordBool = reg5.test(document.regForm.password.value);
```

```

    if (phonenumBool == null) {
        phonenumBool = false;
    }
    else if (phonenumBool.length == 11) {
        phonenumBool = true;
    }
    else {
        phonenumBool = false;
    }

    if (!lastnameBool) errors += "\nФамилия должна состоять от 3 букв или
более.";
    if (!firstnameBool) errors += "\nИмя должно состоять от 3 букв или более.";
    if (!fathersnameBool) errors += "\nОтчество должно состоять от 3 букв или
более.";
    if (!genderBool) errors += "\nПол должен содержать от 3 букв или более.";
    if (!addressBool) errors += "\nАдрес должен содержать от 3 букв или более.";
    if (!birthplaceBool) errors += "\nМесто рождения должно содержать от 3 букв
или более.";
    if (!citizenshipBool) errors += "\nГражданство должно содержать от 3 букв или
более.";
    if (!statusBool) errors += "\nСтатус должен содержать от 3 букв или более.";
    if (!birthdayBool) errors += "\nНеправильный формат даты рождения. Необходим
DD.MM.YYYY.";
    if (!phonenumBool) errors += "\nНомер телефона должен состоять из 11
цифр.";
    if (!emailBool) errors += "\nНеверный формат электронного адреса.";

    if (!passwordBool) errors += "\nПароль должен состоять минимум из 8
символов.";
    else if (passwordBool == true &&
        document.regForm.password.value !=
        document.regForm.passwordtwo.value)
        errors += "\nПароли не совпадают.";

    if (errors == "") errors = "Заявка на регистрацию успешно передана!";
    alert(errors);
}

```

Листинг 2 - Код реализации скрипта.

Уведомление

Фамилия должна состоять от 3 букв или более.
Имя должно состоять от 3 букв или более.
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.

Заккрыть

Пароль

Пароль: *
Повторите пароль: *

Отправить

Рисунок 4 - Результат выполнения скрипта при некорректном вводе данных.

Уведомление

Успешно

Заккрыть

Пол: *
Адрес: *
Дата рождения: *
Место рождения: *
Гражданство: *
Статус: *
Номер телефона: *
Дополнительный номер телефона:
Электронная почта: *
Дополнительная электронная почта:

Муж
Томск
12 . 12 . 2002
Караганда
Российская Федерация
Женат
88005553535

pyrkinVASIA@gmail.com

Пароль
Пароль: *
Повторите пароль: *

Отправить

Рисунок 5 - Результат выполнения скрипта при правильном вводе данных.

Заключение

В ходе данной лабораторной работы были изучены: dom-модель, методы обработки строковых переменных посредством языка javascript, регулярные выражения. Был реализован вывод дерева элементов страницы и форма регистрации.