Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждениевысшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

СИСТЕМУПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

(ТУСУР)

Кафедра КСУП

Отчет по лабораторной работе

по дисциплине «Веб-технологии»

Тема: «DOM-модель. Методы обработки строковых переменных посредством языка JavaScript. Регулярные выражения.»

Студент гр. 571- 2

Вьюгин Кирилл Вадимович

10 декабря 2022 г.

Томск 2022

Оглавление

Оглавление

Введение....3
Теоретическая часть....5
Основная часть....8
1 Реализация скрипта «Дерево»....8
2 Реализация формы с информацией....10
Заключение....15

Введение

Цель работы:

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

Задачи лабораторной работы:

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 <body> не менее 3;
   - добавить внутри элемента <body> 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 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.

Теоретическая часть

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/.

Напомним, что HTML-страница является документом со строгой разметкой. 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.

Базовый уровень функциональности документа обеспечивается объектами, поддерживаемыми даже самыми старыми браузерами. Эта иерархия объектов представляет объектную модель документов уровня 0 (Document Object Model level0 - DOM0), которая исторически появилась до официальных спецификаций W3C.

Применительно к клиентскому Javascript, основным объектом является Window, который ссылается на текущее окно браузера. Остальные объекты, иерархия которых здесь представлена, являются свойствами корневого объекта Window. Почти все эти объекты имеют много полезных свойств, с ними связаны события и методы, использование которых позволяет создавать сценарии, обеспечивающие необходимую функциональность.

Регулярные выражения — это средство обработки данных. Задача, требующая замены или поиска текста, может быть достаточно просто и изящно решена с помощью регулярных выражений. И хотя максимальный эффект от регулярных выражений можно добиться при использовании

серверных языков (например, PHP), всё же не стоит недооценивать возможности этого приложения и на стороне клиента.

Основные понятия

Регулярное выражение (regular expression) — средство для обработки строк или последовательность символов, определяющая шаблон текста.

Модификатор — предназначен для "инструктирования" регулярного выражения.

Метасимволы — специальные символы, которые служат командами языка регулярных выражений.

Общее описание

Регулярное выражение задаётся как обычная переменная, только вместо кавычек используется слеш, например:

var reg=/рег_выражение/

Регулярное выражение может состоять из обычных символов; в этом случае оно будет соответствовать заданной комбинации символов в строке. Например, выражение /счет/ соответствует выделенным подстрокам в следующих строках: "счетчик", "пересчет",

"счеталка".

Регулярные выражения представляют собой образцы для поиска заданных комбинаций символов в текстовых строках (такой поиск называется сопоставлением с образцом). Существует два способа присваивания переменным регулярных выражений:

- использование инициализатора объекта: var re = /pattern/switch?
- использование конструктора RegExp:

var re = new RegExp("pattern"[,"switch"]?).

здесь pattern - регулярное выражение, a switch - необязательные опции поиска.

Инициализаторы объекта, например, var re = /ab+c/, следует применять в тех случаях, когда значение регулярного выражения остается неизменным во время работы сценария. Такие регулярные выражения компилируются в

процессе загрузки сценария и, следовательно, выполняются быстрее.

Вызов конструктора, например, var re = new RegExp("ab+c"), следует применять в тех случаях, когда значение переменной будет меняться. Если вы собираетесь использовать регулярное выражение несколько раз, то имеет смысл скомпилировать его методом compile для более эффективного поиска образцов.

Примечание: регулярное выражение не может быть пустым: два символа // подряд задают начало комментария. Поэтому для задания пустого регулярного выражения используйте выражение /.?/.

Регулярные выражения используются методами exec и test объекта RegExp и методами match, replace, search и split объекта String. Если необходимо просто проверить, содержит ли данная строка подстроку, соответствующую образцу, то используются методы test или search. Если же необходимо извлечь подстроку (или подстроки), соответствующие образцу, то используются методы exec или match. Метод replace обеспечивает поиск заданной подстроки и замены ее на другую строку, а метод split позволяет разбить строку на несколько подстрок, основываясь на регулярном выражении или обычной текстовой строке.

Основная часть

1 Реализация скрипта «Дерево».

```
var addSpace = "     ";

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

    // добавление элемента в дерево
    if (element.nodeName == "#text")
    {
        // проверка, чтобы не было пустой строки
        if (element.data.trim().replace("\n") != "")
        {
            text = text + "<br>" + space + element.data;
        }
    }
    else
    {
        text = text + "<br>" + space + element.nodeName;
    }

    // если дочерних элементов не найдено, то вернуть текст
    // если же найдено, пройтись по всем
    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.body, "", "");
    document.getElementById("output").innerHTML += tree;
}
```

Листинг 1.1 – реализация скрипта для Дерева.

```
<html lang="ru">
    <head>
        <title>Дерево</title>
        <meta charset="UTF-8">
        <script src="../js/script_tree.js" type="text/javascript"></script>
    </head>
    <body onload="generateTree()">
        <h1>Страница для вывода дерева элементов страницы</h1>
        <div>
            <div>
                <div>
                    Это блок в блоке, который находится в блоке
                </div>
            </div>
```

```
        </div>
        <h1>Конец</h1>
        <div id="output" style="border:1px solid black">
            <h1>Вывод</h1>
        </div>
    </body>
</html>
```

Листинг 1.2 – применение скрипта для Дерева.

Страница для вывода дерева элементов страницы

Это блок в блоке, который находится в блоке

Конец

Вывод

```
BODY
    H1
        Страница для вывода дерева элементов страницы
    DIV
        DIV
            DIV
                Это блок в блоке, который находится в блоке
    H1
        Конец
    DIV
        H1
            Вывод
```

Рисунок 1.1 – работа дерева.

2 Реализация формы с информацией.

```
function inf()
{
    var errors = ""
    var reg1 = /[A-Za-zA-Яа-яЁё]/;
    var reg2 = /[0-9]{6}/;
    var reg3 = /\d/g;
    var reg4 = /^\w+@[a-zA-Z]+?\.[a-zA-Z]{2,3}$/
    var reg5 = /^[ 0-9]+$/
    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 addressBool = reg1.test(document.regForm.Address.value);
    var countryBool = reg1.test(document.regForm.country.value);
    var cityBool = reg1.test(document.regForm.city.value)
    var sportBool = reg1.test(document.regForm.favorite_sport.value)
    var postalCode = reg2.test(document.regForm.postal_code.value);
    var phonenumberBool = document.regForm.phonenumber.value.match(reg3)
    var emailBool = reg4.test(document.regForm.email.value);
    var ageBool = reg5.test(document.regForm.Age.value);
    var heightBool = reg5.test(document.regForm.height.value);
    var foot_sizetBool = reg5.test(document.regForm.foot_size.value);
    var weightBool = reg5.test(document.regForm.weight.value);
    if (phonenumberBool == null)
    {
        phonenumberBool = false;
    }
    else if (phonenumberBool.length == 11)
    {
        phonenumberBool = true;
    }
    else
    {
        phonenumberBool = false;
    }

    if (!lastnameBool) errors+= "\nФамилия должна состоять из букв.";
    if (!firstnameBool) errors+= "\nИмя должно состоять из букв.";
    if (!fathersnameBool) errors+= "\nОтчество должно состоять из букв.";
    if (!addressBool) errors+= "\nАдрес должен состоять из букв.";
    if (!countryBool) errors+= "\nСтрана должна состоять из букв.";
    if (!cityBool) errors+= "\nГород должен состоять из букв.";
    if (!sportBool) errors+= "\nСпорт должен состоять из букв.";
    if (!postalCode) errors+= "\nПочтовый индекс должен состоять из 6 цифр.";
    if (!phonenumberBool) errors+= "\nНомер телефона должен состоять из 11
цифр.";
    if (!emailBool) errors+= "\nНеверный формат электронного адреса.";
    if (!ageBool) errors+= "\nВозраст - число";
    if (!heightBool) errors+= "\nРост - число";
    if (!foot_sizetBool) errors+= "\nРазмер ноги - число";
    if (!weightBool) errors+= "\nВес - число";

    if (errors == "") errors = "Успешно";
    alert(errors);
}
```

Листинг 2.1 – реализация скрипта для формы.

```
<html lang="ru">
    <head>
        <title>Информация</title>
        <meta charset="UTF-8">
```

```
        <style>
            #fields {
                float:left;
            }

            #fields label {
                float: left;
                padding-right:10px;
            }

            #fields div {
                clear: both;
                text-align: right;
                line-height:25px;
            }
            span {
                color: red;
            }
        </style>
        <script src="../js/script_information.js"
type="text/javascript"></script>
    </head>
    <body>
        <form name="regForm">
            <div id="fields">
                <fieldset>
                    <legend>Информация о вас</legend>
                    <div>
                        <label>
                            Фамилия:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="lastname">
                    </div>
                    <div>
                        <label>
                            Имя:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="firstname">
                    </div>
                    <div>
                        <label>
                            Отчество:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="fathersname">
                    </div>
                    <div>
                        <label>
                            Возраст:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="number" required name="Age">
                    </div>
```

```
                    <div>
                        <label>
                            Адрес:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="Address">
                    </div>
                    <div>
                        <label>
                            Номер телефона:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="tel" required name="phonenumber">
                    </div>
                    <div>
                        <label>
                            Электронная почта:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="email" required name="email">
                    </div>
                    <div>
                        <label>
                            Страна:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="country">
                    </div>
                    <div>
                        <label>
                            Рост:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="number" required name="height">
                    </div>
                    <div>
                        <label>
                            Размер ноги:
                            <span>
                                *
                            </span>
                        </label><input type="number" required
name="foot_size">
                    </div>
                    <div>
                        <label>
                            Вес:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="number" required name="weight">
                    </div>
```

```
                    <div>
                        <label>
                            Почтовый индекс:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="number" required name="postal_code">
                    </div>
                    <div>
                        <label>
                            Город проживания:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="city">
                    </div>
                    <div>
                        <label>
                            Любимый цвет:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="color" required name="favorite_color">
                    </div>
                    <div>
                        <label>
                            Любимый вид спорта:
                            <span>
                                *
                            </span>
                        </label>
                        <input type="text" required name="favorite_sport">
                    </div>
                </fieldset>
                <input type="button" value="Отправить" onclick="inf()">
            </div>
        </form>
    </body>
</html>
```

Листинг 2.2 – применение скрипта для формы.

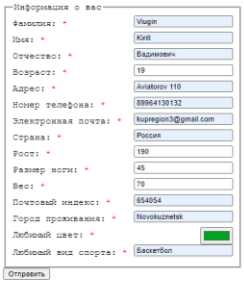

Рисунок 2.3 – работа фомы.

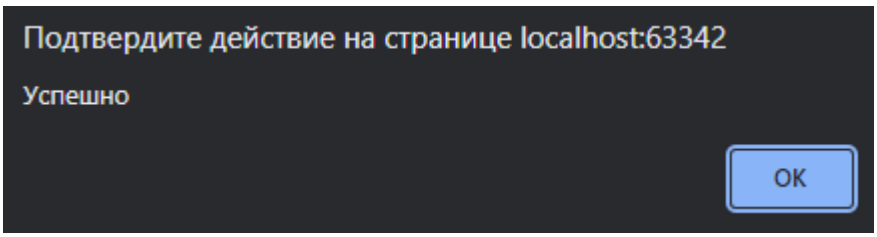

Рисунок 2.4 – работа фомы.

Заключение

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