



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ – 09.03.01 Информатика и вычислительная техника

О Т Ч Е Т

по лабораторной работе № 3

Название: Javascript. Добавление динамического поведения HTML. Средства отладки

Дисциплина: Языки интернет-программирования

Студент

ИУ6-32Б

(Группа)

(Подпись, дата)

Джафаров Э.Э.

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Петрова Я.С.

(И.О. Фамилия)

Москва, 2022

Цель работы:

Целью работы является изучение основных принципов программирования на языке Javascript, изучение способов обхода узлов модели документа DOM, изучение принципов динамического формирования кода страницы HTML, а также получение практических навыков отладки Javascript-приложений.

Задание:

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:

1. Сформировать страницу с произвольным кодом разметки, но обеспечить уровень вложенности внутри элемента не менее 3.
2. добавить внутри элемента секцию, предназначенную для вывода результата обхода дерева элементов страницы.
3. Выбрать способ активации рекурсивной программы обхода дерева элементов, реализовать и подключить эту программу.
4. При проходе по узлам разметки обеспечить отладочный вывод в консоль. Привести в отчете содержимое консоли.
5. Реализовать вывод на странице.

Решение:

1. HTML:

```
<!DOCTYPE html>
<html lang="en" id="first">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="../CSS/lr3.css" rel="stylesheet">
  <title>Playing with tree</title>
  <script src="../JS/lr3.js" type="text/javascript"></script>
</head>

<body>
  <div>
    <h2 class="header">This is a header</h2>
    <div>
      <p class="just">This is just text</p>
    </div>
  </div>

  <div id="result">
    <button class="magic" id="result" onclick="generateTree(document.documentElement)">This is magic button</button>
  </div>
</body>

</html>
```

2. CSS

```
body {  
  align-content: center;  
  font-size: 18px;  
  font-family: "Helvetica Neue", sans-serif;  
  background-color: blue;  
}
```

```
.magic {  
  display: inline-block;  
  padding: 15px 25px;  
  font-size: 24px;  
  cursor: pointer;  
  text-align: center;  
  text-decoration: none;  
  outline: none;  
  color: black; #000;  
  background-color: #4CAF50;  
  border: none;  
  border-radius: 15px;  
  box-shadow: 0 9px #999;  
}
```

```
.magic:hover {  
  background-color: #3e8e41;  
}
```

```
.magic:active {  
  background-color: #3e8e41;  
  box-shadow: 0 5px #666;  
  transform: translateY(4px);  
}
```

```

@keyframes flicker {

  0%,
  18%,
  22%,
  25%,
  53%,
  57%,
  100% {

    text-shadow:
      0 0 4px ■ #000,
      0 0 11px ■ #000,
      0 0 19px ■ #000,
      0 0 40px ■ #0fa,
      0 0 80px ■ #0fa,
      0 0 90px ■ #0fa,
      0 0 100px ■ #0fa,
      0 0 150px ■ #0fa;

  }

  20%,
  24%,
  55% {
    text-shadow: none;
  }

}

.just {
  animation: flicker 1.5s infinite alternate;
  color: ■ #fff;
}

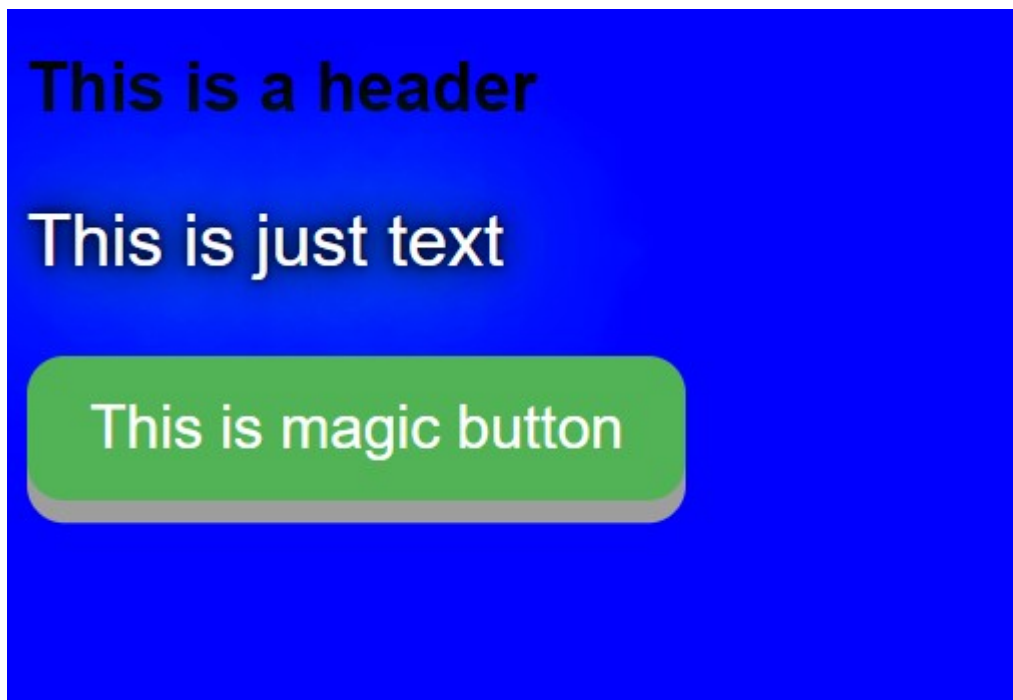
```

3. JavaScript

```
var a = new Array("area", "base", "basefont", "bgsound", "br", "col", "command", "embed", "hr", "img", "input", "isindex",
function generateTree(elem) {
    var child = elem.firstChild;
    var ul = document.createElement('ul');
    while (child) {
        if (a.indexOf(child.tagName.toLowerCase()) !== -1) {
            console.log('Lonely tag', child.tagName);
            var flag = false;
        }
        else {
            console.log('Open', child.tagName);
            var flag = true;
        }
        var li = document.createElement('li');
        li.innerHTML = child.tagName;
        var list = generateTree(child);
        li.appendChild(list);
        ul.appendChild(li);
        if (flag) {
            console.log('Close', child.tagName);
        }
        child = child.nextElementSibling;
    }
    elem.appendChild(ul);
    document.getElementById('result').style.display = "none";
    return ul;
}
```

4. Результат

4.1 До нажатия



4.2 После нажатия



4.3 Консоль

Open HEAD	lr3.js:11:21
Lonely tag META	3 lr3.js:7:21
Lonely tag LINK	lr3.js:7:21
Open TITLE	lr3.js:11:21
Close TITLE	lr3.js:20:21
Open SCRIPT	lr3.js:11:21
Close SCRIPT	lr3.js:20:21
Close HEAD	lr3.js:20:21
Open BODY	lr3.js:11:21
Open DIV	lr3.js:11:21
Open H2	lr3.js:11:21
Close H2	lr3.js:20:21
Open DIV	lr3.js:11:21
Open P	lr3.js:11:21
Close P	lr3.js:20:21
Close DIV	2 lr3.js:20:21
Open DIV	lr3.js:11:21
Open BUTTON	lr3.js:11:21
Close BUTTON	lr3.js:20:21
Close DIV	lr3.js:20:21
Close BODY	lr3.js:20:21

Вывод:

Я изучил основные принципы программирования на языке Javascript, изученил способы обхода узлов модели документа DOM и принципы динамического формирования кода страницы HTML, а также получил практические навыки отладки Javascript-приложений.