



**«Московский государственный технический университет  
имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ \_\_\_\_\_ ИНФОРМАТИКИ И СИСТЕМ УПРАВЛЕНИЯ  
КАФЕДРА \_\_\_\_\_ КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ 6)

## **О т ч е т**

**по лабораторной работе № 4**

**Дисциплина: Языки программирования**

**Название лабораторной работы: Javascript. Добавление динамического поведения HTML. Средства отладки.**

Студент гр. ИУ6-34

(Подпись, дата)

**Габолаев Г.К.**

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

**Изория В.Ф.**

(И.О. Фамилия)

## Задание 1:

Написать Javascript-код для вывода дерева элементов страницы, с которой этот код запущен. Отступы для отображения формировать как символ &nbsp;

В процессе выполнения работы реализовать следующие пункты:

1. Сформировать страницу с произвольным кодом разметки, но обеспечить уровень вложенности внутри элемента <body> не менее 3.
2. добавить внутри элемента <body> секцию <div>, 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.

В отчете привести код страницы с программой обхода, отладочный вывод в консоль и примеры обхода дерева элементов страницы.

## Решение:

```
1.<!DOCTYPE html>
2.<html>
3.
4.<head>
5.    <meta charset="utf-8">
6.    <title>Tests</title>
7.    <style>
8.        .tree {
9.            font-weight: bold;
10.           font-family: monospace;
11.           height: 1px;
12.        }
13.
14.        .tree h4 {
15.            margin: 0;
16.        }
17.
18.        button {
19.            margin: 2px;
20.        }
21.    </style>
22.</head>
23.
24.<body>
25.<div id="tooltip"></div>
26.<div id="signinModal" class="modalWrapper">
27.    <div class="singupModalWrapper">
28.        <div class="mainModalTitle">
29.            <span class="loginAccessTitle-en">Login or sign up</span>
30.        </div>
31.        <div class="modal-close js_closeModal"></div>
32.        <div class="modal-body clearfix">
33.            <form autocomplete="off" class="js-loginFormModal">
34.                <input type="hidden" class="js-redirect" name="redirect"
35.                    value="Fs7rVmIEvUmYNVocHXTG5D9xF5mGODEA2zbeq92AJGc." />
36.                <input type="hidden" name="token" id="token"
37.                    value="MTUwNjcyNDgxNyQhOqtNnN3W_EglAy0919YgvOMcWVlZo3vjs1jSMCwnc-52nhSv8RFXKo6JmRl0RhhyHXy
38.                    bBHJbC-Bj2ABOp0k." />
39.                <input type="hidden" name="remember_me" value="1"/>
40.                <input type="hidden" name="from" value="pc_login_modal:index">
41.            <div class="leftSide floatLeft loginColumnLeft-en">
42.                <p class="signinError" style="display:none;"></p>
43.                <p id="signinLoggingin" style="display:none;">Logging in...</p>
44.            <div><input id="usernameModal" placeholder="Username (6+
    characters) "
```

```

45.         class="js-signinUsernameModal signup_field"
         name="username" maxlength="50" type="text"
46.         value=""/>

```

```

110.         <ul class="filterListItem dropdownWrapper" style="display:
    none;">
111.             <li class="active" name="public"><span>Public</span></li>
112.             <li name="private"><span>Private</span></li>
113.         </ul>
114.     </div>
115.     <input type="hidden">
116. </div>
117. <div class="modal-footer clearfix">
118.     <input type="submit" name="submit" value="Save" id="MHP1138-add-
    playlist"
119.         class="submit-button buttonBase orangeButton">
120.     <a class="buttonBase greyButton cancelButton"
    onclick="clearModalPlaylist();"
    data-modal="createPlaylist">Cancel</a>
121. </div>
122. </form>
123. </div>
124. </div>
125. </div>
126. <div class="tree">
127. </div>
128. <button onclick="nodesTree(document.body, 1)">Build a tree</button>
129. <script>
130.     var tree = document.getElementsByClassName('tree');
131.
132.     function nodesTree(node, depth) {
133.         var newNode;
134.         if (node.nodeName && node.className !== 'tree') {
135.             newNode = document.createElement('h4');
136.             printLine = ('&nbsp;'.repeat(2 * depth) + node.nodeName);
137.             newNode.innerHTML = printLine;
138.             console.log(printLine);
139.             tree[0].appendChild(newNode);
140.         } else {
141.             return;
142.         }
143.         for (var i in node.children)
144.             nodesTree(node.children[i], depth + 1);
145.         console.log(printLine);
146.         tree[0].appendChild(newNode.cloneNode(true));
147.     }
148. </script>
149. </div>
150. </body>
151.
152. </html>

```

The screenshot shows a web browser window with the address bar displaying the file path: file:///Users/gabolaev/Desktop/BMSTU/web/Labs/Lab4/index.html. The browser's developer tools are open, showing the DOM tree on the left and the console on the right.

The DOM tree on the left shows a structure with nested DIV, UL, LI, SPAN, and INPUT elements. The console on the right shows the output of the JavaScript code, including the recursive function nodesTree and the resulting HTML structure.

**Вывод:** Изучена работа с JavaScript на базовом уровне. Создана функция, выполняющая рекурсивное построение DOM дерева HTML страницы. js - игрушка дьявола ежи.