



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 09.03.03

О Т Ч Е Т

по лабораторной работе № 3

Название: Javascript. Добавление динамического поведения HTML. Средства отладки.

Дисциплина: Языки интернет программирования

Студент

ИУ6-35Б

(Группа)

(Подпись, дата)

С.С.Кретов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2021

ПОРЯДОК ВЫПОЛНЕНИЯ ЗАДАНИЯ

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 `<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.

```
<!DOCTYPE html>
<html lang="ru">

<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">
    <title>ICS6 Bootstrap</title>
    <link
        href="https://e-learning.bmstu.ru/iu6/pluginfile.php/7546/mod_folder/content/0/bootstrap.min.css?forcedownload=1"
        rel="stylesheet">
    <style>
        html {
            position: relative;
            min-height: 100%;
        }

        body {
            margin-bottom: 40px;
        }

        main {
            margin-top: 15px;
        }

        footer {
            height: 40px;
            position: absolute;
            bottom: 0;
```



```

        </tr>
        <tr>
            <th scope="row"></th>
            <td>Пароль:</td>
            <td><input type="password" name="PasswordField"></td>
        </tr>
        <tr>
            <th scope="row"></th>
            <td>Пол:</td>
            <td><input type="radio" name="Sex" value="M">M<input type="radio" name="Sex"
value="Ж">Ж <br>
            </td>
        </tr>
        <tr>
            <th scope="row"></th>
            <td colspan="2" align="right"><input type="submit" name="SubmitBtn"></td>
        </tr>
    </table>
</div>
<div class="col-4">
    <table border="2" class="table">
        <tr>
            <th colspan="1" scope="col">#</th>
            <th colspan="3" scope="col">Числитель</th>
            <th colspan="3" scope="col">Знаменатель</th>
        </tr>
        <tr>
            <th scope="col"><strong>Время пар</strong></td>
            <th scope="col"><strong>Название пары</strong></td>
            <th scope="col"><strong>Кабинет</strong></td>
            <th scope="col" colspan="2"><strong>Название пары</strong></td>
            <th scope="col"><strong>Кабинет</strong></td>
        </tr>
        <tr>
            <td>8:30-10:05</td>
            <td rowspan="2">Электротехника</td>
            <td rowspan="2">Каф. Физики</td>
            <td colspan="2" rowspan="2"></td>
        </tr>
        <tr>
            <td>10:15-11:50</td>
        </tr>
        <tr>
            <td>12:00-13:35</td>
            <td colspan="4" align="center">Базы данных</td>
        </tr>
        <tr>
            <td>13:50-14:40</td>
            <td colspan="4" align="center">Языки интернет программирования</td>
        </tr>
    </table>

```

```

        </table>
    </div>
</div>
<div class="row">
    <div class="col">
        Здесь <em>не очень</em> важная информация<br>
        Здесь <strong>особенно важный</strong> текст<br>
        Здесь физическая формула <var>1</var>*<var>w</var>*<var>h</var><br>
        Здесь химическая формула H<sub>2</sub><sub>0</sub><br>
        Здесь математическая формула
        <var>a</var><sup>2</sup>+<var>b</var><sup>2</sup>=<var>c</var><sup>2</sup><br>
        We are working with <mark>Java</mark><br>
        <code>
        <pre>

@Test
public void testGetAllPosts() {
    HttpResponse httpResponse = ConnectionUtils.getUtils().sendGETRequest(Hosts.POSTS.getAll());
    List<Post> posts = Arrays.asList(JSONUtils.getObjectFromHttpResponse(httpResponse, Post[].class));
    posts.sort(Comparator.comparingInt(Post::getId));
    Assert.assertEquals(posts, posts, "List should be sort");
    Assert.assertEquals(httpResponse.getStatusLine().getStatusCode(), HttpStatus.SC_OK);
}

        </pre>
    </code>
    <samp>Test complete successfully</samp><br>
    </div>
</div>
<div>
    <input type="button" value='Показать дерево'
        onclick="el.innerHTML =
'';console.clear();printTree(document.documentElement.childNodes)"><br>
    <p id='my-tree'></p>
    </div>
</main>
<footer class="my-primary">
    <div class="container">
        <p class="footer-text">Footer</p>
    </div>
</footer>
<script>
    const el = document.getElementById('my-tree');
    const TAB = '&nbsp;';
    const TAB_COUNT = 7;
    let tabBuffer = '';
    function printTree(nodes) {
        nodes.forEach((element) => {
            if (element.nodeType !== Node.TEXT_NODE) {
                el.innerHTML += `${tabBuffer}&lt;${element.nodeName}
${outpurAttributes(element)}&gt;<br>`;
                console.log(`${'\t'.repeat(tabBuffer.length / TAB.length /
TAB_COUNT)}<${element.nodeName} ${outpurAttributes(element)}>`);

```

```

        tabBuffer += TAB.repeat(TAB_COUNT);
    }
    else {
        if (element.textContent.trim()) {
            e1.innerHTML += `${tabBuffer}${element.textContent}<br/>`;
            console.log(`${'\t'.repeat(tabBuffer.length / TAB.length /
TAB_COUNT)}${element.textContent}`);
        }

    }

    if (element.hasChildNodes && element.id !== 'my-tree') {
        printTree(element.childNodes);
    }
    if (element.nodeType !== Node.TEXT_NODE) {
        tabBuffer = tabBuffer.substr(0, tabBuffer.length - TAB.length * TAB_COUNT);
        e1.innerHTML += `${tabBuffer}&lt;/${element.nodeName}&gt;<br>`;
        console.log(`${'\t'.repeat(tabBuffer.length / TAB.length /
TAB_COUNT)}</${element.nodeName}>`);
    }
    });
}

function outputAttributes(element) {
    let attributesStr = '';
    element.getAttributeNames().forEach(name => {
        attributesStr += `${name}=${element.getAttribute(name)} `;
    });
    return attributesStr;
}

</script>
</body>
</html>

```

Отладочный вывод в консоль:

```

<DIV class=row >
  <DIV class=col-4 form-group >
    <TABLE class=table >
      <TBODY >
        <TR >
          <TH scope=row >
          </TH>
          <TD >
            Логин:
          </TD>
          <TD >
            <INPUT type=text name=LoginField >
          </INPUT>
          </TD>
        </TR>
        <TR >
          <TH scope=row >
          </TH>
          <TD >
            Пароль:
          </TD>
          <TD >
            <INPUT type=password name=PasswordField >
          </INPUT>
          </TD>
        </TR>
        <TR >
          <TH scope=row >
          </TH>
          <TD >
            Пол:
          </TD>
          <TD >
            <INPUT type=radio name=Sex value=M >
          </INPUT>
            М
            <INPUT type=radio name=Sex value=Ж >
          </INPUT>
            Ж
          <BR >
          </BR>
          </TD>
        </TR>
        <TR >
          <TH scope=row >
          </TH>
          <TD colspan=2 align=right >
            <INPUT type=submit name=SubmitBtn >
          </INPUT>
          </TD>
        </TR>
      </TBODY>
    </TABLE>
  </DIV>

```

1.Фрагмент вывода консоли

```

        @Test public void testGetAllPosts() { HttpResponse httpResponse =
ConnectionUtils.getUtils().sendGETRequest(Hosts.POSTS.getAll()); List
        <POST >
            posts = Arrays.asList(JSONUtils.getObjectFromHttpResponse(httpResponse, Post[].class));
posts.sort(Comparator.comparingInt(Post::getId)); Assert.assertEquals(posts, posts, "List should be sort");
Assert.assertEquals(httpResponse.getStatusLine().getStatusCode(), HttpStatus.SC_OK); }
        </POST >
    </PRE>
</CODE>
<SAMP >
    Test complete successfully
</SAMP>
<BR >
</BR>
</DIV>
</DIV>
<DIV >
    <INPUT type=button value=Показать дерево onclick=el.innerText = "";console.clear();printTree(document.documentElement.childNodes) >
    </INPUT>
    <BR >
    </BR>
    <P id=my-tree >
    </P>
</DIV>
</MAIN>
<FOOTER class=my-primary >
    <DIV class=container >
        <P class=footer-text >
            Footer
        </P>
    </DIV>
</FOOTER>
<SCRIPT >
    const el = document.getElementById("my-tree"); const TAB = ' '; const TAB_COUNT = 7; let tabBuffer = ""; function printTree(nodes) {
nodes.forEach((element) => { if (element.nodeType != Node.TEXT_NODE) { el.innerHTML += `${tabBuffer}<${element.nodeName}
${outputAttributes(element)}>
`; console.log(`${'\t'.repeat(tabBuffer.length / TAB.length / TAB_COUNT)}<${element.nodeName} ${outputAttributes(element)}>`; tabBuffer +=
TAB.repeat(TAB_COUNT); } else { if (element.textContent.trim()) { el.innerHTML += `${tabBuffer}${element.textContent}
`; console.log(`${'\t'.repeat(tabBuffer.length / TAB.length / TAB_COUNT)}${element.textContent}`); } } if (element.childNodes && element.id != 'my-tree') {
printTree(element.childNodes); } if (element.nodeType != Node.TEXT_NODE) { tabBuffer = tabBuffer.substr(0, tabBuffer.length - TAB.length * TAB_COUNT);
el.innerHTML += `${tabBuffer}</${element.nodeName}>
`; console.log(`${'\t'.repeat(tabBuffer.length / TAB.length / TAB_COUNT)}); }); } function outputAttributes(element) { let attributesStr = "";
element.getAttributeNames().forEach(name => { attributesStr += `${name}=${element.getAttribute(name)} `; }); return attributesStr; }
    </SCRIPT>
</BODY>

```

2. Пример вывода дерева

Вывод: научился работать с деревом сайта, консолью и рекурсивными ф-циями.