

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

по лабораторной работе № 3

Дисциплина: Языки Интернет-программирования

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.

Часть 1.

Пункт 1. Сформировать страницу с произвольным кодом разметки, но обеспечить уровень вложенности внутри элемента `<body>` не менее 3.

Код страницы (файл **index.html**):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    table {
      border-collapse: collapse;
    }
    td {
      padding: 3px;
      border: 1px solid grey;
    }
  </style>
  <script src="script.js"></script>
  <title>Document</title>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>caption1</th>
        <th>caption2</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>item1</td>
        <td>item2</td>
      </tr>
      <tr>
        <td>item1</td>
        <td>item2</td>
      </tr>
      <tr>
        <td>item1</td>
        <td>item2</td>
      </tr>
    </tbody>
  </table>
</body>
```

```

</table>
<button id="start">Обход</button>
<div id="result">Результат обхода:</div>
</body>
</html>

```

Программа обхода (файл **script.js**):

```

window.onload = function() {
    var text = "";
    var c = 0;

    function countElements(node) {
        var elementsNum = 0;

        if (node.nodeType == 1) {
            elementsNum++;
            console.log(node.nodeName+', increment the counter +1');
            var a = '';
            for (j = 0; j < c; j++) {
                a = a + '&nbsp;'.repeat(4);
            }
            text = text + a + node.nodeName + "<br>";
        } else {
            console.log(node.nodeName+', not an element')
        }

        var children = node.childNodes;

        for (var i=0; i < children.length; i++) {
            c = c + 1;
            elementsNum += countElements(children[i]);
            c = c - 1;
        }
        return elementsNum;
    }

    function resultInDiv() {
        result = document.getElementById('result');
        var allElementsNum = countElements(document);
        result.innerHTML = "Результат обхода:<br>" + text + 'Количество подсчитанных элементов: ' + allElementsNum + '.';
    }

    var btn = document.getElementById('start');
    btn.addEventListener('mousedown', resultInDiv);
};

```

Пример обхода дерева элементов страницы (см. рис. 1):

caption1 caption2

item1	item2
item1	item2
item1	item2
Обход	

Количество подсчитанных элементов: 25.



Рис. 1 – пример обхода дерева элементов страницы.

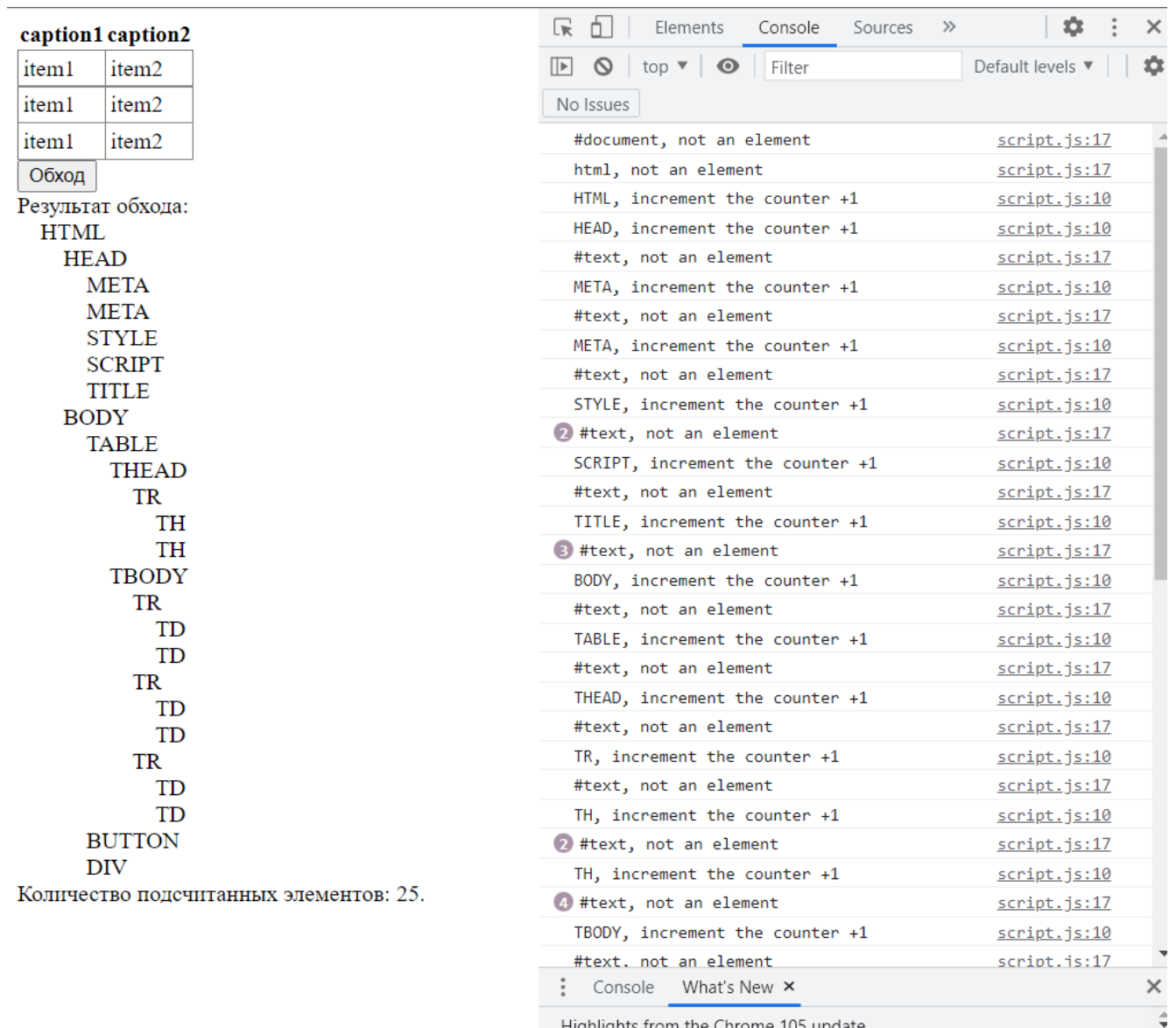


Рис. 2 – пример обхода дерева элементов страницы (с выводом на страницу).

Пункт 2. Добавить внутри элемента `<body>` секцию `<div>`, предназначенную для вывода результата обхода дерева элементов страницы.

Добавлено в коде выше.

Пункт 3. Выбрать способ активации рекурсивной программы обхода дерева элементов, реализовать и подключить эту программу.

Выполнено в коде выше.

Пункт 4. При проходе по узлам разметки обеспечить отладочный вывод в консоль. Привести в отчете содержимое консоли.

Содержимое консоли:

```

script.js:8 #document, not an element
script.js:8 html, not an element

```

```
script.js:6 HTML, increment the counter +1
script.js:6 HEAD, increment the counter +1
script.js:8 #text, not an element
script.js:6 META, increment the counter +1
script.js:8 #text, not an element
script.js:6 META, increment the counter +1
script.js:8 #text, not an element
script.js:6 STYLE, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 SCRIPT, increment the counter +1
script.js:8 #text, not an element
script.js:6 TITLE, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 BODY, increment the counter +1
script.js:8 #text, not an element
script.js:6 TABLE, increment the counter +1
script.js:8 #text, not an element
script.js:6 THEAD, increment the counter +1
script.js:8 #text, not an element
script.js:6 TR, increment the counter +1
script.js:6 TH, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 TH, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 TBODY, increment the counter +1
script.js:8 #text, not an element
script.js:6 TR, increment the counter +1
script.js:8 #text, not an element
script.js:6 TD, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 TD, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 TR, increment the counter +1
script.js:8 #text, not an element
script.js:6 TD, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 TD, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 TR, increment the counter +1
script.js:8 #text, not an element
script.js:6 TD, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
```

```
script.js:6 TD, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 BUTTON, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
script.js:6 DIV, increment the counter +1
script.js:8 #text, not an element
script.js:8 #text, not an element
```

Пункт 5. Реализовать вывод на странице.

Реализовано в коде выше.