



**«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 № 1 (вариант 26)**

**Название лабораторной работы:**

JavaScript, Модель DOM

**Дисциплина:** Языки Интернет-программирования

Студент гр. **ИУ6-33Б**

  
(Подпись, дата)

**Подлесный Р.И.**  
(ФИО)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

**Малахов Д.В.**  
(ФИО)

## Задание:

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:

Верстка страницы:

Файл *lab3.html*

```
<!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>Lab3</title>
  <link rel="stylesheet" href="bootstrap.min.css">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <div class="page-title-block"><h2>Page title</h2></div>
    <div class="cards-block">
      <div class="card">
        <div class="card-top"></div>
        <div class="card-body">
          <h4 class="card-title">Готовая карточка</h4>
          <p class="card-text">Lorem ipsum dolor sit amet consectetur adipisicing elit. Nesciunt,
dignissimos at. Labore porro ipsum temporibus? Ipsam placeat ipsum incidunt sed!</p>
          <a href="#" class="btn btn-primary">Ссылка никуда</a>
        </div>
      </div>
      <div class="card">
        <div class="card-top"></div>
        <div class="card-body">
          <h4 class="card-title">Готовая карточка 2</h4>
          <p class="card-text">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aspernatur,
corrupti. Veritatis recusandae esse cumque laboriosam modi possimus architecto, adipisci culpa.</p>
          <a href="#" class="btn btn-primary">Ссылка никуда</a>
        </div>
      </div>
    </div>
    <button class="btn-secondary" type="submit" onclick="showDom()">Показать DOM-элементы</button>
    <h4 id="output-header">Результат обхода дерева элементов страницы:</h4>
    <div id="dom-output-block"></div>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

## Файл *style.css*:

```
html,
body {
    min-height: auto;
    height: 100%;
    width: 100%;
}

.container {
    height: 100%;
    margin: 0 auto;
    max-width: 1000px;
    border-left-style: solid;
    border-right-style: solid;
    border-color: lightgray;
    border-width: 2px;
    height: auto;
    min-height: 100%;
    padding-bottom: 20px;
}

.page-title-block {
    text-align: center;
    padding: 10px 0 5px 0;
}

.cards-block {
    display: flex;
    width: 100%;
    margin-top: 10px;
    margin-bottom: 15px;
}

.card {
    width: 50%;
    margin: 0 2%;
    text-align: center;
    border-radius: 10px;
}

.card p {
    text-align: justify;
}

.card-top {
    width: 100%;
    height: 50px;
    background-color: rgb(78, 23, 206);
    border-top-left-radius: 10px;
    border-top-right-radius: 10px;
}

.btn-secondary {
    padding: 6px 12px 6px 12px;
    border-radius: 5px;
    display: flex;
    margin: 0 auto;
    margin-bottom: 10px;
}

#dom-output-block p {
    width: 100%;
    margin: 0px;
    padding: 0px;
}

#output-header {
    display: none;
}
```

## Файл *script.js*:

```
function showDom() {
  document.getElementById('output-header').style.display = 'block';
  outputBlock = document.getElementById("dom-output-block");
  outputBlock.innerHTML = ''; // erasing in case of repeated clicking the button
  console.log('Showing DOM structure:');
  showDomElem();
}

function showDomElem(element = document.firstChild, shift = 0) {
  if (element.parentElement != outputBlock) { // if we include the output block, we get into an infinite loop
    let newline = document.createElement('p');
    let str = '&nbsp;'.repeat(shift*4) + '&lt;' + element.tagName + '&gt;';
    outputBlock.append(newline);
    newline.innerHTML = str;
    console.log('\t'.repeat(shift) + '<' + element.tagName + '>');
    if (element.firstChild) { // if elem has childs:
      shift++;
      showDomElem (element.firstChild, shift);
      return;
    } else if (element.nextElementSibling) { // if elem has no childs but has next siblings:
      showDomElem(element.nextElementSibling, shift);
      return;
    } else { //if elem has no childs and no next siblings
      shift--;
      element = element.parentElement; // moving upwards

      let newline = document.createElement('p');
      let str = '&nbsp;'.repeat(shift*4) + '&lt;' + '/' + element.tagName + '&gt;';
      outputBlock.append(newline);
      newline.innerHTML = str;

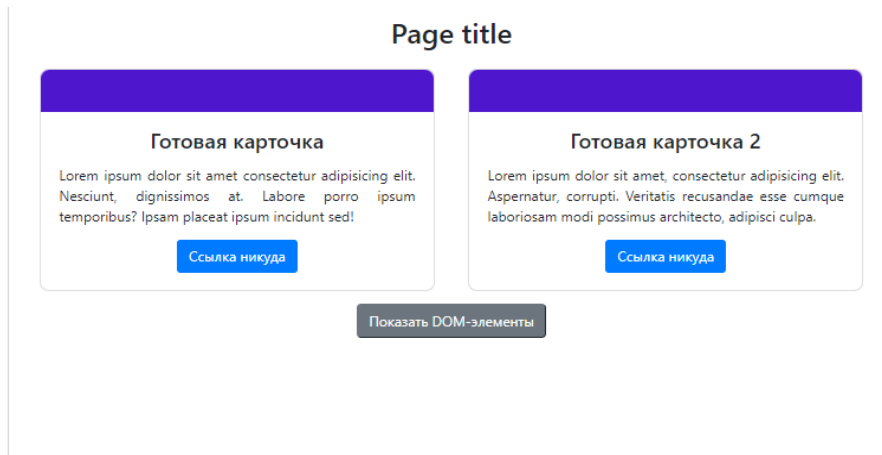
      console.log('\t'.repeat(shift) + '</' + element.tagName + '>'); // closing tag
      if (element.nextElementSibling) { // looking for next siblings
        showDomElem(element.nextElementSibling, shift);
        return;
      }
    }
    else {
      while (element.parentElement) {
        shift--;
        element = element.parentElement;

        let newline = document.createElement('p');
        let str = '&nbsp;'.repeat(shift*4) + '&lt;' + '/' + element.tagName + '&gt;';
        outputBlock.append(newline);
        newline.innerHTML = str;

        console.log('\t'.repeat(shift) + '</' + element.tagName + '>');
        if (element.nextElementSibling){
          showDomElem(element.nextElementSibling, shift);
          return;
        }
      }
    }
  }
}

} else { // if we've come to the output block, terminating the loop and closing all the parent tags:
  while (element.parentElement) {
    shift--;
    element = element.parentElement;
    let newline = document.createElement('p');
    let str = '&nbsp;'.repeat(shift*4) + '&lt;' + '/' + element.tagName + '&gt;';
    outputBlock.append(newline);
    newline.innerHTML = str;
    console.log('\t'.repeat(shift) + '</' + element.tagName + '>');
  }
}
```

Результат в браузере (до нажатия кнопки):



Результат в браузере (до нажатия кнопки):

Page title

### Готовая карточка

Lorem ipsum dolor sit amet consectetur adipisicing elit. Nesciunt, dignissimos at. Labore porro ipsum temporibus? Ipsam placeat ipsum incidunt sed!

[Ссылка никуда](#)

### Готовая карточка 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Aspernatur, corrupti. Veritatis recusandae esse cumque laboriosam modi possimus architecto, adipisci culpa.

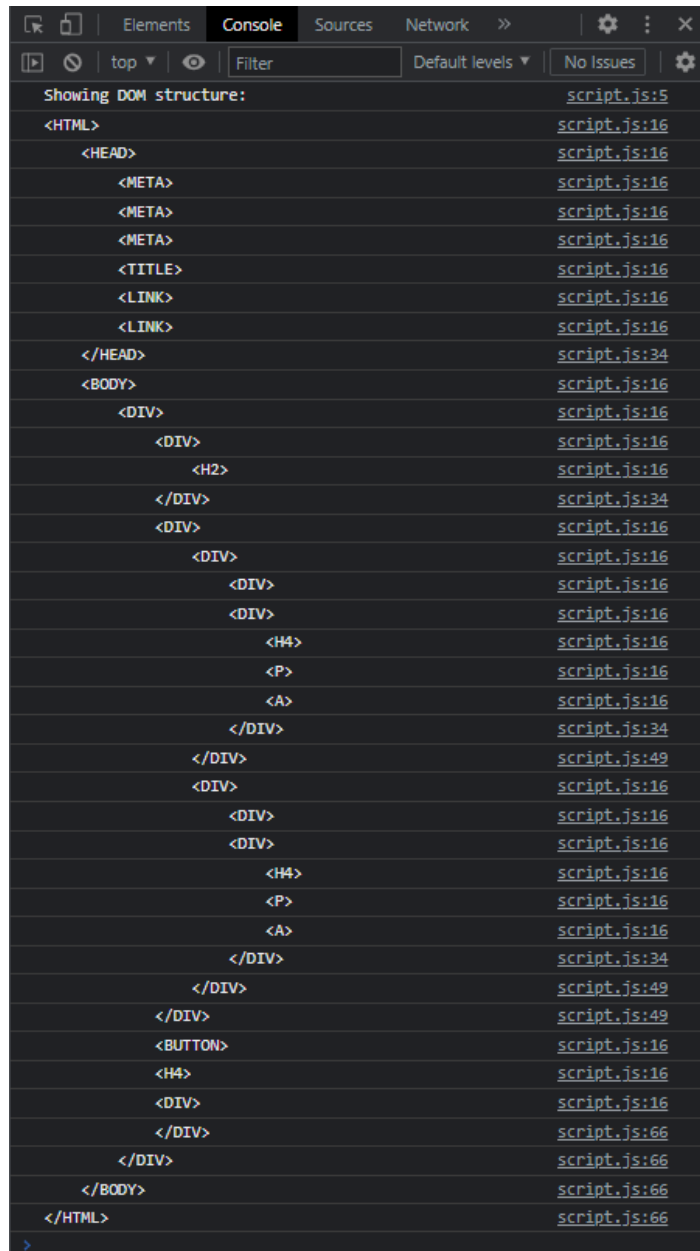
[Ссылка никуда](#)

Показать DOM-элементы

**Результат обхода дерева элементов страницы:**

```
<HTML>
<HEAD>
  <META>
  <META>
  <META>
  <TITLE>
  <LINK>
  <LINK>
</HEAD>
<BODY>
  <DIV>
    <DIV>
      <H2>
    </DIV>
    <DIV>
      <DIV>
        <DIV>
          <DIV>
            <H4>
            <P>
            <A>
          </DIV>
        </DIV>
      </DIV>
      <DIV>
        <DIV>
          <DIV>
            <H4>
            <P>
            <A>
          </DIV>
        </DIV>
      </DIV>
    </DIV>
    <BUTTON>
      <H4>
      <DIV>
    </DIV>
  </DIV>
</BODY>
</HTML>
```

Вывод в консоль:



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the DOM structure of a web page, starting with 'Showing DOM structure:' and listing various HTML elements and their corresponding script locations. The elements are listed in a tree-like structure, with indentation indicating their hierarchy. The elements include <HTML>, <HEAD>, <META>, <TITLE>, <LINK>, </HEAD>, <BODY>, <DIV>, <H2>, </DIV>, <DIV>, <DIV>, <H4>, <P>, <A>, </DIV>, </DIV>, <DIV>, <DIV>, <H4>, <P>, <A>, </DIV>, </DIV>, <DIV>, <DIV>, <H4>, <P>, <A>, </DIV>, </DIV>, <BUTTON>, <H4>, <DIV>, </DIV>, </BODY>, and </HTML>. Each element is followed by a link to the script file and line number, such as 'script.js:5' or 'script.js:16'.

```
Showing DOM structure: script.js:5
<HTML> script.js:16
  <HEAD> script.js:16
    <META> script.js:16
    <META> script.js:16
    <META> script.js:16
    <TITLE> script.js:16
    <LINK> script.js:16
    <LINK> script.js:16
  </HEAD> script.js:34
  <BODY> script.js:16
    <DIV> script.js:16
      <DIV> script.js:16
        <H2> script.js:16
      </DIV> script.js:34
      <DIV> script.js:16
        <DIV> script.js:16
          <DIV> script.js:16
            <H4> script.js:16
            <P> script.js:16
            <A> script.js:16
          </DIV> script.js:34
        </DIV> script.js:49
        <DIV> script.js:16
          <DIV> script.js:16
            <DIV> script.js:16
              <H4> script.js:16
              <P> script.js:16
              <A> script.js:16
            </DIV> script.js:34
          </DIV> script.js:49
        </DIV> script.js:49
      <BUTTON> script.js:16
      <H4> script.js:16
      <DIV> script.js:16
      </DIV> script.js:66
    </DIV> script.js:66
  </BODY> script.js:66
</HTML> script.js:66
```

Файлы лабораторной работы доступны по ссылке:

[https://github.com/Ratiboredom/web\\_languages\\_bmstu/tree/main/Labs/Lab3](https://github.com/Ratiboredom/web_languages_bmstu/tree/main/Labs/Lab3)