



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 № 4

**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

Разработать страницу для оформления заказа товаров.

- Примечание: в качестве усложненного варианта задания допустимо использовать библиотеки jQuery и Dojo.

```
<html lang="en">

<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="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU"
crossorigin="anonymous">
  <title>Document</title>
  <style>
    .row:hover>.price {
      background: rgb(241, 243, 118);
    }

    .addToCart {
      background-image: url('https://cdn.icon-
icons.com/icons2/3136/PNG/32/add_to_cart_trolley_shopping_icon_192426.png');
      background-color: white;
      background-repeat: no-repeat;
      background-position: left;
      border: none;
      margin-left: 10px;
      height: 32px;
      width: 32px;
    }

    #header {
      background-color: inherit;
    }
  </style>

```

```

        .table {
            display: table;
        }

        .row {
            display: table-row;
            white-space: nowrap;
            border-width: thin;
            border-style: solid;
            width: 100%;
        }

        .row *:not(.buy):not(.buy *) {
            display: table-cell;
            border-style: solid;
            border-width: thin;
            text-align: center;
            width: 30%;
        }

        .buy #BuyCount {
            width: 60%;
            border-collapse: collapse;
        }
    </style>
</head>

<body>
    <table align="center">
        <tr>
            <td><label for="ProductName">Наименование товара:</label></td>
            <td><input type="text" name="ProductName" id="ProductName"></td>
        </tr>
        <tr>
            <td><label for="ProductPrice">Цена:</label></td>
            <td><input type="number" name="ProductPrice" id="ProductPrice"></td>
        </tr>
        <tr>
            <td><label for="ProductCount">Количество</label></td>
            <td><input type="number" name="ProductCount" id="ProductCount"></td>
        </tr>
        <tr>
            <td colspan="2" align="right"><input type="button" value="Добавить
товар" name="Add"
                onclick="addRowToShop()">
            </td>
        </tr>
    </table>

```

```

<div class="table" id="products">
  <div style="display: table-caption; text-align:center;">Товары</div>
  <div class="row" id="header">
    <div>Наименование</div>
    <div>Цена</div>
    <div>Кол-во</div>
  </div>
</div>
<div class="table" id="cart" style="width:30%;margin-left: auto; margin-
right: 0px;">
  <div style="display: table-caption; text-align:center;">Корзина</div>
  <div class="row" id="header">
    <div>Продукт</div>
    <div>Цена</div>
    <div>Кол-во</div>
  </div>
</div>
<input type="button" value="Рассчитать" name='fullPrice'
onclick="printFullPrice()" style='float: right;'>
<label style='float: right; padding-right: 25px; color: rgb(223, 75, 49);'
for="fullPrice" id="result"></label>
<script src="Laba4.js"> </script>
</body>
</html>

```

Laba4.js:

```

const productTable = document.getElementById('products');
const cartTable = document.getElementById('cart');
const inputProductName = document.getElementById('ProductName');
const inputProductPrice = document.getElementById('ProductPrice');
const inputProductCount = document.getElementById('ProductCount');
const productsInShop = new Map();
const productsInACart = new Map();

function createRowWithBtn(productsMap, productName, productPrice, productCount) {
  let row = createRowWithoutBtn(productsMap, productName, productPrice,
productCount);
  row.append(createAddToCartBtn());
  return row;
}

function createRowWithoutBtn(productsMap, productName, productPrice,
productCount) {
  productsMap.set(productName, productCount);
  let row = document.createElement('div');
  row.setAttribute('class', 'row');
  let name = document.createElement('div');
  name.innerText = productName;
  name.setAttribute('class', 'name');

```

```

    let price = document.createElement('div');
    price.innerText = productPrice;
    price.setAttribute('class', 'price');
    let count = document.createElement('div');
    count.setAttribute('class', 'count');
    count.innerText = productCount;
    row.append(name, price, count);
    return row;
}

function createAddToCartBtn() {
    let buyDiv = document.createElement('div');
    buyDiv.setAttribute('class', 'buy');
    let inputBuyCount = document.createElement('input');
    inputBuyCount.setAttribute('type', 'number');
    inputBuyCount.setAttribute('name', 'BuyCount');
    inputBuyCount.setAttribute('id', 'BuyCount');
    inputBuyCount.setAttribute('value', 1);
    inputBuyCount.setAttribute('min', 1);
    let addToCart = document.createElement('button');
    addToCart.setAttribute('class', 'addToCart');
    addToCart.setAttribute('onclick',
'addToCart(this.parentElement.parentElement);document.getElementById(\'result\').
innerText=\'\');
    buyDiv.append(inputBuyCount, addToCart);
    return buyDiv;
}

function getTableRows(table) {
    return Array.from(table.children).filter((elem) =>
elem.getAttribute('class')?.includes('row') && !elem?.id.includes('header'));
}

function getDataAndClear(element) {
    let data = element.value;
    clear(element);
    return data;
}

function clear(elem) {
    elem.value = '';
}

function printFullPrice() {
    let sum = 0;
    getTableRows(cartTable).forEach((row) => {
        sum += +row.querySelector('.price').innerText *
row.querySelector('.count').innerText;
    });
    document.getElementById('result').innerText = `Сумма покупок ${sum}`;
}

```

```

function addToCart(row) {
    let rowCount = +row.querySelector('#BuyCount').value;
    if (rowCount < 1) {
        alert('Введите кол-во больше 0');
        row.querySelector('#BuyCount').value = 1;
        return;
    }
    let rowName = row.querySelector('.name').innerText;
    if (productsInShop.get(rowName) - rowCount < 0) {
        alert('На складе не хватает товара');
        return;
    }
    if (productsInACart.has(row.querySelector('.name').innerText)) {
        let countInTable = getTableRows(cartTable)
            .find((elem) => elem.querySelector('.name').innerText == rowName)
            .querySelector('.count');
        countInTable.innerText = +countInTable.innerText + rowCount;
        productsInACart.set(rowName, productsInACart.get(rowName) + rowCount);
    } else {
        cartTable.append(createRowWithoutBtn(productsInACart, rowName,
row.querySelector('.price').innerText, rowCount));
    }
    getTableRows(productTable)
        .find((tableRow) => tableRow == row)
        .querySelector('.count').innerText -= rowCount;
    productsInShop.set(rowName, productsInShop.get(rowName) - rowCount);
}

function addRowToShop() {
    let name = getDataAndClear(inputProductName);
    let count = +getDataAndClear(inputProductCount);
    let price = +getDataAndClear(inputProductPrice);
    console.log(name, count, price);
    if (count < 0 || price < 0 || !count || !price || !name) {
        alert('Все поля должны быть заполнены, цена и количество должны быть
больше 0');
        return;
    }
    if (productsInShop.has(name)) {
        let countInRow = getTableRows(productTable)
            .find((elem) => elem.querySelector('.name').innerText == name)
            .querySelector('.count');
        countInRow.innerText = count + productsInShop.get(name);
        productsInShop.set(name, +countInRow.innerText);
    } else {
        productTable.append(createRowWithBtn(productsInShop, name, price,
count));
    }
}

```

```
//Начальный список продуктов  
productTable.append(createRowWithBtn(productsInShop, 'Колбаса', 200, 15));  
productTable.append(createRowWithBtn(productsInShop, 'Кабачок', 50, 10));  
productTable.append(createRowWithBtn(productsInShop, 'Молоко', 75, 5));
```

Вывод: Я научился взаимодействовать с дочерними элементами и создавать элементы через JS