# Министерство Образования Республики Беларусь

Учреждение образования

«Гомельский государственный технический университет

имени П.О. Сухого»

Кафедра «Информатика»

по курсу: «Разработка приложений для Интернет»

Лабораторная работа №7

# «Объектная модель документа (DOM)»

Выполнил: студент группы ИП-31

Заяц Н. С.

Допуск к защите: Проверил: преподаватель

Дата защиты: Самовенд Н.В.

Гомель 2021

**Цель работы**: изучить возможности взаимодействия JavaScript с элементами страницы и объектами DOM.

**Практическая часть:**

**Задание 1**

На основании задания 4 предыдущей лабораторной работы сформировать таблицу для отображения элементов массива. Каждое свойство объекта выводится в отдельную ячейку. Таблица должна формироваться динамически с использованием JavaScript. Предусмотреть форматирование таблицы

**Решение**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Task\_2</title>

</head>

<body>

<div id="content"></div>

<script>

function getTable(elementsForBody){

let table = document.createElement("table");

table.border = 1;

let thead = document.createElement("thead");

let tbody = document.createElement("tbody");

let headRow = document.createElement("tr");

["sensorModel", "sensorType", "title", "connectionInterface", "dPI", "price", "relizeDate"].forEach((elm) => {

let th=document.createElement("th");

th.appendChild(document.createTextNode(elm));

headRow.appendChild(th);

});

thead.appendChild(headRow);

table.appendChild(thead);

elementsForBody.forEach((elm) => {

let tr = document.createElement("tr");

for (let x in elm) {

let td = document.createElement("td");

td.appendChild(document.createTextNode(elm[x]))

tr.appendChild(td);

}

tbody.appendChild(tr);

})

table.appendChild(tbody);

return table;

}

</script>

<script>

let mouseOne = {

sensorModel: 'IP-1',

sensorType: 'Кружок',

title: 'Мышка для печати информации',

connectionInterface: 'PS/2',

dPI: 8913,

price: 123,

relizeDate: 'todat',

};

let mouseTwo = {

sensorModel: 'IP-2',

sensorType: 'Кружок',

title: 'Мышка для печати информации',

connectionInterface: 'PS/2',

dPI: 8913,

price: 576,

relizeDate: 'todat',

};

let mouseThree = {

sensorModel: 'IP-3',

sensorType: 'Кружок',

title: 'Мышка для печати информации',

connectionInterface: 'PS/2',

dPI: 8913,

price: 22,

relizeDate: 'todat',

};

let mouses = [mouseOne, mouseTwo, mouseThree];

window.onload=function() {

document.getElementById("content").appendChild(getTable(mouses));

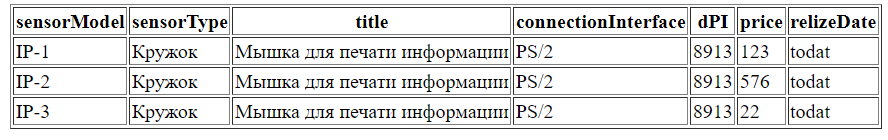
}

</script>

</body>

</html>

**Результат:**

****

**Задание 2**

Создать HTML-документ со списком ссылок. Ссылки на внешние источники (которые начинаются с http:// или https://) необходимо подчеркнуть пунктиром. Искать такие ссылки в списке и устанавливать им дополнительные стили необходимо с помощью JS.

**Решение**

<!DOCTYPE html>

<html lang="en">

<head>

<style>

body {

font-size: 36px;

background-color: rgb(250, 173, 148);

}

</style>

<meta charset="UTF-8">

<title>Task\_2</title>

</head>

<body>

<div id="content"></div>

<script>

function getLingks(){

let list = document.createElement("ul");

["index-html", "https://google.com", "https://www.gstu.by", "file:///G:/users/RPI/test.html", "/local/path", "https://www.tut.by/"].forEach((elm) => {

let liComponent=document.createElement("li");

let linkComponent = document.createElement('a');

linkComponent.setAttribute('href', elm);

linkComponent.innerText =elm;

liComponent.appendChild(linkComponent);

list.appendChild(liComponent);

});

return list;

}

</script>

<script>

window.onload=function() {

document.getElementById("content").appendChild(getLingks());

let links = document.links;

for (let i = 0; i < links.length; i++) {

if (links[i].href.startsWith('http')){

links[i].setAttribute('style', "text-decoration: none; border-bottom: 3px dashed;")

}}

}

</script>

</body>

</html>

Результат:



**Задание 3**

Создать HTML-документ с деревом вложенных директорий (см. рисунок ниже). Размер дерева и содержимое – произвольные. Каждый узел является элементом списка. При клике на элемент списка, он должен сворачиваться или разворачиваться. При наведении на элемент, шрифт должен становится жирным (с помощью свойства classList).

**Решение**

**<!DOCTYPE HTML>**

**<html>**

**<head>**

**<style>**

**body {**

**margin-left: 635px;**

**}**

**li {**

**list-style-type: none;**

**}**

**ul {**

**padding-left: 20px;**

**}**

**img {**

**width: 16px;**

**height: 16px;**

**float: left;**

**margin-right: 3px;**

**}**

**.hidden {**

**display: none;**

**}**

**.visible {**

**display: block;**

**}**

**.allocated:hover {**

**cursor: pointer;**

**font-weight: bold;**

**}**

**</style>**

**<meta charset="utf-8">**

**<script>**

**document.addEventListener("DOMContentLoaded", function (e) {**

**let labels = document.querySelectorAll(".allocated");**

**for (let i = 0; i < labels.length; i++) {**

**labels[i].addEventListener('click', function (e) {**

**this.parentElement.querySelector(".hidden").classList.toggle("visible");**

**})**

**}**

**})**

**</script>**

**</head>**

**<body>**

**<div>**

**<ul>**

**<li><img src="icons/этот\_компьютер.ico" alt="">**

**<p class="allocated"> Этот компьютер</p>**

**<ul class="hidden">**

**</li>**

**<li><img src="icons/локальный диск.ico" alt="">**

**<p class="allocated">Локальный диск (D:)</p>**

**<ul class="hidden">**

**<li><img src="icons/папка.ico" alt="">folder one</li>**

**<li> <img src="icons/папка.ico" alt="">folder two</li>**

**</ul>**

**</li>**

**<li><img src="icons/локальный диск.ico" alt="">**

**<p class="allocated">Локальный диск (E:)</p>**

**<ul class="hidden">**

**<li><img src="icons/папка.ico" alt=""> folder three</li>**

**<li><img src="icons/папка.ico" alt=""> folder five</li>**

**<li><img src="icons/папка.ico" alt="">**

**<p class="allocated">folder with inner files</p>**

**<ul class="hidden">**

**<li><img src="icons/папка.ico" alt=""> inner one </li>**

**<li><img src="icons/папка.ico" alt=""> inner two </li>**

**</ul>**

**</li>**

**</ul>**

**</li>**

**</ul>**

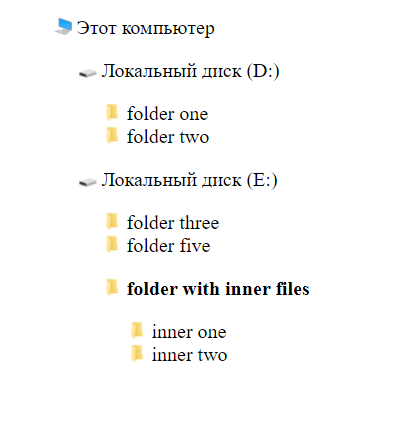
**</li>**

**</ul>**

**</div>**

**</body>**

**</html>**

****

**Задание 4**

Написать сценарий, который позволяет менять фоновое изображение документа выбором из таблицы цветов.

**Решение**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8" />**

**<meta name="viewport" content="width=device-width, initial-scale=1.0" />**

**<title>Nikita</title>**

**</head>**

**<body>**

**<script>**

**function createColorTable(tagName, array) {**

**let table = document.createElement('table');**

**table.style.border = '1px solid black';**

**table.style.background = 'white';**

**let numId = 0;**

**array.forEach(row => {**

**let tr = table.insertRow();**

**row.forEach(column => {**

**let td = tr.insertCell();**

**td.id = numId;**

**td.style.border = '1px solid black';**

**td.style.padding = '30px';**

**td.style.background = column;**

**td.onclick = onCellClick;**

**td.appendChild(document.createTextNode(''));**

**numId += 1;**

**});**

**});**

**let documents = Array.from(document.getElementsByTagName(tagName));**

**documents.forEach(x => x.appendChild(table));**

**}**

**let size = 5;**

**let colors = Array(size).fill()**

**.map(x => x = Array(size).fill()**

**.map(y => y = '#' + Math.random().toString(16).slice(-6)));**

**function onCellClick() {**

**let pos = this.id;**

**let x = Math.floor(pos / size);**

**let y = pos - x \* size;**

**document.body.style.background = colors[x][y];**

**console.log(this.innerHTML + ' ' + (x + 1) + ' ' + (y + 1));**

**}**

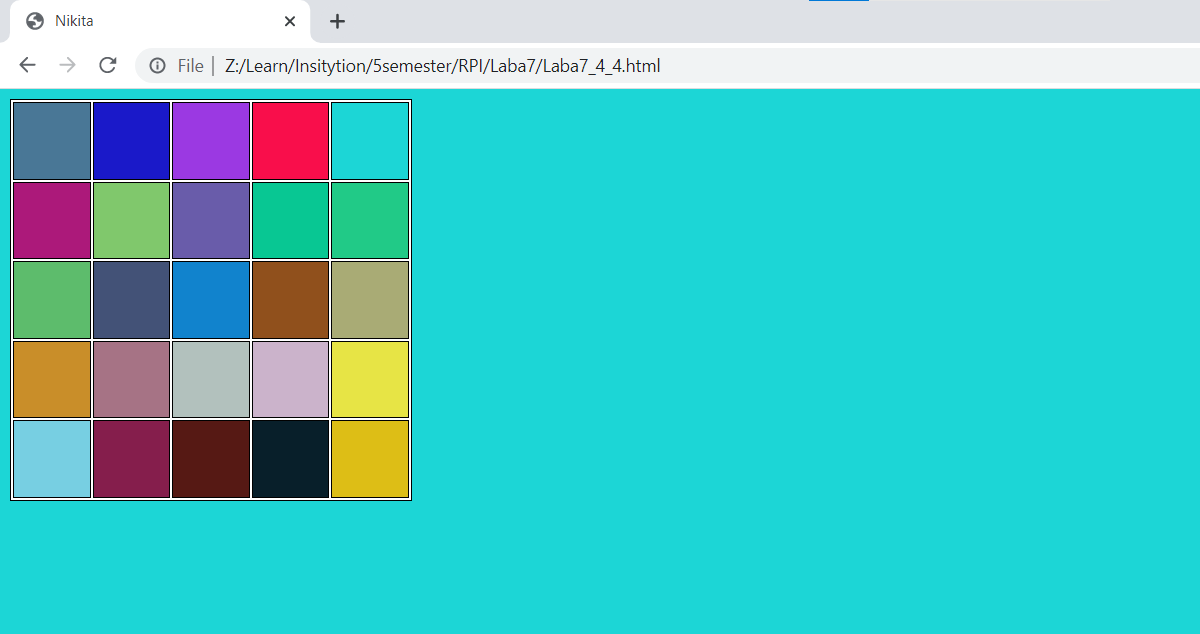
**createColorTable('body', colors);**

**</script>**

**</body>**

**</html>**

**Результат:**

****

**Задание 5**

Создать HTML-документ, в котором присутствуют три перекрывающихся (но не полностью) блока <div> с различным цветом фона и разными значениями z-индекса. Написать сценарий, в котором при клике мыши на блоке, соответствующий блок будет отображаться поверх остальных.

**Решение**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Zayats Mikita</title>

<style>

.red, .yellow, .blue{

width: 200px;

height: 200px;

border: none;

}

.red{

background: red;

top:20%;

left:30%;

position: fixed;

}

.yellow{

background: yellow;

top:30%;

left:40%;

position: fixed;

}

.blue{

background: blue;

top:40%;

left:50%;

position: fixed;

}

</style>

</head>

<body>

<div class="blue" id="blueBlock"></div>

<div class="yellow" id="yellowBlock"></div>

<div class="red" id="redBlock"></div>

<script>

let idsOfBlocks = ["redBlock", "yellowBlock", "blueBlock"];

document.addEventListener("click", function(event) {

let target = event.target;

for(let i in idsOfBlocks){

let tempTarget = document.getElementById(idsOfBlocks[i]);

tempTarget.style.zIndex = "0";

}

target.style.zIndex = "1";

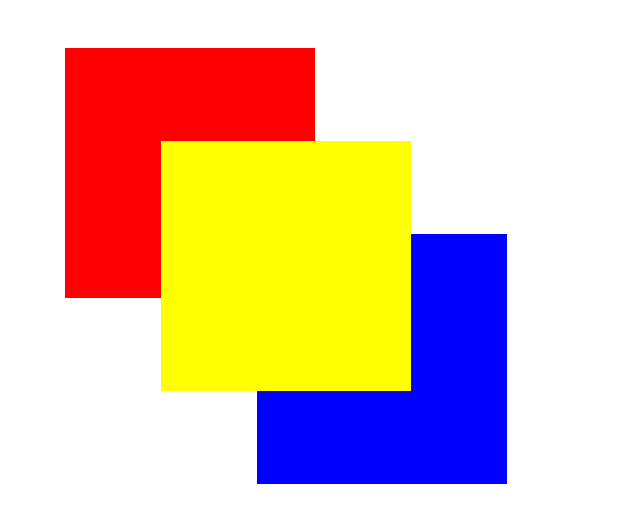
});

</script>

</body>

</html>

Результат:



**Задание 6**

Создать HTML-документ со списком книг (автор и название). При щелчке на элементе спика, цвет текста должен меняться на оранжевый. При повторном щелчке на элементе необходимо возвращать прежний цвет. Если при клике мышкой была нажата клавиша Ctrl, то элемент добавляется/удаляется из выделенных. Если при клике мышкой была нажата клавиша Shift, то к выделению добавляются все элементы в промежутке от предыдущего кликнутого до текущего

**Html файл**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Mikita Zayats</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <style>

        li {

            user-select: none;

            font-size: 24px;

        }

        .selected {

            color: orange;

        }

    </style>

</head>

<body>

    <ul id="books">

        <li>Автор 1, Название 1</li>

        <li>Автор 2, Название 2</li>

        <li>Автор 3, Название 3</li>

        <li>Автор 4, Название 4</li>

        <li>Автор 5, Название 5</li>

        <li>Автор 6, Название 6</li>

        <li>Автор 7, Название 7</li>

        <li>Автор 8, Название 8</li>

        <li>Автор 9, Название 9</li>

        <li>Автор 10, Название 10</li>

    </ul>

    <script>

        let bookClick = ((env) => {

            let previous = null;

            return (env) => {

                let setDefault = (e, ignoreElem) => {

                    let LIes = e.target.parentElement.querySelectorAll("li");

                    LIes.forEach(e => {

                        if (e != ignoreElem) e.classList.remove("selected");

                    });

                }

                let selectList = (e, startElement, foundedElement) => {

                    let LIes = e.target.parentElement.querySelectorAll("li");

                    let array = [];

                    let whoFirst = null;

                    let fillArray = false;

                    let brk = false;

                    LIes.forEach(liItem => {

                        if (brk) {

                            return;

                        }

                        if (fillArray) {

                            array.push(liItem);

                        }

                        if (whoFirst == null && (liItem == foundedElement || liItem == startElement)) {

                            array.push(liItem);

                            fillArray = true;

                            if (liItem == foundedElement) {

                                whoFirst = foundedElement;

                            } else if (liItem == startElement) {

                                whoFirst = startElement;

                            }

                        }

                        if ((whoFirst == startElement && liItem == foundedElement) ||

                            (whoFirst == foundedElement && liItem == startElement)) {

                            array.push(liItem);

                            brk = true;

                        }

                    });

                    array.forEach(liItem => liItem.classList.add("selected"));

                }

                let target = env.target;

                if (target.tagName.toLowerCase() != "li") {

                    return;

                }

                if (!env.shiftKey && !env.ctrlKey) {

                    setDefault(env, target);

                    previous = target;

                    target.classList.toggle("selected");

                }

                if (env.shiftKey) {

                    if (previous == null) {

                        previous = target;

                    } else {

                        selectList(env, previous, target);

                    }

                }

                if (env.ctrlKey) {

                    target.classList.toggle("selected");

                }

            }

        })();

        let books = document.querySelector("#books");

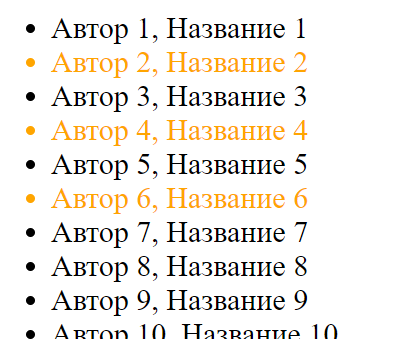
        books.addEventListener("click", bookClick)

    </script>

</body>

</html>

Результат:



**Вывод:** в ходе лабораторной работы былиизучены возможности взаимодействия JavaScript с элементами страницы и объектами DOM.