**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

по дисциплине «Разработка приложений для Интернет»

на тему: «Объектная модель документа (DOM)»

Выполнил: студент гр. ИП-32

Коледа А.С.

Принял: преподаватель

Процкая М.А.

Гомель 2021

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

**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.

Код файла lab\_7\_1.html:

<!DOCTYPE html>

<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 rel="shortcut icon" href="favicon.png" type="image/png">

<title>lab 7.1 Коледа Алексей ИП-32</title>

<style>

table {

border: 1px solid gray;

background-color: gray;

}

th {

height: 20px;

border: 1px solid gray;

background-color: rgb(223, 223, 223);

padding: 5px;

font-family: 'Courier New', Courier, monospace;

font-weight: bold;

}

td {

height: 20px;

border: 1px solid gray;

background-color: rgb(255, 255, 255);

padding: 5px;

font-family: 'Courier New', Courier, monospace;

font-weight: bold;

}

</style>

</head>

<body>

<script>

"use strict"

class HardDrive {

constructor(oType, oCapacity, oInterface, oSpeed) {

this.type = oType;

this.capacity = oCapacity;

this.interface = oInterface;

this.speed = oSpeed;

}

getType() {

return this.type;

}

setType(oType) {

this.type = oType;

}

getCapacity() {

return this.capacity;

}

setCapacity(oCapacity) {

this.capacity = oCapacity;

}

getInterface() {

return this.interface;

}

setInterface(oInterface) {

this.interface = oInterface;

}

getSpeed() {

return this.speed;

}

setSpeed(oSpeed) {

this.speed = oSpeed;

}

getInfo() {

return this.type + ' ' + this.capacity + ' ' + this.interface + ' ' + this.speed;

}

}

class Kingston extends HardDrive {

constructor(oType, oCapacity, oInterface, oSpeed, oReleaseDate, oPrice) {

super(oType, oCapacity, oInterface, oSpeed);

this.releaseDate = oReleaseDate;

this.price = oPrice;

}

getReleaseDate() {

return this.releaseDate;

}

setReleaseDate(oReleaseDate) {

this.releaseDate = oReleaseDate;

}

getPrice() {

return this.price;

}

setPrice(oPrice) {

this.price = oPrice;

}

getInfo() {

return this.type + ' ' + this.capacity + ' ' + this.interface + ' ' + this.speed + ' ' + this.releaseDate + ' ' + this.price;

}

}

let arr = [];

arr.push(new Kingston('SSD', 512, 'PCI', 0, 2007, 145));

arr.push(new Kingston('Flesh', 1000, 'PCI', 0, 2018, 998));

arr.push(new Kingston('SSD', 512, 'PCI', 0, 2010, 723));

arr.push(new Kingston('HDD', 256, 'M.2', 10000, 2008, 234));

generateTable(arr);

function generateTable(array) {

let prop\_name = ['тип', 'ёмкость', 'интерфейс', 'скорость', 'год релиза', 'цена'];

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

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

prop\_name.forEach(a => {

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

cell.innerText = a;

row.appendChild(cell);

});

table.appendChild(row);

array.forEach(element => {

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

let properties = Object.entries(element);

properties.forEach(key => {

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

cell.innerText = key[1];

row.appendChild(cell);

});

table.appendChild(row);

});

document.body.appendChild(table);

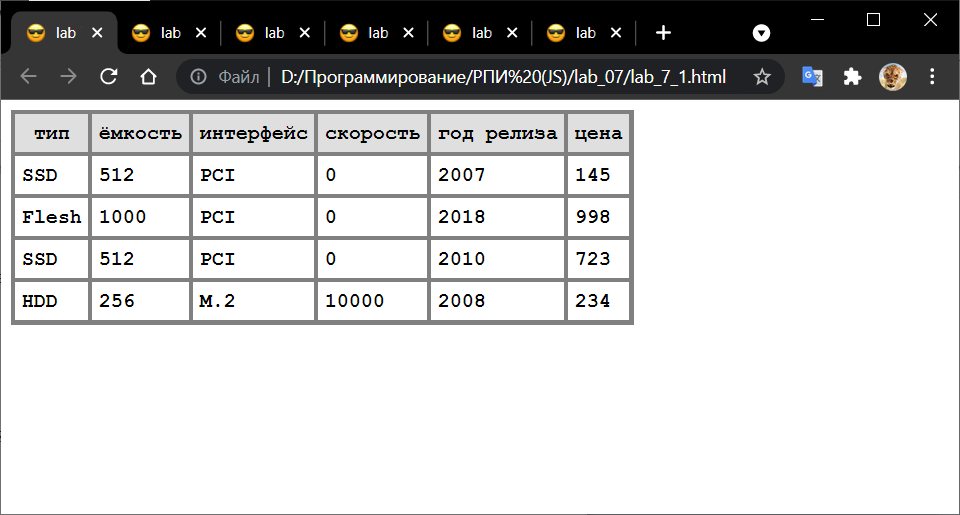
}

</script>

</body>

</html>

Результат выполнения 1-го задания:



**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.

Код файла lab\_7\_2.html:

<!DOCTYPE html>

<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 rel="shortcut icon" href="favicon.png" type="image/png">

<title>lab 7.2 Коледа Алексей ИП-32</title>

</head>

<body>

<ul>

<li><a href="https://example.com/">NASA</a></li>

<li><a href="D:/FAVICON.png">иконка</a></li>

<li><a href="https://example.com/">WikiPedia</a></li>

<li><a href="http://example.com/">Yandex</a></li>

<li><a href="http://example.com/">Google.com</a></li>

<li><a href="lab\_7\_1.html">предыдущая лаба</a></li>

<li><a href="http://example.com/">tut.by</a></li>

<li><a href="D:/[B.E.\_SHtern]\_Kovcheg\_47\_Libra(b-ok.org).pdf">книжка</a></li>

</ul>

<script>

"use strict"

let object = document.getElementsByTagName('a');

for (const key in object) {

if (Object.hasOwnProperty.call(object, key)) {

const element = object[key];

let begin = element.href.slice(0, 5);

if (begin == 'https' || begin == 'http:') {

element.style.textDecoration = 'underline';

element.style.textDecorationStyle = 'dotted';

element.style.color = 'black';

}

else if (begin == 'file:') {

element.style.textDecoration = 'none';

element.style.fontFamily = "'Courier New', Courier, monospace";

}

}

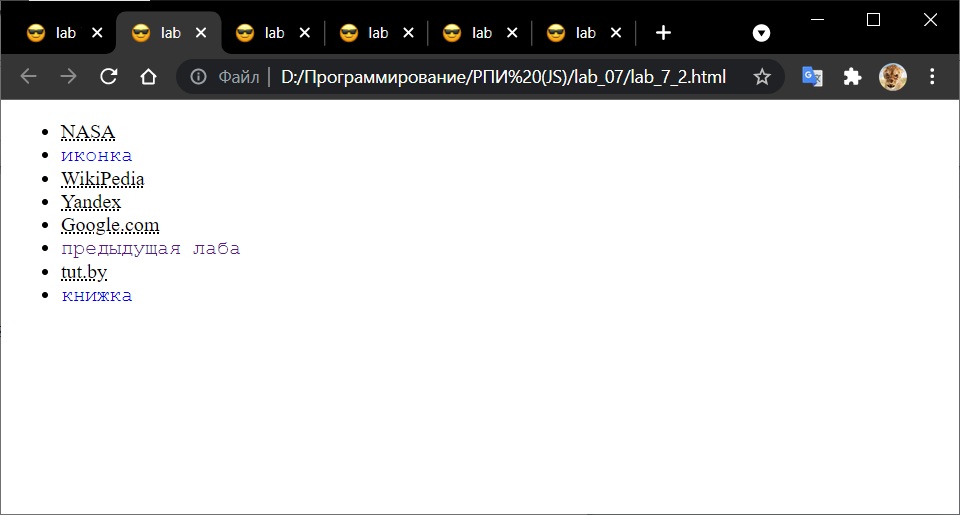
}

</script>

</body>

</html>

Результат выполнения 2-го задания:



**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).

Код файла lab\_7\_3.html:

<!DOCTYPE html>

<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 rel="shortcut icon" href="favicon.png" type="image/png">

<title>lab 7.3 Коледа Алексей ИП-32</title>

<style>

p {

margin: 0;

}

.my {

font-weight: bold;

}

</style>

</head>

<body>

<ul>

<li>

<p>C</p>

<ul>

<li>

<p>Program Files</p>

<ul>

<li>

<p>Microsoft</p>

</li>

<li>

<p>Notepad++</p>

</li>

<li>

<p>WinRAR</p>

</li>

</ul>

</li>

<li>

<p>Users</p>

<ul>

<li>

<p>Alex</p>

</li>

<li>

<p>Admin</p>

</li>

<li>

<p>Public</p>

</li>

</ul>

</li>

<li>

<p>Windows</p>

<ul>

<li>

<p>assembly</p>

</li>

<li>

<p>ru-RU</p>

</li>

<li>

<p>System32</p>

</li>

</ul>

</li>

</ul>

</li>

<li>

<p>D</p>

<ul>

<li>

<p>Загрузки</p>

<ul>

<li>

<p>Картинки</p>

</li>

<li>

<p>Фотографии</p>

</li>

<li>

<p>Лабы</p>

</li>

</ul>

</li>

<li>

<p>Программирование</p>

<ul>

<li>

<p>РПИ (JS)</p>

</li>

<li>

<p>АВС</p>

</li>

<li>

<p>ТиВПО</p>

</li>

</ul>

</li>

</ul>

</li>

</ul>

<script>

"use strict"

let pars = document.getElementsByTagName('p');

for (const key in pars) {

if (Object.hasOwnProperty.call(pars, key)) {

const element = pars[key];

element.addEventListener('mouseover', addToClass);

element.addEventListener('mouseout', removeFromClass);

element.addEventListener('click', toggle\_attribute);

}

}

function addToClass() {

event.target.classList.add('my');

}

function removeFromClass() {

event.target.classList.remove('my');

}

function toggle\_attribute() {

// Получаем родительский элемент li.

let parent = event.target.parentElement;

// Получаем всех потомков li, тоесть всех соседей this.

let childs = parent.childNodes;

for (const key in childs) {

if (Object.hasOwnProperty.call(childs, key)) {

const element = childs[key];

// Переключение атрибута для всех ul из массива потомков.

if (element.nodeName == 'UL') {

if (element.getAttribute('hidden') == 'true')

element.removeAttribute('hidden');

else

element.setAttribute('hidden', true);

}

}

}

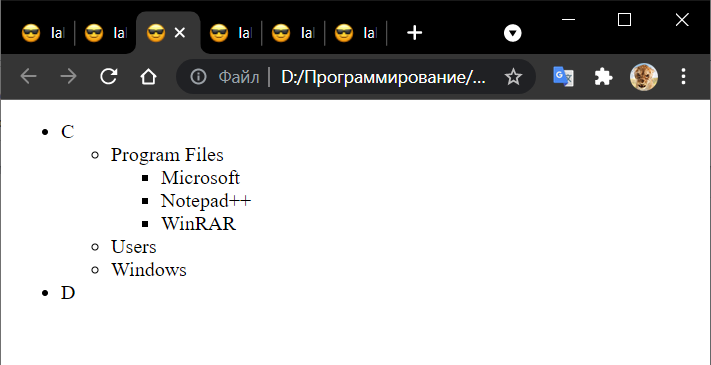
}

</script>

</body>

</html>

Результат выполнения 3-го задания:



**Задание 4:** Написать сценарий, который позволяет менять фоновое изображение документа выбором из таблицы цветов.

Код файла lab\_7\_4.html:

<!DOCTYPE html>

<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 rel="shortcut icon" href="favicon.png" type="image/png">

<title>lab 7.4 Коледа Алексей ИП-32</title>

<style>

table {

border: 2px solid black;

background-color: gray;

}

td {

height: 40px;

width: 40px;

border: 2px solid black;

}

td:hover {

border: 2px solid red;

}

</style>

</head>

<body>

<table>

<tr>

<td style="background-color: rgb(0, 0, 255);"></td>

<td style="background-color: rgb(0, 255, 255);"></td>

<td style="background-color: rgb(255, 255, 255);"></td>

<td style="background-color: rgb(255, 255, 0);"></td>

<td style="background-color: rgb(255, 0, 0);"></td>

</tr>

<tr>

<td style="background-color: rgb(0, 0, 192);"></td>

<td style="background-color: rgb(0, 192, 192);"></td>

<td style="background-color: rgb(192, 192, 192);"></td>

<td style="background-color: rgb(192, 192, 0);"></td>

<td style="background-color: rgb(192, 0, 0);"></td>

</tr>

<tr>

<td style="background-color: rgb(0, 0, 128);"></td>

<td style="background-color: rgb(0, 128, 128);"></td>

<td style="background-color: rgb(128, 128, 128);"></td>

<td style="background-color: rgb(128, 128, 0);"></td>

<td style="background-color: rgb(128, 0, 0);"></td>

</tr>

<tr>

<td style="background-color: rgb(0, 0, 64);"></td>

<td style="background-color: rgb(0, 64, 64);"></td>

<td style="background-color: rgb(64, 64, 64);"></td>

<td style="background-color: rgb(64, 64, 0);"></td>

<td style="background-color: rgb(64, 0, 0);"></td>

</tr>

</table>

<script>

"use strict"

let cells = document.getElementsByTagName('td');

for (const key in cells) {

if (Object.hasOwnProperty.call(cells, key)) {

const element = cells[key];

element.addEventListener('click', () => {

document.body.style.backgroundColor = event.currentTarget.style.backgroundColor;

});

}

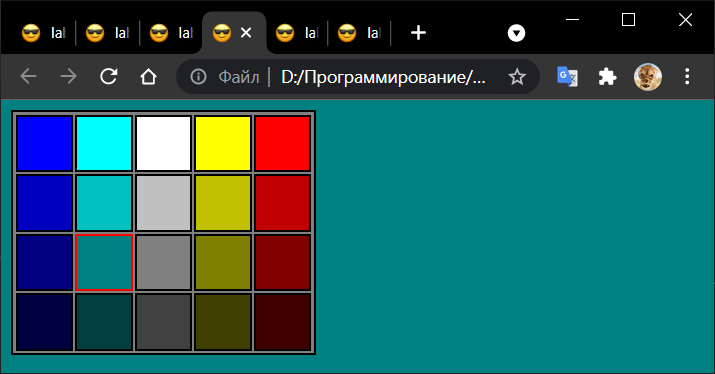
}

</script>

</body>

</html>

Результат выполнения 4-го задания:



**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.

Код файла lab\_7\_5.html:

<!DOCTYPE html>

<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 rel="shortcut icon" href="favicon.png" type="image/png">

<title>lab 7.5 Коледа Алексей ИП-32</title>

<style>

div {

width: 200px;

height: 200px;

position: absolute;

}

#first {

background-color: red;

z-index: 1;

}

#second {

background-color: yellow;

margin-top: 40px;

margin-left: 40px;

z-index: 2;

}

#third {

background-color: blue;

margin-top: 80px;

margin-left: 80px;

z-index: 3;

}

</style>

</head>

<body>

<div id='first'></div>

<div id='second'></div>

<div id='third'></div>

<script>

"use strict"

let divs = document.getElementsByTagName('div');

for (const unit of divs) {

console.log(unit);

unit.addEventListener('click', changeIndex);

}

let maxZ = 10;

function changeIndex() {

event.currentTarget.style.zIndex = maxZ;

maxZ++;

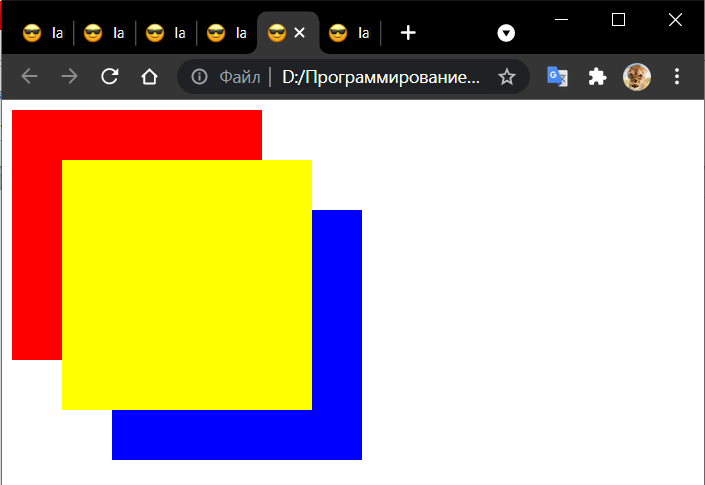
}

</script>

</body>

</html>

Результат выполнения 5-го задания:



**Задание 6:** Создать HTML-документ со списком книг (автор и название). При щелчке на элементе списка, 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.

Код файла lab\_7\_6.html:

<!DOCTYPE html>

<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 rel="shortcut icon" href="favicon.png" type="image/png">

<title>lab 7.6 Коледа Алексей ИП-32</title>

<style>

p {

margin: 5px;

font-weight: normal;

font-family: 'Courier New', Courier, monospace;

}

</style>

</head>

<body>

<ul>

<li>

<p>Сомерсет Моэм - Бремя страстей человеческих</p>

</li>

<li>

<p>Жуль Верн - 20 тысяч лье под водой</p>

</li>

<li>

<p>Толкин - Хоббит</p>

</li>

<li>

<p>Владимир Беляев - Прыжок в ничто</p>

</li>

<li>

<p>Стивен Хоккинг - Краткая история времени</p>

</li>

<li>

<p>Анджей Сапковский - Меч предназначения</p>

</li>

<li>

<p>Френк Герберт - Дюна</p>

</li>

<li>

<p>Lorem ipsum - dolor sit amet consectetur</p>

</li>

</ul>

<script>

"use strict"

let lis = document.getElementsByTagName('li');

let lastInd = -1;

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

const unit = lis[i];

unit.number = i;

unit.addEventListener('click', action);

}

let prevInd = 0;

function action() {

let unit = event.currentTarget.childNodes[1].style;

let unitInd = event.currentTarget.number;

if (event.shiftKey) {

if (lastInd > -1) {

let increment = 1;

let start = lastInd, finish = unitInd;

if (lastInd > unitInd) {

// swap

[start, finish] = [finish, start];

}

for (let i = start; i <= finish; i++) {

lis[i].childNodes[1].style.fontWeight = 'bold';

}

}

}

else if (event.ctrlKey) {

if (unit.fontWeight == 'bold')

unit.fontWeight = 'normal';

else {

unit.fontWeight = 'bold';

// запоминаем последнее выделение

lastInd = unitInd;

}

}

else {

if (unit.color == 'orange')

unit.color = 'black';

else {

unit.color = 'orange';

}

}

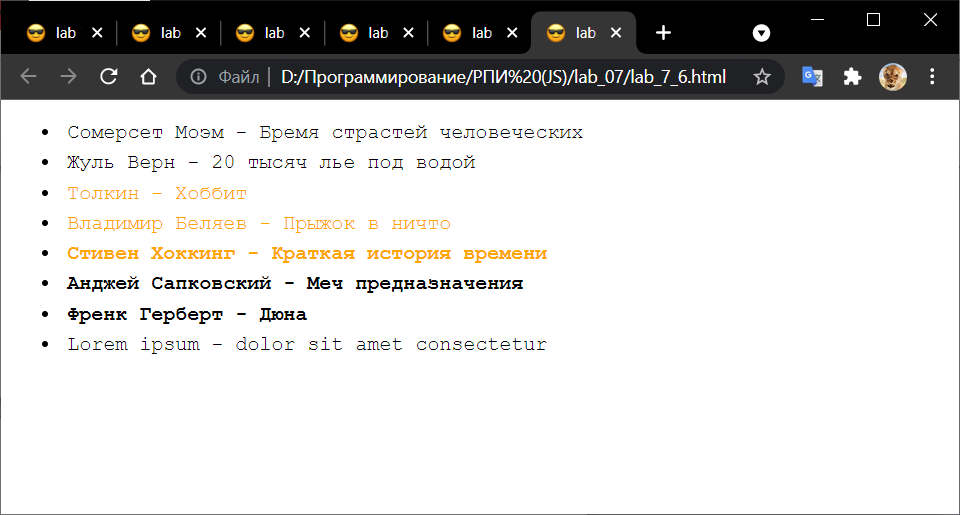
}

</script>

</body>

</html>

Результат выполнения 6-го задания:



**Выводы:** В процессе выполнения лабораторной работы изучил возможности взаимодействия JavaScript с элементами страницы и объектами DOM.