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
Бородина Н.Н.
Принял: преподаватель
Свинтицкий П.В.

Гомель 2023

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">
  <title>lab 7.1 Бородина Наталья ИП-32</title>
  <style>
    table {
      border: 1px solid grey;
      background-color: grey;
      font-family: 'Courier New', Courier, monospace;
      font-weight: bold;
    }

    th {
      height: 20px;
      border: 1px solid grey;
      background-color: rgb(223, 223, 223);
      padding: 5px;
    }

    td {
      height: 20px;
      border: 1px solid grey;
      background-color: rgb(255, 255, 255);
      padding: 5px;
    }
  </style>
</head>

<body>

<script>
   "use strict"

   class Processor {
      constructor(clockSpeed, cores, capacity, technicalProcess) {
         this._clockSpeed = clockSpeed;
         this._cores = cores;
         this._capacity = capacity;
         this._technicalProcess = technicalProcess;
      }
```

```
    set clockSpeed(value) {
        this._clockSpeed = value;
    }

    get clockSpeed() {
        return this._clockSpeed;
    }

    set cores(value) {
        this._cores = value;
    }

    get cores() {
        return this._cores;
    }

    set capacity(value) {
        this._capacity = value;
    }

    get capacity() {
        return this._capacity;
    }

    set technicalProcess(value) {
        this._technicalProcess = value;
    }

    get technicalProcess() {
        return this._technicalProcess;
    }

    getInfo() {
        let str = '<p>' + ' Тактовая частота: ' + this.clockSpeed + ' Гц' + '</p>';
        str += '<p>' + 'Количество ядер: ' + this.cores + '</p>';
        str += '<p>' + 'Разрядность: ' + this.capacity + '</p>';
        str += '<p>' + 'Техпроцесс: ' + this.technicalProcess + ' нм' + '</p>';
        return str;
    }
}

class Intel extends Processor {
    constructor(clockSpeed, cores, capacity, technicalProcess, release, price) {
        // Вызов родительского конструктора
        super(clockSpeed, cores, capacity, technicalProcess);

        this._release = release;
        this._price = price;
    }

    get release() {
        return this._release;
    }

    set release(value) {
        this._release = value;
    }
```

```
    get price() {
      return this._price;
    }

    set price(value) {
      this._price = value;
    }

    getInfo() {
      let str = '<p>' + ' Тактовая частота: ' + this.clockSpeed + ' Гц' + '</p>';
      str += '<p>' + 'Количество ядер: ' + this.cores + '</p>';
      str += '<p>' + 'Разрядность: ' + this.capacity + '</p>';
      str += '<p>' + 'Техпроцесс: ' + this.technicalProcess + ' нм' + '</p>';
      str += '<p>' + 'Выход: ' + this.release + ' год' + '</p>';
      str += '<p>' + 'Цена: ' + this.price + ' денег' + '</p>';
      return str;
    }
  }

  let arr = [];
  arr.push(new Intel(40000, 2, 32, 24, 2007, 1000));
  arr.push(new Intel(20000, 4, 32, 16, 2009, 1500));
  arr.push(new Intel(50000, 6, 32, 16, 2012, 23000));
  arr.push(new Intel(2_000_000_000, 16, 64, 5, 2018, 2000000));
  arr.push(new Intel(1_200_000_000, 24, 64, 4, 2021, 1800000, 2021));

  generateTable(arr);

  function generateTable(array) {
    let prop_name = ['частота', 'ядра', 'разрядность', 'техпроцесс', 'выход', '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:

частота	ядра	разрядность	техпроцесс	выход	цена
40000	2	32	24	2007	1000
20000	4	32	16	2009	1500
50000	6	32	16	2012	23000
2000000000	16	64	5	2018	2000000
1200000000	24	64	4	2021	1800000

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">
  <title>lab 7.2 Бородина Наталья ИП-32</title>
  <style>
     * {
        font-family: 'Courier New', Courier, monospace;
     }
  </style>
</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="https://yandex.com/">Yandex</a></li>
  <li><a href="https://www.google.com/">Google.com</a></li>
  <li><a href="lab_7_1.html">предыдущая лаба</a></li>
  <li><a href="../../labs/lab_1">лаба какая-то</a></li>
  <li><a href="file://">Файловая система (недоступна)</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 === 'http:' || begin === 'https:') {
          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:

- NASA
- иконка
- WikiPedia
- Yandex
- Google.com
- предыдущая лаба
- лаба какая-то
- Файловая система (недоступна)

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">
  <title>lab 7.3 Бородина Наталья ИП-32</title>
  <style>
    p {
      margin: 0;
      font-family: 'Courier New', Courier, monospace;
      user-select: none;
    }

    .my {
      font-weight: bold;
      cursor: pointer;
```

```
    }
  </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>Student</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 children = parent.childNodes;
```

```
      for (const key in children) {
        if (Object.hasOwnProperty.call(children, key)) {
          const element = children[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:

- C
  - Program Files
    - **Microsoft**
    - Notepad++
    - WinRAR
  - Users
    - Student
    - Admin
    - Public
  - Windows
- D

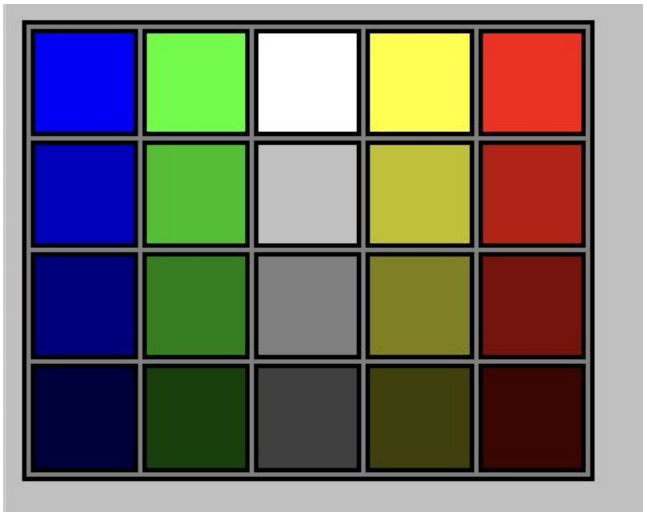
Задание 4: Написать сценарий, который позволяет менять фоновое изображение документа выбором из таблицы цветов.

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">
  <title>lab 7.4 Бородина Наталья ИП-32</title>
  <style>
    table {
      border: 2px solid black;
      background-color: grey;
    }

    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, 0);"></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, 0);"></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, 0);"></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, 0);"></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', (event) => {
        document.body.style.backgroundColor = event.currentTarget.style.backgroundColor;
      });
    }
  }
</script>
</body>

</html>
```

Результат выполнения 4-го задания:



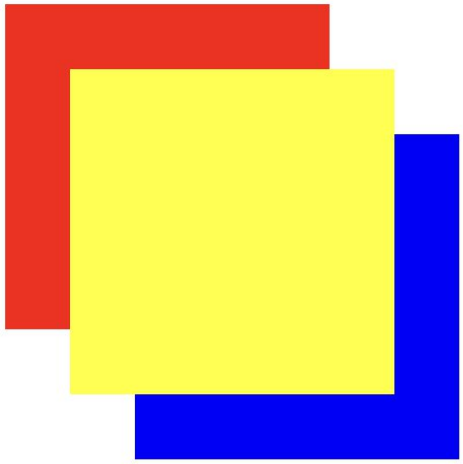
Задание 5: Создать HTML-документ, в котором присутствуют три перекрывающихся (но не полностью) блока с различным цветом фона и разными значениями z-индекса. Написать сценарий, в котором при клике мыши на блоке, соответствующий блок будет отображаться поверх остальных.

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">
  <title>lab 7.5 Бородина Наталья 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) {
      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">
  <title>lab 7.6 Бородина Наталья ИП-32</title>
  <style>
     p {
        margin: 5px;
        font-weight: normal;
        font-family: 'Courier New', Courier, monospace;
        user-select: none;
     }
  </style>
</head>

<body>
<ul id="root">
</ul>
<script>
   "use strict"

   let books = [
      {text: "Сомерсет Моэм - Бремя страстей человеческих", color: "black"},
      {text: "Жуль Верн - 20 тысяч лье под водой", color: "black"},
      {text: "Джон Рональд Руэл Толкин - Хоббит", color: "black"},
      {text: "Владимир Беляев - Прыжок в ничто", color: "black"},
      {text: "Стивен Хоккинг - Краткая история времени", color: "black"},
      {text: "Анджей Сапковский - Меч предназначения", color: "black"},
      {text: "Френк Герберт - Дюна", color: "black"},
      {text: "Lorem ipsum - dolor sit amet consectetur", color: "black"},
   ]

   let lastClickedIndex = -1;

   function render(array) {
      document.getElementById("root").innerHTML = array.map((a, i) => `<li onclick="action(event, ${i})"
style="color: ${a.color}"><p>${a.text}</p></li>`).join("");
   }

   function clear(array) {
      return array.map((b) => ({
         text: b.text,
         color: 'black'
      }))
   }

   render(books)

   function action(event, index) {
      if (event.shiftKey) {
         console.log("shift")
         if (lastClickedIndex > -1) {

            let start = lastClickedIndex, finish = index;
```

```
            if (lastClickedIndex > index) {
               // swap
               [start, finish] = [finish, start];
            }

            books = clear(books);
            for (let i = start; i <= finish; i++) {
               books[i].color = 'orange';
            }
         } else {
            lastClickedIndex = index;
         }
      } else if (event.ctrlKey) {
         if (books[index].color === 'orange')
            books[index].color = 'black';
         else {
            books[index].color = 'orange';
         }

         // запоминаем последий клик
         lastClickedIndex = index;
      } else {
         if (books[index].color === 'black') {
            books = clear(books);

            books[index].color = 'orange';
         } else {
            books = clear(books);
         }

         // запоминаем последий клик
         lastClickedIndex = index;
      }

      render(books);
   }
</script>
</body>

</html>
```

Результат выполнения 6-го задания:

- Сомерсет Моэм - Бремя страстей человеческих
- Жуль Верн - 20 тысяч лье под водой
- Джон Рональд Руэл Толкин - Хоббит
- Владимир Беляев - Прыжок в ничто
- Стивен Хоккинг - Краткая история времени
- Анджей Сапковский - Меч предназначения
- Френк Герберт - Дюна
- Lorem ipsum - dolor sit amet consectetur

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