

**Московский Авиационный Институт
(Национальный исследовательский университет)**

«Информационные технологии и прикладная математика»
Кафедра вычислительной математики и программирования

**Лабораторные работы
по курсу «Сетевое программирование»**

№ ЛР	балл
1	15
2	10
3	10
4	10
6	9
9	9
посещения	4
доклад	20
отчет	20
Итого	99(73)
Оценка	4

Студент: Алексюнина Ю.В.
группа М8О-106М-20

Преподаватель:
Чернышов Л.Н.

Оценка: 4
Дата: 14.04.2022

Москва, 2022

Оглавление

<i>Лабораторная работа 1</i>	3
Задание (Вариант 13):	3
Листинг программы:	3
Выводы	6
<i>Лабораторная работа 2</i>	7
Задание	7
Листинг программы	7
Выводы:	11
<i>Лабораторная работа 3</i>	12
Задача	12
Листинг программы	12
Выводы:	12
<i>Лабораторная работа 4</i>	13
Задание:	13
Листинг программы:	13
Выводы:	24
<i>Лабораторная работа 6</i>	25
Задание:	25
Листинг программы:	25
Выводы:	28
<i>Лабораторная работа 9</i>	29
Задание:	29
Листинг программы:	29
Выводы:	30

Лабораторная работа 1

Свойства и методы объекта document, элементов форм. Динамическое создание элементов формы.

Задание (Вариант 13):

В HTML-документе определены 2 формы.

На первой управляющие кнопки и поля ввода, вторая - пустая.

При нажатии на кнопки первой формы на второй форме генерируются элементы (а).

Параметры и количество элементов определяются в элементах 1-й формы(б), а также заданы в виде массивов на JavaScript.

После построения элементов на второй форме по некоторым кнопкам/элементам

можно произвести некоторые действия (в), результаты которых отображаются на 2-форме.

Для генерации элементов и вывода результатов использовать свойство innerHTML.

(а) INPUT, TABLE (одно строка) создаются при загрузке документа

(б) В элементе INPUT выбрать номер ячейки таблицы.

(в) По кнопке удаляется указанная ячейка

Значения в таблицу заданы в массиве

Листинг программы:

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Лабораторная №1, вариант 13</title>
  <style>
    @import
url("https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css");
    body {
      padding: 10px;
    }
  </style>
```

```

</head>
<body>
  <h1>Алексюнина Юлия Вячеславовна М80-106М-21</h1>
  <p><i>(а)</i> Форма с элементами <code>INPUT</code> и <code>TABLE</code> и
пустой контейнер (работа осуществляется при нажатии на enter)</p>
  <p><i>(б)</i> В элементе <code>INPUT</code> определяется номер ячейки
таблицы</p>
  <p><i>(в)</i> При отправке формы в контейнере генерируется таблица без ячейки,
указанной в <code>INPUT</code></p>
  <form id="form">
    <div class="mb-3">
      <input class="form-control" id="number-input" type="number" />
    </div>
    <table id="table" class="table table-bordered"></table>
  </form>
  <div class="mb-3" id="container"></div>
  <script>
    const container = document.getElementById('container');
    const input = document.getElementById('number-input');
    const table = document.getElementById('table');
    const form = document.getElementById('form');
    const data = ['one', 'two', 'three', 'four', 'five'];

    input.setAttribute('max', data.length - 1);
    input.setAttribute('min', 0);

    const tableAppender = (item, element) => {
      const tr = document.createElement('tr');
      tr.innerHTML = `<td>${item}</td>`;
      element.appendChild(tr);
    }

    data.forEach((item) => tableAppender(item, table));

    form.addEventListener('submit', (e) => {
      e.preventDefault();
      const newData = data.filter((item, index) => index !== Number(input.value));
      const newTable = document.createElement('table');
      newTable.setAttribute('class', 'table table-bordered');
      newData.forEach((item) => tableAppender(item, newTable));
      container.innerHTML = '<hr style="color: #ff0066" size=10>';
      container.appendChild(newTable);
    })
  </script>
  <!-- <script src="./scripts/main.js"></script> -->
</body>
</html>

```

Алексюнина Юлия Вячеславовна М8О-106М-21

(а) Форма с элементами **INPUT** и **TABLE** и пустой контейнер (работа осуществляется при нажатии на enter)

(б) В элементе **INPUT** определяется номер ячейки таблицы

(в) При отправке формы в контейнере генерируется таблица без ячейки, указанной в **INPUT**

one
two
three
four
five

Рисунок 1. Начальное состояние страницы

Алексюнина Юлия Вячеславовна М8О-106М-21

(а) Форма с элементами **INPUT** и **TABLE** и пустой контейнер (работа осуществляется при нажатии на enter)

(б) В элементе **INPUT** определяется номер ячейки таблицы

(в) При отправке формы в контейнере генерируется таблица без ячейки, указанной в **INPUT**

one
two
three
four
five

Рисунок 2. Выбор ячейки для удаления

Алексюнина Юлия Вячеславовна М8О-106М-21

(а) Форма с элементами **INPUT** и **TABLE** и пустой контейнер (работа осуществляется при нажатии на enter)

(б) В элементе **INPUT** определяется номер ячейки таблицы

(в) При отправке формы в контейнере генерируется таблица без ячейки, указанной в **INPUT**

one
two
three
four
five

one
two
four
five

Рисунок 3. Появление сгенерированной таблицы

Выводы

Во время выполнения этой лабораторной работы я познакомилась с методами, которые взаимодействуют с DOM элементами страницы, такими как метод получения элемента `getElementById`; научилась работать с формами и таблицами.

Лабораторная работа 2

Реализовать простой справочник на LocalStorage.

Задание

Справочник должен:

1. отображаться в виде нумерованного списка (1)
2. содержать несколько полей (4). Первое поле ключевое. (Содержание придумать)
3. реализовывать добавление элемента справочника по кнопке.(1)
4. позволять удалять отдельные элементы с) по кнопке у каждого элемента (3)

Листинг программы

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Лабораторная №2</title>
    <style>
      /*@import
url("D:/study(active)/магистратура/сетевоеПрограммирование_Чернышов/task_1/css_1/
bootstrap.min.css");*/
      body {
        padding: 10px;
      }
      h3 {
        font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida
Sans Unicode', Geneva, Verdana, sans-serif;
        font-style: oblique;
      }
      h1 {
        font-family: Arial, Helvetica, sans-serif;
        /* background-image: url(); */
      }
    </style>
  </head>
  <body>
    <div class="body__inner">
      <div class="content">
```

```

<h1>Алексюнина Юлия Вячеславовна М80-106М-21</h1>
<div class="content__card">
  <h3>
    Реализовать простой справочник на LocalStorage <br />
    Справочник должен<br />
    1. отображаться в виде нумерованного списка (1)<br />
    2. содержать несколько полей (4). Первое поле ключевое. (Содержание
придумать)<br />
    3. реализовывать добавление элемента справочника по кнопке.(1)<br />
    4. позволять удалять отдельные элементы с) по кнопке у каждого
элемента (3) <br />
  </h3>
  <form name="firstForm">
    <div class="input__form">
      <label>
        <p> Id </p>
        <input class="inp-text" name="key" style="color:blue; width:
250px; height: 50px; border-color: coral;"/>
      </label>
      <label>
        <p> Name </p>
        <input class="inp-text" name="field1" style="color:blue; width:
250px; height: 50px; border-color: coral;"/>
      </label>
      <label>
        <p> Age </p>
        <input
          class="inp-text" name="field2" style="color:blue; width: 250px;
height: 50px; border-color: coral;"/>
      </label>
      <label>
        <p> Favourite Netflix series </p>
        <input class="inp-text" name="field3" style="color:blue; width:
250px; height: 50px; border-color: coral;"/>
      </label>
    </div>
    <hr>
    <button id="addbutton" style="color:coral; width: 150px; height:
50px;
                                top: 309px; left:1030px;"
                                onclick="addElement()" value="Добавление
элемента"><i>Add element</i></button>
    <hr>
  </form>

  <h2>Catalog: </h2>
  <ol></ol>

```



```

    </div>
  </div>
</div>

<script>
  let itemsArray = localStorage.getItem("items")
    ? JSON.parse(localStorage.getItem("items"))
    : [];

  let changeIndex = -1;

  const ol = document.querySelector("ol");

  localStorage.setItem("items", JSON.stringify(itemsArray));

  const data = JSON.parse(localStorage.getItem("items"));

  const liMaker = (key, field1, field2, field3, index) => {
    const li = document.createElement("li");
    li.innerHTML = `<div class="li__inner"><p><b>${key}</b> ${field1}
${field2} ${field3}</p>
  <button class="btn-remove" onclick="remove(${index})"> Delete element
</button></div>`;
    ol.appendChild(li);
  };

  data.forEach((item, index) => {
    liMaker(item[0], item[1], item[2], item[3], index);
  });

  function addToStorage(key, field1, field2, field3) {
    if (changeIndex !== -1) {
      itemsArray[changeIndex] = [key, field1, field2, field3];
      localStorage.setItem("items", JSON.stringify(itemsArray));
      ol.childNodes[
        changeIndex
      ].innerHTML = `<div class="li__inner"><p><b>${key}</b> ${field1}
${field2} ${field3}</p>
  <button class="btn-remove" onclick="remove(${changeIndex})"> Delete
element </button></div>`;
      changeIndex = -1;
      var formItem = document.querySelector("form");
      var h = formItem.querySelector("h2");
      h.innerHTML = `Add element:`;
    } else {
      itemsArray.push([key, field1, field2, field3]);
      localStorage.setItem("items", JSON.stringify(itemsArray));
    }
  }

```

```

        liMaker(key, field1, field2, field3, itemsArray.length - 1);
    }
}

function remove(index) {
    itemsArray.splice(index, 1);
    localStorage.setItem("items", JSON.stringify(itemsArray));
    while (ol.firstChild) {
        ol.removeChild(ol.firstChild);
    }
    itemsArray.forEach((item, index) => {
        liMaker(item[0], item[1], item[2], item[3], index);
    });
}

function addElement() {
    addToStorage(
        firstForm.key.value,
        firstForm.field1.value,
        firstForm.field2.value, // .slice(0, -1),
        firstForm.field3.value
    );
    // firstForm.key.value = "Текст";
    // firstForm.field1.value = "Текст";
    // firstForm.field2.value = "Текст";
}
</script>
</body>
</html>

```

Алексюнина Юлия Вячеславовна М8О-106М-21

Реализовать простой справочник на *LocalStorage*

Справочник должен

1. отображаться в виде нумерованного списка (1)
2. содержать несколько полей (4). Первое поле ключевое. (Содержание придумать)
3. реализовывать добавление элемента справочника по кнопке.(1)
4. позволять удалять отдельные элементы с) по кнопке у каждого элемента (3)

Id	Name	Age	Favourite Netflix series
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Catalog:

Рисунок 4. Начальное состояние

Алексюнина Юлия Вячеславовна М8О-106М-21

Реализовать простой справочник на *LocalStorage*

Справочник должен

1. отображаться в виде нумерованного списка (1)
2. содержать несколько полей (4). Первое поле ключевое. (Содержание придумать)
3. реализовывать добавление элемента справочника по кнопке.(1)
4. позволять удалять отдельные элементы с) по кнопке у каждого элемента (3)

Id	Name	Age	Favourite Netflix series
3	Hulia	24	SE

Add element

Catalog:

1. 3 ; Hulia ; 24 ; SE

Delete element

Рисунок 5. Пример с добавлением данных

Выводы:

Выполнив данную лабораторную работу я научилась обработке пользовательского ввода в коде на Javascript и работе с локальных хранилищем в браузерах.

Лабораторная работа 3

Реализовать простой справочник с сохранением данных на сервере.
Хранение данных в файлах или в СУБД.

Задача

Реализовать прошлую ЛР с хранением данных в СУБД.

Листинг программы

Реализация сервера, фронтэнд аналогичен Лабораторной №2

```
<?php
    $method = $_SERVER['REQUEST_METHOD'];
    $file = 'db.txt';
    if ($method == 'POST') {
        file_put_contents($file, json_encode($_POST, TRUE).PHP_EOL, FILE_APPEND);
    } elseif ($method == 'PUT') {
        $putdata = file_get_contents('php://input');
        echo $putdata;
        file_put_contents($file, $putdata.PHP_EOL, FILE_APPEND);
    } elseif ($method == 'DELETE') {
        file_put_contents($file, null);
    }
?>
```

Внешний вид не отличается от ЛР 2.

Выводы:

В ходе данной работы был разработан простой HTTP-сервер на языке php для работы со справочником и хранением данных в БД, а так же базовый фронтенд для него.

Лабораторная работа 4

Разработать несколько примеров с использованием JQuery, которые включал бы некоторые из заданных функций.

Задание:

1. `addClass()` +
2. `position()`, `offset()` + +
3. `password` +
4. `innerWidth()` +
5. `wrap()`, `wrapAll()` - -
6. `parent()` -
7. `undelegate()` +
8. `mouseover()` +
9. `keydown()` +
10. `slideToggle()` +

Листинг программы:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Лабораторная №4</title>
  <style>
    @import
url("https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css");
    body {
      padding: 10px;
      background-color: khaki;
    }

    /* 1 */
    p { margin: 8px; font-size:16px; }
      .selected { color:whitesmoke; }
      .highlight { background: rgb(113, 156, 236); }

    /* 2 */
    #sq{
      background: pink;
      width: 50px;
      height: 50px;
      text-align: center;
      padding: 10px;
```

```

        border:5px solid rgb(173, 49, 49);
    }

    /* 3 */
    form { margin:10px;padding:10px;border:5px solid lightcoral;
        height: 380px; width: 300px; }
    /* 4 */
    #pwidth{
        margin:10px;padding:5px;border:2px solid #666;
    }

    /* 5 - */
    /* 6 - */

    /* 7 */
    button {
        margin: 5px;
    }

    img {
        width: 30%;
        height: 30%;
    }

    button#theone {
        color: whitesmoke;
        background: orangered;
    }

    /* 8 */
    #mouseover{
        width: 200px; /* ширина элемента */
        height: 200px; /* высота элемента */
        background: rgb(156, 202, 49); /* цвет заднего фона */
        border-radius: 10px;
        padding: 50px;
    }

    /* 9 */
    #keydown{
        padding: 10px;
    }
    /* 10 */
    #slide{
        display: inline-block; /* блочно-строчные элементы (выстраиваем
элементы в линейку) */
        width: 75px; /* ширина элемента */

```

```

        height: 150px; /* высота элемента */
        margin-right: 20px; /* внешний отступ справа */
        background-color: rgb(162, 63, 201); /* цвет заднего фона */
    }

</style>
<script src="http://code.jquery.com/jquery-1.9.1.js"></script>
</head>
<body>
    <h1><code>Алексюнина Юлия Вячеславовна М80-106М-21</code></h1>
    <h2><code>Лабораторная №4: jQuery</code> </h2>
    <ol>
        <li><a href="1.html"> addClass() + </a>
        </li>
        <li><a href="2.html"> position(), offset() + + </a>
        </li>
        <li><a href="3.html"> password + </a>
        </li>
        <li><a href="4.html"> innerWidth() + </a>
        </li>
        <li><a href="5.html"> wrap(), wrapAll() - - </a>
        </li>
        <li><a href="6.html"> parent() - </a>
        </li>
        <li><a href="7.html"> undelegate() + </a>
        </li>
        <li><a href="8.html"> mouseover() + </a>
        </li>
        <li><a href="9.html"> keydown() + </a>
        </li>
        <li><a href="10.html"> slideToggle() + </a>
        </li>
    </ol>

    <h3>Применение addClass()</h3>
    <h6>addClass() добавляет класс(ы) каждому выбранному элементу страницы.</h6>
    <h6>Добавляем класс "selected"(белый цвет текста) и "highlight"(синий цвет
фона) к первому параграфу.</h6>
    <br> <br>

    <p>Hello</p>
    <p>and</p>
    <p>Goodbye</p>
    <br> <br>

    <h3>Применение position(), offset()</h3>

```

`<h6>position()` получает значение позиции элемента `top` и `left` относительно отступов его родителя.`</h6>`

`<h6>`Получаем положение второго параграфа относительно документа и заносим полученные координаты в него же`</h6>`

`
`

`<!-- position() -->`

`<div>`

`<p>Привет!</p>`

`</div>`

`<p></p>`

`
`

`<h6>offset()` возвращает/устанавливает координаты относительно начала страницы.`</h6>`

`<h6>`Кликаая по квадрату, изменяем его координаты`</h6>`

`
`

`<!-- offset() -->`

`<div>`

`<div id="sq"></div>`

`</div>`

`

`

`<h3>`Применение `password``</h3>`

`<h6>`Селектор `:password` выбирает все элементы `input`, которые имеют атрибут `type` со значением `password`.`</h6>`

`
`

`<form>`

`<input type = "text" placeholder = "Ваш логин">

`

`<input type = "password" placeholder = "Ваш пароль">

`

`<label><input type = "radio" name = "sex" value = "male" checked>`

Мужчина `</label>`

`<label><input type = "radio" name = "sex" value = "female"> Женщина`

`</label>

`

`<label><input type = "checkbox" name = "type1" value = "busy"> Есть собака(-и)</label>

`

`<label><input type = "checkbox" name = "type2" value = "free"> Есть кошка(-и)</label>

`

`<label><input type = "checkbox" name = "type3" value = "childfree"> Нет домашних животных/Есть другие</label>

`

`<input type = "reset">`

`</form>`


```
<br> <br>
```

```
<h3>Применение innerWidth()</h3>
```

```
<h6>innerHTML() возвращает ширину элемента, включая внутренние отступы, в пикселях.</h6>
```

```
<h6>Получаем innerWidth второго параграфа и записываем это значение в него</h6>
```

```
<br>
```

```
<p id="paragraph1"> Первый параграф </p>
```

```
<p id = "pwidth"></p>
```

```
<br> <br>
```

```
<h3>Применение undelegate()</h3>
```

```
<h6>undelegate() удаляет обработчики событий с элементов, подходящих под селектор, установленные методом delegate().</h6>
```

```
<h6>Включаем/отключаем обработку клика по кнопке</h6>
```

```
<br>
```

```
<button id="theone">Does nothing...</button>
```

```
<button id="bind">Bind Click</button>
```

```
<button id="unbind">Unbind Click</button>
```

```
<br> <br>
```

```

```

```
<br> <br>
```

```
<h3>Применение mouseover()</h3>
```

```
<h6>Обработчик событий "mouseover" (срабатывает, когда указатель мыши входит в элемент), или запускает это событие на выбранный элемент.</h6>
```

```
<h6>Вызываем событие mouseover() либо при нажатии на кнопку, либо при выходе указателя мыши из элемента.</h6>
```

```
<br>
```

```
<button>Нажми на меня</button>
```

```
<br>
```

```
<div id = "mouseover"></div>
```

```
<br> <br>
```

```
<h3>Применение keydown()</h3>
```

`<h6>`keydown() позволяет привязать обработчик событий нажатия на любую клавишу клавиатуры.`</h6>`

`<h6>`Получаем код клавиши при нажатии на нее.`</h6>`

`<p>`Нажмите любую кнопку на клавиатуре `</p>`

`
`

`<div id = "keydown"></div>`

`

`

`<h3>`Применение slideToggle()`</h3>`

`<h6>`slideToggle() позволяет плавно отобразить или скрыть выбранные элементы скользящим движением.`</h6>`

`<h6>`Сравниваем применение методов toggle() и slideToggle().`</h6>`

`

`

`<button class = "toggle">`Метод .toggle()`</button>`

`<button class = "slide-toggle">`Метод .slideToggle()`</button>`

`<hr>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`<div id = "slide"></div>`

`

`

`<script>`

`//addClass()`

`//Имя класса(или имена классов, через пробел), которое требуется добавить к связанному элементу.`

`$("#p:first").addClass("selected highlight");`

`//position()` - Получает значение позиции элемента top и left относительно отступов его родителя.

`//Получаем положение последнего параграфа относительно документа`

`//и заносим полученные координаты в него же`

```

var p = $("p:eq(4)");
var position = p.position();
$("p:eq(4)").text( "left: " + position.left + ", top: " + position.top );

// offset()
//.offset( function(index, coords) )
//Устанавливает координаты выбранных элементов, которые возвращает функция
function.
//При вызове ей передаются параметры: index – (позиция элемента в наборе) и
value (текущие координаты элемента).
$("#sq").click(function () {
    $(this).offset(function(i, coord){
        var newCoord = {};
        newCoord.top = coord.top + 10;
        newCoord.left = coord.left + 10;
        return newCoord;
    });
});

//password
//jQuery селектор :password выбирает все элементы <input>, которые имеют
атрибут type со значением password.
$(document).ready(function(){
    $(".:password").css("border", "1px solid orange"); // выбираем все элементы
<input>, которые имеют атрибут type со значением password
});

//innerWidth()
//ширина элемента с учетом размера внутренних отступов (padding).
var p2 = $("p:eq(6)");
$("p:eq(6)").text( "innerWidth:" + p2.innerWidth() );

//undelegate()
//Удаляет обработчики событий с элементов, подходящих под селектор,
установленные методом delegate().
function aClick() {
    $( "img" ).show().fadeOut( "slow" );
}
$( "#bind" ).click(function() {
    $( "body" )
        .delegate( "#theone", "click", aClick )
        .find( "#theone" ).text( "Can Click!" );
});
$( "#unbind" ).click(function() {
    $( "body" )
        .undelegate( "#theone", "click", aClick )
        .find( "#theone" ).text( "Does nothing..." );
});

```

```

});

//mouseover()
//обработчик событий "mouseover" (срабатывает, когда указатель мыши входит в
элемент),
//или запускает это событие на выбранный элемент.
$( document ).ready(function(){
    $( "button:eq(3)" ).click(function(){ // задаем функцию при нажатии на
элемент <button>
        $( "div:eq(3)" ).mouseover(); // вызываем событие mouseover на
элементе <div>
    });
    var num = 0; // переменная (счетчик возникновения события)
    $( "div:eq(3)" ).mouseover(function(){ // задаем функцию при вхождении
указателя мыши в элемент <div>
        num++; // инкремент
        $( "div:eq(3)" ).text( " " + num + " times!"); // выводим
количество срабатываний события
        if(num % 100 == 0) {
            $( "div:eq(3)" ).text( "surprise, mazafaka");
        }
    });
});

//keydown()
//позволяет привязать JavaScript обработчик событий "keydown"
//(нажатие на любую клавишу клавиатуры), или запускает это событие на
выбранный элемент.
$( document ).ready(function(){
    $( "body" ).keydown(function( event ){ // задаем функцию при нажатии любой
клавиши клавиатуры на элементе
        $( "div:eq(4)" ).text( "Код нажатой кнопки: " + event.which);
        // выводим код нажатой клавиши
    });
});

//slideToggle()
//позволяет плавно отобразить или скрыть выбранные элементы скользящим
движением.
//Если элемент изначально отображается, то он будет скрыт, если элемент
скрыт, то он будет отображен.
$( document ).ready(function(){
    $( ".toggle" ).click(function(){ // задаем функцию при нажатии на элемент с
классом toggle
        $( "div:gt(4)" ).toggle(); // скрываем, или отображаем все элементы
<div>
    });
});

```

```

    $( ".slide-toggle" ).click(function(){ // задаем функцию при нажатии на
элемент с классом slide-toggle
        $( "div:gt(4)" ).slideToggle(); // плавно скрываем, или отображаем все
элементы <div>
    });
});
</script>
</body>
</html>

```

Алексюнина Юлия Вячеславовна М80-106М-21

Лабораторная №4: jQuery

1. [addClass\(\)](#) +
2. [position\(\).offset\(\)](#) + +
3. [password](#) +
4. [innerWidth\(\)](#) +
5. [wrap\(\).wrapAll\(\)](#) - -
6. [parent\(\)](#) -
7. [undelegate\(\)](#) +
8. [mouseover\(\)](#) +
9. [keydown\(\)](#) +
10. [slideToggle\(\)](#) +

Применение addClass()

addClass() добавляет класс(ы) каждому выбранному элементу страницы.

Добавляем класс "selected" (белый цвет текста) и "highlight" (синий цвет фона) к первому параграфу.

Hello

and

Goodbye

Применение position(), offset()

position() получает значение позиции элемента top и left относительно отступов его родителя.

Получаем положение второго параграфа относительно документа и заносим полученные координаты в него же

Привет!

left: 10, top: 819.5729370117188

offset() возвращает/устанавливает координаты относительно начала страницы.

Рисунок 6. Начальное состояние, часть 1

Кликав по квадрату, изменяем его координаты



Применение password

Селектор `:password` выбирает все элементы `input`, которые имеют атрибут `type` со значением `password`.

☒ Мужчина ☐ Женщина

☐ Есть собака(-и)

☐ Есть кошка(-и)

☐ Нет домашних животных/Есть другие

Применение `innerWidth()`

`innerWidth()` возвращает ширину элемента, включая внутренние отступы, в пикселях.

Получаем `innerWidth` второго параграфа и записываем это значение в него

Первый параграф

`innerWidth:1219`

Рисунок 7. Начальное состояние, часть 2

Применение undelegate()

undelegate() удаляет обработчики событий с элементов, подходящих под селектор, установленные методом delegate().

Включаем/отключаем обработку клика по кнопке

Does nothing...

Bind Click

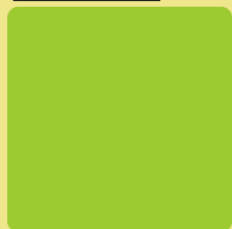
Unbind Click

Применение mouseover()

Обработчик событий "mouseover" (срабатывает, когда указатель мыши входит в элемент), или запускает это событие на выбранный элемент.

Вызываем событие mouseover() либо при нажатии на кнопку, либо при выходе указателя мыши из элемента.

Нажми на меня



Применение keydown()

keydown() позволяет привязать обработчик событий нажатия на любую клавишу клавиатуры.

Получаем код клавиши при нажатии на нее.

Нажмите любую кнопку на клавиатуре

Код нажатой кнопки: 189

Рисунок 8. Начальное состояние, часть 3

Применение slideToggle()

slideToggle() позволяет плавно отобразить или скрыть выбранные элементы скользящим движением.

Сравниваем применение методов toggle() и slideToggle().

Метод .toggle()

Метод .slideToggle()



Рисунок 9. Начальное состояние, часть 4

Выводы:

В ходе выполнения данной лабораторной работы я научилась использовать определенные функции библиотеки JQuery, которые позволяют выполнять различные манипуляции с кодом и внешним видом страницы.

Лабораторная работа 6

Работа с JSON-данными.

Задание:

На JS написать функцию, которая строит таблицу с данными с сайта data.mos.ru. Данные получать через запрос в их API в формате OpenData. Колонок не менее 4-х.

Листинг программы:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Лабораторная №6</title>
  <style>
    @import
url("https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css");
    body {
      padding: 10px;
      background-color: khaki;
    }
  </style>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></scri
pt>
  <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></sc
ript>

  <style>
    table.table-bordered {
      border: 1px solid #e9ecef;
      margin-top: 20px;
    }

    table.table-bordered > thead > tr > th {
      border: 1px solid rgb(121, 113, 113);
      border-bottom: 3px solid #e9ecef;
      background-color: rgb(230, 140, 182);
    }
  </style>
</head>
<body>
  <div>
    <table>
      <thead>
        <tr>
          <th></th>
          <th></th>
          <th></th>
          <th></th>
          <th></th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td></td>
          <td></td>
          <td></td>
          <td></td>
          <td></td>
          <td></td>
        </tr>
      </tbody>
    </table>
  </div>
</body>
</html>
```

```

table.table-bordered > tbody > tr > td {
    border: 1px solid #e9ecef;
}

.table-hover tbody tr:hover td, .table-hover tbody tr:hover th {
    background-color: #e5efff;
}

table.table th:last-child {
    width: 100px;
}

table.table td a {
    cursor: pointer;
    display: inline-block;
    margin: 0 5px;
    min-width: 24px;
}

table.table td i {
    font-size: 22px;
}
</style>
</head>
<body>
<div class="container p-1">
    <h1><code>Алексюнина Юлия Вячеславовна М80-106М-21</code></h1>
    <h2><code>Лабораторная №6: Работа с JSON-данными</code> </h2>
    <table class="table table-hover table-bordered table-light shadow p-3 bg-
white rounded" id="TableWithData">
        <thead>
            <tr>
                <th scope="col">Id</th>
                <th scope="col">External Id</th>
                <th scope="col">Number</th>
                <th scope="col">Cad number</th>
                <th scope="col">Address</th>
                <th scope="col">Square</th>
                <th scope="col">Owner type</th>
                <th scope="col">In renting</th>
                <th scope="col">Permitted use</th>
                <th scope="col">Global ID</th>
            </tr>
        </thead>
        <tbody id="TableContent"></tbody>
    </table>
</div>

```

```

</body>
<script>
    var content = document.getElementById("TableContent");

    function create_column_from_element(tr_object, name) {
        let td = document.createElement("td");
        td.appendChild(document.createTextNode(name));
        tr_object.appendChild(td);
    }

    function orderToRow(elem_id, element) {
        let tr = document.createElement("tr");
        create_column_from_element(tr, elem_id);
        create_column_from_element(tr, element.Id);
        create_column_from_element(tr, element.Number);
        create_column_from_element(tr, element.Cells.CAD_NUM);
        create_column_from_element(tr, element.Cells.ADDRESS_LANDMARKS);
        create_column_from_element(tr, element.Cells.SQUARE);
        create_column_from_element(tr, element.Cells.OWNER_TYPE);
        create_column_from_element(tr, element.Cells.IN_RENTING);
        create_column_from_element(tr, element.Cells.PERMITTED_USE);
        create_column_from_element(tr, element.Cells.GLOBALID);

        return tr;
    }

    let dataset_id = 521;
    let api_key = "300d08650c4543aca2eae9445341e11";
    let url = "https://apidata.mos.ru/v1/datasets/" + dataset_id +
"/rows?api_key=" + api_key + "&$top=100"
    $.get(url, function (data) {
        for (const i in data) {
            try {
                let tr = orderToRow(parseInt(i) + 1, data[i]);
                content.append(tr);
            } catch (e) {
                // pass
            }
        }
    })
</script>
</html>

```

Алексюнина Юлия Вячеславовна М80-106М-21

Лабораторная №6: Работа с JSON-данными

Id	External Id	Number	Cad number	Address	Square	Owner type	In renting	Permitted use	Global ID
1	b6e71207-6eab-4b3e-9780-d4cff343c5da	1	77:05:0004012:100	ул Коломенская д. 27 вл. 2 (г Москва, ул Коломенская, вл 27, корпус 2)	8783.0	город Москва	Нет	ЭКСПЛУАТАЦИИ ЗДАНИЯ И СТРОЕНИЙ ДЕТСКОГО САДА С ПРИЛЕГАЮЩИМ ЗЕМЕЛЬНЫМ УЧАСТКОМ	89673182
2	4327238a-c65c-4be1-8c80-2c6469f94f6c	2	77:01:0002014:139	ул Садовническая д. 73 стр. 24 (г Москва, ул Садовническая, вл 73, стр 24)	2797.0	город Москва	Нет	эксплуатации здания и прилегающей территории под учебно-воспитательные цели	89672204
3	510fa19e-9342-4520-a12b-da9dd483f9bd	3	77:07:0008006:1014	ул Петра Алексеева д. 10Т (г. Москва, ул. Петра Алексеева, вл. 10Т)	41.0	город Москва	Да	объекты размещения помещений и технических устройств газорегуляторных пунктов (ГРП) (1.2.10)	89674607
4	5ccd8e3b-1004-4b36-9780-d4cff343c5da	4	77:03:0002016:69	ул Новосибирская д. 2А (г Москва, ул	13628.0	город Москва	Нет	ЭКСПЛУАТАЦИИ ЗДАНИЯ школы и	89670529

Рисунок 10. Внешний вид построенного справочника по данным о зданиях Москвы

Выводы:

В ходе выполнения лабораторной работы я научилась работать с файлами формата JSON, обрабатывать данные и выводить их в виде таблицы.

Лабораторная работа 9

Построение графика/диаграммы по данным

Задание:

Построение графика с помощью библиотеки canvas

Самостоятельно выбрать форму графика/диаграммы

Листинг программы:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Лабораторная №9</title>
  <style>
    @import
url("https://cdn.jsdelivr.net/npm/bootstrap@5.1.0/dist/css/bootstrap.min.css");
    body {
      padding: 10px;
      background-color: khaki;
    }
  </style>
</head>

<h1><code>Алексюнина Юлия Вячеславовна М80-106М-21</code></h1>
<h2><code>Лабораторная №9: Построение графика/диаграммы по данным на
canvas</code> </h2>
<script>

function finit(){
  cv = document.getElementById("cv").getContext("2d");
  with (cv) {
    strokeRect(0, 0, 605, 300);
    translate(0, 300);

    strokeStyle = "rgb(255,165,0)";
    moveTo(0, 0);
    for(j = 0; j < 31; j++) {
      lineTo(20 * j, -(Math.random() * 300));
    }
  }
}
```

```

        fillStyle = "rgb(123,135,67)";
        x = 0;
        for(j = 0; j < 31; j++) {
            fillRect(x, 100, 5, -(Math.random() * 300) - 100);
            x += 20;
        }

        stroke();
    }
}

</script>
<body onload="finit()">
    <canvas id="cv" width=700 height=300>??</canvas>

```

Алексюнина Юлия Вячеславовна М80-106М-21

Лабораторная №9: Построение графика/диаграммы по данным на canvas

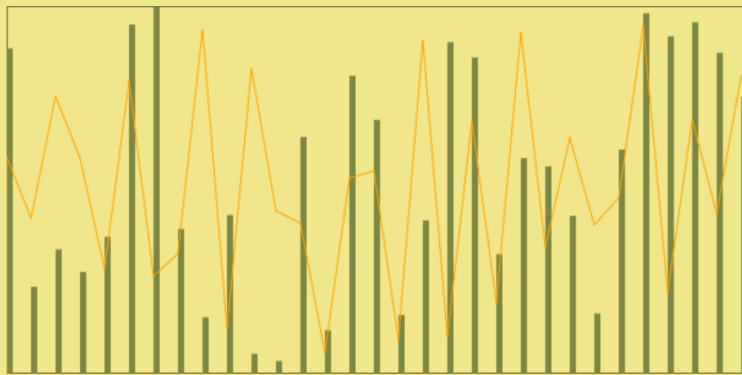


Рисунок 10. Построенный случайным образом график

Выводы:

В ходе выполнения этой лабораторной работы я научилась использовать библиотеку canvas, строить графики и диаграммы по данным и визуализировать их.