**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

учреждение образования

МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ ТЕХНОЛОГИЙ

##### Группа 73 ТП

## Отчет

## по лабораторной работе № 13

**"Использование объектов на языке JavaScript"**

учебный предмет

“Программные средства создания Интернет приложений”

**Исполнитель: А.А. Артюшевский**

**Руководитель: Е.П. Сушкевич**

**Минск, 2025**

**Цель работы:**

Сформировать умение использовать объекты в сценариях на языке JavaScript

**Задания**

1. Создание и вывод информации об объекте

* Создайте на странице форму с двумя полями: "Название города" и "Население".
* После нажатия кнопки "Добавить", создайте объект city с этими свойствами и выведите его данные на страницу в виде списка.
* Добавьте к объекту метод getInfo(), который возвращает строку с информацией о городе, и используйте его для вывода.

2. Работа с массивами и методами Array

* На странице есть поле для ввода чисел через запятую и кнопка "Рассчитать".
* После нажатия кнопки:
  + Создайте массив из введенных чисел.
  + Выведите сумму, среднее значение, отсортированный массив.
  + Используйте методы split, map, reduce, sort.

3. Использование объекта Date

* Создайте кнопку "Показать дату".
* При нажатии выводите текущую дату и время в формате "дд.мм.гггг чч:мм:сс".
* Реализуйте обновление времени каждую секунду с помощью setInterval.

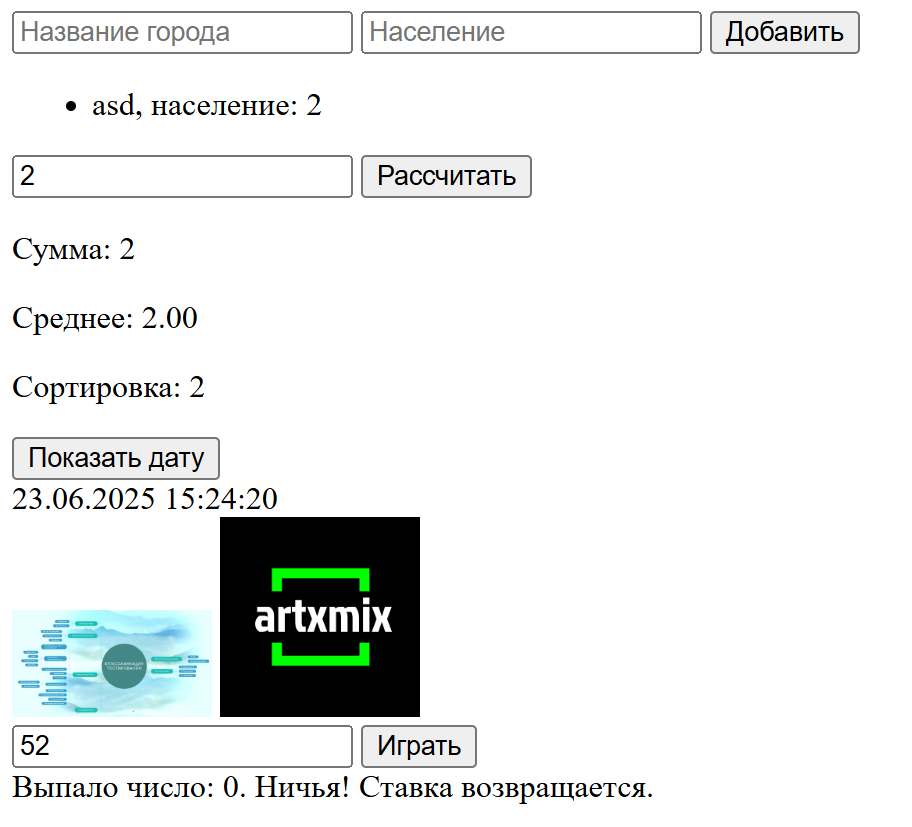
4. Манипуляции с изображениями через объекты

* На странице размещены миниатюры изображений (например, 8 штук).
* При клике на миниатюру появляется модальное окно с увеличенной версией изображения.
* Используйте методы создания элементов (createElement, appendChild) и работу с атрибутами объектов DOM.

5. Генерация и обработка случайных событий

* Реализуйте "лотерею": пользователь вводит ставку, нажимает "Играть".
* Программа генерирует случайное число от -5 до 5.
* В зависимости от результата выводится сообщение о выигрыше или проигрыше.

Решение:



index.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>Document</title>

    <style>

        .modal {

            display: none;

            position: fixed;

            top: 0;

            left: 0;

            right: 0;

            bottom: 0;

            background: rgba(0, 0, 0, 0.7);

            justify-content: center;

            align-items: center;

        }

        .modal img {

            max-width: 80%;

            max-height: 80%;

        }

    </style>

</head>

<body>

    <form id="cityForm">

        <input type="text" id="cityName" placeholder="Название города" required>

        <input type="number" id="population" placeholder="Население" required>

        <button type="submit">Добавить</button>

    </form>

    <ul id="cityList"></ul>

    <input type="text" id="numbers" placeholder="Напр. 3, 5, 1, 9">

    <button onclick="processNumbers()">Рассчитать</button>

    <div id="arrayResult"></div>

    <button onclick="startClock()">Показать дату</button>

    <div id="clock"></div>

    <div id="gallery">

        <img src="imgs/thumb1.jpg" data-full="full1.jpg" class="thumb" width="100">

        <img src="imgs/thumb2.jpg" data-full="full2.jpg" class="thumb" width="100">

    </div>

    <div id="modal" class="modal" onclick="this.style.display='none'">

        <img id="modalImg">

    </div>

    <input type="number" id="bet" placeholder="Введите ставку">

    <button onclick="playLottery()">Играть</button>

    <div id="lotteryResult"></div>

    <script src="/scripts/script1.js"></script>

    <script src="/scripts/script2.js"></script>

    <script src="/scripts/script3.js"></script>

    <script src="/scripts/script4.js"></script>

    <script src="/scripts/script5.js"></script>

</body>

</html>

script1.js:

document.getElementById("cityForm").addEventListener("submit", function (e) {

    e.preventDefault();

    const name = document.getElementById("cityName").value;

    const population = document.getElementById("population").value;

    const city = {

        name,

        population,

        getInfo() {

            return `${this.name}, население: ${this.population}`;

        }

    };

    const li = document.createElement("li");

    li.textContent = city.getInfo();

    document.getElementById("cityList").appendChild(li);

    this.reset();

});

script2.js:

function processNumbers() {

    const input = document.getElementById("numbers").value;

    const numbers = input.split(",").map(n => parseFloat(n.trim())).filter(n => !isNaN(n));

    const sum = numbers.reduce((acc, val) => acc + val, 0);

    const avg = numbers.length ? (sum / numbers.length).toFixed(2) : 0;

    const sorted = [...numbers].sort((a, b) => a - b);

    document.getElementById("arrayResult").innerHTML = `

      <p>Сумма: ${sum}</p>

      <p>Среднее: ${avg}</p>

      <p>Сортировка: ${sorted.join(", ")}</p>

    `;

}

script3.js:

function playLottery() {

    const bet = parseFloat(document.getElementById("bet").value);

    if (isNaN(bet) || bet <= 0) {

        return alert("Введите корректную ставку!");

    }

    const outcome = Math.floor(Math.random() \* 11) - 5;

    let message = `Выпало число: ${outcome}. `;

    if (outcome > 0) {

        message += `Вы выиграли ${bet \* outcome} у.е.! 🎉`;

    } else if (outcome < 0) {

        message += `Вы проиграли ${bet \* Math.abs(outcome)} у.е. 😞`;

    } else {

        message += "Ничья! Ставка возвращается.";

    }

    document.getElementById("lotteryResult").textContent = message;

}

script4.js:

document.querySelectorAll(".thumb").forEach(img => {

    img.addEventListener("click", () => {

        document.getElementById("modalImg").src = img.dataset.full;

        document.getElementById("modal").style.display = "flex";

    });

});

script5.js:

function formatDate(date) {

    const pad = n => n.toString().padStart(2, "0");

    return `${pad(date.getDate())}.${pad(date.getMonth() + 1)}.${date.getFullYear()} ${pad(date.getHours())}:${pad(date.getMinutes())}:${pad(date.getSeconds())}`;

}

function startClock() {

    updateClock();

    setInterval(updateClock, 1000);

}

function updateClock() {

    const now = new Date();

    document.getElementById("clock").textContent = formatDate(now);

}

Контрольные вопросы:

1. Как создать объект в JavaScript и добавить ему свойства и методы?

Ответ:

Объекты в JS можно создавать разными способами. Самый популярный — **литеральный синтаксис**:

const user = {

name: "Артём",

greet() {

return `Привет, я ${this.name}`;

}

};

user.age = 30; // добавили свойство

user.sayBye = function() {

return "Пока!";

}; // добавили метод

1. В чем разница между проверкой существования свойства объекта с помощью оператора in и сравнением с undefined? Когда предпочтительнее использовать оператор in?

Ответ:

**🔹 Сравнение с undefined**

if (obj.prop === undefined) { ... }

Проблема: свойство может **существовать**, но его значение — undefined.

**Оператор in**

if ("prop" in obj) { ... }

Работает **точно**: проверяет, есть ли такое свойство (в том числе унаследованное).

1. Какие методы массива JavaScript можно использовать для перебора и трансформации данных? Как их применить для динамического создания HTML-элементов?

Ответ:

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| forEach | Перебор без возвращаемого значения |
| map | Преобразование массива |
| filter | Отбор элементов |
| reduce | Агрегация (сумма, среднее…) |
| some/every | Проверка условий |

1. Как с помощью объекта Date получить текущую дату и время и отобразить их на странице с обновлением каждую секунду?

Ответ:

<div id="clock"></div>

<script>

function updateClock() {

const now = new Date();

const format = (n) => n.toString().padStart(2, "0");

const timeStr = `${format(now.getDate())}.${format(now.getMonth()+1)}.${now.getFullYear()} ` +

`${format(now.getHours())}:${format(now.getMinutes())}:${format(now.getSeconds())}`;

document.getElementById("clock").textContent = timeStr;

}

setInterval(updateClock, 1000);

updateClock(); // сразу отобразить без задержки

</script>