**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«Чувашский государственный университет имени И.Н. Ульянова»**

**Факультет информатики и вычислительной техники**

**Кафедра вычислительной техники**

Отчет по дисциплине «Web-программирование»

Лабораторная работа №4

«Работа с объектами. Таймеры»

Выполнил:

студент группы ИВТ-11-21

Еремеев Сергей Алексеевич

Принял: Первова Н. В.

Чебоксары, 2023

**Цель работы.**

Изучить основы базового языка по работе с объектами и таймерами JavaScript. Получить основные навыки создания и использования объектов для работы с данными в JavaScript.

**Задание.**

В файле countries.js приведён массив countries с информацией о государствах мира и массив about с заголовочной информацией к массиву countries (см. предыдущую работу).

В качестве индивидуального задания на основе лабораторной работы №3 необходимо разработать два объекта, предназначенных для отображения и изменения информации о государствах.

Для всех вариантов на панель навигации сайта внедрить блок для отображения времени, оставшегося до:

1.конца учебного года (30 июня) – для нечётных вариантов;

2.начала следующего учебного года (1 сентября) – для чётных вариантов.

Формат отображения времени (ХХХХ – год, например, 2013)

Вариант 8. Создать массив объектов.

**Описание структуры объектов obj1 и массива объектов masResult**

var massResult = []; // массив который будет содержать обьекты

var obj1 = {

OutCountries: function() {

function makeTableFromCountry() {

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

var mass1 = countries[i];

var nonObj = [];

var hisObj = [];

var objResult = {

}

for (let j = 0; j < mass1.length; j++) {

if (typeof mass1[j] === "object") {

if (Array.isArray(mass1[j])) {

for (let l = 0; l < mass1[j].length; l++) {

if (Array.isArray(mass1[j][l])) {

litObj = mass1[j];

} else {

geoObj = mass1[j];

}

}

} else {

nonObj.push(mass1[j][1] + " - " + mass1[j][2]);

for (var k = 0; k < Object.keys(mass1[j]).length; k++) {

var key = Object.keys(mass1[j])[k];

hisObj.push(key + " - " + mass1[j][key]);

}

}

} else {

if (typeof mass1[j] === "number") {

nonObj.unshift(mass1[j]);

} else if (typeof mass1[j] === "string") {

nonObj.push(mass1[j]);

}

}

}

objResult[nonObj[1] + "-" + nonObj[2]] = hisObj + "<br>";

massResult.push(objResult);

}

}

makeTableFromCountry();

}

}

obj1.OutCountries();

document.write(JSON.stringify(massResult));

Данный код создает массив massResult, который будет содержать объекты.

Затем определен объект obj1, у которого есть метод OutCountries. Внутри этого метода определена функция makeTableFromCountry.

Функция makeTableFromCountry проходит по каждому элементу массива countries. Для каждого элемента массива mass1 выполняются следующие действия:

1. Создается пустой массив nonObj для хранения значений, которые не являются объектами.

2. Создается пустой массив hisObj для хранения пар "ключ - значение" объектов.

3. Создается пустой объект objResult, который будет содержать результаты для текущего элемента массива mass1.

4. Внутри цикла проходится по каждому элементу массива mass1. Если текущий элемент является объектом:

- Если текущий элемент является массивом, то происходит проверка на наличие внутренних массивов и объектов. Если внутренний массив найден, он присваивается переменной litObj, если внутренний объект найден, он присваивается переменной geoObj.

- Если текущий элемент не является массивом, то его значения добавляются в массив nonObj, а также происходит проход по всем ключам объекта и добавление пар "ключ - значение" в массив hisObj.

5. Если текущий элемент не является объектом, то происходит проверка на его тип. Если тип - число, то оно добавляется в начало массива nonObj, если тип - строка, то оно добавляется в конец массива nonObj.

6. Созданный объект objResult заполняется данными из массивов nonObj и hisObj.

7. Заполненный объект objResult добавляется в массив massResult.

После выполнения функции makeTableFromCountry, вызывается метод OutCountries объекта obj1. Затем результат в виде строки JSON выводится на страницу с помощью метода document.write.

Стандартные функции и операторы javascript:

1. for - цикл, который позволяет выполнять определенный блок кода несколько раз, пока выполняется определенное условие.

2. if - оператор условия, который выполняет определенный блок кода, если определенное условие истинно.

3. typeof - оператор, который возвращает тип данных переменной.

4. Array.isArray() method - метод, который проверяет, является ли переданный аргумент массивом.

5. Object.keys() method - метод, который возвращает массив из всех ключей объекта.

6. push() - метод, который добавляет один или несколько элементов в конец массива.

7. unshift() - метод, который добавляет один или несколько элементов в начало массива.

8. JSON.stringify() - метод, который преобразует значение JavaScript в строку JSON.

**Таймер.**

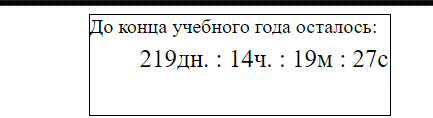


Рисунок 2 – Вид таймера

var necessary\_date = *new* Date("June 30, 2024 00:00:00");

var timer = document.getElementById("timer");

// Функция Math.trunc() возвращает целую часть числа путём удаления всех дробных знаков.

function Timer(){

    var current\_date = Date.now();

    var end = necessary\_date - current\_date;

    var days = Math.trunc(end/1000/60/60/24);

    end = end - days\*1000\*60\*60\*24;

    var hours = Math.trunc(end/1000/60/60);

    end = end - hours\*1000\*60\*60;

    var minutes = Math.trunc(end/1000/60);

    end = end - minutes\*1000\*60;

    var seconds = Math.trunc(end/1000);

    var st = days + "дн.  : " + hours  + "ч. : " + minutes + "м : " + seconds + "с";

    timer.innerText = st; //полностью удалит все предыдущее содержимое и перезапишет его

};

setInterval(Timer,500);

Стандартные функции и операторы в коде выше:

1. new Date() - создает объект типа Date, который содержит дату и время, указанные в параметре.

2. document.getElementById() - возвращает элемент страницы с указанным идентификатором.

3. Math.trunc() - возвращает целую часть числа путем удаления всех дробных знаков.

4. Date.now() - возвращает количество миллисекунд, прошедших с начала эпохи Unix до текущего момента времени.

5. setInterval() - вызывает функцию через определенный интервал времени (в миллисекундах), который указывается вторым параметром.

**Вывод.** Изучил основы базового языка по работе с объектами и таймерами JavaScript. Получить основные навыки создания и использования объектов для работы с данными в JavaScript.