**Лабораторная работа №8**

**Тема работы:** Паттерн «Модуль»

**Цель работы:** закрепить навыки по работе с паттерном «Модуль»

**Теоретические сведения**

**Функция inArray**

**Задача:** реализуйте функцию **inArray**, которая будет проверять, есть ли в массиве элемент с таким значением или нет. Первым параметром функция должна принимать значение, а вторым - массив. Если такой элемент есть - функция должна вернуть true, а если нет - false.

**Решение:**

function inArray(value, arr) {

for (var i **=** 0; i **<** arr.length; i**++**) {

if(arr[i] **==** value) {

return true;

}

}

return false;

}

var arr **=** [1, 2, 3, 4, 5];

alert(inArray(3, arr)); //выведет 3

**Простые числа**

**Задача:** реализуйте функцию **isSimple**, которая параметром будет принимать число и проверять, простое оно или нет. Простое число - это число, которое не делится ни на одно другое число (кроме как на 1 и на само себя - на это делятся все числа). Если число простое - функция должна вернуть true, в противном случае false.

**Решение:** самый простой способ проверить число на простоту - попробовать поделить его на все числа от 2 до самого числа (на само число не делим). Если ни на одно из этих чисел наше число не поделится - оно простое, а если хотя бы на одно поделится - вернем false и выйдем из функции:

function isSimple(num) {

for (var i **=** 2; i **<** num; i**++**) {

if(num **%** i **==** 0) {

return false;

}

}

return true;

}

alert(isSimple(13)); //выведет true

**Делители**

**Задача:** реализуйте функцию **getDivisors**, которая параметром будет принимать число и возвращать массив его делителей, то есть чисел, на которое делится наше число. К примеру, если мы передадим число **24** - мы должны получить массив [1, 2, 3, 4, 6, 12, 24].

**Решение:**

function getDivisors(num) {

var result **=** [];

for (var i **=** 1; i **<=** num; i**++**) {

if(num **%** i **==** 0) {

result.push(num);

}

}

return result;

}

alert(getDividers(24)); //выведет [1, 2, 3, 4, 6, 12, 24]

**Пересечение массивов**

**Задача:** реализуйте функцию **getIntersection**, которая параметрами будет принимать два массива и возвращать массив элементов, которые есть и в одном, и во втором массиве (это называется *пересечение* массивов). К примеру, getIntersection([1, 2, 3], [2, 3, 4]) должно вернуть [2, 3].

**Решение:** переберем один из переданных массивов (не имеет значения какой) с помощью цикла. В цикле воспользуемся функцией **inArray**, которую мы с вами разработали выше. С ее помощью будем проверять, есть ли текущий элемент одного массива во втором. Если есть - записываем его в массив с результатом. Вот решение:

function getIntersection(arr1, arr2) {

var result **=** [];

for (var i **=** 0; i **<** arr1.length; i**++**) {

if(inArray(arr1[i], arr2)) {

result.push(arr1[i]);

}

}

return result;

}

alert(getIntersection([1, 2, 3], [2, 3, 4])); //выведет [2, 3]

**НОД**

**Задача:** реализуйте функцию **getGreatestCommonDivisor**, которая параметрами будет принимать два числа и возвращать наибольший общий делитель (НОД) этих двух чисел. К примеру, для чисел **12** и **18** НОД равен 6.

**Решение:** используем разработанные нами функции. С помощью **getDivisors** найдем массив делителей одного и второго числа, а с помощью **getIntersection** - получим массив общих делителей. Останется просто взять максимальный из общих делителей - и получим результат:

function getGreatestCommonDivisor(num1, num2) {

var divisors **=** getIntersection(getDivisors(num1), getDivisors(num2)); // ['2', '3', '6']

return Number(Max.apply(null, divisors));

}

alert(getGreatestCommonDivisors(12, 18)); //выведет 6

**Порядок выполнения работы**

**Практическая часть**

1. Дана строка. Сделайте заглавным первый символ каждого слова этой строки. Для этого сделайте вспомогательную функцию **ucfirst**, которая будет получать строку, делать первый символ этой строки заглавным и возвращать обратно строку с заглавной первой буквой.
2. Дана строка вида 'var\_text\_hello'. Сделайте из него текст 'varTextHello'.
3. Сделайте функцию **inArray**, которая определяет, есть в массиве элемент с заданным текстом или нет. Функция первым параметром должна принимать текст элемента, а вторым - массив, в котором делается поиск. Функция должна возвращать true или false.
4. Дана строка, например, '123456'. Сделайте из нее '214365'.