**Сумма свойств**

Есть объект salaries с зарплатами. Напишите код, который выведет сумму всех зарплат.

Если объект пустой, то результат должен быть 0.

Например:

var salaries = {

"Вася": 100,

"Петя": 300,

"Даша": 250

};

//... ваш код выведет 650

**Свойство с наибольшим значением**

Есть объект salaries с зарплатами. Напишите код, который выведет имя сотрудника, у которого самая большая зарплата.

Если объект пустой, то пусть он выводит «нет сотрудников».

Например:

"use strict";

var salaries = {

"Вася": 100,

"Петя": 300,

"Даша": 250

};

// ... ваш код выведет "Петя"

**Умножьте численные свойства на 2**

Создайте функцию multiplyNumeric, которая получает объект и умножает все численные свойства на 2. Например:

// до вызова

var menu = {

width: 200,

height: 300,

title: "My menu"

};

multiplyNumeric(menu);

// после вызова

menu = {

width: 400,

height: 600,

title: "My menu"

};

P.S. Для проверки на число используйте функцию:

function isNumeric(n) {

return !isNaN(parseFloat(n)) && isFinite(n)

}

**Создайте калькулятор для введённых значений**

Напишите код, который:

Запрашивает по очереди значения при помощи prompt и сохраняет их в массиве.

Заканчивает ввод, как только посетитель введёт пустую строку, не число или нажмёт «Отмена».

При этом ноль 0 не должен заканчивать ввод, это разрешённое число.

Выводит сумму всех значений массива

**Поиск в массиве**

Создайте функцию find(arr, value), которая ищет в массиве arr значение value и возвращает его номер, если найдено, или -1, если не найдено.

Например:

arr = ["test", 2, 1.5, false];

find(arr, "test"); // 0

find(arr, 2); // 1

find(arr, 1.5); // 2

find(arr, 0); // -1

Открыть песочницу с тестами для задачи.

**Фильтр диапазона**

Создайте функцию filterRange(arr, a, b), которая принимает массив чисел arr и возвращает новый массив, который содержит только числа из arr из диапазона от a до b. То есть, проверка имеет вид a ≤ arr[i] ≤ b. Функция не должна менять arr.

Пример работы:

var arr = [5, 4, 3, 8, 0];

var filtered = filterRange(arr, 3, 5);

// теперь filtered = [5, 4, 3]

// arr не изменился

Открыть песочницу с тестами для задачи.

**Решето Эратосфена**

Целое число, большее 1, называется простым, если оно не делится нацело ни на какое другое, кроме себя и 1.

Древний алгоритм «Решето Эратосфена» для поиска всех простых чисел до n выглядит так:

Создать список последовательных чисел от 2 до n: 2, 3, 4, ..., n.

Пусть p=2, это первое простое число.

Зачеркнуть все последующие числа в списке с разницей в p, т.е. 2\*p, 3\*p, 4\*p и т.д. В случае p=2 это будут 4,6,8....

Поменять значение p на первое не зачеркнутое число после p.

Повторить шаги 3-4 пока p2 < n.

Все оставшиеся не зачеркнутыми числа – простые.

Посмотрите также анимацию алгоритма.

Реализуйте «Решето Эратосфена» в JavaScript, используя массив.

Найдите все простые числа до 100 и выведите их сумму.

**Подмассив наибольшей суммы**

На входе массив чисел, например: arr = [1, -2, 3, 4, -9, 6].

Задача – найти непрерывный подмассив arr, сумма элементов которого максимальна.

Ваша функция должна возвращать только эту сумму.

Например:

getMaxSubSum([-1, 2, 3, -9]) = 5 (сумма выделенных)

getMaxSubSum([2, -1, 2, 3, -9]) = 6

getMaxSubSum([-1, 2, 3, -9, 11]) = 11

getMaxSubSum([-2, -1, 1, 2]) = 3

getMaxSubSum([100, -9, 2, -3, 5]) = 100

getMaxSubSum([1, 2, 3]) = 6 (неотрицательные - берем всех)

Если все элементы отрицательные, то не берём ни одного элемента и считаем сумму равной нулю:

getMaxSubSum([-1, -2, -3]) = 0