

Массивы

№1 (*)

С помощью цикла найти в массиве наименьшее число, удовлетворяющее условиям:

1. 3-х значное
2. делящееся без остатка на 7 или на 3

Исходный массив:

```
const array = [437, 1164, 1119, 517, 367, 975, 586, 663, 131, 60, 492, 338, 54, 386, 966, 322, 503, 1048, 738, 1232, 1263, 198, 1245, 732, 128, 1067, 219, 367, 758, 1159, 1270, 1099, 712, 778, 306, 828, 396, 106, 1000, 1259, 241, 668, 482, 57, 1286, 461, 733, 1205, 963, 165]
```

№2 (*)

С помощью цикла найти сумму всех чисел удовлетворяющих:

1. находится в диапазоне [-10; -3] или [1; 25]
2. больше, чем предыдущий элемент массива

Исходный массив:

```
const array = [46, -45, 26, -17, -32, -19, -43, -18, 16, -28, 17, -40, 42, -32, -36, -39, 35, 43, -34, 47, 9, -23, -26, 0, 19, 48, -3, -38, -26, -29, -1, -10, 24, -36, 10, 42, -34, 24, 2, 33, -13, -42, -2, 40, 38, 11, -46, -13, -41, 1]
```

№3 (*)

Из исходного массива пока это возможно забирать элементы из начала и конца и добавлять их в два других массива по следующим правилам:

1. положительные в начало массива positiveArray
2. отрицательные в конец массива negativeArray

Исходный массив:

```
const array = [46, -45, 26, -17, -32, -19, -43, -18, 16, -28, 17, -40, 42, -32, -36, -39, 35, 43, -34, 47, 9, -23, -26, 0, 19, 48, -3, -38, -26, -29, -1, -10, 24, -36, 10, 42, -34, 24, 2, 33, -13, -42, -2, 40, 38, 11, -46, -13, -41, 1]
```

№4 (*)

Запросите у пользователя 3 числа и выполните с этими числами:

1. если меньше 0-го элемента исходного массива, то добавить в начало исходного массива
2. если больше последнего элемента исходного массива, то добавить в конец исходного массива
3. иначе ничего не делать

Исходный массив:

```
const array = [2, 3, 4]
```

№5 (*)

Разработайте функцию для фильтра массива который делает из исходного массива конечный массив.

Исходный массив:

```
const array = [46, -45, 26, -17, -32, -19, -43, -18, 16, -28, 17, -40, 42, -32, -36, -39, 35, 43, -34, 47, 9, -23, -26, 0, 19, 48, -3, -38, -26, -29, -1, -10, 24, -36, 10, 42, -34, 24, 2, 33, -13, -42, -2, 40, 38, 11, -46, -13, -41, 1]
```

Конечный массив:

```
resultArray = [9, 24, 24]
```

№6 (*)

Запросите у пользователя число. В консоль выведите индекс элемента в исходном массиве. (попробуйте реализовать с помощью `findIndex` и `indexOf`)

Исходный массив:

```
const array = [10, 53, 52, 79, 38, 80, 33, 1, 55, 45, 94, 31, 54, 83, 47, 62, 62, 8, 98, 36]
```

№7 (*)

Даны два массива одинаковой длины (`array1`, `array2`). Создать массив `arrayIs` такой же длины из элементов `true/false` по следующему правилу:

1. если на позиции `N` в массивах `array1` и `array2` один и тот же элемент, то на позиции `N` массива `arrayIs` стоит `true`
2. иначе `false`

Исходные массивы:

```
const array1 = [3, 5, 5, 2, 4, 4, 5, 3, 3, 5, 5, 1, 1, 2, 1, 1, 5, 1, 3, 3]
```

```
const array2 = [3, 4, 4, 5, 2, 2, 4, 3, 2, 3, 1, 5, 2, 1, 2, 5, 5, 4, 1, 3]
```

№8 (*)

Переписать следующий код с помощью `concat` и `reverse`.

```
const array1 = [1, 2, 3, 4, 5]
const array2 = [6, 7, 8, 9, 10]

const resultArray = []

for (let i = 0; i < array1.length; i++) {
  resultArray.push(array1[array1.length - i - 1])
}

array2.forEach(function (element) {
  resultArray.push(element)
})
```

№9 (*)

Если все элемента исходного массива подходят под условия:

1. по модулю больше 20
2. по модулю меньше 100

, - то вывести сообщение "Массив подходит под условие."

Если хотя бы 1 элемент исходного массива подходит под условия выше, то вывести "Есть элементы массива в указанном диапазоне."

Иначе вывести "Совпадений нет".

Примечание: использовать `some`, `every` методы массива.

Исходный массив:

```
const array = [-26, 49, 55, -62, -76, -64, 77, 51, -68, 24]
```

№10(*)

С помощью методов `filter`, `forEach` и `reduce` выполнить последовательную цепочку действия над массивом:

1. удвоить все элементы массива
2. избавиться от отрицательных чисел
3. получить произведение всех чисел

Исходный массив:

```
const array = [46, -45, 26, -17, -32, -19, -43, -18, 16, -28, 17, -40, 42, -32, -36, -39, 35, 43, -34, 47, 9, -23, -26, 0, 19, 48, -3, -38, -26, -29, -1, -10, 24, -36, 10, 42, -34, 24, 2, 33, -13, -42, -2, 40, 38, 11, -46, -13, -41, 1]
```

№11 (**)

Написать функцию `slice` принимающую 3 аргумента:

1. `array` - массив
2. `begin` - число
3. `end` - число

, - и выполняющую тоже самое, что и метод `slice`.

№12 (***)

С помощью метода `splice` заменить самую длинную восходящую подпоследовательность массива на ее реверсивную версию.

Исходный массив:

```
const array = [25, 54, 17, 12, 9, 2, 31, 97, 0, 93, 40, 70, 33, 26, 29, 96, 30, 23, 26, 98, 55, 22, 34, 48, 48, 39, 78, 98, 28, 33, 17, 82, 89, 60, 74, 78, 52, 42, 32, 12, 12, 91, 20, 76, 12, 66, 87, 77, 85, 13, 53, 92, 44, 27, 2, 79, 17, 22, 22, 62, 81, 89, 58, 94, 97, 92, 5, 50, 15, 40, 7, 43, 48, 95, 42, 42, 79, 86, 24, 43, 56, 61, 71, 61, 42, 38, 96, 13, 82, 18, 1, 3, 99, 7, 6, 72, 13, 21, 46, 64]
```

Объекты

Задания 13-16 можно выполнить в одном скрипте. Ответы на вопросы “почему” поясните в комментариях кода.

№13 (*)

Создайте объект `person`, которую вы будите использовать в следующих заданиях этой домашней работы. Включите в объект следующие поля:

1. Имя
2. Фамилию
3. Пол (`gender`)
4. Город проживания
5. Любые другие поля какие захотите

Выведите приветствия в `alert` с использованием шаблонных строк и объекта вашей персоны.

№14 (*)

Создайте объект `car` любого на ваш выбор автомобиля. Самостоятельно выберите и заполните различные поля для это объекта. Присвойте полю `owner` объекта `car` объект `person`. Присвойте полю `car` объекта `person` объект `car`.

Почему `person.car === car` ?

Почему `person === car.owner` ?

Почему `person === person.car.owner` ?

№15 (*)

Создайте объект `user` как копию объекта `person`. Для этого используйте `Object.keys` и цикл `for-of`.

Почему `person.car === user.car` ?

Доработайте скрипт так чтобы поле `car` было и в объекте `person` и в объекте `user`, но чтобы

1. `car !== person.car`
2. `car !== user.car`
3. `person.car !== user.car`

№16 (*)

Создайте объект `building`. Самостоятельно выберите и заполните различные поля для это объекта.

Как лучше всего связать объект `building` и объект `person`?

Как лучше всего связать объект `building` и объект `car`?

Удалите все ссылки на объекты, кроме ссылки на объект `person`.

Строки

№17 (*)

В следующей строке замените все вхождения подстроки “100 рублей”, на подстроку “99 центов”. Используйте для этого while, includes и replace.

Исходная строка:

```
let string = 'Когда у меня было 100 рублей, то курочка стоила 100 рублей. 100 рублей не  
большие деньги, но на 100 рублей можно было прожить целый месяц. Вот бы эти 100 рублей  
иметь сейчас.'
```

№18 (*)

Превратите строку в массив чисел и найдите ее сумму. Выведите строкой сумму элементов получившегося массива.

Исходная строка:

```
const string = '1, 15, 44, -20'
```

№19 (**)

Заполнить массив слов словами из строки с большой буквы.

Исходная строка:

```
const string = 'привет Этот славный дивный Мир! Я так по Тебе скучал. Когда же мы С  
тобой виделись в Последний раз?'
```

№20 (**)

Написать функцию yodaSay, которая принимает на вход 1 аргумент: строку, - и возвращает новую строку с измененным порядком слов. Учитывать капитализацию.

Пример:

```
yodaSay('Здравствуй мой дорогой друг') // 'Мой здравствуй друг дорогой'
```