

Лабораторная Работа №2

Выполнил: Шардт М.А., гр. ИВТ-1.1

Комплект 1: Массивы

1.1: Напишите функцию, которая заполняет новый массив предоставленным значением, например '1', и распечатывает его через console.log().

```
function fillArray(value, count) {  
  let result = []  
  for (let i = 0; i < count; i++) result.push(value)  
  console.log(result)  
}
```

1.2: Напишите сами функцию, которая разворачивает массив в обратном порядке. Не используйте array.reverse().

```
// Чистая функция  
function reverse(array) {  
  let result = []  
  for (const val of array) result.unshift(val)  
  return result  
}
```

```
// Разворот на месте  
function reverseInPlace(array) {  
  for (let i = 0; i < array.length / 2; i++) {  
    const rightIdx = array.length - 1 - i  
    // Меняем правый и левый элементы местами  
    ;[array[i], array[rightIdx]] = [array[rightIdx], array[i]]  
  }  
}
```

1.3: Напишите функцию, возвращает новый массив без предоставленных значений. Используйте примитивные типы

```
// Изначальный массив и значения, которые нужно отфильтровать  
function filter(array, ...args) {  
  let result = []  
  for (const val of array) {  
    if (!args.includes(val)) {
```

```

        result.push(val)
    }
}
return result
}

```

```

// Такой же функционал можно получить с помощью функции filter()
function filter(array, ...args) {
    return array.filter((e) => !args.includes(e))
}

```

1.4: Напишите функцию, которая преобразует глубокий массив (вложенные массивы) в одномерный (сделайте плоский массив, flat array).

```

function flatten(array) {
    let result = []
    for (const val of array) {
        Array.isArray(val)
            ? // Если значение - массив, то вызываем flatten() на нем и
              // расширяем его значения
              result.push(...flatten(val))
            : // В остальных случаях помещаем значение в конец массива
              result.push(val)
    }
    return result
}

```

```

// Такой же функционал можно получить с помощью функции flat() с аргументом
Infinity
function flatten(array) {
    return array.flat(Infinity)
}

```

1.5: Используйте деструктуризацию двумерного массива-матрицы в цикле (for (...)of). Распечатайте матрицу по отдельным строкам.

```

function logMatrix(matrix) {
    for (const row of matrix) {
        console.log(...row)
    }
}

```

1.6: Упражнение 6-6 из курсового учебника по JavaScript. Дан массив IP-адресов пользователей, которые посещали сайт. Необходимо создать частотную таблицу (массив, в котором нет повторяющихся адресов и напротив каждого адреса указана частота посещения сайта пользователем с этого адреса), после чего отсортировать по убыванию частот так, чтобы в верхней части таблицы были самые активные посетители сайта. Отдельно ответьте на вопросы:

1. Сколько всего различных адресов присутствует в массиве?
2. Со скольких адресов сайт посещался по одному разу?
3. Какова максимальная частота посещения?

```
// Изначальные данные
const data = [...]
```

// Создаем хеш-мап со значениями вида <str, number>, где str - IP, а number - количество посещений.

```
const map = new Map()
for (const { ip } of data) {
  // Получаем значение IP. Если полученное значение - undefined (т.е.
  // встречается в первый раз), то устанавливаем значение 1; в противном случае
  // повышаем значение на 1.
  map.set(ip, (map.get(ip) ?? 0) + 1)
}
```

// Конвертируем в массив вида [str, number][] и сортируем по количеству посещений в убывающем порядке.

```
const freq = Array.from(map).sort((a, b) => b[1] - a[1])
// Количество уникальных IP - это длина массива.
const unique = freq.length
// Считаем количество IP, которые встретились одинажды.
const onlyOnce = freq.reduce((pre, cur) => (cur[1] == 1 ? pre + 1 : pre), 0)
// Находим максимальное количество посещений.
const max = freq.reduce((max, cur) => (cur[1] > max ? cur[1] : max), 0)
```

Результаты

Смысл	Переменная	Значение
Количество различных адресов	unique	65
Количество адресов, с которых сайт посещался одинажды	onlyOnce	43
Максимальная частота посещения	max	12