

---

кафедра

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

---

должность, уч. степень, звание

---

подпись, дата

---

инициалы, фамилия

## ОТЧЕТ О ПРАКТИЧЕСКОМ ЗАДАНИИ №2

«Область видимости, асинхронное программирование и работа с данными по  
сети»  
по курсу: ИТ-модуль «JavaScript, его библиотеки и фреймворки в Frontend-  
разработке»

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

---

подпись, дата

---

инициалы, фамилия

Санкт-Петербург 2024

## Цель работы:

формирование практических навыков работы с асинхронным кодом в языке JavaScript, используя callback-функции и объект Promise.

## Задание:

1. Напишите функцию `filterArray`, которая принимает массив чисел и callback функцию. Функция `filterArray` должна вызывать callback функцию для каждого элемента массива и возвращать новый массив, содержащий только те элементы, для которых callback функция вернула `true`. Напишите 2 примера применения этой функции. К примеру, для фильтрации четных и нечетных значений массива.
2. Напишите асинхронную функцию `fetchData`, которая принимает URL в качестве параметра и возвращает Promise. Функция должна использовать `fetch` для получения данных с указанного URL. Если запрос прошел успешно (статус ответа 200), Promise должен быть разрешен с полученными данными в виде строки. Если запрос не удался (любой другой статус), Promise должен быть отклонен с сообщением об ошибке.

## Выполнение задания.

1 пункт:

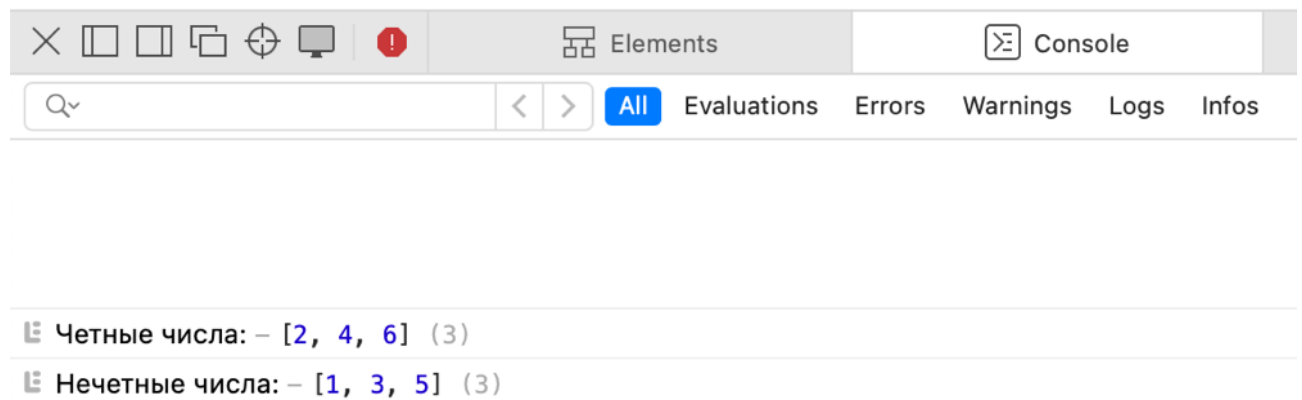


Рисунок 1

На рисунке 1 показан вывод в консоль результатов фильтрации массива от 1 до 6 на четные и нечетные значения.

2 пункт:

Как пример используется JSONPlaceholder, который является бесплатным открытым фейковым онлайн REST API для тестирования и прототипирования.



Рисунок 2

На рисунке 2 изображено, что именно возвращает URL.

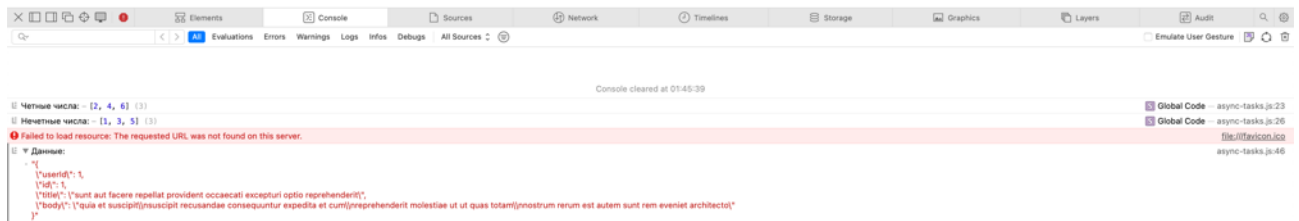


Рисунок 3

На рисунке 3 изображен пример вывода данных с Рисунка 2 в консоль.

## Коды веб-страниц:

### index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Тестирование JavaScript</title>
</head>
<body>
  <script src="async-tasks.js"></script>
</body>
</html>
```

### async-tasks.js:

```
// Задание 1: Функция filterArray
function filterArray(array, callback) {
  const result = [];
  for (let i = 0; i < array.length; i++) {
    if (callback(array[i])) {
      result.push(array[i]);
    }
  }
  return result;
}
```

```
function isEven(number) {
```

```

    return number % 2 === 0;
}

function isOdd(number) {
    return number % 2 !== 0;
}

// Использование filterArray
const numbers = [1, 2, 3, 4, 5, 6];
const evenNumbers = filterArray(numbers, isEven);
console.log('Четные числа:', evenNumbers);

const oddNumbers = filterArray(numbers, isOdd);
console.log('Нечетные числа:', oddNumbers);

// Задание 2: Асинхронная функция fetchData
function fetchData(url) {
    return new Promise((resolve, reject) => {
        fetch(url)
            .then(response => {
                if (response.ok) {
                    return response.text();
                } else {
                    throw new Error('Ошибка запроса: ' + response.status);
                }
            })
            .then(data => resolve(data))
            .catch(error => reject(error));
    });
}

// Использование открытого тестового API для получения данных
fetchData('https://jsonplaceholder.typicode.com/posts/1')
    .then(data => console.log('Данные:', data))
    .catch(error => console.error('Ошибка:', error));

```

### **Вывод.**

В ходе выполнения данной работы я успешно сформировал практические навыки работы с асинхронным кодом в JavaScript. Реализация callback-функций в функции filterArray и использование объекта Promise в асинхронной функции fetchData позволили мне понять основные принципы асинхронного программирования и обработки данных в современном JavaScript. Это опыт углубил мое понимание важных концепций, таких как асинхронность, обработка ошибок и работа с внешними API.