

# JavaScript про ECMAScript (семинары)

## Урок 1. Функциональный JavaScript

### Задание №1

#### Нахождение минимального числа в массиве

Дан массив `const arr = [1, 5, 7, 9]`. Используя метод `Math.min` и оператор распространения (spread operator), найдите минимальное число в массиве. Решение должно быть написано в одной строке.

**Подсказка:** Используйте оператор распространения Spread (...) для преобразования массива в аргументы функции `Math.min`. Это позволяет найти минимальное значение в массиве в одной строке.

#### Эталонное решение:

```
const arr = [1, 5, 7, 9];
const minNumber = Math.min(...arr);
console.log(minNumber); // Ожидаемый вывод: 1
```

### Задание №2

#### Создание счетчика

Напишите функцию `createCounter`, которая создает счетчик и возвращает объект с тремя методами: `increment`, `decrement` и `getValue`. Метод `increment` должен увеличивать значение счетчика на 1, метод `decrement` должен уменьшать значение счетчика на 1, а метод `getValue` должен возвращать текущее значение счетчика. Значение счетчика должно быть доступно только через методы объекта, а не напрямую.

**Подсказка:** Функция `createCounter` возвращает объект с методами для увеличения и уменьшения счетчика. Приватная переменная `count` доступна только через методы объекта, что обеспечивает инкапсуляцию данных.

#### Пример использования

```
const counter = createCounter();
console.log(counter.increment()); // Ожидаемый вывод: 1
console.log(counter.increment()); // Ожидаемый вывод: 2
console.log(counter.decrement()); // Ожидаемый вывод: 1
console.log(counter.getValue()); // Ожидаемый вывод: 1
```

**Эталонное решение:**

```
function createCounter() {  
    let count = 0; // Приватное свойство  
  
    return {  
        increment() {  
            count++;  
            return count;  
        },  
        decrement() {  
            count--;  
            return count;  
        },  
        getValue() {  
            return count; // Для получения текущего значения счетчика  
            (если нужно)  
        }  
    };  
}
```

**Задание №3****Рекурсивный поиск элемента по классу**

Напишите рекурсивную функцию `findElementByClass`, которая принимает корневой элемент дерева DOM и название класса в качестве аргументов и возвращает первый найденный элемент с указанным классом в этом дереве.

**Подсказка:** Функция `findElementByClass` рекурсивно обходит дерево DOM, начиная с корневого элемента. Она проверяет каждый элемент на наличие указанного класса и, если находит совпадение, возвращает его. Если элемент с нужным классом не найден, функция возвращает `null`.

**Пример использования**

```
const rootElement = document.getElementById('root');  
const targetElement = findElementByClass(rootElement, 'my-class');  
console.log(targetElement); // Ожидаемый вывод: элемент с классом  
'my-class', если найден
```

**Эталонное решение:**

```
function findElementByClass(root, className) {
```

```
if (root.classList.contains(className)) {  
    return root;  
}  
  
for (let i = 0; i < root.children.length; i++) {  
    const found = findElementByClass(root.children[i], className);  
    if (found) {  
        return found;  
    }  
}  
  
return null; // Если элемент с таким классом не найден  
}
```