

# Arrays

**Массивы - создание,  
особенности, итерация,  
методы**

# Урок

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18

# Введение

Для хранения упорядоченных коллекций существует особая структура данных, которая называется массив, `Array`.

Массив - это лаконичный способ хранения списка элементов под одним именем.

Массив – это особый подвид объектов

# Введение

```
const names = ['Bob', 'John', 'Donald', 'Stacy', 'Jack']
```



# Создание массива

```
const array = new Array();  
const arr = [];
```

# Особенности работы с массивом

```
const arr = [0, 'cat', null, {}, []]
```

В массиве могут храниться любые типы данных

```
arr[3]
```

Доступ к отдельным элементам в массиве, можем получить используя квадратные скобки

```
arr[2] = 99
```

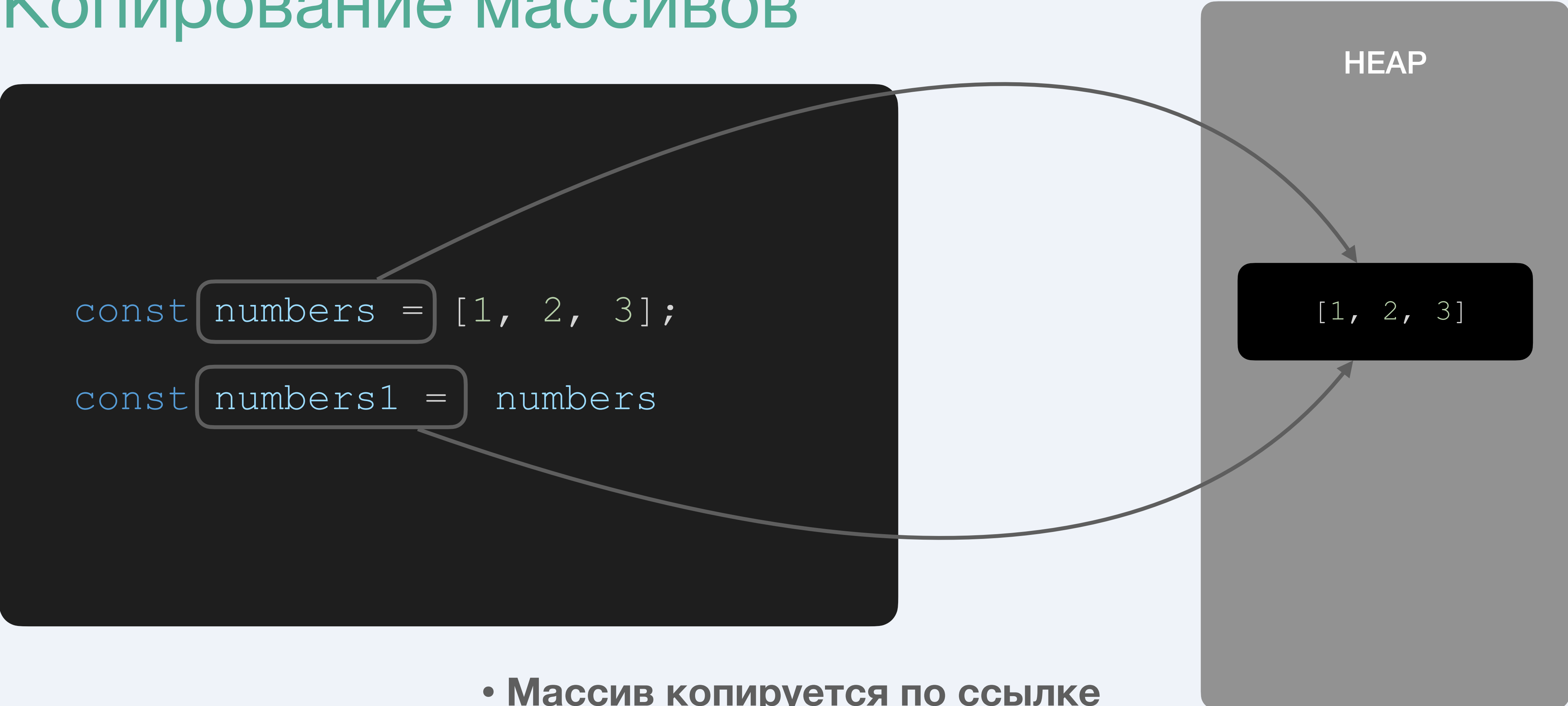
Можно изменять отдельные элементы массива

```
arr.length
```

Вы можете найти длину массива (количество элементов в нём) используя свойство `length`

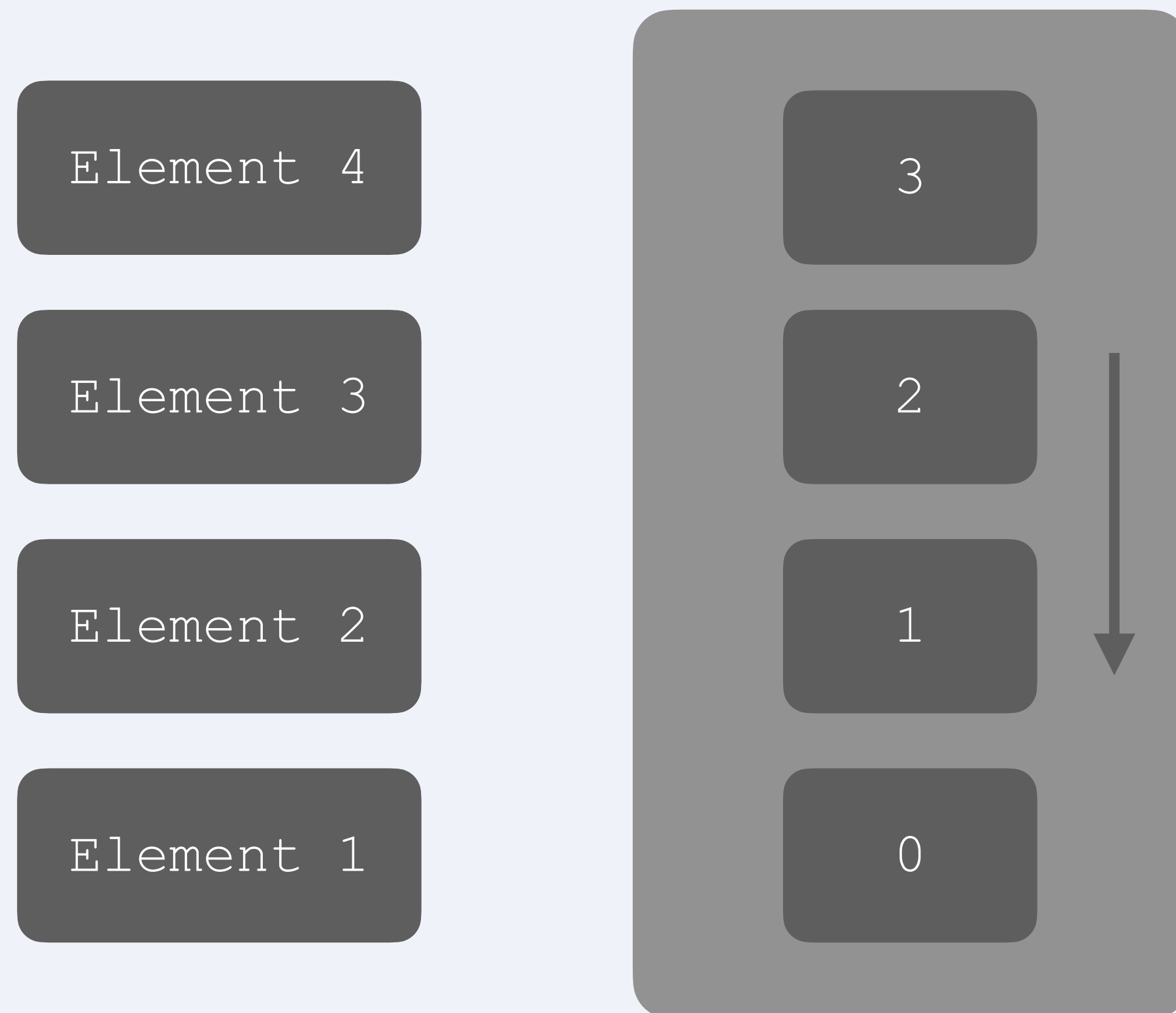
# Копирование массивов

```
const numbers = [1, 2, 3];  
const numbers1 = numbers
```



- Массив копируется по ссылке

# Очередь



- **Отсчет элементов массива начинается с 0**



# Перебор элементов массива

Цикл for

```
const numbers = [1, 2, 3]
```

```
for(let i = 0; i < numbers.length; i++) {  
  console.log(numbers[i]);  
}
```

Цикл for of

```
for(let number of numbers) {  
  console.log(number);  
}
```

# Добавление и удаление элементов массива

```
const emoji = ['🐼', '🐼', '🐼']
```

```
emoji.push('🐰') // ['🐼', '🐼', '🐼', '🐰']
```

Добавляет элемент в конец массива

```
emoji.pop() // ['🐼', '🐼', '🐼']
```

Удаляет последний элемент из массива и возвращает его

```
emoji.unshift('🐰') // ['🐰', '🐼', '🐼', '🐼']
```

Добавляет элемент в начало массива

```
emoji.shift() // ['🐼', '🐼', '🐼']
```

Удаляет из массива первый элемент и возвращает его

# Объединение массивов

```
const pandas = ['🐼', '🐼', '🐼']  
const rabbits = ['🐰', '🐰', '🐰']  
  
const animals = pandas.concat(rabbits)  
  
// ['🐼', '🐼', '🐼', '🐰', '🐰', '🐰']
```

Метод `concat()` возвращает новый массив, состоящий из массива, на котором он был вызван, соединённого с другими массивами и/или значениями, переданными в качестве аргументов.

# Методы массивов

```
const animals = ['🐼', '🐵', '🐣']  
  
animals.indexOf('🐣') // 2  
  
animals.reverse() // ['🐣', '🐵', '🐼']
```

Метод `indexOf()` возвращает первый индекс, по которому данный элемент может быть найден в массиве или `-1`, если такого индекса нет.

Метод `reverse()` на месте обращает порядок следования элементов массива. Первый элемент массива становится последним, а последний — первым.

# Методы массивов

```
const animals = ['🐼', '🐵', '🐣']  
  
const slicedAnimals = animals.slice(0, 2) // ['🐼', '🐵']  
  
animals.splice(0, 1) // ['🐵', '🐣']
```

Метод `slice()` возвращает новый массив, содержащий копию части исходного массива.

Метод `splice()` изменяет содержимое массива, удаляя существующие элементы и/или добавляя новые.

# Методы массивов

```
const animals = ['🐼', '🐵', '🐣']  
  
animals.includes('🦁')           // false  
  
const foo = animals.join('🍌'); // '🐼🍌🐵🍌🐣'
```

Метод `includes()` определяет, содержит ли массив определённый элемент, возвращая в зависимости от этого `true` или `false`.

Метод `join()` объединяет все элементы массива в строку.