

JS: Array Methods

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

Повторим

■ Как получить доступ к значению свойства объекта?

■ Как удалить свойство из объекта?

■ Как проверить есть ли некоторое свойство в объекте?

■ Когда требуется применение брекет-синтаксиса?

■ Какие типы относятся к ссылочным (reference type)?

ЦЕЛЬ

Изучить методы массивов

Map

Метод `map()` создает новый массив с результатами вызова предоставленной функции для каждого элемента в массиве.

```
const numbers = [1, 2, 3, 4, 5];  
const squaredNumbers = numbers.map(num => num ** 2);  
// squaredNumbers: [1, 4, 9, 16, 25]
```

ForEach

Метод `forEach()` выполняет предоставленную функцию один раз для каждого элемента массива.

Возвращаемое значение:
undefined

```
const numbers = [1, 2, 3, 4, 5];  
numbers.forEach(num => console.log(num));  
  
// Output: 1, 2, 3, 4, 5
```

Filter

Метод `filter()` создает новый массив с элементами, для которых предоставленная функция возвращает `true`

```
const students = [
  { name: "Alice", grade: 85 },
  { name: "Bob", grade: 90 },
  { name: "Charlie", grade: 78 },
];

const highGrades = students.filter((student) => student.grade
> 80);

// highGrades: [{ name: 'Alice', grade: 85 }, { name: 'Bob',
grade: 90 }]
```

Reduce

Метод `reduce()` применяет функцию к аккумулятору и каждому значению массива (слева направо), сводя массив к одному значению.

```
const numbers = [1, 2, 3, 4, 5];  
const sum = numbers.reduce((acc, num) => acc + num, 0);  
// sum: 15
```


Reduce

```
[0, 1, 2, 3, 4].reduce(function (previousValue, currentValue, index, array) {  
    return previousValue + currentValue;  
});
```

	previousValue	currentValue	index	array	возвращаемое значение
первый вызов	0	1	1	[0, 1, 2, 3, 4]	1
второй вызов	1	2	2	[0, 1, 2, 3, 4]	3
третий вызов	3	3	3	[0, 1, 2, 3, 4]	6
четвёртый вызов	6	4	4	[0, 1, 2, 3, 4]	10

Значение, возвращённое методом `reduce()` будет равным последнему результату выполнения колбэк-функции — **10**.

reverse

Описание:

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

Синтаксис:

```
array.reverse()
```

```
var myArray = ["один", "два", "три"];
```

```
myArray.reverse();
```

```
console.log(myArray); // ['три', 'два', 'один']
```

join

Описание:

Метод `join()` преобразует все элементы массива в строки и объединяет их в одну большую строку.

Синтаксис:

`arr.join(separator)`

```
1 let arr = ['Вася', 'Петя', 'Маша'];
2
3 let str = arr.join(';'); // объединить массив в строку через ;
4
5 alert( str ); // Вася;Петя;Маша
```



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH