

# ARRAY

# ИМЯ ФАМИЛИЯ



## Должность

- Опыт
- Дополнительная информация

## Контакты

- Корпоративная почта
- Социальные сети

<https://www.instagram.com/aittr.de/>



# 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)

# ЦЕЛЬ

**Изучить особенности и синтаксис работы с массивами в js. Изучить методы push, pull, shift, unshift. Изучить методы строк**

# ПЛАН ЗАНЯТИЯ

- Массивы: доступ к элементам, длина, добавление, удаление
- Циклы
- Класс Math
- Методы строк

## Array

Массивы в JavaScript  
представляют упорядоченные  
списки элементов.

```
let numbers = [1, 2, 3, 4, 5];  
let fruits = ["apple", "orange", "banana"];
```

## Array

Доступ к элементам  
массива по индексу

```
let numbers = [1, 2, 3, 4, 5];  
let fruits = ["apple", "orange", "banana"];  
// Доступ к элементам массива  
let firstNumber = numbers[0]; // 1  
let secondFruit = fruits[1]; // "orange"
```

## Array

Доступ к элементам  
массива по индексу

*// Длина массива*

```
let length = numbers.length; // 5
```

*// Добавление элемента в конец массива*

```
fruits.push("grape");
```

*// Удаление последнего элемента*

```
let lastFruit = fruits.pop();
```



For

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

## Цикл while

```
let count = 0;
```

```
while (count < 5) {  
  console.log(count);  
  count++;  
}
```

## Методы работы со строками

Доступ к элементам  
массива по индексу

```
let text = "Hello, World!";
```

*// Длина строки*

```
let length = text.length; // 13
```

*// Преобразование в верхний/нижний регистр*

```
let upperCase = text.toUpperCase(); // "HELLO, WORLD!"
```

```
let lowerCase = text.toLowerCase(); // "hello, world!"
```

*// Получение подстроки*

```
let substring = text.substring(0, 5); // "Hello"
```

*// Поиск подстроки*

```
let indexOfWorld = text.indexOf("World"); // 7
```

## Объект Math

Объект Math предоставляет математические методы и константы.

*// Округление числа вверх*

```
let roundedUp = Math.ceil(4.2); // 5
```

*// Округление числа вниз*

```
let roundedDown = Math.floor(4.8); // 4
```

*// Возведение в степень*

```
let power = Math.pow(2, 3); // 8
```

*// Генерация случайного числа от 0 до 1*

```
let random = Math.random(); // (например, 0.738496...)
```

The background features a light gray grid. A solid red horizontal bar is at the top. A small red square is located on the left side. The lower half of the slide contains a pattern of squares in various shades of red, orange, and yellow, arranged in a non-uniform, abstract manner.

**Как вы думаете, чем отличается массив в JS от массива в JAVA?**

## Массив JAVA vs JS

### Динамическое изменение размера в JavaScript:

**JS:** Массивы в JavaScript динамически изменяют свой размер. Вы можете добавлять или удалять элементы в массиве без явного указания размера.

**Java:** В Java размер массива определяется при его создании и не может быть изменен. Если вам нужно изменить размер массива, вам придется создать новый массив

### Динамическая типизация в JavaScript:

**JS:** Переменные в JavaScript могут хранить значения разных типов, и массивы не ограничены определенным типом данных. Вы можете хранить в массиве значения различных типов, таких как числа, строки, объекты и другие массивы.

**Java:** В Java массивы строго типизированы. Это означает, что при создании массива вы должны указать тип данных, который он будет содержать, и массив будет принимать только значения этого типа.

### Интерфейс и методы:

**JS:** В JavaScript массивы предоставляют богатый набор методов для работы с данными, таких как push, pop, shift, unshift, splice и другие.

**Java:** В Java массивы предоставляют ограниченный набор методов, и большинство операций с массивами выполняются с использованием стандартных циклов.

## Методы push, pop, shift и unshift

Методы push, pop, shift и unshift предоставляют удобные способы изменения содержимого массива в JavaScript.

## Методы push, pop, shift и unshift

**push():** Добавляет один или несколько элементов в конец массива и возвращает новую длину массива.

```
let fruits = ['apple', 'banana'];  
let length = fruits.push('orange', 'pear');  
// fruits теперь ['apple', 'banana', 'orange', 'pear']  
// length теперь 4
```



## Методы push, pop, shift и unshift

**pop():** Удаляет последний элемент из массива и возвращает его. Массив укорачивается на один элемент.

```
let fruits = ['apple', 'banana', 'orange'];  
let lastFruit = fruits.pop();  
// fruits теперь ['apple', 'banana']  
// lastFruit теперь 'orange'
```

## Методы push, pop, shift и unshift

**shift():** Удаляет первый элемент из массива и возвращает его. Массив укорачивается на один элемент.

```
let fruits = ['apple', 'banana', 'orange'];  
let firstFruit = fruits.shift();  
// fruits теперь ['banana', 'orange']  
// firstFruit теперь 'apple'
```

## Методы push, pop, shift и unshift

**unshift():** Добавляет один или несколько элементов в начало массива и возвращает новую длину массива.

```
let fruits = ['banana', 'orange'];  
let length = fruits.unshift('apple', 'pear');  
// fruits теперь ['apple', 'pear', 'banana', 'orange']  
// length теперь 4
```

# ПОИГРАЕМ ;)

■ Сколько примитивных типов в джава?

■ Какие методы класс Math вы знаете?

■ Какие особенности массива в JS?


■ Чем отличается строгое и нестрогое равенство

■ Что делают методы push, shift, unshift, pop?

# ДОМАШНЕЕ ЗАДАНИЕ



СТАВИМ +, ЕСЛИ ВАМ ПОНЯТНО ДОМАШНЕЕ  
ЗАДАНИЕ



# **Ваша новая IT-профессия – Ваш новый уровень жизни**

Программирование с нуля в  
немецкой школе AIT TR GmbH