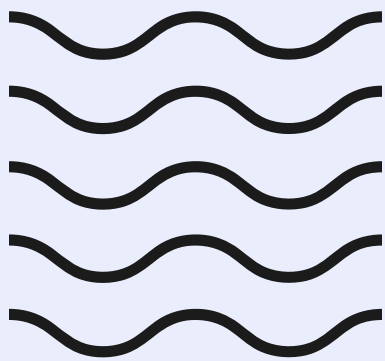


# JS



ПОДГОТОВИЛ: АЛИШЕР ХАМИДОВ



# План лекции

## Основные моменты:

1. Types
2. Operators
3. Function
4. Array





# 7 primitive types

- string "Any text" or 'Any text'
- number 129.2
- boolean true or false
- symbol Symbol('something')
- null null
- undefined undefined
- bigint 9007199254740991n

# Object { key: 'value' }

- Array [1, "text", false]
- Function function name() { }



# Математические операции



\* умножение

/ деление

- вычитание

% остаток от деления

\*\* возведение в степень





# Операторы равенства




=== равенство

== равенство с преобразованием типа(coercion)





# Преобразование типов (coercion)

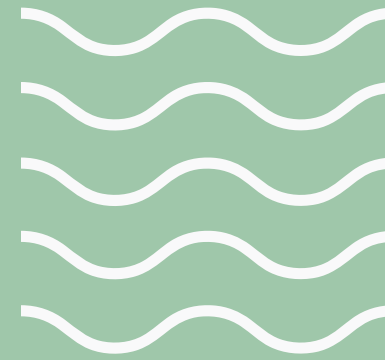
Когда сравниваются разные типы, движок JavaScript  производит попытку привести тип одного элемента к типу другого, чтобы он мог их сравнить

Приоритет преобразований:

1. String
2. Number
3. Boolean



```
2 + "7"; // "27"  
true - 5 // -4
```







# Операторы сравнения



$>=$  умножение

$<=$  деление

$!=$  неравенство

$!=$  неравенство с преобразованием






# Логические операторы



&& AND

|| OR





# Преобразование типов



&& AND

|| OR





# Function

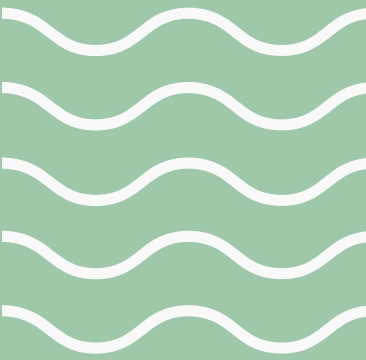
// Объявление функции



```
function makePhrase(param1, param2){  
  // код  
  const a = param1 + " love " + param2;  
  return a;  
}
```

• // Вызов функции

```
makePhrase("I", "You")
```





# Arrow Function

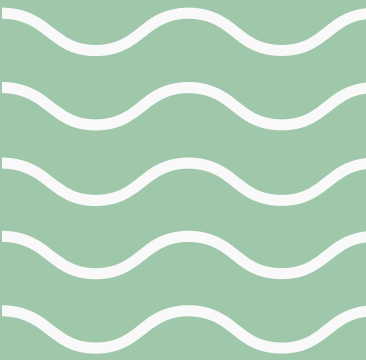


// Объявление функции (без фигурных скобок не нужен return)

```
const makePhrase = (param1, param2) => param1 + " love " + param2;  
  
}
```

// Вызов функции

- `makePhrase("I", "You")`





# Multiline Arrow Function

// Объявление функции (нужен return)



```
const makePhrase = (param1, param2) => {  
  // код  
  const a = param1 + " love " + param2;  
  return a;  
}
```

- // Вызов функции  
makePhrase("I", "You")





# Array

// Два способа создания пустого массива

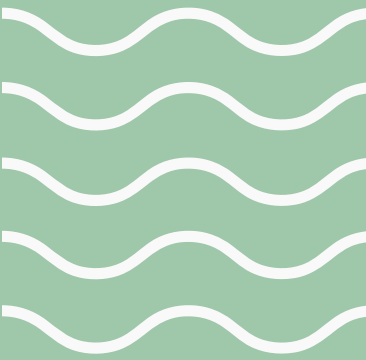
```
let arr = new Array();
```

```
let arr = [];
```

// В квадратных скобках можно сразу указать начальные значения

```
let fruits = ["Apple", "Orange", "Banana"];
```

○

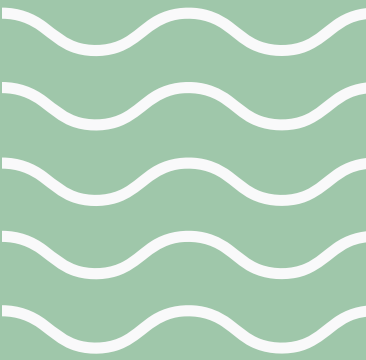




# Array



// Можно получить элемент по индексу  
`let fruits = ["Apple", "Orange", "Banana"];`  
// Получим нулевой элемент  
`console.log(fruits[0]); // Apple`



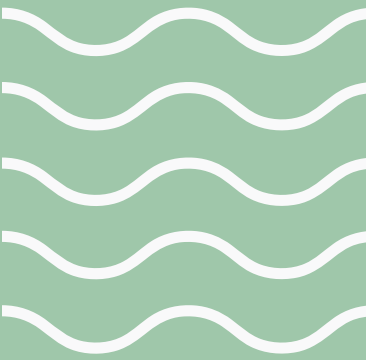




# Array



```
// Можно получить элемент по индексу  
let fruits = ["Apple", "Orange", "Banana"];  
// Можно заменить элемент по индексу  
fruits[0] = "Mango"  
console.log(fruits[0]); // ["Mango", "Orange", "Banana"];
```

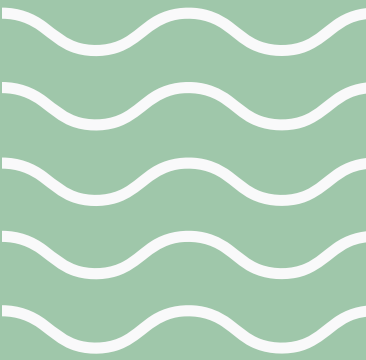




# Методы Array



- **push** - добавляет элемент в конец массива:
- **pop** - удаляет последний элемент из массива и возвращает его
- **shift** - удаляет элемент из начала массива
- **unshift** - добавляет элемент в начало массива





# Дополнительные материалы



- **Учебник по JS: <https://learn.javascript.ru/>**
- **Cheatsheet: <https://htmlcheatsheet.com/js/>**

