

## Урок 0

# Повторение JavaScript. Возможности ES6. Установка Node и npm.

---

## План урока

1. Повторение основ.
2. let и const.
3. Стрелочные функции.
4. Деструктуризация.
5. Оператор spread/rest.
6. Установка Node и npm.

## Повторение основ

В процессе освоения библиотеки React мы будем использовать классические фронтенд-технологии, с которыми вы уже работали:

- HTML
- CSS
- JavaScript

Для успешного понимания материала вам может пригодиться небольшое повторение материала и упоминание возможностей языка, о которых вы могли не знать. Будет полезно пробежаться по [статье](#).

В редакции языка под названием EcmaScript 6 (или просто ES6) было введено множество улучшений, которые сделали JavaScript удобным языком для разработки. Рассмотрим основные из них, с ними мы будем встречаться часто в процессе освоения React.

## let и const

Ключевое слово `let` позволяет объявлять переменные с ограниченной областью видимости — только для блока `{...}`, в котором происходит объявление. Это называется блочной областью видимости. К такому объявлению переменных вы привыкли, используя другие языки программирования. Вместо ключевого слова `var`, которое обеспечивает область видимости внутри функции, стандарт ES6 рекомендует использовать `let`.

Ключевое слово `const` также объявляет переменную в блоке, но не позволяет переписать идентификатор переменной.

```
let a = 10;
{
  let a = 20;
  console.log(a);
}
console.log(a);

const b = 'abcd';
b = 'another string'; // TypeError
```

## Стрелочные функции

Стрелочные функции представляют собой сокращённую запись функций в ES6. Стрелочная функция состоит из списка параметров ( `...` ), за которым следует знак `=>` и тело функции. Тело функции может не оборачиваться в фигурные скобки, если это простое выражение. В таком случае слово `return` подразумевается, делая запись ещё короче.

```
const sum = function(a, b) {
  return a + b;
}

const sumArrow = (a, b) => a + b;

const singular = ['cat', 'dog', 'programmer', 'jeans'];
const plural = singular.map((item) => {
  if (item !== 'jeans') {
    return item + 's';
  }
});
```

```
    }  
    return item;  
  });  
  console.log(plural); // [ 'cats', 'dogs', 'programmers', 'jeans' ]
```

У стрелочных функций есть еще одна интересная деталь: у них нет собственного значения **this**, как у обычных функций. Об этом мы поговорим далее в одном из уроков.

## Деструктуризация

С помощью деструктуризации можно “распаковать” элементы массива или свойства объекта в несколько переменных.

```
// с массивом  
const [first, second] = ['Hello', 'world!'];  
console.log(first, second); // Hello world!
```

```
// с объектом  
const lesson = {  
  title: 'Lesson 0',  
  name: 'Prerequisites',  
  date: '05.10.2021'  
};
```

```
const { title, name, date } = lesson;  
console.log(title, name, date); // Lesson 0 Prerequisites 05.10.2021
```

## Оператор spread/rest

Оператор ... (три точки) называют как spread, так и rest, в зависимости от того, как и где он используется.

При использовании в любом итерируемом объекте (iterable), данный оператор “разбивает” (“spread”) его на индивидуальные элементы:

```
const greetings = (name, surname) => `Greetings, ${name} ${surname}`;  
  
const date = ['John', 'Doe'];  
console.log(greetings(...date)); // Greetings, John Doe
```

В этом примере мы также рассмотрели шаблонные строки — в строки, объявленные с обратными кавычками можно вставить любые корректные JavaScript выражения с помощью `${...}`.

Другим распространенным использованием оператора `...` является объединение набора значений в один массив. В данном случае оператор работает как "rest".

```
const greetings = (...names) => `Greetings, I see ${names.length} new faces!`;
```

```
console.log(greetings('Maria', 'John', 'Pam', 'Steve'));  
// Greetings, I see 4 new faces!
```

## Установка Node и npm

Чтобы работать с React, нам необходимо установить платформу для исполнения JavaScript под названием **Node.js**, а также пакетный менеджер для установки зависимостей **npm**. Поставить обе программы сразу можно по [ссылке](#). Чтобы проверить, есть ли у вас эти программы, выполните в терминале:

