

Учебник

Курсы

Форум ES5 Тесты знаний Скринкасты •

Купить EPUB/PDF



RU

### Продвинутая работа с функциями

Навигация по уроку

У стрелочных функций нет

Стрелочные функции не имеют «arguments»

Итого

Комментарии

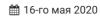
Поделиться



Редактировать на GitHub



→ Язык программирования JavaScript → Продвинутая работа с функциями





# Повторяем стрелочные функции

Давайте вернёмся к стрелочным функциям.

Стрелочные функции - это не просто «сокращение», чтобы меньше писать. У них есть ряд других полезных особенностей.

При написании JavaScript-кода часто возникают ситуации, когда нам нужно написать небольшую функцию, которая будет выполнена где-то ещё.

### Например:

- arr.forEach(func) func выполняется forEach для каждого элемента массива.
- setTimeout(func) func выполняется встроенным планировщиком.
- ...и так далее.

Это очень в духе JavaScript - создать функцию и передать её куда-нибудь.

И в таких функциях мы обычно не хотим выходить из текущего контекста. Здесь как раз и полезны стрелочные функции.

# У стрелочных функций нет «this»

Как мы помним из главы Методы объекта, "this", у стрелочных функций нет this. Если происходит обращение к this, его значение берётся снаружи.

Например, мы можем использовать это для итерации внутри метода объекта:

<

```
let group = {
2
      title: "Our Group",
3
      students: ["John", "Pete", "Alice"],
4
5
      showList() {
6
        this.students.forEach(
          student => alert(this.title + ': ' + student)
7
8
        );
9
     }
10
   };
11
12
   group.showList();
```

Здесь внутри for Each использована стрелочная функция, таким образом this.title в ней будет иметь точно такое же значение, как в методе showList: group.title.

Если бы мы использовали «обычную» функцию, была бы ошибка:

```
let group = {
2
     title: "Our Group",
     students: ["John", "Pete", "Alice"],
3
4
5
     showList() {
6
        this.students.forEach(function(student) {
7
          // Error: Cannot read property 'title' of undefine
          alert(this.title + ': ' + student)
8
9
        });
10
     }
11
   };
12
13 group.showList();
```

Раздел

### Продвинутая работа с функциями

Навигация по уроку

У стрелочных функций нет

Стрелочные функции не имеют «arguments»

Итого

Комментарии

Поделиться



Редактировать на GitHub

Ошибка возникает потому, что for Each по умолчанию выполняет функции c this, равным undefined, и в итоге мы пытаемся обратиться к undefined.title.

Это не влияет на стрелочные функции, потому что у них просто нет this.





## Стрелочные функции нельзя использовать с new

Отсутствие this естественным образом ведёт к другому ограничению: стрелочные функции не могут быть использованы как конструкторы. Они не могут быть вызваны с new.

# ① Стрелочные функции VS bind

Существует тонкая разница между стрелочной функцией => и обычной функцией, вызванной с .bind(this):

- .bind(this) создаёт «связанную версию» функции.
- Стрелка => ничего не привязывает. У функции просто нет this. При получении значения this - оно, как обычная переменная, берётся из внешнего лексического окружения.

# Стрелочные функции не имеют «arguments»

У стрелочных функций также нет переменной arguments.

Это отлично подходит для декораторов, когда нам нужно пробросить вызов стекущими this и arguments.

Haпример, defer(f, ms) принимает функцию и возвращает обёртку над ней, которая откладывает вызов на ms миллисекунд:

<

```
1 function defer(f, ms) {
     return function() {
3
       setTimeout(() => f.apply(this, arguments), ms)
4
     };
5 }
6
7 function sayHi(who) {
8
     alert('Hello, ' + who);
9 }
10
11 let sayHiDeferred = defer(sayHi, 2000);
12 sayHiDeferred("John"); // выводит "Hello, John" через 2
```

То же самое без стрелочной функции выглядело бы так:

```
1 function defer(f, ms) {
    return function(...args) {
2
3
      let ctx = this;
4
      setTimeout(function() {
5
        return f.apply(ctx, args);
6
      }, ms);
7
    };
8 }
```

Здесь мы были вынуждены создать дополнительные переменные args и ctx, чтобы функция внутри setTimeout могла получить их.

# Итого

Стрелочные функции:

- He имеют this.
- Не имеют arguments.

Раздел

# Продвинутая работа с функциями

Навигация по уроку

У стрелочных функций нет

Стрелочные функции не имеют «arguments»

Итого

Комментарии

Поделиться





Редактировать на GitHub

- Не могут быть вызваны с new.
- (У них также нет super, но мы про это не говорили. Про это будет в главе Наследование классов).



Всё это потому, что они предназначены для небольшого кода, который не имеет своего «контекста», выполняясь в текущем. И они отлично справляются с этой задачей!



Проводим курсы по JavaScript и фреймворкам.



перед тем как писать...

×

© 2007—2020 Илья Кантор | о проекте | связаться с нами | пользовательское соглашение | политика конфи



