

Функции

Автор последнего обновления: Pavel.Tokarenko@evraz.com | 14 сент. 2023 г. в 05:19 GMT+3

В проектах ДОЛЖНЫ использоваться обычные функции вместо стрелочных. Стрелочные функции не являются заменой обычным, так как несут определённую функциональность и ограничения. Например, могут быть объявлены только как функциональные выражения, что накладывает ограничение на порядок объявления.

Стрелочные функции всегда остаются стрелочными, нельзя предугадать в каком ключе будет использован модуль другим разработчиком, а значит, что стрелочные функции могут наложить ряд ограничений, либо не ожидаемые свойства в случае передачи ссылки на функцию.

```
// плохо
const useArrowFunction = () => {
  // arrow body
  ...
}

// хорошо
function useClassicFunction() {
  // function body
  ...
}
```

Стрелочные функции ДОЛЖНЫ применяться только в качестве callback ов с простыми выражениями. Если возникла необходимость добавить фигурные скобки тела стрелочной функции, ДОЛЖНА использоваться классическая функция.

```
// плохо
useSomething((a, b) => {
  if (a == null && b == null) return [];
  return [...a, ...b];
});

// хорошо
useSomething((a, b) => a - b);

useSomething(combineArrays);

function combineArrays<T>(a: T[], b: T[]): T[] {
  if (a == null, b == null) return [];
  return [...a, ...b];
}
```

Стрелочные функции НЕЛЬЗЯ использовать в качестве методов объектов. Это связано с возможностью использования ссылки на метод, вместо вызова. Особенно чревато подобное использование, если в методе используется this для доступа к другим элементам объекта, что может создать ситуацию, когда в этот this попадёт контекст другого объекта или функции.

```
// плохо
const myObject = {
  bestMethod = () => {
    this.anotherMethod();
  },
  anotherMethod = () => {}
}

useFunction(myObject.bestMethod) // опасно, this.anotherMethod() выдаст ошибку,
// либо никогда не выполнится
```