//Функция принимает объект, делает spread

*function* toIterable(*obj*) {

    return {

        //Делается spread

        ...*obj*,

        //Метод будет вызываться автоматически, когда будем проходится по объекту

        [Symbol.iterator] () {

            //Извлекаем ключи из текущего объекта. Получим массив ключей [name, age, city]

*const* keys = *Object*.keys(this);

            //Определяем кол-во элеметов в объекте, путем подсчета элементов в массиве keys

            //Т.к. все свойства объекта попали туда

*const* limit = keys.length;

            //В переменную помещаю конекст объекта?????

*const* $this = this;

            //Задаем счетчик

*let* counter = 0;

            //Возвращается объект с методом next()

            return {

                //next возвращает объект из 2х свойств

                next() {

                    //Если значение счетчика меньше кол-ва элементов в объекте

                    if(counter < limit) {

                        //Возвращается объект

                        return {

                            done: false,

                            value: $this[keys[counter++]]

                        }

                    }

                    return {

                        done: true

                    }

                }

            }

        }

    }

}

*const* person = {

    name: 'Alex',

    age: 25,

    city: 'Odessa',

}

//Вывод элементов

for(*let* value of toIterable(person)) {

    console.log(value);

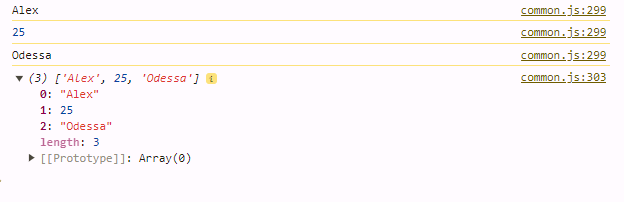
}

//Можем сделать spread obj -> array

console.log([...toIterable(person)]);

//БЕЗ ПРИМЕНЕНИЯ ФУНКЦИИ БЫЛА БЫ ОШИБКА

Результат



ТАКИМ ОБРАЗОМ МЫ СДЕЛАЛИ ОБЪЕКТ ПЕРЕБИРАЕМЫМ (FOR OF)