**Object.keys, values, entries**

map.keys(), map.values(), map.entries().

Это универсальные методы, и существует общее соглашение использовать их для структур данных. Если бы мы делали собственную структуру данных, нам также следовало бы их реализовать.

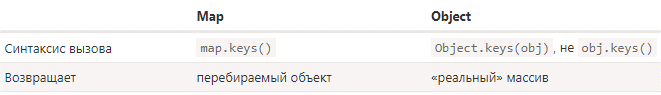
Методы поддерживаются для структур:

* Map
* Set
* Array

Для простых объектов доступны следующие методы:

* [Object.keys(obj)](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Object/keys) – возвращает массив ключей.
* [Object.values(obj)](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Object/values) – возвращает массив значений.
* [Object.entries(obj)](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Object/entries) – возвращает массив пар [ключ, значение].

различия (по сравнению с map, например):



Первое отличие в том, что мы должны вызвать Object.keys(obj), а не obj.keys().

Почему так? Основная причина – гибкость. Помните, что объекты являются основой всех сложных структур в JavaScript. У нас может быть объект data, который реализует свой собственный метод data.values(). И мы всё ещё можем применять к нему стандартный метод Object.values(data).

Второе отличие в том, что методы вида Object.\* возвращают «реальные» массивы, а не просто итерируемые объекты. Это в основном по историческим причинам.

Например:

let user = {

name: "John",

age: 30

};

* Object.keys(user) = ["name", "age"]
* Object.values(user) = ["John", 30]
* Object.entries(user) = [ ["name","John"], ["age",30]

Вот пример использования Object.values ​​для перебора значений свойств в цикле:

let user = {

name: "John",

age: 30

};

// перебор значений

for (let value of Object.values(user)) {

alert(value); // John, затем 30

}

**Object.keys/values/entries игнорируют символьные свойства**

Так же, как и цикл for..in, эти методы игнорируют свойства, использующие Symbol(...) в качестве ключей.

Обычно это удобно. Но если требуется учитывать и символьные ключи, то для этого существует отдельный метод [Object.getOwnPropertySymbols](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Object/getOwnPropertySymbols), возвращающий массив только символьных ключей. Также, существует метод [Reflect.ownKeys(obj)](https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Reflect/ownKeys), который возвращает *все* ключи.

Если мы хотели бы их применить, то можно использовать Object.entries с последующим вызовом Object.fromEntries:

1. Вызов Object.entries(obj) возвращает массив пар ключ/значение для obj.
2. На нём вызываем методы массива, например, map.
3. Используем Object.fromEntries(array) на результате, чтобы преобразовать его обратно в объект.

Например, у нас есть объект с ценами, и мы хотели бы их удвоить:

let prices = {

banana: 1,

orange: 2,

meat: 4,

};

let doublePrices = Object.fromEntries(

// преобразовать в массив, затем map, затем fromEntries обратно объект

Object.entries(prices).map(([key, value]) => [key, value \* 2])

);

alert(doublePrices.meat); // 8