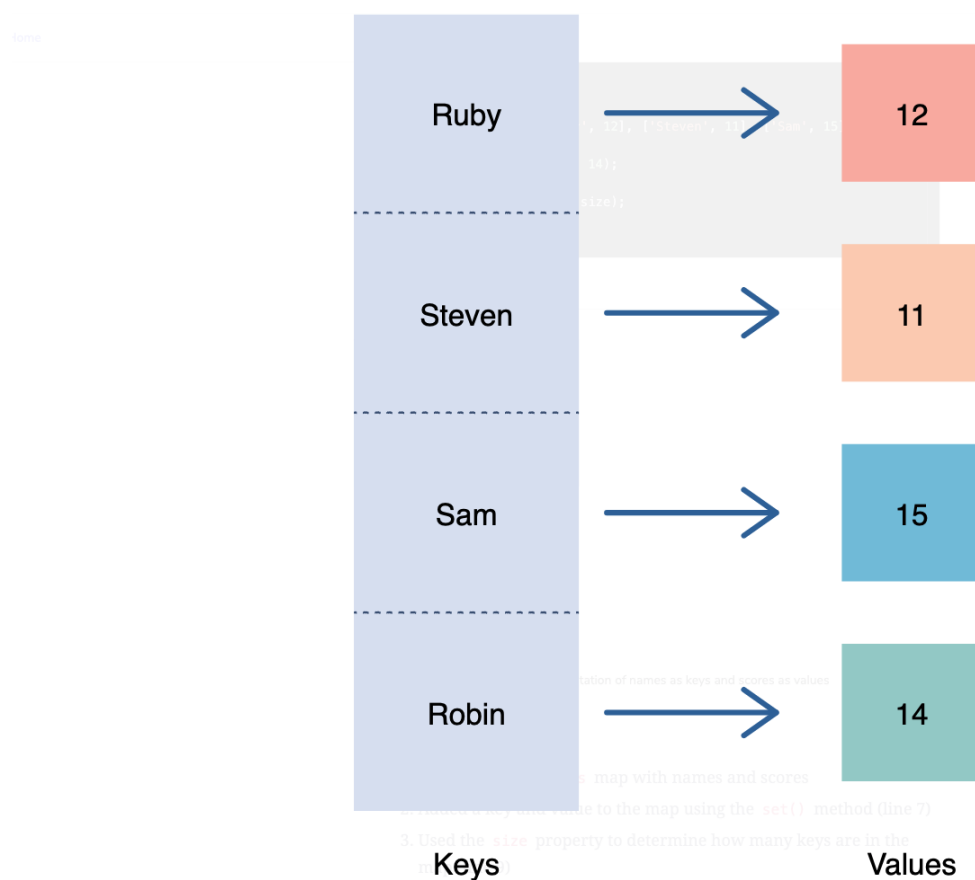


# JS Built-in object - Map



Source: <https://www.educative.io/blog/what-is-javascript-map#whatismap>

## AGENDA

1. Tổng quan về Map
2. Một số phương thức của Map
3. So sánh map vs object

## 1. Tổng quan về Map

- **What:** Map là một cấu trúc dữ liệu trong đó mỗi bộ dữ liệu được lưu và truy xuất key duy nhất.
- **Why:** truy cập nhanh, không cần duyệt qua từng items như array
- **When:** easy to access data with key and frequently adding/remove key

Trước **ES6**, thì mình dùng object

```
const cityList = [
  { id: 1, name: 'TP. Hồ Chí Minh' },
  { id: 2, name: 'TP. Phan Thiết' },
]

const studentList = [
  { id: 123, name: 'Alice', cityId: 1 },
  { id: 321, name: 'Bob', cityId: 2 },
]

// Question: how to show city name for each student?
const cityMap = cityList.reduce((map, city) => {
  map[city.id] = city;
  return map;
}, {});

cityMap[1].name; // TP. Hồ Chí Minh
cityMap[2].name; // TP. Phan Thiết
```

Còn **từ ES6 trở đi** thì mình cũng dùng object 🙄 , à không mình nhầm, còn tùy nhen mn =))) Mình sẽ so sánh map vs object ở bên dưới.

## 2. Một số phương thức của Map

| #  | Name  | Desc                             |
|----|---|----------------------------------|
| 1  | <code>new Map()</code>                      | Khởi tạo map                     |
| 2  | <b>size</b> (thuộc tính, ko phải hàm)       | Lấy size của map                 |
| 3  | <code>set(key, value)</code>                | Tạo mới / cập nhật value cho key |
| 4  | <code>get(key)</code>                       | Lấy value của key                |
| 5  | <code>has(key)</code>                       | Kiểm tra map có key không?       |
| 6  | <code>delete(key)</code>                    | Xoá một key                      |
| 7  | <code>clear()</code>                        | Xoá tất cả keys                  |
| 8  | <code>keys()</code>                         | LOOP qua tất cả keys             |
| 9  | <code>values()</code>                       | LOOP qua tất cả values           |
| 10 | <code>entries()</code>                      | LOOP qua tất cả cặp [key, value] |
| 11 | <code>forEach((value, key) =&gt; {})</code> | LOOP qua tất cả values           |

Create a new Map

```
const map = new Map();
map.size; // 0

const map = new Map([
  ['a', 1],
  ['b', 2],
  [3, 3],
]);
map.size; // 3
```

```
> const map = new Map([
  ['a', 1],
  ['b', 2],
  [3, 3],
])
< undefined

> map
< ▶ Map(3) {"a" => 1, "b" => 2, 3 => 3}
```

## Get / Set

```
const map = new Map();

map.set('a', 1);
map.set('a', 2);
map.size; // 1

map.get('a'); // 2
map.get('b'); // undefined

map.delete('b'); // false
map.delete('a'); // true

map.get('a'); // undefined
map.size; // 0
```

## Loop through Map

```
const map = new Map();

map.set('a', 1);
map.set('b', 2);
map.set('c', 3);

for (const [key, value] of map) {
  console.log(key, value)
}

map.forEach((value, key) => {
  console.log(key, value)
})
// a 1
// b 2
// c 3
```

## Show city name using Map

```
const cityList = [
  { id: 1, name: 'TP. Hồ Chí Minh' },
  { id: 2, name: 'TP. Phan Thiết' },
]

const studentList = [
  { id: 123, name: 'Alice', cityId: 1 },
  { id: 321, name: 'Bob', cityId: 2 },
]

// Question: how to show city name for each student?
// using forEach
const cityMap = new Map();
cityList.forEach(city => {
  cityMap.set(city.id, city);
})

// or using reduce
const cityMap = cityList.reduce((map, city) => {
  map.set(city.id, city);
  return map;
}, new Map());

cityMap.get(1).name; // TP. Hồ Chí Minh
cityMap.get(2).name; // TP. Phan Thiết
```

## 3. So sánh Map vs Object

|                           | Map                               | Object  |
|---------------------------|-----------------------------------|---|
| Accidental Keys           | only contains what we add         | has prototype keys, so be careful to be conflicted with your keys |
| Key Types                 | can be any type                   | string or symbol ONLY   |
| Key Order                 | keep inserted order               | better to not depend on key's order                               |
| Size                      | using <code>map.size</code> props | do it your self <code>Object.keys(obj).length</code>              |
| Iteration                 | Yes, it support to loop           | Not support, need to do it with props                             |
| Performance               | Better for frequent add/remove    | Not optimize for frequent add/remove                              |
| Serialization and Parsing | Do it yourself                    | Native support  |

Source: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Map](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map)

## Tham khảo

- <https://flaviocopes.com/javascript-data-structures-map/#what-is-a-map>
  - <https://www.educative.io/blog/what-is-javascript-map#whatismap>
  - <https://medium.com/front-end-weekly/es6-map-vs-object-what-and-when-b80621932373>
  - <https://codeburst.io/array-vs-set-vs-map-vs-object-real-time-use-cases-in-javascript-es6-47ee3295329b>
- 

## Khoá học Javascript cho người mới bắt đầu 2021 🎨

- Tác giả: **Hậu Nguyễn** - Founder Easy Frontend
- Khoá học chỉ được published trên Udemy, không thông qua trung gian.
- Khoá học không bán dạng videos upload trên Google Drive hay bất cứ hình thức nào tương tự.
- Khoá học có nhóm discord để hỗ trợ trong quá trình học tập.

📞 Liên hệ tác giả để được hỗ trợ:

- ✅ Facebook: <https://www.facebook.com/nvhauesmn/>
- ✅ Fanpage: <https://www.facebook.com/learn.easyfrontend>
- ✅ Youtube Channel: <https://www.youtube.com/easyfrontend>