

map传入数据必须是两个，第一个为key，第二个为value，字典类型，key的类型是任意的，map中数据顺序由添加时决定，据说性能更好

```
let map = new Map([["A", 1]]);
```

增

```
map.set("A", 1) // 写入数据
```

```
map.set("A", 5) // 传入相同key修改数据
```

删

```
map.delete("A") // 删除数据，传入key
```

```
map.clear("A", 5) // 清空数据
```

查

```
map.has('A'); // 是否存在某一项，传入key
```

```
console.log(s.size); // set的项数
```

```
console.log(map.get('A')); // 查找map的值，传入key
```

```
console.log(s.keys()); // set的key
```

```
console.log(s.values()); // set的value
```

```
console.log(s.entries()); // set的键值对
```

```
map.forEach((value, key) => { // 遍历map 第一个参数为value第二个参数为key
```

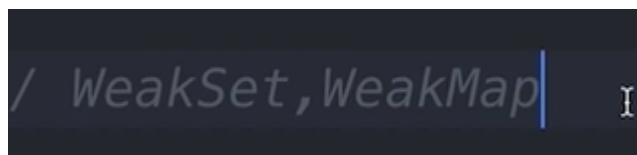
```
    console.log(key, value)
```

```
});
```

```
for(let [key, value] of map) { // 所有数据都能用for...of...遍历
```

```
    console.log(key, value);
```

```
}
```



同set, map，不同点——weakSet只接受对象，weakMap可以接受对象

其实 Object 也是按键值对存储和读取的，那么他俩之间除了我们之前说的区别以外还有其他的吗？

- 键的类型

一个 Object 的键只能是字符串或者 Symbols，但一个 Map 的键可以是任意值，包括函数、对象、基本类型。

- 键的顺序

Map 中的键值是有序的，而添加到对象中的键则不是。因此，当对它进行遍历时，Map 对象是按插入的顺序返回键值。

- 键值对的统计

你可以通过 size 属性直接获取一个 Map 的键值对个数，而 Object 的键值对个数只能手动计算。

- 键值对的遍历

Map 可直接进行迭代，而 Object 的迭代需要先获取它的键数组，然后再进行迭代。

- 性能

Map 在涉及频繁增删键值对的场景下会有些性能优势。