

## 手写 JS 深拷贝，考虑 set map 函数等

如果不考虑函数，set、map，则使用 JSON.stringify 和 JSON.parse 效率很高。

考虑函数等类型时，就不能使用 JSON.stringify，因为会函数造成丢失：

```
JSON.stringify({fn: () => {}, a: 3});  
// '{"a":3}' 函数丢失
```

```
export function deepClone(obj: any, map = new WeakMap()): any {  
  if (typeof obj !== 'object' || obj == null) {  
    return obj;  
  }  
  // 避免循环引用  
  const objFromMap = map.get(obj);  
  if (objFromMap) {  
    return objFromMap;  
  }  
  
  let target: any = {};  
  map.set(obj, target);  
  
  // Map  
  if (obj instanceof Map) {  
    target = new Map();  
    obj.forEach((v, k) => {  
      const v1 = deepClone(v, map);  
      const k1 = deepClone(k, map);  
      target.set(k1, v1);  
    });  
  }  
  
  // Set  
  if (obj instanceof Set) {  
    target = new Set();  
    obj.forEach((v) => {  
      const v1 = deepClone(v, map);  
      target.add(v1);  
    });  
  }  
  
  // Array  
  if (obj instanceof Array) {  
    target = obj.map(item => deepClone(item, map));  
  }  
  
  // Object  
  for (const key in obj) {
```

```
    const val = obj[key];  
    const val1 = deepClone(val, map);  
    target[key] = val;  
  }  
  return target;  
}
```