

1、后端拿到的数组排序之后还原

让对象属性不可修改

es6:

```
let d = new Proxy(obj, {  
  get(target, key) { // 读这个对象的操作  
    if(key == 'price') {  
      return target[key] += 100  
    } else {  
      return target[key]  
    }  
  },  
  set(target, key, value) { // 写这个对象的操作  
    return false  
  }  
})
```

es5:

```
for(let [key] of Object.entries(obj)) {  
  Object.defineProperty(obj, key, {  
    writable:false  
  })  
}
```

2、校验，拦截无效数据

```
let test=(target, key, value)=>{ // 验证模块封装  
  if(Reflect.has(target, key)) {  
    if(key == 'price') {  
      if(value>300) {  
        return false;  
      } else {  
        target[key] = value  
      }  
    } else {  
      target[key] = value  
    }  
  }  
}
```

```

    }else{
        return false;
    }
}
let d = new Proxy(obj, {
    get(target, key) {
        return target[key] || ''
    },
    set: test
})
d.price = 350
console.log(d.name+':', d.price);

```

3、监控，上报

```

window.addEventListener('error', (e)=>{
    console.log(e.message);
    report('...')
}, true)
let test=(target, key, value)=>{
    if(Reflect.has(target, key)) {
        if(key == 'price') {
            if(value>300) {
                throw new TypeError('price exceed 300')
            }else{
                target[key] = value
            }
        }else{
            target[key] = value
        }
    }else{
        return false;
    }
}

```

4、每次生产随机id，每个实例对象id不同，id不可修改

```

class component{

```

```
constructor() {  
    this.proxy = new Proxy({ // 代理这个id对象，让get()方法每次拿到的都是相  
同的id  
        id: Math.random().toString(36).slice(-8)  
    }, {})  
}  
get id() { // 如果写在此处则每次拿数据就要执行一次造成生成的随机数不同  
    return this.proxy.id  
}  
}  
  
let com1 = new component()  
let com2 = new component()  
for(let i=0; i<10; i++) {  
    console.log(com1.id, com2.id);  
}
```