

规则

1、注重命名

2、IF语句简化

3、及早返回

4、解构赋值

1、注重命名

为一个事件命名是很困难的。虽然困难但是非常有必要。

想象以下，将两个数组合并成一个数组，并生成一个唯一值的数组。那么你会怎么命名它呢？我们也许会这样命名？

```
1 function mergeNumberListIntoUniqueList(listOne, listTwo) {  
2   return [...new Set([...listOne, ...listTwo])]  
3 }
```

上面的命名并不糟糕，但是还不是很友好。你可以将一个功能函数拆分为两个功能函数，这样命名更友好和函数复用性更好。

```
1 function mergeLists(listOne, listTwo) {  
2   return [...listOne, ...listTwo]  
3 }  
4  
5 function createUniqueList(list) {  
6   return [...new Set(list)]  
7 }  
8
```

2、IF语句简化

假设我们有下面的代码：

```
1 if(value === 'duck' || value === 'dog' || value === 'cat') {  
2   // ...  
3 }
```

我们可以这样解决：

```
1 const options = ['duck', 'dog', 'cat'];  
2 if (options.includes(value)) {  
3   // ...
```

```
4 }
```

将乱七八糟的条件判断放到一个变量中存储，比看臃肿的表达式要好很多。

3、及早返回

有下面的代码：

```
1 function handleEvent(event) {
2   if (event) {
3     const target = event.target;
4     if (target) {
5       // Your awesome piece of code that uses target
6     }
7   }
8 }
9 复制代码
```

及早返回使得我们的代码更加易读：

```
1 function handleEvent(event) {
2   if (!event || !event.target) {
3     return;
4   }
5   // Your awesome piece of code that uses target
6 }
7 复制代码
```

4、解构赋值

在javascript中，我们可以对objects和arrays进行解构赋值。

例如：

```
1 // object 解构赋值
2 const numbers = {one: 1, two: 2};
3 const {one, two} = numbers;
4 console.log(one); // 1
5 console.log(two); // 2
6
7 // array 解构赋值
8 const numbers = [1, 2, 3, 4, 5];
9 const [one, two] = numbers;
10 console.log(one); // 1
11 console.log(two); // 2
```