

规范文档

术语

- 小骆驼: getUser
- 大骆驼: GetUser
- 下划线: get_user
- 大写下划线命名: GET_USER
- 中杠命名: learning-hub

变量作用域

- 私有变量: private 使用 `_` 下划线开头

命名规范

JAVA

属性: 小骆驼

方法: 小骆驼

类: 大骆驼

文件: 大骆驼

常量: 大写下划线

JS

变量: 小骆驼

属性: 小骆驼

方法: 小骆驼

类: 大骆驼

文件: 下划线或中杠命名

常量: 大写下划线

代码规范

- 尽量使用单层if

```

1  // good
2  if(0 == a && 0 == b){
3  }
4
5  // bad
6  if(0 == a){
7      if(0 == b){
8
9      }
10 }

```

- 尽量不要超出双层for
- 尽量使用 `0 == a` 这样的方式
- 运算符左右加一个空格
- 每条语句末尾加分号
- 如果一个函数有多个校验操作可以使用return或throw来截断错误的校验

```

1  // good
2  function sum (a = 0, b = 0) {
3      if ( !isNumber(a) ) {
4          throw new Error("a不是一个数字");
5      }
6
7      if ( !isNumber(b) ) {
8          throw new Error("b不是一个数字");
9      }
10
11     return a + b;
12 }
13 // bad
14 function sum (a = 0, b = 0) {
15     if ( isNumber(a) && isNumber(b) ) {
16         return a + b;
17     } else {
18         throw new Error("a或b不是一个数字");
19     }
20 }

```

- 代码尽量不要超出屏幕范围，如果超出请适当换行
- js中必须明确变量的类型
- js中引用类型必须使用const定义
- 尽量使用函数或方法，而不是for

```

1  const a = [
2      {
3          a:1
4      },
5      {
6          a:5
7      },
8      {
9          a:8

```

```
10     },
11 ];
12
13 // good
14 console.log(a.find(item => 5 === item.a));
15
16 // bad
17 for (let i = 0; i < a.length; i++){
18     if(5 === a[i].a){
19         console.log(a[i]);
20         break;
21     }
22 }
```

注释规范

- 除非变量名可以清楚的表达他自身的意思，否则必须在后面加注释
- 方法函数必须加注释，必须标出他的功能，参数，参数类型，参数含义，返回值类型、返回值含义，可以使用特殊的注解方式。
- 全大写的常量必须加注释
- if和for和switch的上面必须加注释
- 使用 `// TODO: <说明>` 来插入代码书签，表示真正做，但是在解决问题后需要删除
- `//` 后面尽量加空格