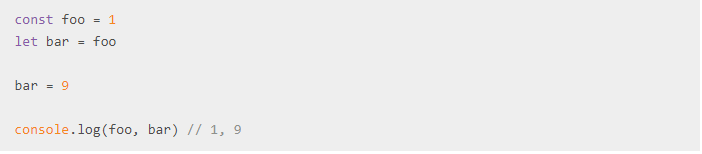
语言规范

1. 类型

1、基本类型

* 1. 字符串
  2. 数值
  3. 布尔类型
  4. Null
  5. Undefined

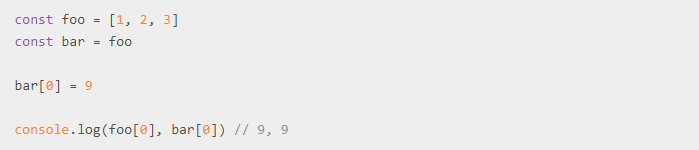


2、复杂类型

a) object

b) array

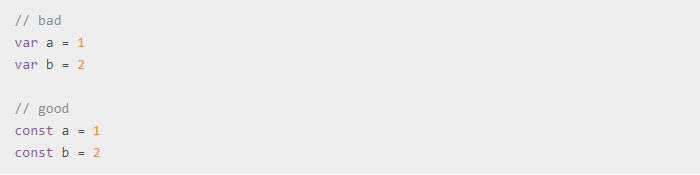
c) function



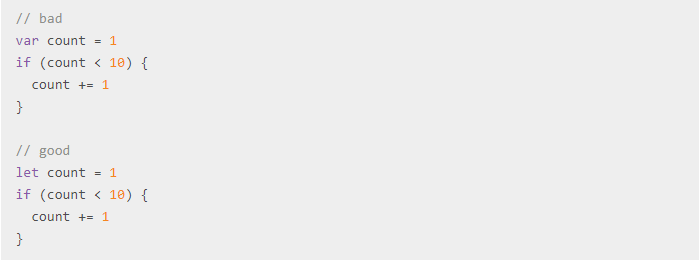
1. 引用

Const 和 let 都是块级作用域，var是函数级作用域

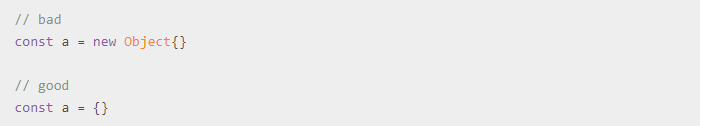
1. 对所有引用都使用const，不要使用var



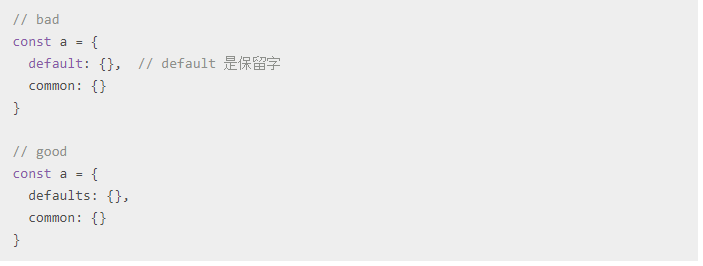
1. 如果引用是可变动的，则使用let



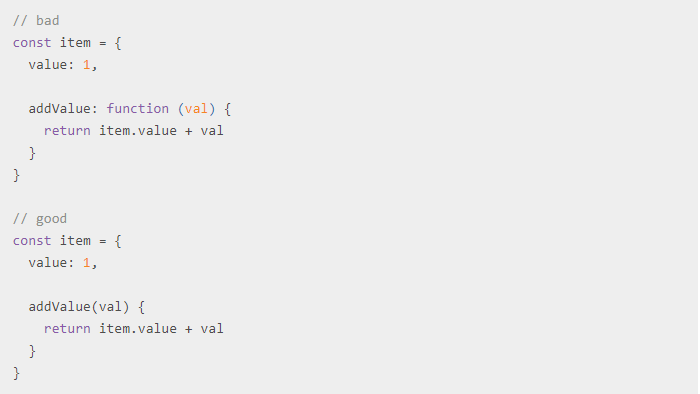
1. 对象
2. 请使用字面量创建对象



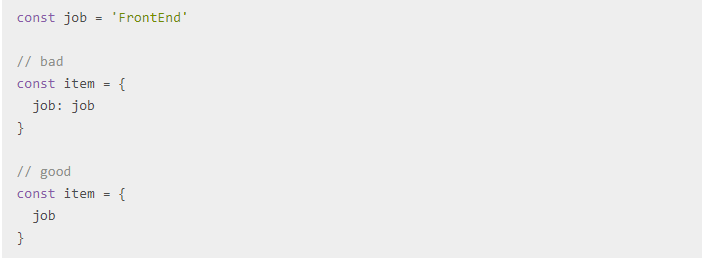
1. 别使用保留字段作为对象的键值，这样在ie8下不会运行（虽然项目不考虑ie8,但还是希望不要用保留字段作为对象键值）



1. 请使用对象方法的简写方式



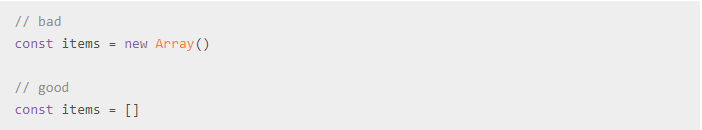
1. 请使用对象属性值得简写方式



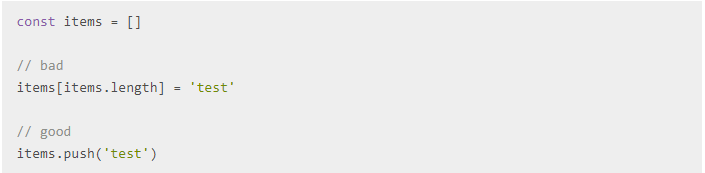
1. 对象属性值的简写方式要和声明式的方式分组



1. 数组
2. 请使用字面量值创建数组



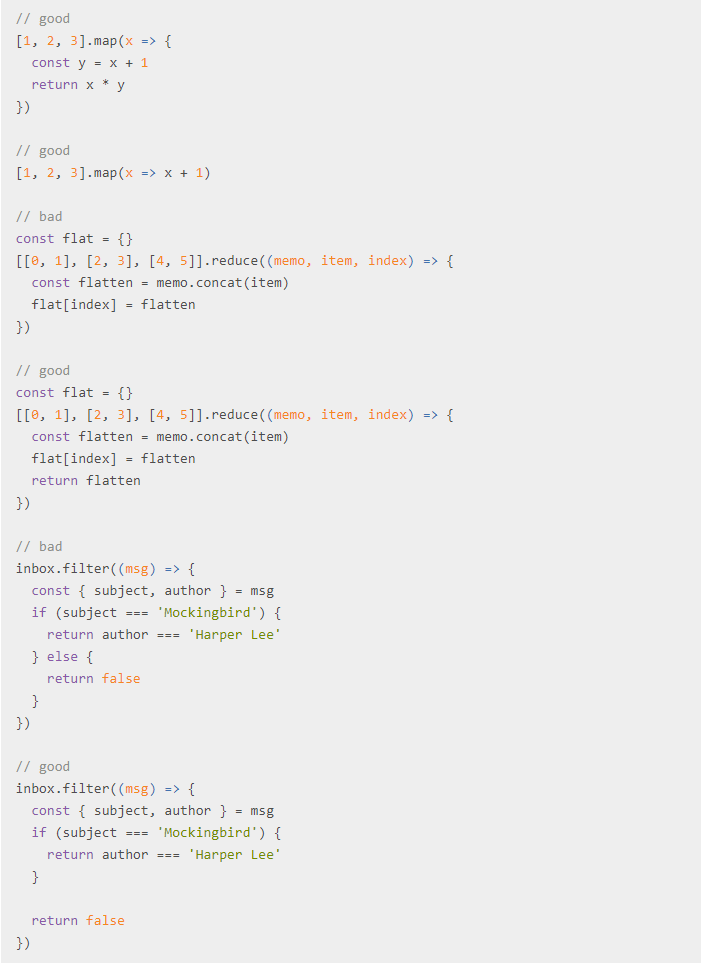
1. 向数组中添加元素时，请使用push方法



1. 使用拓展运算符…复制数组



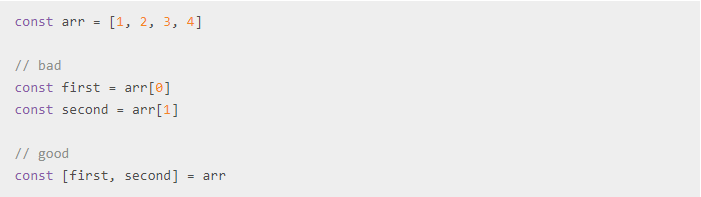
1. 使用数组的map等方法时，请使用return声明，如果是单一声明语句的情况，可省略retrun



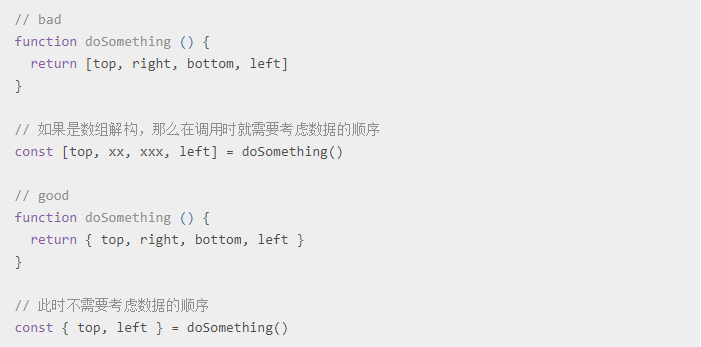
1. 解构赋值
2. 当需要是用对象的多个属性时，请使用解构赋值



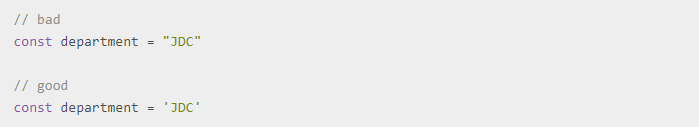
1. 当需要使用数组的多个值时，请同样使用解构赋值



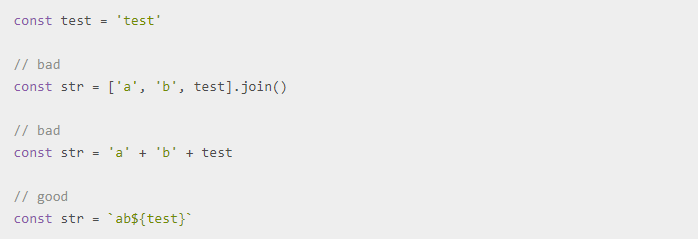
3、函数需要回传多个值时，请使用对象的解构，而不是数组的解构’



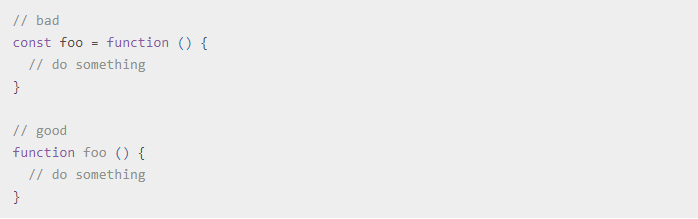
1. 字符串
2. 字符串统一使用单引号的形式 ‘’



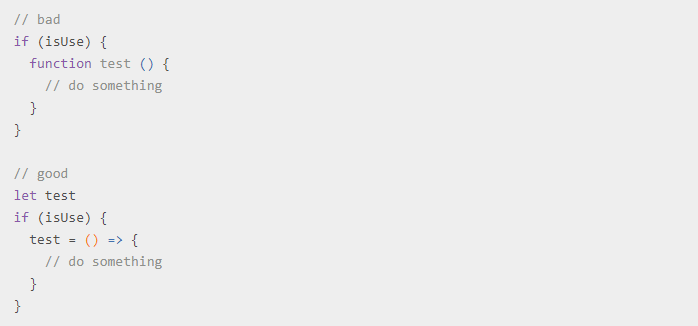
1. 程序化生成字符串时，请使用魔板字符串



1. 函数
2. 请使用函数声明，而不是函数表达式



1. 不要在非函数代码块中声明函数

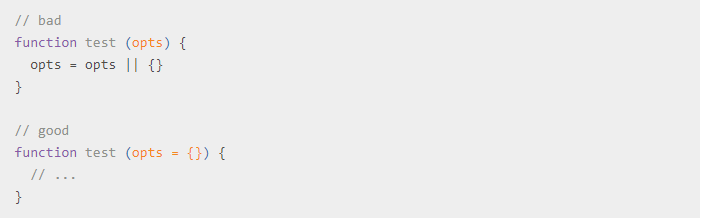


1. 不要使用arguments,可以选择使用…

Arguments是一个类数组，而…是一个真正的数组



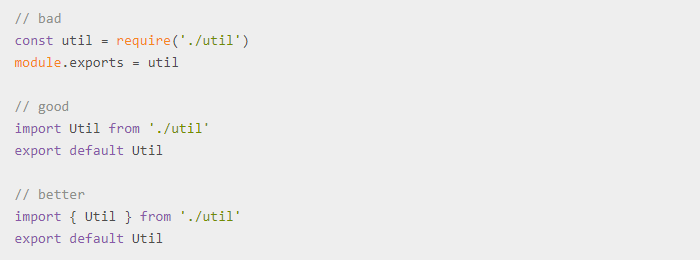
4、不要更改函数参数的值



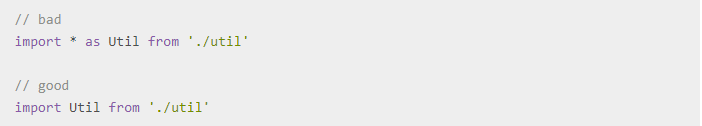
1. 原型
2. 使用class，避免直接操作prototype



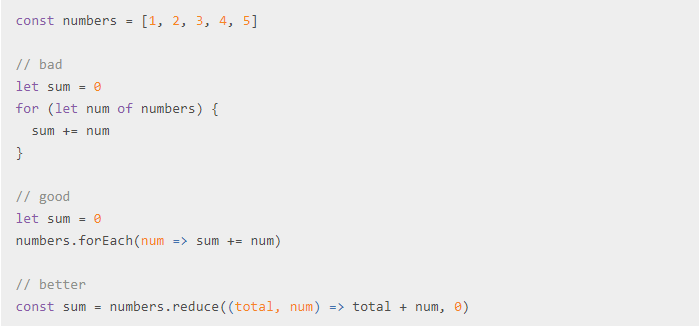
1. 模块
2. 使用标准的ES6模块语法import 和 export



1. 不要使用import的通配符\*，这样可以确保你只有一个默认的export



1. 迭代器
2. 不要使用iterators



十一、对象属性

1. 使用.来访问对象属性

