1. 命名规范
2. 文件命名

英文具象命名，命名要体现含义；单词全拼不得使用缩写，全部采用小写方式，若复式则采用中划线分隔，如：out-stock.js

（注：命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。 说明：正确的英文拼写和语法可以让阅读者易于理解，避免歧义。注意，即使纯拼音命名方式也要避免采用）

1. 变量、参数、方法命名

采用小写驼峰命名 lowerCamelCase，代码中的命名均不能以下划线开始，也不能以下划线或美元符号结束。

推荐：

customerNumber

不推荐：

\_CustomerNum

1. 方法命名必须是动词+名词格式

如：getNumber()

1. 常量命名

常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚不要缩写，不要嫌名字长。

如：MAX\_COUNT

二、编码规范

1、代码格式

1. 使用 2 个空格进行缩进（这个会在eslint中做设置）

如：

if (x < y) {

x += 10;

} else {

x += 1;

}

1. 不同逻辑、不同语义、不同业务的代码之间插入一个空行分隔开来以提升可读性。（注：任何情形，没有必要插入多个空行进行隔开。）
2. 字符串

统一使用单引号(‘)，不使用双引号(“)，这个会在eslint中设置规范。相反，在html中用双引号添加属性、事件。两者不会造成冲突。

如：

let tips = 'foo';

let tipsDiv = '<div id="test"></div>';

1. 对象声明

使用字面量来代替对象构造器

推荐：

var user = {

age: 0,

name: 1,

city: 3

}

不推荐：

var user = new Object();

user.age = 0;

user.name = 0;

user.city = 0;

1. 关于es6

优先使用es6的语法糖和函数，比如箭头函数、await/async ， 解构， let ， for…of 等等

1. 关于括号

下列关键字后必须有大括号（即使代码块的内容只有一行）：if, else, for, while, do, switch, try, catch, finally, with。

推荐：

if (condition) {

doSomething();

}

不推荐：

if (condition) doSomething()

1. undefined 判断

永远不要直接使用 undefined 进行变量判断；使用 typeof 和字符串’undefined’对变量进行判断。

推荐：

if (typeof person === 'undefined') {

...

}

不推荐：

if (person === undefined) {

...

}

1. 条件判断和循环最多三层

条件判断能使用三目运算符和逻辑运算符解决的，就不要使用条件判断，但是谨记不要写太长的三目运算符。如果超过 3 层请抽成函数，并写清楚注释。

1. this 的转换命名

对上下文 this 的引用只能使用’self’来命名

1. 慎用 console.log

因 console.log 大量使用会有性能问题，所以在非 webpack 项目中谨慎使用 log 功能。

1. 函数声明

每个函数必须增加注释，注释中要写明函数功能，有入参的话要注明参数数据类型和参数定义，若参数是枚举值的话，也要列举上所有的枚举和对应枚举的含义。如果有返回值也要定义返回值数据类型及定义。

例：

/\*

\* 日期文本装换

\* @param { String } dateString 日期文本

\* @returns { String } 转换后的文本

\*/

1. 关于函数参数声明

定义函数参数时，必要参数定义为一个参数，可选参数放在一个可选对象中。

例：

/\*

\* 日期文本装换

\* @param { String } dateString 日期文本

\* @param { Object} options 格式参数

\* @param { String} options.delimiter 年月日分隔符（可选参数）

\* @returns { String } 转换后的文本

\*/

function getDate(dateString, options) {}