官方网站

https://cn.eslint.org/



# 前言

ESLint 由 JavaScript 红宝书 作者 Nicholas C. Zakas 编写， 2013 年发布第一个版本。 NCZ 的初衷不是重复造一个轮子，而是在实际需求得不到 JSHint 团队响应 的情况下做出的选择：以可扩展、每条规则独立、不内置编码风格为理念编写一个 lint 工具。

JSX就是Javascript和XML结合的一种格式。React发明了JSX，利用HTML语法来创建虚拟DOM。

[中文官方网站](https://link.jianshu.com?t=http://eslint.cn/" \t "_blank)

[英文官方网站](https://link.jianshu.com?t=http://eslint.org/" \t "_blank)

可以去上方网站进行相关资料查阅，长话短说，直接开始入门吧~

# 安装

建议采用全局安装方式

npm install -g eslint

# 初始化

如果你的项目还没有配置文件（.eslintrc）的话，可以通过指定--init参数来生成一个新的配置文件：

eslint --init

运行eslint --init后，会在当前目录下生成一个.eslintrc文件，你可以在这个文件中配置一些规则。

# 配置

ESLint被设计为完全可配置的，主要有两种方式来配置ESLint：

1. 内嵌配置：将配置信息以注释的形式写在要检验的文件内
2. 文件配置：利用javascript、json、yml语言的语法，生成名为.eslintrc.\*的配置文件，将配置信息写到文件内

## 文件配置方式

**env**：你的脚本将要运行在什么环境中

Environment可以预设好的其他环境的全局变量，如brower、node环境变量、es6环境变量、mocha环境变量等

// env：你的脚本将要运行在什么环境中  
'env': {  
 'browser': **true**,  
 'commonjs': **true**,  
 'node': **true**,  
 'es6': **true**,  
},

**globals**：额外的全局变量

示例：

globals: {  
 vue: **true**,  
 wx: **true**},

// 启用推荐的规则，报告一些常见的问题  
'extends': 'eslint:recommended',

\*\* parserOptions: \*\* JavaScript 语言选项

// \*\* parserOptions: \*\* JavaScript 语言选项  
'parserOptions': {  
 // ECMAScript 版本  
 'ecmaVersion': 6,  
 'sourceType': 'script', // module  
 // 想使用的额外的语言特性:  
 'ecmaFeatures': {  
 // 允许在全局作用域下使用 return 语句  
 'globalReturn': **true**,  
 // impliedStric  
 'impliedStrict': **true**,  
 // 启用 JSX  
 'jsx': **true**,  
 },  
},

**rules**：开启规则和发生错误时报告的等级  
规则的错误等级有三种：  
0或'off'：关闭规则。  
1或'warn'：打开规则，并且作为一个警告（并不会导致检查不通过）。  
2或'error'：打开规则，并且作为一个错误 (退出码为1，检查不通过)。

参数说明：  
参数1 ： 错误等级  
参数2 ： 处理方式

"comma-dangle": ["error", "never"],

示例：

'rules': {  
 // /////////////  
 // 可能的错误 //  
 // //////////////  
  
 // 禁止条件表达式中出现赋值操作符  
 'no-cond-assign': 2,  
 // 禁用 console  
 'no-console': 0,  
 // 禁止在条件中使用常量表达式  
 // if (false) {  
 // doSomethingUnfinished();  
 // } //cuowu  
 'no-constant-condition': 2,  
 // 禁止在正则表达式中使用控制字符 ：new RegExp("\x1f")  
 'no-control-regex': 2,  
 // 数组和对象键值对最后一个逗号， never参数：不能带末尾的逗号, always参数：必须带末尾的逗号，  
 // always-multiline：多行模式必须带逗号，单行模式不能带逗号  
 'comma-dangle': [1, 'always-multiline'],  
 // 禁用 debugger  
 'no-debugger': 2,  
 // 禁止 function 定义中出现重名参数  
 'no-dupe-args': 2,  
 // 禁止对象字面量中出现重复的 key  
 'no-dupe-keys': 2,  
 // 禁止重复的 case 标签  
 'no-duplicate-case': 2,  
 // 禁止空语句块  
 'no-empty': 2,  
 // 禁止在正则表达式中使用空字符集 (/^abc[]/)  
 'no-empty-character-class': 2,  
 // 禁止对 catch 子句的参数重新赋值  
 'no-ex-assign': 2,  
 // 禁止不必要的布尔转换  
 'no-extra-boolean-cast': 2,  
 // 禁止不必要的括号 //(a \* b) + c;//报错  
 'no-extra-parens': 0,  
 // 禁止不必要的分号  
 'no-extra-semi': 2,  
 // 禁止对 function 声明重新赋值  
 'no-func-assign': 2,  
 // 禁止在嵌套的块中出现 function 或 var 声明  
 'no-inner-declarations': [2, 'functions'],  
 // 禁止 RegExp 构造函数中无效的正则表达式字符串  
 'no-invalid-regexp': 2,  
 // 禁止在字符串和注释之外不规则的空白  
 'no-irregular-whitespace': 2,  
 // 禁止在 in 表达式中出现否定的左操作数  
 'no-negated-in-lhs': 2,  
 // 禁止把全局对象 (Math 和 JSON) 作为函数调用 错误：var math = Math();  
 'no-obj-calls': 2,  
 // 禁止直接使用 Object.prototypes 的内置属性  
 'no-prototype-builtins': 0,  
 // 禁止正则表达式字面量中出现多个空格  
 'no-regex-spaces': 2,  
 // 禁用稀疏数组  
 'no-sparse-arrays': 2,  
 // 禁止出现令人困惑的多行表达式  
 'no-unexpected-multiline': 2,  
 // 禁止在return、throw、continue 和 break语句之后出现不可达代码  
 /\*  
 function foo() {  
 return true;  
 console.log("done");  
 }//错误  
 \*/  
 'no-unreachable': 2,  
 // 要求使用 isNaN() 检查 NaN  
 'use-isnan': 2,  
 // 强制使用有效的 JSDoc 注释  
 'valid-jsdoc': 1,  
 // 强制 typeof 表达式与有效的字符串进行比较  
 // typeof foo === "undefimed" 错误  
 'valid-typeof': 2,  
 // ////////////  
 // 最佳实践 //  
 // ////////////  
  
 // 定义对象的set存取器属性时，强制定义get  
 'accessor-pairs': 2,  
 // 强制数组方法的回调函数中有 return 语句  
 'array-callback-return': 0,  
 // 强制把变量的使用限制在其定义的作用域范围内  
 'block-scoped-var': 0,  
 // 限制圈复杂度，也就是类似if else能连续接多少个  
 'complexity': [2, 9],  
 // 要求 return 语句要么总是指定返回的值，要么不指定  
 'consistent-return': 0,  
 // 强制所有控制语句使用一致的括号风格  
 'curly': [2, 'all'],  
 // switch 语句强制 default 分支，也可添加 // no default 注释取消此次警告  
 'default-case': 2,  
 // 强制object.key 中 . 的位置，参数:  
 // property，'.'号应与属性在同一行  
 // object, '.' 号应与对象名在同一行  
 'dot-location': [2, 'property'],  
 // 强制使用.号取属性  
 // 参数： allowKeywords：true 使用保留字做属性名时，只能使用.方式取属性  
 // false 使用保留字做属性名时, 只能使用[]方式取属性 e.g [2, {"allowKeywords": false}]  
 // allowPattern: 当属性名匹配提供的正则表达式时，允许使用[]方式取值,否则只能用.号取值 e.g [2, {"allowPattern": "^[a-z]+(\_[a-z]+)+$"}]  
 'dot-notation': [2, {'allowKeywords': **false**}],  
 // 使用 === 替代 == allow-null允许null和undefined==  
 'eqeqeq': [2, 'allow-null'],  
 // 要求 for-in 循环中有一个 if 语句  
 'guard-for-in': 2,  
 // 禁用 alert、confirm 和 prompt  
 'no-alert': 0,  
 // 禁用 arguments.caller 或 arguments.callee  
 'no-caller': 2,  
 // 不允许在 case 子句中使用词法声明  
 'no-case-declarations': 2,  
 // 禁止除法操作符显式的出现在正则表达式开始的位置  
 'no-div-regex': 2,  
 // 禁止 if 语句中有 return 之后有 else  
 'no-else-return': 0,  
 // 禁止出现空函数.如果一个函数包含了一条注释，它将不会被认为有问题。  
 'no-empty-function': 2,  
 // 禁止使用空解构模式no-empty-pattern  
 'no-empty-pattern': 2,  
 // 禁止在没有类型检查操作符的情况下与 null 进行比较  
 'no-eq-null': 1,  
 // 禁用 eval()  
 'no-eval': 2,  
 // 禁止扩展原生类型  
 'no-extend-native': 2,  
 // 禁止不必要的 .bind() 调用  
 'no-extra-bind': 2,  
 // 禁用不必要的标签  
 'no-extra-label:': 0,  
 // 禁止 case 语句落空  
 'no-fallthrough': 2,  
 // 禁止数字字面量中使用前导和末尾小数点  
 'no-floating-decimal': 2,  
 // 禁止使用短符号进行类型转换(!!fOO)  
 'no-implicit-coercion': 0,  
 // 禁止在全局范围内使用 var 和命名的 function 声明  
 'no-implicit-globals': 1,  
 // 禁止使用类似 eval() 的方法  
 'no-implied-eval': 2,  
 // 禁止 this 关键字出现在类和类对象之外  
 'no-invalid-this': 0,  
 // 禁用 \_\_iterator\_\_ 属性  
 'no-iterator': 2,  
 // 禁用标签语句  
 'no-labels': 2,  
 // 禁用不必要的嵌套块  
 'no-lone-blocks': 2,  
 // 禁止在循环中出现 function 声明和表达式  
 'no-loop-func': 1,  
 // 禁用魔术数字(3.14什么的用常量代替)  
 'no-magic-numbers': [1, {'ignore': [0, -1, 1]}],  
 // 禁止使用多个空格  
 'no-multi-spaces': 2,  
 // 禁止使用多行字符串，在 JavaScript 中，可以在新行之前使用斜线创建多行字符串  
 'no-multi-str': 2,  
 // 禁止对原生对象赋值  
 'no-native-reassign': 2,  
 // 禁止在非赋值或条件语句中使用 new 操作符  
 'no-new': 2,  
 // 禁止对 Function 对象使用 new 操作符  
 'no-new-func': 0,  
 // 禁止对 String，Number 和 Boolean 使用 new 操作符  
 'no-new-wrappers': 2,  
 // 禁用八进制字面量  
 'no-octal': 2,  
 // 禁止在字符串中使用八进制转义序列  
 'no-octal-escape': 2,  
 // 不允许对 function 的参数进行重新赋值  
 'no-param-reassign': 0,  
 // 禁用 \_\_proto\_\_ 属性  
 'no-proto': 2,  
 // 禁止使用 var 多次声明同一变量  
 'no-redeclare': 2,  
 // 禁用指定的通过 require 加载的模块  
 'no-return-assign': 0,  
 // 禁止使用 javascript: url  
 'no-script-url': 0,  
 // 禁止自我赋值  
 'no-self-assign': 2,  
 // 禁止自身比较  
 'no-self-compare': 2,  
 // 禁用逗号操作符  
 'no-sequences': 2,  
 // 禁止抛出非异常字面量  
 'no-throw-literal': 2,  
 // 禁用一成不变的循环条件  
 'no-unmodified-loop-condition': 2,  
 // 禁止出现未使用过的表达式  
 'no-unused-expressions': 0,  
 // 禁用未使用过的标签  
 'no-unused-labels': 2,  
 // 禁止不必要的 .call() 和 .apply()  
 'no-useless-call': 2,  
 // 禁止不必要的字符串字面量或模板字面量的连接  
 'no-useless-concat': 2,  
 // 禁用不必要的转义字符  
 'no-useless-escape': 0,  
 // 禁用 void 操作符  
 'no-void': 0,  
 // 禁止在注释中使用特定的警告术语  
 'no-warning-comments': 0,  
 // 禁用 with 语句  
 'no-with': 2,  
 // 强制在parseInt()使用基数参数  
 'radix': 2,  
 // 要求所有的 var 声明出现在它们所在的作用域顶部  
 'vars-on-top': 0,  
 // 要求 IIFE 使用括号括起来  
 'wrap-iife': [2, 'any'],  
 // 要求或禁止 “Yoda” 条件  
 'yoda': [2, 'never'],  
 // 要求或禁止使用严格模式指令  
 'strict': 0,  
 // ////////////  
 // 变量声明 //  
 // ////////////  
  
 // 要求或禁止 var 声明中的初始化(初值)  
 'init-declarations': 0,  
 // 不允许 catch 子句的参数与外层作用域中的变量同名  
 'no-catch-shadow': 0,  
 // 禁止删除变量  
 'no-delete-var': 2,  
 // 不允许标签与变量同名  
 'no-label-var': 2,  
 // 禁用特定的全局变量  
 'no-restricted-globals': 0,  
 // 禁止 var 声明 与外层作用域的变量同名  
 'no-shadow': 0,  
 // 禁止覆盖受限制的标识符  
 'no-shadow-restricted-names': 2,  
 // 禁用未声明的变量，除非它们在 /\*global \*/ 注释中被提到  
 'no-undef': 2,  
 // 禁止将变量初始化为 undefined  
 'no-undef-init': 2,  
 // 禁止将 undefined 作为标识符  
 'no-undefined': 0,  
 // 禁止出现未使用过的变量  
 'no-unused-vars': [2, {'vars': 'all', 'args': 'none'}],  
 // 不允许在变量定义之前使用它们  
 'no-use-before-define': 0,  
 // ////////////////////////  
 // Node.js and CommonJS //  
 // ////////////////////////  
  
 // require return statements after callbacks  
 'callback-return': 0,  
 // 要求 require() 出现在顶层模块作用域中  
 'global-require': 1,  
 // 要求回调函数中有容错处理  
 'handle-callback-err': [2, '^(err|error)$'],  
 // 禁止混合常规 var 声明和 require 调用  
 'no-mixed-requires': 0,  
 // 禁止调用 require 时使用 new 操作符  
 'no-new-require': 2,  
 // 禁止对 \_\_dirname 和 \_\_filename进行字符串连接  
 'no-path-concat': 0,  
 // 禁用 process.env  
 'no-process-env': 0,  
 // 禁用 process.exit()  
 'no-process-exit': 0,  
 // 禁用同步方法  
 'no-sync': 0,  
 // ////////////  
 // 风格指南 //  
 // ////////////  
  
 // 指定数组的元素之间要以空格隔开(, 后面)， never参数：[ 之前和 ] 之后不能带空格，always参数：[ 之前和 ] 之后必须带空格  
 'array-bracket-spacing': [2, 'never'],  
 // 禁止或强制在单行代码块中使用空格(禁用)  
 'block-spacing': [1, 'never'],  
 // 强制使用一致的缩进 第二个参数为 "tab" 时，会使用tab，  
 // if while function 后面的{必须与if在同一行，java风格。  
 'brace-style': [2, '1tbs', {'allowSingleLine': **true**}],  
 // 双峰驼命名格式  
 'camelcase': 2,  
 // 控制逗号前后的空格  
 'comma-spacing': [2, {'before': **false**, 'after': **true**}],  
 // 控制逗号在行尾出现还是在行首出现 (默认行尾)  
 // http://eslint.org/docs/rules/comma-style  
 'comma-style': [2, 'last'],  
 // "SwitchCase" (默认：0) 强制 switch 语句中的 case 子句的缩进水平  
 // 以方括号取对象属性时，[ 后面和 ] 前面是否需要空格, 可选参数 never, always  
 'computed-property-spacing': [2, 'never'],  
 // 用于指统一在回调函数中指向this的变量名，箭头函数中的this已经可以指向外层调用者，应该没卵用了  
 // e.g [0,"that"] 指定只能 var that = this. that不能指向其他任何值，this也不能赋值给that以外的其他值  
 'consistent-this': [1, 'that'],  
 // 强制使用命名的 function 表达式  
 'func-names': 0,  
 // 文件末尾强制换行  
 'eol-last': 2,  
 'indent': [2, 4, {'SwitchCase': 1}],  
 // 强制在对象字面量的属性中键和值之间使用一致的间距  
 'key-spacing': [2, {'beforeColon': **false**, 'afterColon': **true**}],  
 // 强制使用一致的换行风格  
 'linebreak-style': [1, 'unix'],  
 // 要求在注释周围有空行 ( 要求在块级注释之前有一空行)  
 'lines-around-comment': [1, {'beforeBlockComment': **true**}],  
 // 强制一致地使用函数声明或函数表达式，方法定义风格，参数：  
 // declaration: 强制使用方法声明的方式，function f(){} e.g [2, "declaration"]  
 // expression：强制使用方法表达式的方式，var f = function() {} e.g [2, "expression"]  
 // allowArrowFunctions: declaration风格中允许箭头函数。 e.g [2, "declaration", { "allowArrowFunctions": true }]  
 'func-style': 0,  
 // 强制回调函数最大嵌套深度 5层  
 'max-nested-callbacks': [1, 5],  
 // 禁止使用指定的标识符  
 'id-blacklist': 0,  
 // 强制标识符的最新和最大长度  
 'id-length': 0,  
 // 要求标识符匹配一个指定的正则表达式  
 'id-match': 0,  
 // 强制在 JSX 属性中一致地使用双引号或单引号  
 'jsx-quotes': 0,  
 // 强制在关键字前后使用一致的空格 (前后腰需要)  
 'keyword-spacing': 2,  
 // 强制一行的最大长度  
 'max-len': [1, 200],  
 // 强制最大行数  
 'max-lines': 0,  
 // 强制 function 定义中最多允许的参数数量  
 'max-params': [1, 7],  
 // 强制 function 块最多允许的的语句数量  
 'max-statements': [1, 200],  
 // 强制每一行中所允许的最大语句数量  
 'max-statements-per-line': 0,  
 // 要求构造函数首字母大写 （要求调用 new 操作符时有首字母大小的函数，允许调用首字母大写的函数时没有 new 操作符。）  
 'new-cap': [2, {'newIsCap': **true**, 'capIsNew': **false**}],  
 // 要求调用无参构造函数时有圆括号  
 'new-parens': 2,  
 // 要求或禁止 var 声明语句后有一行空行  
 'newline-after-var': 0,  
 // 禁止使用 Array 构造函数  
 'no-array-constructor': 2,  
 // 禁用按位运算符  
 'no-bitwise': 0,  
 // 要求 return 语句之前有一空行  
 'newline-before-return': 0,  
 // 要求方法链中每个调用都有一个换行符  
 'newline-per-chained-call': 1,  
 // 禁用 continue 语句  
 'no-continue': 0,  
 // 禁止在代码行后使用内联注释  
 'no-inline-comments': 0,  
 // 禁止 if 作为唯一的语句出现在 else 语句中  
 'no-lonely-if': 0,  
 // 禁止混合使用不同的操作符  
 'no-mixed-operators': 0,  
 // 不允许空格和 tab 混合缩进  
 'no-mixed-spaces-and-tabs': 2,  
 // 不允许多个空行  
 'no-multiple-empty-lines': [2, {'max': 2}],  
 // 不允许否定的表达式  
 'no-negated-condition': 0,  
 // 不允许使用嵌套的三元表达式  
 'no-nested-ternary': 0,  
 // 禁止使用 Object 的构造函数  
 'no-new-object': 2,  
 // 禁止使用一元操作符 ++ 和 --  
 'no-plusplus': 0,  
 // 禁止使用特定的语法  
 'no-restricted-syntax': 0,  
 // 禁止 function 标识符和括号之间出现空格  
 'no-spaced-func': 2,  
 // 不允许使用三元操作符  
 'no-ternary': 0,  
 // 禁用行尾空格  
 'no-trailing-spaces': 2,  
 // 禁止标识符中有悬空下划线\_bar  
 'no-underscore-dangle': 0,  
 // 禁止可以在有更简单的可替代的表达式时使用三元操作符  
 'no-unneeded-ternary': 2,  
 // 禁止属性前有空白  
 'no-whitespace-before-property': 0,  
 // 强制花括号内换行符的一致性  
 'object-curly-newline': 0,  
 // 强制在花括号中使用一致的空格  
 'object-curly-spacing': 0,  
 // 强制将对象的属性放在不同的行上  
 'object-property-newline': 0,  
 // 强制函数中的变量要么一起声明要么分开声明  
 'one-var': [2, {'initialized': 'never'}],  
 // 要求或禁止在 var 声明周围换行  
 'one-var-declaration-per-line': 0,  
 // 要求或禁止在可能的情况下要求使用简化的赋值操作符  
 'operator-assignment': 0,  
 // 强制操作符使用一致的换行符  
 'operator-linebreak': [2, 'after', {'overrides': {'?': 'before', ':': 'before'}}],  
 // 要求或禁止块内填充  
 'padded-blocks': 0,  
 // 要求对象字面量属性名称用引号括起来  
 'quote-props': 0,  
 // 强制使用一致的反勾号、双引号或单引号  
 'quotes': [2, 'single', 'avoid-escape'],  
 // 要求使用 JSDoc 注释  
 'require-jsdoc': 1,  
 // 要求或禁止使用分号而不是 ASI（这个才是控制行尾部分号的，）  
 'semi': [2, 'always'],  
 // 强制分号之前和之后使用一致的空格  
 'semi-spacing': 0,  
 // 要求同一个声明块中的变量按顺序排列  
 'sort-vars': 0,  
 // 强制在块之前使用一致的空格  
 'space-before-blocks': [2, 'always'],  
 // 强制在 function的左括号之前使用一致的空格  
 'space-before-function-paren': [2, 'always'],  
 // 强制在圆括号内使用一致的空格  
 'space-in-parens': [2, 'never'],  
 // 要求操作符周围有空格  
 'space-infix-ops': 2,  
 // 强制在一元操作符前后使用一致的空格  
 'space-unary-ops': [2, {'words': **true**, 'nonwords': **false**}],  
 // 强制在注释中 // 或 /\* 使用一致的空格  
 'spaced-comment': [2, 'always', {'markers': ['global', 'globals', 'eslint', 'eslint-disable', '\*package', '!']}],  
 // 要求或禁止 Unicode BOM  
 'unicode-bom': 0,  
 // 要求正则表达式被括号括起来  
 'wrap-regex': 0,  
 // ////////////  
 // ES6.相关 //  
 // ////////////  
  
 // 要求箭头函数体使用大括号  
 'arrow-body-style': 2,  
 // 要求箭头函数的参数使用圆括号  
 'arrow-parens': 2,  
 'arrow-spacing': [2, {'before': **true**, 'after': **true**}],  
 // 强制在子类构造函数中用super()调用父类构造函数，TypeScrip的编译器也会提示  
 'constructor-super': 0,  
 // 强制 generator 函数中 \* 号周围使用一致的空格  
 'generator-star-spacing': [2, {'before': **true**, 'after': **true**}],  
 // 禁止修改类声明的变量  
 'no-class-assign': 2,  
 // 不允许箭头功能，在那里他们可以混淆的比较  
 'no-confusing-arrow': 0,  
 // 禁止修改 const 声明的变量  
 'no-const-assign': 2,  
 // 禁止类成员中出现重复的名称  
 'no-dupe-class-members': 2,  
 // 不允许复制模块的进口  
 'no-duplicate-imports': 0,  
 // 禁止 Symbol 的构造函数  
 'no-new-symbol': 2,  
 // 允许指定模块加载时的进口  
 'no-restricted-imports': 0,  
 // 禁止在构造函数中，在调用 super() 之前使用 this 或 super  
 'no-this-before-super': 2,  
 // 禁止不必要的计算性能键对象的文字  
 'no-useless-computed-key': 0,  
 // 要求使用 let 或 const 而不是 var  
 'no-var': 0,  
 // 要求或禁止对象字面量中方法和属性使用简写语法  
 'object-shorthand': 0,  
 // 要求使用箭头函数作为回调  
 'prefer-arrow-callback': 0,  
 // 要求使用 const 声明那些声明后不再被修改的变量  
 'prefer-const': 0,  
 // 要求在合适的地方使用 Reflect 方法  
 'prefer-reflect': 0,  
 // 要求使用扩展运算符而非 .apply()  
 'prefer-spread': 0,  
 // 要求使用模板字面量而非字符串连接  
 'prefer-template': 0,  
 // Suggest using the rest parameters instead of arguments  
 'prefer-rest-params': 0,  
 // 要求generator 函数内有 yield  
 'require-yield': 0,  
 // enforce spacing between rest and spread operators and their expressions  
 'rest-spread-spacing': 0,  
 // 强制模块内的 import 排序  
 'sort-imports': 0,  
 // 要求或禁止模板字符串中的嵌入表达式周围空格的使用  
 'template-curly-spacing': 1,  
 // 强制在 yield\* 表达式中 \* 周围使用空格  
 'yield-star-spacing': 2,  
},

上述配置只是简要将最基本的一些配置罗略出来，如果有需要建议去上方官网中查看。

## 配置代码注释方式

有时我们可能要在代码中忽略eslint的某种检查，或者加入某种特定检查，此时我们可以用如下的方式：

示例：

1. 忽略 no-undef 检查

/\* eslint-disable no-undef \*/

1. 忽略 no-new 检查

/\* eslint-disable no-new \*/

1. 设置检查

/\*eslint eqeqeq: off\*//\*eslint eqeqeq: 0\*/

## eslint 检查指令

1. 检查且修复

eslint \* --fix

1. 检查指定文件

eslint app.js --fix

## 开发工具上的使用

### WebStorm

Preferences -> Languages & Frameworks -> JavaScript -> Code Quality Tools -> Eslint -> Enable (勾选) -> Apply -> OK

### Atom

安装linter-eslint

### Sublime

安装SublimeLinter

安装SublimeLinter-contrib-eslint

# 结语

Eslint 对于每一个开发者而言都是非常值得使用的，这样会强制你写出高质量且整洁的代码，所以建议无论大家是前端开发者、weex开发者、还是小程序开发者都建议大家讲Eslint集成进入，来提高项目的质量。

链接：https://www.jianshu.com/p/22e6197e5cff