

Rules

为了让你对规则有个更好的理解，ESLint 对其进行了分门别类。

所有的规则默认都是禁用的。在配置文件 (`../user-guide/configuring#extending-configuration-files`) 中，使用 `"extends": "eslint:recommended"` 来启用推荐的规则，报告一些常见的问题，在下文中这些推荐的规则都带有一个✔标记。



命令行 (`../user-guide/command-line-interface#fixing-problems`) 的 `--fix` 选项用来自动修复规则所报告的问题（目前，大部分是对空白的修复），在下文中会有一个🔧的图标。

Possible Errors

这些规则与 JavaScript 代码中可能的错误或逻辑错误有关：

✔	for-direction (for-direction)	强制“for”循环中更新子句的计数器朝着正确的方向移动
✔	getter-return (getter-return)	强制 getter 函数中出现 return 语句
✔	no-async-promise-executor (no-async-promise-executor)	禁止使用异步函数作为 Promise executor
	no-await-in-loop (no-await-in-loop)	禁止在循环中出现 await
✔	no-compare-neg-zero (no-compare-neg-zero)	禁止与 -0 进行比较
✔	no-cond-assign (no-cond-assign)	禁止条件表达式中出现赋值操作符
	no-console (no-console)	禁用 console
✔	no-constant-condition (no-constant-condition)	禁止在条件中使用常量表达式
✔	no-control-regex (no-control-regex)	禁止在正则表达式中使用控制字符
✔	no-debugger (no-debugger)	禁用 debugger

✓	no-dupe-args (no-dupe-args)	禁止 function 定义中出现重名参数
✓	no-dupe-keys (no-dupe-keys)	禁止对象字面量中出现重复的 key
✓	no-duplicate-case (no-duplicate-case)	禁止出现重复的 case 标签
✓	no-empty (no-empty)	禁止出现空语句块
✓	no-empty-character-class (no-empty-character-class)	禁止在正则表达式中使用空字符集
✓	no-ex-assign (no-ex-assign)	禁止对 catch 子句的参数重新赋值
✓	✂ no-extra-boolean-cast (no-extra-boolean-cast)	禁止不必要的布尔转换
	✂ no-extra-parens (no-extra-parens)	禁止不必要的括号
✓	✂ no-extra-semi (no-extra-semi)	禁止不必要的分号
✓	no-func-assign (no-func-assign)	禁止对 function 声明重新赋值
✓	no-inner-declarations (no-inner-declarations)	禁止在嵌套的块中出现变量声明或 function 声明
✓	no-invalid-regexp (no-invalid-regexp)	禁止 RegExp 构造函数中存在无效的正则表达式字符串
✓	no-irregular-whitespace (no-irregular-whitespace)	禁止不规则的空白
✓	no-misleading-character-class (no-misleading-character-class)	不允许在字符类语法中出现由多个代码点组成的字符
✓	no-obj-calls (no-obj-calls)	禁止把全局对象作为函数调用
✓	no-prototype-builtins (no-prototype-builtins)	禁止直接调用 Object.prototype 的内置属性


✓ 	no-regex-spaces (no-regex-spaces)	禁止正则表达式字面量中出现多个空格
✓	no-sparse-arrays (no-sparse-arrays)	禁用稀疏数组
	no-template-curly-in-string (no-template-curly-in-string)	禁止在常规字符串中出现模板字面量占位符语法
✓	no-unexpected-multiline (no-unexpected-multiline)	禁止出现令人困惑的多行表达式
✓	no-unreachable (no-unreachable)	禁止在 <code>return</code> 、 <code>throw</code> 、 <code>continue</code> 和 <code>break</code> 语句之后出现不可达代码
✓	no-unsafe-finally (no-unsafe-finally)	禁止在 <code>finally</code> 语句块中出现控制流语句
✓ 	no-unsafe-negation (no-unsafe-negation)	禁止对关系运算符的左操作数使用否定操作符
✓	require-atomic-updates (require-atomic-updates)	禁止由于 <code>await</code> 或 <code>yield</code> 的使用而可能导致出现竞态条件的赋值
✓	use-isnan (use-isnan)	要求使用 <code>isNaN()</code> 检查 <code>NaN</code>
✓	valid-typeof (valid-typeof)	强制 <code>typeof</code> 表达式与有效的字符串进行比较

Best Practices

这些规则是关于最佳实践的，帮助你避免一些问题

	accessor-pairs (accessor-pairs)	强制 <code>getter</code> 和 <code>setter</code> 在对象中成对出现
	array-callback-return (array-callback-return)	强制数组方法的回调函数中有 <code>return</code> 语句
	block-scoped-var (block-scoped-var)	强制把变量的使用限制在其定义的作用域范围内
	class-methods-use-this (class-methods-use-this)	强制类方法使用 <code>this</code>
	complexity (complexity)	指定程序中允许的最大环路复杂度

	consistent-return (consistent-return)	要求 <code>return</code> 语句要么总是指定返回的值，要么不指定
🔧	curly (curly)	强制所有控制语句使用一致的括号风格
	default-case (default-case)	要求 <code>switch</code> 语句中有 <code>default</code> 分支
🔧	dot-location (dot-location)	强制在点号之前和之后一致的换行
🔧	dot-notation (dot-notation)	强制尽可能地使用点号
🔧	eql (eql)	要求使用 <code>===</code> 和 <code>!==</code>
	guard-for-in (guard-for-in)	要求 <code>for-in</code> 循环中有一个 <code>if</code> 语句
	max-classes-per-file (max-classes-per-file)	强制每个文件中包含的类的最大数量
	no-alert (no-alert)	禁用 <code>alert</code> 、 <code>confirm</code> 和 <code>prompt</code>
	no-caller (no-caller)	禁用 <code>arguments.caller</code> 或 <code>arguments.callee</code>
✓	no-case-declarations (no-case-declarations)	不允许在 <code>case</code> 子句中使用词法声明
🔧	no-div-regex (no-div-regex)	禁止除法操作符显式的出现在正则表达式开始的位置
🔧	no-else-return (no-else-return)	禁止 <code>if</code> 语句中 <code>return</code> 语句之后有 <code>else</code> 块
	no-empty-function (no-empty-function)	禁止出现空函数
✓	no-empty-pattern (no-empty-pattern)	禁止使用空解构模式
	no-eq-null (no-eq-null)	禁止在没有类型检查操作符的情况下与 <code>null</code> 进行比较
	no-eval (no-eval)	禁用 <code>eval()</code>
	no-extend-native (no-extend-native)	禁止扩展原生类型
🔧	no-extra-bind (no-extra-bind)	禁止不必要的 <code>.bind()</code> 调用

	no-extra-label (no-extra-label)	禁用不必要的标签
✓	no-fallthrough (no-fallthrough)	禁止 case 语句落空
	no-floating-decimal (no-floating-decimal)	禁止数字字面量中使用前导和末尾小数点
✓	no-global-assign (no-global-assign)	禁止对原生对象或只读的全局对象进行赋值
	no-implicit-coercion (no-implicit-coercion)	禁止使用短符号进行类型转换
	no-implicit-globals (no-implicit-globals)	禁止在全局范围内使用变量声明和 function 声明
	no-implied-eval (no-implied-eval)	禁止使用类似 eval() 的方法
	no-invalid-this (no-invalid-this)	禁止 this 关键字出现在类和类对象之外
	no-iterator (no-iterator)	禁用 __iterator__ 属性
	no-labels (no-labels)	禁用标签语句
	no-lone-blocks (no-lone-blocks)	禁用不必要的嵌套块
	no-loop-func (no-loop-func)	禁止在循环语句中出现包含不安全引用的函数声明
	no-magic-numbers (no-magic-numbers)	禁用魔术数字
	no-multi-spaces (no-multi-spaces)	禁止使用多个空格
	no-multi-str (no-multi-str)	禁止使用多行字符串
	no-new (no-new)	禁止使用 new 以避免产生副作用
	no-new-func (no-new-func)	禁止对 Function 对象使用 new 操作符
	no-new-wrappers (no-new-wrappers)	禁止对 String, Number 和 Boolean 使用 new 操作符

✓	no-octal (no-octal)	禁用八进制字面量
	no-octal-escape (no-octal-escape)	禁止在字符串中使用八进制转义序列
	no-param-reassign (no-param-reassign)	禁止对 <code>function</code> 的参数进行重新赋值
	no-__proto__ (no-__proto__)	禁用 <code>__proto__</code> 属性
✓	no-redeclare (no-redeclare)	禁止多次声明同一变量
	no-restricted-properties (no-restricted-properties)	禁止使用对象的某些属性
	no-return-assign (no-return-assign)	禁止在 <code>return</code> 语句中使用赋值语句
	no-return-await (no-return-await)	禁用不必要的 <code>return await</code>
	no-script-url (no-script-url)	禁止使用 <code>javascript: url</code>
✓	no-self-assign (no-self-assign)	禁止自我赋值
	no-self-compare (no-self-compare)	禁止自身比较
	no-sequences (no-sequences)	禁用逗号操作符
	no-throw-literal (no-throw-literal)	禁止抛出异常字面量
	no-unmodified-loop-condition (no-unmodified-loop-condition)	禁用一成不变的循环条件
	no-unused-expressions (no-unused-expressions)	禁止出现未使用过的表达式
✓ 🔧	no-unused-labels (no-unused-labels)	禁用出现未使用过的标
	no-useless-call (no-useless-call)	禁止不必要的 <code>.call()</code> 和 <code>.apply()</code>

✓	no-useless-catch (no-useless-catch)	禁止不必要的 catch 子句
	no-useless-concat (no-useless-concat)	禁止不必要的字符串字面量或模板字面量的连接
✓	no-useless-escape (no-useless-escape)	禁用不必要的转义字符
🔧	no-useless-return (no-useless-return)	禁止多余的 return 语句
	no-void (no-void)	禁用 void 操作符
	no-warning-comments (no-warning-comments)	禁止在注释中使用特定的警告术语
✓	no-with (no-with)	禁用 with 语句
	prefer-named-capture-group (prefer-named-capture-group)	建议在正则表达式中使用命名捕获组
	prefer-promise-reject-errors (prefer-promise-reject-errors)	要求使用 Error 对象作为 Promise 拒绝的原因
	radix (radix)	强制在 parseInt() 使用基数参数
	require-await (require-await)	禁止使用不带 await 表达式的 async 函数
	require-unicode-regexp (require-unicode-regexp)	强制在 RegExp 上使用 u 标志
	vars-on-top (vars-on-top)	要求所有的 var 声明出现在它们所在的作用域顶部
🔧	wrap-iife (wrap-iife)	要求 IIFE 使用括号括起来
🔧	yoda (yoda)	要求或禁止 “Yoda” 条件

Strict Mode

该规则与使用严格模式和严格模式指令有关：



strict (strict)

要求或禁止使用严格模式指令

Variables

这些规则与变量声明有关：

	init-declarations (init-declarations)	要求或禁止 var 声明中的初始化
✓	no-delete-var (no-delete-var)	禁止删除变量
	no-label-var (no-label-var)	不允许标签与变量同名
	no-restricted-globals (no-restricted-globals)	禁用特定的全局变量
	no-shadow (no-shadow)	禁止变量声明与外层作用域的变量同名
✓	no-shadow-restricted-names (no-shadow-restricted-names)	禁止将标识符定义为受限的名字
✓	no-undef (no-undef)	禁用未声明的变量，除非它们在 <code>/*global */</code> 注释中被提到
	no-undef-init (no-undef-init)	禁止将变量初始化为 undefined
	no-undefined (no-undefined)	禁止将 undefined 作为标识符
✓	no-unused-vars (no-unused-vars)	禁止出现未使用过的变量
	no-use-before-define (no-use-before-define)	禁止在变量定义之前使用它们

Node.js and CommonJS

这些规则是关于Node.js 或 在浏览器中使用CommonJS 的：

	callback-return (callback-return)	强制数组方法的回调函数中有 return 语句
--	-----------------------------------	-------------------------







global-require (global-require)	要求 <code>require()</code> 出现在顶层模块作用域中
handle-callback-err (handle-callback-err)	要求回调函数中有容错处理
no-buffer-constructor (no-buffer-constructor)	禁用 <code>Buffer()</code> 构造函数
no-mixed-requires (no-mixed-requires)	禁止混合常规变量声明和 <code>require</code> 调用
no-new-require (no-new-require)	禁止调用 <code>require</code> 时使用 <code>new</code> 操作符
no-path-concat (no-path-concat)	禁止对 <code>__dirname</code> 和 <code>__filename</code> 进行字符串连接
no-process-env (no-process-env)	禁用 <code>process.env</code>
no-process-exit (no-process-exit)	禁用 <code>process.exit()</code>
no-restricted-modules (no-restricted-modules)	禁用通过 <code>require</code> 加载的指定模块
no-sync (no-sync)	禁用同步方法






Stylistic Issues


这些规则是关于风格指南的，而且是非常主观的：





 <code>array-bracket-newline</code> (array-bracket-newline)	在数组开括号后和闭括号前强制换行
 <code>array-bracket-spacing</code> (array-bracket-spacing)	强制数组方括号中使用一致的空格
 <code>array-element-newline</code> (array-element-newline)	强制数组元素间出现换行
 <code>block-spacing</code> (block-spacing)	禁止或强制在代码块中开括号前和闭括号后有空格
 <code>brace-style</code> (brace-style)	强制在代码块中使用一致的大括号风格

	camelcase (camelcase)	强制使用骆驼拼写法命名约定
	capitalized-comments (capitalized-comments)	强制或禁止对注释的第一个字母大写
	comma-dangle (comma-dangle)	要求或禁止末尾逗号
	comma-spacing (comma-spacing)	强制在逗号前后使用一致的空格
	comma-style (comma-style)	强制使用一致的逗号风格
	computed-property-spacing (computed-property-spacing)	强制在计算的属性的方括号中使用一致的空格
	consistent-this (consistent-this)	当获取当前执行环境的上下文时，强制使用一致的命名
	eol-last (eol-last)	要求或禁止文件末尾存在空行
	func-call-spacing (func-call-spacing)	要求或禁止在函数标识符和其调用之间有空格
	func-name-matching (func-name-matching)	要求函数名与赋值给它们的变量名或属性名相匹配
	func-names (func-names)	要求或禁止使用命名的 <code>function</code> 表达式
	func-style (func-style)	强制一致地使用 <code>function</code> 声明或表达式
	function-paren-newline (function-paren-newline)	强制在函数括号内使用一致的换行
	id-blacklist (id-blacklist)	禁用指定的标识符
	id-length (id-length)	强制标识符的最小和最大长度
	id-match (id-match)	要求标识符匹配一个指定的正则表达式
	implicit-arrow-linebreak (implicit-arrow-linebreak)	强制隐式返回的箭头函数体的位置
	indent (indent)	强制使用一致的缩进
	jsx-quotes (jsx-quotes)	强制在 JSX 属性中一致地使用双引号或单引号

 key-spacing (key-spacing)	强制在对象字面量的属性中键和值之间使用一致的间距
 keyword-spacing (keyword-spacing)	强制在关键字前后使用一致的空格
line-comment-position (line-comment-position)	强制行注释的位置
 linebreak-style (linebreak-style)	强制使用一致的换行风格
 lines-around-comment (lines-around-comment)	要求在注释周围有空行
 lines-between-class-members (lines-between-class-members)	要求或禁止类成员之间出现空行
max-depth (max-depth)	强制可嵌套的块的最大深度
max-len (max-len)	强制一行的最大长度
max-lines (max-lines)	强制最大行数
max-lines-per-function (max-lines-per-function)	强制函数最大代码行数
max-nested-callbacks (max-nested-callbacks)	强制回调函数最大嵌套深度
max-params (max-params)	强制函数定义中最多允许的参数数量
max-statements (max-statements)	强制函数块最多允许的的语句数量
max-statements-per-line (max-statements-per-line)	强制每一行中所允许的最大语句数量
 multiline-comment-style (multiline-comment-style)	强制对多行注释使用特定风格
multiline-ternary (multiline-ternary)	要求或禁止在三元操作数中间换行
new-cap (new-cap)	要求构造函数首字母大写








	new-parens (new-parens)	强制或禁止调用无参构造函数时有圆括号
	newline-per-chained-call (newline-per-chained-call)	要求方法链中每个调用都有一个换行符
	no-array-constructor (no-array-constructor)	禁用 Array 构造函数
	no-bitwise (no-bitwise)	禁用按位运算符
	no-continue (no-continue)	禁用 continue 语句
	no-inline-comments (no-inline-comments)	禁止在代码后使用内联注释
	no-lonely-if (no-lonely-if)	禁止 if 作为唯一的语句出现在 else 语句中
	no-mixed-operators (no-mixed-operators)	禁止混合使用不同的操作符
	no-mixed-spaces-and-tabs (no-mixed-spaces-and-tabs)	禁止空格和 tab 的混合缩进
	no-multi-assign (no-multi-assign)	禁止连续赋值
	no-multiple-empty-lines (no-multiple-empty-lines)	禁止出现多行空行
	no-negated-condition (no-negated-condition)	禁用否定的表达式
	no-nested-ternary (no-nested-ternary)	禁用嵌套的三元表达式
	no-new-object (no-new-object)	禁用 Object 的构造函数
	no-plusplus (no-plusplus)	禁用一元操作符 ++ 和 --
	no-restricted-syntax (no-restricted-syntax)	禁用特定的语法
	no-tabs (no-tabs)	禁用 tab
	no-ternary (no-ternary)	禁用三元操作符

 <code>no-trailing-spaces (no-trailing-spaces)</code>	禁用行尾空格
<code>no-underscore-dangle (no-underscore-dangle)</code>	禁止标识符中有悬空下划线
 <code>no-unneeded-ternary (no-unneeded-ternary)</code>	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
 <code>no-whitespace-before-property (no-whitespace-before-property)</code>	禁止属性前有空白
 <code>nonblock-statement-body-position (nonblock-statement-body-position)</code>	强制单个语句的位置
 <code>object-curly-newline (object-curly-newline)</code>	强制大括号内换行符的一致性
 <code>object-curly-spacing (object-curly-spacing)</code>	强制在大括号中使用一致的空格
 <code>object-property-newline (object-property-newline)</code>	强制将对象的属性放在不同的行上
 <code>one-var (one-var)</code>	强制函数中的变量要么一起声明要么分开声明
 <code>one-var-declaration-per-line (one-var-declaration-per-line)</code>	要求或禁止在变量声明周围换行
 <code>operator-assignment (operator-assignment)</code>	要求或禁止在可能的情况下使用简化的赋值操作符
 <code>operator-linebreak (operator-linebreak)</code>	强制操作符使用一致的换行符
 <code>padded-blocks (padded-blocks)</code>	要求或禁止块内填充
 <code>padding-line-between-statements (padding-line-between-statements)</code>	要求或禁止在语句间填充空行
 <code>prefer-object-spread (prefer-object-spread)</code>	禁止使用以对象字面量作为第一个参数的 <code>Object.assign</code> ，优先使用对象扩展。

 quote-props (quote-props)	要求对象字面量属性名称用引号括起来
 quotes (quotes)	强制使用一致的反勾号、双引号或单引号
 semi (semi)	要求或禁止使用分号代替 ASI
 semi-spacing (semi-spacing)	强制分号之前和之后使用一致的空格
 semi-style (semi-style)	强制分号的位置
sort-keys (sort-keys)	要求对象属性按序排列
 sort-vars (sort-vars)	要求同一个声明块中的变量按顺序排列
 space-before-blocks (space-before-blocks)	强制在块之前使用一致的空格
 space-before-function-paren (space-before-function-paren)	强制在 <code>function</code> 的左括号之前使用一致的空格
 space-in-parens (space-in-parens)	强制在圆括号内使用一致的空格
 space-infix-ops (space-infix-ops)	要求操作符周围有空格
 space-unary-ops (space-unary-ops)	强制在一元操作符前后使用一致的空格
 spaced-comment (spaced-comment)	强制在注释中 <code>//</code> 或 <code>/*</code> 使用一致的空格
 switch-colon-spacing (switch-colon-spacing)	强制在 <code>switch</code> 的冒号左右有空格
 template-tag-spacing (template-tag-spacing)	要求或禁止在模板标记和它们的字面量之间有空格
 unicode-bom (unicode-bom)	要求或禁止 Unicode 字节顺序标记 (BOM)
 wrap-regex (wrap-regex)	要求正则表达式被括号括起来

ECMAScript 6

这些规则只与 ES6 有关, 即通常所说的 ES2015:

	arrow-body-style (arrow-body-style)	要求箭头函数体使用大括号
	arrow-parens (arrow-parens)	要求箭头函数的参数使用圆括号
	arrow-spacing (arrow-spacing)	强制箭头函数的箭头前后使用一致的空格
✓	constructor-super (constructor-super)	要求在构造函数中有 <code>super()</code> 的调用
	generator-star-spacing (generator-star-spacing)	强制 generator 函数中 <code>*</code> 号周围使用一致的空格
✓	no-class-assign (no-class-assign)	禁止修改类声明的变量
	no-confusing-arrow (no-confusing-arrow)	禁止在可能与比较操作符相混淆的地方使用箭头函数
✓	no-const-assign (no-const-assign)	禁止修改 <code>const</code> 声明的变量
✓	no-dupe-class-members (no-dupe-class-members)	禁止类成员中出现重复的名称
	no-duplicate-imports (no-duplicate-imports)	禁止重复模块导入
✓	no-new-symbol (no-new-symbol)	禁止 <code>Symbol</code> <code>new</code> 操作符和 <code>new</code> 一起使用
	no-restricted-imports (no-restricted-imports)	禁止使用指定的 <code>import</code> 加载的模块
✓	no-this-before-super (no-this-before-super)	禁止在构造函数中，在调用 <code>super()</code> 之前使用 <code>this</code> 或 <code>super</code>
	no-useless-computed-key (no-useless-computed-key)	禁止在对象中使用不必要的计算属性
	no-useless-constructor (no-useless-constructor)	禁用不必要的构造函数
	no-useless-rename (no-useless-rename)	禁止在 <code>import</code> 和 <code>export</code> 和解构赋值时将引用重命名为相同的名字

	no-var (no-var)	要求使用 <code>let</code> 或 <code>const</code> 而不是 <code>var</code>
	object-shorthand (object-shorthand)	要求或禁止对象字面量中方法和属性使用简写语法
	prefer-arrow-callback (prefer-arrow-callback)	要求回调函数使用箭头函数
	prefer-const (prefer-const)	要求使用 <code>const</code> 声明那些声明后不再被修改的变量
	prefer-destructuring (prefer-destructuring)	优先使用数组和对象解构
	prefer-numeric-literals (prefer-numeric-literals)	禁用 <code>parseInt()</code> 和 <code>Number.parseInt()</code> ，使用二进制，八进制和十六进制字面量
	prefer-rest-params (prefer-rest-params)	要求使用剩余参数而不是 <code>arguments</code>
	prefer-spread (prefer-spread)	要求使用扩展运算符而非 <code>.apply()</code>
	prefer-template (prefer-template)	要求使用模板字面量而非字符串连接
✓	require-yield (require-yield)	要求 <code>generator</code> 函数内有 <code>yield</code>
	rest-spread-spacing (rest-spread-spacing)	强制剩余和扩展运算符及其表达式之间有空格
	sort-imports (sort-imports)	强制模块内的 <code>import</code> 排序
	symbol-description (symbol-description)	要求 <code>symbol</code> 描述
	template-curly-spacing (template-curly-spacing)	要求或禁止模板字符串中的嵌入表达式周围空格的使用
	yield-star-spacing (yield-star-spacing)	强制在 <code>yield*</code> 表达式中 <code>*</code> 周围使用空格

Deprecated

Deprecated rule

Replaced by

Deprecated rule	Replaced by
indent-legacy (indent-legacy)	indent (indent)
lines-around-directive (lines-around-directive)	padding-line-between-statements (padding-line-between-statements)
newline-after-var (newline-after-var)	padding-line-between-statements (padding-line-between-statements)
newline-before-return (newline-before-return)	padding-line-between-statements (padding-line-between-statements)
no-catch-shadow (no-catch-shadow)	no-shadow (no-shadow)
no-native-reassign (no-native-reassign)	no-global-assign (no-global-assign)
no-negated-in-lhs (no-negated-in-lhs)	no-unsafe-negation (no-unsafe-negation)
no-spaced-func (no-spaced-func)	func-call-spacing (func-call-spacing)
prefer-reflect (prefer-reflect)	(no replacement)
require-jsdoc (require-jsdoc)	(no replacement)
valid-jsdoc (valid-jsdoc)	(no replacement)

Removed

These rules from older versions of ESLint (before the deprecation policy (/docs/user-guide/rule-deprecation) existed) have been replaced by newer rules:

Removed rule	Replaced by
generator-star (generator-star)	generator-star-spacing (generator-star-spacing)
global-strict (global-strict)	strict (strict)
no-arrow-condition (no-arrow-condition)	no-confusing-arrow (no-confusing-arrow), no-constant-condition (no-constant-condition)

Removed rule	Replaced by
no-comma-dangle (no-comma-dangle)	comma-dangle (comma-dangle)
no-empty-class (no-empty-class)	no-empty-character-class (no-empty-character-class)
no-empty-label (no-empty-label)	no-labels (no-labels)
no-extra-strict (no-extra-strict)	strict (strict)
no-reserved-keys (no-reserved-keys)	quote-props (quote-props)
no-space-before-semi (no-space-before-semi)	semi-spacing (semi-spacing)
no-wrap-func (no-wrap-func)	no-extra-parens (no-extra-parens)
space-after-function-name (space-after-function-name)	space-before-function-paren (space-before-function-paren)
space-after-keywords (space-after-keywords)	keyword-spacing (keyword-spacing)
space-before-function-parentheses (space-before-function-parentheses)	space-before-function-paren (space-before-function-paren)
space-before-keywords (space-before-keywords)	keyword-spacing (keyword-spacing)
space-in-brackets (space-in-brackets)	object-curly-spacing (object-curly-spacing), array-bracket-spacing (array-bracket-spacing)
space-return-throw-case (space-return-throw-case)	keyword-spacing (keyword-spacing)
space-unary-word-ops (space-unary-word-ops)	space-unary-ops (space-unary-ops)
spaced-line-comment (spaced-line-comment)	spaced-comment (spaced-comment)

Edit this page (<https://github.com/eslint/eslint.github.io/edit/master/docs/rules/index.md>)

Mailing List (<https://groups.google.com/group/eslint>) GitHub (<https://github.com/eslint/eslint>)

Twitter (<https://twitter.com/geteslint>) Chat Room (<https://gitter.im/eslint/eslint>)

Copyright JS Foundation and other contributors, <https://js.foundation/> (<https://js.foundation>)