

02

OPEN ORIENTED

凹凸实验室

prettier-代码格式美化神器

2018.03.12

prettier是什么

前端格式化代码的一个整体方案

不能修正语法错误

How does it compare to ESLint/TSLint/stylelint, etc.?

Linters have two categories of rules:

Formatting rules: eg: `max-len`, `no-mixed-spaces-and-tabs`, `keyword-spacing`, `comma-style`...

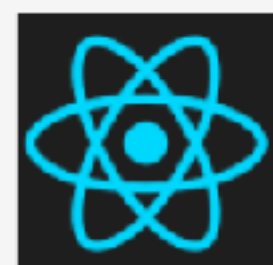
Prettier alleviates the need for this whole category of rules! Prettier is going to reprint the entire program from scratch in a consistent way, so it's not possible for the programmer to make a mistake there anymore :)

Code-quality rules: eg `no-unused-vars`, `no-extra-bind`, `no-implicit-globals`, `prefer-promise-reject-errors`...

Prettier does nothing to help with those kind of rules. They are also the most important ones provided by linters as they are likely to catch real bugs with your code!

Who's Using This?

A few of the [many projects](#) using Prettier



- 1、使用环境：编辑器插件、CLI、nodejs调用、git pre-commit、eslint plugins
- 2、json配置语法，支持读取editorconfig
- 3、支持 JavaScript, CSS, SCSS, Vue, JSON, Less, TypeScript, GraphQL, Markdown

.prettierrc文件

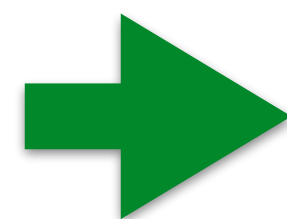
package.json里的prettier

prettier.config.js文件

singleQuote	false/true，是否单引号
trailingComma	none/es5/all，为对象、数组最后一行添加逗号
arrowParens	avoid/always，为箭头函数添加括号
requirePragma	false/true，检测到注释关键字，保存时候自动格式化
tabWidth	2，tab宽度。从editorconfig读取
bracketSpacing	false/true，对象前后加空格，{ a: 1 }
semi	true/false，语句结尾加分号
printWidth	80，自动换行宽度
useTabs	false/true，使用tab代替空格
rangeStart	
rangeEnd	
parser	
insertPragma	

trailingComma的好处

...	...	@@ -1,4 +1,5 @@
1	1	const fruits = [
2	2	'apples',
3		- 'banana'
	3	+ 'banana',
	4	+ 'pineapple'
4	5];



...	...	@@ -1,4 +1,5 @@
1	1	const fruits = [
2	2	'apples',
3	3	'banana',
	4	+ 'pineapple',
4	5];

```
const fruits = [  
  'pear',  
  'banana'  
];
```



```
const fruits = [  
  'pear',  
];
```



```
const fruits = [  
  'pear',  
];
```



```
const fruits = [  
  'pear'|  
];
```

```
const myFruits = [  
  'apple',  
];
```

```
const myFruits = [  
  'banana'  
  'apple',  
];
```

```
const myFruits = [  
  'banana',  
  'apple',  
];
```

```
const myFruits = [  
  'banana',  
  'apple',  
];
```


使用方法-配置IDE

- 1、安装IDE插件 (VSCode / SublimeText)
 - 2、“command + ,”打开设置。把editor.formatOnSave、auto_format_on_save设置为true
 - 3、文件头第一个注释块加上@prettier
- done! 保存文件时自动格式化

```
/**
 * @prettier
 * @author yuluyu
 * @deprecated 2018.01.24
 * @description 领券中心
 */
```

使用方法-使用git hook

02

1、npm i husky prettier pretty-quick —save-dev

2、设置钩子

```
"scripts": {  
  "precommit": "pretty-quick --staged"  
},
```

3、文件头第一个注释块加上@prettier

4、git commit

```
yuluyu@yuluyu-mbp > ~/work/git/wxapp_gitjd > feature_test_prettier ●+ > git commit  
husky > npm run -s precommit (node v8.9.3)  
  
🔍 Finding changed files since git revision 54f671b.  
🔗 Found 2 changed files.  
🔧 Fixing up pages/market/demo/demo.js.  
✅ Everything is awesome!
```

有什么效果？

方法链式调用对齐

加分号

强制单引号

长语句换行

对象、操作符前后空格

4空格、转换tab为空格 (editconfig)

- 1、某些写法格式化后可能会看不习惯
- 2、IDE插件：根目录没设置`.prettierrc`保存时候也会根据默认规则格式化代码（改为手动格式化）
- 3、git hook缺点：node $\geq 6.x$
- 4、不同类型代码（js/css等）用同一套配置，比如都是单引号

也许，你并不喜欢这种风格。比如，我不喜欢`else`放在大括号后面，也不喜欢把链式方法全部写在同一行。但是，我发现使用**Prettier**有很多益处：

- 几乎不需要做决定，因为Prettier的配置选项很少。
- 团队成员不需要为规则去争论。
- 开源代码开发者不需要去学习项目的代码风格。
- 不需要去修复ESLint报告的风格问题。
- 保存文件的时候可以自动统一风格。

One more thing

husky & lint-staged + prettier

```
feature_test_prettier
npm i
// 修改任意js文件
git add .
git commit // 会自动执行hook
```

配置项: <https://prettier.io/docs/en/configuration.html>

SublimeText插件: <https://packagecontrol.io/packages/JsPrettier#install-jsprettier-via-package-control>

VSCode插件: <https://github.com/prettier/prettier-vscode>

关于对象、数组使用逗号多行的分析 <https://medium.com/@nikgraf/why-you-should-enforce-dangling-commas-for-multiline-statements-d034c98e36f8>

我为什么推荐Prettier来统一代码风格

<https://blog.fundebug.com/2017/10/23/format-code-use-Prettier/>

T H A N K S
FOR YOUR WATCHING

