

# 一款超人气代码格式化工具prettier

## 一、prettier 介绍

prettier / prettier

Watch 396 Star

Code Issues 580 Pull requests 41 Projects 0 Wiki Insights

CONTENTS

1. 一、prettier 介绍

2. 二、prettier 的使用

2.1. 1、使用编辑器的插件

2.2. 2、使用脚本的方式

3. 三、以上两种方式对比

官方自己介绍说，prettier是一款强势武断的代码格式化工具，它几乎移除了编辑器本身所有的对代码的操作格式，然后重新显示。就是为了让所有用这套规则的人有完全相同的代码。在团队协作开发的时候更是体现出它的优势。与eslint，tslint等各种格式化工具不同的是，prettier只关心代码格式化，而不关心语法问题。

prettier 的优势也很明显，它支持 HTML/JS/JSSX/TS/JSON/CSS/SCSS/LESS/VUE 等主流文件格式。下面这张图可以很好的进行说明：

Works with the Tools You Use

JS

JavaScript  
JSX  
Flow  
TypeScript  
JSON

CSS  
Less  
SCSS  
styled-components  
styled-jss

Markdown  
CommonMark  
GitHub-Flavored Markdown  
MDX

Community Plugins  
Elm (via elm-format)  
Java  
PHP  
PostgreSQL  
Ruby  
Swift  
TOML

HTML  
Vue  
Angular

GraphQL  
GraphQL Schemas

YAML

YAML

也支持目前市面上所有主流的编辑器：

Editor Support

Atom  
prettier-atom  
mprettier  
miniprettier

Emacs  
prettier-emacs

Espresso  
espresso-prettier

Sublime Text  
JsPrettier

Vim  
neoformat  
ale  
vim-prettier

Visual Studio  
JavaScriptPrettier

VS Code  
prettier-vscode  
unibeautify-vscode

WebStorm  
Built-in support

## 二、prettier 的使用

prettier的使用可分为两种方式：

### 1、使用编辑器的插件

使用编辑器插件是最为方便的一种方法，编写完代码，只需要一键即可格式化编写的代码，非常方便。这里已vscode为例进行说明，下面的配置是我自己的对于HTML/CSS/JS/LESS文件的prettier格式化规则：



```
"[html]": {
  "editor.defaultFormatter": "esbenp.prettier-vscode"
},
"[css]": {
  "editor.defaultFormatter": "esbenp.prettier-vscode"
},
"[less]": {
  "editor.defaultFormatter": "esbenp.prettier-vscode"
},
"[javascript]": {
  "editor.defaultFormatter": "esbenp.prettier-vscode"
},

/* prettier的配置 */
"prettier.printWidth": 100, // 超过最大值换行
"prettier.tabWidth": 4, // 缩进字节数
"prettier.useTabs": false, // 缩进不使用tab, 使用空格
"prettier.semi": true, // 句尾添加分号
"prettier.singleQuote": true, // 使用单引号代替双引号
"prettier.proseWrap": "preserve", // 默认值。因为使用了一些折行敏感型的渲染器（如GitHub comment）而按照markdown文本样式
"prettier.arrowParens": "avoid", // (x) => {} 箭头函数参数只有一个时是否要有小括号。avoid: 省略括号
"prettier.bracketSpacing": true, // 在对象, 数组括号与文字之间加空格 "{ foo: bar }"
"prettier.disableLanguages": ["vue"], // 不格式化vue文件, vue文件的格式化单独设置
"prettier.endOfLine": "auto", // 结尾是 \n \r \n\r auto
"prettier.eslintIntegration": false, // 不让prettier使用eslint的代码格式进行校验
"prettier.htmlWhitespaceSensitivity": "ignore",
"prettier.ignorePath": ".prettierignore", // 不使用prettier格式化的文件填写在项目的.prettierignore文件中
"prettier.jsxBracketSameLine": false, // 在jsx中把'>' 是否单独放一行
"prettier.jsxSingleQuote": false, // 在jsx中使用单引号代替双引号
"prettier.parser": "babylon", // 格式化的解析器, 默认是babylon
"prettier.requireConfig": false, // Require a 'prettierconfig' to format prettier
"prettier.stylelintIntegration": false, // 不让prettier使用stylelint的代码格式进行校验
"prettier.trailingComma": "es5", // 在对象或数组最后一个元素后面是否加逗号（在ES5中加尾逗号）
"prettier.tslintIntegration": false // 不让prettier使用tslint的代码格式进行校验
}
```

## CONTENTS

- 1. 一、prettier 介绍
- 2. 二、prettier 的使用
  - 2.1. 1、使用编辑器的插件
  - 2.2. 2、使用脚本的方式
- 3. 三、以上两种方式对比

上面只是一些基本的语言的格式化规范, prettier 每一个属性的配置都有详细的说明, 大家可以根据自己的情况进行调整。

相信每个在vscode上编写vue的都会下载 Vetur 插件, 它目前是 vscode 上面最好用的一款vue插件。现在要说的是, 如何使用 prettier 格式化vue的代码。你没法使用类似格式化html/css/js的方式来格式化vue格式的代码, 像下面这样子的:

```
{
  "[vue]": {
    "editor.defaultFormatter": "esbenp.prettier-vscode"
  }
}
```

Copy

这样prettier是不认识的。不过幸运的是, Vetur插件内部默认使用prettier进行格式化的, 但是由于Vetur的默认格式化配置与我们期望的有所出入, 所以我们需要单独对Vetur的prettier进行配置, 如下:

```
{
  "vetur.format.defaultFormatter.html": "prettier",
  "vetur.format.defaultFormatter.js": "prettier",
  "vetur.format.defaultFormatter.less": "prettier",
  "vetur.format.defaultFormatterOptions": {
    "prettier": {
      "printWidth": 160,
      "singleQuote": true, // 使用单引号
      "semi": true, // 末尾使用分号
      "tabWidth": 4,
      "arrowParens": "avoid",
      "bracketSpacing": true,
      "proseWrap": "preserve" // 代码超出是否要换行 preserve保留
    }
  }
}
```

Copy

这些配置是不会和之前配置的prettier规则冲突的。

值得提一句的是，Vetur对于html文件默认使用的是 prettyhtml，但是由于prettier也可以支持html的格式化，prettier对全语言的格式化是比较简洁的，也希望prettier能够对更多的语言进行支持。

## 2、使用脚本的方式

这种方式就是使用prettier指令在命令行窗口对单一文件进行格式化。

首先需要安装prettier全局指令：

```
npm install -g prettier
```

Copy

可以使用 `prettier -v` 检查是否安装完成。

安装好之后，使用下面指令对xxx.js文件进行格式化（使用的是prettier默认的配置规则）。

```
// //prettier--write <文件路劲+文件名>

prettier --write ./xxx.js
```

Copy

当然，默认的配置规则是不符合我们的需求的，我们需要自定义配置规则。

书写自定义规则的文件需要是下面几种文件和格式：

- .prettierrc 文件，支持yaml和json格式；或者加上 .yaml/.yml/.json 后缀名
- .prettierrc.toml 文件（当为toml格式的时候，后缀是必须的）
- prettier.config.js 或者 .prettierrc.js，需要返回一个对象
- package.json文件中加上"prettier"属性

每种文件的书写格式如下：

JSON

```
{
  "trailingComma": "es5",
  "tabWidth": 4,
  "semi": false,
  "singleQuote": true
}
```

Copy

JS

```
// prettier.config.js or .prettierrc.js
module.exports = {
  trailingComma: "es5",
  tabWidth: 4,
  semi: false,
  singleQuote: true
};
```

Copy

YAML

```
# .prettierrc or .prettierrc.yaml
trailingComma: "es5"
tabWidth: 4
semi: false
singleQuote: true
```

TOML

CONTENTS

1. 一、prettier 介绍

2. 二、prettier 的使用

2.1. 1、使用编辑器的插件

2.2. 2、使用脚本的方式

3. 三、以上两种方式对比

```
tabWidth = 4
semi = false
singleQuote = true
```

prettier 查找配置的方式首先会找当前目录下，使用以下指令格式化代码：

```
//prettier --config --write <文件路劲+文件名>

prettier --config --write ./xxx.js
```

如果prettier在当前目录找不到配置文件，会一直向上级目录查找，直到找到或找不到。如果我们配置文件放在别的地方，则需要手工指定配置文件的路径：

```
// prettier --config <配置文件路径+文件名> --write <文件路劲+文件名>

prettier --config ./prettier/.prettierrc --write ./xxx.js
```

Copy

如果觉得每次格式化一个文件比较麻烦，可以使用下面的指令，一次格式化所有文件：

```
prettier --config ./prettier/.prettierrc --write './*.{ts,js,css,json}'
```

Copy

我们一般使用这种方式的时候，就把这个配置文件写在项目根路径下，然后使用命令行一次性格式化项目下的所有文件。

### 三、以上两种方式对比

上面两种方式各有优劣，我们来分析一下各自的使用场景和一些问题：

第一种方式其实适合个人开发，第二种方式适合团队开发。

至于为什么这么说，就要考虑到二者的优先级问题了。上面两种方式如果同时存在的话，会有优先级的问题。  
.prettierrc 的优先级会高于在vscode全局配置settings.json中格式化配置的优先级

也就是说，如果你在一个项目中有 .prettierrc 配置文件，然后你又在settings.json也配置了格式化规则，那么当你在vscode编辑器中 对一个文件点击鼠标右键[格式化文档]的时候，格式化规则会以 .prettierrc 为准。

所以，由于编辑器settings.json每个人的设置可能都不一样，要求每个人统一设置也不方便操作，而嵌入在项目中的配置文件则可以随着项目到达各个开发者，而且会覆盖每个开发者的不同代码喜好，真正做到团队代码统一的效果。

以上就是所有我对prettier理解的内容，希望对你有帮助。更多精彩内容可以关注我的微信公众号[前端队长]，我们一同成长，一同 领略技术与生活“落霞与孤鹜齐飞，秋水共长天一色”的美好。

参考链接：

- <https://blog.csdn.net/wxl1555/article/details/82857830>
- <https://juejin.im/post/5bfcdee25188251d9e0c40f2>
- <https://segmentfault.com/a/1190000012909159>

作者：Daotin  
出处：<https://www.cnblogs.com/lvonve/p/10785682.html>  
版权：本文采用「署名-非商业性使用-相同方式共享 4.0 国际」知识共享许可协议进行许可。  
联系方式：Github，微信公众号：前端队长

分类: 技术综合  
标签: prettier, VSCode

推荐 0 反对 0



请先登录

博客园 首页 标签 归档 新随笔 管理

« 上一篇: [你可能不知道的css-doodle](#)  
» 下一篇: [你的代码好看吗](#)

posted @ 2019-04-28 18:25 Daotin 阅读(614) 评论(0) 刷新评论

登录后才能发表评论, 立即 [登录](#) 或 [注册](#), [访问 网站首页](#)

Copyright © 2021 Daotin  
Powered by .NET 5.0 on Kubernetes & Theme Silence v2.0.3

≡ CONTENTS ×

- 1. 一、prettier 介绍
- 2. 二、prettier 的使用
  - 2.1. 1、使用编辑器的插件
  - 2.2. 2、使用脚本的方式
- 3. 三、以上两种方式对比