# 官网

https://code.visualstudio.com/

# 下载地址

https://code.visualstudio.com/download

# 官方文档

https://vscode.js.cn/docs

https://jeasonstudio.gitbooks.io/vscode-cn-doc/content/

# 使用教程

https://zhuanlan.zhihu.com/p/113222681

# 快捷键

## ctrl+,

设置

## ctrl+shift+F

在文件中查找（全局）

## ctrl+P

到指定的文件

## ctrl+shift+N

新建窗口

## ctrl+shift+P

显示所有命令

## ctrl+alt+I

打开聊天

## ctrl+O

打开文件

## ctrl+K ctrl+O

打开文件夹

## ctrl+`

切换终端

## ctrl+shift+`

新建终端

## ctrl+向下箭头

滚动到下一条命令

## ctrl+向上箭头

滚动到上一条命令

## ctrl+J

切换面板

## ctrl+/

切换行注释

## shift+alt+A

切换块注释

## F2

重命名

## F5

开始调试

# 插件

## 开发Java的插件

### Extension Pack for Java

这个插件包含下面的这些插件

**Language Support for Java™ by Red Hat**

**Debugger for Java**

**Test Runner for Java**

**Maven for Java**

**Project Manager for Java**

**Visual Studio IntelliCode**

#### Language Support for Java™ by Red Hat

##### 功能

1、Java语法高亮

2、智能代码补全

3、错误检测和修复建议

4、代码格式化

5、重构功能

6、导入组织

##### 配置示例



#### Debugger for Java

##### 功能

1、断点调试

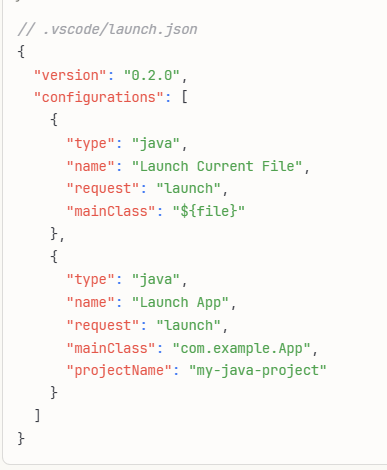
2、变量监视

3、调用栈查看

4、条件断点

5、远程调试

##### 使用方法



#### Test Runner for Java

#### Maven for Java

#### Project Manager for Java

## 开发Vue的插件

### ****Vetur****

Vetur 是VSCode中最受欢迎的Vue插件之一。它提供了语法高亮、智能感知、Emmet、linting/错误检查、格式化、自动补全、调试等功能。Vetur的的linting功能可以帮助您检查代码中的错误和不良实践，而自动补全功能则可以大大提高编写Vue代码的速度。

### ****Volar****

### ****Vue VSCode Snippets****

Vue VSCode Snippets插件提供了大量的Vue代码片段，可以帮助您快速编写常见的Vue代码结构，如组件、指令、混入等。这个插件极大地提高了编写Vue代码的效率。

### ****Vue 3 Snippets****

对于正在使用Vue 3的开发者来说，Vue 3 Snippets插件是一个不错的选择。它提供了针对Vue 3的代码片段，包括Composition API等。

### ****ESLint****

ESLint是一个强大的linting工具，可以帮助您检查JavaScript代码中的错误和不良实践。对于Vue项目来说，使用ESLint可以确保代码的一致性和可维护性。

### ****Prettier - Code formatter****

Prettier是一个流行的代码格式化工具，可以自动调整您的代码风格，使其符合一致的规范。在Vue项目中，使用Prettier可以确保代码风格的统一，减少不必要的格式争议。

### ****Debugger for Chrome（已废弃）****

Debugger for Chrome插件允许您在VSCode中直接调试Chrome浏览器中的代码。这对于Vue开发者来说非常有用，因为您可以在编写代码的同时实时查看和调试页面效果。



说明：推荐使用 [JavaScript Debugger](vscode-file://vscode-app/d:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)插件。

## Path Intellisense

### 作用

主要用于提供路径自动补全功能。

### 主要功能

#### 1、路径自动补全

当您在代码中输入文件路径时，插件会自动显示可用的文件和文件夹选项，大大提高了开发效率。

#### 2、支持多种引用方式

支持相对路径和绝对路径的补全，适用于JavaScript的import语句、HTML的src和href属性、CSS的url引用等多种场景。

#### 3、减少错误

手动输入文件路径很容易出现拼写错误、大小写错误或者忘记了确切的文件夹结构。Path Intellisense 通过提供准确的路径建议，可以有效避免这类错误，从而节省您的调试时间。

#### 4、提高效率

无需手动浏览文件树或记住复杂的路径结构，Path Intellisense 让您能够快速、准确地插入所需的文件路径，从而加快编码速度。

### 如何使用

1、开始输入路径：在您的代码文件中，当您需要输入文件路径时，例如在一个 import 语句中输入 ./ 或模块名的一部分，Path Intellisense 就会自动激活并显示建议列表。

2、**选择建议：** 使用上下箭头键浏览建议，按 Enter 键或 Tab 键选择并插入路径。

### 总结

Path Intellisense 插件通过提供快速、准确的文件路径自动补全功能，极大地简化了在 VSCode 中处理文件路径的过程，帮助开发者提高编码效率并减少潜在的路径错误。其丰富的配置选项和对项目配置文件的良好支持使其成为许多开发者工具箱中的必备插件。

## Error Lens

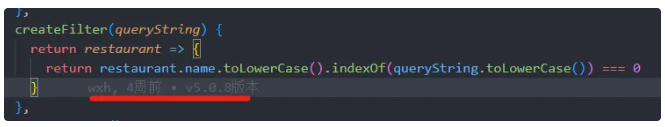
### 基本概念

Error Lens 是一个超实用的 VS Code 插件，它可以让你的代码错误和警告变得非常显眼。当你在代码中犯错时，这个插件会在你出错的那一行直接显示错误信息，而不仅仅是底部的状态栏。这种即时反馈让你可以快速发现并修复问题。再也不用一直盯着状态栏看错在哪里了。

## GitLens

### 基本概念

GitLens 是强大的Git集成插件，可以帮助我们实时查看Git提交信息，轻松进行代码溯源。安装GitLens 插件之后在每行代码的末尾，GitLens 都会公开最后一次提交的作者、提交的时间以及其他详细的备注信息。



如今AI 编程早已成为日常，很多人写的代码 80% 都靠 AI 完成。但 AI 不是全能的，团队协作时也容易出问题。

## TODO Highlight

### 作用

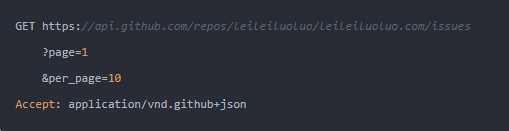
高亮 // TODO、// FIXME 等注释标签。



## REST Client

### 作用

直接在 VSCode 中发送 HTTP 请求，调试接口。和后端联调、测试 API 返回结果，我还是喜欢在 VSCode 内部直接请求，不必切到 Postman 或 curl。



将如上内容保存为一个.http文件后，REST Client 扩展会自动检测到如上内容，并在 GET 上面显示「Send Request」按钮；点击该按钮即可发送请求。

## stagewise

### 基本概念

Stagewise 是一个VS Code扩展插件，它连接Web浏览器与VS Code/Cursor环境中的AI代理。这个插件使开发者能够在浏览器中选择UI元素，提取包括截图在内的上下文信息，并将这些数据直接传送给AI编程助手

### 主要功能

1、元素选择：在Web应用中选择任意UI元素

2、实时评论：直接在页面元素上留下评论

3、AI智能处理：让AI代理根据上下文自动处理代码修改

4、截图上下文：自动捕获DOM元素、截图和元数据发送给AI代理

### 技术特性

1、开箱即用：无需复杂配置

2、高度可定制：支持自定义配置文件

3、MCP服务器连接：可连接到自己的MCP服务器

4、零影响：不影响项目打包大小

5、多框架支持：提供React、Vue、Svelte的示例

### 使用流程

1、启动开发服务器：运行你的前端项目

2、打开浏览器：访问你的本地开发地址

3、选择元素：在页面上点击任意UI元素

4、添加评论：描述你想要的修改

5、AI处理：AI代理会根据上下文自动修改代码

6、实时预览：在浏览器中查看修改效果

### 适用场景

1、前端开发：快速调整UI组件样式

2、原型迭代：基于视觉反馈快速修改

3、代码重构：通过可视化方式指导重构

4、团队协作：通过视觉标注传达修改需求

### 总结

Stagewise 本质上是一个革命性的前端开发工具，它将传统的"描述-编码"流程转变为"点击-描述-自动编码"的可视化开发体验，大大提高了前端开发的效率和精准度。

## Markdown Preview Mermaid Support

### 基本概念

**Markdown Preview Mermaid Support** 是 VS Code 的一个扩展插件，主要功能是让你在 VS Code 的 Markdown 预览里直接渲染 **Mermaid 图表**。

### 主要作用

1、让 VS Code 的 Markdown 预览支持Mermaid 语法。

2、直接在 .md 文件中写流程图、时序图、类图等，不需要切换到其他工具。

3、预览时即时更新，改完保存就能看到效果

### 支持的图表类型（Mermaid 常用）

graph（流程图 / 流向图）

sequenceDiagram（时序图）

classDiagram（类图）

stateDiagram（状态图）

gantt（甘特图）

erDiagram（实体关系图）

journey（用户旅程图）

### 使用方法

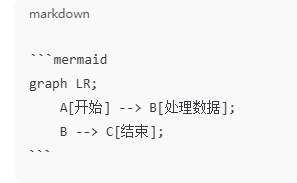
1、创建 Markdown 文件并写 Mermaid 代码

Mermaid 语法块的写法要注意：

代码块声明必须是```mermaid（mermaid 前没有空格）

结尾用 ````（三个反引号）关闭

**示例**（test.md）：



2、打开 Markdown 预览

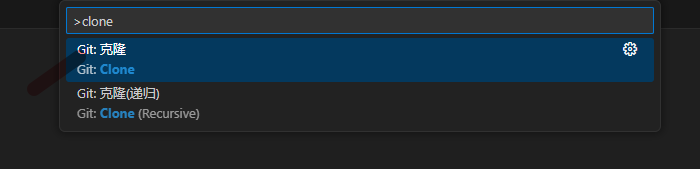
插件安装完成后，有两种方法预览 Mermaid 图：

**方法 1（快捷键）**  
打开 test.md → 按 Ctrl+Shift+V 进入预览模式

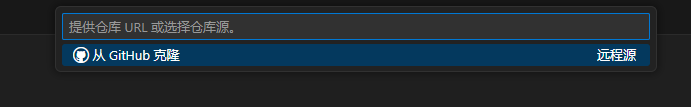
**方法 2（右键）**  
右键 test.md → 选择 **Open Preview**

# 怎么通过vscode拉取git项目

1、打开命令面板，命令面板中键入“Git: Clone”并选择它



2、在弹出的输入框中，输入Git仓库的URL地址。

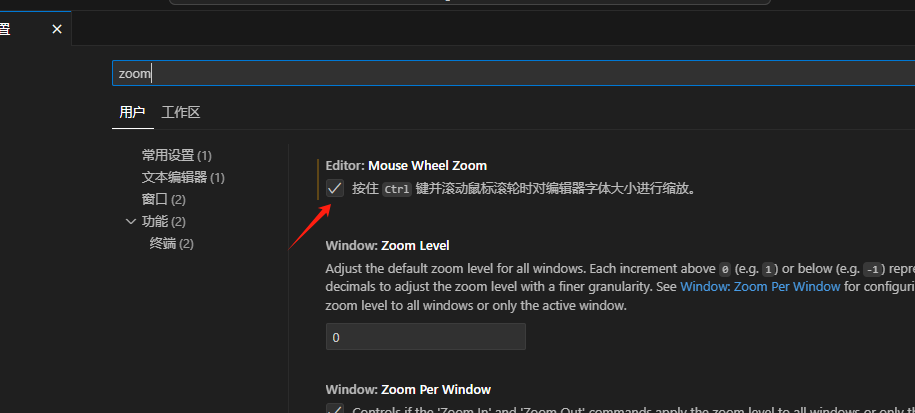


3、选择要将该项目克隆到本地的目录。单击“Enter”键开始克隆项目。

4、一旦克隆完成，VS Code将会打开一个新窗口，显示项目的内容。你可以在VS Code中使用所有的Git功能来管理和修改项目。

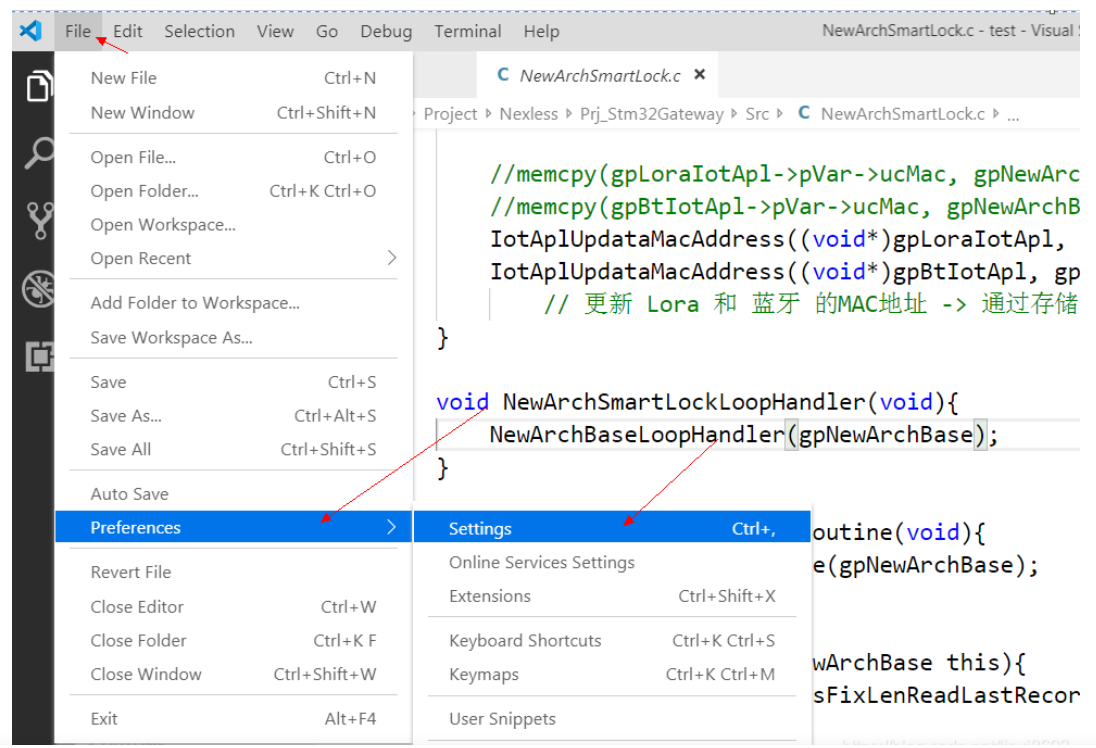
# 开启鼠标滚轮缩放字体大小设置

打开首选项，进入设置，搜索zoom，给滚轮缩放字体选项打上钩

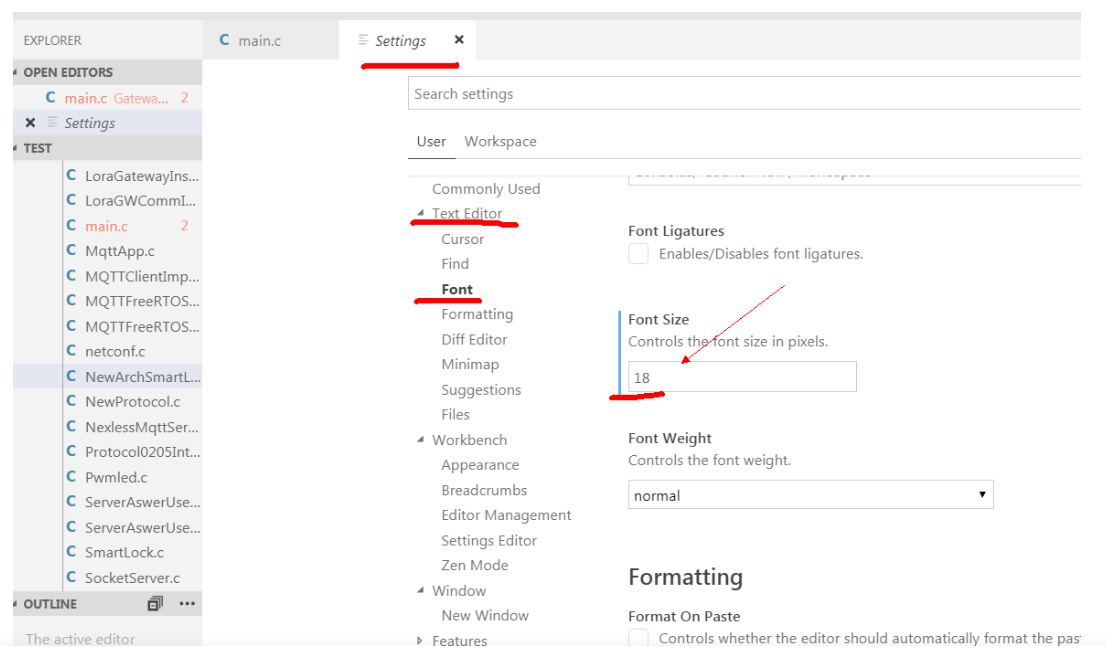


# 设置字体大小（代码）

菜单的选择 -> 参考下图

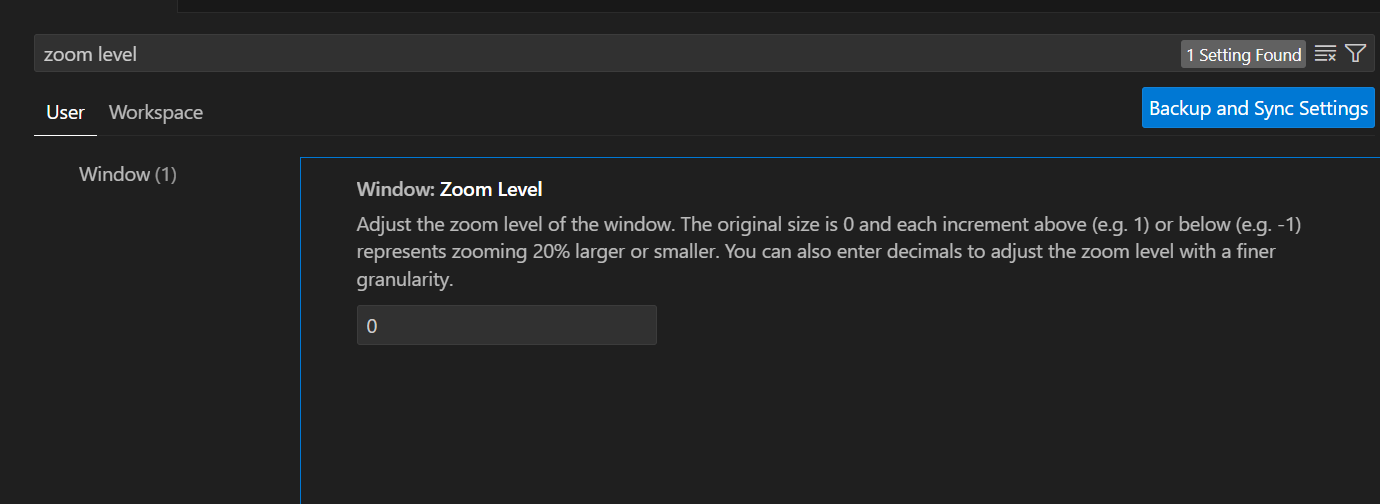


对话框的设置 -> 参考下图



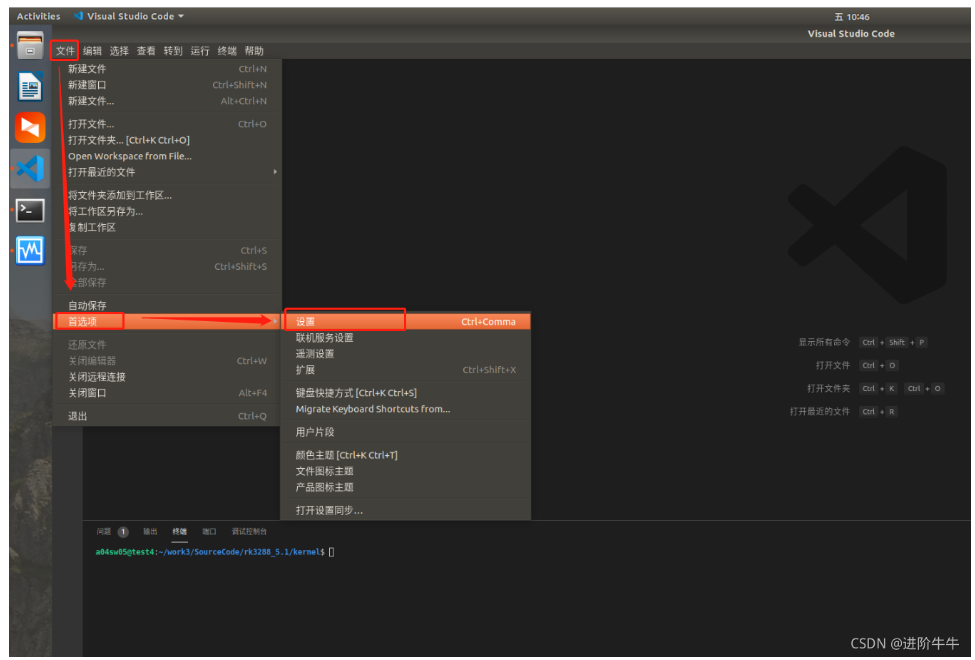
# 设置非代码字体大小

按ctrl+,然后搜索Zoom Level，然后出现一个输入框，把输入框的 -1，改成1

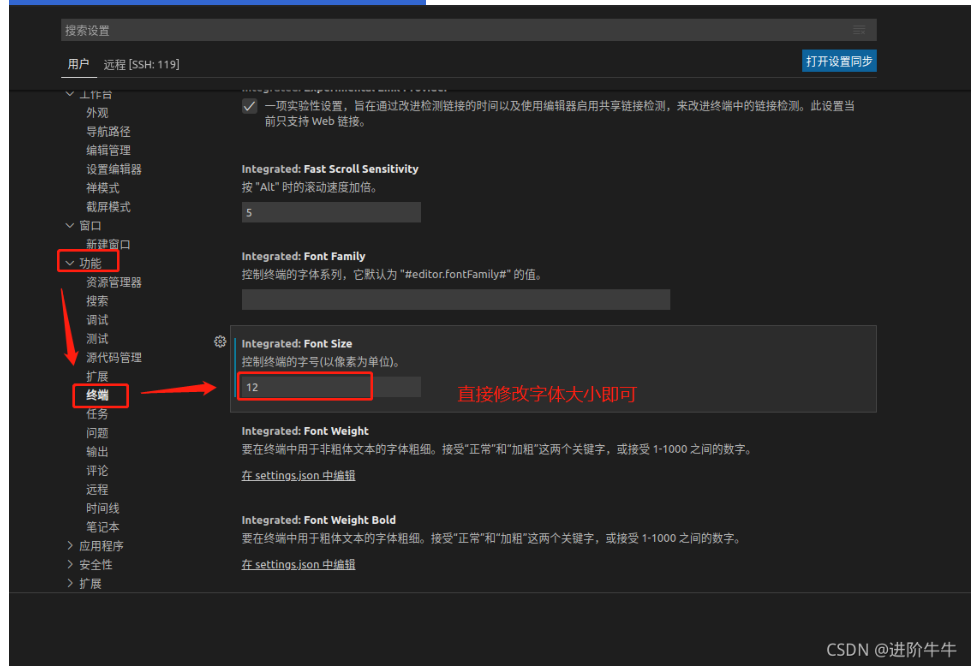


# 修改终端控制台字体大小

1、点击“文件”-> “首选项” -> “设置” ，操作如下图所示：

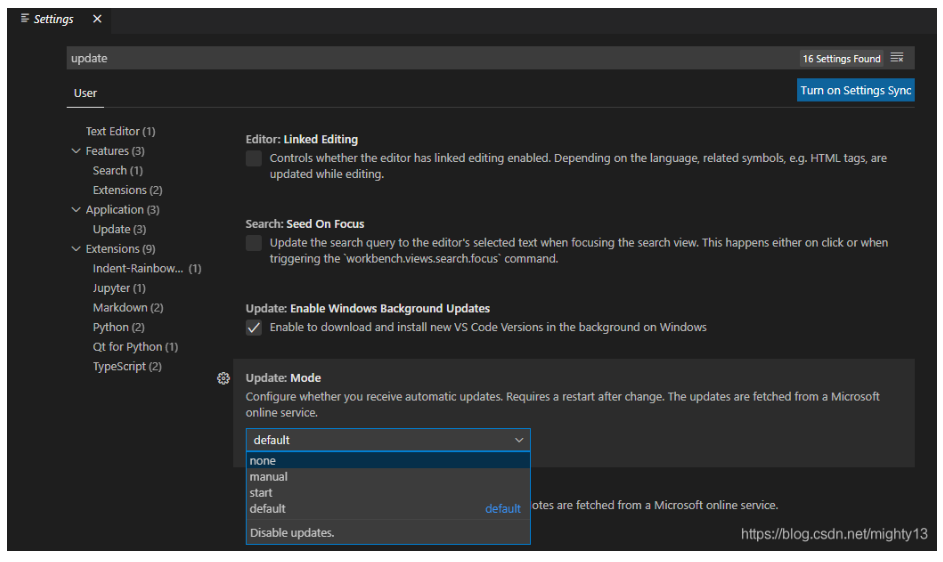


2、点击"设置“后，会弹出设置界面，在启动选择”功能“ -> ”终端“，然后下拉找到 ”Font Size“选项，设置合适的字体大小即可；如果觉得下拉滚轮找麻烦的话，可以直接在 settings.json 文本中编辑。



# 禁止更新

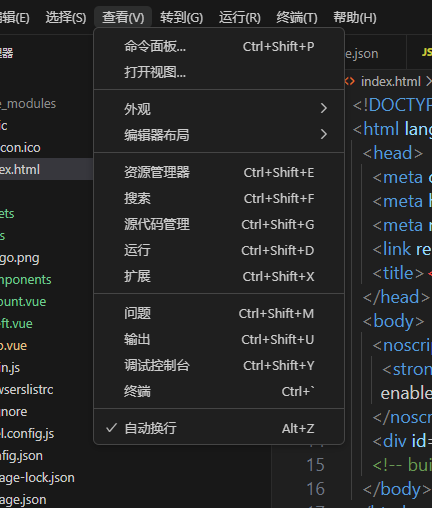
打开菜单File中Preferences子菜单中选择Settings项，搜索update mode，将其设置为none。



# 自动换行

## 第一种方法（使用这个）

选择查看-自动换行

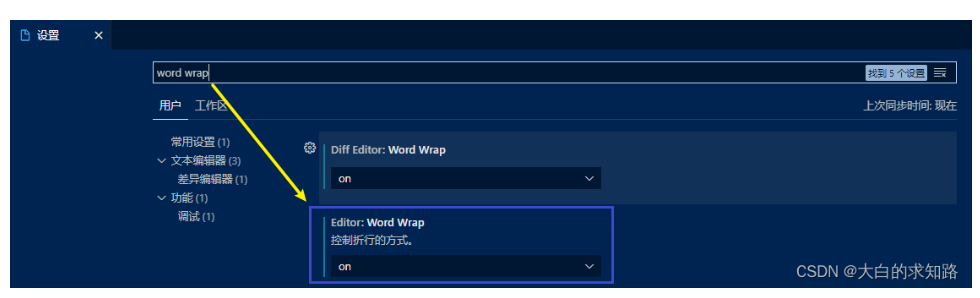


## 第二种方法

按下“Ctrl+Shift+P”，输入“Open user settings”



搜索栏里输入“word wrap”，更改Editor: Word Wrap为“on”，即可



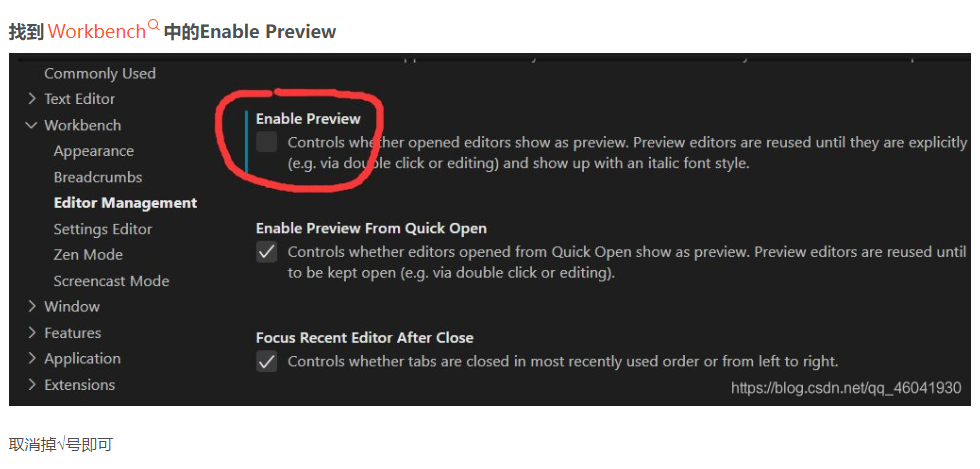
# 同时打开多个文件

## 第一种方法（使用这个）

打开文件后双击该文件名，也可以实现文件保持打开。

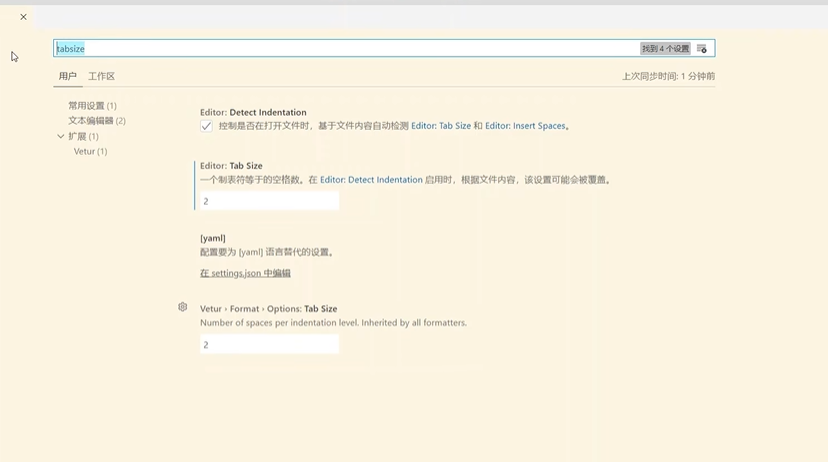
## 第二种方法

进入设置后，打开下面的页面：



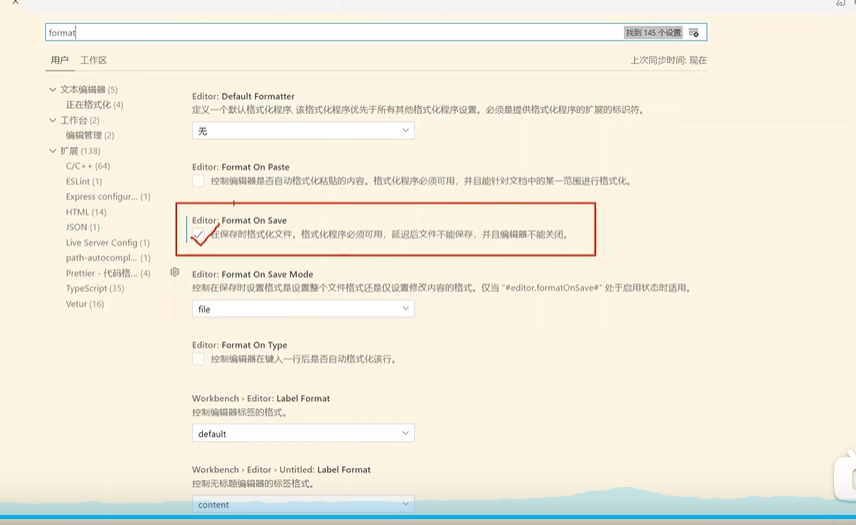
# 设置Tab为2个空格

输入tab size

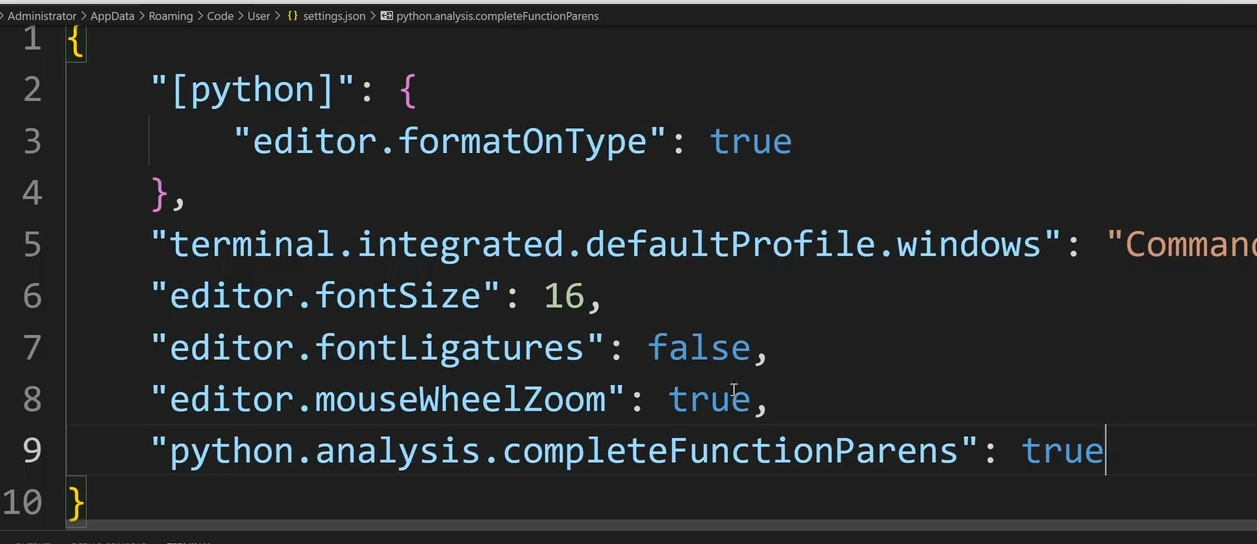


# 保存时格式化

输入format



# 安装python环境



# 设置前端包管理器

设置npm为默认包管理器

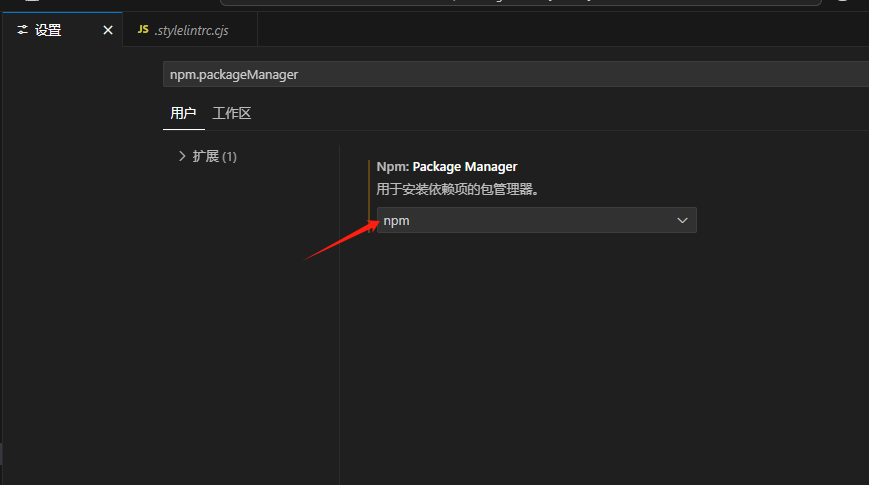
1、打开VS Code设置

2、在设置搜索框中输入 npm.packageManager

3、修改设置

（1）找到 "Npm: Package Manager" 选项

（2）将值从 "auto" 改为 "npm"



# 怎么把vscode中的git提示信息关闭

