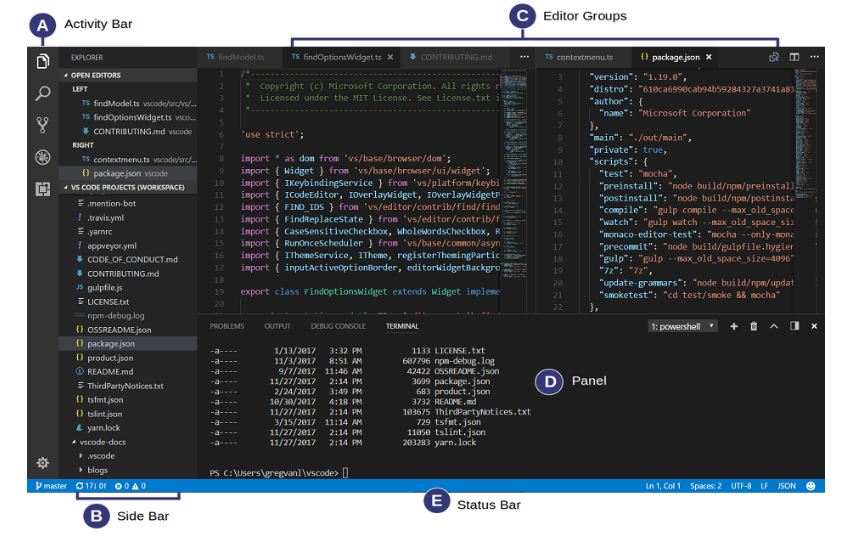
说明

* ：表示配置设置

# 基础布局



**编辑器：**用于编辑代码

**侧边栏**：显示文件夹和文件信息

**状态栏**：显示正在编辑的项目和文件的信息

**活动栏**：用于切换视图和显示专用上下文信息，如在Git中改变的数目。

**面板：**显示输出、调试信息、错误或警告信息，还有终端。

* 每一次VSC将记录上一次关闭时的状态，并在下一次打开时显示。

## 自定义外观

菜单栏“查看”=》“外观”

# 编辑器Editor

## 编辑器布局

菜单栏“查看”=》“编辑器布局”

## 并排编辑

可以打开多个编辑器，而且可以水平或垂直并排排列多个编辑器。

如果当前已经打开了一个编辑器，可以采取以下操作再并排打开一个编辑器：

* Alt + 鼠标点击侧边栏中的文件
* Ctrl+“\”，将原有的编辑器拆分成两个
* 在侧边栏中选定文件，然后Ctrl+Enter
* 点击拆分按钮
* 鼠标拖动文件到编辑器
* Ctrl+P,快速打开文件
* 默认情况下，编辑器会优先激活右边的编辑器。可以通过设置workbench.editor.openSideBySideDirection。

## 编辑器标签Tabs

VSC会在每个编辑器顶部显示一个标签Tab，用于选取和拖动编辑器。

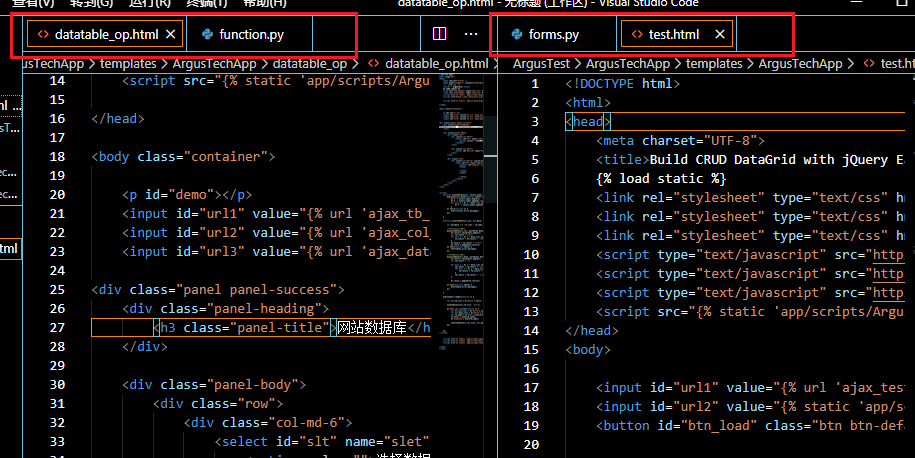
* workbench.editor.showTabs：设置是否显示编辑器标签

默认情况下，新打开的编辑器会出现在当前编辑器的右侧。

* workbench.editor.openPositioning：设置新编辑器显示位置

## 编辑器组

当拆分编辑器时，实际上是创建了新的编辑器组。在编辑器组内可以包含有多个编辑器。



## 选择编辑器的快捷键

* Ctrl+PageDown 前往右边的编辑器
* Ctrl+PageUp 前往左边的编辑器
* Ctrl+Tab 切换编辑器
* Ctrl+1 前往最左的编辑器组
* Ctrl+2 前往中间的编辑器组
* Ctrl+3 前往最右的编辑器组
* Ctrl+F4 关闭当前编辑器
* Ctrl+K W 关闭当前编辑器组内的所有编辑器
* Ctrl+K Ctrl+W 关闭所有编辑器

## 导航痕迹（Breadcrumb）

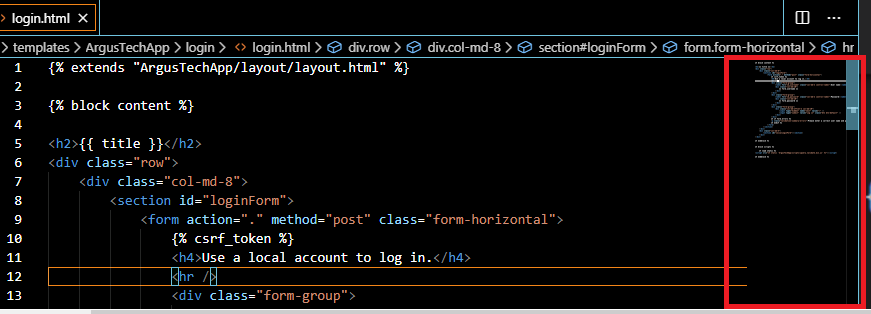
导航痕迹显示文件夹、文件和符号的相对关系。



显示导航痕迹：查看=》显示导航痕迹

## 迷你地图

迷你地图是对代码的概览，在编辑器右侧显示。可以在地图中点击或拖动对代码进行快速定位。

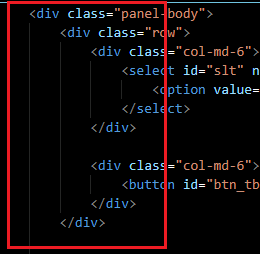


* 默认在编辑器右侧显示。

"editor.minimap.side": "left"：在左侧显示

"editor.minimap.enabled": false：不显示迷你地图

## 缩进提示



VSC会提示代码缩进情况。

* "editor.renderIndentGuides": false：取消代码缩进提示

## 禅模式

只保留编辑器窗口，并进入全屏。进入全屏时，可以数据拖动边条调整宽度，或按下“Alt”键拖动单边边条，只调整一边的宽度。

启用方式：

* 命令面板
* 快捷键Ctrl+K Z

退出方式：

* 双击ESC
* zenMode.fullScreen：禅模式下是否全屏

zenMode.hideStatusBar：是否隐藏状态栏

zenMode.hideTabs：是否隐藏编辑器标签

zenMode.restore：是否恢复

zenMode.centerLayout：编辑器是否居中对齐

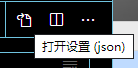
## 设置编辑器配置

可以设定一个通用的编辑器配置或对不同的项目/文件夹进行个别设定。所有设定都存放在settings.json文件中。

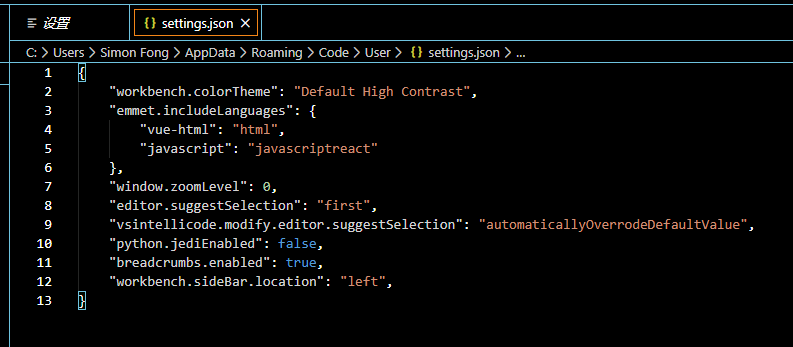
进入设置UI界面：

* 用户设置：菜单栏“文件”=》“首选项”=》“设置”（或快捷键Ctrl+“，”），然后“用户”。
* 工作区设置：菜单栏“文件”=》“首选项”=》“设置”（或快捷键Ctrl+“，”），然后“工作区”。

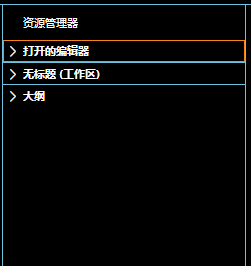
进入json代码文件：

如要进入json文件手动设置，点击右上角“打开设置json” 

进入json文件：



# 资源管理器Explorer

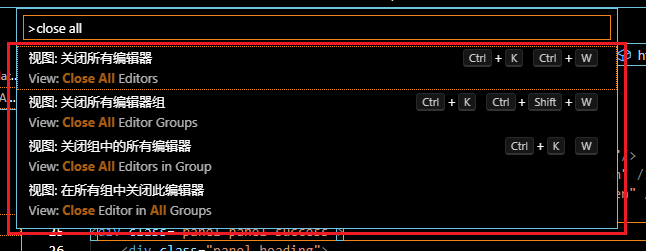


在资源管理器中一共有三个窗口：

* 打开的编辑器：显示当前在编辑器中开打的所有文件
* 工作区：显示当前工作区中的所有文件和文件夹
* 大纲：显示当前编辑器中文件的代码符号结构

## 打开的编辑器Open Editors

在资源管理器的顶部。以列表形式显示当前被打开的文件。点击其中的文件，会激活该文件。当完成代码编辑后，可以在该窗口中注意关闭文件；也可以在视图中选择“关闭所有编辑器”或“关闭组中的所有编辑器”命令。



## 工作区Workspace

用于管理全部文件和文件夹。

如果目标文件是命令行工具（a command-line tool），可以直接选中文件，然后右键菜单中选择“在终端中打开”

* 默认情况下，VSC会排除一些文件夹在资源管理器中显示，如.git。

files.exclude设置在资源管理器中隐藏的文件和文件夹。

这样的设置可以有效地派生的资源文件，如在Unity中的\\*.meta文件，在TypeScript的\\*.js文件，

### 预览模式

当在资源管理器的工作区中单击文件时，只会新打开一个预览编辑器，并在新的编辑器中显示代码。同时在“打开的编辑器”和标签中显示的文件名是斜体的。

* 此时编辑器只是在预览模式，不会直接进入文件。
* 此时如选取不同的文件，都只会预览编辑器中显示内容，不会再打开新的编辑器。

当双击文件时，预览编辑器变成可编辑的编辑器，文件名变成是正常显示。

* workbench.editor.enablePreview是否启用一个用于预览的编辑器

workbench.editor.enablePreviewFromQuickOpen当快速打开Quick Open时，是否启用预览编辑器

### 选择多个文件

比较多个文件：在工作区选择多个文件，右键菜单“将已选择项比较”

* 默认情况下，Ctril+鼠标左键进行多项选择。

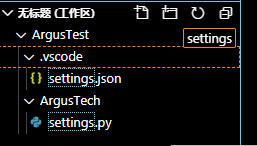
workbench.list.multiSelectModifier设置在选择多个文件时的控制键。

### 过滤显示文件

用鼠标激活工作区，并在其顶部输入，这是将激活过滤器。一旦在当前层级文件夹中存在与过滤器输入文件或文件夹名称相符的，将高亮显示。

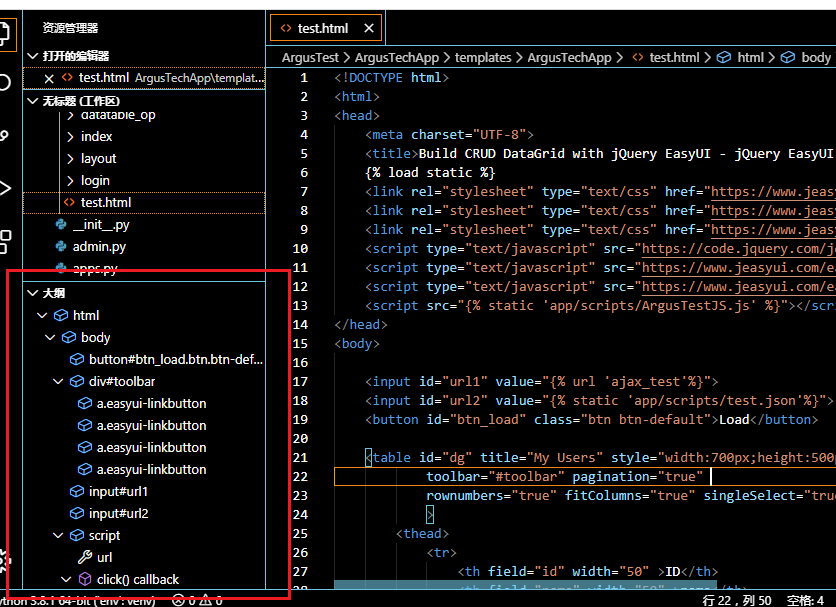


此时在过滤器中点击匹配的文件，会只显示匹配的文件。

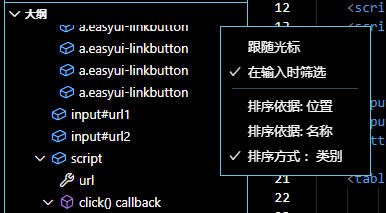


## 大纲Outline

大纲窗口位于侧边栏底部，用于显示当前文件代码中的符号结构。



在大纲中可以以不同排序方式查看代码符号结构：



还可以过滤查找代码符号：查找包含link的符号



错误和警告都会显示在大纲窗口中，从而快速地定位问题所在。

大纲窗口的显示内容取决于在VSC中安装的不同类型文件的扩展包有关。

* outline.icons是否显示符号元素的图标

outline.problems.enabled是否显示错误和警告

outline.problems.badges是否使用徽章显示错误和警告

outline.problems.colors是否使用不同的颜色标记错误和警告

# 命令面板Command Palette

快捷键：Ctrl+Shift+P

通过命令面板可以实现VSC的全部功能。

## 常用命令快捷键：

Ctrl+Shift+P：打开命令面板

Ctrl+P：根据输入搜索文件或视图

Ctrl+Tab在打开的编辑器列表中切换

Ctrl+Shift+O：根据输入搜索文件中的变量、类、方法

Ctrl+G:快速定位到一个指定代码行

## 显示可用的命令列表

在命令面板中输入“?”，可以返回可用的命令列表。

注意：输入的是问号，不是“>?”。



## 视图Views

视图命令可以进入以下窗口：

* 资源管理器
* 搜索
* 源代码管理
* 运行
* 扩展包
* 自定义视图（如果安装相关扩展包）



## 活动栏

可以实现视图的快速切换的工具

# 主题Theme

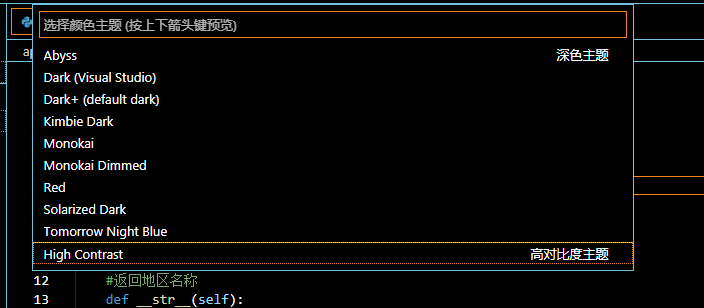
在VSC中，主题包括VSC显示的主题、文件图标的主题、代码符号显示的主题。这些主题都可以在VSC内置主题列表中选择，也可以安装扩展包，也可以自定义，也可以创建自己的主题。本章节主要集中在内置主题，其他均略去。

## VSC颜色主题

### 选择内置主题

打开颜色主题挑选框：文件=》首选项=》颜色主题

快捷键：Ctrl+K Ctrl+T



* **workbench.colorTheme**

### 禁用内置主题

打开扩展包管理器，选择内置扩展包，在主题列表中选中并禁用相应扩展包。

### 安装主题扩展包

略

### 自定义

略

* 在workbench.colorCustomizations中设定

### 创建新的主题

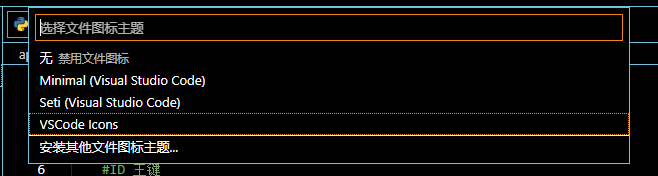
略

## 图标主题

图标主要是在资源管理器和编辑器标签中显示，以直观地区分不同类型的文件、文件夹。在VSC中可以为这些图标选择不同的主题。

### 选择文件图标主题

打开对话框：文件=》首选项=》文件图标主题



* 在workbench.iconTheme设置

### 安装扩展包

略

### 创建新的文件图标主题

略

## 代码符号的主题

代码符号的高亮显示因应不同的语言而有所不同。

* 可以在**editor.tokenColorCustomizations** 进行设置

# 设置Settings

在vsc中，设置包括：

* 用户设置User Settings：适用于所有在VSC中打开的实例
* 工作区设置Workspace Settings：适用于特定的工作区

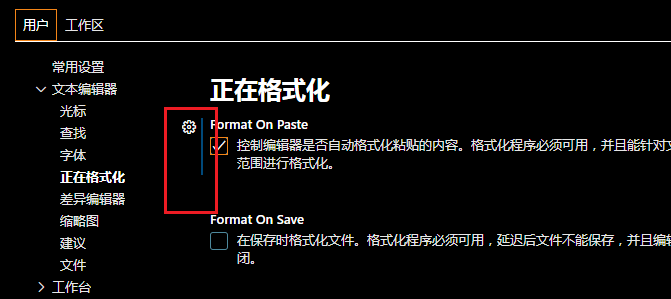
工作区设置会覆盖用户设置，并只适用于特定项目。当该项目在多个开发者中共享时，该工作区设置也会被其他开发者共享。

注：所谓的VSC工作区Workspace，仅仅指项目所在的根目录文件夹。对工作区的设置、调试器配置和任务设定都会存储在根目录中的.vscode文件夹中。

## 打开设置界面

打开设置：文件=》首选项=》设置

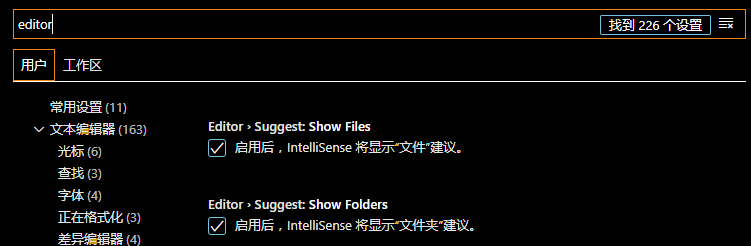
当设置被修改时，会在被修改的项目上用蓝色竖线标记提示。



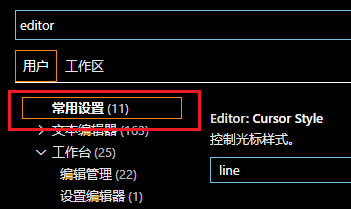
所有设置都会被保存在一个json文件中。

* 搜索特定设置

可以在设置对话框中的搜索栏输入信息来搜索相关的设置项。这也适用于扩展包的设置。



在“常用设置”中会显示经常被使用到的设置项。



## 设置文件的存放路径（settings.json）

默认情况下，vsc的会使用UI界面显示设置。可以通过切换按钮切换到json文件去设置。

* workbench.settings.editor打开vsc设置的方式，默认是在UI界面中打开
* 在Windows下，用户设置文件路径：

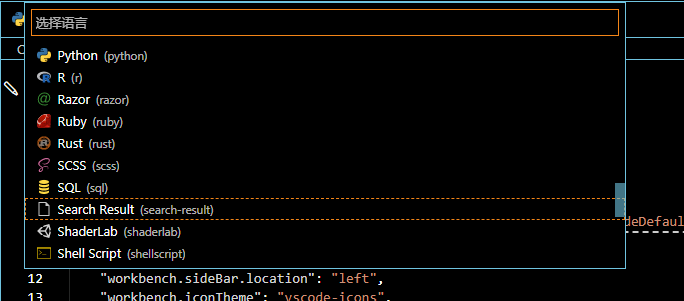
%APPDATA%\Code\User\settings.json

* 在Windows下，工作区设置文件路径：

工作区根目录里的.vscode文件夹中

## 通过语言自定设置对话框

在命令面板输入：Preferences: Configure Language Specific Settings，然后选择需要的语言。



* workbench.action.configureLanguageBasedSettings

例子：略

## 安全性

处于安全的考虑，某一些设置只能在用户设置中设置，不能再工作区设置中设置。

以下为不能在工作区设置：

* git.path
* terminal.integrated.shell.linux
* terminal.integrated.shellArgs.linux
* terminal.integrated.shell.osx
* terminal.integrated.shellArgs.osx
* terminal.integrated.shell.windows
* terminal.integrated.shellArgs.windows
* terminal.external.windowsExec
* terminal.external.osxExec
* terminal.external.linuxExec

## VSC默认设置

略。相见VSC文档

## 语言

VSC默认语言为英语，其他语言通过安装扩展包获取。

修改VSC语言

命令面板：**Configure Display Language**

使用参数文件argv.json：略

## 常见问题

* 无法写入设置

当自定义设置但显示"Unable to write settings."时，多数是因为自定义的语句出现语法错误。

* 恢复初始设置

删除settings.json文件中的两个大括号之间的内容，保存文件。然后VSC会使用默认设置。

* 何时使用工作区设置

当自定义的设置只想适用于当前工作区，但不想适用于其他项目时。如对特定语言的智能提示规则。

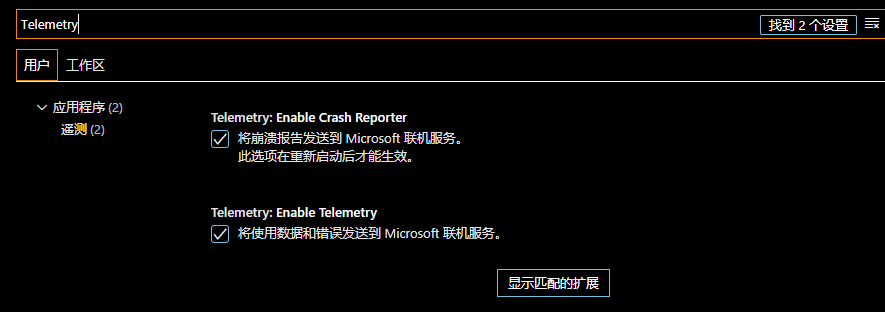
## 向VSC发送数据

禁止发送改善数据（Telemetry）

* "telemetry.enableTelemetry":false 禁止发送数据

"telemetry.enableCrashReporter": false 禁止发送崩溃报告

文件=》首选项=》设置，搜索Telemetry



安装扩展包：略

在输出面板查看发送报告：略

# 按键绑定

## 键盘快捷键编辑器

用于设置快捷键。

打开编辑器：文件=》首选项=》键盘快捷方式

## 键盘扩展包

安装扩展包。略

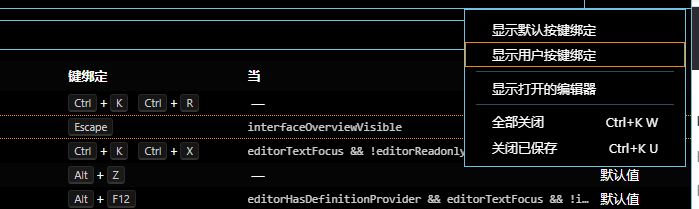
## 查看默认快捷键列表

帮助=》键盘快捷方式参考。会打开一个包含VSC默认快捷键列表的PFD文件

## 探测冲突的快捷键设定

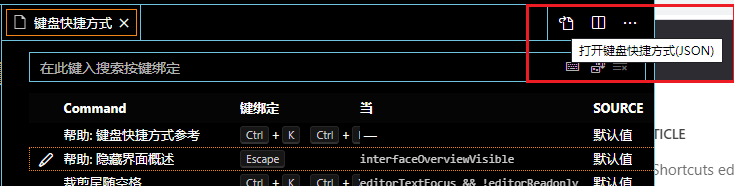
当安装多个快捷键扩展包时，会产生冲突。解决的方法是打开键盘快捷键编辑器，选定一个冲突的快捷键设定，右键菜单“显示相同的按键设定”。

查看用户自定义的快捷键设定：



## 自定义快捷键文件（keybingdings.json）

所有快捷键的设定都可以在keybingdings.json文件中进行自定义。



键盘规则，用于在keybingdings.json文件中设定快捷键。

例子：

// Keybindings that are active when the focus is in the editor

{ "key": "home", "command": "cursorHome", "when": "editorTextFocus" },

{ "key": "shift+home", "command": "cursorHomeSelect", "when": "editorTextFocus" },

规则包含：

key：指定按键

command：包含执行的命令

when：可选，指明规则起作用的条件

首先判断key和when是否符合规则，则执行command命令。

VSC运行时随时调用该json文件，即在VSC运行时随时修改json文件随时使用自定义的快捷键设定。

## 接受的key值

key值由修饰符和按键组成。

修饰符（Windows）包含：Ctrl+，Shift+，Alt+，Win+

按键包括：

* f1-f19, a-z, 0-9
* `, -, =, [, ], \, ;, ', ,, ., /
* left, up, right, down, pageup, pagedown, end, home
* tab, enter, escape, space, backspace, delete
* pausebreak, capslock, insert
* numpad0-numpad9, numpad\_multiply, numpad\_add, numpad\_separator
* numpad\_subtract, numpad\_decimal, numpad\_divide

## 命令Command的参数

可以向command命令传递参数。

如，设定Enter键：

{

"key": "enter",

"command": "type",

"args": { "text": "Hello World" },

"when": "editorTextFocus"

}

当按下回车键时，会执行输入操作，输入的内容则由参数args定义。

## 移除特定的快捷键设置规则

略

## 键盘布局

不同的键盘布局会产生冲突。略

## When语句

When语句定义了快捷键生效的情况。如果省略该语句，表示任何时候都有效。

在when语句中使用的条件运算符。略

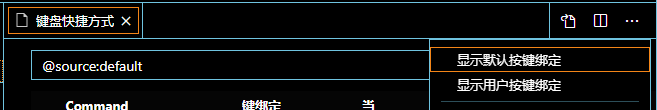
when适用的情景表。略

## 自定义快捷键用于函数重构

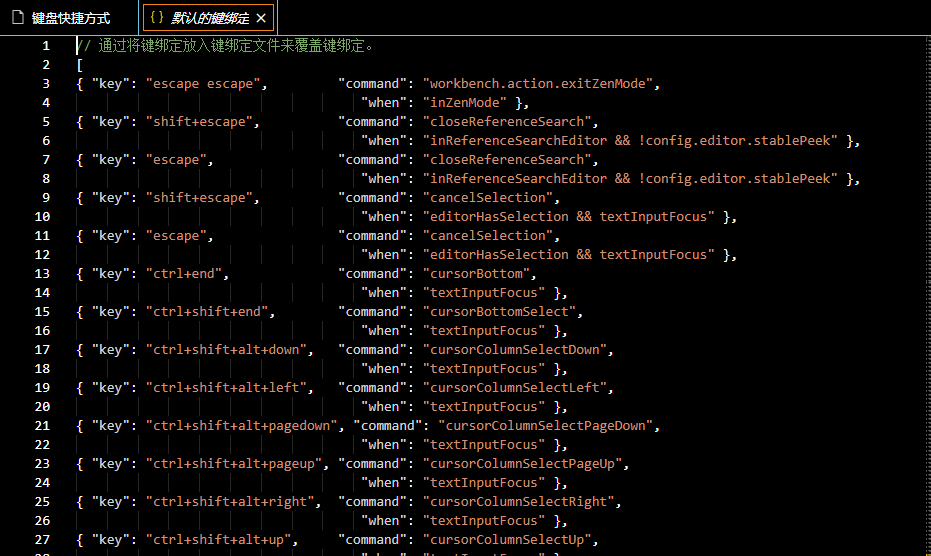
略

## VSC默认快捷键

在UI界面中查看



在json文件中查看：命令面板：Preferences: Open Default Keyboard Shortcuts (JSON)



详细列表：略。或见以下快捷键参考表

# 调试

launch.json文件

{  
 "name": "调试 Node.js 程序",  
 "type": "node",  
 "request": "launch",  
 "program": "${workspaceFolder}/src/index.js"  
}

* type，必填项，调试类型，当前为 node，如果是 PHP 调试，则在安装 PHP 调试插件后写 php；
* request，必填项，有两种类型，分别是 launch 和 attach，前者的意思就是 VSCode 会打开这个程序然后进入调试，后者的意思是你已经打开了程序，然后接通 Node.js 的内部调试协议进行调试，如果你看过上面的“Node.js 的调试原理”一文，应该可以大致理解；
* program，程序的启动入口；

# VSC配置

在配置中常用到的标识符：

视图：

* workbench.view.explorer - File Explorer
* workbench.view.search - Search
* workbench.view.scm - Source Control
* workbench.view.debug - Run
* workbench.view.extensions - Extensions

面板

* workbench.panel.markers - Problems
* workbench.panel.output - Output
* workbench.panel.repl - Debug Console
* workbench.panel.terminal - Integrated Terminal
* workbench.panel.comments - Comments
* workbench.view.search - Search when search.location is set to panel

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名 | 作用 | 备注 | 场景 |
|  |  |  |  |
|  |  |  |  |
| workbench.settings.editor | 打开VS Code设置的方式， | 默认是在UI界面中打开 | Setting |
| workbench.action.configureLanguageBasedSettings | 指定特定语言进行VSC设置 |  | Setting |
| workbench.colorTheme | 设置VS Code主题 |  | Theme |
| workbench.colorCustomizations | 自定义VS Code主题 |  | Theme |
| editor.tokenColorCustomizations | 自定义编辑器语句显示的主题 |  | Theme |
| workbench.iconTheme | 设置文件图标主题 |  | Theme |
| files.exclude | 设置在资源管理器中隐藏的文件和文件夹 | 例子："\*\*/\*.cs.meta": true  隐藏Unity的\\*.cs.meta文件  "\*\*/\*.js": {"when": "$(basename).ts"}  隐藏TypeScript的\\*.js文件 | Explorer |
| workbench.list.multiSelectModifier | 设置在资源管理器选择多个文件时的控制键 | 默认值："Ctrl"。当修改为"alt"时，按下alt+鼠标右键可以在侧边栏控制器中进行多项选取。 | Explorer |
|  |  |  |  |
| outline.icons | 在大纲窗口中是否显示符号元素的图标 | 默认值："enable",显示图标 | Outline |
| outline.problems.enabled | 在大纲窗口中是否显示错误和警告 | 默认值："enable",显示错误和警告 | Outline |
| outline.problems.badges | 是否使用徽章显示错误和警告 | 默认值："enable",使用徽章显示错误和警告 | Outline |
| outline.problems.colors | 是否使用不同的颜色标记错误和警告 | 默认值："enable",使用不同的颜色标记错误和警告 | Outline |
|  | | | |
| workbench.editor.openSideBySideDirection | 指定并排显示编辑器时的方向 | 默认值：在右边打开并激活编辑器 | Editor |
| editor.minimap.side | 迷你地图显示位置：左/右 | 默认值：right，在编辑器右侧显示 | Editor |
| editor.minimap.enabled | 显示/隐藏迷你地图 | 默认值：true，显示迷你地图 | Editor |
| editor.renderIndentGuides | 显示/隐藏代码缩进提示 | 默认值：true，显示代码缩进提示 | Editor |
| workbench.editor.showTabs | 设置是否显示编辑器标签 | 默认值：true，显示标签 | Editor |
| workbench.editor.openPositioning | 设置新编辑器显示位置 | 默认值："right"，在右侧显示新编辑器 | Editor |
|  |  |  |  |
|  | | | |
| zenMode.fullScreen | 禅模式下是否全屏 | 默认值："enable" | Zen Mode |
| zenMode.hideStatusBar | 禅模式下是否隐藏状态栏 |  | Zen Mode |
| zenMode.hideTabs | 禅模式下是否隐藏编辑器标签 |  | Zen Mode |
| zenMode.restore | 禅模式下是否恢复 |  | Zen Mode |
| zenMode.centerLayout | 禅模式下编辑器是否居中对齐 |  | Zen Mode |
|  | | | |
| workbench.editor.enablePreview | 是否启用一个用于预览的编辑器 | 默认值："enable"，启用 | Preview |
| workbench.editor.enablePreviewFromQuickOpen | 当快速打开Quick Open时，是否启用预览编辑器 |  | Preview |
|  |  |  |  |
|  |  |  |  |

# 常用快捷键：

## 代码编辑

|  |  |  |  |
| --- | --- | --- | --- |
| 组合键 | | 作用 | 备注 |
| Ctrl+X | **剪切一行（空选时）** | | **行剪切** |
| Shift+Alt+F | **代码自动对齐** | | **对齐** |
| Ctrl+C | **复制一行（空选时）** | | **行复制** |
| Ctrl+Shift+K | **删除代码行** | | **行删除** |
| Ctrl+Enter | **在当前行下插入一行** | | **行插入** |
| Ctrl+Shift+Enter | **在当前行上插入一行** | | **行插入** |
| Alt+Down | **将当前行移动到下一行** | | **行移动** |
| Alt+Up | **将当前行移动到上一行** | | **行移动** |
| Shift+Alt+Down | **将当前行复制并粘贴到下一行** | | **行复制** |
| Shift+Alt+Up | **将当前行复制并粘贴到上一行** | | **行复制** |
| Ctrl+Z | **撤销上一个操作** | | **撤销** |
| Ctrl+Y | **向前恢复操作** | | **撤销** |
| Ctrl+D | **添加相同的代码到选中范围内** | | **查找下一个** |
| Ctrl+K Ctrl+D | **移动到下一个相同的代码** | | **查找下一个** |
| Ctrl+U | **撤销上一次光标操作（移动或选中）** | | **光标移动** |
| Shift+Alt+I | **选中多个时，将光标置于每一选中内容的末尾** | | **光标移动** |
| Ctrl+Shift+L | **查找所有配对的内容并选中** | | **查找所有** |
| Ctrl+F2 | **查找相同的内容并选中** | | **查找所有** |
| Ctrl+L | **选中当前行** | | **行选中** |
| Ctrl+Shift+\ | **光标调到匹配的括号处** | | **光标移动** |
| Ctrl+] | **缩进行** | | **行对齐** |
| Ctrl+[ | **突进行** | | **行对齐** |
| Home | **移动到行首** | | **光标移动** |
| End | **移动到行尾** | | **光标移动** |
| Ctrl+End | **移动到文档尾** | | **光标移动** |
| Ctrl+Home | **移动到文档首** | | **光标移动** |
| Ctrl+Down | **相当于鼠标滚轮向上滚动** | | **光标移动** |
| Ctrl+Up | **相当于鼠标滚轮向下滚动** | | **光标移动** |
| Alt+PageDown | **向下翻一页** | | **光标移动** |
| Alt+PageUp | **向上翻一页** | | **光标移动** |
| Ctrl+Shift+[ | **折叠当前区域** | | **代码块** |
| Ctrl+Shift+] | **展开当前区域** | | **代码块** |
| Ctrl+K Ctrl+[ | **折叠区域，包括所有子区域** | | **代码块** |
| Ctrl+K Ctrl+] | **展开区域，包括所有子区域** | | **代码块** |
| Ctrl+K Ctrl+0 | **折叠文档内所有区域** | | **代码块** |
| Ctrl+K Ctrl+J | **展开文档内所有区域** | | **代码块** |
| Ctrl+K Ctrl+C | **注释当前行** | | **注释** |
| Ctrl+K Ctrl+U | **取消注释当前行** | | **注释** |
| Ctrl+/ | **注释/取消注释当前行** | | **注释** |
| Shift+Alt+A | **添加注释块** | | **注释** |
| Ctrl+F | **查找** | | **查找** |
| Ctrl+H | **查找并替换** | | **查找** |
| Enter | **（查找对话框中）查找下一个** | | **查找** |
| Shift+Enter | **（查找对话框中）查找上一个** | | **查找** |
| Alt+Enter | **（查找对话框中）选中所有匹配的内容** | | **查找** |
| Alt+C | **（查找对话框中）切换大小写** | | **查找** |
| Alt+R | **（查找对话框中）切换正则** | | **查找** |
| Alt+W | **（查找对话框中）全部匹配** | | **查找** |
| Alt+Z | **切换是否自动换行** | |  |

## 语言编辑

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| F12 | **转到定义** | **定义** |
| Alt+F12 | **快速查看定义** | **定义** |
| Shift+F12 | **显示引用** | **定义** |
| Ctrl+K Ctrl+I | **显示悬停信息** | **提示** |
| F2 | **重命名** | **重命名** |
| Shift+Alt+Right | **智能选择扩选** | **框选** |
| Shift+Alt+Left | **智能选择缩选** | **框选** |

## 导航

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| Ctrl+T | **显示项目所有符号（类、函数、变量等）** |  |
| Ctrl+G | **定位到行** |  |
| Ctrl+P | **快速打开文件** |  |
| Ctrl+Shift+O | **定位到符号** |  |
| Ctrl+Shift+M | **显示问题** |  |
| F8 | **定位到下一个错误或警告** |  |
| Shift+F8 | **定位到上一个错误或警告** |  |
| Ctrl+Shift+P or F1 | **打开命令面板** |  |
| Ctrl+Tab | **切换编辑器** |  |

## 窗口管理

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| Ctrl+Shift+N | **新开一个VSC窗口** |  |
| Ctrl+W | **关闭窗口** |  |
| Ctrl+F4 | **关闭当前编辑器** |  |
| Ctrl+K F | **关闭文件夹** |  |
| Ctrl+\ | **拆分编辑器** |  |
| Ctrl+1 | **激活第一个编辑器组** |  |
| Ctrl+2 | **激活第二个编辑器组** |  |
| Ctrl+3 | **激活第三个编辑器组** |  |
| Ctrl+K Left | **激活左侧编辑器** |  |
| Ctrl+K Right | **激活右侧编辑器** |  |
| Ctrl+K W | **关闭当前编辑器组内的所有编辑器** |  |
| Ctrl+K Ctrl+W | **关闭所有编辑器** |  |

## 文件管理

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| Ctrl+N | **新建一个文件** |  |
| Ctrl+O | **打开文件** |  |
| Ctrl+S | **保存文件** |  |
| Ctrl+K S | **保存所有文件** |  |
| Ctrl+Shift+S | **另存为** |  |
| Ctrl+Shift+T | **重新打开关闭的编辑器** |  |
| Ctrl+K R | **打开当前文档所在的窗口** |  |
| Ctrl+K O | **打开文件夹** |  |

## 显示

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| F11 | **切换到全屏** |  |
| Ctrl+K Z | **进入到禅模式** |  |
| Escape Escape | **离开禅模式** |  |
| Ctrl+= | **放大显示vsc** |  |
| Ctrl+- | **缩小显示vsc** |  |
| Ctrl+B | **切换侧边栏** |  |
| Ctrl+Shift+E | **在资源管理与编辑器之间切换** |  |
| Ctrl+Shift+F | **显示搜索** |  |
| Ctrl+Shift+G | **显示源代码管理** |  |
| Ctrl+Shift+D | **显示运行** |  |
| Ctrl+Shift+X | **显示扩展** |  |
| Ctrl+Shift+U | **显示输出** |  |
| Ctrl+Q | **在视图View列表中切换** |  |
| Ctrl+Shift+C | **打开新的CMD窗口** |  |
| Ctrl+` | **打开新的终端** |  |
| Ctrl+K Ctrl+T | **显示主题选择器** |  |

## 调试

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| F9 | **设置/取消断点** |  |
| F5 | **开始调试** |  |
| F5 | **继续调试** |  |
| Ctrl+F5 | **不调试直接运行程序** |  |
| F6 | **暂停调试** |  |
| F11 | **运行下一行语句** |  |
| Shift+F11 | **跳到父语句** |  |
| F10 | **跳过子语句** |  |
| Shift+F5 | **停止调试** |  |

## 任务

|  |  |  |
| --- | --- | --- |
| 组合键 | 作用 | 备注 |
| Ctrl+Shift+B | **运行任务** |  |

## 创建调试器

* 选择调试器（命令面板），生成launch.json文件。

## 查看函数定义

* 右键“转到定义”；或F12
* 右键“快速查看”=》“查看定义”；或Alt+F12

## 创建一个代码片段（code snippet）

1. 文件=》首选项=》用户代码片段
2. 选择语言
3. 修改.json