

Vscode Workshop

代码片段, 基础设置等

auth

SSTIA

2021 年 10 月 8 日

- ① vscode 基础配置
- ② JSON 文件
- ③ 用户代码片段 User snippets
- ④ 用户代码片段示例
- ⑤ vscode 多光标特性

① vscode 基础配置

② JSON 文件

③ 用户代码片段 User snippets

④ 用户代码片段示例

⑤ vscode 多光标特性

- ① 尽可能使用英文
- ② 使用 windows 系统作为例子,mac 相似, 部分键位不同可参考[macOS 快捷键](#)

restore window : none hotexit: off

- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
- 4 用户代码片段示例
- 5 vscode 多光标特性

JSON

JSON (JavaScript Object Notation, JavaScript 对象表示法) 是一种由道格拉斯·克罗克福特构想和设计、轻量级的数据交换语言, 该语言以易于让人阅读的文字为基础, 用来传输由属性值或者序列性的值组成的数据对象。尽管 JSON 是 JavaScript 的一个子集, 但 JSON 是独立于语言的文本格式, 并且采用了类似于 C 语言家族的一些习惯。JSON 数据格式与语言无关, 脱胎于 JavaScript, 但当前很多编程语言都支持 JSON 格式数据的生成和解析。JSON 的官方 MIME 类型是 application/json, 文件扩展名是.json。 <https://zh.wikipedia.org/wiki/JSON>

JSON 示例

```
1 {  
2   "name": "zhangsan",  
3   "age": 18,  
4   "gender": "male"  
5 }
```

- 字符串
- 对象
- 数组
- 数字

```
1 {  
2   "students": [  
3     { "firstName": "san", "lastName": "zhang" },  
4     { "firstName": "si", "lastName": "li" },  
5     { "firstName": "wu", "lastName": "wang" }  
6   ]  
7 }
```


① vscode 基础配置

② JSON 文件

③ 用户代码片段 User snippets

用户代码片段用途
进入用户代码片段
用户代码框架

④ 用户代码片段示例

⑤ vscode 多光标特性

<https://snippet-generator.app/>

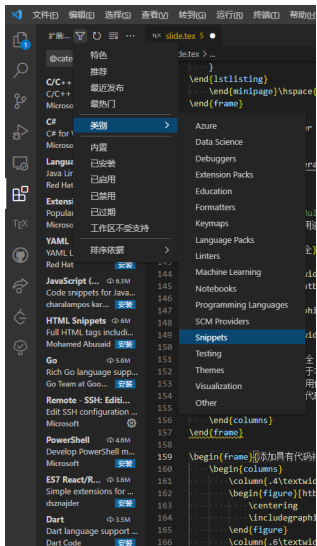
- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
 用户代码片段用途
 进入用户代码片段
 用户代码框架
- 4 用户代码片段示例
- 5 vscode 多光标特性

插件自带补全

```
 snippets.json - VCPlugin.wksp - Visual Studio Code
snippets.json x 1 latex.json
C:\Users\Hyeon> .\vscode> extensions> thomasmq.math-snippets-0.1.1> snippets> 1 snippets.json > ...
1
2 "latex - inline math (dollars)": {
3   "scope": "",
4   "prefix": "li",
5   "body": ["$$$ $@"],
6   "description": "Create inline math formula"
7 },
8
9 "latex - display math (dollars)": {
10  "scope": "",
11  "prefix": "ld",
12  "body": ["$$ $@"],
13  "description": "Create display math formula"
14 },
15
16 "latex - inline math with display style (dollars)": {
17  "scope": "",
18  "prefix": "lid",
19  "body": ["$\\displaystyle $@"],
20  "description": "Create inline math formula with display style"
21 },
22
23 "latex - inline math (brackets)": {
24  "scope": "",
25  "prefix": "li",
26  "body": ["\\(\\$\\$\\$ $@"],
27  "description": "Create inline math formula"
28 },
29
30 "latex - display math (brackets)": {
31  "scope": "",
32  "prefix": "ld",
33  "body": ["\\[\\$\\$\\$ $@"],
34  "description": "Create display math formula"
35 },
```

- 代码补全（定义于 vscode 插件）
- 可查看于本地
”*C : \Users < UserName > \.vscode\extensions\ms – code. – \snippets.*”

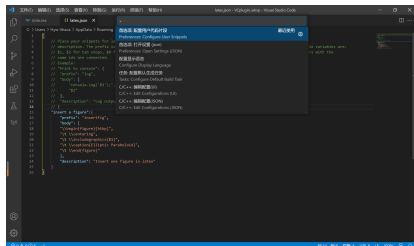
添加具有代码补全功能的插件



- 在插件商店中输入关键字，并使用过滤器搜索出带有代码片段的插件安装。
- 很多常用代码捆绑在一起或某些补全插件中不支持
- 自定义代码补全

- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
 - 用户代码片段用途
 - 进入用户代码片段
 - 用户代码框架
- 4 用户代码片段示例
- 5 vscode 多光标特性

进入用户代码片段



进入用户代码片段方式

- 快捷键 Ctrl+Shift+p, 然后输入 snippets 即可对各个文件类型进行编辑。
- 或图形界面操作
- 选择文件类型 (vscode 支持的语言)

- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
 - 用户代码片段用途
 - 进入用户代码片段
 - 用户代码框架
- 4 用户代码片段示例
- 5 vscode 多光标特性

用户代码片段框架：名字与前缀

```
"DateTime Snippet": { // <-- Snippet Name  
  "prefix": "dt", // <-- Prefix  
  "body": [ // <-- Body
```

设置名字和前缀

- 前缀尽可能简明
- 前缀越短，使用时更方便

用户代码片段框架：主体

定义主体

- 每一行需要被双引号包裹
- 每一行结束需要有逗号（最后一行除外）
- 主体的每一行需要被中括号[]包裹

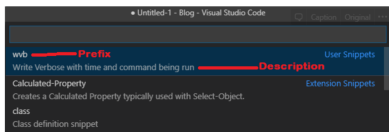
```
"body": [ // <-- Body
  "import datetime", // <-- Body
  "dt = datetime.datetime.now()", // <-- Body
  "print(f\"The time is {dt} and this is a message\")", // <-- Body
], // <-- Body
```

用户代码片段框架：描述

```
}, // <-- Body  
"description": "Prints a message with current time"  
}
```

定义描述

- 描述片段的信息
- 描述会在代码片段列表中显示以便于区分



- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
- 4 用户代码片段示例
- 5 vscode 多光标特性

示例: L^AT_EX

```
1  "insert a figure":{
2  | "prefix": "insertFig",
3  | "body": [
4  |   "\\begin{figure}[htbp]",
5  |   "\\t \\centering",
6  |   "\\t \\includegraphics{$1}",
7  |   "\\t \\caption{$2}",
8  |   "\\t \\end{figure}"
9  | ],
10 | "description": "Insert one figure in latex"
11 | }
```

- \$ 为光标位置, 按 tab 键移动 \$0 为光标最终位置
- 注意要转义特殊字符
- 更多转义参考链接

示例: \LaTeX

```
1 "equation of several lines":{
2   "prefix": "align",
3   "body": [
4     "\\begin{${1|align, align*|}}",
5     "\\label{${2}}",
6     "$0 ",
7     "\\t \\end{${1}}"
8   ],
9   "description": "insert align equation"
10 }
```

- \$ 后 label 相同则内容相同
- 为光标处设置选项
- $\{N|$ 选项 1, 选项 2[, 选项] $\}$

示例: C++

```
1  "myForLoop":{
2      "prefix": "myfor",
3      "body": [
4          "for(int i = 1; i < 10; i = i++)",
5          "{",
6          "${1:cout<<\"hello world\\\"}",
7          "${2|break,continue|}",
8          "$0",
9          "}"
10     ],
11     "description": "custom for loop in cpp"
12 }
```

- 为光标处设置默认值
- `${1: 默认值}`
- 为光标处设置选项
- `${N| 选项 1, 选项 2[, 选项]}`

- ① vscode 基础配置
- ② JSON 文件
- ③ 用户代码片段 User snippets
- ④ 用户代码片段示例
- ⑤ **vscode 多光标特性**
 - 多光标用途
 - 添加多光标

- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
- 4 用户代码片段示例
- 5 **vscode 多光标特性**
多光标用途
添加多光标

为什么用到多光标

```
1 .foo {  
2     Padding: 5;  
3     Margin: 5;  
4     Font-size: 5;  
5 }
```

- 将每个属性首字母小写
- 添加 px 到每行最后
- 与多选快捷键结合使用

- 1 vscode 基础配置
- 2 JSON 文件
- 3 用户代码片段 User snippets
- 4 用户代码片段示例
- 5 **vscode 多光标特性**
多光标用途
添加多光标

使用鼠标添加多光标

使用鼠标：在键盘上按住 Alt，然后鼠标点在第二个 “5” 之前，那么第二个光标就创建好了。现在你可以看到两个光标，第二个光标比第一个要细一点。

```
1  .foo {  
2    padding: 5;  
3    margin: 5;  
4    font-size: 5;  
5  }
```

参考

<https://zh.wikipedia.org/wiki/JSON>

[https:](https://www.tutorialspoint.com/json_simple/json_simple_escape_characters.htm)

[//www.tutorialspoint.com/json_simple/json_simple_escape_characters.htm](https://www.tutorialspoint.com/json_simple/json_simple_escape_characters.htm)

Thanks