

.vimrc文件的语句语法

原创

MikelSun

于 2024-04-02 21:38:19 发布

阅读量826

收藏 2

点赞数 7

分类专栏:

vim编辑器

文章标签:

linux

ubuntu

vim

#vim

vim编辑器 专栏收录该内容

0 订阅 20 篇文章

摘要

本文详细介绍了Vim编辑器的配置文件.vimrc，涵盖了设置选项、映射、函数、条件语句、循环、插件加载、注释、自定义命令、数据自动命令等内容，帮助用户更好地定制Vim的工作环境。

摘要由CSDN通过智能技术生成

本文结构：

a、简介

b、详细解释其中的一些常见语句和语法。

a、.vimrc 文件是 Vim 编辑器用于配置用户设置和自定义行为的文件。当 Vim 启动时，它会读取 .vimrc 文件中的命令和设置，并根据这些指令来配置的行为。

b、.vimrc 文件中的语法和命令基于 Vim 脚本语言，下面是其中的一些常见语句和语法。

1. 设置选项

Vim 有许多内部选项，你可以通过 set 命令来配置它们。例如：

```
1 | set number           " 显示行号
2 | set tabstop=4        " 设置制表符宽度为4
3 | set shiftwidth=4     " 设置缩进的空格数为4
4 | set expandtab         " 使用空格代替制表符
5 | set hlsearch         " 高亮显示搜索结果
```

2. 映射 (Mappings)

映射允许你定义快捷键。常见的映射命令包括 nnoremap（普通模式下非递归映射）、inoremap（插入模式下非递归映射）等。

```
1 | nnoremap <C-j> :w<CR> " 将 Ctrl+j 映射为保存文件
2 | inoremap <C-k> <Esc>:w<CR>i " 在插入模式下，Ctrl+k 映射为退出插入模式、保存文件并重新进入插入模式
```

3. 函数 (Functions)

Vim 脚本也支持函数的定义和调用。

```
1 | function! MyFunction()
2 |     echo "Hello from MyFunction!"
3 | endfunction
4 |
5 | nnoremap <Leader>f :call MyFunction()<CR> " 调用上面定义的函数
```

4. 条件语句 (Conditional Statements)

你可以使用 if、elseif 和 else 语句来根据条件执行不同的命令。

```
1 | if &filetype == 'python'
2 |     setlocal indentexpr=GetPythonIndent()
3 |     setlocal indentkeys+=0{,0},:.,0#,!^F,o,o,*<Return>
4 | endif
```

5. 循环 (Loops)

Vim 脚本支持 for 和 while 循环。



MikelSun

关注

👍 7



🌟 2

💬 0



专栏

```
1 | for i in range(1, 5)
2 |     echo "This is iteration " . i
3 | endfor
```

6. 插件和脚本的加载

你可以在 `.vimrc` 文件中使用 `source` 命令来加载其他的 Vim 脚本文件，或者通过插件管理器（如 [Vim-Plug](#) 、[Vundle](#) 等）来安装和配置插件。

```
1 | source ~/.vim/my_custom_script.vim " 加载自定义脚本
2 | Plugin 'tpope/vim-fugitive' " 使用 Vim-Plug 安装插件（需要先安装 Vim-Plug）
```

7. 注释

`.vimrc` 文件中的注释以双引号 `"` 开头，用于解释代码的目的或行为。

```
" 这是一个注释，不会被 Vim 执行
```

8. 自定义命令 (Custom Commands)

你可以使用 `command!` 语句定义自己的 Vim 命令。

```
command! ToggleNumber toggle number
```

9. 列表和字典

Vim 脚本支持列表（数组）和字典（关联数组）数据结构，可以在 `.vimrc` 文件中定义和使用它们。

```
1 | let myList = ['apple', 'banana', 'cherry']
2 | let myDict = {'name': 'Alice', 'age': 30}
```

10. 自动命令 (Autocmds)

你可以使用 `autocmd` 来定义当特定事件发生时自动执行的命令。

```
autocmd FileType python setlocal indentexpr=GetPythonIndent()
```

