

Vim 配置入门

Vim 是最重要的编辑器之一，主要有下面几个优点。

- 可以不使用鼠标，完全用键盘操作。
- 系统资源占用小，打开大文件毫无压力。
- 键盘命令变成肌肉记忆以后，操作速度极快。
- 服务器默认都安装 Vi 或 Vim。

Vim 的配置不太容易，它有自己的语法，许许多多的命令。我总是记不清楚，所以就整理了下面这篇文章，列出主要配置项的含义。

一、基础知识

Vim 的全局配置一般在 `/etc/vim/vimrc` 或者 `/etc/vimrc`，对所有用户生效。用户个人的配置在 `~/.vimrc`。

如果只对单次编辑启用某个配置项，可以在命令模式下，先输入一个冒号，再输入配置。举例来说，`set number` 这个配置可以写在 `.vimrc` 里面，也可以在命令模式输入。

```
:set number
```

配置项一般都有“打开”和“关闭”两个设置。“关闭”就是在“打开”前面加上前缀“no”。

```
" 打开
set number

" 关闭
set nonumber
```

上面代码中，双引号开始的行表示注释。

查询某个配置项是打开还是关闭，可以在命令模式下，输入该配置，并在后面加上问号。

```
:set number?
```

上面的命令会返回 `number` 或者 `nonumber`。

如果想查看帮助，可以使用 `help` 命令。

```
:help number
```

二、基本配置

```
set nocompatible
```

不与 Vi 兼容（采用 Vim 自己的操作命令）。

(2)

```
syntax on
```

打开语法高亮。自动识别代码，使用多种颜色显示。

(3)

```
set showmode
```

在底部显示，当前处于命令模式还是插入模式。

(4)

```
set showcmd
```

命令模式下，在底部显示，当前键入的指令。比如，键入的指令是 `2y3d`，那么底部就会显示 `2y3`，当键入 `d` 的时候，操作完成，显示消失。

(5)

```
set mouse=a
```

支持使用鼠标。

(6)

```
set encoding=utf-8
```

使用 utf-8 编码。

(7)

```
set t_Co=256
```

启用256色。

(8)

```
filetype indent on
```

开启文件类型检查，并且载入与该类型对应的缩进规则。比如，如果编辑的是 `.py` 文件，Vim 就会找 Python 的缩进规则 `~/.vim/indent/python.vim`。

三、缩进

(9)

```
set autoindent
```

按下回车键后，下一行的缩进会自动跟上一行的缩进保持一致。

(10)

```
set tabstop=2
```

按下 Tab 键时，Vim 显示的空格数。

(11)

```
set shiftwidth=4
```

在文本上按下 >>（增加一级缩进）、<<（取消一级缩进）或者 ==（取消全部缩进）时，每一级的字符数。

(12)

```
set expandtab
```

由于 Tab 键在不同的编辑器缩进不一致，该设置自动将 Tab 转为空格。

(13)

```
set softtabstop=2
```

Tab 转为多少个空格。

四、外观

(14)

```
set number
```

显示行号

(15)

```
set relativenumber
```

显示光标所在的当前行的行号，其他行都为相对于该行的相对行号。

(16)

```
set cursorline
```

光标所在的当前行高亮。

(17)

```
set textwidth=80
```

设置行宽，即一行显示多少个字符。

(18)

```
set wrap
```

自动折行，即太长的行分成几行显示。

```
set nowrap
```

关闭自动折行

(19)

```
set linebreak
```

只有遇到指定的符号（比如空格、连词号和其他标点符号），才发生折行。也就是说，不会在单词内部折行。

(20)

```
set wrapmargin=2
```

指定折行处与编辑窗口的右边缘之间空出的字符数。

(21)

```
set scrolloff=5
```

垂直滚动时，光标距离顶部/底部的位置（单位：行）。

(22)

```
set siderscrollf=15
```

水平滚动时，光标距离行首或行尾的位置（单位：字符）。该配置在不折行时比较有用。

(23)

```
set laststatus=2
```

是否显示状态栏。0 表示不显示，1 表示只在多窗口时显示，2 表示显示。

(24)

```
set ruler
```

在状态栏显示光标的当前位置（位于哪一行哪一列）。

五、搜索

(25)

```
set showmatch
```

光标遇到圆括号、方括号、大括号时，自动高亮对应的另一个圆括号、方括号和大括号。

(26)

```
set hlsearch
```

搜索时，高亮显示匹配结果。

(27)

```
set incsearch
```

输入搜索模式时，每输入一个字符，就自动跳到第一个匹配的结果。

(28)

```
set ignorecase
```

搜索时忽略大小写。

(29)

```
set smartcase
```

如果同时打开了 `ignorecase`，那么对于只有一个大写字母的搜索词，将大小写敏感；其他情况都是大小写不敏感。比如，搜索 `Test` 时，将不匹配 `test`；搜索 `test` 时，将匹配 `Test`。

六、编辑

(30)

```
set spell spelllang=en_us
```

打开英语单词的拼写检查。

(31)

```
set nobackup
```

不创建备份文件。默认情况下，文件保存时，会额外创建一个备份文件，它的文件名是在原文件名的末尾，再添加一个波浪号（~）。

(32)

```
set noswapfile
```

不创建交换文件。交换文件主要用于系统崩溃时恢复文件，文件名的开头是 `!`、结尾是 `.swp`。

(33)

```
set undofile
```

保留撤销历史。

Vim 会在编辑时保存操作历史，用来供用户撤销更改。默认情况下，操作记录只在本次编辑时有效，一旦编辑结束、文件关闭，操作历史就消失了。

打开这个设置，可以在文件关闭后，操作记录保留在一个文件里面，继续存在。这意味着，重新打开一个文件，可以撤销上一次编辑时的操作。撤销文件是跟原文件保存在一起的隐藏文件，文件名以 `.un~` 开头。

(34)

```
set backupdir=~/.vim/.backup//  
set directory=~/.vim/.swp//  
set undodir=~/.vim/.undo//
```

设置备份文件、交换文件、操作历史文件的保存位置。

结尾的 `//` 表示生成的文件名带有绝对路径，路径中用 `%` 替换目录分隔符，这样可以防止文件重名。

(35)

```
set autochdir
```

自动切换工作目录。这主要用在一个 Vim 会话之中打开多个文件的情况，默认的工作目录是打开的第一个文件的目录。该配置可以将工作目录自动切换到，正在编辑的文件的目录。

(36)

```
set noerrorbells
```

出错时，不要发出响声。

(37)

```
set visualbell
```

出错时，发出视觉提示，通常是屏幕闪烁。

(38)

```
set history=1000
```

Vim 需要记住多少次历史操作。

(39)

```
set autoread
```

打开文件监视。如果在编辑过程中文件发生外部改变（比如被别的编辑器编辑了），就会发出提示。

(40)

```
set listchars=tab:»■,trail:■  
set list
```

如果行尾有多余的空格（包括 Tab 键），该配置将让这些空格显示成可见的小方块。

(41)

```
set wildmenu  
set wildmode=longest:list,full
```

命令模式下，底部操作指令按下 Tab 键自动补全。第一次按下 Tab，会显示所有匹配的操作指令的清单；第二次按下 Tab，会依次选择各个指令。