

# VIM语法总结

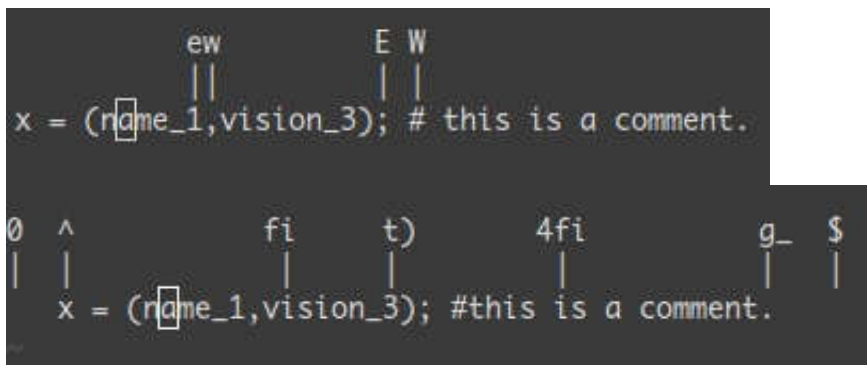
## 命令模式下的编辑

语法	描述	语法	描述
: wq!	强制保存	: q!	强制退出不保存
:set nu	显示行号	:set nonu	取消显示行号
u	undo	ctrl + r	redo
.	重复前面命令	2.	重复前面命令2次
dd	向下删除一行	2dd	删除两行
dt\	删除行之后内容直到遇到“\”		
ye或 yw	从当前位置拷贝到本单词的最后一个字符	gu(gU)	换成小写(大写)字母

## 如何移动

语法	描述	语法	描述
w	移动到单词开头	e	移动到单词结尾
gg	移动到第一行	G	移动到最后一行
0	移动到行的开头	\$	移动到行的结尾
^	移动到行的开头非blank字符	g_	移动到行的结尾非blank字符
fa	到下一个a出现的位置	3fa	第3次出现a的位置
2g	移动到第二行	%	匹配括号移动，包括 (, {, [.
*	匹配光标当前所在单词移到下一个匹配到的单词	#	移动到上一个匹配到的单词
/patten	搜索，按n继续向下搜索	N	按N继续向上搜索

ctrl + o 跳到上一次修改的地方



## 以什么样的形式进入插入模式

语法	描述	语法	描述
A	移动到行末并进入插入模式	x	删除光标所在位置的字符
I	移到到行首并进入插入模式	X	删除光标所在位置之前的字符
o	在当前行之后新建一行并进入插入模式		

## 区域选择

其格式为 `<action>a<object>` 和 `<action>i<object>`

action可以是任何的命令，如 d (删除), y (拷贝), v (可以视模式选择)。

object 可能是：w 一个单词，W 一个以空格为分隔的单词，s 一个句子，p 一个段落。也可以是一个特别的字符：", ' , ), }, ]。

举例：`daw` 删除一个单词

## 块操作

命令模式下按 **ctrl + v** 移动光标选中需要修改的行（区域），然后按 **x** 或者 **d** 删除该内容

批量在行首增加内容：

`ctrl + v >> 0 >> I >> 写下想增加的内容 >> Esc`

```
-- Screen size
type Size = (Int,Int)
type Point = (Int,Int)
type TimePoint = (Int,Int,Int)
screenSize = (25,79)

type Drawer = Size -> Int -> [Float]

fabs x | x < 0 = -x
```

批量在行尾增加内容：

`ctrl + v >> $ >> A >> 写下想增加的内容 >> Esc`

块移动 `<` , `>` 左右缩进。 `=` 自动给缩进。

## 自动补全

vim 下在编辑模式下也有自动补全功能便不是tab键，而是 `ctrl + n`，`ctrl + p` 如果匹配到多个可以按多次来找到自己想要的单词

## 宏录制

`qa` 开始录制宏，并把宏的名字取名为a

`q` 停止录制宏

`@a` 用@来调用宏并执行

`@@` 快捷的执行最新操作的宏

```
95
96
97
98
99
100
101
102
103
103,3
```

`qa >> Y >> p >> ctrl + a >> q >> @a >> 100@@`

**结语：**不论什么命令都可以在前面加数字来表示重复n次该命令

**sandow 于2015年11月1日完结。版权所有，禁止用于其它商业用途**