# vim解决问题的方式

## 技巧1 结识 .命令

* .命令可以重复上次的修改
  + >g会增加从当前行到文档末尾处的缩进层级
  + 每次进入插入模式也会形成一次修改。从进入插入模式的那一刻起，直到返回普通模式为止，vim会记录每一个按键操作。
  + .命令是一个微型的宏 ## 技巧2 不要自我重复
* a命令在当前光标之后添加内容
* a命令在当前行的结尾添加内容 ## 技巧3 以退为进
* f{char}命令让vim查找下一处指定字符出现的位置，如果找到了，就直接把光标移动到那里
* ;命令会重复上次f命令所查找的字符
* s命令先删除光标下的字符，然后进入插入模式 ## 技巧4 执行、重复、回退
* @:可以重复任意ex命令
* &重复上次的:substitute命令

|  |  |  |  |
| --- | --- | --- | --- |
| 目的 | 操作 | 重复 | 回退 |
| 做出一个修改 | {edit} | . | u |
| 在行内查找下一指定字符 | f{char}/t{char} | ; | , |
| 在行内查找上一指定字符 | f{char}/t{char} | ; | , |
| 在文档中查找下一处匹配项 | /pattern<cr> | n | n |
| 在文档中查找上一处匹配项 | ?pattern<cr> | n | n |
| 执行替换 | :s/target/replacement | & | u |
| 执行一系列修改 | qx{changes}q | @x | u |

## 技巧五 查找并手动替换

* \*命令可以查找当前光标下的单词 ## 技巧六 结识.范式 # 普通模式 ## 技巧7 停顿时请移开画笔 ## 技巧8 把撤销单元切成块 ## 技巧9 构造可重复的修改
* 如果在插入模式中使用了<up>、<down>、<left>、<right>这些光标键，将会产生一个新的撤销块。会对.命令的操作产生影响
* 使用更为精准的aw文本对象 ## 技巧10 用次数做简单的算术运算
* <c-a>和<c-x>命令对数字执行加和减操作。在不带数字执行时，它们会逐个加减，但如果带一个次数前缀，那么就可以用它们加减任意整数。
  + 如果光标不在数字上，会把当前光标之上或之后的数值加上[count]
  + vim会把0开头的数字解释为八进制值，可以加入set nrformats=将所有数字设为十进制 ## 技巧11 能够重复，就别用次数 ## 技巧12 双剑合璧，天下无敌 | 命令 | 用途 | | —- | ———————————- | | c | 修改 | | d | 删除 | | y | 复制到寄存器 | | g~ | 反转大小写 （可视化模式下） | | gu | 转为小写 | | gu | 转为大写 | | > | 增加缩进 | | < | 减小缩进 | | = | 自动缩进 | | ! | 使用外部程序过滤{motion}所跨越的行 |
* >>缩进当前行
* guap把整段文字转换为大写
* vim的语法只有一条额外的规则，即当一个操作符命令被连续调用两次时，它会作用于当前行
  + gUgU或gUU可以作用于当前行 # 插入模式 ## 技巧13 在插入模式中可即时更正错误
* <C-h>删除前一个字符（同退格键）
* <C-w>删除前一个单词
* <C-u>删至行首 ## 技巧14 返回普通模式 | 按键操作 | 用途 | |:———|:——————–| | Esc> | 切换普通模式 | | <C-[ | 切换到普通模式 | | <C-o> | 切换到插入-普通模式 |
* 插入-普通模式 在此模式中，可以先执行一个普通模式，执行完后，马上回到插入模式
* zz让当前行显示在窗口正中；<C-o>zz在插入-普通模式中触发这条命名让正中显示后直接j回到插入模式 ## 技巧15 不离开插入模式，粘贴寄存器中的文本
* <C-r>{register}插入寄存器中的内容，但是当textwidth或autoindent选项被激活了的话，那么最终会出现不必要的换行或额外的缩进
* <C-r><C-p>{register}会更智能些，它会按照原意插入寄存器中的文本，并修正任何不必要的缩进 ## 技巧16 随时随地做运算
* 表达式寄存器允许我们做一些运算，并把运算结果直接插入到文档中
* 在插入模式中，输入 <C-r>=就可以访问这一寄存器。这条命令会在屏幕的下方显示一个提示符，我们可以在其后输入要执行的表达式。输入表达式后敲一下<CR>，Vim就会把执行的结果插入到文档的当前位置了。 ## 技巧17 用字符编码插入非常用字符
* 在插入模式下输入<C-v>{code}其中{code}是要插入字符的编码
* 把光标移动到字符上面并按ga命令，下方就会列出编码信息（十进制和十六进制）

|  |  |
| --- | --- |
| 按键操作 | 用途 |
| <C-v>{123} | 以十进制字符编码插入字符 |
| <C-v>u{1234} | 以十六进制字符c编码插入字符 |
| <C-v>{nondigit} | 按原义插入非数字字符 |
| <C-j>{char1}{char2} | 插入以二合字母{char1}{char2}表示的字符 |

## 技巧18 用二合字母插入非常用字符

## 技巧19 用替换模式替换已有文本

* 虚拟替换模式gR