

# Practical Vim

## 摘要

对于vim来说，熟练的盲打是很重要的。一旦完成了vimtutor的训练，并了解如何为vimrc配置一些基本选项之后，就可以用vim完成实际工作了。虽然步履蹒跚，但终有回报。

## 1 Appendix 1 Vimtutor

在终端中键入vimtutor即可进入教程。

表格 1.

Lesson1	
q!	强制退出编辑器并不保存任何改动
x	普通模式下删除字符
A	在当前句的最后添加字符(append)
Lesson2	
d	d is the delete opeator
dw	删除一个单词
d\$	删除从光标到句尾的一切
de	删除从光标到单词尾的一切
w	移动到下一个单词(word)
e	移动到下一个单词尾(end of the word)
0	移动到句子开头
数字numbers	重复操作的次数
eg:d2w	删除两个单词
dd	删除一整行
u	撤销上一个命令(undo)
U	撤销对当前行的所有操作
<C-R>	撤销undo
Lesson3	
P	将删除的文本复制在光标处
r	替换光标处的字符
c	(change)用的方式与d(delete)相同，只是操作后会进入insert模式
cc	删除整句后输入
Lesson4	
<C-G>	显示当前文件名以及光标在文件中的位置
G	移动到文件尾部
gg	移动到文件头部
numbers+G	跳到numbers所指的行数
/(word)	使用/进行单词搜索
/(word)+n	搜索到相应单词后按n会跳转到下一个相应单词
/(word)+N	反方向跳转
?(word)	搜索到相应单词后按n会跳转到上一个相应单词
<C-O>	回到跳转前的位置
<C-I>	<C-O>的逆过程
%	在{, [, (上使用会跳转到对应的括号。对于括号类debug很有用
:s/old/new	仅改动当前句子中的第一个old—>new
:s/old/new/g	仅改动当前句子中的全部的old—>new
:#,#s/old/new/g	仅改动从#—#句子中的全部的old—>new (两个#是行号)
:%s/old/new/g	改动当前文件中的全部的old—>new
:%s/old/new/gc	改动当前文件中的全部的old—>new，但改动前会先请求许可

表格 2.

Lesson5	
:!	后面可以跟随shell命令 eg: :!ls
:w [filename]	将当前文件保存
V	进入visual模式，可以以文件块的形式大面积地选中文本
:r [filename]	将filename中的内容粘贴在光标处
:r ![shell command]	将shell中的内容粘贴在光标处 eg: :r !dir
Lesson6	
o	新开一行并直接进入插入模式
a	在当前光标的后方插入
R	进入replace模式，可以替换多个字符
v	以光标移动的方式选中字符
y	复制选中的文本
yw	复制单词
yy	复制整行
p	粘贴
:set	设置vim配置
Lesson7	
:help	查询 eg: :help w
:+<C-D>	显示可执行命令 eg: :e+<C-D>
tab	自动补全