# Vim入门

## vim是什么？

vim是Vi IMproved，是编辑器Vi的一个加强版，一个极其强大并符合IT工程师（程序员、运维）习惯的编辑器。如果你是一名职业的SE，那么一定在寻找一款出色的能够自由定制、满足灵活编辑功能的编辑器。那么答案，就是vim或者Emacs。而这一套连续的博文，就为您介绍vim编辑器。至于另一款强大的编辑器Emacs，我们会在今后的一个系列博文中看到。

## 配置文件在哪？

Windows系统的配置文件是vim安装目录下的vimrc文件。

Linux系统，RHEL和CentOS的配置文件是/etc/vimrc，Debian和Ubuntu的配置文件是/usr/share/vim/vimrc。

Mac OS X的配置文件是/usr/share/vim/vimrc。

## vimRC中的RC是什么意思？

为什么把这么个没有意义的问题也列出来？原因很简单，就是我在接触vim之初第二个想问的问题就是这个（第一个就是上面的“1.1”）。一般在运行某个软件、环境或工具之初，要初始执行的一些命令，称为Run Commands，即RC。根据Wikipedia.org的记载，RC缩写的用法源自MIT的相容分时移动系统（Compatible Time-Sharing System，CTSS）。所以以后如果你看到了SystemRC或者ScreenRC，也就知其缘由了。

## 三种基本模式

在《大家来学vim》中提到vim有6+5种模式，但除了细分程度不同外，实际上常用的只有3种模式：**NormalMode、Insert Mode和Command-line Mode。**

（1）从Normal Mode进入Insert Mode，可以按i、I、a、A、o、O、r、R即可

（2）从Insert Mode进入Normal Mode，只需要按ESC键即可

1. 从Normal Mode进入Command-line Mode，可以按“:”、“/”、“?”。

“:”表示输入vim命令,“/”表示后面输入要向下搜索的字符串,“?”表示后面输入要向上搜索字符串。

（4）从Command-line Mode进入Normal Mode，只需要按ESC键即可了。

你可以把Normal Mode理解为常态，进入其他任何一种模式，需要某种特殊方式，而从其他特殊模式切换到NormalMode，就按ESC键即可。

## vim中那些最最常用的配置

当你尝试去Google一些其他人的vimrc配置时，你一定会找到一篇叫做《The ultimate vim configuration》的文章，不过它的作者，Amix后来在他的博客上提到[参考2]，这份在google搜索vimrc会排在前十的vim配置文件，如今已经过时了，所以他提供了一些更新信息。

注：如果使用命令行进行设置，只对本次vim移动有效，要对所有vim移动有效，需进入vimrc进行设置。

**（1）颜色设置**

syntax on 开启代码高亮

syntax off 关闭代码高亮

syntax enable 开启代码高亮

**（2）搜索设置**

set hlsearch 开启搜索高亮

set nohlsearch 关闭搜索高亮

set incsearch 输入搜索字符串的同时进行搜索

set ignorecase 搜索时忽略大小写

**（3）用户界面**

set showmode 开启模式显示

set ruler 开启光标位置提示

set number 显示行号

set nonu 不显示行号

set cursorline 强调光标所在行

set cmdheight=1 命令部分高度为1

**（4）编辑辅助配置**

set autoindent 自动缩进

set noautoindent 不自动缩进

set smartindent 智能缩进

set autoread 当文件在外部被改变时，vim自动更新载入

set showmatch 显示匹配的括号

参考：

Run Commands，http://en.wikipedia.org/wiki/Run\_commands

The Ultimate vim Configuration(vimRC)，<http://amix.dk/blog/post/19486>

# 常用的状态切换按键

## Normal Mode -> Command-line Mode

“:” 表示输入vim命令

“/” 表示后面输入要向下搜索的字符串

“?” 表示后面输入要向上搜索字符串。

## 回到 Normal Mode

ESC

## 打开、保存、退出、改变文件（Buffer）

:e <path/to/file> 打开一个文件

:saveas <path/to/file> 另存为<path/to/file>文件

:w 保存文件（前提是已有文件）

:w! 强制保存文件（前提是用户有修改文件访问权限的权限）

:n1,n2 w 将n1行到n2行的数据另存为名为filename文件

:x 如果文件有更改，则保存后退出；否则直接退出

:q 退出缓冲区（即目前编辑文档使用的Vim界面）

:q! 强制退出缓冲区而不保存(qa! 强制退出所有的正在编辑的文件)

:wq,ZZ 保存文件并退出缓冲区

:wq! 强制保存文件并退出缓冲区（前提是用户有修改文件访问权限的权限）

:w+文件名 另存为名为filename文件

## 多个文件切换

:bn和:bp 如果同时打开多个文件，可以再多个文件间切换上一个或下一个。

:n和:N 同上。

## Normal Mode -> Insert Mode

i 小写字母i，在光标位置插入（常用 100i+插入的字符串+esc,则插入100个字符串）

a 小写字母a，在光标的下一个位置插入

I 大写字母I，在光标所在行的第一个非空格处插入

A 大写字母A，在光标所在行的最后一个字符处插入

o 小写字母o，在光标所在行的下一行处插入新行

O 大写字母O，在光标所在行的上一行处插入新行

r 小写字母r，替换光标所在处的字符一次

R 大写字母R，持续替换光标所在处的字符，直到按下ESC

# 常用光标移动按键

## 光标字符移动

h/j/k/l 左/下/上/右

## 光标词移动

w 光标移动到后一词的词首

W 光标移动到后一词的词首且忽略标点符号

e 光标移动到后一词的词尾

E 光标移动到后一词的词尾且忽略标点符号

b 光标移动到前一词的词首

B 光标移动到前一词的词首且忽略标点符号

ge 光标移动到前一词的词尾

gE 光标移动到前一词的词尾且忽略标点符号

## 移动到行首和行尾

$ 光标移动到行尾（End），注意要按Shift键

0 光标移动到行首（Home）

^ 光标移动到行首第一个非空白字符（Home），注意要按Shift键

## 移动到指定字符

f+字符 移动到某个字符（f前加数字表示移动到本行第几个字符）

F+字符 反向移动

t+字符 移动到字符的前一个字符（T表示反向）

; 重复移动命令

, 反向重复

## 光标句移动

) 光标移动到后一句的句首(以.为判断)

( 光标移动到前一句的句首

## 光标行移动

G 光标移动到文档的最后一行的第一个非空白字符

nG 光标移动到文档的第n行，相当于”:n”

gg 光标移动到文档的第1行的第一个非空白字符，相当于”1G”，也相当于”:1″

：+数字 移动到指定行

50% 移动到50%位置

## 光标段移动

} 光标移动到下一段的段首(以.为判断)

{ 光标移动到上一段的段首

ctr+p

## 光标页移动

Ctrl+f 向下翻页（Pagedown）

Ctrl+b 向上翻页（Pageup）

Ctrl+d 向下翻半页

Ctrl+u 向上翻半页

H 光标移动到目前显示页的第一行

M 光标移动到目前显示页的中间行

L 光标移动到目前显示页的最后一行

## 括号匹配

% 配合“(”和“)”使用，即在括号上按“%”号，寻找相匹配的另一半，同理对{}、[]也是适用的

## 光标自由移动

Ctrl+o 回到光标的上一位置

## 确定当前位置

ctrl+g

## 使用标记

跳转：即移出本行之外的位置

使用“G”进行跳转时，可以使用’’返回该次跳转之前的位置

ctrl+o(older) 跳转到一个较老的位置

ctrl+I 跳转到一个较新的位置

:jumps 显示跳往的位置列表，“>”回到最后一个跳转位置

ma 使用a标记当前位置

‘a 跳转到标记位置

:marks 显示所有的标记

特殊标记

‘ 跳转前的光标位置

“ 最后编辑的光标位置

[ 最后修改的开始位置

] 最后修改的结束位置

# 常用编辑移动按键

## 撤销与重做

u 撤销

ctrl+r 重做

## 删除移动（delete）

dd 删除光标所在行

ndd 删除从光标所在行开始，向下的n行

d1G 删除从光标所在行开始，到第一行的所有行

dG 删除从光标所在行开始，到最后一行的所有行

d$ 删除从光标所在位置，到行尾的所有字符

d0 删除从光标所在位置，到行首的所有字符

# 查找

## 简单查找

在普通模式下按在/即进入查找模式：

/字符串 查找指定字符串（查找的是第一个字符串，移动到下一个用n,3n表示移动到第三个查找的字符串）

？字符串 反向查找

n 重复查找（N 反向查找）

## 替换中的特殊字符

1. **查找整个单词**

**利用文中单词：**

利用文中单词查找整个单词，移动光标到指定单词，并按“\*”键。（“#”是反向）

利用文中单词部分匹配 g+\*或g+#

**在命令行输入：**

字符+\> 匹配单词末尾

\<+字符 匹配单词开头

1. **行首与行尾匹配**

^字符 字符匹配行首

$字符 字符匹配行尾

**3.匹配任何字符**

. 匹配任何单个字符 如，c.m->com

## 历史查找命令

/+向上箭头 查看上一个查找的字符

/+字符+向上箭头 部分匹配查找字符

## 命令

:set ignorecase 忽略大小写（:set noignorecase）

:set hlsearch 高亮查找(:set nohlsearch)

# 替换

vi/vim 中可以使用 :s 命令来替换字符串，其完整的形式为：**[range]s/search/replace/[flags][count]**。

## [range]的表示形式

**1.单行：**

（1）不写range或者点（“.”）表示替换范围为光标所在行；

（2）单个数字，表示替换某一行，如1、33、$分别表示替换第1、33以及最后一行；

（3）标记，如’as/ vivian / sky表示替换之前由标记a(ma做标记)进行标记的行；

（4）相对位置在前，如.+3s/ vivian / sky，表示对当前光标所在行的下面三行的内容进行替换；

（5）相对位置在后，如s/ vivian / sky/g4，表示对当前行以及它下面的三行的内容进行替换。

**2.多行：**

1,.s/ vivian /sky:替换第一行到当前行中每一行的第一个vivan为sky;

33,$ s/ vivian /sky: :替换第33行到最后一行中每一行的第一个vivan为sky;

%：表示替换所有行，相当于：1,$

:%s/vivian/sky/（等同于 :g/vivian/s//sky/） 替换每一行的第一个 vivian 为 sky

**:g/vivian/s//sky/g->表示全部替换的另一种格式**

1. **指定替换范围**

:/from/to/;/s/=/<>/g，表示将本行from到to中的“=”替换为“<>”

## [flags]表示形式

1.无：只对指定范围内的第一个匹配项进行替换

2.g(global):对指定范围内的所有匹配项进行替换

3.c(confirm):在替换前请求用户确认

replace with Teacher(y/n/a/q/l/^E/^Y)?的回答如下：

（1）y YES：执行这个替换

（2）n No：取消这个替换

（3）a All：执行所有替换而不要在询问

（4）q Quit：退出而做任何改动

（5）l Last:替换完当前匹配点后退出

（6）Ctrl+E:向上翻滚一行

（7）Ctrl+Y: 向下翻滚一行

4.i(ignorecase):不区分大小写

5.e:忽略执行过程中的错误

注：上面的所有flags都可以组合起来使用，比如gc表示对指定范围内的所有匹配项进行替换，并且在每一次替换之前都会请用户确认

6.:%s/vivian/sky/g（等同于 **:g/vivian/s//sky/g**） 替换每一行中所有 vivian 为 sky

从上述替换命令可以看到：

* g 放在命令末尾，表示对搜索字符串的每次出现进行替换；
* 不加 g，表示只对搜索字符串的首次出现进行替换；
* g 放在命令开头，表示对正文中所有包含搜索字符串的行进行替换操作。

## 其他分隔符

还可以使用 # 、+、?作为分隔符，此时中间出现的 / 不会作为分隔符，?表示向上搜索替换。

:s#vivian/#sky/# 替换当前行第一个 vivian/ 为 sky/

:%s+/oradata/apras/+/user01/apras1+ （使用+ 来替换 / ）： /oradata/apras/替换成/user01/apras1/

# 正则表达式

## 查找

# 其他命令

## 缩进

先进行缩进设置：

:set shiftwidth=4 softtablestop=4 expandtab

缩进快捷键：shift+>>或shift+<<