## vi/vim 的使用

基本上 vi/vim 共分为三种模式，分别是**命令模式（Command mode）**，**输入模式（Insert mode）**和**底线命令模式（Last line mode）**。 这三种模式的作用分别是：

### **命令模式：**

用户刚刚启动 vi/vim，便进入了命令模式。

此状态下敲击键盘动作会被Vim识别为命令，而非输入字符。比如我们此时按下i，并不会输入一个字符，i被当作了一个命令。

以下是常用的几个命令：

* **i** 切换到输入模式，以输入字符。
* **x** 删除当前光标所在处的字符。
* **:** 切换到底线命令模式，以在最底一行输入命令。
* gg 贯标跳到开头
* / 查找，如 输入/user按下Enter，查找user，接着输入n查找下一处
* dd 删除一行
* ndd 删除关闭所在的n行

若想要编辑文本：启动Vim，进入了命令模式，按下i，切换到输入模式。

命令模式只有一些最基本的命令，因此仍要依靠底线命令模式输入更多命令。

### **输入模式**

在命令模式下按下i就进入了输入模式。

在输入模式中，可以使用以下按键：

* **字符按键以及Shift组合**，输入字符
* **ENTER**，回车键，换行
* **BACK SPACE**，退格键，删除光标前一个字符
* **DEL**，删除键，删除光标后一个字符
* **方向键**，在文本中移动光标
* **HOME**/**END**，移动光标到行首/行尾
* **Page Up**/**Page Down**，上/下翻页
* **Insert**，切换光标为输入/替换模式，光标将变成竖线/下划线
* **ESC**，退出输入模式，切换到命令模式

### **底线命令模式**

在命令模式下按下:（英文冒号）就进入了底线命令模式。

底线命令模式可以输入单个或多个字符的命令，可用的命令非常多。

在底线命令模式中，基本的命令有（已经省略了冒号）：

* q 退出程序
* w 保存文件
* ! 强制执行

按ESC键可随时退出底线命令模式。

简单的说，我们可以将这三个模式想成底下的图标来表示：



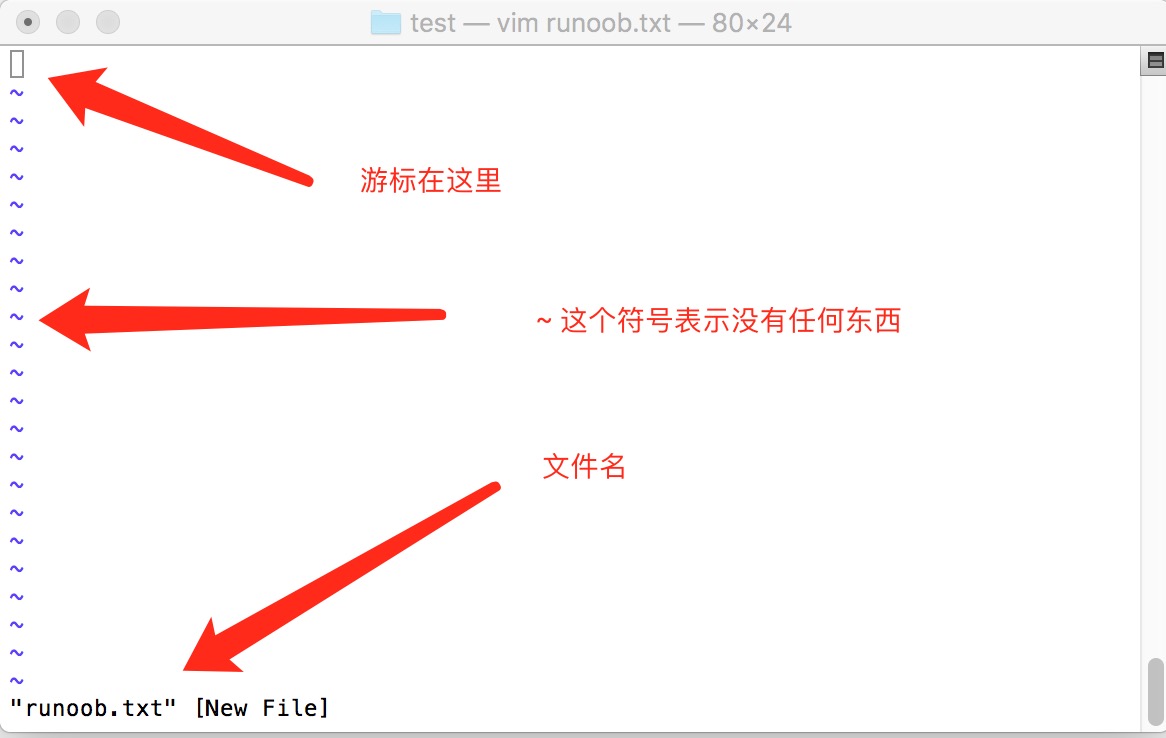
## vi/vim 使用实例

### **使用 vi/vim 进入一般模式**

如果你想要使用 vi 来建立一个名为 test.txt 的文件时，你可以这样做：

$ vim runoob.txt

直接输入**vi 文件名**就能够进入 vi 的一般模式了。请注意，记得 vi 后面一定要加文件名，不管该文件存在与否！



### **按下 i 进入输入模式(也称为编辑模式)，开始编辑文字**

在一般模式之中，只要按下 i, o, a 等字符就可以进入输入模式了！

在编辑模式当中，你可以发现在左下角状态栏中会出现 –INSERT- 的字样，那就是可以输入任意字符的提示。

这个时候，键盘上除了 **Esc** 这个按键之外，其他的按键都可以视作为一般的输入按钮了，所以你可以进行任何的编辑。



### **按下 ESC 按钮回到一般模式**

好了，假设我已经按照上面的样式给他编辑完毕了，那么应该要如何退出呢？是的！没错！就是给他按下 **Esc** 这个按钮即可！马上你就会发现画面左下角的 – INSERT – 不见了！

### **在一般模式中按下 **:wq** 储存后离开 vi**

OK，我们要存档了，存盘并离开的指令很简单，输入 **:wq** 即可保存离开！



OK! 这样我们就成功创建了一个 runoob.txt 的文件。

## vi/vim 按键说明

除了上面简易范例的 i, Esc, :wq 之外，其实 vim 还有非常多的按键可以使用。

### **第一部份：一般模式可用的光标移动、复制粘贴、搜索替换等**

|  |  |
| --- | --- |
| **移动光标的方法** | |
| h 或 向左箭头键(←) | 光标向左移动一个字符 |
| j 或 向下箭头键(↓) | 光标向下移动一个字符 |
| k 或 向上箭头键(↑) | 光标向上移动一个字符 |
| l 或 向右箭头键(→) | 光标向右移动一个字符 |
| [Ctrl] + [f] | 屏幕『向下』移动一页，相当于 [Page Down]按键 (常用) |
| [Ctrl] + [b] | 屏幕『向上』移动一页，相当于 [Page Up] 按键 (常用) |
| 0 或功能键[Home] | 这是数字『 0 』：移动到这一行的最前面字符处 (常用) |
| $ 或功能键[End] | 移动到这一行的最后面字符处(常用) |
| G | 移动到这个档案的最后一行(常用) |
| nG | n 为数字。移动到这个档案的第 n 行 |
| v | 按v开始选择，通过上下左右移动光标，经过的区域为选择的文本，按v结束选择， |
| d | 删除光标选择的文本 |
| y | 复制光标选择的文本 |
| **搜索替换** | |
| /word | 向光标之下寻找一个名称为 word 的字符串 |
| n | 这个 n 是英文按键。代表重复前一个搜寻的动作。举例来说， 如果刚刚我们执行  /vbird ，则按下 n 后，会向下继续搜寻下一个名称为 vbird 的字符串。 |
| **:%s/word1/word2/g** | 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2 ！(常用) |
| **:%s/word1/word2/gc** | 从第一行到最后一行寻找 word1 字符串，并将该字符串取代为 word2 ！  且在取代前显示提示字符给用户确认 (confirm) 是否需要取代！(常用) |
| **删除、复制与贴上** | |
| x | 在一行字当中，x 为向后删除一个字符 (相当于 [del] 按键) |
| dd | 删除游标所在的那一整行(常用) |
| yy | 复制游标所在的那一行(常用) |
| u | 复原前一个动作。(常用) |
| [Ctrl]+r | 重做上一个动作。(常用) |
| p, P | p 为将已复制的数据在光标下一行贴上，P 则为贴在游标上一行！ (常用) |

### **第二部份：一般模式切换到编辑模式的可用的按钮说明**

|  |  |
| --- | --- |
| **进入输入或取代的编辑模式** | |
| i, I | 进入输入模式(Insert mode)： i 为『从目前光标所在处输入』， I 为『在目前所在行的第一个非空格符处开始输入』。 (常用) |
| r, R | 进入取代模式(Replace mode)： r 只会取代光标所在的那一个字符一次；R会一直取代光标所在的文字，直到按下 ESC 为止；(常用) |
| [Esc] | 退出编辑模式，回到一般模式中(常用) |

### **第三部份：一般模式切换到指令行模式的可用的按钮说明**

|  |  |
| --- | --- |
| **指令行的储存、离开等指令** | |
| :w | 将编辑的数据写入硬盘档案中(常用) |
| :w! | 若文件属性为『只读』时，强制写入该档案。不过，到底能不能写入，  还是跟你对该档案的档案权限有关啊！ |
| :q | 离开 vi (常用) |
| :q! | 若曾修改过档案，又不想储存，使用 ! 为强制离开不储存档案。 |
| 注意一下啊，那个惊叹号 (!) 在 vi 当中，常常具有『强制』的意思～ | |
| :wq | 储存后离开，若为 :wq! 则为强制储存后离开 (常用) |
| :w [filename] | 将编辑的数据储存成另一个档案（类似另存新档） |
| :r [filename] | 在编辑的数据中，读入另一个档案的数据。  亦即将 『filename』 这个档案内容加到游标所在行后面 |
| **vim 环境的变更** | |
| :set nu | 显示行号，设定之后，会在每一行的前缀显示该行的行号 |
| :set nonu | 与 set nu 相反，为取消行号！ |

特别注意，在 vi/vim 中，数字是很有意义的！数字通常代表重复做几次的意思！ 也有可能是代表去到第几个什么什么的意思。

举例来说，要删除 50 行，则是用 『50dd』 对吧！ 数字加在动作之前，如我要向下移动 20 行呢？那就是『20j』或者是『20↓』即可。