

vim 的模式

创建加密文档

```
vim -x file1
```

普通模式

vim打开后默认的模式就是普通方式

安Esc进入普通模式，在该模式下使用方向键或者 h, j, k, l 键可以移动光标。

按键	说明
h	左
l	右 (小写 L)
j	下
k	上
w	移动到下一个单词
b	移动到上一个单词

一般按 i 或者 a 进入 插入模式

命令	说明
i	在当前光标处进行编辑
I	在行首插入
A	在行末插入
a	在光标后插入编辑
o	在当前行后插入一个新行
O	在当前行前插入一个新行
cw	替换从光标所在位置后到一个单词结尾的字符

普通模式下退出 vim

普通模式下按下 Shift+zz 即可保存退出 vim。

普通模式下删除 vim 文本信息

进入普通模式，使用下列命令可以进行文本快速删除：

命令	说明
x	删除光标所在的字符
X	删除光标所在前一个字符
Delete	同 x
dd	删除整行
dw	删除一个单词（不适用中文）
d\$或D	删除至行尾
d^	删除至行首
dG	删除到文档结尾处
d1G	删至文档首部

除此之外，你还可以在命令之前加上数字，表示一次删除多行，如：

2dd 表示一次删除 2 行。


普通模式下 . 可以用作重复上一次的命令的操作

N+命令,表示该命令重复执行N次


复制及粘贴文本

- 普通模式中使用 `y` 复制
- 普通模式中, `yy` 复制光标所在的整行 (`3yy` 表示复制 3 行)
- 普通模式中, `y^` 复制至行首, 或 `y0`。不含光标所在处字符。
- 普通模式中, `y$` 复制至行尾。含光标所在处字符。
- 普通模式中, `yw` 复制一个单词。
- 普通模式中, `y2w` 复制两个单词。
- 普通模式中, `yG` 复制至文本末。
- 普通模式中, `y1G` 复制至文本开头。
- 普通模式中使用 `p` 粘贴
- 普通模式中, `p`(小写)代表粘贴至光标后 (下)
- 普通模式中, `P`(大写)代表粘贴至光标前 (上)

1. `p` (小写 `p`) :

- **作用:** 在当前光标位置的**后面**或**下面**粘贴内容。
- **效果:**
 - 如果是粘贴一行 (之前用 `dd` 或其他命令剪切/复制的行), 则粘贴到光标所在行的**下一行**。
 - 如果是粘贴部分文本 (例如用 `yw` 复制的一个单词), 则粘贴到光标所在字符的**后面**。

2. `P` (大写 `P`) :

- **作用:** 在当前光标位置的**前面**或**上面**粘贴内容。
- **效果:**
 - 如果是粘贴一行, 粘贴到光标所在行的**上一行**。
 - 如果是粘贴部分文本, 则粘贴到光标所在字符的**前面**。

替换和撤销(Undo)命令

替换和 Undo 命令都是针对普通模式下的操作：

命令	说明
<code>r+<待替换字母></code>	将游标所在字母替换为指定字母
<code>R</code>	连续替换，直到按下 <code>Esc</code>
<code>cc</code>	替换整行，即删除游标所在行，并进入插入模式
<code>cw</code>	替换一个单词，即删除一个单词，并进入插入模式
<code>C(大写)</code>	替换游标以后至行末
<code>~</code>	反转游标所在字母大小写
<code>{n}u</code>	撤销一次或 n 次操作
<code>U(大写)</code>	撤销当前行的所有修改
<code>Ctrl+r</code>	redo，即撤销 undo 的操作

- 普通模式下输入 `15G`，跳转到 15 行
- 普通模式下输入 `>>` 整行将向右缩进（使用，用于格式化代码超爽）
- 普通模式下输入 `<<` 整行向左回退
- 普通模式下输入 `:` 进入命令行模式下对 `shiftwidth` 值进行设置可以控制缩进和回退的字符数

shiftwidth 命令

`shiftwidth` 命令是指上一节 `>>` 命令产生的缩进（可以简写成`sw`），普通模式下输入 `:` 进入命令行模式下对 `shiftwidth` 值进行设置可以控制缩进和回退的字符数。

获取目前的设定值：

```
:set shiftwidth?
```

设置缩进为 10 个字符：

```
:set shiftwidth=10
```

输入 `ESC` 回到普通模式，再次尝试 `>>` 看缩进量是否变化。

调整文本位置

命令行模式下输入 `:ce`(center)命令使本行内容居中：

```
:ce
```

命令行模式下输入 `:ri`(right)命令使本行文本靠右：

```
:ri
```

命令行模式下输入 `:le`(left)命令使本行内容靠左：

```
:le
```

快速查找

普通模式下输入 `/` 然后键入需要查找的字符串，按回车后就会进行查找。`?` 与 `/` 功能相同，只不过 `?` 是向上而 `/` 是向下查找。

进入查找之后，输入 `n` 和 `N` 可以继续查找。`n` 是查找下一个内容，`N` 查找上一个内容。

高级查找

- 普通模式下输入 `*` 寻找光标所在处的单词
- 普通模式下输入 `\#` 同上，但 `\#` 是向前（上）找，`*` 则是向后（下）
- 普通模式下输入 `g*` 同 `*`，但部分符合该单词即可
- 普通模式下输入 `g\#` 同 `\#`，但部分符合该单词即可

以上查找 `n`，`N` 的继续查找命令依然可以用。

- 命令行模式下输入 `:n` 编辑 2.txt 文件，可以加 `!` 即 `:n!` 强制切换，之前一个文件的输入没有保存，仅仅切换到另一个文件
- 命令行模式下输入 `:N` 编辑 1.txt 文件，可以加 `!` 即 `:N!` 强制切换，之前文件内的输入没有保存，仅仅是切换到另一个文件

进入 vim 后打开新文件

- 命令行模式下输入 `:e 3.txt` 打开新文件 3.txt
- 命令行模式下输入 `:e#` 回到前一个文件
- 命令行模式下输入 `:ls` 可以列出以前编辑过的文档
- 命令行模式下输入 `:b 2.txt` (或者编号) 可以直接进入文件 2.txt 编辑
- 命令行模式下输入 `:bd 2.txt` (或者编号) 可以删除以前编辑过的列表中的文件项目
- 命令行模式下输入 `:e! 4.txt`，新打开文件 4.txt，放弃正在编辑的文件
- 命令行模式下输入 `:f` 显示正在编辑的文件名
- 命令行模式下输入 `:f new.txt`，改变正在编辑的文件名字为 new.txt

恢复文件

由于在线环境无法复现此场景，请在本地尝试。

如果因为断电，终端意外关闭等原因造成文档没有保存，可以采用恢复方式。

vim 在编辑的时候会自动在当前文件目录下生成一个交换文件，一般以 `<filename>.swp` 的格式保存，如果有多个版本的交换文件，还可能是 `.swp`，`.swm` 等字母。注意这个交换文件的更新不是实时的，因为实时的更新会占用磁盘，影响系统其他正常进程的速度。

当 vim 意外终止且没有保存已编辑的内容时，可以使用交换文件对文件进行恢复，注意需要在编辑这个文件的地方进行编辑，因为交换文件默认在编辑的目录生成。

例如假设 1.txt 意外关闭，我们可以输入 vim 然后输入 `:recover 1.txt`。

```
vim
:recover 1.txt

# 出现如下提示按下回车即可恢复，如果有多个版本需要输入对应的交换文件前的数字选择要恢复的版本。
Using swap file ".1.txt.swp"
Original file "~/test/1.txt"
Recovery completed. You should check if everything is OK.
(You might want to write out this file under another name
and run diff with the original file to check for changes)
You may want to delete the .swp file now.
```

或者输入 `vim -r` 后直接选择用于恢复的文档的交换文件即可：

```
vim -r 1.txt
```

插入模式

插入文本,按 *ESC*返回普通模式

命令模式

命令行模式下保存文档

从普通模式输入 *:* 进入命令行模式，输入 *w* 回车，保存文档。输入 *:w <filename>* 可以将文档另存为其他文件名或存到其它路径下。

命令行模式下退出 vim

从普通模式输入 *:* 进入命令行模式，输入 *wq* 回车，保存并退出编辑。

以下为其它几种退出方式：

命令	说明
<i>:q!</i>	强制退出，不保存
<i>:q</i>	退出
<i>:wq!</i>	强制保存并退出
<i>:w <文件路径></i>	另存为
<i>:saveas 文件路径</i>	另存为
<i>:x</i>	保存并退出
<i>:wq</i>	保存并退出

可视模式

可视模式命令简介

- 在普通模式下输入 *v*（小写），进入字符选择模式，就可以移动光标，光标走过的地方就会选取。再次按下 *v* 后就会取消选取。
- 在普通模式下输入 *Shift+v*（小写），进入行选择模式，按下 *V* 之后就会把整行选取，您可以上下移动光标选更多的行，同样，再按一次 *Shift+v* 就可以取消选取。
- 在普通模式下输入 *Ctrl+v*（小写），这是区域选择模式，可以进行矩形区域选择，再按一次 *Ctrl+v* 取消选取。
- 在可视模式下输入 *d* 删除选取区域内容
- 在可视模式下输入 *y* 复制选取区域内容

视窗操作

视窗操作简介

vim 可以在一个界面里打开多个窗口进行编辑，这些编辑窗口称为 vim 的视窗。打开方法有很多种，例如可以使用在命令行模式下输入 `:new` 打开一个新的 vim 视窗，并进入视窗编辑一个新文件（普通模式下输入 `Ctrl+w` 也可以），除了 `:new` 命令，下述列举的多种方法也可以在命令模式或普通模式下打开新的视窗：

注意：快捷键可能会与浏览器的快捷键冲突，可换为 IE 浏览器进行实验或者在浏览器设置里禁用浏览器快捷键。

- 命令行模式下输入 `:sp 1.txt` 打开新的水平分屏视窗来编辑 1.txt
- 命令行模式下输入 `:vsp 2.txt` 打开新的垂直分屏视窗来编辑 2.txt
- 普通模式下 `Ctrl+w s` 将当前窗口分割成两个水平的窗口
- 普通模式下 `Ctrl+w v` 将当前窗口分割成两个垂直的窗口
- 普通模式下 `Ctrl+w q` 即 `:q` 结束分割出来的视窗。如果在新视窗中有输入需要使用强制符！即`:q!`
- 普通模式下 `Ctrl+w o` 打开一个视窗并且隐藏之前的所有视窗
- 普通模式下 `Ctrl+w j` 移至下面视窗
- 普通模式下 `Ctrl+w k` 移至上面视窗
- 普通模式下 `Ctrl+w h` 移至左边视窗
- 普通模式下 `Ctrl+w l` 移至右边视窗
- 普通模式下 `Ctrl+w J` 将当前视窗移至下面
- 普通模式下 `Ctrl+w K` 将当前视窗移至上面
- 普通模式下 `Ctrl+w H` 将当前视窗移至左边
- 普通模式下 `Ctrl+w L` 将当前视窗移至右边
- 普通模式下 `Ctrl+w -` 减小视窗的高度
- 普通模式下 `Ctrl+w +` 增加视窗的高度

在vim执行外部命令

`!!ls`:用于显示当前目录的内容 `!rm FILENAME`:用于删除名为FILENAME的文件 `:w FILENAME`:将当前vim中正在编辑的文件另存为FILENAME文件

vim的功能设定

vim 的功能设定

可以在编辑文件的时候进行功能设定，如命令行模式下输入 `:set nu` (显示行数)，设定值退出 vim 后不会保存。要永久保存配置需要修改 vim 配置文件。

vim 的配置文件 `~/.vimrc` (实验楼环境中配置文件在 `/etc/vim/vimrc`)，可以打开文件进行修改，不过务必小心不要影响 vim 正常使用。

获取目前的设定

- 命令行模式下输入 `:set` 或者 `:se` 显示所有修改过的配置
- 命令行模式下输入 `:set all` 显示所有的设定值
- 命令行模式下输入 `:set <option>?` 显示 option 的设定值
- 命令行模式下输入 `:set nooption` 取消当前设定值

set 功能的说明

- 命令行模式下输入 `:set autoindent(ai)` 设置自动缩进
- 命令行模式下输入 `:set autowrite(aw)` 设置自动存档，默认未打开
- 命令行模式下输入 `:set background=dark` 或 `light`，设置背景风格
- 命令行模式下输入 `:set backup(bk)` 设置自动备份，默认未打开
- 命令行模式下输入 `:set cindent(cin)` 设置 C 语言风格缩进