

# Vim：命令模式

## 命令模式

前面的教程大多集中在 Normal 模式

命令模式提供了除了 Normal 模式的编辑方式

命令模式的操作对象以“行”为基本单位

normal 和 global 命令提供强大的批量行操作

## Ex 命令格式

`: [range] {excommand} [args]`

- `range` : 作用的范围, 不给的话默认是本行
- `excommand` : 特殊的命令, 适用于 Command 模式
- `args` : 后续的参数, 视命令而定

一些 Ex Command ( `[x]` 为寄存器, 是可选项 ) :

- `: [range] delete [x]` : 删除 `range` 中的行 (到寄存器 `x` ), `delete` 可简写为 `d`
- `: [range] yank [x]` : 复制 `range` 中的行 (到寄存器 `x` ), `yank` 可简写为 `y`
- `: [range] print` : 将 `range` 中行打印出来, `print` 可简写为 `p`

## range 与 address：指定范围

range 由一个或两个 address 构成，即 {address} 或 {address},{address}

address 可以是：

- {lineno}：行号，如 3 代表第三行（0 代表第一行上面的虚拟行）
- \$：最后一行
- .：光标所在行
- /{pattern}/：下一个 pattern 所在的行

address 可以做加减法，.+3 代表光标往下第三行，\$-3 代表倒数第 4 行

## address 组合成 range

address 组合出 range (可以混用) : 例子:

- 1, 3 : 文件的 1-3 行
- ., .+4 : 当前-当前往下 4 行 (共 5 行)
- \$-3, \$ : 文件的最后 4 行
- 1, 3 delete : 删除 1~3 行
- ., .+4 yank : 复制当前-当前往下 4 行
- \$-3, \$ : 打印文件的最后 4 行

% : 特殊的 range , 代表当前文件的所有行

'< / '> : 可视模式中选中范围的开头和结尾 (可视模式下直接按 : 可以直接设置)

## 行的复制、移动、粘贴

- `:[range] copy {address}` : 把 `range` 中的行复制到 `address` 后面
- `:[range] move {address}` : 把 `range` 中的行移动到 `address` 后面
- `:[address] put [x]` : 把寄存器 `x` 的内容粘贴到 `address` 后面

0 作为虚拟行的 `address`, 可以用来将内容插入第一行

## Ex 命令

删除、拷贝、打印：

- `: [range] delete [x]`
- `: [range] yank [x]`
- `: [range] print`

复制、移动、粘贴

- `: [range] copy {address}`
- `: [range] move {address}`
- `: [address] put [x]`

Ex 命令和 Normal 模式下的编辑操作有区别吗？

- 命令模式下的编辑操作以“行”为单位
- 命令模式下的编辑操作无需移动光标

## 批量操作：normal 命令

格式： `: [range] normal {commands}`

含义：对 `range` 中的所有行执行 Normal 模式下的命令 `commands`

- 将 `range` 设置为 `%`，可以对整个文件的所有行执行
- `: [range] normal .`：配合 `.` 命令，效果拔群

常用做法：先做一次修改操作，再用 normal 命令在指定的行上完成操作

- `.` 命令只能记录一次修改，用宏可以实现记录多个操作

`: [range] normal @ {register}`

常用做法：先把想要的操作录制成宏，再用 normal 命令在指定的行上重放宏



## 批量操作：global 命令

格式： `: [range] global / {pattern} / [cmd]`

含义：对 `range` 中包含 `pattern` 的所有行执行 Command 模式下的 Ex 命令

`[cmd]`：Ex 命令，不给的话默认是打印（`print`）

注意，`normal` 命令也是 Ex 命令！

`: [range] global / {pattern} / normal {commands}`：对 `range` 中所有带 `pattern` 的行，执行 Normal 模式下的命令 `commands`

例子：

- `:% global /TODO/delete`：删除所有带 TODO 的行

## 替换命令

```
: [range] s/{pattern}/{string}/[flags]
```

将 `pattern` 替换为 `string`

`flags` :

- `g` : 替换每一行的所有匹配
- `i` : 忽视大小写
- `c` : 替换前进行确认
- `n` : 计数而不是替换

`:%s/Vim//gn` : 统计文件中所有 `Vim` 出现的次数（此时替换为什么无所谓，加了 `n` 就不会执行替换操作）

## 小结

- 了解常见的 Ex 命令
- 了解 `address` 和 `range` 的概念
- **(重要)** 掌握如何使用 normal 命令批量运行 Normal 模式的命令
- **(重要)** 掌握如何使用 global 命令批量运行 Command 模式的 Ex 命令