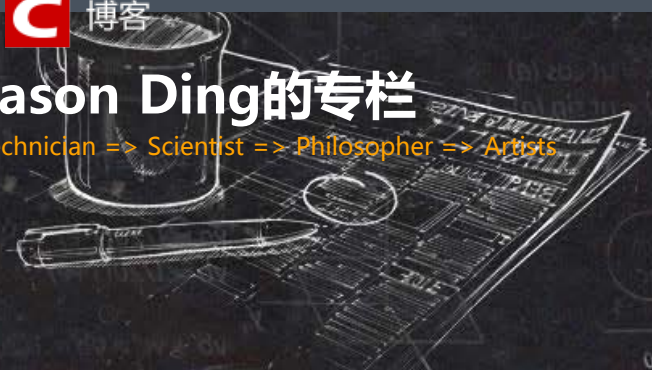


Technician => Scientist => Philosopher => Artists



 快速回复

## 算入门



译文: 3篇 评论: 71条

【Computer Vision】 (14)

- 用于普通模式: 输入命令时。
- 用于可视模式: 可视区域高亮并输入命令时。
- 用于操作符等待模式: 操作符等待中 ( “d” , “y” , “c” 等等之后)。
- 用于插入模式: 也用于替换模式。
- 用于命令行模式: 输入 “:” 或 “/” 命令时。

- 【Design Pattern】 (2)
- 【C++ Programming】 (11)
- 【Python Programming】 (7)
- 【Scala Programming】 (36)
- 【About Programming】 (7)
- 【Spark】 (23)
- 【Energy Big Data】 (2)
- 【Developer Kits】 (9)
- 【ML News Flash】 (3)
- 【Thoughts】 (8)

文章搜索

博客专栏



**Akka编程实战**  
文章：5篇  
阅读：8391



**Scala编程实例**  
文章：19篇  
阅读：24113



**机器学习实验**  
文章：11篇  
阅读：37759

- 文章存档
- 2016年05月 (3)

2016年03月 (9)

2016年02月 (5)

2016年01月 (10)

2015年12月 (1)
- 展开

- 阅读排行
- Spark交互式分析平台Ap (9699)

【机器学习基础】支持向 (7331)

【scikit-learn】交叉验证 (6258)

【Vim】使用Vundle管理 (5599)

【scikit-learn】评估分类 (5244)

【python】matplotlib.py (4662)

【E2LSH源码分析】LSI (4320)

Command 命令	Normal 常规模式	Visual 可视化模式	Operator Pending 运算符模式	Insert Only 插入模式	Command Line 命令行模式
:map	y	y	y		
:nmap	y				
:vmap		y			
:omap			y		
:map!				y	y
:imap				y	
:cmap					y

### 几种模式的介绍

1. Normal Mode
- 也就是最一般的普通模式，默认进入vim之后，处于这种模式。
2. Visual Mode
- 一般译作可视模式，在这种模式下选定一些字符、行、多列。  
在普通模式下，可以按v进入。
3. Insert Mode
- 插入模式，其实就是指处在编辑输入的状态。普通模式下，可以按i进入。
4. Select Mode
- 选择模式。用鼠标拖选区域的时候，就进入了选择模式。和可视模式不同的是，在这个模式下，选择完了高亮区域后，敲任何按键就直接输入并替换选择的文本了。和windows下的编辑器选定编辑的效果一致。普通模式下，可以按gh进入。
5. Command-Line/Ex Mode
- 命令行模式和Ex模式。两者略有不同，普通模式下按冒号(:)进入Command-Line模式，可以输入各种命令，  
使用vim的各种强大功能。普通模式下按Q进入Ex模式，其实就是多行的Command-Line模式。

### 命令的组合

同Vim下的其他命令一样，命令的名字往往由好几段组成。前缀作为命令本身的修饰符，微调命令的效果。对于map而言，可能有这么几种前缀：

- none  
表示非递归。  
递归的映射。其实很好理解，也就是如果键a被映射成了b，c又被映射成了a，如果映射是递归的，那么c就被映射成了b。
- n  
表示在普通模式下生效
- v  
表示在可视模式下生效
- i

【scikit-learn】scikit-learn	(4292)
【E2LSH源码分析】p稳	(4240)
【特征工程】特征工程技	(4209)

评论排行	
【E2LSH源码分析】p稳	(9)
【Scala】Vector内部结构	(4)
【计算机视觉】提取视频	(4)
机器学习与深度学习资料	(3)
本博客系列文章导航	(3)
【机器学习基础】理解为	(3)
番茄工作法分享	(2)
【机器学习基础】从感知	(2)
【leetcode边做边学】从	(2)
【E2LSH源码分析】LSH	(2)

最新评论	
【机器学习实验】使用朴素贝叶斯	yongge1981: 大师你好, 有几个问题请教一下, 希望你不要吝赐教, 多谢多谢! 1, 下载的数据集在哪个目录下, 文件名是什么? ...
【机器学习实验】使用朴素贝叶斯	yongge1981: 大师你好, 有几个问题请教一下, 希望你不要吝赐教, 多谢多谢! 1, 下载的数据集在哪个目录下, 文件名是什么? ...
利用ffmpeg和opencv进行视频的	JAVA_CLASSm: 很有参考价值.
【机器学习中的数学】贝叶斯框架	laowantongliulizhi: 挺好的东西
【scikit-learn】交叉验证及其用	aawurongrong: 写的很好
【机器学习基础】正则化	hadoopX: 简直太棒了!!!!

广告

表示在插入模式下生效

- c

表示在命令行模式下生效

## 普通模式的映射命令

map

命令格式：

```
:map {lhs} {rhs}
```

其含义是，在:map作用的模式中将键序列 {lhs} 映射为 {rhs}，{rhs}可进行映射扫描，也就是可递归映射。

举例：

```
:map td :tabnew .<cr>
```

含义：在其作用模式（普通、可视、操作符）下，输入td等价于输入 `:tabnew .<cr>`。而普通模式下输

入 `:tabnew .<cr>` 就是打开当前目录

如果再定义绑定 `:map ts td`，就是指在其作用模式下输入ts等价于td，也就是打开当前目录。不过如果没有特殊需要，一般不建议递归映射。

noremap

:noremap和:map命令相对，作用模式和命令格式都相同，只不过不允许再对{rhs}进行映射扫描，也就是{lhs}

定义后的映射就是{rhs}的键序列，不会再对{rhs}键序列重新解释扫描。它一般用于重定义一个命令，当然如

果:map不需要递归映射的话，建议使用:noremap

比如：

```
:noremap ts td
```

它的意思是在其作用模式下，输入ts就是输入td，但是和:map不同的是，此时td再不会做进一步扫描解释。虽然之前已经定义了td，但是不会对td再做扫描。

unmap

:unmap是对应取消:map绑定的 { lhs }，作用模式相同，命令格式:unmap {lhs}。

例如：

```
:unmap td
```

就是取消在其作用模式中td的绑定，比如之前td被绑定为 `:tabnew .<cr>`，此时此绑定消失。

mapclear

:mapclear时对应取消所有:map绑定的，慎用！

只用于普通模式的

:nmap

:nmap是:map的普通模式板，也就是说其绑定的键只作用于普通模式。

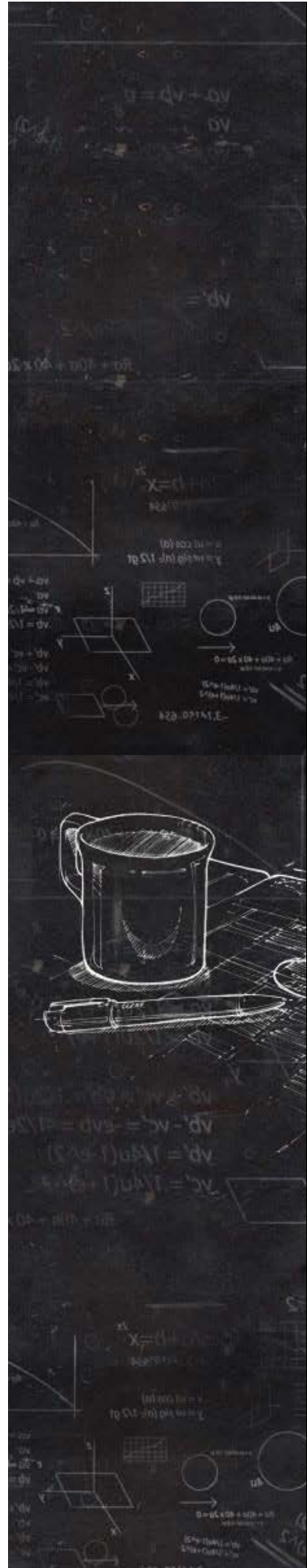
例如：

```
:nmap td :tabnew .<cr> 和 :map td :tabnew .<cr> 在普通模式下等效
```

:nnoremap

:nnoremap和:nmap的关系和:noremap和:map的关系一样，只是:nmap的非递归版

:nunmap



:nunmap和:nmap的关系和:unmap和:map的关系一样，取消:nmap的绑定。

**:nmapclear**

:nmapclear是对应取消所有:map绑定的，慎用！

另外

{rhs} 之前可能显示一个特殊字符:

- \* 表示它不可重映射
- & 表示仅脚本的局部映射可以被重映射
- @ 表示缓冲区的局部映射

到这一步你可以轻松的长吸一口气，因为相关的命令已经都了解了，记不住没关系，可以随时:help map一下。

键表

```
<k0> - <k9> 小键盘 0 到 9
<S-...> Shift + 键
<C-...> Control + 键
<M-...> Alt + 键 或 meta + 键
<A-...> 同 <M-...>
<Esc> Escape 键
<Up> 光标上移键
<Space> 插入空格
<Tab> 插入Tab
<CR> 等于 <Enter>
```

特殊参数

有些特殊参数必须映射命令的后边，在其他任何参数的前面。

**<buffer>**

**<buffer>** 如果这些映射命令的第一个参数是 **<buffer>**，映射将只局限于当前缓冲区（也就是你此时正编辑的文件）内。比如：

```
:map <buffer> ,w /a<CR>
```

它的意思是在当前缓冲区里定义键绑定，“,w”将在当前缓冲区里查找字符a。同样你可以在其他缓冲区里定义：

```
:map <buffer> ,w /b<CR>
```

比如我经常打开多个标签(:tabedit)，想要在各自己标签里定义“,w”键绑定，那么你只要在每个标签页里分别定义就可，其作用域也只在各自的标签里。同样要清除这些缓冲区的键绑定也要加上 **<buffer>** 参数，比如：

```
:unmap <buffer> ,w
```

```
:mapclear <buffer>
```

**<silent>**

**<silent>** 是指执行键绑定时不在命令行上回显，比如：

```
:map <silent> ,w /abcd<CR>
```

你在输入,w查找abcd时，命令行上不会显示/abcd，如果没有 **<silent>** 参数就会显示出来。

**<special>**





`<special>` 一般用于定义特殊键怕有副作用的场合。比如：

```
:map <special> <F12> /Header<CR>
```

`<expr>`

`<expr>`. 如果定义新映射的第一个参数是 `<expr>`，那么参数会作为表达式来进行计算，结果使用实际使用的，例如：

```
:inoremap <expr> . InsertDot()
```

这可以用来检查光标之前的文本并在一定条件下启动全能 (omni) 补全。

一个例子：

```
1 let counter = 0
2 inoremap <expr> <C-L> ListItem()
3 inoremap <expr> <C-R> ListReset()
4
5 func ListItem()
6 let g:counter += 1
7 return g:counter . ' '
8 endfunc
9
10 func ListReset()
11 let g:counter = 0
12 return ''
13 endfunc
```

在插入模式下，CTRL-L插入顺序的列表编号，并返回；CTRL-R复位列表编号到0，并返回空。

`<unique>`

`<unique>` 一般用于定义新的键映射或者缩写命令的同时检查是否该键已经被映射，如果该映射或者缩写已经存在，则该命令会失败

`<Leader>` 和mapleader变量

mapleader变量对所有map映射命令起效，它的作用是将参数 `<leader>` 替换成mapleader变量的值，比如：

```
:map <Leader>A oanother line<Esc>
```

如果mapleader变量没有设置，则用默认的反斜杠代替，因此这个映射等效于：

```
:map \A oanother line<Esc>
```

意思时输入\A键时，在下一行输入another line并返回到普通模式。

如果设置了mapleader变量，比如：

```
let mapleader = " , "
```

那么就等效于：

```
:map ,A oanother line<Esc>
```

`<LocalLeader>` 和maplocalleader变量

`<LocalLeader>` 和`<Leader>` 类似，只不过它只作用于缓冲区。

因此在设置mapleader和maplocalleader时最好区分开，不要出现冲突。

转载请注明作者Jason Ding及其出处

GitCafe博客主页(<http://jasonding1354.gitcafe.io/>)

Github博客主页(<http://jasonding1354.github.io/>)

CSDN博客(<http://blog.csdn.net/jasonding1354>)



简书主页([http://www.jianshu.com/users/2bd9b48f6ea8/latest\\_articles](http://www.jianshu.com/users/2bd9b48f6ea8/latest_articles))

百度搜索jasonding1354进入我的博客主页



顶  
0

踩  
0

- ▲ 上一篇
- ▼ 下一篇
- 【Vim】使用Vundle管理配置Vim基本插件
- 【Linux Mint】通过vino-server共享Cinnamon桌面

我的同类文章

【Developer Kits】 (8)

•

【Linux】使用update-alter...

2016-01-06

阅读 1551

•

【Linux Mint】通过vino-se...

2015-05-02

阅读 1132

•

【Vim】让Vim成为你双手...

2015-04-29

阅读 690

•

【项目工具】接触Maven

2014-11-29

阅读 852

•

【Git】认识各种开源协议...

2015-05-11

阅读 904

•

【Vim】使用Vundle管理配...

2015-04-29

阅读 5585

•

Sublime Text使用指南

2014-12-28

阅读 1123

•

Eclipse中高效的快捷键、...

2014-10-31

阅读 1788

阿里云

云服务器9.9元 学生专享

阿里云助你实现创业梦想

参考知识库



.NET知识库

3674 关注 | 833 收录

猜你在找

- Windows Server 2012 DHCP Server 管理

▪ Windows Server 2012 DHCP Server 管理

▪ 使用CSDN Markdown编辑器写博客

▪ Windows Server 2012 Active Directory 管理

▪ Windows Server 2012 Active Directory 管理
- aaa安装监控软件nagioscacti软件

▪ 三方库

▪ Go开发工具

▪ GitHub上史上最全的iOS开源项目分类汇总

▪ GIT上优秀的IOS框架

查看评论



暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry






Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

 网站客服  杂志客服  微博客服  webmaster@csdn.net  400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技  
京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved 