

Neovim 手册

最终稿

完整版

精订版

2025 年 10 月 1 日 校订

Neovim 概述

Neovim 是 Linux 下著名的编辑器 Vim (Vi 通常是 Vim 的符号链接) 的现代化分支，在保证兼容 Vim 的配置和插件的情况下，增加了对脚本语言 lua 的支持，因此增多了不少用 lua 语言编写的插件，初始界面较为简陋，经过定制后可以达成各种各样的功能，同时 Neovim 增加了对鼠标的支持，因此除了使用传统 Vim 键位移动也可以直接用鼠标点击（不过效率更低）

Neovim 下载

以 Linux 为例

首先先下载 Neovim：

apt 和 dnf 都可以直接下载 Neovim，

输入命令：

```
sudo apt install neovim
```

或

```
sudo dnf install neovim
```

即可。若想要更新版本，可以使用 ppa 或手动克隆仓库 <https://github.com/neovim/neovim> 并编译

其次解决终端 Neovim 图标可能的问题：

前往 Neovide 官网 neovide.dev 并下载 Linux 版本

Neovim普通模式键表（不全）

A	在行尾插入
a	在光标后插入
	文本对象表示一个
B	移动到前一个WORD头
b	移动到前一个word尾
C	剪切到行尾并插入
c	后跟移动对象，移动并剪切后插入
	两个c连用，同大写S
D	剪切到行尾
d	后跟移动对象，移动并剪切
	两个d连用，剪切当前行
E	移动到WORD尾
e	移动到word尾
F	
f	
G	跳转到文件尾
	跟在数字后，跳转到指定行
g	两个g连用，跳转到文件头
H	
h	向左移动
I	在行首插入
i	在光标前插入
	文本对象表示内部
J	
j	向下移动
K	

k	向上移动
L	
l	向右移动
M	
m	
N	
n	
O	在上方新建行并插入
o	在下方新建行并插入
P	在光标前粘贴
p	在光标后粘贴
	跟在寄存器后，粘贴指定寄存器内容
	文本对象跟在a/i后表段落
Q	
q	开始录制宏
	第二次按下q结束录制宏
R	进入替换模式
r	替换当前字符
S	清空当前行并插入
s	清空当前字符并插入
	文本对象跟在a/i后表句子
T	
t	
U	
u	撤销
V	
v	进入可视化模式
W	移动到下一个WORD头

W	文本对象跟在a/i后表WORD
w	移动到下一个word头
	文本对象跟在a/i后表word
X	
x	剪切当前字符
Y	
y	后跟移动对象，移动并复制
	两个y连用，复制当前行
Z	
z	
0	移动到行首
@	调用宏
\$	移动到行尾
%	跟在数字后，跳转到指定百分比
^	同0，移动到行首
:	进入命令模式
'	文本对象跟在a/i后表"对象
"	后跟",0,a-z,表寄存器
	跟在"后，表无名寄存器
	文本对象跟在a/i后表""对象
?	往文件头查找
/	往文件尾查找
<	两个<连用，当前行取消一层缩进
	文本对象跟在a/i后表<>对象
>	两个>连用，当前行增加一层缩进
(文本对象跟在a/i后表()对象
{	文本对象跟在a/i后表{}对象
Ctrl+R	重做

Ctrl+V	进入列模式
注：word代表单词（分界线为空格、标点、横杠等），WORD代表广义单词（分界线只有空格）；文本对象里a包含边界，i不包含边界	

Neovim普通模式键表分类版（不全）

移动命令

B	移动到前一个WORD头
b	移动到前一个word尾
E	移动到WORD尾
e	移动到word尾
G	跳转到文件尾
	跟在数字后，跳转到指定行
g	两个g连用，跳转到文件头
h	向左移动
j	向下移动
k	向上移动
l	向右移动
W	移动到下一个WORD头
w	移动到下一个word头
O	移动到行首
^	同O，移动到行首
\$	移动到行尾
%	跟在数字后，跳转到指定百分比

文本操作

D	剪切到行尾
d	后跟移动对象，移动并剪切
	两个d连用，剪切当前行
P	在光标前粘贴
p	在光标后粘贴
	跟在寄存器后，粘贴指定寄存器内容
r	替换当前字符

x	剪切当前字符
y	后跟移动对象，移动并复制
	两个y连用，复制当前行
<	两个<连用，当前行取消一层缩进
>	两个>连用，当前行增加一层缩进
插入模式	
A	在行尾插入
a	在光标后插入
C	剪切到行尾并插入
c	后跟移动对象，移动并剪切后插入
	两个c连用，同大写S
I	在行首插入
i	在光标前插入
O	在上方新建行并插入
o	在下方新建行并插入
S	清空当前行并插入
s	清空当前字符并插入
文本对象	
a	表示一个
i	表示内部
p	跟在a/i后表段落
s	跟在a/i后表句子
W	跟在a/i后表WORD
w	跟在a/i后表word
'	跟在a/i后表"对象
"	跟在a/i后表""对象
<	跟在a/i后表<>对象
(文本对象跟在a/i后表()对象

{	文本对象跟在a/i后表{}对象
其他模式	
R	进入替换模式
v	进入可视化模式
:	进入命令模式
Ctrl+V	进入列模式
特殊对象	
"	后跟",0,a-z,表寄存器
	跟在"后，表无名寄存器
特殊操作	
q	开始录制宏
	第二次按下q结束录制宏
u	撤销
@	调用宏
?	往文件头查找
/	往文件尾查找
Ctrl+R	重做
注：word代表单词（分界线为空格、标点、横杠等），WORD代表广义单词（分界线只有空格）；文本对象里a包含边界，i不包含边界	

Neovim其他模式键表（不全）

插入模式

Ctrl-R	跟寄存器名，粘贴指定寄存器内容
Esc	退回普通模式

可视化模式

B	移动到前一个WORD头
b	移动到前一个word尾
E	移动到WORD尾
e	移动到word尾
G	跳转到文件尾
	跟在数字后，跳转到指定行
g	两个g连用，跳转到文件头
h	向左移动
j	向下移动
k	向上移动
l	向右移动
W	移动到下一个WORD头
w	移动到下一个word头
0	移动到行首
^	同0，移动到行首
\$	移动到行尾
%	跟在数字后，跳转到指定百分比
d	剪切选中文本
y	复制选中文本
c	剪切选中文本并进入插入模式
Esc	退回普通模式

Neovim底行模式常用命令（不全）

:ab <a> 	以后但凡输入a并输入非字母数字字符，自动替换为b
:bn	下一个缓冲区
:bp	上一个缓冲区
:bd	删除缓冲区
:e	新编辑一个文件
:n	新建缓冲区
:q	退出
:qa	退出所有
:w	保存文件
:wa	保存所有
:wq	保存退出
:wqa	保存所有，退出所有
:lua	运行lua
:make	在当前目录下运行make
:!	运行Shell命令

Neovim使用案例

默认在普通模式下，<CR>表示换行，<C-键>表示Ctrl+键，<M-键>表示Meta（现在的Alt）+键

清空本行并进入插入模式	S
	ddO
删除本行第一个单词	0dw
删除本行第一个广义单词	0dW
删除当前双引号里的所有内容，不包括双引号	di"
删除本行及往下的25行	d25j
替换本行最后一个单词	\$bcw
在当前目录下运行make	:make<CR>
	:!make<CR>
清空文件内容	ggdG
跳转到第51行	51G
	:51<CR>
在行前插入并粘贴h寄存器内容	I<C-r>h
	Oi<C-r>h
跳转到文件50%	50%
强制退出	:qa!<CR>
保存所有并退出所有	:wqa!<CR>
替换当前字符	r
剪切到行尾	d\$
	D
	v\$d
粘贴上次复制的内容	"0p
剪切7个单词	d7w

剪切7个单词

v7wd

Neovim常用配置（不全）

以下配置均为vim格式，init.lua需调用vim.cmd[[内容]]

<code>syntax on</code>	开启语法高亮
<code>set nu</code>	启用行号
<code>set noshowmode</code>	不在底行显示当前模式
<code>set nohlsearch</code>	搜索时不高亮
<code>nnoremap</code>	设定普通模式键映射
<code>cnoremap</code>	设定底行模式键映射
<code>vnoremap</code>	设定可视化模式键映射
<code>inoremap</code>	设定插入模式键映射
<code>set ai</code>	启用自动缩进
<code>let g:var=</code>	定义变量，var为变量名

Neovim 相关网址

官网：neovim.io

插件：dotfyle.com

Github 地址：[neovim/neovim](https://github.com/neovim/neovim)

Neovide 相关网址

官网：neovide.dev

Github 地址：[neovide/neovide](https://github.com/neovide/neovide)

我的 Neovim 配置

Github 地址：[hunter-hongg/Nvim-Setting](https://github.com/hunter-hongg/Nvim-Setting)

我的 Neovim 练习文件

Github 地址：[hunter-hongg/Nvim-Practise](https://github.com/hunter-hongg/Nvim-Practise)

Neovim 提示

1. 使用 Neovide, 可以解决终端字体问题, 同时 Neovide 的字体的 `<= =>` `<=>` `->` `==` 等字体自动优化
2. Neovim 常用四种模式: 普通模式 (命令模式), 底行模式, 插入模式, 可视化模式; 其中尽量多地使用普通模式, 底行模式用于保存、编辑文件等操作; 可视化模式其实可以等价在普通模式下运行, 例如 `d25G` 在可视化模式下表示为 `v25Gd`, 但可视化模式支持多次移动, 普通模式下的 `d c y` 做不到; 插入模式其实就是日常接触的文本编辑器, 但尽量少地停留在插入模式, 会影响 Neovim 的效率
3. Neovim 配置文件有两个, `~/.config/nvim/init.{vim, lua}`, 一般的插件配置都是 lua 所以我使用 `init.lua`
4. Neovim 配置可以进行改动
5. 尽量少使用鼠标 (尤其是在初学的时候), 可以在配置文件里添加 `set mouse= (init.vim)` 或 `vim.cmd("set mouse=")(init.lua)` 来禁用鼠标, 同时应当禁用上下左右, 使用 `hjkl` 进行移动, 效率更高

Neovim 补充

1. 进行/或?搜索后，可以按下 n 进入下一个匹配项，或按下 N 进入上一个匹配项
2. 大写的 J 可以合并两行
3. 我的配置的使用方法可以参见
~/.config/nvim/lua/conf/myconf.lua, 注意有一条 Ctrl+P
打开文件模糊匹配里面没写
4. 我的配置还有多方面增强

Neovim 补充 20251005

1. 后跟移动的命令如 `d c y`, 大写表示 `n$`, 例如 `D` 表示 `d$`, `Y` 表示 `y$`, 以此类推
2. `hjkl` 等命令前可以跟数字, 表示重复 `n` 次对应命令
3. `$`前可以跟数字, 表示后 `n` 行的最后
4. 后跟移动的命令如 `d c y` 以及 `<` `>` 等部分命令, 双写表示对当行执行操作
5. 后跟移动的命令如 `d c y` 的双写时, 可以在头部或中间插入数字, 如 `d3d` 和 `3yy`, 表示删除/复制 (以及删除并插入) `N` 行