# VIM

## عملیات پایه

نوشتن شماره خط : :set number

فعال کردن کرسر لاین : :set cursorline

نکته : برای پایدار کردن بعضی از این کانفیگ ها (مثلا نشان دادن شماره خط یا کرسرلاین) کافی است که یک فایل به اسم .vimrc را در مسیر /home/user(my user,e.g: ali) ایجاد می کنیم (برای کابر روت در مسیر /root) در صورتی که وجود دارد آن فایل را باز میکنیم و دو دستور بالا (بدون : ) می زنیم و فایل را ذخیره می کنیم. حالا در دفعه بعد این دو دستور هنگام بازشدن اعمال شده اند.

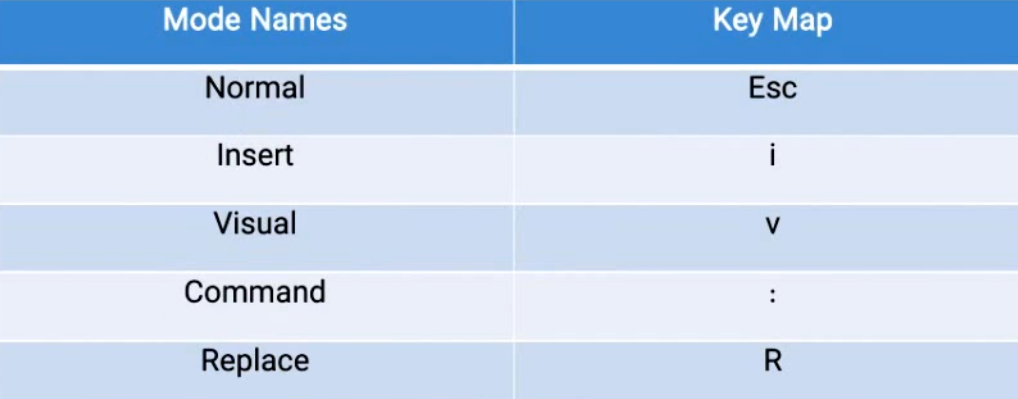
Undo کردن : دکمه u

Paste کردن : p

Delete کردن : d

کپی کردن (yank کردن) : y

## حالتهای کاری :



که وقتی وارد می شویم در حالت normal هستیم و vim منتظر دریافت دستورات است.

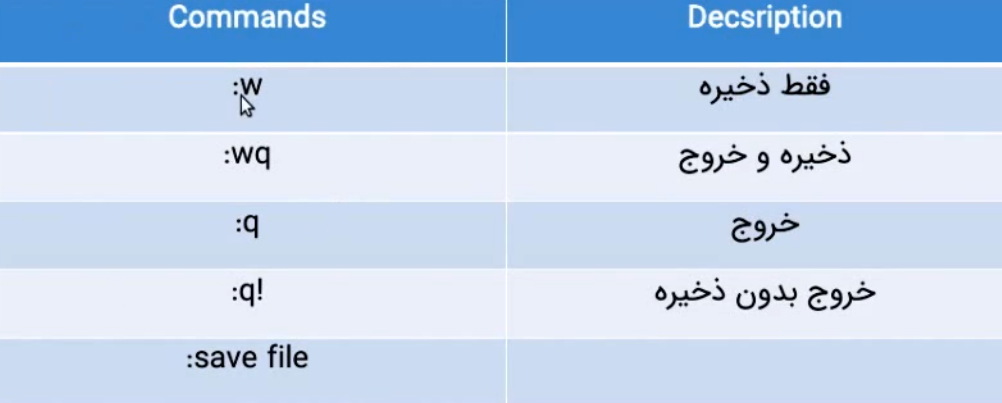
حالت visual خود به سمت نوع زیر تقسیم می شود.

۱- v : که در این حالت با استفاده از کرسر می توان بالا، پایین و چپ و راست رفت و کاراکتر به کاراکتر انتخاب کرد.

۲- V: که در این مدل خطی می توان انتخاب کرد

۳- Ctl + v : که در این مدل visual block فعال می شود که می توان به صورت ستونی انتخاب کرد.

## حالت های ذخیره



به وسیله حالت آخر می توان هم زمان تغییرات را در یک فایل جدید ذخیره کرد. مثلا اگر فایل حاضر file باشد و ما دستور save file2 را بنویسیم تغییرات در فایل دوم که file2 است ذخیره می شود. (در واقع شبیه save as است) البته معمولا به جای کلمه save می توان از w نیز استفاده کرد.

## جابجایی

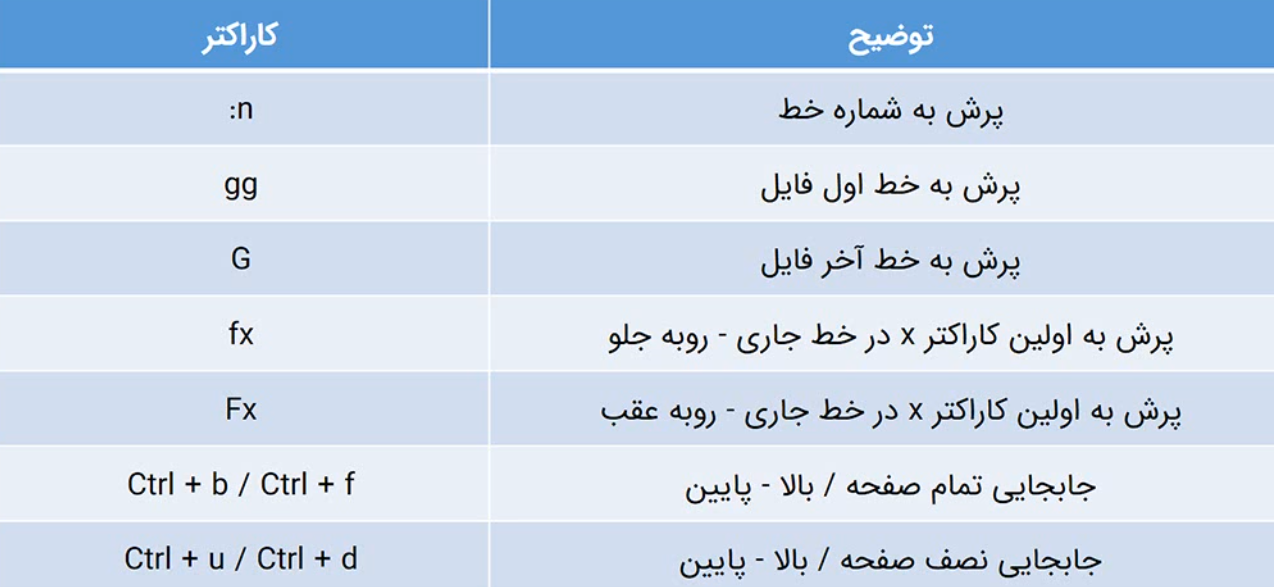


حروف روی کیبورد که کنار همدیگر هستند یعنی hjkl به ترتیب چپ و پایین و بالا و راست می روند و اگر این حروف با عدد هم همراه باشند به همان اندازه انجام می شود. مثلا 4h ، ۴ حرف به چپ می رود.

w در حالت نرمال یک کلمه به بعد می رود و b یک کلمه به قبل باز می گردد و e یک کلمه به بعد می رود ولی انتهای کلمه بعدی می ایستد.

0 به ابتدای خط فعلی می رود و $ به انتهای خط فعلی می رود.

## پرش در متن

رفتن به خط شماره ۲۵ : :25

رفتن به ابتدای فایل : gg

رفتن به انتهای فایل : G

رفتن به اولین کاراکتر مشخص در خط جاری رو به جلو : fx که اولین x به سمت جلو در خط حاضر می رود

رفتن به اولین کاراکتر مشخص در خط جاری رو به عقب : Fx که اولین x به سمت عقب در خط حاضر می رود

رفتن به صفحات :‌Ctrl + b صفحه قبل و Ctrl + f صفحه بعد

رفتن به نیم صفحات : Ctrl + u نیم صفحه بالا و Ctrl + d نیم صفحه پایین

## تعریف bookmark



بوکمارک یک نشانه است که به سادگی به وسیله آن نشانه به آن خط خاص از فایل می پریم.

به وسیله m و بعد از آن یک حرف می توان یک بوکمارک تعریف کرد. یعنی می رویم روی خطی که میخواهیم بوکمارک کنیم و مثلا mr را می زنیم حالا هرجای فایل که باشیم با زدن `r به این خط بر می گردیم.

برای دیدن بوکمارک ها می توان با نوشتن :marks لیست آنها را مشاهده کرد.

برای پاک کردن بوکمارکها از دستورr,l delmarks (در واقع r و l همان کاراکترهای نشانه هستند) استفاده می کنیم.

پاک کردن تا بوکمارک : d`r از جایی که هستیم تا بوکمارک r را پاک میکند

کپی کردن تا بوکمارک : y`r از جایی که هستیم تا بوکمارک r را کپی می کند

## دستورات مربوط به کپی



به طور کلی برای پاک کردن (یا کات کردن) از d استفاده می شود و از dw برای پاک کردن یک کلمه و از dd برای پاک کردن کل خط با d2d برای پاک کردن ۲ خط استفاده می شود.

به طور کلی برای کپی کردن (یا کات کردن) از y استفاده می شود و از yw برای کپی کردن یک کلمه و از yy برای کپی کردن کل خط با y2y برای کپی کردن ۲ خط استفاده می شود.

به طور کلی از p برای paste کردن بعد از محل کرسر و از P برای paste کردن قبل از محل کرسر استفاده می شود.

## عوض کردن

## 

برای عوض کردن یک کاراکتر: روی کاراکتری که می خواهیم عوض کنیم می رویم و دکمه r را میزنیم و حالا کاراکتر جدید را وارد می کنیم.

عوض کردن یک یا چند خط: با cc یا cnc (چند خط) را پاک می کند و وارد insert mode می شود و منتظر ورود کاراکترهای جدید از ما می ماند.

عوض کردن از جایی که هستیم تا انتهای خط : C از جایی که هستیم تا انتهای خط را پاک می کند و منتظر ما می ماند تا کاراکتر های جدید را وارد کنیم.

Undo کردن : u عمل undo را انجام می دهد

Redo کردن : Ctrl + r عمل redo را انجام می دهد.

تکرار دستور قبل : با . دستور قبلی هر چه که باشد تکرار می شود.

تمام دستورات کپی و paste را می توان بعد از انواع حالت های کپی در مدل visual نیز استفاده کرد (یعنی مثلا در حالت visual block یک ستون را انتخاب کرد و با d آن ستون را حذف کرد)



کاراکتر ~ حروف کوچک و بزرگ را به هم تبدیل می کند .

## جستجو



جستجو رو به جلو‌: با استفاده از /word

جستجو رو به عقب : با استفاده از ?word

مورد بعدی جستجو شده : n به مورد بعدی جستجو شده می رود و N به مورد قبلی جستجو شده می رود

جستجو و جایگزینی یک کلمه در خط : می توان از :s/old/new/ در همان خطی که هستیم مقدار قدیمی را با مقدار جدید جایگزین می کند

جستجو و جایگزینی تمام کلمات در خط : می توان از :%s/old/new/ در همان خطی که هستیم تمام مقادیر قدیمی را با مقدار جدید جایگزین می کند

جستجو و جایگزینی تمام کلمات در تمام فایل : می توان از :%s/old/new/g تمام مقادیر قدیمی را با مقدار جدید در کل فایل جایگزین می کند

جستجو و جایگزینی تمام کلمات در تمام فایل و بررسی : می توان از :%s/old/new/gc تمام مقادیر قدیمی را با مقدار جدید در کل فایل جایگزین می کند و به هنگام جایگزینی می پرسد که جایگزین کند یا نه

جستجو و جایگزینی تمام کلمات در تمام فایل در حالت case-insensitive: می توان از :%s/old/new/gi تمام مقادیر قدیمی را با مقدار جدید در کل فایل جایگزین می کند بدون در نظر گرفتن بزرگ یا کوچک بودن حروف

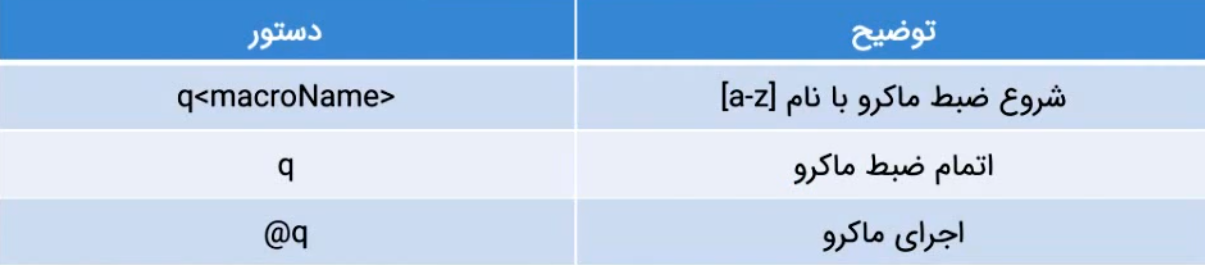
جستجو و جایگزینی چند کلمه به طور همزمان در کل فایل : می توان از :%s/old1\|old2\|old3/new/g تمام مقادیر قدیمی را با مقدار جدید در کل فایل جایگزین می کند

هایلایت کردن تمام کلمات جستجو شده : اگر قبل از جستجو :set hlsearch را زده باشیم تمام موارد جستجو شده را هایلات نشان می دهد و برای پاک کردن هایلات نیز کافی است که :noh را بزنیم.

جستجوی کلمه ای که کرسر روی آن است: کافی است از \* استفاده کنیم به این ترتیب تمام کلمات مشابه در کل فایل هایلایت می شوند و با زدن مجدد \* مدام به کلمه بعدی می رویم. برای رفتن به کلمات قبلی باید از # استفاده کنیم.

استفاده از رگولار اکپرشن در جستجو: می توان دقیقا پترن مد نظر را با /pat در فایل جستجو کرد.

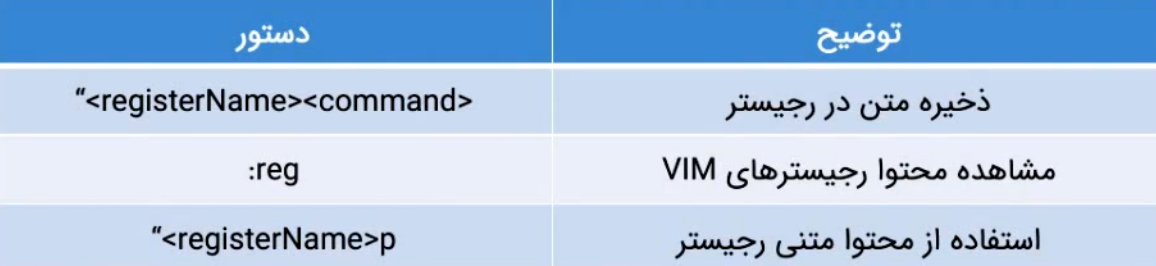
## ماکرو و رجیستر



می توانیم یک سری دستور را به عنوان ماکرو ذخیره کرد و بعدا با زدن یک دکمه تمام موارد اجرا شود.

تولید ماکرو : با زدن q و بعد یک حرف یک ماکرو به نام آن ماکرو ایجاد می شود و مجددا با زدن q آن ماکرو متوقف می شود.

اجرای ماکرو : با استفاده از @q ماکروی ذخیره شده اجرا می شود



رجیستر در واقع یک کلیپ بورد جدید است به جای کلیپ بورد استاندارد

ذخیره محتوی در یک رجیستر: با زدن “ry2y یک رجیستر به نام r ساخته می شود که از جایی که کرسر وجود دارد تا دو خط در رجیستر ذخیره می شود

بازیابی محتوی رجیستر : با زدن “rp که نام رجیستر r است محتوی موجود در رجیستر paste می شود.