

编辑器vim (二)

主要内容

- 插入模式
- 正常模式：
 - 光标移动
 - 行内搜索和行间搜索
 - 文本对象移动
 - 编辑命令
 - 文本对象选择
 - 复制和粘贴
- 可视模式

插入模式

- 原样粘贴paste:
 - 选项"paste": 暂时关闭自动缩进, expandtab等, 在状态栏中会显示-- INSERT (paste)---
 - 将系统剪贴板中来自于其他应用的内容粘贴到vi窗口时, 希望原样paste时设置
 - 粘贴完后 set nopaste关闭paste选项
- 选项'pastetoggle/pt': 插入模式和正常模式下通过指定的键切换paste选项

set pastetoggle=<F10>

- 插入指定寄存器的内容 **<Ctrl-R> <寄存器名>**,
 - 键入Ctrl-R后会显示", 表示等待输入寄存器名字, 可输入{0-9a-z"%#*+:-=}
 - Ctrl-A: 插入最近插入的内容, 相当于<Ctrl-R>.
- <Up> <Down> <Left> <Right> <Home> <End> 光标键, 移动位置且仍在插入模式
- Ctrl-O<cmd> 执行一个命令后返回到插入模式

```
inoremap <silent> <C-S> <C-O>:update<CR>  
noremap <silent> <C-S> :update<CR>
```

```
.bashrc中增加  
stty -ixon -ixoff
```

插入模式：自动完成

```
netrc.py [+]  
-- ^X mode (^] ^D ^E ^F ^I ^K ^L ^N ^O ^P s ^U ^V ^Y)
```

- 支持**单词的自动完成** <Ctrl-P>/<Ctrl-N>
 - 查找前一个或下一个以目前光标所在位置的单词前缀开始的单词，如果只有一个匹配，自动完成
 - 如果有多个选择时，会弹出选择框
 - 通过箭头或者<Ctrl-N>或<Ctrl-P>从多个可选项中选择
 - **键入其他键表示接受选择**，同时新键入的键也会插入
 - **<Ctrl-E>放弃选择，不进行自动完成** (end)
 - <Ctrl-Y>接受选择(yes)
 - complete选项控制查找的范围，缺省为当前文件以及其他打开的文件、tag文件和包含文件等
- <Ctrl-X>进入子模式来进行自动完成
 - 输入非法的Ctrl-X子模式命令(比如Ctrl-C，表示中止命令)时退出子模式
 - 在自动完成的过程中，箭头及<Ctrl-N>或<Ctrl-P>从多个可选项中选择，输入<Ctrl-E>放弃选择
 - **<Ctrl-X><Ctrl-L>：整行**
 - <Ctrl-X> <Ctrl-F>：文件名
 - <Ctrl-X> <Ctrl-N>或<Ctrl-X><Ctrl-P> 当前文件的单词
 - <Ctrl-X> <Ctrl-K> 从字典文件中查找，首先设置字典文件 :set dictionary+=/usr/share/dict/words
 - <Ctrl-X> <Ctrl-]> 从标签(tags)文件中查找
 - **<Ctrl-X> <Ctrl-O> 万能补全(omnicomplete)**，即根据当前光标部分单词的内容来猜测进行补全
 - 弹出一个预览preview窗口显示相关信息

正常模式：左右上下移动光标

set wrap
set linebreak

- 行回绕选项wrap: 某行长度超过屏幕宽度时多行显示, 这些行称为display line
- 选项linebreak/lbr: 在某行太长时根据breakat设置多行显示
- g0/g\$/g^ 移动光标到显示行的相应位置
- gj/gk 与jk类似, 区别在于有显示行时, 为下一个和上一个显示行



```
3 time.localtime()
4
5 text = ' Other codes may be available on your platform.  See documentation for
  the C library strftime function.  Other codes may be available on your platform.
  See documentation for the C library strftime function.'
6
7 print(text)
8
```

键0	动作
j或下方向键	下移一行
k或上方向键	上移一行
h或左方向键	左移一位
l或右方向键	右移一位
0或<Home>	当前行开头
\$或<End>	当前行末尾
^或下划线_	当前行第一个非空格类字符
g_	当前行最后一个非空格类字符
<Return>或+	下一行第一个非空字符
-	上一行第一个非空字符
<Space>	下一个字符, 缺省允许跨越行
<BS>	上一个字符, 缺省允许跨越行

whichwrap选项: 左右移动光标键是否允许跨越行, 缺省为'b,s', 仅允许<BS><Space>

正常模式： 跳转到指定行和滚屏(scrolling)

- 选项number: 开启行号显示
- 选项ruler: 状态栏中包含当前行和列
- 选项relativenumber/rnu: 开启相对行号，方便使用jk快速跳转

键	动作
Ctrl-G	查看当前光标所在位置等信息
nG或ngg或:n	文件第n行，最后一行可:\$
gg/G	文件第1行/最后一行
n%	光标移动到文件的n%位置
n<Motion>	表示执行n次光标移动命令，比如nj表示向下移动n行
	pipe字符跳转到指定列

- jk等光标移动命令移动光标要超出窗口可显示的行时会自动滚动
- 选项scrolloff/so: 光标所在行离窗口顶部或底部至少多少行，缺省为0，可以设置为2

键	动作
<C-F>	PageDown下翻一页
<C-B>	PageUp上翻一页
<C-D>	向下翻半页
<C-U>	向上翻半页
<C-E>	向下翻一行(extra)
<C-Y>	向上翻一行
H	光标移动到屏幕顶部(high)对应的行
M	光标移动到屏幕中间(middle)对应的行
L	光标移动到屏幕底部(low)对应的行
zt	滚屏使得当前行到屏幕顶部
zz	滚屏使得当前行到屏幕中间
zb	滚屏使得当前行到屏幕底部

```
set scrolloff=2
" set relativenumber
```

```
22
23 inoremap <silent> <C-S> <C-O>:update<CR>
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540

```

移动光标：行内搜索

行内搜索：即在光标所在的行范围内搜索某个字符

- 正向还是反向搜索
- 如果找到后光标是停在查找的字符上，还是该字符的所在搜寻方向上的前一个位置

键	动作
fc	行内向右(正向) 搜索字符c
Fc	行内向左(反向)搜索字符c
tc	行内向右(正向)搜索字符c， 光标在字符左边(till)
Tc	行内向左(反向)搜索字符c， 光标在字符右边
分号;	继续上次fFtT行内搜索命令
逗号,	继续上次fFtT行内搜索命令， 但是是之前的反方向

```
if (answer=='yes' or answer=='YES'):
    print('yes')
else:
    print('no')
```

f'	光标在'y e s'的第一个字符
;	重复上次f',光标在'y e s'
;	再次重复， 光标在'y E S'
,	重复最近的f',只是反方向, 光标在'y e s'
;	重复最近的f',光标在'y E S'
t'	光标在'y E S' s '

移动光标： 行间搜索

- /和?搜索命令进入命令行模式， 键入/或?切换到命令行模式等待用户输入模式， /和?表示正向(往右往下)和反向(往左往上)搜索指定的模式， 该模式是一个正则表达式
- n 重复最近搜索命令， 正向还是反向是由最近的搜索命令决定。N重复最近搜索命令， 但是是该搜索命令的反方向
- vim记忆行间搜索之前使用的模式， 可通过箭头和<C-N><C-P>浏览

```
import time
print(time.localtime())
print(time.time()) #TIME
```

/time<CR>	第1行
n	第2行第1个
n	第2行第2个
n	第3行第1个
N	第2行第2个
?<CR>	第2行第1个
n	第1行第1个

/pattern[/]<Return>	正向搜索文本正则表达式pattern
?pattern[?]<Return>	反向搜索文本正则表达式pattern
/<Return>	正向搜索， 采用的模式为最近搜索的模式
? <Return>	反向搜索， 采用的模式为最近搜索的模式
n	继续最近的搜索命令(/或?或*或g*等)
N	继续最近的搜索命令(/或?或*或g*等), 只是为反方向
选项wrapscan/ws	到文件两端时回绕
选项ignorecase	搜索时大小写无关
选项smartcase	搜索模式中有大写， 则大小写相关， 否则大小写无关

- set ignorecase smartcase
- /time 大小写无关 /TIME 仅仅搜索TIME

```
set wrapscan
set ignorecase
set smartcase
```


移动光标： 行间搜索

- * 和# 正向(反向) 搜索光标所在或最近的下一个word，而且搜索找到的内容要求也是一个完整的word(字母数字和下划线组成)
 - 等价于\<word\>，会保存在搜索历史记录中
 - 比如要查找word， word1以及1word都不匹配， 而word.或'word'中的word匹配
- n和N: 重复的搜索命令也包括了搜索光标所在单词的命令给* # g* g#
- 如果hlsearch选项开启， 则匹配的内容以高亮显示。可通过:noh[lsearch]清除高亮显示
- 可映射<BS>增加nohl功能

```
import time
print(time.localtime())
print(time.time()) # CTIME

text = ' Other codes may be available on your platform
library strftime function. Other codes may be available
documentation for the C library strftime function.'

print(text)

if (answer=='yes' or answer=='YES'):
    print('yes')
else:
    print('no')

~
~
~
~
test.py [+]
/time
```

*	向前搜索当前光标所在整个word，即\<word\>
#	与*类似，只是向后(反向)搜索
g* g#	与* #类似，只是搜索只要包含word即可 /word或?word
选项hlsearch	开启高亮搜索选项
:noh[lsearch]	清除高亮显示
选项incsearch/is	在输入模式的过程中就开始搜索

```
set incsearch
set hlsearch
nnoremap <silent> <BS> :noh<CR><BS>
```

移动光标: 单词间移动

e.g. we're going too slow ... another ...4,

WORD:

- 相比word包括的范围更大
- 一系列非空格类字符组成
- 分隔符: 连续的空格类字符

e.g. we're going too slow ... another ...4, 分解成WORD

e.g.	we're	going	too	slow	...	another	...4,
------	-------	-------	-----	------	-----	---------	-------

e.g. we're going too slow ... another ...4, 分解成word

e	.	g	.	we	'	re	going	too	slow	...	another	...	4	,
---	---	---	---	----	---	----	-------	-----	------	-----	---------	-----	---	---

word:

- iskeyword选项给出了单词类字符的定义, 缺省包括字母数字和下划线
- 由一系列单词类字符(字母数字和下划线)组成, 或者一系列非单词类字符即标点符号) 组成
- 分隔符: 一系列的空格类字符或者一系列非单词类字符(标点符号)
- 标点符号属于某个word

键	动作	键	动作
w	下一个单词的开始	W	同w, 只是空格类字符确定单词
e	右边单词, 即当前单词或下一个单词(当前光标在空格处或单词结尾)的结尾	E	同e, 只是空格类字符确定单词
b	左边单词, 即当前单词或上一个单词(当前光标在空格处或单词开始)的开始	B	同B, 只是空格类字符确定单词
ge	上一个单词的结尾	gE	上一个单词的结尾, 只是WORD定义

移动光标: 文本对象间移动

- vi还定义了句子(sentence), 段落(paragraph)、 section(章节) 的概念
- sentence(句子)一般以英文的句号、感叹号和问号(!.?)结尾, 后面跟一个空格类(换行,空格或Tab)字符
- **paragraph(段落) 之间一般以空行(注意不是包含空格的行)隔开, 对应语句块**
- section(章节) 常用于找到(最外层)函数的开始和结束位置

键	动作
)	下一个句子的开始(不含)的前一位置
(上一个句子的开始的前一位置
}	下一个段落的开始(不含)的前一位置
{	上一个段落的开始的前一位置
]]	下一个章节的开始位置
][下一个章节的结束位置
[[上一个章节的开始位置
[]	上一个章节的结束位置
%	包含了光标所在位置字符的最里面的括号的位置之间跳转

章节间移动记忆:

- 第一个字符为:
 -]: 下一章节
 - [: 上一章节
- 第二个字符:
 - 前一字符一致: 开始位置
 - 不一致: 结束位置

移动光标： 标记(help mark-motions)

- 标记(mark)指记录的文件位置，标记名可以是：
 - 小写英文字母：使用范围为当前文件
 - 大写英文字母：也称为文件标记，表示某个文件的位置
 - 十进制数字0-9: 用户不能设置，由vim自身维护，0记录了上次退出vim时光标所在的哪个文件以及位置，1记录了上上次，如此…
- 特殊的标点符号标记：使用范围为当前文件

标点mark	含义
< >	可视选择区域的开始和结束位置
"	当前文件上次退出位置
^	上次退出插入模式位置
.(dot)	上次修改位置
`反引号 或'单引号	大范围跳转时之前的位置，如行间跳转G及行间搜索(/?)等

```
autocmd BufReadPost * if line("'\"") >
0 && line("'\"") <= line("$") |
    \ execute "normal! g`\"" |
    \ endif
```

命令	含义
:marks	查看当前活跃标记
ma	标记当前位置为a，可取值为{a-zA-Z}以及标点标记名
`(反引号)a	跳转到前面标记的位置a
'(单引号)a	跳转到前面标记的位置a的那一行
``(双反引号)	跳转到大范围跳转时之前的位置
"(双单引号)	与``类似，只是跳转到对应位置的那一行
gi	跳到上次插入结束位置(标记^) 开始插入

在实践中用得最多是通过`或"在当前位置与上一个跳转位置间切换

主要内容

- 插入模式
- 正常模式：
 - 光标移动
 - 行内搜索和行间搜索
 - 文本对象移动
 - **编辑命令**
 - 文本对象选择
 - 复制和粘贴
- 可视模式

插入文本

- 在正常模式下通过键入ilaAoOcCsS等命令进入插入模式

键	动作
i	当前位置前插入
I	当前行首第一个非空白字符处插入，等价于^i
a	当前位置后插入
A	当前行尾插入，等价于\$a
o	当前行下面插入
O	当前行上方插入

- 正常模式的命令一般在前面可包含一个整数n，**[#1] command**，表示重复该命令n次
 - 比如4j表示光标向下移动4行(可开启相对行号relativenumber/rnu)
 - 比如4ohello<ESC>，会插入4行hello

删除文本

[#1] operation [#2] target

#1: 表示重复多少次operation, 缺省为1

#2: 表示一次operation对应的target有多少个

- 删除单个字符，可以使用x或X命令
- 删除多个字符，可以使用d命令
- d/c/y等命令属于操作符(operator)命令，操作符+ target 组合在一起才是完整的命令
 - 选项showcmd开启，可在状态栏下方看到正在输入的命令
 - 在正常模式下键入操作符进入到操作符等待(Operator-pending)模式
 - 可输入光标移动(motion)以及文本对象命令，此时执行操作符命令，并回到正常模式
 - 可输入<ESC>取消并返回到正常模式
- dw从当前光标删除到下一个单词开始
- db 从当前光标删除到左边单词开始
- de 从当前光标删除到右边单词结束
- dj 删除当前行和下一行
- 4x或d4l: 删除4个字符
- d4w或4dw: 删除4个word
- dt": 删除从当前位置直到"(但不包括) 之前的所有字符。如果df", 则"也会被删除
- 10dd: 删除10行
- 4J: 将当前行开始接下来4行合并为同一行

键	动作
x	删除当前位置处的字符，等价于dl
X	删除当前位置左边的字符，等价于dh
d{motion}	删除从当前位置到motion所给位置的字符，motion为前面介绍的光标移动命令
d{textobject}	删除指定文本对象
D	删除当前位置到行尾的字符，等价与d\$
dd	删除当前行
J	将多行(至少两行)合并为一行，相当于删除换行符。如果要一行分成多行，则在指定位置插入或者替换为换行符(即r<CR>)即可

修改和替换文本

```
1
13 if (answer=='yes' or answer=='YES'):
1     print('yes')
2 else:
3     print('no')
```

- 替换(replace)单个字符:
 - rc命令表示当前光标所在位置的字符替换为c，仍然在正常模式
 - s(substitute) 命令等价于xi，将当前光标所在位置的字符删除，进入插入模式，直到<ESC>退出，相当于新输入的内容替换原来的字符
- 修改(change)多个字符:
 - c{motion}命令，相当于d{motion}i，进入插入模式。
 - C表示c\$
 - cc表示替换当前行，S和cc等价，也是替换当前行
- R命令进入**替换(Replace)模式**，会从光标位置开始逐个替换原来的字符

c0: 更改行首到当前位置的内容
注意光标所在位置为非空白字符时
cw/cW等价于ce和cE，不会删除之后的空格
c5w:更改5个word的内容

- ✓ 原有位置的字符被删除，替代以新输入的字符
- ✓ 如果到了行尾，由于没有字符可以删除，因此新输入的字符被附加到行尾
- ✓ 输入<ESC>退出替换模式，返回到正常模式

键	动作
rc	替换当前位置字符为c，还在正常模式。 r<CR> 分割行
s	替换当前位置字符，进入插入模式，等价于xi
c{motion}	替换当前位置到motion所指定位置处的内容，相当于d{motion}i
C	替换当前位置到行尾的字符，进入插入模式，相当于c\$
S或cc	替换当前行。注意S与D/C的不同， D表示d\$, C表示c\$
R	进入replace模式，逐个替换后面字符直到ESC

格式化

- 缩进命令主要包括 < >, 每次缩进的宽度来自于shiftwidth选项
- 改变大小写命令: ~/g~/gu{motion}/gU{motion}等
- 格式化命令:
 - = {motion}, 依赖于外部程序equalprg或选项indentexpr的设置实现
 - gq {motion}, 一般依赖于选项formatexpr或外部程序formatprg实现
- <C-A> 或 <C-X>
 - 光标所在处或之后的数值或字母增加1或减少1
 - 选项nrformats如果包含了alpha, 则允许字母形式, set nrformats+=alpha

filetype plugin on

键	动作
<< >>	当前行往左或往右缩进shiftwidth个空格
<{motion} >{motion}	选定的行往左或往右缩进shiftwidth个空格
= = 或={motion}	当前行或者选定的行进行格式化
<C-A>或<C-X>	光标所在处或之后的数值或字母增加1或减少1

1. xxx
1. xxx

文本编辑命令撤销和重复

- .(dot) 表示重复最近的编辑(插入、删除、更改等)命令
- u可撤销最近的插入、删除、更改等编辑操作
- U表示撤销从移动到当前行开始后所有对当前行的修改
- Ctrl-R或:redo, 相当于undo undo, 即重新执行撤销的命令

注意： 传统vi中仅允许撤销一级:

- u撤销修改, 再次u撤销刚才的撤销修改, 即相当于redo

help undo-tree

- vim对于之前的编辑和撤销操作以树的方式维护, 提供了非常强大的回到之前的某个状态的功能
- :undolist 查看undotree的叶子节点
- 有相关的vim插件提供了undotree的浏览和切换功能, 比如Gundo和undotree

键	动作
u	撤销上一命令对编辑缓冲区的修改
U	撤销移动到当前行开始后对当前行的修改
Ctrl-R或:redo	redo,撤销最近的撤销 (重新执行撤销的命令)
.	重复上一编辑命令
/<CR>	向前重复搜索相应的模式 /text...
? <CR>	向后方向重复搜索相应的模式
n	重复上次搜索
N	反向重复上次搜索
;	重复行内搜索f,F或t,T
,	相反方向重复行内搜索

文本对象选择(help text-objects)

- Vim还支持文本对象选择命令，这些命令可以出现在操作符(比如d)之后，也可在可视(visual)模式中使用
- 对于word/WORD等有两种文本选择方式：
 - 两者的区别在于如果该文本对象的周围有空格时选择的范围不同
 - “i”命令(如iw或iW)表示“inner word”或“inner WORD”
 - 如果光标在word/WORD范围内，则iw包含该word/WORD
 - 如果光标在word/WORD前面的空格，则iw包含了前面的所有空格
 - “a”命令(如aw或aW)表示“a word”或“a WORD”
 - 如果光标在word/WORD(WORD以空格隔开) 范围内，则aw/aW包含了该word/WORD以及后面的空格(如果有)。如果后面没有空格，则包含前面空格(如果有)
 - 如果光标在word/WORD前面的空格，则aw/aW包含了word/WORD前面的所有空格以及word/WORD

"inner or around word?"

空格	word	空格
	aw/aW	
aw/aW		
	iw/iW	
iw/iW		

- vi定义了句子(sentence), 段落(paragraph)的概念
 - 可用as或is选择句子以及是否包含周围空格
 - 可用ap和ip选择段落以及是否包含周围空格

文本对象选择(help text-objects)

- vi还可对于文本对象块进行选择
 - 配对的括号: [] () {} < >等括号
 - 配对的引号: 单引号、双引号和反引号
 - 配对的tag block: <aaa> ... </aaa>
- 对于文本对象块有两种文本选择方式:
 - “a”命令表示“a block”，选择的文本除了相应的分界符包含的文本外还包含了前后的分界符本身
 - “i”命令表示“inner block”，选择的文本仅包含文本，不包含前后的分界符

键	动作
a[或a]	从光标所在位置开始反向查找未闭合的[, 然后正向查找配对的], 选择所包含的文本块以及[]
i[或i]	与a[或a]类似, 只是不包括外围的[]
a(a) i(i) a{ a} i{ i} a< a> i< i> at it	与上面介绍的a[i[类似 圆括号包含的block也可采用ab/ib, 大括号包含的block也可采用aB/iB at/it表示tag block
a' a" a`	选择相应引号包含的字符串, 包括外围的引号。如果最后的引号后面有空格, 则包含了后面的那些空格; 如果最后引号后面没有空格, 则包含开始引号前面的空格(如果有)
i' i" i`	选择相应引号包含的字符串, 不包括外围的引号

主要内容

- 插入模式
- 正常模式：
 - 光标移动
 - 行内搜索和行间搜索
 - 文本对象移动
 - 编辑命令
 - 文本对象选择
 - **复制和粘贴**
- 可视模式

复制和粘贴(yank and paste)

- 要进行复制和粘贴，首先选择要复制的内容到寄存器(y命令)；然后移动到新的位置，将寄存器的内容粘贴(p或P命令)
- Vi支持10种类型的寄存器，可通过:reg(isters)显示当前寄存器的内容
- "加上寄存器名表示使用哪个寄存器，**复制、粘贴和删除类命令**(dcsx)缺省使用**无名寄存器**" (即上次使用的寄存器内容)
 - "ayw表示从当前位置到下一个word开始的内容复制到寄存器a中
 - "b4dd表示删除4行并且保存到寄存器b中
 - "ap表示将寄存器a的内容粘贴插入到当前位置之后
- 寄存器的内容如果以字符为单位(如yiw)，则p在当前位置之后粘贴，P表示在之前
- 寄存器的内容如果以行为单位(如y4j或4yy)，则p表示当前行之后，而P表示当前行之前粘贴
- 在插入模式以及命令行模式中可通过<Ctrl-R>(会等待输入寄存器名字)插入相应寄存器的内容

键	动作
Y或yy	复制(yank)一行到寄存器
y{motion text}	复制当前位置到motion指定的内容或者文本对象到寄存器
小写的p	将寄存器中的内容粘贴插入到当前位置或当前行之后
大写的P	将寄存器中的内容粘贴插入到当前位置或当前行之前

复制和粘贴: 寄存器

名字	动作
a-z A-Z	命名寄存器，在执行删除复制粘贴命令时使用“a形式显式指定。大写表示附加到原有内容
"	无名寄存器，保存复制和删除(不管是否指定了寄存器，黑洞除外)的内容，类似于 最近使用寄存器
0-9	寄存器0保存 不指定寄存器的y命令 复制的内容；寄存器1-9保存最近不指定寄存器的删除类(dcs)命令删除的 多行内容 。寄存器1为最近删除的，2为之前删除的，如此...
-	小删除寄存器， 不指定寄存器的删除命令 (dcsx) 删除内容以字符为单位时 放在-中
. : %	只读。“ .保存最近插入的文本 ”，“ :保存最近执行的命令行 ”，“ %保存文件名 ”
#	alternate buffer register，保存替代文件名，即最近编辑过的文件的文件名
/	搜索模式寄存器，保存最近 搜索时使用的模式
=	表达式寄存器，保存要计算的表达式，读该寄存器时返回运算后的结果
_	黑洞寄存器，这样不会污染其他寄存器(包括无名寄存器)
*+~	GUI的粘贴板， macOS/Windows中*对应系统粘贴板，Linux系统中*为高亮选择内容, +为系统粘贴板

- “最近一次复制或删除的内容，可以通过p/P命令粘贴（不需要指定寄存器）
- 0: 最近不指定寄存器的y复制的内容
- 1-9: 不指定寄存器时多行删除的内容
- -: 不指定寄存器时删除内容以字符为单位的内容
- . 插入的内容
- 执行cs等命令时，原来的内容(如果不指定寄存器) 在-或者1，而新的内容在.

选项clipboard决定不指定寄存器时使用哪个寄存器

```
set clipboard=unnamed  
set clipboard=unnamedplus
```


复制和黏贴: 寄存器

```
:reg
--- 寄存器 ---
"" insert some texts.....
"0 print('yes')^J
"1 insert some texts.....^J
"2 insert something^J
"3 if (answer=='yes' or answer=='YES'):^J  pprint('yes')^J
"4 ^J
"5 "word1"... word2... ..word3  word4 )^J^Jif (answer=='yes' or answer=='Y
"6 set-window-option -g window-status-attr dim^J
"7 set-window-option -g window-status-bg default^J
"8 set-window-option -g window-status-fg cyan^J
"9 set-window-option -g window-status-current-fg white^Jset-window-option -g
"a yes
"- insert some texts.....
"." print('hello world')
": w
"% vifile.txt
"/ ^#
"= 4*5*6
请按 ENTER 或其它命令继续
```

最近复制或删除的内容

最近不指定寄存器的y命令复制的内容

不指定寄存器近几次多行删除的内容

最近插入的内容

不指定寄存器的行内删除的内容

常用的vi组合命令

键	动作
xp	调换两个字符
ddp	调换两行

可视模式 :help visual-mode

- 通过高亮选择一块文本后，键入相应的编辑和复制(d/c/y等)命令，对于文本进行相应的操作，退出可视模式
- 选择文本可以有三种方式：字符方式、行方式、块方式
- 在正常模式下通过v/V或<Ctrl-V>开始进入可视模式，当前光标位置为选择文本的开始位置
 - v进入字符方式的可视模式
 - V进入行方式的可视模式，选择文本时以行为单位
 - <Ctrl-V>进入块方式的可视模式，以块(也称为列)方式选择文本
 - 在可视模式下，通过v/V或<C-V>可以改变选择字符的方式(比如从v变为V)。如果键入命令与当前模式相同，则退出可视模式
 - 按<ESC>或<Ctrl-C> 退出可视模式
 - gv进入可视模式，选择的为最近进入可视模式所选择的文本，选择方式也和上次一样
 - gn或gN命令，与正常模式下的搜索命令类似，重复上次搜索或反方向重复上次搜索，在找到匹配时进入可视模式并选择搜索到的内容
 - 在可视模式下，gn或gN命令会将选择的内容扩展到这次搜索所匹配的内容

可视模式

o Go to Other end of highlighted text: The current cursor position becomes the start of the highlighted text and the cursor is moved to the other end of the highlighted text. The highlighted area remains the same. v_o

Go to Other end of highlighted text: The current cursor position becomes the start of the highlighted

v_o
Other end of highlighted text: position becomes the start of and the cursor is moved to the highlighted text. The highlighted area

进入可视模式后

- 通过正常模式下的光标移动和文本对象命令(如iw/ib等)选择文本
- o命令切换选择文本的方向，光标当前位置成为选择文本的开始位置，光标移动到选择文本的另一端
- 块选择方式时，O命令使得光标移动到同一行的另一边，o命令切换到选择块的另一边（即对角线方向），即o为对角切换，而O为横向切换
- 标记`<`>`分别记录了可视模式选择的文本的开始和结束位置
- 键入相应的编辑和复制命令，对于目前高亮选择的文本进行相应的操作，退出可视模式
 - d(delete)或者x: 删除选择的内容
 - c(change)或s: 删除选择的内容并进入插入模式
 - y: 复制选择的内容
 - p或P(paste): 替换所选择的内容
 - 缩进相关命令: >(往右缩进) <(往左缩进) 格式化(=或gq)
 - J(join) 选择的行合并为一行
 - :或!, 进入命令行模式, 前面添加 '<,>', 表示对于选择的文本执行后面的Ex命令

可视模式: blockwise visual mode

- Ctrl-V进入块方式的可视模式
- o以及O切换选择文本的对角和横向方向
- l{string}<ESC> 选择的文本块的第一列之前都插入string
- A{string}<ESC>选择的文本块的最后一列之后插入string; 如果文本块是通过<Ctrl-V>\$方式选择的, 则在文本块的行尾附加string
- c{string}<ESC> 选择的文本块部分的内容被多行的string替代

Other end of highlighted text:
position becomes the start of
nd the cursor is moved to the
ghted text. The highlighted at

v_o

v_b_A

append {string} to the end of
differing behavior where the
lengths:

键	动作
v	进入字符方式可视模式
V	进入行方式可视模式
Ctrl-V	进入块方式可视模式
gv	进入可视模式, 选择上次可视模式的内容
gn	搜索最近使用的模式, 选择或扩充选择到找到的内容
gN	与gn类似, 只是反方向搜索
ESC或<Ctrl-C>	在可视模式下键入时退出可视模式
o或O	交换可视模式下选择的文本的开始和结束位置
`<`>	该标记分别记录了最近可视模式选择的开始和结束位置