本文档暂不考虑设置问题

https://vimhelp.org/usr_toc.txt.html

移动命令

最常用

动作	键	备注
向上	k	
向下	j	
向左	h	
向右	l	
光标移动到行首	0	
光标移动到一行的第一个非空字符	^	
光标移动到行尾	$	
下个单词首字符	w	
下个广义单词首字符	W	
单词尾字符	e	
广义单词尾字符	E	
上一个单词首字符	b	
上一个广义单词首字符	B	
到一行内光标后一个为x的字符	fx	
到一行内光标前一个为x的字符	Fx	
到一行内光标后一个为x的字符前的位置	t	
到一行内光标前一个为x的字符后的位置	T	
可以在括号两边之间跳	%	如果不在一个有用的字符上, %会先正向查找找到一个
查找下一个光标所在的单词	*	
查找上一个光标所在的单词	#	
非空格符的下一行	+	Vscode里暂时用不了
非空格符的上一行	-	Vscode里暂时用不了

*广义单词指被空格分隔的文本，如两个空格间的一段文字

*cw的功能实际上与ce相同这是一个历史遗留的习惯问题

行跳转

动作	键	备注
跳到当前文件第n行	nG	
光标移动到当前文件的首行	gg / 1G	
光标移动到当前文件的末尾	G	
按比例跳转（文档xx%处）	xx%	
跳到当前文件第n行	:n	
跳到当前文件第1行	:1 / :0	
光标移动到当前文件的末尾	:$	
选中几行	:xx,xx	

看着不常用的跳转

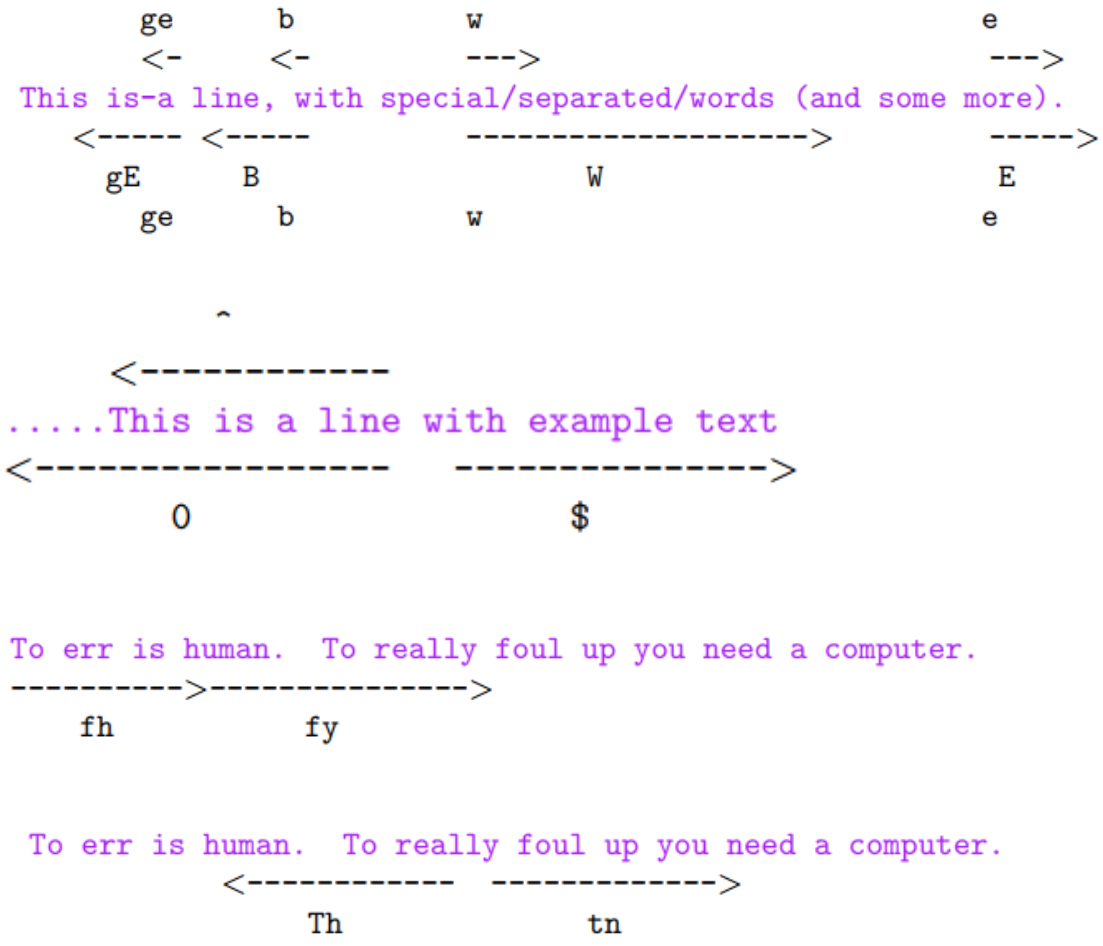
动作	键	备注
向下移一屏	Ctrl+f	在Vscode中被替换了
向上移一屏	Ctrl+b	
向下移半屏	Ctrl+d	
向上移半屏	Ctrl+u	
跳到当前屏最上方一行第一个字母	H	
跳到当前屏中间一行第一个字母	M	
跳到当前屏最下方一行第一个字母	L	

调整光标所在行的位置

不移动光标所在行而调整光标显示在屏幕上的位置

调整光标所在行的位置	键	备注
顶部	zt	
中部	zz	
底部	zb	

部分指令图例


ge b w e
This is-a line, with special/separated/words (and some more).
gE B W E
ge b w e
.....This is a line with example text
0 $
To err is human. To really foul up you need a computer.
fh fy
To err is human. To really foul up you need a computer.
Th tn

操作命令

键		备注
o	从当前光标下面创建新的一行	A, a, I, i暂略
O	从当前光标上面创建新的一行	
x	删光标后字符	
X	删光标前字符	
d	删除	
D	d$	
dd	删除整行	
y	复制	
yy	复制整行	
p	粘贴	
v	可视模式	
V	整行选择	
CR-v	块选择模式	
c	删除并进插入模式	
C	删至行末并进插入模式	
cc	删整行并进插入模式	
r	单个替换	
R	连续替换	
s	删除光标盖住的字符	Vscode里暂时用不了
S	删除光标所在行	Vscode里暂时用不了
J	将光标所在行与下一行的数据结合成同一行	会自动加空格
.	重复上一次操作	只是移动操作不会重复， 宏操作在vscode不会重复， u，Ctrl-r，冒号命令也不会被重读
co	多行复制	:1,10 co 20 将1-10行插入到第20行之后 :1,$ co $ 将整个文件复制一份并添加到文件尾部
m	多行移动	1, 10 m 20 将第1-10行移动到第20行之后
u	复原上一个操作	
U	撤销所有操作	?

键		备注
CR-r	撤销复原	

文本对象

不同于操作符-动作与可视模式，**操作符 - 文本对象**是执行修改的第三方法

这种方法与 "操作符 - 动作" 很相似，但它不是操作于从当前位置到移动目标间的内容，而是对光标所在位置的 "文本对象" 进行操作。文本对象是作为一个整体来处理的。现在光标在对象中的位置无关紧要。

文本对象操作涉及到范围和操作，范围主要是主要是 i(inner) 和 a(around)，文本对象有 w (word), s (sentence), p (paragraph), 和各种 引号和括号。当然通过插件还能增强 text object 的功能

主要新字符是i， a（inner，a）前者不包括文字后面的空格，而后者会包括。a会倾向处理整体，i会倾向处理对象内

范围	键	备注
单词	w	
广义单词	W	
句子	s	
段落	p	
括号	({[ ]})	写出括号的哪一边都行
引号	'"`	单引号，双引号，反引号

*广义单词指被空格分隔的文本，如两个空格间的一段文字

以下实例中的红色 [ ] 表示作用范围：

```
iw        This is a [test] sentence.
aw        This is a [test ]sentence.
iW        This is a [...test...] sentence.
aW        This is a [...test... ]sentence.
is        ...sentence. [This is a sentence.] This...
as        ...sentence. [This is a sentence. ]This...
          End of previous paragraph.

ip        [This is a paragraph. It has two sentences.]

          The next.
          End of previous paragraph.

ap        [This is a paragraph. It has two sentences.

          ]The next.
i( or i)  1 * ([2 + 3])
a( or a)  1 * [(2 + 3)]
i< or i>  The <[tag]>
a< or i>  The [<tag>]
i{ or i}  some {[ code block ]}
a{ or a}  some [{ code block }]
i[ or i]  some [[ code block ]]
a[ or a]  some [[ code block ]]
i"        The "[best]"
a"        The[ "best"]
i`        The `[best]`
a`        The[ `best`]
```

如果安了这个插件Indent Object就能支持把缩进视为文本对象了

替换

用v选中再p可以大段替换

normal模式的替换操作规则

- 基本格式：`:s/old/new/` 这个指令代表替代当前行第一个匹配的old。替换是可以使用正则表达式的
- 在s前面加%： 相当于选中所有行进行操作

- 在s前面加:n,m： 对选中的行进行操作（:1,$s/old/new/和:%s/old/new/一样）
- 在最后的/后面加c： 在每个旧字符替换时进行确认
- 在最后的/后面加g： 替换范围内所有的旧字符

	作用	备注
:s/old/new/	用old替换new，替换当前行的第一个匹配	
:s/old/new/g	用old替换new，替换当前行的所有匹配	
:%s/old/new/	用old替换new，替换所有行的第一个匹配	
:%s/old/new/g	用old替换new，替换整个文件的所有匹配	
:%s/old/new/gc	用old替换new，替换整个文件的所有匹配，并在每个单词替换时进行提示	
:10,20 s/^/	在第10行知第20行每行前面加四个空格，用于缩进	

一部分查找相关

命令模式下的查找操作（`/`与`?`应该都是支持正则表达式的）（查找.*[]^%/?~$时注意转义）

键		备注
/xxx	向光标之下查找关键字xxx	
?xxx	:向光标之上查找关键字xxx	
n/N	n重复上一个搜索操作，N反向	
*	查找下一个光标所在的单词	和上面不同，这个本质上属于移动，可以和v,p...配合
#	查找上一个光标所在的单词	同上

区分这个大小写，要设置ignorecase选项

此外这个可以查找以某些字符开头，结尾的单词

键		备注
/\<xxx	搜索xxx开头的单词	
/xxx\>	搜索xxx结尾的单词	

插入模式下的快捷键

键		备注
Ctrl + y	复制并粘贴位于光标上方的文本	
Ctrl + e	复制并粘贴位于光标下方的文本	
Ctrl + p	单词不全	在Vscode中被覆盖了

Jump

这个是可以跨文件的，

一般，每次你执行一个会将光标移动到本行之外的命令，该移动即被称为一个 "跳转"。这包括查找命令 "/" 和 "n" (无论跳转到多远的地方)。但不包括 "fx" 和 "tx" 这些行内查找命令或者 "w" ，"e"，"j"，"k" 等词移动命令。

动作	键	备注
查看跳转表	:jump	
跳回上次跳转发生前的位置	''	``好像也一样（？）
跳回上一个位置	Ctrl+o	
跳到下一个位置	Ctrl+i	

书签

动作	键	备注
创建书签	m{bookmark-name}（Ex : ma）	只能是单个字符，最好是字母吧
跳到书签确切位置	`{bookmark-name}	注意是反引号
跳到书签行的开头	‘{bookmark-name}	正引号
列出所有书签	:marks	
删除书签	:delmarks {bookmark-name}	Vscode下好像用不了
全局书签		书签名是大写

默认书签

默认书签	键	备注
‘	跳转前光标的位置	
"	最后编辑的光标位置	Vscode里用不了
[	最后修改的开始位置	
]	最后修改的结束位置	

寄存器

动作	键	备注
使用寄存器中文本	"<register-name><command>	<command>可以是y系列可以是p系列
查看可用寄存器	:registers	

寄存器类型

寄存器类型	键	备注
未命名寄存器	"	
命名寄存器	a-z / A-Z	共26个，默认情况下VIM不会使用这些寄存器。 在从寄存器中使用内容时，使用大写小写的行为一致。 在向寄存器中添加内容时，如果使用小写字母， 会先清空这个寄存器，然后写入； 如果使用大写字母，则会在原有寄存器上内容后面添加
编号寄存器	0-9	编号寄存器`0`包含最近的`yank`命令中的文本。 编号寄存器`1`包含最近删除或更改命令删除的文本。 其他的呢【？】
默认寄存器	`% # : . "`	

默认寄存器名称	名称	备注
%	当前文件的名称	
#	当前窗口的备用文件的名称	
:	最近执行的命令	
.	包含最后插入的文本	好像有用
"	最后使用过的注册	Vscode里发挥不明显

宏

动作	键	备注
录制宏	q+小写字母	小写字母是宏名称
结束录制宏	q	
播放录制	@{macro-name}	
查看宏	:registers	和查看寄存器那个一样

宏好像用不了`.`，可以数字+@{macro-name}重复执行

Vscode中的操作

简便操作

- d$：D
- 删除整行：dd
- 删除整行并进入编辑：cc
- 复制整行：yy
- 交换光标所在行和其下一行：ddp
- :1,10d 删除1-10行

  :11,$d 删除11行及以后所有的行

  :1,$d 删除所有行

  （：xx,xx）

TODO：

VSCODE中操作

tag操作

搜索的更多操作