

# VimTutor

[VimTutor-note](#) | [Vim on git](#) | [Vim-galore](#) | [Vim](#) | [SpaceVim](#) | [Hack SpaceVim](#)

感觉vimtutor是众多vim帮助文档中最好的入门教程，其中的每一个知识点都有相应的习题，可以让你在短时间内掌握基本的VIM操作。

想要入门vim的童鞋最好还是一边看文档，一边用vim自己尝试每个命令。

请通过vim练习:`vim Vimtutor.txt`

```
=====
=====
```

## Welcome to the VIM Tutor - Version 1.7

### 欢迎使用VIM教程 1.7版

```
=====
=====
```

Vim is a very powerful editor that has many commands, too many to explain in a tutor such as this. This tutor is designed to describe enough of the commands that you will be able to easily use Vim as an all-purpose editor.

VIM是一个强大的编辑器，它拥有大量的命令,不可能在本教程中全部解释。但是，本教程描述了足够多的命令,使你能轻松将VIM作为通用编辑器。

The approximate time required to complete the tutor is 25-30 minutes, depending upon how much time is spent with experimentation.

完成本教程的时间大约是25-30分钟，主要依赖于你在实验中花去的时间。

#### ATTENTION:

The commands in the lessons will modify the text. Make a copy of this file to practise on (if you started "vimtutor" this is already a copy).

## 注意:

本课程中的命令会改变本文的内容，请创建一个副本来练习（如果你用命令“vimtutor”

开始，本文已经是自动生成的副本了）。

注:创建副本的命令为:

```
vim -u NONE -c 'e $VIMRUNTIME/tutor/tutor' -c 'w! vimtutor_copy' -c 'q';ls -l
```

It is important to remember that this tutor is set up to teach by use. That means that you need to execute the commands to learn them properly. If you only read the text, you will forget the commands!

很重要的一点是，要记住，本教程是用于指导VIM的使用。这意味着你需要执行命令以正确地学习它们。如果你只是单纯阅读本文，你会很快忘记这些命令！

Now, make sure that your Shift-Lock key is NOT depressed and press the j key enough times to move the cursor so that Lesson 1.1 completely fills the screen.

现在，请确保你的shift键是启用的，多次按 j 键移动到1.1课,使之居于屏幕中心。

~~

## Lesson 1.1: MOVING THE CURSOR

### 1.1课:移动光标

To move the cursor, press the h,j,k,l keys as indicated.

要移动光标，则按示意按下 h,j,k,l 键

^	
k	Hint: The h key is at the left and moves left.
< h l >	The l key is at the right and moves right.
j	The j key looks like a down arrow.
v	

## 示意:

- 左边的h键是往左移
- 右边的l键是往右移

- j键是往下移

1. Move the cursor around the screen until you are comfortable.

在屏幕中移动光标到你觉得合适的地方。

2. Hold down the down key (j) until it repeats.

Now you know how to move to the next lesson.

按住j键，使之重复作用。

现在你知道了怎么移动到下一节了吧。

3. Using the down key, move to Lesson 1.2.

使用下移键，移动到1.2节。

**NOTE:** If you are ever unsure about something you typed, press to place you in Normal mode. Then retype the command you wanted.

**注意:**如意你不确定你按下了什么，按回到正常模式。再按你想要输入的命令。

**NOTE:** The cursor keys should also work. But using hjkl you will be able to move around much faster, once you get used to it. Really!

**注意:**光标键同样生效。但hjkl键会使你移动得更快，只要你熟悉了它。真的！

~~

## Lesson 1.2: EXITING VIM

### 1.2节:退出VIM

**!! NOTE:** Before executing any of the steps below, read this entire lesson!!

**!! 注:**执行以下步骤前，请读完全节！！

1. Press the key (to make sure you are in Normal mode).

按下键（确保你处于正常模式）。

2. Type: :q! .

This exits the editor, DISCARDING any changes you have made.

按: :q! .

此命令使编辑器退出并放弃你作的作用改动。(即不保存对文本的编辑)

3. When you see the shell prompt, type the command that got you into this tutor. That would be: vimtutor

当你看见shell提示时, 输入以下命令并进入本教程:vimtutor

注:事实上你vimtutor是打开一个副本, 你也可以打开你之前创建的副本:

vim vimtutor\_copy

4. If you have these steps memorized and are confident, execute steps 1 through 3 to exit and re-enter the editor.

如果你已经记住的这些步骤并有信心了, 就执行步骤1-3退出, 然后再进入编辑器。

**NOTE:** :q! discards any changes you made. In a few lessons you will learn how to save the changes to a file.

注:q! 会取消你对文本所作的任何改动。在新的一节你会学到保存文件的修改。

5. Move the cursor down to Lesson 1.3.

移动光标到1.3节。

~~

## Lesson 1.3: TEXT EDITING - DELETION

### 1.3节:文本编辑之删除

**Press x to delete the character under the cursor.**

按下x键以删除光标所在的字符.

1. Move the cursor to the line below marked --->.

移动光标到--->标记的行。

2. To fix the errors, move the cursor until it is on top of the character to be deleted.

为了修复错误, 移动光标到要删除字符上。

3. Press the x key to delete the unwanted character.

按下x键以删除不想要的字符。

4. Repeat steps 2 through 4 until the sentence is correct.

重复步骤2到4，直到语句正确。

---> The ccow jumpedd ovverr thhe mooon.

5. Now that the line is correct, go on to Lesson 1.4.

现在语句正确了吧，请移步1.4节。

NOTE: As you go through this tutor, do not try to memorize, learn by usage.

注:浏览本教程时，不要试着去记，而是学习其用法。

~~

## Lesson 1.4: TEXT EDITING - INSERTION

### 1.4节:文本编辑之插入

**Press i to insert text.**

按下 i键 以插入文本

1. Move the cursor to the first line below marked --->.

移动光标到--->标记的行。

2. To make the first line the same as the second, move the cursor on top of the first character AFTER where the text is to be inserted.

为使第一行同第二行相同，移动光标到要插入文本处字符上。

3. Press i and type in the necessary additions.

按下i键，键入需要的字符。

4. As each error is fixed press to return to Normal mode.  
Repeat steps 2 through 4 to correct the sentence.

当错误被修复后，按下回到正常模式。重复步骤2到4以修改完整。

---> There is text misng this .

---> There is some text missing from this line.

5. When you are comfortable inserting text move to lesson 1.5.

当你正确插入完文本后，移步1.5节。

~~

## Lesson 1.5: TEXT EDITING - APPENDING

### 1.5节:文本编辑之追加

**Press A to append text.**

**按下A键以追加文本**

1. Move the cursor to the first line below marked --->.

It does not matter on what character the cursor is in that line.

移动光标到--->标记的行。

该行同下一行在字符上不匹配。

2. Press A and type in the necessary additions.

按下A键，然后键入要添加的内容。

3. As the text has been appended press to return to Normal mode.

当添加完文本后按下<esc>,返回到正常模式。

4. Move the cursor to the second line marked ---> and repeat steps 2 and 3 to correct this sentence.

移动光标到第二个--->标记的行。

---> There is some text missing from th

There is some text missing from this line.

---> There is also some text miss

There is also some text missing here.

5. When you are comfortable appending text move to lesson 1.6.

当你正确添加了文本后，请移步1.6节。

~~

## Lesson 1.6: EDITING A FILE

### 1.6节:编辑一个文件

Use `:wq` to save a file and exit.

`:wq` 保存文件并退出

!! NOTE: Before executing any of the steps below, read this entire lesson!!

!! 注: 执行以下步骤之前, 请读完全节!!

1. Exit this tutor as you did in lesson 1.2: `:q!`

Or, if you have access to another terminal, do the following there.

像1.2节一样, 退出本教程: `:q!`

或者, 你也可以访问其它的终端, 并执行以下步骤。

2. At the shell prompt type this command: `vim tutor`

'vim' is the command to start the Vim editor, 'tutor' is the name of the file you wish to edit. Use a file that may be changed.

在shell提示符下, 键入此命令: `vim tutor`

'vim' 是要启动的vim编辑器, 'tutor'是编辑的文件名。该文件将被编辑修改。

3. Insert and delete text as you learned in the previous lessons.

按照你在前几节你所学到的插入、删除文本。

4. Save the file with changes and exit Vim with: `:wq`

保存对文件的修改, 并退出vim: `:wq`

5. If you have quit vimtutor in step 1 restart the vimtutor and move down to the following summary.

如果你在步骤1中退出了本教程, 启动本教程并移步到以的总结。

6. After reading the above steps and understanding them: do it.

阅读完以下步骤后, 理解并执行它们。

~~

# Lesson 1 SUMMARY

## 课程1 的 总结

1. The cursor is moved using either the arrow keys or the hjkl keys.  
h (left) j (down) k (up) l (right)

可以使用光标键或者hjkl键移动光标

h (左移) j (下移) k (上移) l (右移)

2. To start Vim from the shell prompt type: vim FILENAME

从shell提示符启动VIM, 请键入:vim FILENAME <ENTER>

3. To exit Vim type: <ESC> :q! <ENTER> to trash all changes.  
OR type: <ESC> :wq <ENTER> to save the changes.

要退出VIM: <ESC> :q! <ENTER> 放弃所有修改。

或者键入: <ESC> :wq <ENTER> 保存所有的修改。

4. To delete the character at the cursor type: x

删除光标处的一个字符: x

5. To insert or append text type:

i type inserted text <ESC> insert before the cursor

A type appended text <ESC> append after the line

插入或者追加文本:

i type inserted text <ESC> 插入文本

A type appended text <ESC> 追加文本

NOTE: Pressing <ESC> will place you in Normal mode or will cancel  
an unwanted and partially completed command.

注: 按下<ESC> 会回到正常模式, 或者取消未完成的命令。

Now continue with Lesson 2.

现在继续进行第2课。

~~

## Lesson 2.1: DELETION COMMANDS



## 2.1节:删除命令

Type **dw** to delete a word.

键入 **dw** 以删除一个单词。

1. Press <ESC> to make sure you are in Normal mode.

按下 <ESC> 以确保你处于正常模式。

2. Move the cursor to the line below marked --->.

移动光标到以下以--->标记的行。

3. Move the cursor to the beginning of a word that needs to be deleted.

移动光标到需要删除的单词头。

4. Type **dw** to make the word disappear.

键入 **dw** 以使该单词消失（被删除）。

NOTE: The letter **d** will appear on the last line of the screen as you type it. Vim is waiting for you to type **w** . If you see another character than **d** you typed something wrong; press <ESC> and start over.

注: 当你键入**d**时, 字**d** 会在屏幕最低处显示, 此时VIM在等待你键入**w**。当你错误的键入时, 你会看见其它的字符而非**d**;按下 <ESC> 重新开始。

---> There are a some words fun that don't belong paper in this sentence.

5. Repeat steps 3 and 4 until the sentence is correct and go to Lesson 2.2.

重新步骤3到4,直到语句正确, 然后移步2.2节。

~~

## Lesson 2.2: MORE DELETION COMMANDS

### 2.2节:其它的删除命令

Type **d\$** to delete to the end of the line.

键入 **d\$** 以删除光标到行尾间处的字符

1. Press <ESC> to make sure you are in Normal mode.

按下 <ESC> 以确保你处于正常模式。

2. Move the cursor to the line below marked --->.

移动光标到以下以--->标记的行。

3. Move the cursor to the end of the correct line (AFTER the first . ).

移动光标到正确行的末端（在第一个.号后）。

4. Type d\$ to delete to the end of the line.

键入 d\$ 以删除到行尾。

---> Somebody typed the end of this line twice. end of this line twice.

5. Move on to Lesson 2.3 to understand what is happening.

移动到2.3节，并理解发生了什么。

~~

## Lesson 2.3: ON OPERATORS AND MOTIONS

### 2.3节:关于操作码和对象

(operator我理解为操作码，motions我理解为对象)

Many commands that change text are made from an operator and a motion.

The format for a delete command with the d delete operator is as follows:

d motion

Where:

d - is the delete operator.

motion - is what the operator will operate on (listed below).

A short list of motions:

w - until the start of the next word, EXCLUDING its first character.

e - to the end of the current word, INCLUDING the last character.

\$ - to the end of the line, INCLUDING the last character.

Thus typing de will delete from the cursor to the end of the word.

NOTE: Pressing just the motion while in Normal mode without an operator will move the cursor as specified.

许多修改文本的命令由两部分组成:操作码和对象。

删除命令d的格式如下:

d 对象

其中:

d -是操作码

motion -是操作码要操作的对象 (译者:更确切地说应是范围吧?) (罗列在下方)

关于对象的小清单:

w -到下一个单词头, 除开所在的第一个字符。

e -到下一个单词尾, 包括所在单词的最后一个字符。

\$- 到行尾, 包括了最后一个字符。

0- 到行首, 包含空格。

^-到行首, 不含空格。

G -到文本最后一行首

gg -到文本首行。

所以de会删除光标处到单词尾下的字符。

注: 不带操作码按下对象键时, 仅会按对象指定的方式 移动光标。

~~

## Lesson 2.4: USING A COUNT FOR A MOTION

### 2.4节:对对象使用计数

**Typing a number before a motion repeats it that many times.**

1. Move the cursor to the start of the line marked ---> below.
2. Type 2w to move the cursor two words forward.
3. Type 3e to move the cursor to the end of the third word forward.
4. Type 0 (zero) to move to the start of the line.
5. Repeat steps 2 and 3 with different numbers.

---> This is just a line with words you can move around in.

6. Move on to Lesson 2.5.

在对象前键入重复的次数.

1. 移动光标到以下以--->标记的行。
2. 键入 2w 以使光标向后移动两个单词。
3. 键入 3e 以使光标向后移动到第三个单词尾。
4. 键入 0 移动到行首。
5. 使用不同的数字重复步骤2到3。

---> This is just a line with words you can move around in.

6. 移步到2.5节。

~~

## Lesson 2.5: USING A COUNT TO DELETE MORE

### 2.5节:使用计数以删除得更多

**Typing a number with an operator repeats it that many times.**

为操作码键入 一个数字，使之重复多次。

In the combination of the delete operator and a motion mentioned above you insert a count before the motion to delete more:

d number motion

合并上文提到的删除操作码和对象，在对象前插入一个数字以删除更多  
d 数字 对象

1. Move the cursor to the first UPPER CASE word in the line marked --->.

移动光标到以下以--->标记的行的第一个大写单词处。

2. Type d2w to delete the two UPPER CASE words

键入 d2w 以删除2个大写 单词。

3. Repeat steps 1 and 2 with a different count to delete the consecutive UPPER CASE words with one command

用不同的计数重复步骤1和2， 以一个命令删除多个连续的大写单词。

---> this ABC DE line FGHI JK LMN OP of words is Q RS TUV cleaned up.

~~

## Lesson 2.6: OPERATING ON LINES

### 2.6节:行的操作

Type **dd** to delete a whole line.

键入 **dd** 以删除一整行.

Due to the frequency of whole line deletion, the designers of Vi decided it would be easier to simply type two d's to delete a line.

由于需要频繁的删除一整行，vi的设计者决定通过简单的键入两d轻松删除一整行。

1. Move the cursor to the second line in the phrase below.
2. Type **dd** to delete the line.
3. Now move to the fourth line.
4. Type **2dd** to delete two lines.
5. 移动到以下短语的第二行。
6. 键入 **dd** 以删除行。
7. 现在移动到第四行。
8. 键入 **2dd** 以删除后两行。

---> 1) Roses are red,  
---> 2) Mud is fun,  
---> 3) Violets are blue,  
---> 4) I have a car,  
---> 5) Clocks tell time,  
---> 6) Sugar is sweet  
---> 7) And so are you.

~~

## Lesson 2.7: THE UNDO COMMAND

### 2.7节:恢复命令

Press **u** to undo the last commands, **U** to fix a whole line.

按下**u** 以恢复最后的命令，**U** 是修复一整行.

1. Move the cursor to the line below marked ---> and place it on the first error.
2. Type x to delete the first unwanted character.
3. Now type u to undo the last command executed.
4. This time fix all the errors on the line using the x command.
5. Now type a capital U to return the line to its original state.
6. Now type u a few times to undo the U and preceding commands.
7. Now type CTRL-R (keeping CTRL key pressed while hitting R) a few times to redo the commands (undo the undo's).
8. 移动光标到以--->标记的行，定位到第一处错误。
9. 键入 x 以删除第一个不要的字符。
10. 现在 键入 u 以恢复最后执行的命令。
11. 这次用x命令修复该行的所有错误。
12. 现在键入一个大写的 U 以回到行的原始状态。
- 6.现在按几次 u 以恢复U 和之间的命令。
13. 现在 按几次 CTRL+R（按住CTRL同时敲击R），以撤消恢复命令（与u命令相反）。

---> Fiix the errors oon thhis line and reeplace them wittth undo.

8. These are very useful commands. Now move on to the Lesson 2 Summary.
9. 这些都是非常有用的命令。现在移步到第2课的总结。

~~

## Lesson 2 SUMMARY

### 第2课总结

1. To delete from the cursor up to the next word type: dw

删除光标到下一个单词: dw

2. To delete from the cursor to the end of a line type: d< br/ > 删除光标到行尾 : d
3. To delete a whole line type: dd

删除整行: dd

4. To repeat a motion prepend it with a number: 2w

预置数字以重复对象: 2w

5. The format for a change command is:

operator [number] motion

where:

operator - is what to do, such as d for delete

[number] - is an optional count to repeat the motion

motion - moves over the text to operate on, such as w (word),

\$ (to the end of line), etc

更改命令的格式:

操作码 [数字] 对象

6. To move to the start of the line use a zero: 0

用0移动到行首: 0

7. To undo previous actions, type: u (lowercase u)

To undo all the changes on a line, type: U (capital U)

To undo the undo's, type: CTRL-

恢复之前的动作, 键入: u (小写)

恢复对一行的所有改变, 键入:U(大写)

恢复恢复, 键入: CTRL-R

~~

## Lesson 3.1: THE PUT COMMAND

### 3.1节:PUT命令

Type p to put previously deleted text after the cursor.

输入 p 以在光标处粘贴先前删除的文本。

1. Move the cursor to the first ---> line below.

移动光标到以下以--->标记的行。

2. Type dd to delete the line and store it in a Vim register.

输入 dd 以删除该行, 该行会被存储至VIM寄存器中。

3. Move the cursor to the c) line, ABOVE where the deleted line should go.

移动光标到c)行, 即被删除行要到达的前行。

4. Type p to put the line below the cursor.

键入 p 将那行粘贴到光标之下。

5. Repeat steps 2 through 4 to put all the lines in correct order.

重复步骤2到4以粘贴所有的行到正确的位置。

---> d) Can you learn too?  
---> b) Violets are blue,  
---> c) Intelligence is learned,  
---> a) Roses are red,

~~

## Lesson 3.2: THE REPLACE COMMAND

### 3.2节:替换命令

Type rx to replace the character at the cursor with x .

键入 rx 以替换光标处的字符

1. Move the cursor to the first line below marked --->.

移动光标到以--->标记的行。

2. Move the cursor so that it is on top of the first error.

移动光标到第一个错误。

3. Type r and then the character which should be there.

键入 r和要更正的字符。

4. Repeat steps 2 and 3 until the first line is equal to the second one.

重复步骤2和3，直到一二行相同。

---> Whan this lime was tued in, someone presswd some wrojg keys!

---> When this line was typed in, someone pressed some wrong keys!

5. Now move on to Lesson 3.3.

现在移动到3.3节吧。



NOTE: Remember that you should be learning by doing, not memorization.

注:记住你应该学会做而非记。

~~

## Lesson 3.3: THE CHANGE OPERATOR

### 3.3:变更操作码

To change until the end of a word, type **ce** .

要变更光标到词尾，键入:ce

1. Move the cursor to the first line below marked --->.

移动光标到以--->标记的行。

2. Place the cursor on the u in lubw.

定位光标至 lubw 的 u 上。

3. Type **ce** and the correct word (in this case, type **ine** ).

键入 ce 和 正确的单词（本例中，应该键入 ceine）。

4. Press <ESC> and move to the next character that needs to be changed.

按<esc>，并移动到下一个需要更改的字符。

5. Repeat steps 3 and 4 until the first sentence is the same as the second.

重复步骤3和4，以使第一句同第二句相同。

---> This lubw has a few wptfd that mrrf changing usf the change operator.

---> This line has a few words that need changing using the change operator.

Notice that **ce** deletes the word and places you in Insert mode.

注意，ce 是删除单词并定位到插入模式（译者:其实后面的e就是对象，而前面的 c是操作码）

~~

## Lesson 3.4: MORE CHANGES USING **c**

## 3.4节:其它的c更改命令

The change operator is used with the same motions as delete.

同delete 命令一样，更改操作码也使用相同的对象。

1. The change operator works in the same way as delete. The format is:

```
c    [number]    motion<br />
```

更改操作码的用法与删除操作码相同，格式为：

c [number] motion

2. The motions are the same, such as w (word) and \$ (end of line).

所有的对象是相同的，例如w(单词)、\$（行尾）。

3. Move to the first line below marked --->.

移动到以下以--->标记的第一行。

4. Move the cursor to the first error.

移动光标到第一处错误。

5. Type c\$ and type the rest of the line like the second and press <ESC>.

键入 c\$ 和 余下的同第二行相同的字符，并按下。

---> The end of this line needs some help to make it like the second.

---> The end of this line needs to be corrected using the c\$ command.

NOTE: You can use the Backspace key to correct mistakes while typing.

注: 在键入时，你可以使用退格以删除错误。

~~

## Lesson 3 SUMMARY

### 第3课总结

1. To put back text that has just been deleted, type p . This puts the deleted text AFTER the cursor (if a line was deleted it will go on the line below the cursor).

要粘贴刚刚删除的文本，键入 p.这会把被删文本粘贴到光标之后（如果某行被删，它会被粘贴到光标之后的一行）。

2. To replace the character under the cursor, type r and then the character you want to have there.

要替换光标处的字符，键入 r 和要插入的字符。

3. The change operator allows you to change from the cursor to where the motion takes you. eg. Type ce to change from the cursor to the end of the word, c\$ to change to the end of a line.

更改操作码允许你更改从光标到对象定义处。例如:键入 ce 以更改光标到词尾，键入c\$会更改光标到行尾。

4. The format for change is:

c [number] motion

更改的格式:

c [number] motion

Now go on to the next lesson.

现在进入 下一课吧。

~~

## Lesson 4.1: CURSOR LOCATION AND FILE STATUS

### 4.1节:光标定位和文件状态

Type **CTRL-G** to show your location in the file and the file status.

Type **G** to move to a line in the file.

键入 **CTRL-G** 以显示你在文件中的位置和状态.

键入**G** 以移动到文件行尾。

NOTE: Read this entire lesson before executing any of the steps!!

注:执行任何一步前请读完本课。

1. Hold down the Ctrl key and press g . We call this CTRL-G.

A message will appear at the bottom of the page with the filename and the position in the file. Remember the line number for Step 3.

按住CTRL键，并键入 g. 我们称之为CTRL-G。

在页面的底部会呈现一个信息，指出文件名和当前文件位置。

记住行号

以在第三步使用。

NOTE: You may see the cursor position in the lower right corner of the screen  
This happens when the 'ruler' option is set (see :help 'ruler' )

注: 你可能见到 处于屏幕右下角的光标

这发生在设置了'ruler'选项时（见 :help 'ruler'）

2. Press G to move you to the bottom of the file.

Type gg to move you to the start of the file.

按 G 以移动到文件底。

键入 gg 以移动到文件首。

3. Type the number of the line you were on and then G . This will return you to the line you were on when you first pressed CTRL-G.

键入你之前所在 的行号，然后键入G。

这样你会返回到你之前，即你按CTRL-G时的那一行。

4. If you feel confident to do this, execute steps 1 through 3.

如果你有信心去做它，请执行1到3步。

~~

## Lesson 4.2: THE SEARCH COMMAND

### 4.2节:搜索命令

Type / followed by a phrase to search for the phrase.

按 / 并接一个要搜索的短语

1. In Normal mode type the / character. Notice that it and the cursor appear at the bottom of the screen as with the : command.

在正常模式，键入 / 字母。注意，像:命令一样，光标会出现在屏幕底

2. Now type 'errroor' . This is the word you want to search for.

现在，键入 'errroor'.这是你想要搜索的单词。

3. To search for the same phrase again, simply type n .  
To search for the same phrase in the opposite direction, type N .

想再次搜索相同的短语，简单的键入 n.

想再次搜索相同的短语并以相反的方向，键入 N.

4. To search for a phrase in the backward direction, use ? instead of / .

要反向搜索一个短语，使用?替换/.

5. To go back to where you came from press CTRL-O (Keep Ctrl down while pressing the letter o). Repeat to go back further. CTRL-I goes forward.

要返回之前的位置，键入 CTRL-O（按住 CTRL时，按下字母o).重复几次回到更早的那刻。CTRL-I是向前移动。

---> "errroor" is not the way to spell error; errroor is an error.

NOTE: When the search reaches the end of the file it will continue at the start, unless the 'wrapscan' option has been reset.

注: 当搜索到达文件尾时，它会从文件头开始搜索，除非'wrapscan'选项被设置了。

~~

## Lesson 4.3: MATCHING PARENTHESES SEARCH

### 4.3节:括号匹配搜索

Type % to find a matching ),], or } .

键入 % 以找到所匹配的),],or }.

1. Place the cursor on any (, [, or { in the line below marked --->.

在以--->标记的行中，将光标定位到任一(,[,{.

2. Now type the % character.

现在按下 % 字符。

3. The cursor will move to the matching parenthesis or bracket.

将光标移动到匹配的括号上。

4. Type % to move the cursor to the other matching bracket.

按 % 以移动光标到其它的匹配括号。

5. Move the cursor to another (, ), [, ], { or } and see what % does.

移动光标到 另外的(, ), [, ], { or }, 看看 % 是怎么做的。

---> This ( is a test line with ('s, ['s ] and {'s } in it. ))

NOTE: This is very useful in debugging a program with unmatched parentheses!

注:在高度程序的非匹配括号是这个是相当有用的。

~~

## Lesson 4.4: THE SUBSTITUTE COMMAND

### 4.4: 替换命令

Type `:s/old/new/g` to substitute 'new' for 'old'.

键入 `:s/old/new/g` 以用 'new' 替换 'old'。

1. Move the cursor to the line below marked --->.

移动光标到以下以--->标记的行。

2. Type `:s/thee/the` . Note that this command only changes the first occurrence of "thee" in the line.

键入 `:s/thee/the` . 注意此命令仅会更改本行的第一例 "thee".

3. Now type `:s/thee/the/g` . Adding the g flag means to substitute globally in the line, change all occurrences of "thee" in the line.

现在 键入 `:s/three/the/g`. 加入的 g 标志 意味着 会替换本行所有的 "thee".

---> thee best time to see thee flowers is in thee spring.

4. To change every occurrence of a character string between two lines,
  - type `:#,#s/old/new/g` where `#, #` are the line numbers of the range of lines where the substitution is to be done.
  - Type `:%s/old/new/g` to change every occurrence in the whole file.
  - Type `:%s/old/new/gc` to find every occurrence in the whole file, with a prompt whether to substitute or not.

要更改两行间的所有的匹配字符串:

- type `:#,#s/old/new/g` 其中, `#, #`是要更改的行号的范围
- type `:%s/old/new/g` 更改全文件中的所有事件。
- Type `:%s/old/new/gc` 更改全文件中的所有事件,并给出替换与否的提示。

~~

## Lesson 4 SUMMARY

### 第4课总结

1. CTRL-G displays your location in the file and the file status.

CTRL-G 显示 当前文件位置和状态。

- G moves to the end of the file.

G 移动到文件尾。

- number G moves to that line number.

数字 G 移动到某行。

- gg moves to the first line.

gg 移动到文件头。

2. Typing `/` followed by a phrase searches FORWARD for the phrase.

键入 `/` ,接上要向前搜索的短语。

- Typing `?` followed by a phrase searches BACKWARD for the phrase.

键入 `?` 接上要向后搜索的短语

- After a search type n to find the next occurrence in the same direction
- or N to search in the opposite direction.

搜索后，按n 以相同的方向搜索下一事件，按N以相反的方向搜索。

- CTRL-O takes you back to older positions, CTRL-I to newer positions.

CTRL-O 使你返回到以前的位置，CTRL-I 回到以后的位置。

3. Typing % while the cursor is on a (,),[,],{, or } goes to its match.

键入 % 时括号上的光标会匹配自己的括号。

4. To substitute new for the first old in a line type :s/old/new

在一行中以new替换old. :s/old/new

- To substitute new for all 'old's on a line type :s/old/new/g

在一行中以new替换所有的old :s/old/new/g

- To substitute phrases between two line #'s type :#,#s/old/new/g

在#到#两行间，以new 替换所有的old :#,#s/old/new/g

- To substitute all occurrences in the file type :%s/old/new/g

替换文件中的所有事件 :%s/old/new/g

- To ask for confirmation each time add 'c' :%s/old/new/gc

每次替换前确认，增加'c' :%s/old/new/gc

~~

## Lesson 5.1: HOW TO EXECUTE AN EXTERNAL COMMAND

### 5.1节:怎样执行外部命令

Type **:! followed by an external command to execute that command.**

按 **:! 并加上要执行的外部命令。**



1. Type the familiar command `:` to set the cursor at the bottom of the screen. This allows you to enter a command-line command.

键入类似的命令: 使光标处于屏幕底。这允许你键入一个命令行。

2. Now type the `!` (exclamation point) character. This allows you to execute any external shell command.

现在键入 `!` 字符。这允许你执行一个任何外部的shell命令

3. As an example type `ls` following the `!` and then hit `.` This will show you a listing of your directory, just as if you were at the shell prompt. Or use `!dir` if `ls` doesn't work

作为一个例子, 键入 `ls` 接一个`!`,然后敲击. 这会显示一个你的目录, 就像你在一个shell提示符一样。或者 使用 `!dir` (如果 `ls`不起作用的话) 。

NOTE: It is possible to execute any external command this way, also with arguments.

注: 通过这种方式执行任何的外部命令是可能的, 同样可以跟参数。

NOTE: All `:` commands must be finished by hitting  
From here on we will not always mention it.

注: 所有的 `:` 命令必须以敲击结束。  
从这里开始, 我们不会问题提及这些问题。

~~

## Lesson 5.2: MORE ON WRITING FILES

### 5.2节:其它的写文件命令

To save the changes made to the text, type `:w FILENAME`.

要保存对文本作的改动, 键入 `:w FILENAME`.

1. Type `!dir` or `!ls` to get a listing of your directory.  
You already know you must hit after this.

键入 `!dir` or `!ls` 以得到你的目录的清单。  
你已经 知道了你必须在这句后 敲击。

2. Choose a filename that does not exist yet, such as TEST.

选择一个不存在的文件名，如TEST。

3. Now type: :w TEST (where TEST is the filename you chose.)

现在 键入 :w TEST (其中的TEST是你选择的文件名) 。

4. This saves the whole file (the Vim Tutor) under the name TEST.

To verify this, type :!dir or :!ls again to see your directory

这会以TEST的名字 保存全部文件（即这个 VIM教程）。

要确认之，请键入 :!dir or :!ls 以查看你的目录。

NOTE: If you were to exit Vim and start it again with vim TEST , the file would be an exact copy of the tutor when you saved it.

注:如果你之前退出了VIM并以vim TEST再次启动了它,这个文件就会是你保存时的教程的副本。

5. Now remove the file by typing (MS-DOS): :!del TEST  
or (Unix): :!rm TEST

现在键入以下命令删除这个文件:

(MS-DOS): :!del TEST

or (Unix): :!rm TEST

~~

## Lesson 5.3: SELECTING TEXT TO WRITE

### 5.3节:选择要写的文本

To save part of the file, type v motion :w FILENAME

要保存部分文件，键入 v 对象 :w FILENAME

1. Move the cursor to this line.

移动光标到此行。

2. Press v and move the cursor to the fifth item below. Notice that the text is highlighted.

按 v 并移动 光标到 以下的第5项。注意这些文本会理高亮。

3. Press the : character. At the bottom of the screen : '<,>' will appear.

按 : 字符。在屏幕底会显示 : '<,>' will appear.

4. Type w TEST , where TEST is a filename that does not exist yet. Verify that you see : '<,>'w TEST before you press Enter.

按 w TEST ,其中 TEST 是一个不存在 的文件。

确认你按前你看到了 : '<,>'w TEST.

5. Vim will write the selected lines to the file TEST. Use :!dir or !ls to see it. Do not remove it yet! We will use it in the next lesson.

vim 会把选定的行进文件TEST中,使用 :!dir or !ls 作检查。先别删除它! 我们会下节使用它。

NOTE: Pressing v starts Visual selection. You can move the cursor around to make the selection bigger or smaller.

Then you can use an operator

to do something with the text. For example, d deletes the text.

注: 按下v 开始选择选择。你可以将光标移动到其它地方以使选择的区变大或者 变小。

然后你可使用操作码对文本做一些事。比如, d 删除文件。

~~

## Lesson 5.4: RETRIEVING AND MERGING FILES

### 5.4节:恢复和合并文件

To insert the contents of a file, type :r FILENAME

要插入一个文件的内容键入:r FILENAME

1. Place the cursor just above this line.

NOTE: After executing Step 2 you will see text from Lesson 5.3. Then movek DOWN to see this lesson again.

只需将光标放到这一行。

注: 执行第二步后你会看到5.3节的内容。然后下移并再次查看这节。

2. Now retrieve your TEST file using the command `:r TEST` where TEST is the name of the file you used.

The file you retrieve is placed below the cursor line.

现在 恢复 你的TEST文件, 使用命令 `:r TEST` 其中, TEST 是你使用的文件。  
你要恢复的文件将被放置在光标行下面。

3. To verify that a file was retrieved, cursor back and notice that there are now two copies of Lesson 5.3, the original and the file version.

要确认文件被恢复了, 用光标返回, 注意这里有两5.3节的副本, 原本的文件版。

NOTE: You can also read the output of an external command. For example, `:r !ls` reads the output of the `ls` command and puts it below the cursor.

注: 你可能会阅读外部命令的输出。如 `:r !ls` 读入`ls`命令的输出并将其放置到光标下。

~~

## Lesson 5 SUMMARY

### 第5课总结

1. `!:command` executes an external command.

Some useful examples are:

(MS-DOS) (Unix)

`!:dir !ls` - shows a directory listing.

`!:del FILENAME !:rm FILENAME` - removes file FILENAME.

`!:command` 执行外部命令

一些有用的例子:

(MS-DOS) (Unix)

`!:dir !ls` - 显示目录清单

`!:del FILENAME !:rm FILENAME` - 移除 FILENAME文件。

2. `:w FILENAME` writes the current Vim file to disk with name FILENAME.

`:w FILENAME` 将当前的vim 文件以FILENAME 文件名写进磁盘。

3. `v motion :w FILENAME` saves the Visually selected lines in file FILENAME.

`v` 对象 `:w FILENAME` 保存选定选择行到FILENAME 文件。

4. `:r FILENAME` retrieves disk file FILENAME and puts it below the cursor position.

`:r FILENAME` 恢复磁盘文件FILENAME到光标下。

5. `:r !dir` reads the output of the `dir` command and puts it below the cursor position.

`:r !dir` 读取`dir`命令的输出并粘贴到光标下。

~~

## Lesson 6.1: THE OPEN COMMAND

### 6.1节:OPEN 命令

Type `o` to open a line below the cursor and place you in Insert mode.

键入 `o` 以在光标打开一行，并置于插入模式。

1. Move the cursor to the line below marked `--->`.

移动光标到以`--->`标记的行。

2. Type the lowercase letter `o` to open up a line BELOW the cursor and place you in Insert mode.

键入 小写字母 `o` 以在光标下插入一行，并置于插入模式。

3. Now type some text and press to exit Insert mode.

现在 键入 一些文本，并按以退出插入模式。

`--->` After typing `o` the cursor is placed on the open line in Insert mode.

4. To open up a line ABOVE the cursor, simply type a capital O , rather than a lowercase o. Try this on the line below.

要在光标前插入一行，只需键入 大写字母 O，而非小写字母 o，在下一行试试。

---> Open up a line above this by typing O while the cursor is on this line.

~~

## Lesson 6.2: THE APPEND COMMAND

### 6.2: 追加命令

Type a to insert text AFTER the cursor.

键入 a 以在光标后插入文本。

1. Move the cursor to the start of the line below marked --->.

移动 下面以--->标记的行首。

2. Press e until the cursor is on the end of li .

按 e 直到 光标在li 尾。

3. Type an a (lowercase) to append text AFTER the cursor.

按 一个 a(小写) 以追加文本到光标后。

4. Complete the word like the line below it. Press to exit Insert mode.

按照下一行完成这个单词。按退出插入模式。

5. Use e to move to the next incomplete word and repeat steps 3 and 4.

使用 e 移动到未完成的单词，并重复步骤3和4。

---> This li will allow you to pract appendi text to a line.

---> This line will allow you to practice appending text to a line.

NOTE: a, i and A all go to the same Insert mode, the only difference is where the characters are inserted.

注: a,i 和A 都会进入插入模式，仅有的不同是字符在哪里插入。

~~

## Lesson 6.3: ANOTHER WAY TO REPLACE

### 6.3节: 以别一种方式替换

Type a capital R to replace more than one character.

键入 大写 R 以替换 更多的字符而非一个。

1. Move the cursor to the first line below marked --->. Move the cursor to the beginning of the first xxx .

移动光标到以---》标记的第一行。移动光标到第一个XXX。

2. Now press R and type the number below it in the second line, so that it replaces the xxx .

现在 按下 R 并键入 在第二行显示的数字，以其替换xxx.

3. Press to leave Replace mode. Notice that the rest of the line remains unmodified.

按下 以 离开替换模式。注意 剩余的行仍没有改变。

4. Repeat the steps to replace the remaining xxx.

重复以上步骤以替换剩余的XXX。

---> Adding 123 to xxx gives you xxx.

---> Adding 123 to 456 gives you 579.

NOTE: Replace mode is like Insert mode, but every typed character deletes an existing character.

注: 替换模式下类似于插入模式，但是键入字符会删除已经存在的字符。

~~

## Lesson 6.4: COPY AND PASTE TEXT

### 6.4: 复制粘贴文本

## Use the y operator to copy text and p to paste it

用y 操作码来复制文本，以 p 粘贴此文本。

1. Go to the line marked with ---> below and place the cursor after "a)"

移动到以--->标记的行并定位到“a)”后。

2. Start Visual mode with v and move the cursor to just before "first".

以v 开启选择模式,移动光标到“first”前。

3. Type y to yank (copy) the highlighted text.

按 y 以复制高亮的文本。

4. Move the cursor to the end of the next line: j< br/ > 移动光标到下一行的 : j

5. Type p to put (paste) the text. Then type: a second <ESC> .

按p 以粘贴文本，然后键入:a second <esc>.

6. Use Visual mode to select " item.", yank it with y , move to the end of the next line with j\$ and put the text there with p .

使用选择模式以选择” item。“，以y复制之，以j\$移动到下行并以p粘贴到那里。

---> a) this is the first item.

b)

NOTE: you can also use y as an operator; yw yanks one word.

注: 你可能会使用y 作操作码;yw 复制一个单词。

~~

## Lesson 6.5: SET OPTION

### 6.5节:设置选项

Set an option so a search or substitute ignores case

设置搜索和替换时忽略大小写的选项

1. Search for 'ignore' by entering: /ignore <ENTER>  
Repeat several times by pressing n .



搜索'ignore'，键入：/ignore <enter>

重复按n几次。

2. Set the 'ic' (Ignore case) option by entering: :set ic

设置 'ic'（忽略大小写）选项，键入: :set ic

3. Now search for 'ignore' again by pressing n  
Notice that Ignore and IGNORE are now also found.

现在 以n再次搜索'ignore'.

注意，Ignore and IGNORE都可以被搜到。

4. Set the 'hlsearch' and 'incsearch' options: :set hls is

设置选项'hlsearch' and 'incsearch': set hls is

5. Now type the search command again and see what happens: /ignore <ENTER>

同在键入搜索命令，看看发生 了什么: /ignore <enter>

6. To disable ignoring case enter: :set noic

要禁止忽略大小写，键入: :set noic

NOTE: To remove the highlighting of matches enter: :nohlsearch

注意: 要删除高亮匹配，键入: :nohlsearch

NOTE: If you want to ignore case for just one search command, use \c  
in the phrase: /ignore\c <ENTER>

注: 如果 你只是想在某次搜索命令时忽略大小写，在短语后加上\c: /ignore\c <enter>

~~

## Lesson 6 SUMMARY

### 第6课总结

1. Type o to open a line BELOW the cursor and start Insert mode.  
Type O to open a line ABOVE the cursor.

按o 以插入一行，在光标后，并进入插入模式。

按O 以在光标前插入一行。

2. Type a to insert text AFTER the cursor.

Type A to insert text after the end of the line.

按a 以在光标后插入文本。

按A在行尾插入文本。

3. The e command moves to the end of a word.

e命令使光标移动到单词尾。

4. The y operator yanks (copies) text, p puts (pastes) it.

y操作码 复制文本，p 粘贴此文本。

5. Typing a capital R enters Replace mode until <ESC> is pressed.

调入大写字母R，进入替换模式，直到<esc>按下。

6. Typing ":set xxx" sets the option "xxx". Some options are:

'ic' 'ignorecase' ignore upper/lower case when searching

'is' 'incsearch' show partial matches for a search phrase

'hls' 'hlsearch' highlight all matching phrases

You can either use the long or the short option name.

键入":set xxx" 以设置选项 "xxx". 一些选项:

'ic' 'ignorecase' 搜索时忽略大小写

'is' 'incsearch' 部分显示匹配的搜索短语。

'hls' 'hlsearch' 高亮所有匹配的短语

你可选用长的或者短的选项名。

7. Prepend "no" to switch an option off: :set noic

前置"no"以关闭选项: :set noic

~~

# Lesson 7.1: GETTING HELP

## 7.1节:使用帮助

### Use the on-line help system

#### 使用在线帮助系统

Vim has a comprehensive on-line help system. To get started, try one of these three:

- press the <HELP> key (if you have one)
- press the <F1> key (if you have one)
- type :help <ENTER>

vim 有一个广泛的在线帮助系统。要启用之，可试试这三种之一：

- 按 <help> (如果你有)
- 按 <F1> (如果你有)
- 按 :help <enter>

Read the text in the help window to find out how the help works.

Type CTRL-W CTRL-W to jump from one window to another.

Type :q <ENTER> to close the help window.

阅读help窗口以查看help是如何进行的。

Type CTRL-W CTRL-W 从一个窗口跳转到另一个窗口。

Type :q <ENTER> 关闭帮助 窗口。

You can find help on just about any subject, by giving an argument to the ":help" command. Try these (don't forget pressing <ENTER>):

```
:help w
:help c_CTRL-D
:help insert-index
:help user-manual
```

你可以通过为":help"给一个参数找到任何主题的帮助,试试这些（别忘了<enter>）

```
:help w
:help c_CTRL-D
:help insert-index
:help user-manual
```

~~

## Lesson 7.2: CREATE A STARTUP SCRIPT

### 7.2节:创建启动脚本

#### Enable Vim features

#### 启用VIM特性

Vim has many more features than Vi, but most of them are disabled by default. To start using more features you have to create a "vimrc" file.

VIM 拥有比vi更多的特性，但是大多数是被默认禁止的。要使用更多的特性，你可创建“vimrc”文件。

1. Start editing the "vimrc" file. This depends on your system:

`:e ~/.vimrc` for Unix

`:e $VIM/_vimrc` for MS-Windows

开始编辑“vimrc”文件，这依赖于你的系统。

`:e ~/.vimrc` for Unix

`:e $VIM/_vimrc` for MS-Windows

2. Now read the example "vimrc" file contents:

`:r $VIMRUNTIME/vimrc_example.vim`

现在阅读下“vimrc”文件的例子:

`:r $VIMRUNTIME/vimrc_example.vim`

3. Write the file with:保存文件

`:w`

The next time you start Vim it will use syntax highlighting.

You can add all your preferred settings to this "vimrc" file.

For more information type `:help vimrc-intro`

下次，你启动VIM后，它会使语法高亮，你可以在“vimrc”中添加所有你喜欢的设置。

更多信息请键入 `:help vimrc-intro`

~~

## Lesson 7.3: COMPLETION

## 7.3节:补齐

### Command line completion with CTRL-D and <TAB>

#### 使用CTRL-D和<TAB>使命令行补齐

1. Make sure Vim is not in compatible mode: `:set nocompatible`

确保VIM 处于非兼容模式: `:set nocompatible`

2. Look what files exist in the directory: `!ls` or `!dir`

看看目录下有什么文件存在: `!ls` or `!dir`

3. Type the start of a command: `:e`

键入命令的开始部分: `:e`

4. Press CTRL-D and Vim will show a list of commands that start with "e".

按 CTRL-D , VIM会显示以“e”开头的清单。

5. Press <TAB> and Vim will complete the command name to ":edit".

按下<TAB>, VIM会以“:edit”补齐命令名。

6. Now add a space and the start of an existing file name: `:edit FIL`

现在 添加一个空格和已存的文件名头部: `:edit FIL`

7. Press <TAB>. Vim will complete the name (if it is unique)

按<tab>, vim会完成余下的文件名（如果它是独有的话）。

NOTE: Completion works for many commands. Just try pressing CTRL-D and <TAB>. It is especially useful for `:help` .

注:补齐针对大多数命令有效, 只需按CTRL-D和<TAB>.特别是对 `:help` 非常有用。

~~

## Lesson 7 SUMMARY

### 第七课总结

1. Type :help or press <F1> or <Help> to open a help window.

键入 :help 或 <F1> 或<help>以打开帮助窗口

2. Type :help cmd to find help on cmd .

键入 :help cmd 以找出关于cmd的帮助 。

3. Type CTRL-W CTRL-W to jump to another window

键入 CTRL-W 以跳转到另一窗口。

4. Type :q to close the help window

键入 :q 以关闭帮助窗口。

5. Create a vimrc startup script to keep your preferred settings.

创建 vimrc 启动脚本以保存你喜欢的设置。

6. When typing a : command, press CTRL-D to see possible completions.

Press <TAB> to use one completion.

当键入 :command 时，按下CTRL-D 以查看可能的补齐。

按<tab>使用其中一个补齐。

~~

This concludes the Vim Tutor. It was intended to give a brief overview of the Vim editor, just enough to allow you to use the editor fairly easily. It is far from complete as Vim has many many more commands. Read the user manual next: ":help user-manual".

本文总结了vim教程。它原本给出了VIM编辑器的简短的概要，只能足以让你很容易地使用编辑器。这远没有完成，因为VIM在如此众多的命令。请阅读下面的用户手册：

:help user-manual

For further reading and studying, this book is recommended:

Vim - Vi Improved - by Steve Oualline

Publisher: New Riders

The first book completely dedicated to Vim. Especially useful for beginners.

There are many examples and pictures.

See <http://iccf-holland.org/click5.html>

要进一步的阅读和学习，我们推荐这本书：

Vim - Vi Improved -作者Steve Oualline

出版社:New Riders

第一本书专门针对 Vim. 特别适合 新手.

其中有许多实例和图示。

See <http://iccf-holland.org/click5.html>

This book is older and more about Vi than Vim, but also recommended:

Learning the Vi Editor - by Linda Lamb

Publisher: O'Reilly & Associates Inc.

It is a good book to get to know almost anything you want to do with Vi.

The sixth edition also includes information on Vim.

这本书比较老并且主要内容是VI而非vim，但是仍值得推荐：

Learning the Vi Editor - by Linda Lamb

出版社: O'Reilly & Associates Inc.

它是一本能告诉你你想利用VI做的几乎所有事情的好书。

此书的第三版也包含了有关VIM的内容。

This tutorial was written by Michael C. Pierce and Robert K. Ware,

Colorado School of Mines using ideas supplied by Charles Smith,

Colorado State University. E-mail: [bware@mines.colorado.edu](mailto:bware@mines.colorado.edu).

Modified for Vim by Bram Moolenaar.

本教程由科罗拉多矿业学院的Michael C. Pierce 和 Robert K. Ware编写，同时Charles Smith 提供了很多创意。

E-mail: [bware@mines.colorado.edu](mailto:bware@mines.colorado.edu).

Bram Moolenaar 为VIM对本教程作了修订。

~~

## 下面是一些补充:

CTRL+F:往下翻页

CTRL+B:往上翻页

CTRL+E: 往下滚动

CTRL+Y: 往上滚动

## 行的拼接:

在选择模式选中要拼接的行:ggvG

输入: :<,>j<enter>

注:普通的j用于拼接单行。

## 列操作:

-选中全文的第一列

-按gg跳至开头

-按CTRL——V进入列选择状态:visula block

-按G到末尾

## 将每行的第一列变为大写

-按U将选中内容变大写

-按u将选中内容变小写

-按~将大小翻转。

## 列操作:在第行前加一个星号和空格:

-选中第一列

-按I进入插入状态

-输入:

```
*<space><ESC>
```

-选中前两列后按x, 可时行列删除

## 宏的录制:将所有偶数行后加入OK!



-按gg进入第一行

-按qm进入宏录制

-按j\$进入第二行尾

-按aok!以输入ok!

-按<esc>返回

-按j到下一行

-按q结束宏录制

## 宏的播放：（接上面的步骤）

单次播放：

-按下@m,执行m寄存器中的宏

-可以发现第4行中的尾已经加入了ok!

多次播放：

-按下100@m，这里重复播放100次宏。

点号（重复上一次命令）：

## 将第行尾加一个感叹号

-按ggA!以在第一行添加感叹号

-按<esc>j以到下一行。

-按.表示重复前一个操作，即在执行A!<esc>以在行尾添加一个感叹号。

-反复按j.j.j.j.

:%s/\$!/g

## 基本设置

set nocp 设置不兼容模式

set ru 打开状态栏标尺

set hls 搜索时高亮显示找到的文本

set is 递进搜索

syntax on 打开关键字上色

set backspace=indent,eol,start

set whichwrap=b,s,<,>[,]

\ h,l,~

set sw=4 缩进尺寸4个空格

set ts=4 Tab宽度为4个字符

set et 将所有的TAB替换为空格

C/C++编码设置

set cin 自动缩进

set ai 还是自动缩进

set cino=:0g0t0(sus

set sm 显示括号匹配

在console下粘贴版式的问题

:set paste

:set nopaste

= 命令重调格式

断行设置

set lbr 不在单词中间断行

图形界面与字符界面分别设置

if (has("gui\_running"))

图形界面设置

set nowrap 不拆行

set guioptions+=b 水平流动条

colo torte 配色方案

else

字符界面设置

set wrap 拆行

colo ron 配色方案

endif