

## VI 使用完全教程

### vi 简介

vi 文本编辑器使用了两种主要的模式：命令模式和插入模式。本教程的第一部分将重点关注于导航文件，这个任务可以在命令模式中完成。当您处于命令模式中时，普通的键盘操作用来执行命令，而不是创建文本。当您进入到插入模式，可以使用键盘输入文本，例如在命令行中。要退出命令模式，可以按 **Esc** 键。

vi 中的命令有些是单键命令，有些是使用 Shift 或 Ctrl 或按键序列的命令。在使用引用一个大写字母的命令时，您应该使用 Shift 键加上这个字母。在使用引用两个字母或符号的命令时，您应该按顺序按下这些键，而不是同时按下。

要开始练习，首先您将在命令行中使用 vi 命令加上新文件的名称，以创建一个空白文件。在本教程中，您在 vi 中从头开始建立了一个文档，然后学习使用有用的 vi 命令对该文档进行编辑。在完成本教程之后，您将了解所有主要的 vi 命令，这些命令可以用来完成日常的编辑任务，以及一些功能强大的命令，您可以在适当的时候使用它们。登录到您最喜欢的类 UNIX 操作系统，然后使用 vi 打开一个新的文件（请参见图 1）。

图 1. 使用 vi 打开一个新的文件



vi 打开了一个名为 tutorial.txt 的新文件（请参见图 2）。您马上将看到奇怪的地方：文本编辑器最左边的一栏中填满了波浪符号。不要担心，这是 vi 表示文档中未定义的部分的方式。换句话说，因为该文件没有任何内容，所以这些行并不存在。

图 2. vi 中的空白文件



在开始进行任何操作之前，您应该了解如何保存文件以及如何编辑文件。要输入这些类型的命令，可以按冒号（:）键加上描述所需操作的字母序列。要保存新的文件，可以按 : 键、**w** 键，然后按 **Enter** 键。要退出 vi，可以按 : 键、**q** 键，然后按 **Enter** 键。现在，重新在命令行中打开 vi。如果您希望退出 vi 而不保存所做的更改，那么它会发出警告并提示您按**感叹号（!）**以确认您的操作。如果您希望忽略系统警告，可以在按键序列的后面追加一个感叹号，退出 vi 的按键序列是：

```
: key, q key, ! key, Enter key
```

您可以将命令组合使用，如 **wq**，表示保存文件并退出 vi。

您需要完成的第一项任务是插入一些文本，以便能够学习如何编辑和操作文件。在缺省情况下，进入 vi 后处于命令模式。所以，如果您这时开始输入，那么可能会对得到的结果感到困惑。稍后将对每种模式进行介绍，现在请按 **i** 键以进入到插入模式，然后输入一些文本行内容，这与在普通文本编辑器中的操作相同。在完成输入后，按 **Esc** 键返回到命令模式。请参见图 3 和 4。

图 3. 当您进入到插入模式时，编辑器中最下面的一行将会说明这一点



图 4. 在示例文档中输入一些内容，以便可以对其进行操作



在按了 **Esc** 键之后，光标仍然位于输入的最后一个字符处，并且返回到了命令模式。下一部分将向您介绍如何导航文件。

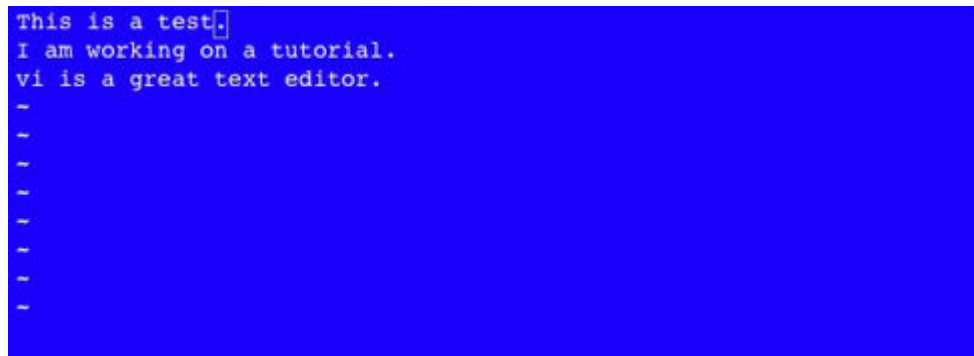
### 在 vi 中进行导航

在命令模式中，您的键盘变成了与界面的交互工具，而不是文本输入工具。vi 可以让那些需要使用所有常用命令的用户将手保持在标准键位（a-s-d-f 和 j-k-l-;）上，并延伸到附近的字母。首先要学习的基本操作是移动光标。大多数最新版本的 vi 允许您使用键盘上的箭头键，但是高级的 vi 用户更喜欢使用手指易于触及的按键 **h-j-k-l**：

- **h** 和 **l** 分别表示向左和向右，这是非常直观的，因为将它们绑定到了四个方向键中最左边和最右边的按键上。
- **k** 向上移动光标。
- **j** 向下移动光标。

您将很快地记住这些按键。要将光标移动到您所创建的包含三行内容的文件中的第一行，可以按两次 **k** 键。光标现在位于第一行的末尾。继续进行操作，并使用 **h-j-k-l** 将光标移动到文件中的其他位置，然后再次让它返回到第一行的末尾。请参见图 5。

图 5. 使用方向键 **h-j-k-l** 让光标返回到文件的第一行



```
This is a test.  
I am working on a tutorial.  
vi is a great text editor.  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

### 使用快捷方式移动光标

#### 在一行中移动

现在您的光标已经位于第一行的末尾，您可能希望将它移动到该行的开头，但却不希望反复地按 **h** 键以使光标移动到那里。在命令模式中，**vi** 具有一些基于键盘的快捷方式，它们允许您快速地移动到文件中的各个位置，与您将手移动到鼠标并指向文件中的某个位置，或使用箭头键一次移动一个位置相比，这种方式的速度更快。这些快捷方式中的第一个是 **0**：

- 要移动到一行的开头，可以按 **0**，您的光标将跳转到该位置。
- 要移动到一行的末尾，可以按 **\$**。

继续练习并尝试这种方法。

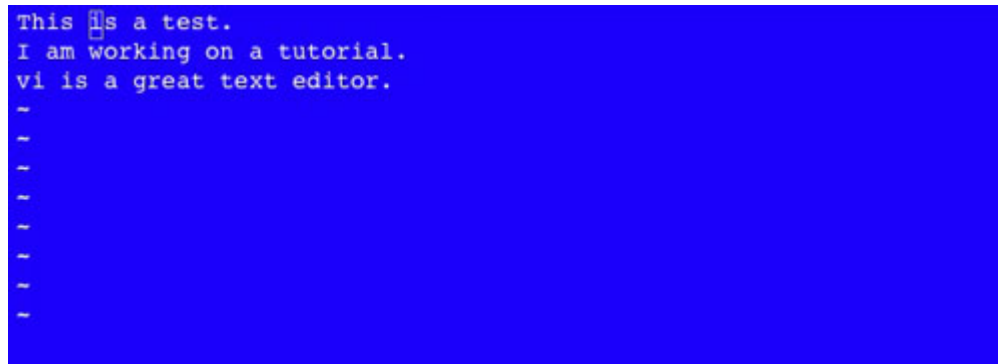
现在，您可以一次移动一个字符。您也可以将光标移动到每一行的开头和末尾，但是这两种操作在移动粒度方面相差很大。另一种选择是一次移动一个单词。您可以使用 **w** 和 **b** 键来完成这项任务：

- 按 **w** 键将向前移动一个单词。
- 按 **b** 键将向后移动一个单词。

您可以尝试这种方法，先将光标移动到第一行的开头（按 **0**），然后按三次 **w** 键将光标移动到单词 *test* 的开头。接下来，按两次 **b** 键以返回到单词 *is*。

您可能注意到了，**w** 键和 **b** 键将光标定位于每个单词的开头。您还可以使用 **e** 键向前移动光标，导航到单词的末尾，或按 **g** 键。按 **g** 向后移动。请参见图 6。

图 6. 使用 **w** 和 **b** 键逐个单词地移动光标，这是一种快速导航到拼写错误或需要更改的单词的好方法



### 从一行移动到另一行

现在，您可以在一行中快速地进行移动，但是如果正在处理一个较大的文件，还需要快速地从一行移动到另一行。在 **vi** 中，有许多方法可以用来完成这项任务。您可以使用向下或向上的移动命令 (**k** 和 **j**)，或者您可以使用 **page-up** 和 **page-down** 命令。**vi** 中的大多数命令不需要按 **Ctrl** 键，但是 **page-up** 和 **page-down** 这对命令是这条规则的例外：

- 按 **Ctrl-u** 向上翻一页。
- 按 **Ctrl-d** 向下翻一页。

要快速地导航到文件的开头或末尾，您可以按 **gg** 或 **G**：

- 按 **gg** 可以将光标移动到文档中的第一行。
- 按 **G** 可以将光标移动到文档中的最后一行。

还可以在附加方法中指定行数，这些内容将在在 **vi** 命令前面加上数字部分中讨论。

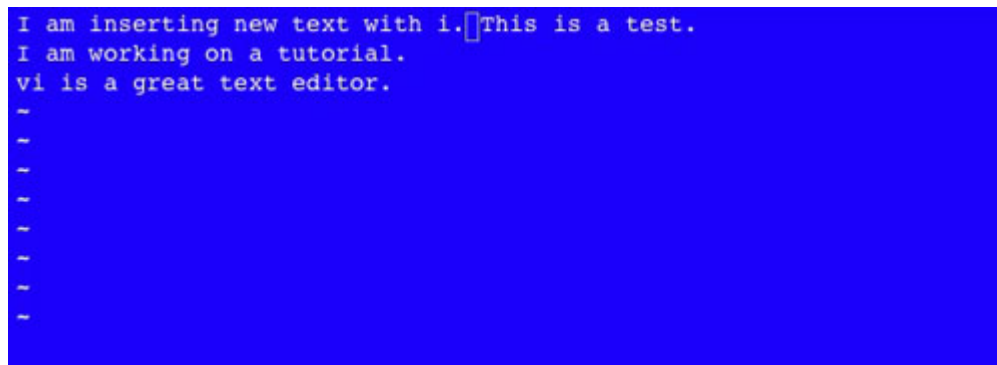
### 在 **vi** 中进行插入和编辑

在导航文档的过程中，您仍然处于命令模式，并且使用键盘作为界面交互工具来移动光标。下一组命令模式按键提供了各种进入插入模式的方法，可以用来向文件中输入新的文本。在

本教程开始处输入初始文本时，您使用了最基本的方式进入到插入模式：按 **i** 键，它表示 *insert*。按 **i** 可以在当前光标位置的前面进入到插入模式。

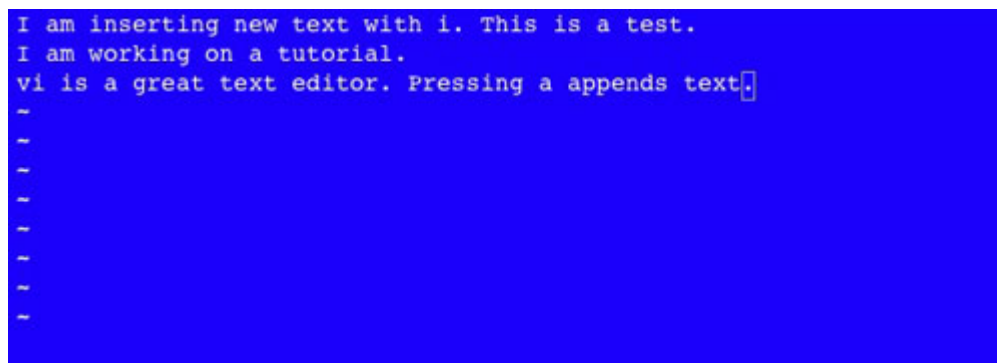
例如，如果您按 **gg** 导航到文件的开头，那么您可以按 **i** 键，这将使得您输入的任何文本都出现在当前行文本之前。按 **gg**、**i**，然后输入 I am inserting new text with i。在完成输入后，按 **Esc** 键返回到命令模式。请记住，在输入完新的文本之后，您必须返回到命令模式，否则无法对文档进行导航。请参见图 7。

图 7. 添加新文本的最简单的方法是使用 **i** 键进入到插入模式

A screenshot of a terminal window with a blue background and white text. The text displayed is: "I am inserting new text with i. This is a test." on the first line, "I am working on a tutorial." on the second line, and "vi is a great text editor." on the third line. Below the third line, there are several tilde (~) characters representing empty lines in the file. The cursor is positioned at the end of the first line, after the space following "i".

添加新文本的另一个基本方法是使用 **a** 键，它表示 *append*。使用 **a** 键可以进入到插入模式，但它将在光标当前位置之后添加文本，而不是在光标当前位置之前。要测试该命令，定位到文档的最后一行，按 **G** 和 **\$** 键定位到该行的末尾。然后按 **a** 键，输入 Pressing a appends text，并按 **Esc** 键返回到命令模式。请参见图 8。

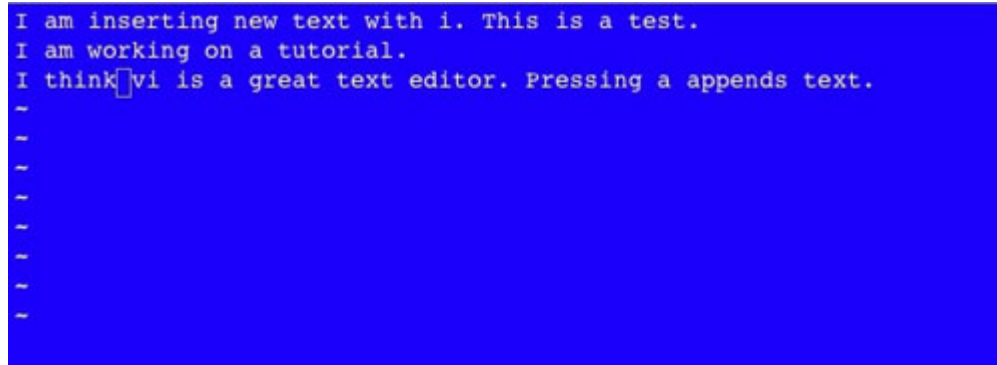
图 8. 另一种插入文本的方法是使用 **a** 键，它表示 *append*

A screenshot of a terminal window with a blue background and white text. The text displayed is: "I am inserting new text with i. This is a test." on the first line, "I am working on a tutorial." on the second line, and "vi is a great text editor. Pressing a appends text" on the third line. Below the third line, there are several tilde (~) characters representing empty lines in the file. The cursor is positioned at the end of the third line, after the space following "text".

现在您的光标位于文件最后一行的句号处。如果您现在按 **i** 键，那么会将文本插入到句号之前。如果您按 **a** 键，则会将文本插入到句号之后。通过按 **I** 键（大写字母），您可以在一行的开头进行输入，即使您的光标位于该行的末尾。与之类似，如果您按大写 **A** 键，

那么您可以在该行的末尾输入文本，无论光标当前处于什么位置。要测试这个命令，可以按 **I**，输入 **I think**，然后按 **Esc** 键。请参见图 9。

图 9. 要在一行的开头插入文本（无论光标当前处于什么位置），可以按 **I** 键

A screenshot of a terminal window with a blue background and white text. The text displayed is: "I am inserting new text with i. This is a test." followed by "I am working on a tutorial." and "I think" followed by a cursor (a small white box) and "vi is a great text editor. Pressing a appends text." Below this, there are several tilde (~) characters representing additional lines of text.

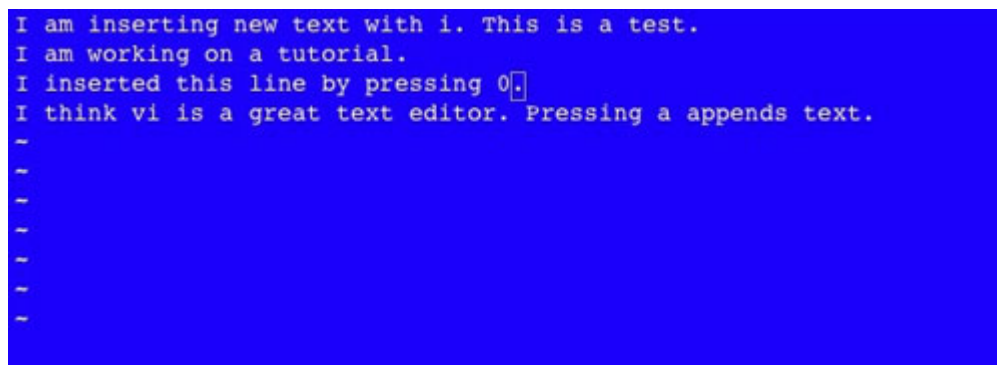
```
I am inserting new text with i. This is a test.  
I am working on a tutorial.  
I think $\square$ vi is a great text editor. Pressing a appends text.  
~  
~  
~  
~  
~  
~  
~  
~
```

插入新文本的另一种有用的方法是，在进入插入模式的同时，向您的文本文件添加新的一行。与普通的文本插入一样，新的一行可以插入到光标位置之前或之后：

- 要在当前光标位置之前插入新的一行，可以按 **O** 键。
- 要在当前光标位置之后插入新的一行，可以按 **o** 键。

要尝试使用这个命令，可以按 **O**，输入 **I inserted this line by pressing O**，然后按 **Esc** 键以返回到命令模式。请参见图 10。

图 10. 要在光标之前插入新的文本行，可以按 **O** 键

A screenshot of a terminal window with a blue background and white text. The text displayed is: "I am inserting new text with i. This is a test." followed by "I am working on a tutorial." and "I inserted this line by pressing O" followed by a cursor (a small white box) and "vi is a great text editor. Pressing a appends text." Below this, there are several tilde (~) characters representing additional lines of text.

```
I am inserting new text with i. This is a test.  
I am working on a tutorial.  
I inserted this line by pressing O $\square$   
I think vi is a great text editor. Pressing a appends text.  
~  
~  
~  
~  
~  
~  
~  
~
```

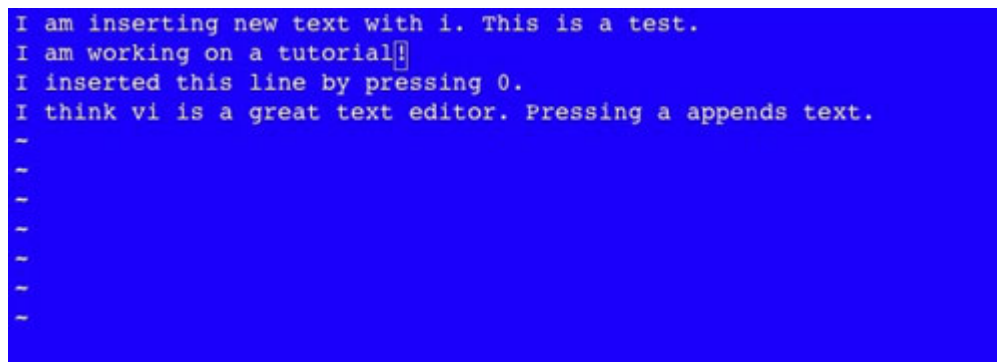
您现在已经使用了一些主要的方法来插入新的文本。回顾一下，这些键分别是 **a**、**i**、**A**、**I**、**o** 和 **O**。您能记住其中每个字母表示什么操作吗？即使现在没有记住，请不要担心，在使用一段时间之后您就会记住了。

## 替换文本

既然已经向 `tutorial.txt` 文件中输入了一些内容，您可能发现其中存在一些错误，或者可能希望更改其中的一些单词。在学习如何删除文本之前，您应该学习如何替换文本。为什么呢？因为如果您学习了如何添加新的文本和删除旧的文本，您可能会养成使用删除命令的习惯，而在某些情况下 `replace` 命令可能更加有效。与先删除一个单词，然后在该处添加一个新单词的方法相比，使用单步处理对单词进行替换更加便捷。

`vi` 使用了两种重要的替换命令。第一种是 `r` 键，它将删除光标位置的字符，并进入到插入模式以便输入单个替换的字符。换句话说，您可以在按了 `r` 之后输入一个字符。因此，`vi` 自动地返回到命令模式（不需要按 `Esc` 键）。要尝试该命令，可以使用 `k` 键和 `l` 键导航到第二行的末尾。您的光标应该位于句号处。要将句号更改为感叹号，可以按 `r` 键，然后按 `!` 键。请参见图 11。

图 11. 要替换单个字符，可以按 `r` 键，然后输入文档中需要的字符



```
I am inserting new text with i. This is a test.
I am working on a tutorial
I inserted this line by pressing o.
I think vi is a great text editor. Pressing a appends text.
-
-
-
-
-
-
-
```

还有一种更有价值的 `replacement` 命令，可以按 `c` 键，然后按 `w` 键，它们组合在一起表示 *change word*。这个命令删除当前的单词，并进入到插入模式，以便您可以立即输入替换的单词。在这种情况下，输入完新的单词后，您需要按 `Esc` 键以通知 `vi` 完成了替换任务。将光标向下移动到单词 *great*，按 `c` 和 `w` 键，输入 `cool`，然后按 `Esc` 键。请参见图 12。



图 12. 要替换单词，可以按 **c** 和 **w** 键，然后输入新的单词

```
I am inserting new text with i. This is a test.
I am working on a tutorial!
I inserted this line by pressing 0.
I think vi is a cool text editor. Pressing a appends text.
~
~
~
~
~
~
~
```

如果您希望删除文本，而不是对其进行替换，那么您需要使用 `delete` 命令。与 `vi` 中其他的操作一样，您有多种方法可供选择，这取决于您希望一次删除多少数据。最基本的 `delete` 命令是 `x` 键，它一次删除一个字符。要尝试使用该命令，可以返回到文本文件的开头，按 **Ctrl-u** (**page-up**) 键。按 **\$** 键定位到该行的末尾，然后按五次 **x** 键，以删除 *test.*。请参见图 13。

## 删除文本

图 13. 要逐个地删除字符，可以使用 **x** 键

```
I am inserting new text with i. This is a
I am working on a tutorial!
I inserted this line by pressing 0.
I think vi is a cool text editor. Pressing a appends text.
~
~
~
~
~
~
~
```

按五次 **x** 键可以完成这项任务，但是您可能希望 `vi` 能够为各种操作提供更简单的方法。按 **d** 键和 **w** 键，您可以一次删除一个单词，这两个键组合在一起表示 *delete word*。按三次 **b** 键，返回到单词 *This*。要删除这个单词，按 **d** 键和 **w** 键。请参见图 14。

图 14. 要删除单词，可以使用 dw 命令

```
I am inserting new text with i. Iis a
I am working on a tutorial!
I inserted this line by pressing 0.
I think vi is a cool text editor. Pressing a appends text.
~
~
~
~
~
~
~
```

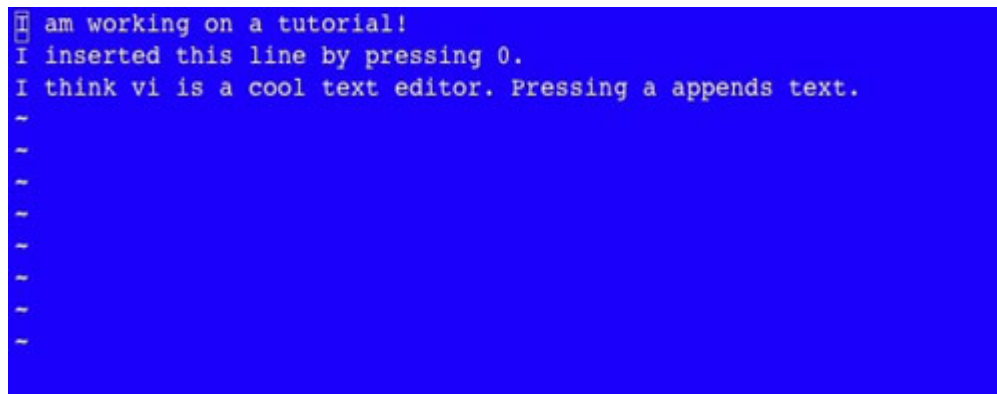
现在第一行中包含一个半截的、不完整的句子。要删除这一行中光标位置之后所有的内容，您可以使用 d 键加上 \$ 键，前面曾经介绍过，后者用来表示到一行末尾的操作。按 d 键和 \$ 键，以便删除该行末尾的 *is a*。请参见图 15。

图 15. 要删除光标到一行末尾的文本，可以使用 d\$

```
I am inserting new text with i.I
I am working on a tutorial!
I inserted this line by pressing 0.
I think vi is a cool text editor. Pressing a appends text.
~
~
~
~
~
~
~
```

vi 中最后一个常用的删除命令是 delete-line 命令，可以按两次 d 键来完成该操作。它会删除一行文本，并将下面一行内容上移，这样一来，文档中就不会出现空行。要删除文件中的第一行，可以按 d 键，然后再按一次该键。请参见图 16。

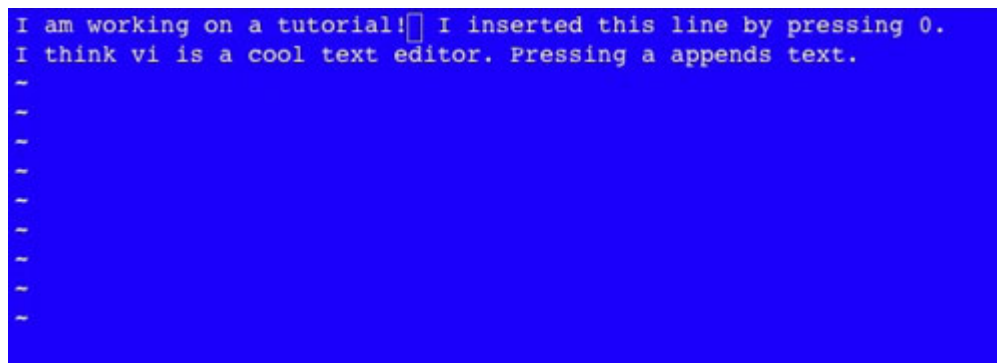
图 16. 要删除整行文本，可以按 dd



```
I am working on a tutorial!  
I inserted this line by pressing 0.  
I think vi is a cool text editor. Pressing a appends text.  
~  
~  
~  
~  
~  
~  
~  
~
```

谈到向上移动一行文本，您可以按 J 键以使用 join 命令，该命令将光标所在行的下面一行文本向上移动到光标所在行，而不会删除任何文本。您的光标应该位于文档的新行（第一行）。按 J 键将第二行上移至第一行。请参见图 17。

图 17. 要在 vi 中将两行文本合并在一起，可以按 J 键



```
I am working on a tutorial! I inserted this line by pressing 0.  
I think vi is a cool text editor. Pressing a appends text.  
~  
~  
~  
~  
~  
~  
~  
~
```

### 更好地完成其他操作

现在，您应该已经掌握了在 vi 中创建、导航和编辑文本文件的一些基本技能。在熟练掌握了这些基本的命令之后，您可以像在其他更加常规的文本编辑器中那样创建和编辑文件。但是您的指令表中还缺少了一些内容。本部分内容将向您介绍如何剪切、复制和粘贴。您可以一次反复地执行命令、再次执行命令、搜索文档、并使用撤销（undo）和重做（redo）命令。这个编辑器提供了其他文本编辑器中主要的功能，并且使用了快速访问的键盘命令方式。

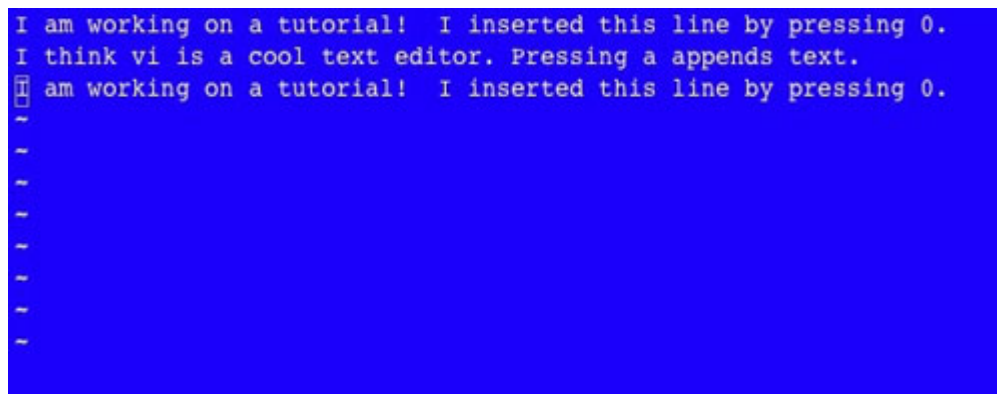
## 剪切、复制和粘贴

在 vi 中删除文本的同时，会将这些文本自动地保存到缓冲区中（类似于 Windows 中的剪贴板）。您已经了解了如何完成剪切命令，可以使用 `x`、`dd`、`dw` 和 `d$`。可以使用类似的命令来复制数据，而不删除该数据，在 vi 中，这种操作称为 *yank*：

- 按两次 `y` 键，可以复制整行文本。
- 按 `y` 键和 `w` 键，可以复制一个单词。
- 按 `y` 键和 `$` 键，可以从当前光标位置开始复制该行文本。

只有了解了如何粘贴数据，对数据进行复制才能真正派上用场。所以，在测试这些命令之前，您应该学习 `paste` 命令，可以使用 `p` 键来执行该命令。与 vi 中其他的许多命令一样，小写的 `p` 键将数据粘贴到光标位置之后，而大写的 `P` 键将数据粘贴到光标位置之前。要进行复制和粘贴，导航到文本文件的第一行，按两次 `y` 键。然后，将光标向下移动到第二行，按一次 `p` 键。该操作将第一行的文本复制到了第三行。请参见图 18。

图 18. 要复制一行文本，可以使用 `yy` 命令，并使用 `p` 命令进行粘贴

A screenshot of a vi editor window with a blue background and white text. The first line is "I am working on a tutorial! I inserted this line by pressing 0." The second line is "I think vi is a cool text editor. Pressing a appends text." The third line is "I am working on a tutorial! I inserted this line by pressing 0." The cursor is at the start of the third line. There are several tilde (~) characters below the third line, indicating more text in the file.

```
I am working on a tutorial! I inserted this line by pressing 0.
I think vi is a cool text editor. Pressing a appends text.
I am working on a tutorial! I inserted this line by pressing 0.
~
~
~
~
~
~
~
~
```

您可以尝试进行剪切和粘贴操作，将光标移动到第二行，按两次 `d` 键。然后，按 `p` 键将该行文本粘贴到第二行的下面。请参见图 19。

图 19. 要剪切并粘贴一行文本，可以使用 dd 命令和 p 命令

```
I am working on a tutorial! I inserted this line by pressing 0.  
I am working on a tutorial! I inserted this line by pressing 0.  
~ think vi is a cool text editor. Pressing a appends text.  
~  
~  
~  
~  
~  
~  
~  
~
```

### 在 vi 命令前面加上数字

此时，您可能为如何一次对多块数据执行这些命令而感到困惑。例如，您可能经常需要复制和粘贴整个段落，而不是单独的行。vi 允许您在使用每个命令时，在它们的前面加上一个数字，这样就会多次执行该命令。这种用法的功能非常强大，这也是使得 vi 能够成为专业用户手中优秀的编辑器的重要因素之一。要尝试一次剪切并粘贴两行文本，可以导航到文件的第一行，按 2 键，按两次 d 键，然后按 p 键。请参见图 20。

图 20. 可以在 dd 命令和 p 命令前面加上 2，一次剪切并粘贴两行文本

```
I think vi is a cool text editor. Pressing a appends text.  
~ am working on a tutorial! I inserted this line by pressing 0.  
I am working on a tutorial! I inserted this line by pressing 0.  
~  
~  
~  
~  
~  
~  
~  
~
```

在使用 paste 命令多次复制一行文本的过程中，也可以使用相同的概念。要完成这样的任务，可以先返回到第一行，使用 yy 命令复制该行文本，然后按 10，再按 p 键。现在将会增加 10 行关于 vi 的评价。在继续进行其他操作之前，按 5 键，然后按两次 d 键以删除其中一些多余的行。请参见图 21。

图 21. 在 p 命令前面加上数字 10, 可以粘贴多行文本, 然后在 dd 前面加上数字 5, 删除其中的几行文本

```

I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I am working on a tutorial! I inserted this line by pressing 0.
I am working on a tutorial! I inserted this line by pressing 0.
~
~
~
5 fewer lines

```

也可以尝试在导航命令的前面加上数字。例如, 按 **30** 键, 然后按 **1** 键, 这样可以将光标向右移动 30 个字符。按 **7** 键, 然后按 **G** 键, 这样可以将光标移动到文件的第七行。按 **5** 键, 然后按 **w** 键, 这样可以将光标移动到第五个单词。在完成了这些实验之后, 您可以继续进行下一步操作。

### 再次执行 vi 命令

vi 中另一个非常有用的命令是 **.** (句号) 键。**.** 键可以再次执行最后一条命令, 这个重要的特性可以让您快速地完成工作。例如, 将光标导航到第一行的单词 *cool*, 然后使用 **cw** 命令将这个单词更改为 *fast*。在输入完单词 *fast* 后, 不要忘记按 **Esc** 键。将光标向下移动到另一个 *cool* 单词, 然后按 **.** 键将这个单词更改为 *fast*。您还可以移动到文件中最后两行的单词 *line*, 并使用 **.** 键替换它们。请参见图 22。

图 22. 使用句号键再次执行命令, 这使得您可以快速地进行文档编辑

```

I think vi is a fast text editor. Pressing a appends text.
I think vi is a fast text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I am working on a tutorial! I inserted this fast by pressing 0.
I am working on a tutorial! I inserted this fast by pressing 0.
~
~
~

```



## 在 vi 中搜索文本

在 vi 中搜索文本也是非常快速且高效的。要开始搜索一个字符串,可以按 `/` 键(斜杠键),再加上希望搜索的字符串,然后按 **Enter** 键。要组合使用您的 vi 技能,可以按 `/` 键,输入 `think`,按 **Enter**,然后使用 `cw` 命令将这个单词更改为 `know`。在完成操作之后,不要忘记按 **Esc** 键。

如果您希望对第二行进行相同的操作,按 `n` 键以找到下一个 `think`,然后按 `.` 键将该单词更改为 `know`。`?` 键可以像 `/` 键那样进行搜索,但它将反向而不是正向搜索文档。在将 `think` 替换为 `know` 之后,按 `?` 键加上单词 `fast` 以便进行反向搜索。请参见图 23 和 24。

图 23. 使用斜杠键加上希望搜索的字符串,可以搜索该字符串

```
I know vi is a fast text editor. Pressing a appends text.
I know vi is a fast text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I am working on a tutorial! I inserted this fast by pressing 0.
I am working on a tutorial! I inserted this fast by pressing 0.
~
~
~
~/think
```

图 24. 使用 `?` 键加上希望搜索的字符串,可以反向搜索该字符串

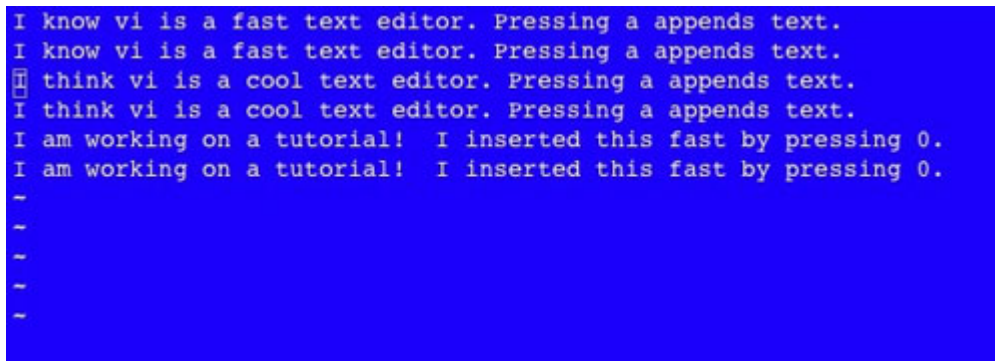
```
I know vi is a fast text editor. Pressing a appends text.
I know vi is a fast text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I am working on a tutorial! I inserted this fast by pressing 0.
I am working on a tutorial! I inserted this fast by pressing 0.
~
~
~
?fast
```

## vi 中的撤销 (undo) 和重做 (redo)

如果在操作中出现了错误，vi 为您提供撤销和重做机制，以确保可以将文档恢复到正确的状态。在命令模式中，按 **u** 键可以执行撤销命令，按 **Ctrl-r** 可以执行重做命令。可以尝试撤销和重做操作，如下所示（请参见图 25）：

1. 定位到文本文件的第三行，并删除一些行。
2. 按 **3** 键和 **G** 键定位到第三行。
3. 按 **2** 键和 **dd** 命令以删除两行文本。
4. 糟糕！不应该进行删除操作，并且您希望恢复这两行文本。要完成这个任务，可以按 **u** 键以撤销上一个命令。
5. 如果您又改变了主意，想要删除这两行文本，可以按 **Ctrl-r** 以重新执行该命令。

图 25. 要撤销一个命令，可以使用 **u** 键；要重新执行一个命令，可以按 **Ctrl-r**



```
I know vi is a fast text editor. Pressing a appends text.
I know vi is a fast text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I think vi is a cool text editor. Pressing a appends text.
I am working on a tutorial! I inserted this fast by pressing 0.
I am working on a tutorial! I inserted this fast by pressing 0.
~
~
~
~
~
```





本系列中后续的教程将介绍 Shell 的诀窍与技巧。同时，您应该不断地使用命令行并练习 vi，您将很快会成为一名专业的 UNIX 用户！