**保存文件并退出**

说起来有些惭愧，我也是最近才学到这个命令

: x

和下面的命令是等价的：

: wq

都是保存当前文件并退出。

（译者注：这两个命令实际上并不完全等价，当文件被修改时两个命令时相同的。但如果未被修改，使用 : x 不会更改文件的修改时间，而使用 :wq 会改变文件的修改时间。）

**基本计算器**

在插入模式下，你可以使用 Ctrl+r 键然后输入 =，再输入一个简单的算式。按 Enter 键，计算结果就会插入到文件中。例如，尝试输入：

Ctrl+r '=2+2' ENTER



然后计算结果“4 ”会被插入到文件中。

**查找重复的连续的单词**

当你很快地打字时，很有可能会连续输入同一个单词两次，就像 this this。这种错误可能骗过任何一个人，即使是你自己重新阅读一遍也不可避免。幸运的是，有一个简单的正则表达式可以用来预防这个错误。使用搜索命令（默认是 /）然后输入：

微博桌面截图_20150306094004

这会显示所有重复的单词。要达到最好的效果，不要忘记把下面的命令：

set hlsearch

放到你的 .vimrc 文件中高亮所有的匹配。



**缩写**

一个很可能是最令人印象深刻的窍门是你可以在 Vim 中定义缩写，它可以实时地把你输入的东西替换为另外的东西。语法格式如下：

:ab [缩写] [要替换的文字]

一个通用的例子是：

:ab asap as soon as possible

会把你输入的 “asap” 替换为 “as soon as possible”。

**在你忘记用 root 方式打开文件时的文件保存**

这可能是一个在论坛中一直受欢迎的命令。每当你打开一个你没有写入权限的文件（比如系统配置文件）并做了一些修改，Vim 无法通过普通的 “:w” 命令来保存。

你不需要重新以 root 方式打开文件再进行修改，只需要运行：

:w !sudo tee %

这会直接以 root 方式保存。

**实时加密文本**

如果你不想让别人看懂你的屏幕上的内容，你可以使用一个内置的选项，通过下面的命令使用 [ROT13](https://en.wikipedia.org/wiki/ROT13) 来对文本进行编码：

ggVGg?



gg 把光标移动到 Vim 缓冲区的第一行，V 进入可视模式，G 把光标移动到缓冲区的最后一行。因此，ggVG 使可视模式覆盖这个当前缓冲区。最后 g? 使用 ROT13 对整个区域进行编码。

注意它可以被映射到一个最常使用的键。它对字母符号也可以很好地工作。要对它进行撤销，最好的方法就是使用撤销命令：u。

**自动补全**

这是另外一个令我感到惭愧的功能，但我发现周围很多人并不知道。Vim 默认有自动补全的功能。的确这个功能是很基本的，并且可以通过插件来增强，但它也很有帮助。方法很简单。Vim 尝试通过已经输入的单词来预测单词的结尾。比如当你在同一个文件中第二次输入 “compiler” 时，仅仅输入 “com” 然后保持在插入模式，按 Ctrl+n 键就可以看到 Vim 为你补全了单词。很简单，但也很有用。

**比较两个文件的不同**

你们中的大多数很可能都知道 vimdiff 命令，它可以使用分离模式打开 Vim 并比较两个文件的不同。语法如下：

$ vimdiff [文件1] [文件2]

但同样的结果也可以通过下面的 Vim 命令来获得：

:diffthis

首先在 Vim 中打开原始文件。然后使用分离模式带来第二个文件：

:vsp [文件2]

最后在第一个缓冲区里输入：

:diffthis

通过 Ctrl+w 来切换缓冲区并再次输入：

:diffthis

这样两个文件中不同的部分就会被高亮。

（译者注：可以直接在一个缓冲区里使用命令 :windo diffthis，而不用输入 :diffthis 两次）

要停止比较，使用：

:diffoff

**按时间回退文件**

Vim 会记录文件的更改，你很容易可以回退到之前某个时间。该命令是相当直观的。比如：

:earlier 1m

会把文件回退到 1 分钟以前的状态。

注意，你可以使用下面的命令进行相反的转换：

:later

**删除标记内部的文字**

当我开始使用 Vim 时，一件我总是想很方便做的事情是如何轻松的删除方括号或圆括号里的内容。转到开始的标记，然后使用下面的语法：

di[标记]

比如，把光标放在开始的圆括号上，使用下面的命令来删除圆括号内的文字：

di(

如果是方括号或者是引号，则使用：

di{

和：

di"

**删除指定标记前的内容**

和删除标记内部有些相似，但目的不同。命令如下：

dt[标记]

会删除所有光标和标记之间的内容（保持标记不动），如果在同一行有这个标记的话。例如

dt.

会删除至句子的末尾，但保持 ‘.’ 不动。

**把 Vim 变为十六进制编辑器**

这不是我最喜欢的窍门，但有时会很有趣。你可以把 Vim 和 xxd 功能连起来来把文件转换为十六进制模式。命令如下：

:%!xxd



类似的，你可以通过下面的命令恢复原来的状态：

:%!xxd -r

**把光标下的文字置于屏幕中央**

我们所要做的事情如标题所示。如果你想强制滚动屏幕来把光标下的文字置于屏幕的中央，在可视模式中使用命令（译者注：在普通模式中也可以）：

zz

**跳到上一个／下一个位置**

当你编辑一个很大的文件时，经常要做的事是在某处进行修改，然后跳到另外一处。如果你想跳回之前修改的地方，使用命令：

Ctrl+o

来回到之前修改的地方

类似的：

Ctrl+i

会回退上面的跳动。

**把当前文件转化为网页**

这会生成一个 HTML 文件来显示文本，并在分开的窗口显示源代码：

:%TOhtml

（译者注：原文是 :%Tohtml，但在我的电脑上是 :%TOhtml）



很基本但很不错。

总的来说，这一系列命令是在我读了许多论坛主题和 [Vim Tips wiki](http://vim.wikia.com/wiki/Vim_Tips_Wiki)（如果你想学习更多关于编辑器的知识，我非常推荐这篇文章） 之后收集起来的。

如果你还知道哪些非常有用但你认为大多数人并不知道的命令，可以随意在评论中分享出来。就像引言中所说的，一个“鲜为人知但很有用的”命令也许只是你自己的看法，但分享出来总是好的。