编辑神器Vim新教程出炉，GitHub 3400星，复杂命令轻松搞定

机器之心 2020-09-07

机器之心报道

编辑：陈萍、杜伟

> 用聪明的方式打开 Vim，提高编辑效率。

提起文本编辑器，你一定会想到编辑器之神 Vim。作为一个快 30 岁的「老牌」编辑器，直到现在还很受欢迎。Vim 编辑文件非常高效，可支持多个操作系统，如 Unix/Linux、Windows、macOS，甚至 iOS 和 Android 上都能找到 Vim 的移植版本。

Vim 虽然非常强大，但是需要记住很多操作命令。如果没有有效的学习方法，操作起来很不方便，只有做到非常熟练才能感受到它带来的快捷。下图展示了 Vim 中关于删除、复制与粘贴的部分命令：

删除、复制与贴上	
x, X	在一行字当中，x 为向后删除一个字符 (相当于 [del] 按键)， X 为向前删除一个字符(相当于 [backspace] 亦即是退格键) (常用)
nx	n 为数字，连续向后删除 n 个字符。举例来说，我要连续删除 10 个字符，『10x』。
dd	删除游标所在的那一整行(常用)
ndd	n 为数字。删除光标所在的向下 n 行，例如 20dd 则是删除 20 行 (常用)
d1G	删除光标所在到第一行的所有数据
dG	删除光标所在到最后一行的所有数据
d$	删除游标所在处，到该行的最后一个字符
d0	那个是数字的 0 ，删除游标所在处，到该行的最前面一个字符
yy	复制游标所在的那一行(常用)
nyy	n 为数字。复制光标所在的向下 n 行，例如 20yy 则是复制 20 行(常用)
y1G	复制游标所在行到第一行的所有数据
yG	复制游标所在行到最后一行的所有数据
y0	复制光标所在的那个字符到该行行首的所有数据
y$	复制光标所在的那个字符到该行行尾的所有数据

其实 Vim 的操作命令还远不如此，如果没有很好的学习方法，可能很难使用这种编辑器。

那么有没有更好的方式来进行 Vim 学习呢?

近日，在 GitHub 上开源了一个用聪明的方式学习 **Vim** 的项目，上线短短几天，便收获了 3400 星。让我们来看看它是怎么做到聪明学习 Vim 的。

项目地址：https://github.com/iggredible/Learn-Vim

用聪明的方式打开 **Vim**

该项目提供了学习 Vim 的具体步骤，共分为 19 个章节。通过学习，你将掌握 vim 的语法、对文件的移动操作、撤销等等多种操作命令。

学习目录如下图所示：

这个「聪明」学习 Vim 到底体现在哪些方面呢？与别的 Vim 学习教程的不同之处又有哪些呢？接下来一一解析。

Vim 语法

在所有章节中，首先介绍一下 Vim 语法。一旦理解了 Vim 命令的语法结构，就可以与 Vim

进行「对话」了，就如想要学习一门语言，语法规则是绕不开的一步。而这个语法，就是「动词 + 名词」。现在使用基本的 Vim 动词和名词来建立词汇表。

Vim 名词与动词

Vim 中的名词（移动）：移动是在 Vim 中进行上下左右等操作。Vim 中的一些动作如下所示：

```
h：左
j：下
k：上
l：右
w：移动到下个单词的开头
}：跳到下一段
$：移动到本行末尾
```

Vim 中的动词（操作符）：以 h 操作符为例，Vim 中有 16 种。但只需掌握其中的 3 种，就能满足 80% 的编辑要求。

```
y     复制文本
d     删除文本，并保存到寄存器
c     删除文本，保存到寄存器，并开启「插入」模式
```

现在了解了 Vim 中基本的名词和动词，就可以根据语法规则组合上述名词和动词：

- y$：把当前所有的内容，从当前位置拖至行尾；
- dw：从当前位置删除到下一个单词的开头；

- c}：从当前位置到当前段落末尾进行更改。

当然移动也接受数字作为参数，如果你想上移 3 行，不需要键入 k 三次，直接使用 3k 就行了：

- y2h：向左移动 2 个字符；
- d2w：删除后面的 2 个单词；
- c2j：更改接下来 2 行。

除此以外，Vim 还允许通过键入操作符命令两次来执行行操作。例如：键入「dd」，删除整行；键入「cc」，更改整行内容。

Vim 可组合性和语法

在学习了 Vim 语法之后，下面介绍一下 Vim 中的可组合性。可组合性意味着拥有一组可以组合且执行更复杂命令的通用命令。

在 Vim 中可以组合较简单的命令执行复杂的命令。当 Vim 与外部程序集成时，可组合性的真正威力就显露出来了。

假如你有一个非常混乱的文本，如下所示，你想把它变成列表形式：

```
Id|Name|Cuteness
01|Puppy|Very
02|Kitten|Ok
03|Bunny|Ok
```

用 Vim 命令不容易做到，但是你可以用 column 终端命令快速完成。将光标放在「Id」上，

运行「!}column -t -s "|"」。就可以得到如下表格数据：

```
Id  Name    Cuteness
01  Puppy   Very
02  Kitten  Ok
03  Bunny   Ok
```

上述的操作过程可分解成这样：动词为「!」(过滤操作符)，名词为「}」(转到下一段)。过滤器操作符「!」接受另一个参数，一个终端命令「column -t -s "|"」。

假设你不仅希望将文本列表化，还想筛选出带有「Ok」的行。awk 可以轻松地完成这项任务。可以执行如下操作:

```
!}column -t -s "|" | awk 'NR > 1 && /Ok/ {print $0}'
```

得到的结果：

```
02  Kitten  Ok
03  Bunny   Ok
```

对操作符、动作和终端命令了解得越多，编写复杂动作的能力就会成倍增加。

基于 **Vim** 语法的更多操作符

在了解了 Vim 语法后，就可以完成更多的文件操作。例如第 5 章就介绍了对文件的操作。

对文件的操作，最基本的是对字符操作，字符操作的运动单元是左、下、上、右。

```
h    左
j    下
k    上
l    右
```

接下来转到一个更大的移动单元，词：

```
w      移到下一个单词的开头
e      移到下一个单词的末尾
b      移到前一个单词的开头
ge     移到前一个单词的末尾
```

词的操作介绍完，下面介绍一下对行的操作：

```
0      移动到当前行的第一个字符；
^      移动到当前行中的第一个非空字符；
g_     移动到当前行的最后一个非空白字符；
$      移动到当前行的最后一个字符；
n|     移动到当前行的第 n 列。
```

此外还有对句子和段落、匹配、行号、窗口等的操作，这里不再赘述。

该教程还在持续更新中，通过教程的学习，希望能更好地掌握 Vim 命令，并更快地进行代码

编辑。

© THE END

喜欢此内容的人还喜欢

日漫迷有福了！这个系统可以全自动翻译日漫，再也不用啃生肉了

机器之心

朋友圈超皮的句子，经典逗趣，其乐无穷！

毒鸡汤王

幼儿园期末评语，这样写有趣又有范！

幼师联盟官微