14. 交互式编辑和编辑历史

某些版本的 Python 解释器支持编辑当前输入行和编辑历史记录，类似 Korn shell 和 GNU Bash shell 的功能 。这个功能使用了 GNU Readline 来实现，一个支持多种编辑方式的库。这个库有它自己的文档，在这里我们就不重复说明了。

14.1. Tab 补全和编辑历史

在解释器启动的时候变量和模块名补全功能将 自动启用 以便在按下 Tab 键时唤起补全函数；它会查找 Python 语句名称、当前局部变量和可用的模块名称。 对于带点号的表达式如 `string.a`，它会对该表达式最后一个 `'.'` 之前的部分求值然后根据结果对象的属性给出补全建议。 请注意如果具有 `__getattr__()` 方法的对象是该表达式的一部分这可能会执行应用程序定义的代码。 默认配置还会将你的编辑历史保存到你的用户目录下名为 `.python_history` 的文件。 该历史在下一次交互式解释器会话期间将继续可用。

14.2. 默认交互式解释器的替代品

此功能相比较早版本的解释器是很大的进步；不过，还有一些需求没有实现：如果能为连续行提示正确的缩进就更好了（解析器知道接下来是否需要 `INDENT` 词元）。 补全机制或许可以使用解释器的符号表。 使用一个命令来检查（甚至提示）匹配括号、引号等也会很有帮助。

一个可选的增强型交互式解释器是 IPython，它已经存在了有一段时间，它具有 tab 补全，探索对象和高级历史记录管理功能。它还可以彻底定制并嵌入到其他应用程序中。另一个相似的增强型交互式环境是 bpython。