# 概述

笔者编写iRepl的初衷有两个：一是为了学习熟练Clojure编程；二是想自己编写一个可以提供实用功能的称手的工具，这个工具应当包含以下功能：

* 能够调用系统程序
* 能够像REPL一样被使用
* 能够注册、管理各个工具程序

Clojure本身的REPL就是一个很强大灵活的工具，但具有一定的局限性。例如，它不提供与系统（OS命令及其shell命令）的直接交互，譬如在REPL中就不能直接调用“dir”（或者在Linux下就是“ls”）；又譬如，REPL虽然提供了灵活的编程能力，但是它并不具有工具管理的功能……简而言之，笔者是想把能想到的功能都放进同一个工具里面，而且这个工具还是高度可配置的。

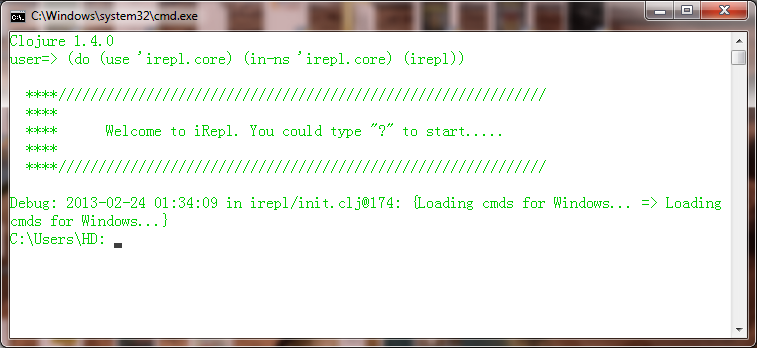
基于这个“美好的”想法，笔者开始了尝试编写iRepl之旅（注：目前为止，该程序只在开始阶段，很多东西都还没落实——但这不影响我们对它开展的各种尝试）。

## 启动iRepl

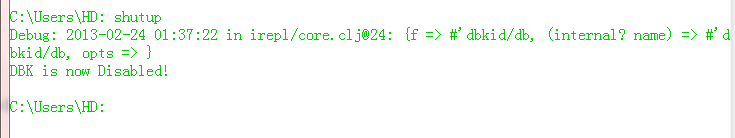
要启动iRepl，首先需要保证iRepl文件夹（及其文件）都在系统的CLASSPATH中，然后：

1. 启动REPL
2. 在REPL中用“(do (use 'irepl.core) (in-ns 'irepl.core) (irepl))”来启动iRepl

启动之后，我们可以看到类似于以下的界面：



题为“Debug: ……”的那句是因为默认开启了调试模式，在这个模式中，每当运行到使用了调试工具的地方，都会打印出相关信息，该功能目前不是很成熟而且对非开发者没有用处，因此我们可以使用“shutup”命令来关闭这个模式：

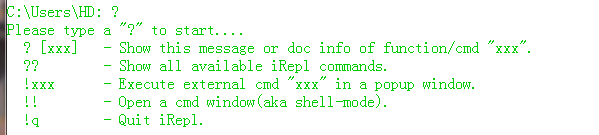


在看到“DBK is now Disabled!”之后，就说明该模式就被禁掉了（注：*如果要重新启用调试模式，我们可以再次使用“shutup”命令*）。

就像启动时欢迎标语所说，在启动iRepl之后，我们可以直接使用“?”命令来开始探索我们的iRepl。

## “?”——iRepl的帮助命令

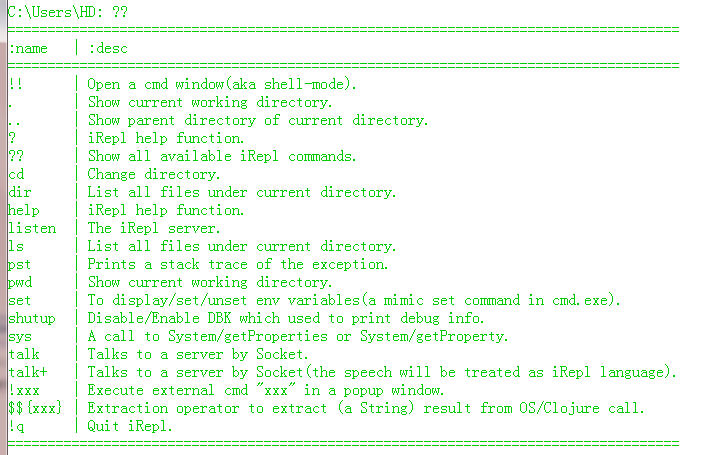
“?”实际上是“help”命令的别名（命令的别名通常是为了便于记忆和减少敲键盘的次数。更多关于命令的别名，请参考后文），它是iRepl的帮助命令：



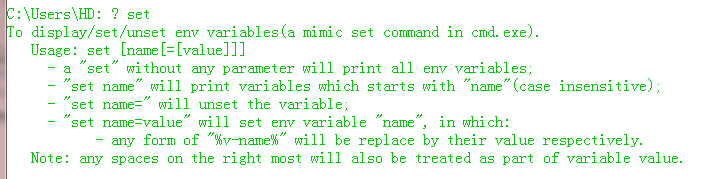
不带参数的帮助命令将显示命令本身的说明，以及其它几个主要的具有指导作用的命令。注意其中带有中括号的如“? [xxx]”命令格式表示“?”命令后面的参数xxx是可选的，因此，使用“?”和“? date”都是合法的，但是其显示结果将有所不同。

## “??”——列出已有命令

双问号“??”命令将显示所有已注册的iRepl命令以及关于它们的简短说明（iRepl命令指的是iRepl本身提供的、以及用户自己加载、注册的命令，但不包括系统命令），在查看系统中当前的iRepl提供什么功能的时候，这是一个方便的命令：



“??”常常与“?”一起使用，因为“?”能提供更多的文档说明：



## “!”——执行系统命令或进入shell模式

像其它常见工具一样，我们使用“!”来指示iRepl执行系统命令，如“!date”（或者是包含空格的“! date”）。

有时候可能会遇到需要执行多个系统命令的情况，因此我们提供了shell模式，使用命令“!!”可以让iRepl直接打开一个shell窗口，这样子就可以像平常使用shell一样执行系统命令了——直到用户关闭该shell为止。不过需要注意的是，每次使用shell模式打开的shell都其环境都是全新的——它并不能继承在iRepl中所设置的环境变量（关于在iRepl中设置环境变量，请参考后文）。

## 退出iRepl

我们使用命令“!q”来退出iRepl。

## 更多……

现在，更有趣的地方来了：我们可以像使用REPL一样使用iRepl：

