



# IPython: Beyond Normal Python

## IPython：超越Python解释器

There are many options for development environments for Python, and I'm often asked which one I use in my own work. My answer sometimes surprises people: my preferred environment is [IPython](#) plus a text editor (in my case, Emacs or Atom depending on my mood). IPython (short for *Interactive Python*) was started in 2001 by Fernando Perez as an enhanced Python interpreter, and has since grown into a project aiming to provide, in Perez's words, "Tools for the entire life cycle of research computing." If Python is the engine of our data science task, you might think of IPython as the interactive control panel.

对于Python来说，开发环境有很多种选择，作者经常会被问到他在自己工作中使用哪一个。答案有时会让提问者感到吃惊：作者习惯的环境是[IPython](#) 在加上一个文本编辑器（取决于作者的心情，可能是Emacs或者Atom）。IPython（是交互式Python的缩写）是Fernando Perez在2001年创建的一个增强Python解释器的项目，目前已经发展成为一个超级工具，目标是提供（用Perez自己的话来说）“研究计算领域完整生命周期的工具”。如果类比Python是我们数据科学的引擎的话，那么你可以认为IPython就是一个交互式的控制面板。

As well as being a useful interactive interface to Python, IPython also provides a number of useful syntactic additions to the language; we'll cover the most useful of these additions here. In addition, IPython is closely tied with the [Jupyter project](#), which provides a browser-based notebook that is useful for development, collaboration, sharing, and even publication of data science results. The IPython notebook is actually a special case of the broader Jupyter notebook structure, which encompasses notebooks for Julia, R, and other programming languages. As an example of the usefulness of the notebook format, look no further than the page you are reading: the entire manuscript for this book was composed as a set of IPython notebooks.

除了提供Python十分方便的交互式界面外，IPython还提供了一些语言的语法扩展；我们会在这里介绍其中最有用的扩展内容。并且，IPython紧密的绑定在[Jupyter项目](#)之上，Jupyter能够提供一个浏览器界面的笔记本（译者注：后续一律称为notebook，不再翻译该名词），能够非常方便的开发、合作、分享甚至发布数据科学的结果。IPython notebook是Jupyter这个庞大项目中的Python部分，后者希望为Julia、R和其他编程语言都能提供notebook功能。本书就可以作为notebook格式的一个有力证明：所有本书的内容都是使用IPython notebook编写的。

IPython is about using Python effectively for interactive scientific and data-intensive computing. This chapter will start by stepping through some of the IPython features that are useful to the practice of data science, focusing especially on the syntax it offers beyond the standard features of Python. Next, we will go into a bit more depth on some of the more useful "magic commands" that can speed-up common tasks in creating and using data science code. Finally, we will touch on some of the features of the notebook that make it useful in understanding data and sharing results.

IPython的目标是让科学和数据计算在Python中更加有效和具有交互性。本章将会介绍许多对于数据科学实践来说非常有用的IPython特性，尤其聚焦在它提供在Python标准之外的语法扩展。然后，我们将会进一步深入到一些有用的“魔法命令”中，这些命令能够提高你在编写和使用数据科学代码的时候的生产效率。最后，我们将接触到在notebook当中有用的数据分析和分享结果的特性。

### Shell or Notebook?

### Shell 还是 Notebook?

There are two primary means of using IPython that we'll discuss in this chapter: the IPython shell and the IPython notebook. The bulk of the material in this chapter is relevant to both, and the examples will switch between them depending on what is most convenient. In the few sections that are relevant to just one or the other, we will explicitly state that fact. Before we start, some words on how to launch the IPython shell and IPython notebook.

IPython提供了两种主要的使用方法，Shell和Notebook。本章都会使用到，例子将会根据最方便的方式切换。如果某些小节只会使用到其中一个，作者会明确指出。在我们开始之前，我们先来学习如何启动shell和notebook。

### Launching the IPython Shell

#### 启动IPython Shell

This chapter, like most of this book, is not designed to be absorbed passively. I recommend that as you read through it, you follow along and experiment with the tools and syntax we cover: the muscle-memory you build through doing this will be far more useful than the simple act of reading about it. Start by launching the IPython interpreter by typing `ipython` on the command-line; alternatively, if you've installed a distribution like Anaconda or EPD, there may be a launcher specific to your system (we'll discuss this more fully in [Help and Documentation in IPython](#)).

本章和本书大多数章节一样，不是希望你只是被动学习。作者建议当你阅读的时候，能够跟着内容进行实践，对我们介绍的工具和语法进行试验：在此过程中你将会获得肌肉记忆，这远比简单的阅读牢靠的多。启动IPython解释器，你只需要在命令行上输入 `ipython` 即可；同样的，如果你安装的是Anaconda或者EPD这样的发行版，那么你的操作系统上可能会有相应的启动图标（我们会在[IPython帮助和文档](#)中更详细的讨论）。

Once you do this, you should see a prompt like the following:

```
IPython 4.0.1 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.
In [1]:
```

With that, you're ready to follow along.

当你输入命令后，你应该会看到如下的一个提示符：

```
IPython 4.0.1 -- An enhanced Interactive Python.
?          -> Introduction and overview of IPython's features.
%quickref  -> Quick reference.
help       -> Python's own help system.
object?    -> Details about 'object', use 'object??' for extra details.
In [1]:
```

然后，你就可以接着往下进行了。

### Launching the Jupyter Notebook

#### 启动Jupyter Notebook

The Jupyter notebook is a browser-based graphical interface to the IPython shell, and builds on it a rich set of dynamic display capabilities. As well as executing Python/IPython statements, the notebook allows the user to include formatted text, static and dynamic visualizations, mathematical equations, JavaScript widgets, and much more. Furthermore, these documents can be saved in a way that lets other people open them and execute the code on their own systems.

Jupyter notebook是一个浏览器图形界面的IPython shell，上面构建了一整套丰富的动态展示功能。除了能够执行Python/IPython代码，notebook还允许用户书写格式化的文本，静态和动态的展示数学公式，JavaScript组件和其他很多功能。然后这些文档能被保存成一种能让其他人在他们自己的系统中打开和执行的文件格式。

Though the IPython notebook is viewed and edited through your web browser window, it must connect to a running Python process in order to execute code. This process (known as a "kernel") can be started by running the following command in your system shell:

```
$ jupyter notebook
```

虽然IPython notebook在你的浏览器窗口中展示和编辑，但是它需要连接到一个执行中的Python进程才能真正执行代码。这个进程（被称为“kernel”）可以在命令行中使用下面的命令启动：

```
$ jupyter notebook
```

This command will launch a local web server that will be visible to your browser. It immediately spits out a log showing what it is doing; that log will look something like this:

```
$ jupyter notebook
[NotebookApp] Serving notebooks from local directory: /Users/jakevdp/PythonDataScienceHandbook
[NotebookApp] 0 active kernels
[NotebookApp] The IPython Notebook is running at: http://localhost:8888/
[NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confi
rmation).
```

这个命令会启动一个web服务器让你的浏览器访问。它会立刻在你的控制台打印出日志，日志的格式类似下面：

```
$ jupyter notebook
[NotebookApp] Serving notebooks from local directory: /Users/jakevdp/PythonDataScienceHandbook
[NotebookApp] 0 active kernels
[NotebookApp] The IPython Notebook is running at: http://localhost:8888/
[NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confi
rmation).
```

Upon issuing the command, your default browser should automatically open and navigate to the listed local URL; the exact address will depend on your system. If the browser does not open automatically, you can open a window and manually open this address (<http://localhost:8888/> in this example).

输入上述命令之后，你系统的默认浏览器应该会自动打开然后指向本地的地址；完整的地址取决于你的系统。如果你的浏览器没有自动打开，你可以手动打开它并输入地址(本例中是<http://localhost:8888/>)。

