第一章：Python简介与环境安装

Table of Contents

[1. 序言 2](#_Toc49592097)

[1.1 章节简介 2](#_Toc49592098)

[2. Python简介 2](#_Toc49592099)

[2.1 编程语言是什么？ 2](#_Toc49592100)

[2.2 Python是什么？ 3](#_Toc49592101)

[2.3 Python的历史 3](#_Toc49592102)

[2.4 Python的优点 4](#_Toc49592103)

[2.5 Python可以做什么？ 6](#_Toc49592104)

[3. 开发环境安装 6](#_Toc49592105)

[3.1 安装Python 3.6 6](#_Toc49592106)

[3.2 在Windows上安装Python 7](#_Toc49592107)

[3.3 验证Python正确安装 7](#_Toc49592108)

[3.4 Windows环境安装Jupyter学习环境 8](#_Toc49592109)

[3.5 Jupyter的使用 9](#_Toc49592110)

# 序言

## 章节简介

本章节帮助没有编程基础的同学了解Python的背景以及Python开发环境的安装，为后续章节的学习做好准备。课程目标为：

a. 什么是编程语言

b.为什么学习Python

c. Python安装流程

d. 学习中使用的开发工具jupyter notebook

# Python简介

## 编程语言是什么？

编程语言（programming language）可以简单的理解为一种计算机和人都能识别的语言。一种计算机语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。

我们通过编程语言来编写程序代码，再通过语言处理程序执行向计算机发送指令，让计算机完成对应的工作。简单来说，编程语言就是人类指挥计算机进行工作的语言，是人类与计算机沟通的语言。

## Python是什么？

Python也是编程语言的一种，并且是高级的编程语言。Python语言可能是第一种即简单又功能强大的编程语言。它不仅适合于初学者，也适合专业人员使用，更加重要的是，用Python编程是一种愉快的事。本书将帮助你学习这个奇妙的语言，并且向你展示如何即快捷又方便地完成任务。

## Python的历史

Python的作者，**Guido von Rossum**，荷兰人。1982年，Guido从阿姆斯特丹大学获得了数学和计算机硕士学位。然而，尽管他算得上是一位数学家，但他更加享受计算机带来的乐趣。他的话说，尽管拥有数学和计算机双料资质，他总趋向于做计算机相关的工作，并热裏于做任何和编程相关的活动。在那个时候，Guido接触并使用过诸如*Pascal*、*C*、*Fortran*等语言。这些语言的基本设计原则是让机器能更快运行。在80年代，虽然IBM和苹果已经掀起了个人电脑浪潮，但这些个人电脑的配置很低。比如早期的Macintosh，只有8MHz的CPU主频和128KB的RAM，一个大的数组就能占满内存。所有的编译器的核心是做优化，以便让程序能够运行。为了堆进效率，语言也迫使程序员像计算机一样思考，以便能写出更符合机器囗味的程序。在那个时代，程序员恨不得榨取计算机每一滴的能力。有人甚至认为c语言的指针是在浪费内存。至于动态型，内存自动管理，面向对象……别想了，那会让你的电脑陷入瘫痪。

这种编程方式让Guido感到苦恼。Guido知道如何用c语言写出一个功能，但整个编写过程需要耗费大量的时间，即使他已经准确的知道了如何实现。他的另一个选择是shell。Bourne Shell作为UNIX系统的解释器已经长期存在。UNIX的管理员们常常用shell去写一些简单的脚本，以进行一些系统维护的工作，比如定期备份、文件系统管理等等。shell可以像胶水一样，将UNIX下的许功能连接在一起。许多C语言下上百行的程序，在shell下只用几行就可以完成。然而，shell的本质是调用命令。它并不是一个真正的语言。比如说，shell没有数值型的数据类型，加法运算都很复杂。总之，shell不能全面的调动计算机的功能。

Guido希望有一种语言，这种语言能够像C语言那样，能够全面调用计算机的功能接口，又可以像shell那样，可以轻松的编程。ABC语言让Guido看到希望，ABC是由荷兰的数学和计算机研究所开发的。Guido在该研究所工作，并参与到A语言的开发。ABC语言以教学为目的。与当时的大部分语言不同，ABC语言的目标是"让户感觉更好"。ABC语言希望让语言变得容易阅读，容易使用，容易记忆，容易学习，并以此来激发人们学习编程的兴趣。

**一门语言的诞生**：1991年，第一个Python编译器诞生。它是C语言实现的，并能够调用C语言的库文件。从一出生，Python已经具有了：类，函数，异常处理，包含表和词典在内的核心数据类型，以及模块为基础的拓展系统。Python语法很多来自C，但又受到ABC语言的强烈影响。来自ABC语言的一些规定直到今天还富有争议，比如强制缩进。但这些语法规定让Python容易读。另一方面，Python聪明的选择服从一些惯例，特别是C语言的惯例，比如回归等号赋值。Guido认为，如果常识上确立的东西，没有必要过度纠结。Python从一开始就特别在意可拓展性。Python可以在多个层次上拓展。从高层上，你可以直接引入py文件。在底层，你可以引用c语言的库。Python程序员可以快速的使用Python的.py文件作为拓展模块。但当性能是考虑的重要因素时，Python程序员可以深入底层，写c程序，编译为so文件引入到Python中使用。Python就好像是使钢构建房一样，先规定好大的框架。而程序员可以在此框架下相当自由的拓展或更改。最初的Python完全庄Guido本人开发。Python得到Guido同事的欢迎。他们迅速的反馈使用意见，并参与到Python的改进。Guido和一些同事组成Python的核心团队。他们将自己大部分的业余时间用于hack Python。随后，Python拓展到研究所之外。Python将许机器层面上的细节隐藏，交给编译器处理，并凸显出逻辑层面的编程思考。Python程序员可以花更的时间于思考程序的逻辑，而不是具体的实现细节。这一特征吸引了广大的程序员。Python开始流行。

## Python的优点

1. **简单**：Python是一种代表简单主义思想的语言。阅读一个良好的Python程序就感觉是在读英语一样，尽管这个英语的要求非常严格！Python的这种伪代码本质是它最大的优点之一。它使你能够专注于解决问题而不是去搞明白语言本身。
2. **易学**：就如同你即将看到的一样，Python极其容易上手。前面已经提到了，Python有极其简单的语法。
3. **免费、开源**：Python是FLOSS（自由/开放源码软件）之一。简单地说，你可以自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中。FLOSS是基于一个团体一它是由一群希望看到一个更加优秀的分享知识的概念。这是为什么Python如此优秀的原因之一—它是由一群希望看到一个更加优秀的Python的人创造并经常改进着的。
4. **层语言**：当你用Python语言编写程序的时候，你无需考虑诸如如何管理你的程序使用的内存一类的底层细节。
5. **可移植性**：由于它的开源本质，Python已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。如果你小心地避免使用依赖于系统的特性，那么你的所有Python程序无需修改就可以在下述任何平台上面运行。这些平台包括Linux、Windows、FreeBSD、Macintosh、Solaris、Google基于Linux开发的Android平台！
6. **解释性**：这一点需要一些解释。一个用编译性语言比如C或C++写的程序可以从源文件（即C或C++语言）转换到一个你的计算机使用的语言（二进制代码，即0和1）。这个过程通过编译器和不同的标记、选项完成。当你运行你的程序的时候，连接/转载器软件把你的程序从硬盘复制到内存中并且运行。而Python语言写的程序不需要编译成二进制代码。你可以直接从源代码运行程序。在计算机内部，Python解释器把源代码转换成称为字节码的中间形式，然后再把它翻译成计算机使的机器语言并运行。事实上，由于你不再需要担心如何编译程序，如何确保连接转载正确的库等等，所有这一切使得使用Python更加简单。由于你只需要把你的Python程序拷贝到另外一台计算机上，它就可以工作了，这也使得你的Python程序更加易于移植。
7. **面向对象**：Python既支持面向过程的编程也支持面向对象的编程。在“面向过程“的语言中，程序是由过程或仅仅是可重用代码的函数构建起来的。在"面向对象"的语言中，程序是由数据和功能组合而成的对象构建起来的。与其他主要的语言如C++和Java相比，Python以一种非常强大又简单的方式实现面向对象编程。
8. **可扩展性**：如果你需要你的一段关键代码运行得更快或者希望某些算法不公开，你可以把你的部分程序C或C++编写，然后在你的Python程序中使用它们。
9. **丰富的库**：Python标准库确实很庞大。它可以帮助你处理各种工作，包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV文件、密码系统、GUI（图形用户界面）、Tk和其他与系统有关的操作。记住，只要安装了Python，所有这些功能都是可用的。这被称作Python的"功能齐全"理念。賒了柝准库以外，还有许多其他高质量的库，如scikit-learn、TensorFlow和Python图像库等等。
10. **规范的代码**：Python采用强制缩进的方式使得代码具有极佳的可读性。

## Python可以做什么？

1. **web应用开发**：服务器端的编程，具有很丰富的web开发框架，如Django，能够快速完成一个网站的开发和web服务，比如Google、豆瓣等。
2. **数据分析**：Python被广泛的应用到科学与数字分析中，比如图像可视化分析，生物信息学，常用的是Numpy、pandas等。
3. **系统网络运维**：在运维的工作中，有大量重复性的工作，并需要管理，监控，发布系统等，将工作自动化起来，提高工作效率，这样的场景非常适合Python。
4. **界面开发**：Python在图形界面开发上很强大，可以用tkinter/PyQT框架开发各种桌面软件！
5. **爬虫**：在爬虫领域，Python几乎是霸主地位，将网络一切数据作为资源，通过自动化程序进行有针对性的数据采集以及处理。从事该领域应学习爬虫策略、高性能异IO、分布式爬虫等，并针对Scrapy框架源码进行深入剖析，从而理解其原理并实现自定义爬虫框架。
6. **人工智能**：MASA和Google早期大量使用Python，为Python积累了丰富的科学运算库，当AI时代来临后，Python从众多编程语言中脱颖而出，各种人工智能算法都基于Python编写，尤其PyTorch之后，Python作为AI时代头牌语言的位置基本确定。

# 开发环境安装

Python是一种跨平台的语言，可以运行在Windows、Mac和Linux系统上。也就是说我们在Windows写的Python程序，放在同样安装了Python的Mac电脑上也是可以运行的。要学习Python的语法并使用Python，就需要我们安装Python和对应的包。通过本小节的学习，我们会一起安装Python，并得到一个相对简单好用的编程调试环境。

## 安装Python 3.6

目前Python的所有可下载使用版本，可以在这个链接查看[Python官方下载页](https://www.python.org/downloads/)。我们使用稳定版本的3.6作为Python学习的环境，Mac用户可以通过下载[Mac用户Python下载地址](https://www.python.org/downloads/release/python-3612/)下载对应的Python进行安装，如果你使用过Linux并有一定Linux系统管理经验，自行安装Python3应该没有问题，对于不熟悉Mac OS及Linux的同学，请切换到Windows环境下，让我们一起来安装Python。

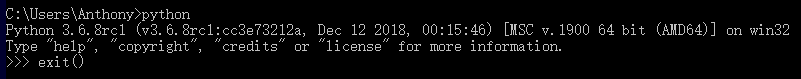
## 在Windows上安装Python

请根据自己的Windows操作系统版本选择64位[64位](https://www.python.org/ftp/python/3.6.8/python-3.6.8-amd64.exe)或者[32位](https://www.python.org/ftp/python/3.6.8/python-3.6.8.exe)的Python安装程序下载，然后双击运行exe安装包，请注意要勾选*Add Python 3.6 to path*，然后点击install按钮即可。



## 验证Python正确安装

安装Python后，用CMD打开命令窗，在提示符处输入python并回车，如果看到下面情况，就说明Python已经正确安装。如果你看到>>>的提示符，说明已经进入了Python的交互环境并进行编程，我们可以编写Python程序并回车作为代码的编写得到结果，这里我们先不学习具体内容，通过输入exit();退出交互环境。



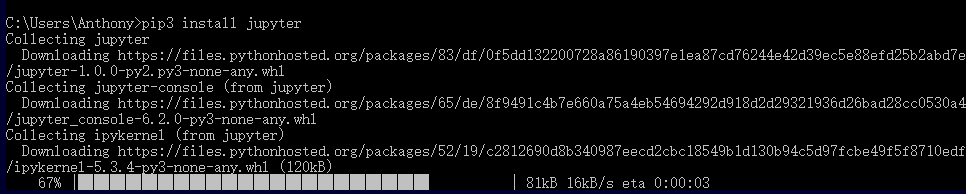
## Windows环境安装Jupyter学习环境

Jupyter是什么？官网的介绍是：Jupyter Notebook是一个Web应用程序，允许您创建和共享包含实时代码，方程，可视化和说明文本的文档。 用途包括：数据清理和转换，数值模拟，统计建模，机器学习等等。

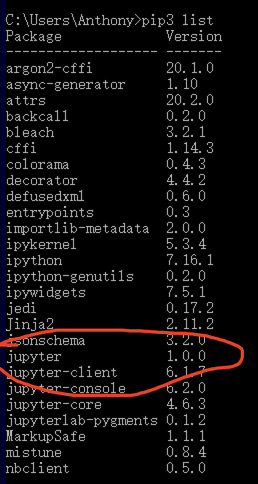
用大家更容易理解的介绍就是：Jupyter Notebook是我们之前看到的使用命令行交互Ipython的升级版，而Ipython可以说是一个加强版的交互式 Shell。也就是说，它比在terminal里运行python会更方便，界面更友好，功能也更强大，也更有利我们学习。如果说刚才我们安装的Python包是Python语言的内核的话，Jupyter就是一个让我们使用更友好的交互界面方便我们学习研究。

jupyter notebook的安装和打开。安装非常简单，只需要在CMD终端输入：

*pip3 install jupyter*

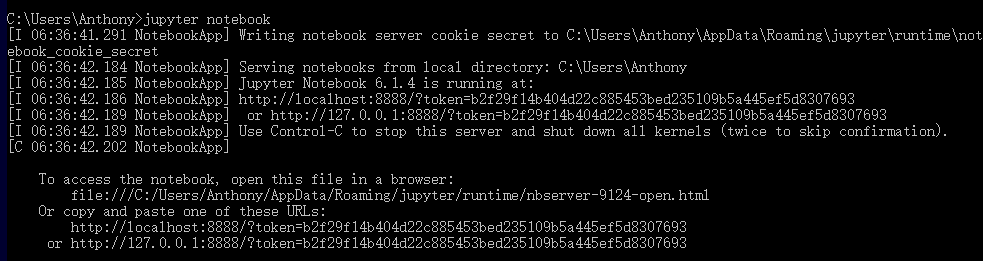
**

等到安装结束，让我们键入***pip3 list***确认Jupyter正确安装(pip3可以理解为安装Python的安装助手，在我们安装了Python后，帮助Python安装管理更新各种需要的模块和资源)



## Jupyter的使用

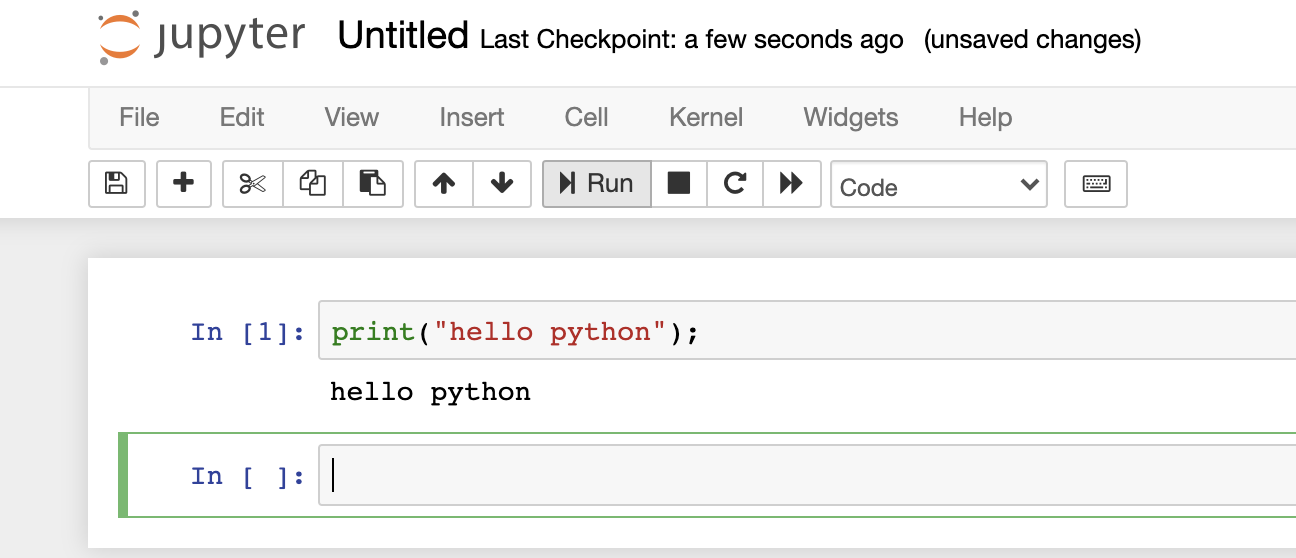
成功安装Jupyter后，让我们启动并使用。输入命令*jupyter notebook*启动Jupyter编辑器。



打开浏览器，在浏览器地址栏中输入*<http://localhost:8888>* 我们就打开了Jupyter编辑器。点击*New* 下面*Python 3*让我们新建一个Python文件。



在新打开的Python3文件中，让我们输入一行Python命令*print(“hello python”);* 然后点击Run按钮来检查它的输出。(这行代码的作用是输出一行字符串hello pyhon)



至此我们已经顺利的可以通过使用Jupyter来进行Python的学习了，更多关于如何发挥jupyter强大功能的详细，请参阅[如何优雅地使用 Jupyter？](https://www.zhihu.com/question/59392251) 后续的学习中，我们将会通过Jupyter来编写Python代码进行学习。

## 执行Python文件

Jupyter的特点是可以编辑并执行整个文件的一部分代码，它的主要的用途是用来学习及科学研究，另外一种很常见的执行Python程序的方式是通过Python解释器来直接执行Python程序。当我们编写Python代码时，我们得到的是一个包含Python代码的以.py为扩展名的文本文件。要运行代码，就需要Python解释器去执行.py文件。

让我们看个示例，先用文本编辑工具（如“记事本”）来写一小段Python程序如下：

*print (“Hello Python!”)*

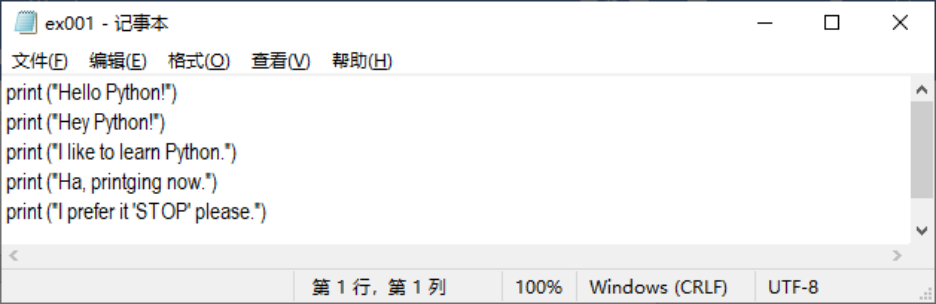
*print (“Hey Python!”)*

*print (“I like to learn Python.”)*

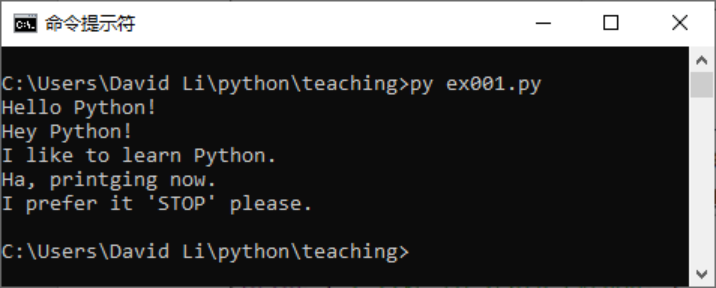
*print (“Ha, printing now.”)*

*print (“I prefer it ‘STOP’ please.”)*

把这段小程序存在电脑上，起个名字叫ex001.py。然后在命令行终端上运行它：python3 ex001.py。



在CMD终端里(Mac及Linux环境里使用shell)运行这个小程序，得到输出结果：



本教程的内容中会交替使用两种方式调用程序执行。