优点  
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  
觉像是在读英语一样。它使你能够专注于解决问题而不是去搞明白语言本身。  
易学：Python 极其容易上手，因为Python 有极其简单的说明文档。  
速度快：Python 的底层是用 C 语言写的，很多标准库和第三方库也都是用 C 写  
的，运行速度非常快。免费、开源：Python 是 FLOSS（自由/开放源码软件）之  
一。使用者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是基于一个团体分享知识的概念。  
高层语言：用 Python 语言编写程序的时候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。  
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）。这些平台包括Linux、Windows、FreeBSD、Macintosh、  
Solaris、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、PalmOS、QNX、  
VMS、Psion、AcomRISCOS、VxWorks、PlayStation、SharpZaurus、WindowsCE、  
PocketPC、Symbian 以及Google基于linux 开发的android平台。  
解释性：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。  
运行程序的时候，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  
它翻译成计算机使用的机器语言并运行。这使得使用 Python 更加简单。也使得  
Python 程序更加易于移植。  
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。  
可扩展性：如果需要一段关键代码运行得更快或者希望某些算法不公开，可以部  
分程序用C 或C++编写，然后在Python 程序中使用它们。

可嵌入性：可以把Python 嵌入C/C++程序，从而向程序用户提供脚本功能。  
丰富的库：Python 标准库确实很庞大。它可以帮助处理各种工作，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。这被称作 Python 的“功能齐全”理念。除了标准库  
以外，还有许多其他高质量的库，如wxPython、Twisted和Python 图像库等等。  
规范的代码：Python 采用强制缩进的方式使得代码具有较好可读性。而 Python  
语言写的程序不需要编译成二进制代码。  
缺点  
单行语句和命令行输出问题：很多时候不能将程序连写成一行，如importsys;fori  
insys.path:printi。而 perl 和 awk 就无此限制，可以较为方便的在 shell 下完成简  
单程序，不需要如Python 一样，必须将程序写入一个.py文件。  
独特的语法  
这也许不应该被称为局限，但是它用缩进来区分语句关系的方式还是给很多初学  
者带来了困惑。即便是很有经验的Python 程序员，也可能陷入陷阱当中。  
运行速度慢：这里是指与C 和C++相比。