Python 应用领域   
  
　　Python 不仅仅是一个设计优秀的程序语言，它能够完成现实中的各种任务，包括开发者们日复一日所做的事情。作为编制其他组件、实现独立程序的工具，它通常应用于各种领域。实际上，作为一种通用语言，Python 的应用角色几乎是无限的：你可以在任何场合应用Python, 从网站和游戏开发到机器人和航天飞机控制。   
　　尽管如此，Python 的应用领域分为如下几类。下文将介绍一些Python 如今最常见的应用领域，以及每个应用领域内所用的一些工具。我们不会对各个工具进行深入探讨，如果你对这些话题感兴趣，请从Python 网站或其他一些资源中获取更多的信息。   
　　**系统编程**   
　　Python 对操作系统服务的内置接口，使其成为编写可移植的维护操作系统的管理工具和部件（有时也被称为Shell 工具）的理想工具。Python 程序可以搜索文件和目录树，可以运行其他程序，用进程或线程进行并行处理等等。   
　　Python 的标准库绑定了POSIX 以及其他常规操作系统（OS ）工具：环境变量、文件、套接字、管道、进程、多线程、正则表达式模式匹配、命令行参数、标准流接口、Shell 命令启动器、文件名扩展等。此外，很多Python 的系统工具设计时都考虑了其可移植性。例如，复制目录树的脚本无需做任何修改就可以在几乎所有的Python 平台上运行。   
　　**用户图形接口**   
　　Python 的简洁以及快速的开发周期十分适合开发GUI 程序。Python 内置了TKinter 的标准面向对象接口Tk GUI API ，使Python 程序可以生成可移植的本地观感的GUI 。Python/ Tkinter GUI 不做任何改变就可以运行在微软Windows 、X Windows（UNIX 和Linux ）以及Mac OS（Classic 和OS X 都支持）等平台上。一个免费的扩展包PMW ，为Tkinter 工具包增加了一些高级部件。此外，基于C++ 平台的工具包wxPython GUI API 可以使用Python 构建可移植的GUI 。   
　　诸如PythonCard 和Dabo 等一些高级工具包是构建在wxPython 和Tkinter 的基础API 之上的。通过适当的库，你可以使用其他的GUI 工具包，例如，Qt 、GTK 、MFC 和Swing 等。对于运行于浏览器中的应用或在一些简单界面的需求驱动下，Jython（Java 版本的Python ，我们将会在第2章中进行介绍）和Python 服务器端CGI 脚本提供了其他一些用户界面的选择。   
　　**Internet 脚本**   
　　Python 提供了标准Internet 模块，使Python 能够广泛地在多种网络任务中发挥作用，无论是在服务器端还是在客户端都是如此。脚本可以通过套接字进行通信；从发给服务器端的CGI 脚本的表单中解析信息；通过URL 获取网页；从获取的网页中解析HTML 和XML 文件；通过XML-RPC 、SOAP 和Telnet 通信等。Python 的库使这一切变得相当简单。   
　　不仅如此，从网络上还可以获得很多使用Python 进行Internet 编程的第三方工具。例如，HTMLGen 可以从Python 类的描述中生成HTML 文件，mod\_python 包可以使在Apache 服务器上运行的Python 程序更具效率并支持Python Server Page 这样的服务器端模板，而且支持客户端运行的服务器端Applet 。此外，Python 涌现了许多Web 开发工具包，例如，Django 、TurboGears 、Pylons 、Zope 和WebWare ，使Python 能够快速构建功能完善和高质量的网站。   
　　**组件集成**   
　　在介绍Python 作为控制语言时，曾涉及它的组件集成的角色。Python 可以通过C/C++ 系统进行扩展，并能够嵌套C/C++ 系统的特性，使其能够作为一种灵活的粘合语言，脚本化处理其他系统和组件的行为。例如，将一个C库集成到Python 中，能够利用Python 进行测试并调用库中的其他组件；将Python 嵌入到产品中，在不需要重新编译整个产品或分发源代码的情况下，能够进行产品的单独定制。   
　　为了在脚本中使用，在Python 连接编译好组件时，例如，SWIG 和SIP 这样的代码生成工具可以让这部分工作自动完成。更大一些的框架，例如，Python 的微软Windows 所支持的COM 、基于Java 实现的Jython 、基于.NET 实现的IronPython 和各种CORBA 工具包，提供了多种不同的脚本组件。例如，在Windows 中，Python 脚本可利用框架对微软Word 和Excel 文件进行脚本处理。   
　　**数据库编程**   
　　对于传统的数据库需求，Python 提供了对所有主流关系数据库系统的接口，例如，Sybase 、Oracle 、Informix 、ODBC 、MySQL 、PostgreSQL 、SQLite 。Python 定义了一种通过Python 脚本存取SQL 数据库系统的可移植的数据库API ，这个API 对于各种底层应用的数据库系统都是统一的。例如，因为厂商的接口实现为可移植的API ，所以一个写给自由软件MySQL 系统的脚本在很大程度上不需改变就可以工作在其他系统上（例如，Oracle ）-- 你仅需要将底层的厂商接口替换掉就可以实现。   
　　Python 标准的pickle 模块提供了一个简单的对象持久化系统：它能够让程序轻松地将整个Python 对象保存和恢复至文件和文件类的对象中。在网络上，同样可以找到名叫ZODB 的第三方系统，它为Python 脚本提供了完整的面向对象数据库系统，系统SQLObject 可以将关系数据库映射至Python 的类模块。并且，从Python 2.5 版本开始，SQLite 已经成为Python 自带标准库的一部分了。   
　　**快速原型**   
　　对于Python 程序来说，使用Python 或C编写的组件看起来都是一样的。正因为如此，我们可以在一开始利用Python 做系统原型，之后再将组件移植到C或C++ 这样的编译语言上。和其他的原型工具不同，当原型确定后，Python 不需要重写。系统中不需要像C++ 这样执行效率的部分可以保持不变，从而使维护和使用变得轻松起来。   
　　**数值计算和科学计算编程**   
　　我们之前提到过的NumPy 数值编程扩展包括很多高级工具，例如，矩阵对象、标准数学库的接口等。通过将Python 与出于速度考虑而使用编译语言编写的数值计算的常规代码进行集成，NumPy 将Python 变成一个缜密严谨并简单易用的数值计算工具，这个工具通常可以替代已有的代码，而这些代码都是用FORTRAN 或C++ 等编译语言编写的。其他一些数值计算工具为Python 提供了动画、3D 可视化、并行处理等功能的支持。   
　　**游戏、图像、人工智能、XML 、机器人等**   
　　Python 的应用领域很多，远比这里提到的多得多。例如，可以利用pygame 系统使用Python 对图形和游戏进行编程；用PIL 和其他的一些工具进行图像处理；用PyRo 工具包进行机器人控制编程；用xml 库、xmlrpclib 模块和其他一些第三方扩展进行XML 解析；使用神经网络仿真器和专业的系统shell 进行AI 编程；使用NLTK 包进行自然语言分析：甚至可以使用PySol 程序下棋娱乐。可以从Vaults of Parnassus 以及新的PyPI 网站（请在Google 或http://www.python.org 上获得具体链接）找到这些领域的更多支持。   
　　一般来说，这些特定领域当中有许多在很大程度上都是Python 组件集成角色的再次例证。采用C这样的编译语言编写库组件，增加Python 至其前端，这样的方式使Python 在不同领域广泛地发挥其自身价值。对于一种支持集成的通用型语言，Python 的应用极其广泛。   
　　企业与政务应用   
　　目前，Python已经成功的实现企业级应用，在全球，已经有很多公司采用Python进行企业级软件的开发和应用，比如：ERP和CRM这样的应用。同时，通过Python技术，成功的实现了许多政务应用。   
　　用Python写的较大的系统有：   
　　Zope：一个应用程序服务器，具有内容管理、团队开发、XML、面向对象、SOAP接口等一系先进特性，且开放源码。参见：www.zope.com。   
　　Gadfly：一个用Python写的面向对象关系型数据库，具有小巧、快速、可移植性好，具有大部分SQL语言特性，且开放源码。参见：http://www.chordate.com/gadfly.html。   
　　还有一些比较有名的公司也在用Python进行原型的开发，如Microsoft、IBM等，还有：   
　　红帽(Red Hat )曾用Python和Tk一起成功开发配置和管理操作系统的可视界面。整个系统可以全面控制Linux操作系统，并根据用户选择对配置文件作自动更新。   
　　Infoseek在其公用搜索引擎使用了Python。该公司还用Python对其软件进行定制，使最终用户能对该网站内容进行方便下载。

**云计算平台(运维系统)**

OpenStack是IaaS(基础设施即服务)组件，让任何人都可以自行建立和提供云端运算服务,全部采用python开发

OpenStack 是一个旨在为公共及私有云的建设与管理提供软件的开源项目。它的社区拥有超过130家企业及1350位开发者，这些机构与个人都将OpenStack作为基础设施即服务（简称IaaS）资源的通用前端。OpenStack项目的首要任务是简化云的部署过程并为其带来良好的可扩展性。本文希望通过提供必要的指导信息，帮助大家利用OpenStack前端来设置及管理自己的公共云或私有云。

企业内部采用python开发运维管理系统，因为python在web , 系统管理方面

都比较擅长，还有面向对象特性，这是shell脚本不能胜任的，相比perl他语法简单，可读性强，所是可以大面积使用的语言

**Python 和其他语言比较**   
　　比Tcl 强大。Python 支持"大规模编程"，使其适宜于开发大型系统。   
　　有着比Perl 更简洁的语法和更简单的设计，这使得Python 更具可读性、更易于维护，有助于减少程序bug 。   
　　比Java 更简单、更易于使用。Python 是一种脚本语言，Java 从C++ 这样的系统语言中继承了许多语法和复杂性。   
　　比C++ 更简单、更易于使用，但通常也不与C++ 竞争。因为Python 作为脚本语言，常常扮演多种不同的角色。   
　　比Visual Basic 更强大也更具备跨平台特性。由于Python 是开源的，也就意味着它不可能被某一个公司所掌控。   
　　比Ruby 更成熟、语法更具可读性。与Ruby 和Java 不同的是，OOP 对于Python 是可选的：这意味着Python 不会强制用户或项目选择OOP 进行开发。   
　　具备SmallTalk 和Lisp 等动态类型的特性，但是对开发者及定制系统的终端用户来说更简单，也更接近传统编程语言的语法。   
　　特别对不仅仅做文本文件扫描还有也许未来会被人们读到（或者说你）的程序而言，很多人会发现Python 比目前任何的可用的脚本或编程语言都划得来。不仅如此，除非你的应用要求最尖端的性能，Python 往往是C、C++ 和Java 等系统开发语言的一个不错的替代品：Python 将会减少很多编写、调试和维护的麻烦。

**目前使用Python的企业**   
　　Google 在其网络搜索系统中广泛应用了Python ，并且聘用了Python 的创作者。   
　　YouTube 视频分享服务大部分是由Python 编写的。   
　　流行的P2P 文件分享系统Bittorrent 是一个Python 程序。   
　　Intel 、Cisco 、Hewlett-Packard 、Seagate 、Qualcomm 和IBM 使用Python 进行硬件测试。   
　　Industrial Light & Magic 、Pixar 等公司使用Python 制作动画电影。   
　　在经济市场预测方面，JPMorgan Chase 、UBS 、Getco 和Citadel 使用Python 。   
　　NASA 、Los Alamos 、Fermilab 、JPL 等使用Python 实现科学计算任务。   
　　IRobot 使用Python 开发了商业机器人真空吸尘器。   
　　ESRI 在其流行的GIS 地图产品中使用Python 作为终端用户的定制工具。   
　　NSA 在加密和智能分析中使用Python 。   
　　IronPort 电子邮件服务器产品中使用了超过100 万行的Python 代码实现其作业。   
　　OLPC 使用Python 建立其用户界面和动作模块。

**总结：**

个人总结python有两个主要优点：

1. 胶水语言

Python在功能方面是个折中的语言，在各个领域相比来说没有特别突出，他的优点在于，他与各语言的集成度很高，就像一个胶水，其实这在目前开发语言百花齐放的阶段，各个系统独立分散，环境复杂，确实需要一个通用的语言能架起他们之前的桥梁,它对各语言的支持最广泛，甚至都可以调用其它语言的类库，在云时代迫切需要一个环境单一，便于维护的，开源的，支持广泛应用的系统，python虽然是一门很老的语言，但他的应用领域还是一些比较新的领域，他的潜力很大，他经历了这么久的历练，相信在云时代，他会有不可替代的作用

1. 可维护性

一个好的代码，不仅仅是自己能看懂，其它人能看懂，这才是很好的代码，python代码倡导简洁，可读性强，这是他最明显的优点，相比java他语法更简单，同功能代码量更少,他是新一代的面向对象的脚本语言。