

Python 简介

Python 是一个高层次的结了解释性、编译性、互动性和面向对象的脚本语言。

Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，其他语言的一些标点符号，它具有比其他语言更有特色语法结构。

- **Python 是一种解释型语言：**这意味着开发过程中没有了编译这个环节。类似于PHP和Perl语言。
- **Python 是交互式语言：**这意味着，您可以在一个 Python 提示符 `>>>` 后直接执行代码。
- **Python 是面向对象语言：**这意味着Python支持面向对象的风格或代码封装在对象的编程技术。
- **Python 是初学者的语言：**Python 对初级程序员而言，是一种伟大的语言，它支持广泛的应用程序开发，从简单的文字处理到 WWW 浏览器再到游戏。

Python 发展历史

Python 是由 Guido van Rossum 在八十年代末和九十年代初，在荷兰国家数学和计算机科学研究所设计出来的。

Python 本身也是由诸多其他语言发展而来的,这包括 ABC、Modula-3、C、C++、Algol-68、SmallTalk、Unix shell 和其他的脚本语言等等。

像 Perl 语言一样，Python 源代码同样遵循 GPL(GNU General Public License)协议。

现在 Python 是由一个核心开发团队在维护，Guido van Rossum 仍然占据着至关重要的作用，指导其进展。

Python 特点

- **1.易于学习：**Python有相对较少的关键字，结构简单，和一个明确定义的语法，学习起来更加简单。
- **2.易于阅读：**Python代码定义的更清晰。
- **3.易于维护：**Python的成功在于它的源代码是相当容易维护的。
- **4.一个广泛的标准库：**Python的最大的优势之一是丰富的库，跨平台的，在UNIX，Windows和Macintosh兼容很好。
- **5.互动模式：**互动模式的支持，您可以从终端输入执行代码并获得结果的语言，互动的测试和调试代码片断。
- **6.可移植：**基于其开放源代码的特性，Python已经被移植（也就是使其工作）到许多平台。
- **7.可扩展：**如果你需要一段运行很快的关键代码，或者是想要编写一些不愿开放的算法，你可以使用C或C++完成那部分程序，然后从你的Python程序中调用。

- **8.数据库**：Python提供所有主要的商业数据库的接口。
- **9.GUI编程**：Python支持GUI可以创建和移植到许多系统调用。
- **10.可嵌入**：你可以将Python嵌入到C/C++程序，让你的程序的用户获得"脚本化"的能力。

[← Python 基础教程](#)[Python 环境搭建 →](#)

1 篇笔记

[✎ 写笔记](#)

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，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。
- **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。
- **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。
- **高层语言** -- 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。
- **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、OS/2、Amiga、AROS、AS/400、BeOS、OS/390、z/OS、Palm OS、QNX、VMS、Psion、Acom RISC OS、VxWorks、PlayStation、Sharp Zaurus、Windows CE 甚至还有 PocketPC、Symbian 以及 Google 基于 Linux 开发的 Android 平台！
- **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用的机器语言并运行。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。
- **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。
- **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用 C 或 C++ 编写，然后到你的 Python 程序中使用它们。

- 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。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极佳的可读性。

缺点

- 运行速度，有速度要求的话，用 C++ 改写关键部分吧。
- 国内市场较小（国内以 Python 来做主要开发的，目前只有一些 web2.0 公司）。但时间推移，目前很多国内软件公司，尤其是游戏公司，也开始规模使用他。
- 中文资料匮乏（好的 Python 中文资料屈指可数，现在应该变多了）。托社区的福，有几本优秀的教材已经被翻译了，但入门级教材多，高级内容还是只能看英语版。
- 构架选择太多（没有像 C# 这样的官方 .net 构架，也没有像 ruby 由于历史较短，构架开发的相对集中。Ruby on Rails 构架开发中小型web程序天下无敌）。不过这也从另一个侧面说明，python 比较优秀，吸引的人才多，项目也多。

wwoo55 3周前 (02-26)