## 1 python解释器

自己总结：

类似于jvm的编译器，将代码编译为可执行文件。、

解释器就是帮助我们将Python代码，也就是.py文件，交给机器可以执行的工具。

通常我们将Python和Java语音归为解释型语言，而对于C/C++则归为编译型语音，

…

### 1. python解释器是什么？

计算机的大脑是CPU， 中文名叫中央处理器，它仍然不能直接处理 Python 语言。CPU 只能直接处理机器指令语言，那是一种由0和1数字组成的语言，这是一种我们人很难直接写出来的语言。所以，我们需要一个翻译， 把Python语言翻译成 计算机CPU 能听懂的机器指令语言，这样计算机才能按照 我们的Python程序的要求去做事。

.py结尾的文件需要解释器去运行执行

1. python解释器的构成及其各部分功能

解释器由一个编译器和一个虚拟机构成，编译器负责将源代码转换成字节码文件，而虚拟机负责执行字节码。

所以，解释型语言其实也有编译过程，只不过这个编译过程并不是直接生成目标代码，而是中间代码（字节码），然后再通过虚拟机来逐行解释执行字节码

2. 执行过程原理

1).执行 python XX.py 后，将会启动 Python 的解释器

2).python解释器的编译器会将.py源文件编译（解释）成字节码生成PyCodeObject字节码对象存放在内存中。

3).python解释器的虚拟机将执行内存中的字节码对象转化为机器语言，虚拟机与操作系统交互，使机器语言在机器硬件上运行。

4).运行结束后python解释器则将PyCodeObject写回到pyc文件中。当python程序第二次运行时，首先程序会在硬盘中寻找pyc文件，如果找到，则直接载入，否则就重复上面的过程。

所以我们应该这样来定位PyCodeObject和pyc文件，我们说pyc文件其实是PyCodeObject的一种持久化保存方式。

pyc文件，文件中包含python的magic number（来说明编译时使用的python版本号）、源文件的mtime（使pyc和py文件保持同步）、编译出的code对象

Python解释器执行程序的三个阶段，例如执行（python3 D:\test.py）

第一步：先启动python3解释器

第二步：Python3解释器就像一个文本编辑器一样将文件python3 D：\test.py从硬盘读入内存

第三步：Python3解释器解释执行文件代码

只有第三阶段才识别python的语法

### 2. python解释器的种类

Cpython

官方下载的python2.7均为Cpython。Cpython是用C语言开发的，因此得名

Ipython

IPython是基于CPython之上的一个交互式解释器，也就是说，IPython只是在交互方式上有所增强，但是执行Python代码的功能和CPython是完全一样的。好比很多国产浏览器虽然外观不同，但内核其实都是调用了IE。

CPython用>>>作为提示符，而IPython用In [序号]:作为提示符。

PyPy

PyPy是另一个Python解释器，它的目标是执行速度。PyPy采用JIT技术，对Python代码进行动态编译（注意不是解释），所以可以显著提高Python代码的执行速度。

绝大部分Python代码都可以在PyPy下运行，但是PyPy和CPython有一些是不同的，这就导致相同的Python代码在两种解释器下执行可能会有不同的结果。如果你的代码要放到PyPy下执行，就需要了解PyPy和CPython的不同点

Jython

Jython是运行在Java平台上的Python解释器，可以直接把Python代码编译成Java字节码执行。

Ironpython

IronPython和Jython类似，只不过IronPython是运行在微软.Net平台上的Python解释器，可以直接把Python代码编译成.Net的字节码。

3.现在使用的解释器

Cpython

## 2 python主流框架

<https://blog.csdn.net/h330531987/article/details/86589130>

|  |
| --- |
| 从GitHub中整理出的15个最受欢迎的Python开源框架。这些框架包括事件I/O，OLAP，Web开发，高性能网络通信，测试，爬虫等。    Django: Python Web应用开发框架  Django 应该是最出名的Python框架，GAE甚至Erlang都有框架受它影响。Django是走大而全的方向，它最出名的是其全自动化的管理后台：只需要使用起ORM，做简单的对象定义，它就能自动生成数据库结构、以及全功能的管理后台。    Diesel：基于Greenlet的事件I/O框架  Diesel提供一个整洁的API来编写网络客户端和服务器。支持TCP和UDP。    Flask：一个用Python编写的轻量级Web应用框架  Flask是一个使用Python编写的轻量级Web应用框架。基于Werkzeug WSGI工具箱和Jinja2  模板引擎。Flask也被称为“microframework”，因为它使用简单的核心，用extension增加其他功能。Flask没有默认使用的数  据库、窗体验证工具。    Cubes：轻量级Python OLAP框架  Cubes是一个轻量级Python框架，包含OLAP、多维数据分析和浏览聚合数据（aggregated data）等工具。    Kartograph.py：创造矢量地图的轻量级Python框架  Kartograph是一个Python库，用来为ESRI生成SVG地图。Kartograph.py目前仍处于beta阶段，你可以在virtualenv环境下来测试。    Pulsar：Python的事件驱动并发框架  Pulsar是一个事件驱动的并发框架，有了pulsar，你可以写出在不同进程或线程中运行一个或多个活动的异步服务器。    Web2py：全栈式Web框架  Web2py是一个为Python语言提供的全功能Web应用框架，旨在敏捷快速的开发Web应用，具有快速、安全以及可移植的数据库驱动的应用，兼容Google App Engine。    Falcon：构建云API和网络应用后端的高性能Python框架  Falcon是一个构建云API的高性能Python框架，它鼓励使用REST架构风格，尽可能以最少的力气做最多的事情。    Dpark：Python版的Spark  DPark是Spark的Python克隆，是一个Python实现的分布式计算框架，可以非常方便地实现大规模数据处理和迭代计算。DPark由豆瓣实现，目前豆瓣内部的绝大多数数据分析都使用DPark完成，正日趋完善。    Buildbot：基于Python的持续集成测试框架  Buildbot是一个开源框架，可以自动化软件构建、测试和发布等过程。每当代码有改变，服务器要求不同平台上的客户端立即进行代码构建和测试，收集并报告不同平台的构建和测试结果。    Zerorpc：基于ZeroMQ的高性能分布式RPC框架  Zerorpc是一个基于ZeroMQ和MessagePack开发的远程过程调用协议（RPC）实现。和 Zerorpc 一起使用的 Service API 被称为 zeroservice。Zerorpc 可以通过编程或命令行方式调用。    Bottle: 微型Python Web框架  Bottle是一个简单高效的遵循WSGI的微型python Web框架。说微型，是因为它只有一个文件，除Python标准库外，它不依赖于任何第三方模块。    Tornado：异步非阻塞IO的Python Web框架  Tornado的全称是Torado Web Server，从名字上看就可知道它可以用作Web服务器，但同时它也是一个Python Web的开发框架。最初是在FriendFeed公司的网站上使用，FaceBook收购了之后便开源了出来。    webpy: 轻量级的Python Web框架  webpy的设计理念力求精简（Keep it simple and powerful），源码很简短，只提供一个框架所必须的东西，不依赖大量的第三方模块，它没有URL路由、没有模板也没有数据库的访问。    Scrapy：Python的爬虫框架  Scrapy是一个使用Python编写的，轻量级的，简单轻巧，并且使用起来非常的方便。 |

## 3 pip

pip 是 Python 包管理工具，该工具提供了对Python 包的查找、下载、安装、卸载的功能。

目前如果你在 [python.org](https://www.python.org/) 下载最新版本的安装包，则是已经自带了该工具。

Python 2.7.9 + 或 Python 3.4+ 以上版本都自带 pip 工具。

pip 官网：<https://pypi.org/project/pip/>

你可以通过以下命令来判断是否已安装：

pip --version

如果你还未安装，则可以使用以下方法来安装：

$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py # 下载安装脚本

$ sudo python get-pip.py # 运行安装脚本

***注意：****用哪个版本的 Python 运行安装脚本，pip 就被关联到哪个版本，如果是 Python3 则执行以下命令：*

$ sudo python3 get-pip.py # 运行安装脚本。

*一般情况 pip 对应的是 Python 2.7，pip3 对应的是 Python 3.x。*

部分 Linux 发行版可直接用包管理器安装 pip，如 Debian 和 Ubuntu：

sudo apt-get install python-pip

### pip 最常用命令

**显示版本和路径**

pip --version

**获取帮助**

pip --help

**升级 pip**

pip install -U pip

*如果这个升级命令出现问题 ，可以使用以下命令：*

sudo easy\_install --upgrade pip

**安装包**

pip install SomePackage # 最新版本

pip install SomePackage==1.0.4 # 指定版本

pip install 'SomePackage>=1.0.4' # 最小版本

比如我要安装 Django。用以下的一条命令就可以，方便快捷。

pip install Django==1.7

**升级包**

pip install --upgrade SomePackage

升级指定的包，通过使用==, >=, <=, >, < 来指定一个版本号。

**卸载包**

pip uninstall SomePackage

**搜索包**

pip search SomePackage

**显示安装包信息**

pip show

**查看指定包的详细信息**

pip show -f SomePackage

**列出已安装的包**

pip list

**查看可升级的包**

pip list -o

### 注意事项

如果 Python2 和 Python3 同时有 pip，则使用方法如下：

Python2：

python2 -m pip install XXX

Python3:

python3 -m pip install XXX