# 第一章 Python的概述与环境安装

本节所讲内容：

1.1 Python介绍

1.2 Python2与Python3的比较

1.3 Python3的安装

1.4 Python环境变量配置

1.5 Python安装目录介绍

1.6 Python脚本结构

1.7 Python脚本运行

## 1.1 Python介绍

### 1.1.1 Python语言介绍

Python是一门高级的、面向对象的、解释性、脚本语言。

高级语言：贴近开发者，对应底层语言，底层语言贴近机器；java、C#、php 。

面向对象对应于面向过程，是开发人员在开发过程当中的思路，是程序员的世界观，python 一切皆对象。

解释性对应于编译性语言。

解释性语言：

解释型语言在程序执行时，有一个的编译过程，这样解释性语言每执行一次就要翻译一次。

编译性语言：  
编译型语言在程序执行之前，有一个单独的编译过程，将程序翻译成机器语言，以后执行这个程序的时候，就不用再进行翻译了。

脚本语言是一种解释性的语言,例如vbscript,javascript,installshield script,ActionScript等等,它不像c\c++等可以编译成二进制代码,以可执行文件的形式存在.脚本语言不需要编译，可以直接用，由解释器来负责解释。 脚本语言一般都是以文本形式存在,类似于一种命令。

Python 起源

Python的作者，Guido von Rossum（吉多·范罗苏姆）我们称他为‘龟叔’，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。用他的话说，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，并热衷于做任何和编程相关的活儿。在那个时候，Guido接触并使用过诸如Pascal [ˈpæskəl]、C、Fortran[ˈfɔ:træn] 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(麦金塔电脑)，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。为了增进效率，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口味的程序。

在那个时代，程序员恨不得用手榨取计算机每一寸的能力。有人甚至认为C语言的指针是在浪费内存。至于动态类型，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(是一个交换式的命令[解释器](https://baike.baidu.com/item/%E8%A7%A3%E9%87%8A%E5%99%A8" \t "https://baike.baidu.com/item/Bourne%20Shell/_blank)和命令编程语言)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下的许多功能连接在一起。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。

HOW TO用于定义一个函数。一个Python程序员应该很容易理解这段程序。ABC语言使用冒号和缩进来表示程序块、行尾没有分号、for和if结构中也没有括号() 、赋值采用的是PUT，而不是更常见的等号。这些改动让ABC程序读起来像一段文字。 尽管已经具备了良好的可读性和易用性，ABC语言最终没有流行起来。

在当时，ABC语言编译器需要比较高配置的电脑才能运行。而这些电脑的使用者通常精通计算机，它们更多考虑程序的效率，而非它的学习难度。除了硬件上的困难外，ABC语言的设计也存在一些致命的问题： 可拓展性差。

ABC语言不是模块化语言。如果想在ABC语言中增加功能，比如对图形化的支持，就必须改动很多地方，它不能直接进行IO。

ABC语言不能直接操作文件系统。尽管你可以通过诸如文本流的方式导入数据，但ABC无法直接读写文件。输入输出的困难对于计算机语言来说是致命的。你能想像一个打不开车门的跑车么？

尽管ABC语言很特别，但学习难度 也很大；

传播困难ABC编译器很大，必须被保存在磁带上；

当时Guido在访问的时候，就必须有一个大磁带来给别人安装ABC编译器。 这样，ABC语言就很难快速传播。

1989年，为了打发圣诞节假期（寂寞难耐），Guido开始写Python语言的编译器。Python这个名字，来自Guido所挚爱的电视剧Monty Python's Flying Circus(飞行马戏团)。他希望这个新的叫做Python的语言，能符合他的理想：创造一种C和shell之间，功能全面，易学易用，可拓展的语言。

Guido作为一个语言设计爱好者，已经有过设计语言的尝试。这一次，也不过是一次纯粹的hacking(搬运)行为。

一门语言的诞生

1991年，第一个Python编译器诞生。它是用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写. 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就好像是使用钢构建房一样，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完全由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和一些同事构成Python的核心团队。他们将自己大部分的业余时间用于hack(熟练地编辑) Python。随后，Python拓 展到研究所之外。 Python将许多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程序员可以花更多的时间用于思考程序的逻辑，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。



人生苦短，我用python

启示录

Python崇尚优美、清晰、简单，是一个优秀并广泛使用的语言。Python在TIOBE排行榜中排行第四，它是Google的第三大开发语言，Dropbox的基础语言，豆瓣的服务器语言。Python的发展史可以作为一个代表，带给我许多启示。 在Python的开发过程中，社区起到了重要的作用。

Guido自认为自己不是全能型的程序员，所以他只负责制订框架。如果问题太复杂，他会选择绕过去，也就是cut the corner（走捷径）。这些问题最终由社区中的其他人解决。社区中的人才是异常丰富的，就连创建网站，筹集基金这样与开发稍远的事情，也有人乐意于处理。如今的项目开发越来越复杂，越来越庞大，合作以及开放的心态成为项目最终成功的关键。

Python从其他语言中学到了很多，无论是已经进入历史的ABC，还是依然在使用的C和Perl，以及许多没有列出的其他 语言。可以说，Python的成功代表了它所有借鉴的语言的成功。同样，Ruby借鉴了Python，它的成功也代表了Python某些方面的成功。每个语言都是混合体，都有它优秀的地方，但也有各种各样的缺陷。

同时，一个语言“好与不好”的评 判，往往受制于平台、硬件、时代等等外部原因。程序员经历过许多语言之争。其实，以开放的心态来接受各个语言，说不定哪一天，程序员也可以如Guido那样，混合出自己的语言。

关键点常识

Python/ˈpaɪθən/ 的发音与拼写

Python的意思是蟒蛇，源于作者喜欢的一部电视剧

Python的作者是Guido van Rossum（吉多·范罗苏姆）

Python是龟叔在1989年圣诞节期间，为了打发无聊的圣诞节而用C编写的一个编程语言

Python正式诞生于1991年

Python的解释器如今有多个语言实现，我们常用的是CPython（官方版本的C语言实现），其他还有Jython（可以运行在Java平台）、IronPython（可以运行在.NET和Mono平台）、PyPy（Python实现的，支持JIT即时编译）

Python目前有两个版本，Python2和Python3，最新版分别为2.7.15和3.7，现阶段大部分公司用的是Python2 和python3

Life is shot, you need Python. 人生苦短，我用Python。

2019年3月份 编程语言流行排行榜 （https://www.tiobe.com/tiobe-index/）



### 1.1.2 Python优缺点

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，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，然后在你的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（是用[Python](http://lib.csdn.net/base/11" \t "http://blog.csdn.net/hanhuili/article/details/_blank" \o "Python知识库)实现的基于事件驱动的网络引擎框架）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

1. 速度慢 由于Python是解释型语言，所有它的速度会比，C、C++慢一些，但是不影响使用。由于，现在的硬件配置都非常高，基本上没有影响，除非是一些实时性比较强的程序可能会受到一些影响，但是也有解决办法，可以嵌入C程序。

2. 强制缩进 如果你有其他语言的编程经验，例如：C语言或者Java语言，那么Python的强制缩进一开始会让你很不习惯。但是如果你习惯了Python的缩进语法，你会觉得它非常优雅。

3. 单行语句 由于Python可以在尾部不写分号，所以一行只能有一条语句，这可能也算是一个不足吧，不过这真的微不足道。

### 1.1.3 Python应用场景

Web应用开发

Python经常被用于Web开发。比如，通过mod\_wsgi模块，Apache可以运行用Python编写的Web程序。Python定义了WSGI（是Python应用程序或框架和Web服务器之间的一种接口）标准应用接口来协调Http服务器与基于Python的Web程序之间的通信。一些Web框架，如Django,flask,tornado,Zope等，可以让程序员轻松地开发和管理复杂的Web程序。

操作系统管理、服务器运维的自动化脚本

在很多操作系统里，Python是标准的系统组件。 大多数Linux发行版以及NetBSD、OpenBSD和Mac OS X都集成了Python，可以在终端下直接运行Python。有一些Linux发行版的安装器使用Python语言编写，比如Ubuntu的Ubiquity安装器,Red Hat Linux和Fedora的Anaconda安装器。Gentoo Linux使用Python来编写它的Portage包管理系统。Python标准库包含了多个调用操作系统功能的库。

通过pywin32这个第三方软件包，Python能够访问Windows的COM服务及其它Windows API。使用IronPython，Python程序能够直接调用.Net Framework。一般说来，Python编写的系统管理脚本在可读性、性能、代码重用度、扩展性几方面都优于普通的shell脚本。

科学计算

NumPy,SciPy,Matplotlib可以让Python程序员编写科学计算程序。

桌面软件

PyQt、PySide、wxPython、PyGTK是Python快速开发桌面应用程序的利器。

服务器软件（网络软件）

Python对于各种网络协议的支持很完善，因此经常被用于编写服务器软件、网络爬虫。第三方库Twisted[ˈtwɪstɪd] 支持异步网络编程和多数标准的网络协议(包含客户端和服务器)，并且提供了多种工具，被广泛用于编写高性能的服务器软件。

游戏

很多游戏使用C++编写图形显示等高性能模块，而使用Python或者Lua编写游戏的逻辑、服务器。相较于Python，Lua的功能更简单、体积更小；而Python则支持更多的特性和数据类型。

YouTube、Google、Yahoo!、NASA都在内部大量地使用Python。

上课主要以Python3.5.0为主(为什么要以3为主呢？)

Python核心团队计划在2020年停止支持Python 2。NumPy项目自2010年以来一直支持Python 2和Python 3，并且发现支持Python 2对我们有限的资源增加了负担；因此，我们最终计划将停止支持Python 2。

到2018年12月31日为止，所有的NumPy版本都将完全支持Python2和Python3。

从2019年1月1日开始，任何新的功能版本都只支持Python3。

国内是python2和python3并存，官方和我现在极力推荐使用Python3

## 1.2 Python2与Python3的比较

### 1.2.1 print语句

python2中print是一个语句，无论想输出什么，直接放到print关键字后面即可。

python3里，print()是一个函数，像其他函数一样，print()需要你将要输出的东西作为参数传给它。

| python2 | python3 | 备注 |
| --- | --- | --- |
| print | print() | 输出一个空白行，python3需要调用不带参数的print() |
| print 1 | print(1) | 输出一个值，将值传入print()函数 |
| print 1, 2 | print(1,2) | 输出使用空格分割的两个值，使用两个参数调用print() |

### 1.2.2 <>比较运算符

Python2支持<>作为!=的同义词，python3只支持!=, 不再支持<>

### 1.2. 3 unicode字符串

python2中有两种字符串类型：Unicode字符串和非Unicode字符串。Python3中只有一种类型：Unicode字符串。

### 1.2. 4 long 长整型

Python 2有为非浮点数准备的int和long类型。int类型的最大值不能超过[sys.maxint](http://sebug.net/paper/books/dive-into-python3/porting-code-to-python-3-with-2to3.html" \l "renames" \t "https://blog.csdn.net/samxx8/article/details/_blank)，而且这个最大值是平台相关的。可以通过在数字的末尾附上一个L来定义长整型，显然，它比int类型表示的数字范围更大。

在Python 3里，[只有一种整数类型int](http://sebug.net/paper/books/dive-into-python3/native-datatypes.html" \l "numbers" \t "https://blog.csdn.net/samxx8/article/details/_blank)，大多数情况下，它很像Python 2里的长整型。由于已经不存在两种类型的整数，所以就没有必要使用特殊的语法去区别他们。

## 1.3 Python3的安装

### 1.3.1 python3安装

直接到官网https://www.python.org/下载，安装就可以了。

安装比较简单，点exe文件一直下一步就可以了（注意：安装的时候有个选择是否添加环境变量，这个选是，之后就不用添加环境变量了）如果没有，请添加环境变量。

🡪右键点击计算机

🡪点击属性

🡪高级系统设置

🡪环境变量

🡪系统变量里的path

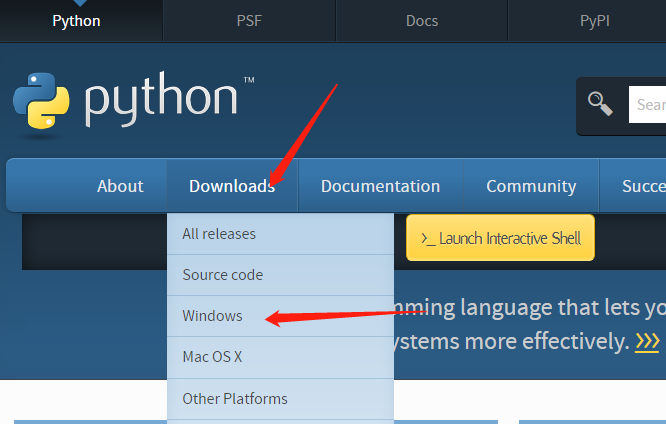
🡪编辑添加安装python的路径

（例如E:xuegod;E:\xuegod\script)

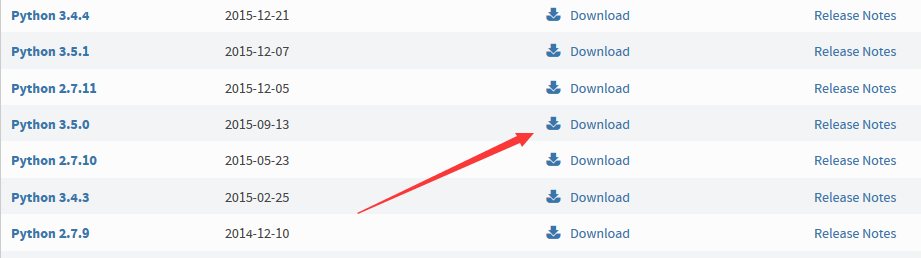
修改python3安装目录下的python.exe文件重命名为python3.exe。

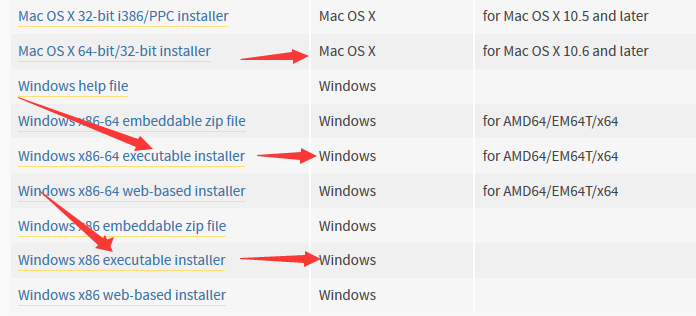
（这样命令窗口运行python3输入python3，运行 python3）

注意：我们选择为64位安装包

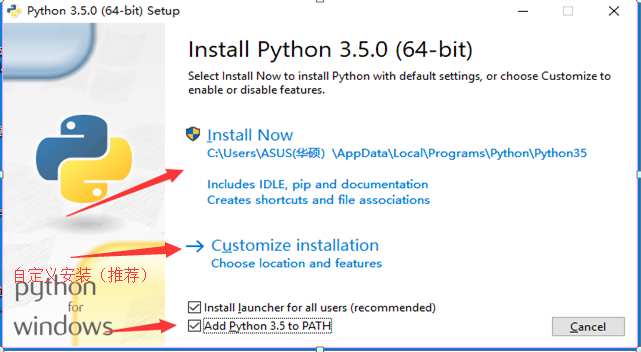


第一步：

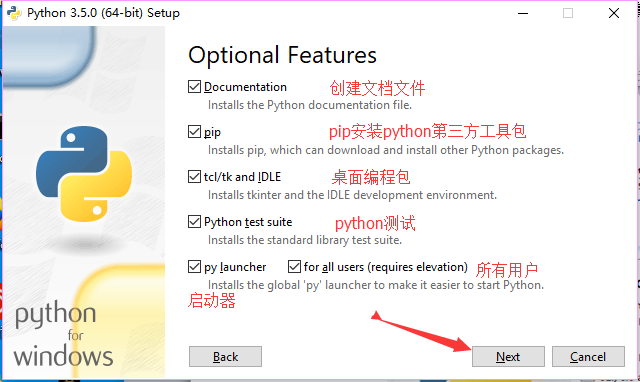


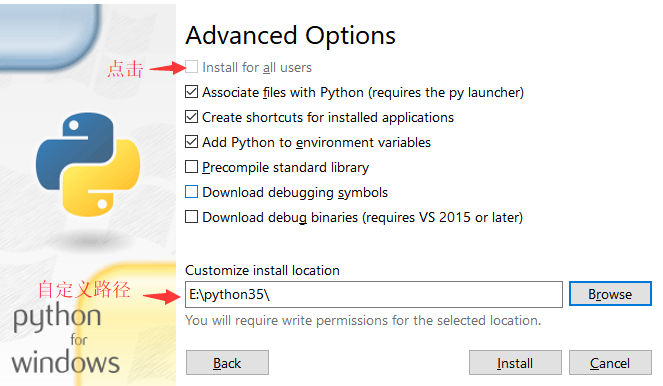


第二步：可以install now，或者是自定义（当然老师推荐大家是自定义安装）

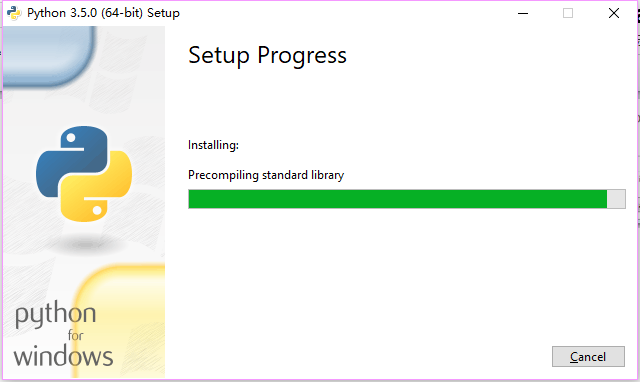


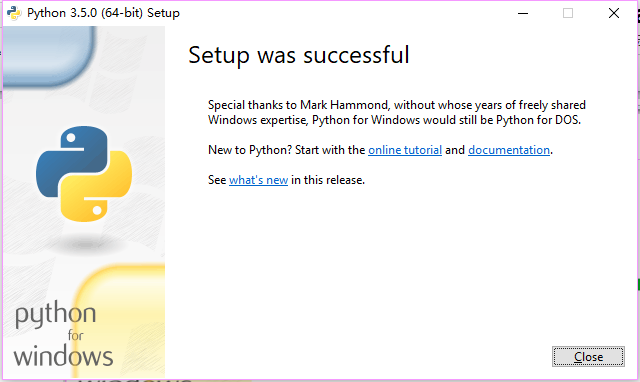
第三步：





安装完成





## 1.4 python环境变量配置

### 1.4.1 配置环境变量

🡪右键点击我的电脑

🡪 点击属性

🡪 点击高级系统设置

🡪 点击环境变量

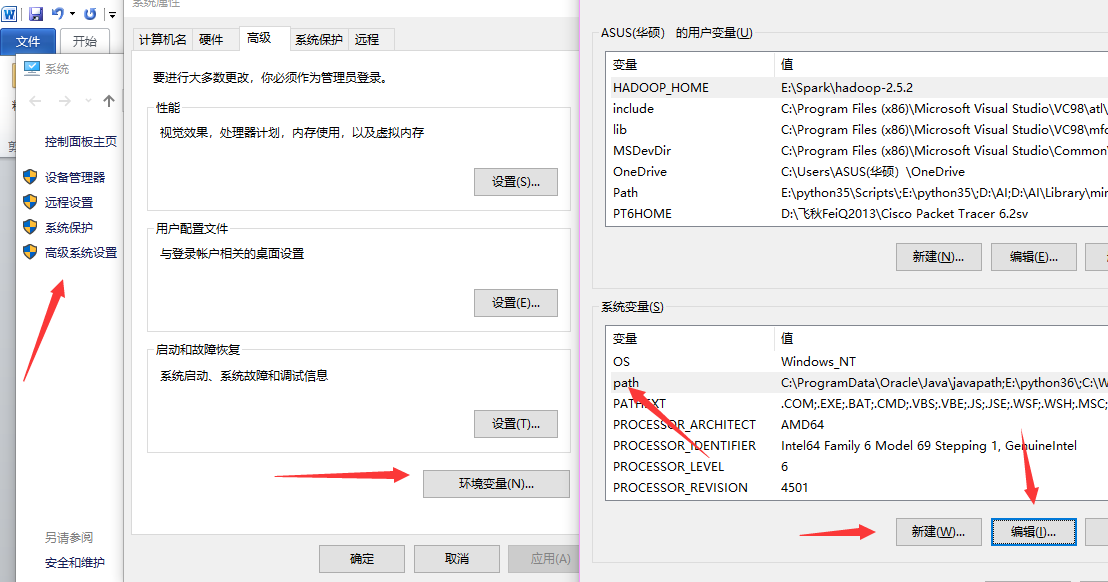
🡪 在系统变量里寻找path（没有新建）

🡪 有path的点击添加

🡪 将你python安装目录以及python中scripts的文件路径添加到path中。

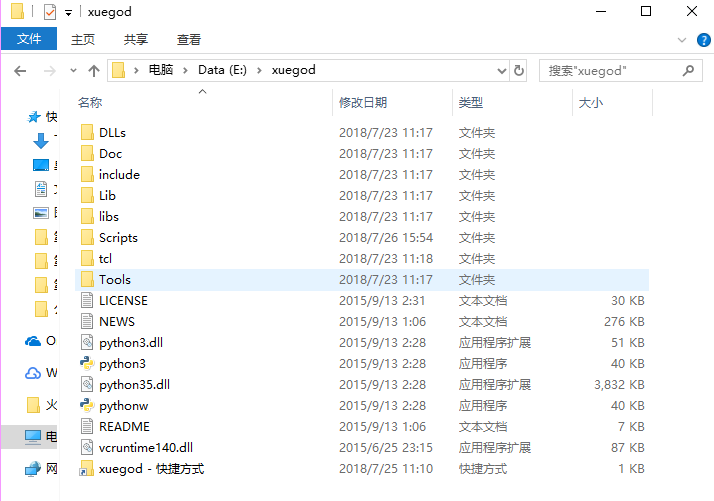
例如老师（E:\xuegod，E:\xuegod\Scripts）。

恭喜你整个python已经安装成功。



## 1.5 python安装目录介绍

### 1.5.1 安装目录介绍



DLLS: python的动态链接库，跟编译器有关，和python编程关系不大

Doc: python 的参考书，有很多实例

Include: python 编译器的C语言头文件源码

Lib: 这个库很重要，是python的标准库，你扩充的库都安装在这里面。

Libs： python的内置库，语法存在的地方

Scripts: 项目用到的各种脚本

Tcl：包含python默认内置的GUI工具Tkiner

Tools目录：Python 提供的一些工具

Tcl: 桌面编程包

python目录 lib 和libs有什么区别？

Lib是python的标准库，我们安装的第三方模块都在这里面，libs是python的内置库，下载python的时候自带的。所以我们不要把API（应用程序接口）放到libs中，要放到lib中。

## 1.6 Python脚本结构

### 1.6.1 Python脚本结构

.py 文件

以 .py 作扩展名的文件是 Python 源代码文件，由 python.exe 解释，可在控制台下运行。当然，也可用文本编辑器进行修改。

.pyc 文件

以 .pyc 作扩展名的文件是 python 编译文件。.pyc 文件是不能用文本编辑器进行编辑的，其优点是 .pyc 文件的执行速度要远快于 .py 文件。至于为什么要有 .pyc 文件，这个需求太明显了，因为 .py 文件是可直接看到源码的，若是软件开发商的话，不可能把源码泄漏出去？所以，就需编译成 .pyc 后再发布。

.pyw 文件

.pyc 文件执行时，桌面会出现类似 CMD 命令的黑色 shell 窗口，十分难看，于是 .pyw 文件就应运而生了。.pyw 文件与 .pyc 文件的执行，本质上并没什么区别，只是 .pyw 文件执行的时候不会出现类似 CMD 命令的黑色 shell 窗口。.pyw 文件格式主要是设计用来运行纯 GUI 图形用户界面程序的。 纯 GUI 图形用户界面程序的用户不需要看到类似 CMD 命令的黑色 shell 控制台窗口。.pyw 文件运行时，所有 stdout、stderr 输出无效，所有原 stdin 的读取只会得到 EOF。   
值得一提的是，开发纯 GUI 图形用户界面程序时，可暂时把 .pyw 改成 .py，以便运行时调出控制台窗口，看到所有错误信息，方便修改、调试。

.pyo 文件

.pyo 文件是优化编译后的程序。 “python -O 源文件” 即可将源程序编译为 .pyo 文件。同样，.pyo 文件也是不能用文本编辑器进行编辑的。

.pyd 文件

.pyd 文件并不是用 python 编写成的，.pyd 文件一般是其他语言编写的 python 扩展模块。.pyd 文件是用 D 语言按照一定格式编写，并处理成二进制的文件。

窗用 python.exe 运行 .py ，用 pythonw.exe 运行 .pyw 。

这纯粹是因为安装视窗版 Python 时，扩展名 .py 自动被登记为用 python.exe 运行的文件，

而 .pyw 则被登记为用 pythonw.exe 运行。

.py 和 .pyw 之间的“其它差别”全都是 python.exe 和 pythonw.exe 之间的差别。

跟 python.exe 比较起来，pythonw.exe 有以下的不同：

(1）执行时不会弹出控制台窗口（也叫 DOS 窗口）

(2）所有向原有的 stdout 和 stderr 的输出都无效

(3）所有从原有的 stdin 的读取都只会得到 EOF

.pyw 格式是被设计来运行开发完成的纯图形界面程序的。

纯图形界面程序的用户不需要看到控制台窗口。

### 1.6.2 python成熟的编译器

Sublime

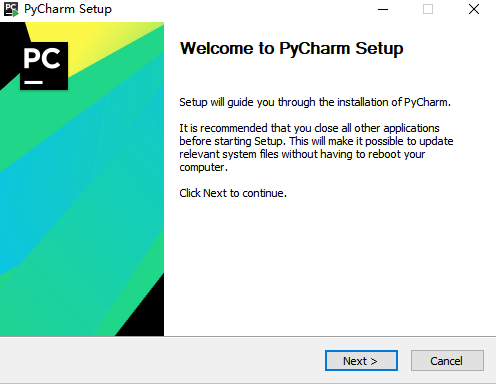
Pycharm

Ipython

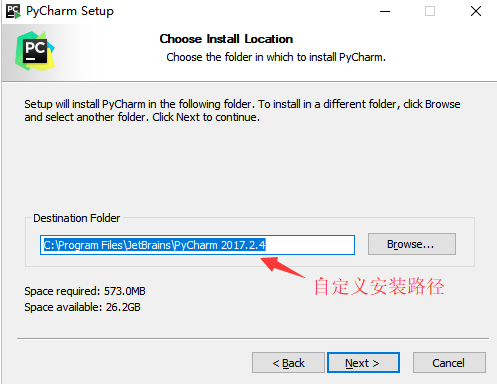
IDLE python gui

(IDLE是python自带的简单的集成开发环境)

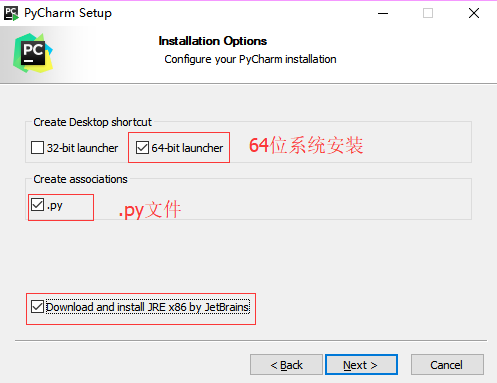
我们主要使用pycharm，PyCharm是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发（老师会在群里分享pycharm安装包）。   
安装包下载地址：https://www.jetbrains.com/pycharm/download/#section=windows

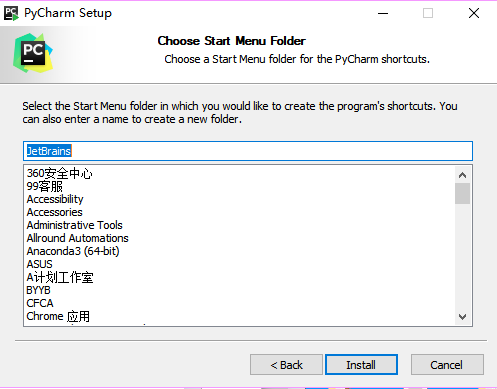


点击next

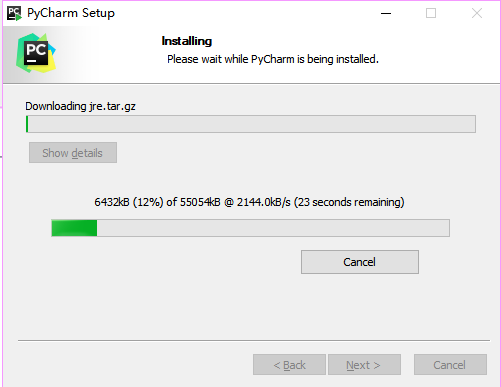


自定义文件安装路径

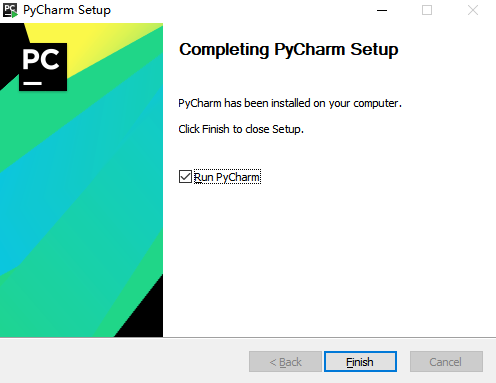




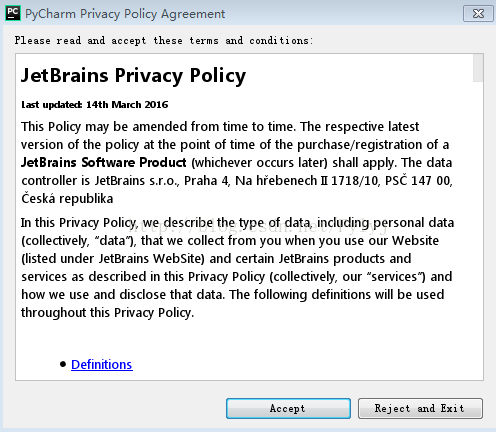
点击install



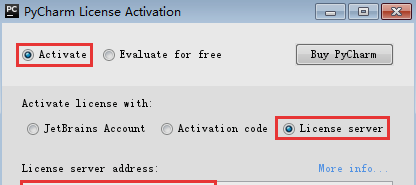
等待安装好，点击next



点击finish



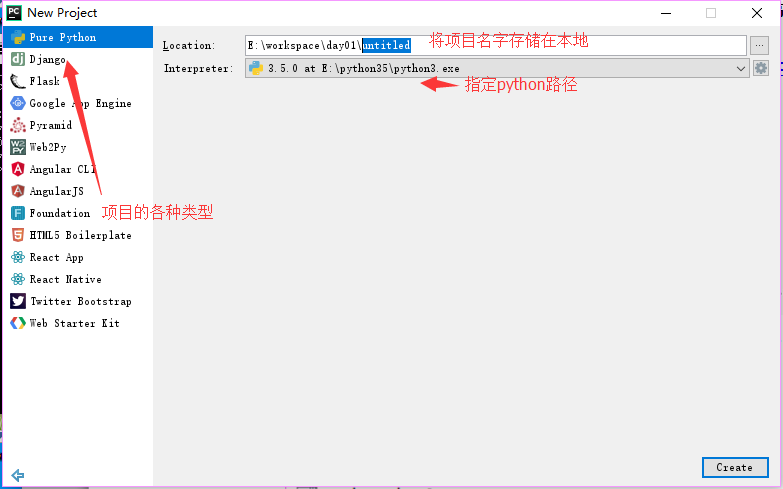
之后就进入激活阶段



老师在群里已经发过激活码，配置好后，直接出现以下界面。

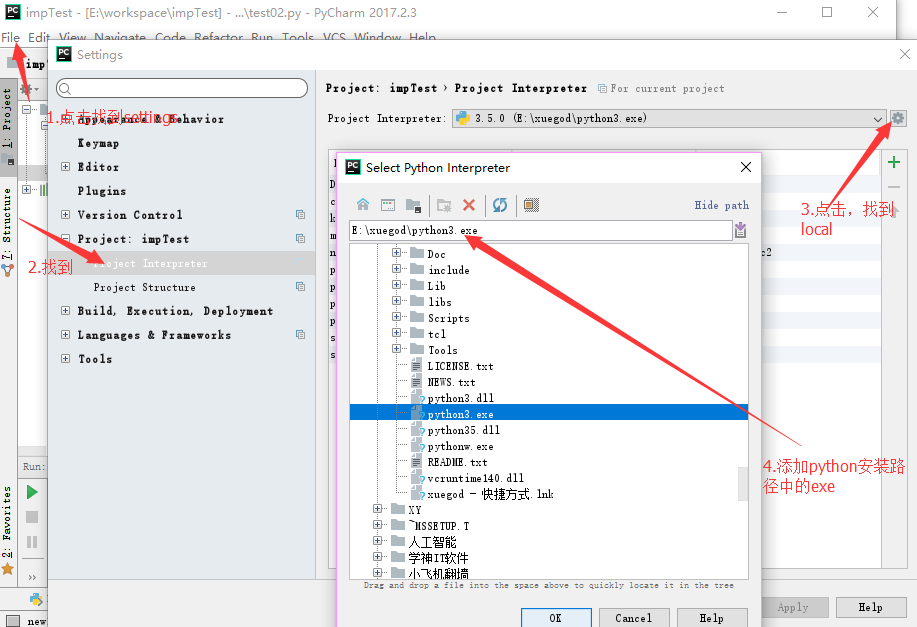


安装好pycharm后点开就会出现以上界面🡪点击Create New project （创建新的project）,



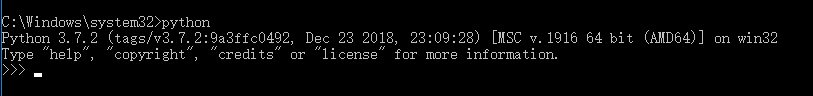
然后点击create，就会直接创建一个文件夹。然后就可以在里面创建相应的py文件。

配置python3.exe。运行项目时用python3运行。

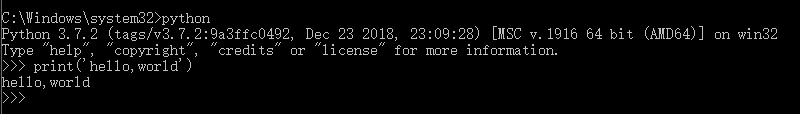


## 1.7 Python脚本运行

### 1.7.1 安装完成，打开命令窗口（键盘：win+R）：输入python3 注意：使用最高管理员权限打开cmd，否则后面操作安装模块时会出现各种错误。



打印Python的第一个程序hello world



更新pip

（在 Windows命令提示符(cmd)中输入以下命令进行 pip的更新:python -m pip install -U pip）

更新pip





趣味实战：  
print('\n'.join([''.join([('Love'[(x-y)%len('Love')]if ((x\*0.05)\*\*2+(y\*0.1)\*\*2-1)\*\*3-(x\*0.05)\*\*2\*(y\*0.1)\*\*3 <= 0 else ' ') for x in range(-30, 30)]) for y in range(30, -30, -1)]))

  
下载软件百度链接：

https://pan.baidu.com/s/1vItLQblghxHUnF\_oRcyXQA 密码：3ewm

总结：

* 1. Python介绍
  2. Python2与Python3的比较
  3. Python3的安装
  4. Python环境变量配置
  5. Python安装目录介绍
  6. Python脚本结构
  7. Python脚本运行