Python科学计算与数据处理

中国科学院大学
数学科学学院

本节目录

- 课程介绍
- Python简介
  - Python简史
  - Python的特征
  - Python的应用
  - Python 科学计算软件的选择

Python科学计算与数据处理

——课程介绍

课程介绍

- 通过课程的学习掌握用Python进行科学计算与数据处理的方法。

- 课程内容
  - Python 简介与Python 科学计算软件的选择。
  - Python 基础。
  - NumPy-快速处理数据。
  - SciPy-数值计算。
  - Matplotlib-绘制图表。
  - SymPy-符号运算。
  - Pandas-数据分析。

python8102@163.com　　密码：python456

参考资料

- Python程序设计语言
  - Python编程实践。
  - Python核心编程中文版（第二版）。
  - Python学习手册(第4版)。
- Python科学计算与数据处理
  - Python数据分析基础教程：NumPy学习指南
  - Python科学计算。
  - 利用Python进行数据分析。
  - OReilly Python for Finance, Analyze Big Financial Data (2015)

课程考核及方式

- 成绩的组成（暂定）

  平时作业与点名（40%）+读资料提交报告（60%）

- 参考资料报告

  与Python科学计算与数据处理相关的、课堂上未讲授的内容。（可参考百度深度学习资料）

- 评判报告方式

  报告同学互判，同学给的平均成绩占报告总分的80%。

百度深度学习资料

- 百度网盘链接：https://pan.baidu.com/s/1OEvJy8oYltcuyFykAHTnWg 密码：eu7v
- 飞桨官网：https://www.paddlepaddle.org.cn/ 有详细的资料，包括软件如何下载安装、API如何使用等等；
- 飞桨github：https://github.com/paddlepaddle/paddle
- 飞桨模型库：https://github.com/paddlepaddle/models

Python科学计算与数据处理

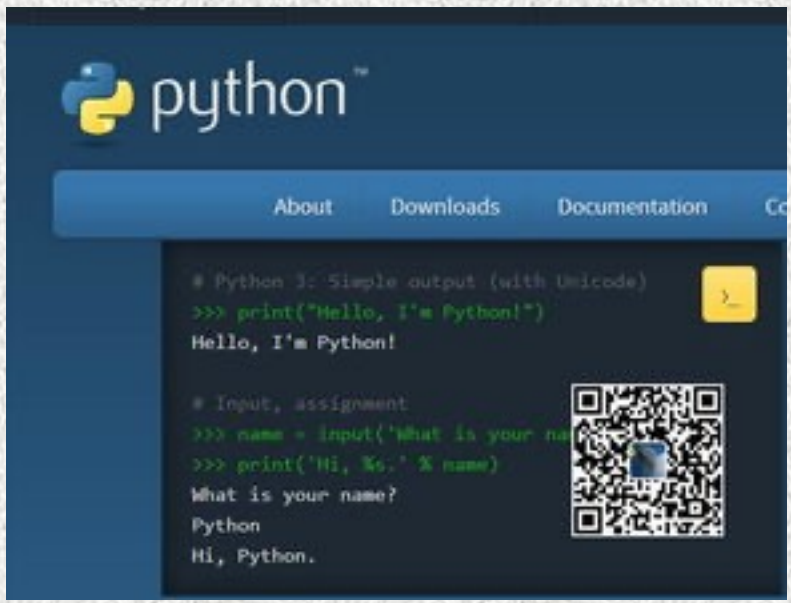
——Python简介

Python简史

- 什么是Python？
  - Python英语单词是蟒蛇的意思。
  - Python语言是少有的一种可以称得上既**简单**又**功能强大**的编程语言。
  - 注重的是如何**解决问题**而不是编程语言的**语法和结构**。

Python简史

- Python是一种简单易学，功能强大的编程语言，它有高效率的高层数据结构，简单而有效地实现面向对象编程。
- Python简洁的语法和对动态输入的支持，再加上解释性语言的本质，使得它在大多数平台上的许多领域都是一个理想的脚本语言，特别适用于快速的应用程序开发。



Python简史

Python的作者，吉多·范罗苏姆(Guido von Rossum)，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。

Python简史

Python语言诞生的时间是在1989年。在阿姆斯特丹，Guido在圣诞节家中正为ABC语言编写一个插件。ABC是由荷兰的数学和计算机研究所开发的，专为方便数学家、物理学家使用。Guido在该研究所工作，并参与到ABC语言的开发。

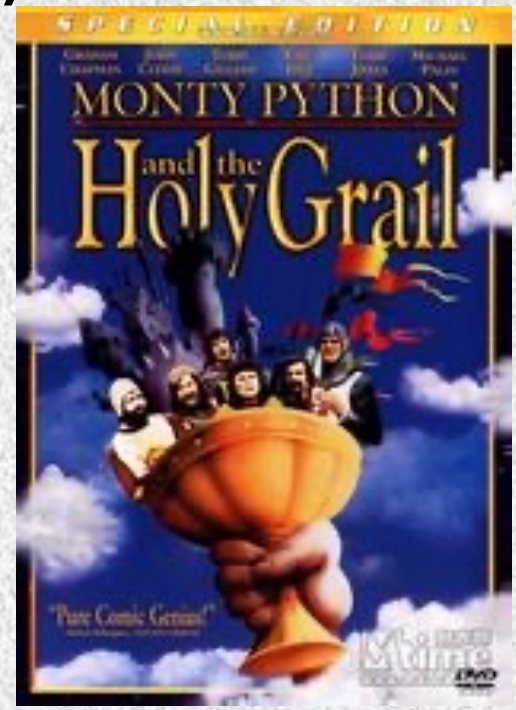
Python简史

Guido希望有一种语言能够像C语言那样，全面调用计算机的功能接口，同时又可以轻松的编程。ABC语言让Guido看到希望。ABC语言以教学为目的。ABC语言的目标是“让用户感觉更好”，希望让语言变得容易阅读，容易使用，容易记忆，容易学习，并以此来激发人们学习编程的兴趣。

在这个圣诞节假期，Guido开发的这个插件，实际实现了一个脚本语言，且功能强大。 Guido 以自己的名义发布了这门语言，且命名其为Python。

Python简史

Python英语单词的由来是因为Guido von Rossum是天空马戏团忠实的fans，用一个大蟒蛇飞行马戏团的名字中的一个单词“Python”作为这门新语言的名字。



Python简史

- Python的发展可经历几个重要的阶段：
  - CNRI时期：CNRI是资助Python发展初期的重要资助重要单位，Python1.5版前的主要成果大部分在此时期完成。
  - BeOpen时期：Guido von Rossum与BeOpen公司合作，Python1.6与Python2.0基本上同时推出，但原则上已经分别维护。Python2.0的许多功能与Python1.6不同。

Python简史

- DC时期：Guido离开BeOpen公司，将开发团队带到Digital Creations（DC）公司，该公司以发展Zope系统闻名，由于Guido的加入，因此这个项目也颇受关注。

- Python3.0：Python2.x和Python3.x差异挺大、前后不兼容，虽然有2to3的工具可以转换，但不能解决所有的问题。Python3.x尚未完全普及开来，很多第三方的库都没用官方支持Python3.x。考虑到前后版本的这个不兼容性，这会让一些人对采用Python开发项目产生顾忌。

Python简史

- 里程碑

Python由于在2010年获得较大市场份额的增长（1.81%，增长速度最快的）获年度Tiobe编程语言大奖。

Year	Winner
2010	Python
2009	Go
2008	C
2007	Python
2006	Ruby
2005	Java
2004	PHP
2003	C++

2020年获年度Tiobe编程语言大奖。

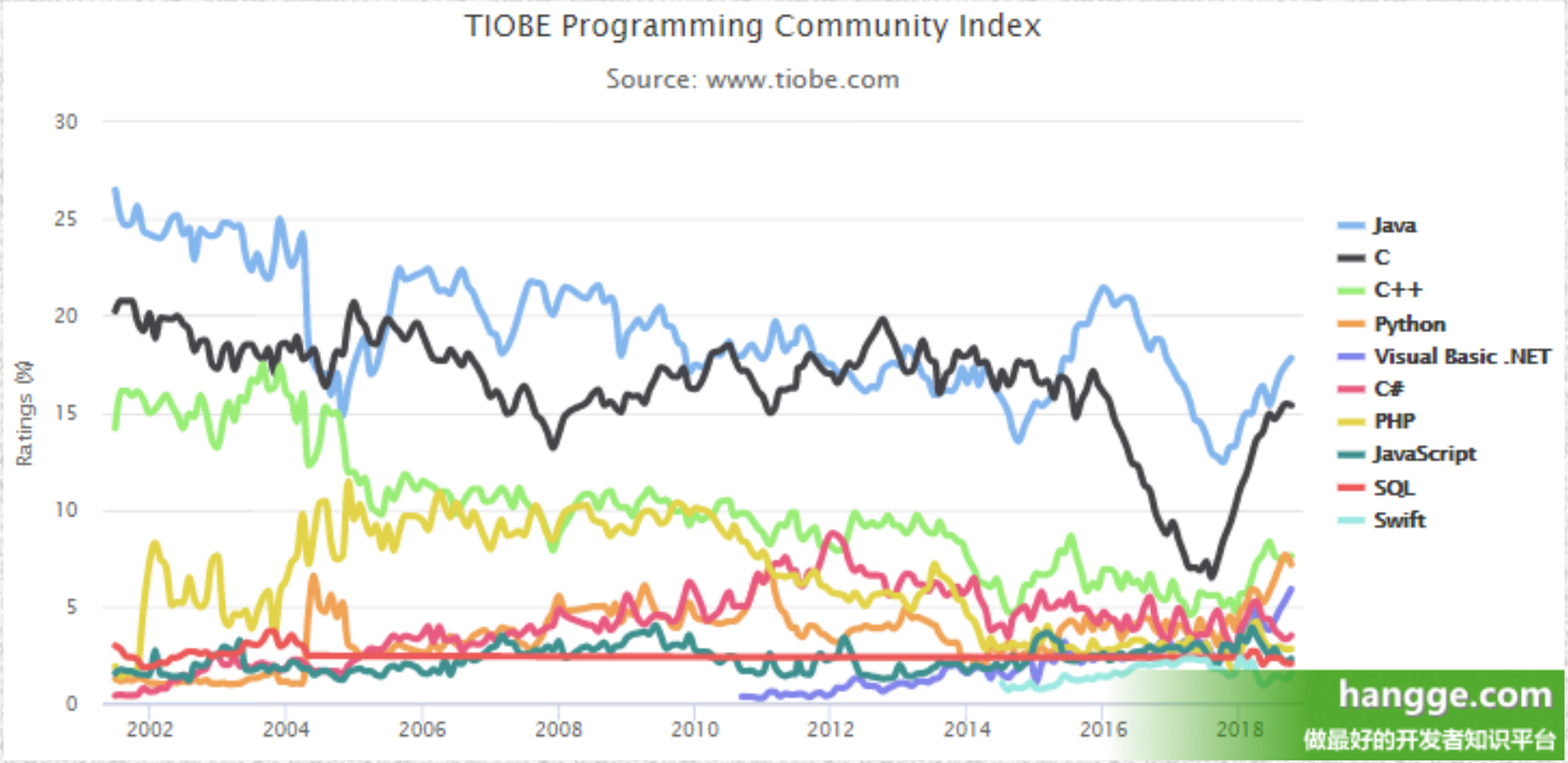
Python简史

Oct 2018	Oct 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.801%	+5.37%
2	2		C	15.376%	+7.00%
3	3		C++	7.593%	+2.59%
4	5	↑	Python	7.156%	+3.35%
5	8	↑	Visual Basic .NET	5.884%	+3.15%
6	4	↓	C#	3.485%	-0.37%
7	7		PHP	2.794%	+0.00%
8	6	↓	JavaScript	2.280%	-0.73%
9	-	⇈	SQL	2.038%	+2.04%
10	16	⇈	Swift	1.500%	-0.17%
11	13	↑	MATLAB	1.317%	-0.56%
12	20	⇈	Go	1.253%	-0.10%
13	9	⇊	Assembly language	1.245%	-1.13%
14	15	↑	R	1.214%	-0.47%
15	17	↑	Objective-C	1.202%	-0.31%
16	12	⇊	Perl	1.168%	-0.80%
17	11	⇊	Delphi/Object Pascal	1.154%	-1.03%
18	10	⇊	Ruby	1.108%	-1.22%
19	19		PL/SQL	0.779%	-0.63%
20	18	↓	Visual Basic	0.652%	-0.77%

Python简史

Programming Language	2018	2013	2008	2003	1998	1993	1988
Java	1	2	1	1	17	-	-
C	2	1	2	2	1	1	1
C++	3	4	3	3	2	2	5
Python	4	7	6	12	24	16	-
C#	5	5	7	9	-	-	-
Visual Basic .NET	6	13	-	-	-	-	-
JavaScript	7	10	8	7	21	-	-
PHP	8	6	4	5	-	-	-
Ruby	9	9	9	19	-	-	-
Perl	10	8	5	4	3	10	-
Objective-C	18	3	45	48	-	-	-
Ada	30	16	17	14	7	7	2
Lisp	31	12	15	13	6	4	3
Pascal	140	14	19	97	11	3	13

Python简史


TIOBE Programming Community Index
Source: www.tiobe.com
Ratings (%)
30
25
20
15
10
5
0
2002
2004
2006
2008
2010
2012
2014
2016
2018
Java
C
C++
Python
Visual Basic .NET
C#
PHP
JavaScript
SQL
Swift
hangge.com
做最好的开发者知识平台

Python简史

Dec 2021	Dec 2020	Change	Programming Language	Ratings	Change
1	3	↑	Python	12.90%	+0.69%
2	1	↓	C	11.80%	-4.69%
3	2	↓	Java	10.12%	-2.41%
4	4		C++	7.73%	+0.82%
5	5		C#	6.40%	+2.21%
6	**6**		**Visual Basic**	**5.40%**	**+1.48%**
7	7		JavaScript	2.30%	-0.06%
8	12	⇈	Assembly language	2.25%	+0.91%
9	10	↑	SQL	1.79%	+0.26%

本次 TIOBE 指数榜单前 10 位里，Python 已连续三个月霸榜第一；

Python的特征

Python语言的定位

- 脚本语言（Scripting language）

- 高阶动态编程语言

  Python不能与JavaScript等只能处理简单任务的编程语言相提并论。

Python的特征

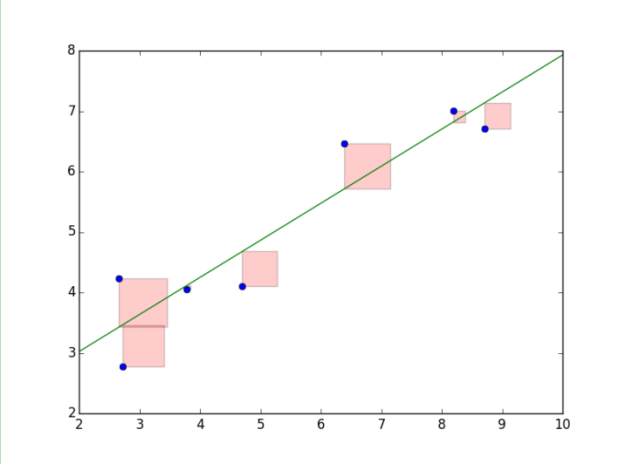
- 简单易学
  - 简洁：不用结束符
  - 可读性强：每一级缩进都是固定的若干个空格（如：4个空格）
  - 上手快：会其它语言的上手更快

Python上手容易，易学，读它的代码就像是在读文章。稍微有点逻辑思维的人只要看几分钟就能知道是什么意思。

从0开始学习编程的话，Python是一个不错的开始。

Python的特征

```
# 线性拟合-使用math
import math
def linefit(x , y):
    N = float(len(x))
    sx,sy,sxx,syy,sxy=0,0,0,0,0
    for i in range(0,int(N)):
        sx  += x[i]
        sy  += y[i]
        sxx += x[i]*x[i]
        syy += y[i]*y[i]
        sxy += x[i]*y[i]
    a = (sy*sx/N -sxy)/( sx*sx/N -sxx)
    b = (sy - a*sx)/N
    r = abs(sy*sx/N-sxy)/math.sqrt((sxx-sx*sx/N)*(syy-
sy*sy/N))
    return a,b,r
```



Python的特征

```
if __name__ == '__main__':
    X=[ 1 ,2  ,3 ,4 ,5 ,6]
    Y=[ 2.5 ,3.51 ,4.45 ,5.52 ,6.47 ,7.51]
    a,b,r=linefit(X,Y)
    print("X=",X)
    print("Y=",Y)
    print("拟合结果: y = %10.5f x + %10.5f , r=%10.5f" %
(a,b,r) )



#结果为：  %run "E:/2017python/线性拟合-使用math.py"
('X=', [1, 2, 3, 4, 5, 6])
('Y=', [2.5, 3.51, 4.45, 5.52, 6.47, 7.51])
拟合结果: y =    1.00000 x +    1.49333 , r=   0.99989
```

Python的特征

```
#线性拟合-使用numpy


import numpy as np
X=[ 1 ,2 ,3 ,4 ,5 ,6]
Y=[ 2.5 ,3.51 ,4.45 ,5.52 ,6.47 ,7.51]
z1 = np.polyfit(X, Y, 1) #一次多项式拟合，相当于线性拟合
p1 = np.poly1d(z1)
print z1 #[ 1. 1.49333333]
print p1 # 1 x + 1.493



#结果为%run "E:/2017python/codes/00/线性拟合-使用numpy.py"
[ 1.         1.49333333]

1 x + 1.493
```

Python的特征

```
#二次多项式拟合-使用numpy
import numpy

def polyfit(x, y, degree):
    results = {}
    coeffs = numpy.polyfit(x, y, degree)
    results['polynomial'] = coeffs.tolist()

    # r-squared
    p = numpy.poly1d(coeffs)
    # fit values, and mean
    yhat = p(x)                         # or [p(z) for z in x]
    ybar = numpy.sum(y)/len(y)          # or sum(y)/len(y)
    ssreg = numpy.sum((yhat-ybar)**2)   # or sum([ (yihat - ybar)**2 for yihat in yhat])
    sstot = numpy.sum((y - ybar)**2)    # or sum([ (yi - ybar)**2 for yi in y])
    results['determination'] = ssreg / sstot #准确率
    return results
```

Python的特征

```
x=[ 1 ,2  ,3 ,4 ,5 ,6]
y=[ 2.5 ,3.51 ,4.45 ,5.52 ,6.47 ,7.2]
z1 = polyfit(x, y, 2)
print z1

# %run "E:/2017python/codes/00/二次多项式拟合-使用numpy.py"
{'polynomial': [-0.024285714285714233, 1.125714285714285,
1.3700000000000017], 'determination': 0.9989301416685599}

x=[ 1 ,2  ,3 ,4 ,5 ,6]
y=[ 2.5 ,3.51 ,4.45 ,5.52 ,6.47 ,7.2]
z1 = numpy.polyfit(x, y, 2)

z1
array([-0.02428571,  1.12571429,  1.37      ])
```

Python的特征

- 解释性&编辑性
  - 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。
  - Python中也有编译执行的特性。

- 高级语言

  使用Python语言编写程序，无需考虑诸如管理内存一类的底层。

Python的特征

- 面向对象

  Python即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。

Python的特征

- 可扩展性与可嵌入性

  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。与此相反，可以把Python嵌入C/C++程序，提供脚本功能。

- 免费、开源

  可自由地发布这个软件的拷贝、阅读它的源代码、对它进行改动、把它的一部分用到新的自由软件中。

Python的特征

可移植性

由于它开源的本质，Python已被移植到许多平台上。如果能避免使用依赖系统的特性，那么所有的Python程序无需修改就可在任何平台上面运行。包括Linux，window，Macintosh等等。

胶水语言

可以调用别的语言编写的功能模块，将他们有机的结合在一起形成更高效的新程序。Python可以把C++、Java写的模块轻松结合起来协同工作，这样就能把c++的针对底层，java的面向对象两大优势统一到一个完整的程序中来。

Python的特征

- 丰富的库

丰富的库似乎已变成判断一种编程语言是否强大的重要标准。Python拥有一个强大的标准库。Python语言的核心只包含数字、字符串、列表、字典、文件等常见类型和函数，而由Python标准库提供了系统管理、网络通信、文本处理、科学计算、数据库接口、图形系统、XML处理等额外的功能。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。

Python的特征

- Python和其他语言的比较
  - 比TCL（ tool command language ）强大，支持“大规模编程”，适宜于开发大型系统
  - 比Perl语法简洁，更具可读性、更易于维护，有助于减少Bug
  - 比Java、C++更简单、更易于使用
  - 比VB更强大也更具备跨平台特性
  - 比Ruby更成熟、语法更具可读性

Python的特征

Python的语法特点

Python是一种语法表达极其优美的脚本语言

- 运行方式　命令行、交互式、图形集成环境
- 面向对象　甚至还支持异常处理
- 模块和包　与Java类似，还开发了Jpython
- 语言扩展　可以用C/C++/Java编写新的语言模块
- 丰富的库　数据分析/系统管理/web/GUI/… …

Python的特征

- Python的缺点

  运行速度不够快。

  Python程序运行的效率不如Java或者C代码高，但是我们可以使用Python调用C编译的代码。

- **开发速度与运行速度之间的矛盾**

  至今还没有一门编程语言，开发速度比Python快，运行速度比C快。

Python的应用

Python是用标准C语言写成的一种面向对象的脚本语言，语法相对简单，符合人的思维习惯，通过集成环境或解释器直接执行源程序。它可以运行在windows、linux等操作系统平台上，具有丰富的功能库以处理各种工作。

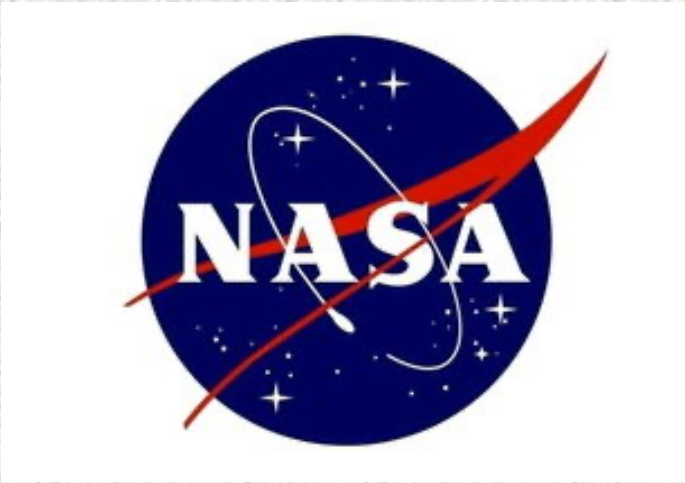
在编程领域的应用也日渐广泛，可以用于系统编程、图形处理、科学计算、文本处理、数据库编程、网络编程、多媒体编程等方面。也被一些公司应用于产品开发上。

Python的应用

实现Web爬虫和搜索引擎中的许多组件。

使用它（包括其它技术）管理讨论组。

Python的应用



完成科学计算任务。在它的几个系统中既用了Python开发，又将其作为脚本语言.

视频分享服务大部分是由Python编写的.

Python的应用

Python是豆瓣的主要开发语言

Python与Ruby比较：

Ruby用的人太少了。至少Python在当时中国有真正的使用者。从技术管理的角度来看，没有用Java已经有些不切实际了，用Ruby就太过理想主义了。

还有一个问题是库。Python有大量现成的库，而且很多库都经过大型商业应用。

Python更简洁。

Ruby 是日本人创建的！不能保证其技术支持性！！ Python 是世界的！

Python科学计算发行版的比较

- Python环境的搭建

  http://www.python.org/download/

  python2.7.x同python3.x比较改了不少地方。

  http://www.compileonline.com/execute_python_online.php

  ？？一个在线的python运行环境，可在这里练习，无需下载安装配置。左边页面是写代码的地方，点击左上角的“Execute Sctipt”，就可以在右边页面看到输出结果。

Python科学计算发行版的比较

Python是一门优秀的程序语言，其还拥有出色的数据处理能力，尤其是在数据量巨大的时候，因而也吸引了不少数据分析人员的关注和使用。

一般的Python数据分析并不直接在Python shell中运行代码，而是选择了IPython，IPython 是一个 python 的交互式 shell，比传统的Python shell 好用，支持变量自动补全，自动缩进，支持 bash shell 命令，内置了许多很有用的功能和函数。

Python科学计算发行版的比较

Python的数据处理能力主要依赖于NumPy,SciPy,Matplotlib,Pandas这4个库，其中NumPy提供了矩阵运算的功能，SciPy则在NumPy的基础上添加了许多科学计算的函数库，而这两个库就使Python具有和Matlab一样的数据处理能力了。Matplotlib库提供了绘图，可以实现数据的可视化，pandas是基于NumPy的一种工具，该库提供了高效地操作大型数据集所需的工具。而这四个库都需要我们进行单独安装，Python自身并不具备这些库。

Python科学计算发行版的比较

Ipython及各种科学计算库的安装及升级更新较为麻烦和复杂，幸好有大神将科学计算所需要的模块以及IPython打包供用户使用。

Python科学计算发行版的比较

- **Python（x，y）**

  GUI基于PyQt，曾经是功能最全也是最强大的，而且是Windows系统中科学计算免费Python发行版的不二选择.不过今时已不同往昔! Python（x，y）里面的许多科学计算包部分有兼容性的问题，无法使用最新的程序包。

Python科学计算发行版的比较

- **WinPython**

  WinPython功能也是比较全的,软件包比较新，GUI基于PyQt，不过相对于Python（x，y），它主要是关注便携式安装体验：你可以把它装在u盘里面。

Python科学计算发行版的比较

- **Anaconda**

  Anaconda Python 是完全免费的企业级的Python发行大规模数据处理、预测分析和科学计算工具。

  Anaconda 是 Python 科学技术包的合集，功能和 Python(x,y) 类似。包管理使用 conda，GUI基于PySide，容量适中，但该有的科学计算包都有。Anaconda 支持所有操作系统平台，它的安装、更新和删除都很方便,且所有的东西都只安装在一个目录中。

Python科学计算发行版的比较

Anaconda目前提供 Python 2.X,Python 3.X系列发行包，这也是其他发行版所望尘莫及的。

简言之，安装了Anaconda，你就安装了Python+NumPy+SciPy+Matplotlib+IPython+IPython Notebook。所以，我们仅仅安装Anaconda就可以了！

Anaconda下载地址：
https://www.anaconda.com/distribution/#download-section

Anaconda+PyCharm安装

Python科学计算发行版的比较

- **Enthought Canopy**

  GUI基于wxpython,包含PySide。Canopy有自己的集成开发环境（IDE），里面的代码智能提示和自动补全功能不比IPython差。

  Canopy是Enthought公司开发的一款Python集成开发环境，之前的版本叫EPD，附带了超过50个Python模块，包括numpy、scipy、panda、matplotlib等常用模块，同时提供免费版和供科研使用的学术版。

Python科学计算发行版的比较

学术版可享受完整版的所有功能，只需验证一个教育邮箱，便可享受Canopy所有的功能和服务。

在注册并完成教育邮箱的验证后，便可下载Canopy。Canopy支持Window、Linux和Mac平台，并提供32位和64位系统的安装包。

https://store.enthought.com/downloads/#default

Python科学计算发行版的比较

在软件中登陆Enthought上注册的用户名和密码，会出现一个”Training on Demand“图标，点击这个图标便可在网上学习Canopy自带的Python学习教程，其中包括：Introduction to Python、NumPy、Advanced Python、SciPy、Interfacing with other languages五个教程，教程的质量相当高，同时提供用Ipython Notebook编写的文档可供练习，且会不断加入新的专题，是一笔不可多得的入门好资源！

Python科学计算发行版的比较

同时，Canopy提供一键升级Package的功能，可以根据需要，方便快捷地安装和管理各个Package。

Python科学计算发行版的比较

软件选择和推荐

- Python(x,y)和WinPython都是开源项目，其项目负责人都是Pierre Raybaut。

- Canopy和Anaconda是公司推的，带免费版和商业版/插件。这两款发行版也牵扯到一个人，那就是Travis Oliphant。Travis是SciPy的原始作者，同时也是NumPy的贡献者。Travis在2008年以副总裁身份加入Enthought，2012年以总裁的身份离开，创立了一个新公司continuum.io，并推出了Python的科学计算平台Anaconda。

Python科学计算发行版的比较

- Anaconda的开发和维护中有Python创始人和社区的核心成员。因此在各种操作系统中，无论是Linux，还是Windows、Mac都推荐Anaconda！
- Canopy的性能和稳定性超强!也提供免费的free版本和学术版本（用于教育科研也是免费的）。

- Python用于科学计算的一些常用工具和库
  - NumPy-数学计算基础库：N维数组、线性代数计算、傅立叶变换、随机数等。
  - SciPy-数值计算库：线性代数、拟合与优化、插值、数值积分、稀疏矩阵、图像处理、统计等。
  - SymPy-符号运算
  - Pandas-数据分析库：数据导入、整理、处理、分析等。
  - Matplotlib-绘图库：绘制二维图形和图表。
  - scikit-learn: Machine Learning in Python
  - Beautiful soup：爬虫工具

- nltk：Natural Language Toolkit
- Chaco-交互式图表
- TVTK-数据的三维可视化
- Mayavi-更方便的可视化
- VPython-制作3D演示动画
- OpenCV-图像处理和计算机视觉
- Cython-Python转C的编译器：编写高效运算扩展库的首选工具
- BioPython-生物科学

附： Canopy简单操作

- 设置路径

```
In [1]:import os      #导入os的标准库
In [2]:os.getcwd() #查看现在的工作目录
Out[2]: 'C:\\WinPython-64bit-
    3.5.1.1\\python-3.5.1.amd64\\Scripts'
In [3]: os.chdir('E:\python') #设置工作目录
    是‘E:\\python’，且此目录必须已存在。
In [4]:os.getcwd()
Out[4]: 'E:\\python’
```

Ipython操作简介

- Ipython 是一个 python 的交互式 shell，比默认的Python shell 好用得多，支持变量自动补全（ Tab 键），自动缩进，支持bash shell 命令，内置了许多很有用的功能和函数。
- Ipython 中运行编写好的py脚本文件：

  run test.py　　!python test.py

  IPython 中用！表示执行 **shell** 命令，用 $ 将 python 的变量转化成shell 变量。通过这种两个符号，就可以做到和 shell命令之间的交互，可以非常方便地做许多复杂的工作。

Ipython操作简介

- Ipython功能介绍资料：

help(len)； len? object特性的介绍和概述，

这种表示方式几乎适用于一切，包括对象方法：

ss1 = [1, 2, 3, 4, 5]; ss1.insert? ； ss1?

也适用于你自己创建的函数或其它对象：

```
def square(a):
    """Return the square of a."""
    return a**2
```

square? 代码中内嵌文档，可在docstring获取。

square?? 可通过？？获取源代码。

len?? 若查询对象不是用python实现的，如C或其它编译扩展语言实现，则？？后缀等同？后缀。

Ipython操作简介

- Ipython shell执行过程记录: In[] Out[]

  In [] 记录用户输入命令

  Out [] 记录 变量、命令输出信息。

```
import math

math.sin(2)
Out[17]: 0.9092974268256817

math.cos(2)
Out[18]: -0.4161468365471424

print(In)

Out
```

```
print(In[17])
math.sin(2)

print(Out[18])
-0.416146836547

Out[17]**2+Out[18]**2
Out[24]: 1.0
```

Ipython操作简介

- Ipython魔法命令：Ipython在普通python语法基础之上的增强功能。单行用%，多行用%%。

  %quickref：一份手册，包含了所有的命令

  %magic【%lsmagic】，[ipython特性的介绍和概述，快速查看可使用魔力函数(内置的一些方法)]

Ipython操作简介

- 几个简单好用的 **magic函数:**

在IPython里面可以使用一些标准unix命令, 比如cd,pwd等...,其实这些unix命令是IPython的**magic commands**, 这些magic commands一般用%作为前缀.

%run, 运行一个.py脚本, 运行完了后这个.py文件里的变量都可以在Ipython里继续访问.

%timeit, 测试一个命令(或者一个函数)的运行时间

```
%timeit [x*x for x in range(1000)]
1000 loops, best of 3: 221 µs per loop

%timeit [x*x for x in xrange(1000)]
1000 loops, best of 3: 198 µs per loop
```

Ipython操作简介

%ed 或 %edit编辑一个文件并执行。

%env显示环境变量。

%history显示历史记录。

%pwd显示当前目录。

%pycat filename用语法高亮显示一个 python 文件(不用加.py后缀名)。

%save filename 将执行过的代码保存为文件。

%pylab 可以查看代码中 import 了哪些包。。。

```
%pylab inline
Populating the interactive namespace from numpy and matplotlib
In [35]: x=linspace(-2,2,10)
In [36]: plot(x,sin(x))
```