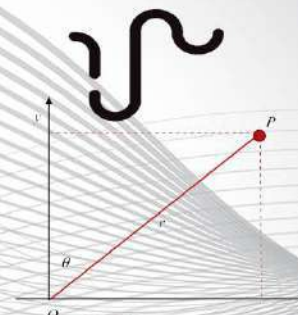
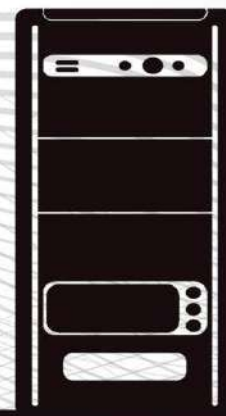
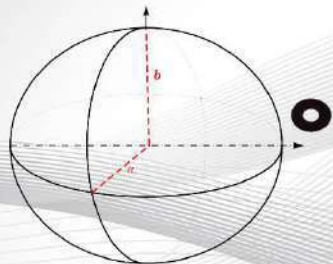


JAVA
+ xml
+ c#



1024

(X,Y,H)

cout << " 测绘软件设计与实现"

第10章 Python程序开发

朱晓峻

Email: zhuxiaojunahu@126.com

QQ: 302838249

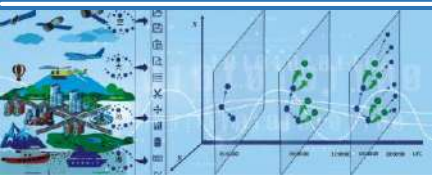
安徽大学 资源与环境工程学院

学习内容

Contents

测绘程序设计

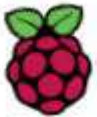
- ★ Python语言基础
- ★ Python的学习资料
- ★ Python安装与设置
- ★ Exercises 上机练习
- ★ 基于Python平差程序
- ★ Python网络爬虫
- ★ 人工智能程序开发



Python语言基础



Python是什么？

- Python是一种**解释性**编程语言，简单易学，功能强大，适合快速开发。
- 有高效率的**高层数据结构**，能够简单、有效地实现**面向对象**编程。
- 提高生产力、降低维护成本
- 跨平台，适合于移动平台的开发(**Raspberry Pi**) 
- Python是**人工智能**Artificial Intelligence首选编程语言



Python的来源

Guido van
Rossum

吉多·范罗苏
姆

(1956 -)

Python的创始人Guido van Rossum。1989年圣诞节期间，在阿姆斯特丹，Guido**为了打发圣诞节的无趣**，决心开发一个新的脚本解释程序，作为ABC语言的一种继承。所以选中Python（大蟒蛇的意思）命名是因为他是一个叫Monty Python的喜剧团体的爱好者。





PEP 572 -- Python 之父宣布退位

[python-committers] Transfer of power

Guido van Rossum guido@python.org

Thu Jul 12 10:57:35 EDT 2018

- Previous message (by thread): [\[python-committers\] A different way to focus discussions](#)
- Next message (by thread): [\[python-committers\] Transfer of power](#)
- **Messages sorted by:** [\[date\]](#) [\[thread\]](#) [\[subject\]](#) [\[author\]](#)

Now that PEP 572 is done, I don't ever want to have to fight so hard for a PEP and find that so many people despise my decisions.

I would like to remove myself entirely from the decision process. I'll still be there for a while as an ordinary core dev, and I'll still be available to mentor people — possibly more available. But I'm basically giving myself a permanent vacation from being BDFL, and you all will be on your own.

After all that's eventually going to happen regardless — there's still that bus lurking around the corner, and I'm not getting younger... (I'll spare you the list of medical issues.)

I am not going to appoint a successor.

So what are you all going to do? Create a democracy? Anarchy? A dictatorship? A federation?



PEP: Python Enhancement Proposals

增强建议书

PEP 8 -- Style Guide for Python Code

PEP:	8
Title:	Style Guide for Python Code
Author:	Guido van Rossum <guido at python.org>, Barry Warsaw <barry at python.org>, Nick Coghlan <ncoghlan at gmail.com>
Status:	Active

Type:
Created:
Post-
History:

PEP 20 -- The Zen of Python

PEP:	20
Title:	The Zen of Python
Author:	tim.peters at gmail.com (Tim Peters)
Status:	Active
Type:	Informational
Created:	19-Aug-2004
Post-History:	22-Aug-2004



世界编程语言排行榜 2018年度语言

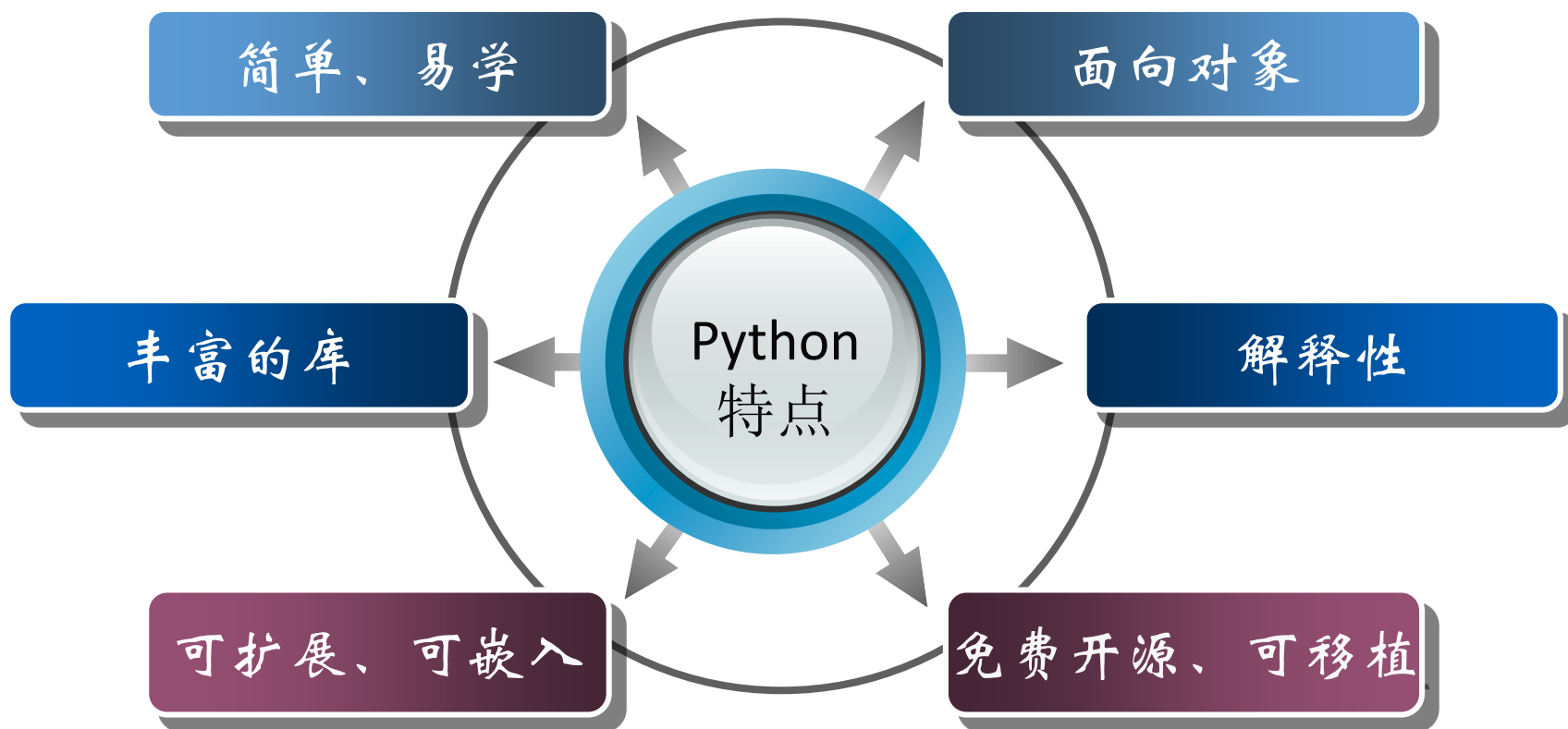
- TIOBE排行榜

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲▲	SQL	2.062%	+2.06%
10	18	▲▲	Objective-C	1.509%	+0.00%



Python语言特点

- 实际需要；高效、易懂；众多优点集于一身
 - （人生苦短，我用Python！）



Python语言特点

Python的特点

1) 简单、易学

Python是一种代表简单主义思想的语言，有简单的语法，容易上手。

Python的这种伪代码本质是它最大的优点之一。

Python使你能够专注于解决问题而不是去搞明白语言本身。

2) 面向对象的高级语言

无需关注底层细节，而C/C++中需要操作指针。

与其他语言相比，Python以强大而又简单的方式实现面向对象编程。

3) 解释性

Python程序不需要编译成二进制代码，可以直接在源代码上运行。

对于编译性语言（C/C++），源文件->编译/链接器->可执行文件。

Python语言特点

Python的特点

4) 免费开源，可移植性

Linux, Windows, Androids/iOS, 游戏机PSP等等。

5) 可扩展性，可嵌入性

如果一段关键代码希望运行得更快或者希望算法不公开，你可以把这部分程序用C或C++编写，然后在Python程序中使用它们。

你可以把Python嵌入到C/C++程序，从而向程序用户提供脚本功能。

6) 丰富的库

Python标准库确实很庞大，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、等等。

此外，还有其他高质量的库。



The Zen of Python

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```



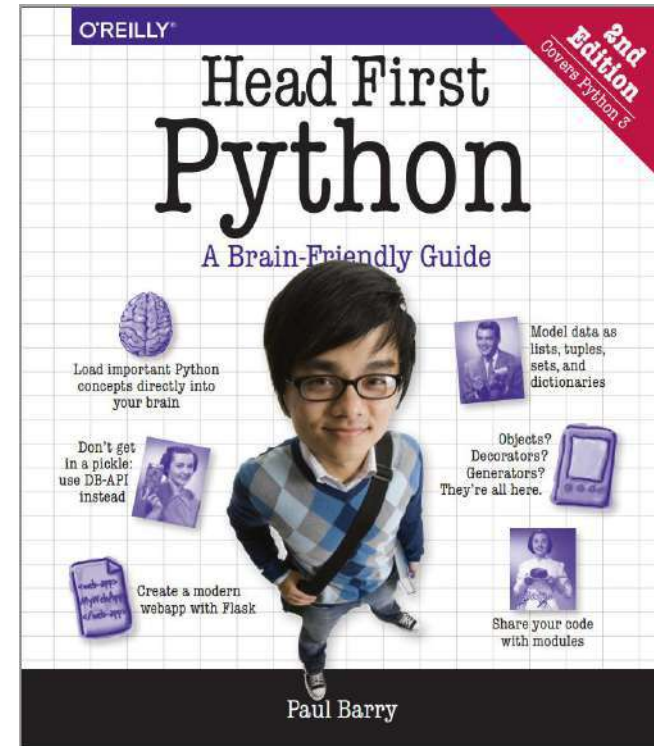
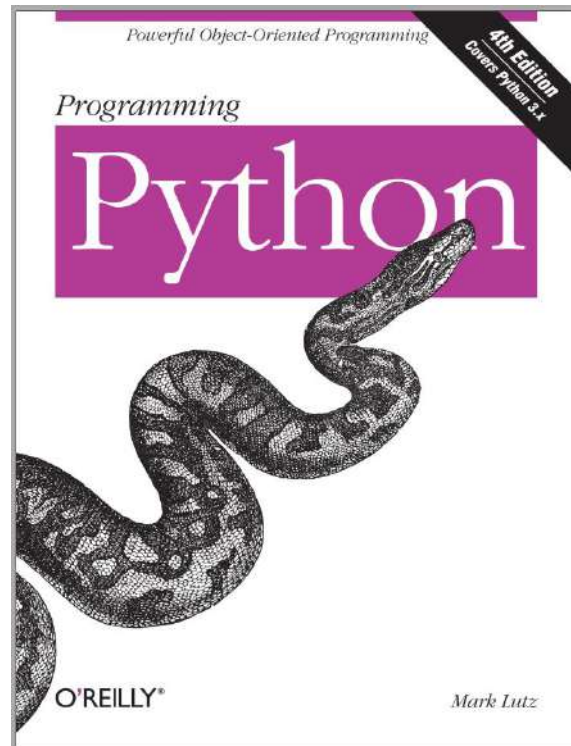
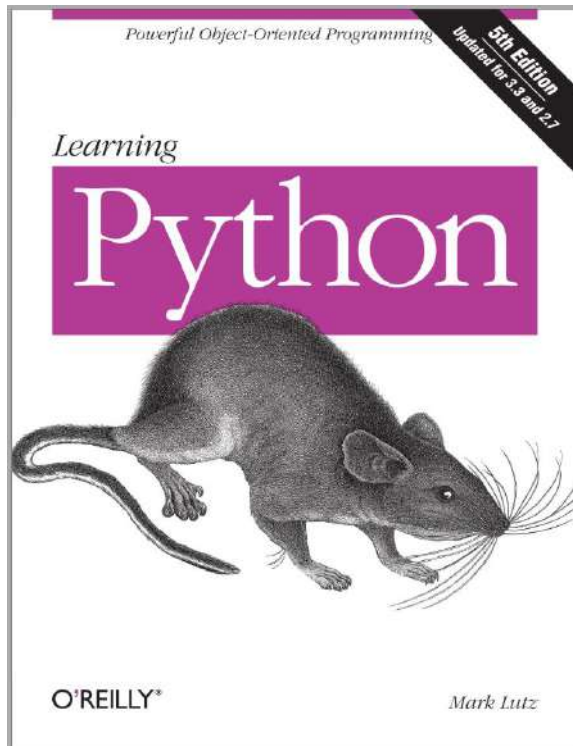
Python之禅

- 优美胜于丑陋
- 明了胜于晦涩
- 简洁胜于复杂
- 复杂胜于凌乱
- 扁平胜于嵌套
- 间隔胜于紧凑
- 可读性很重要

Python的学习资料

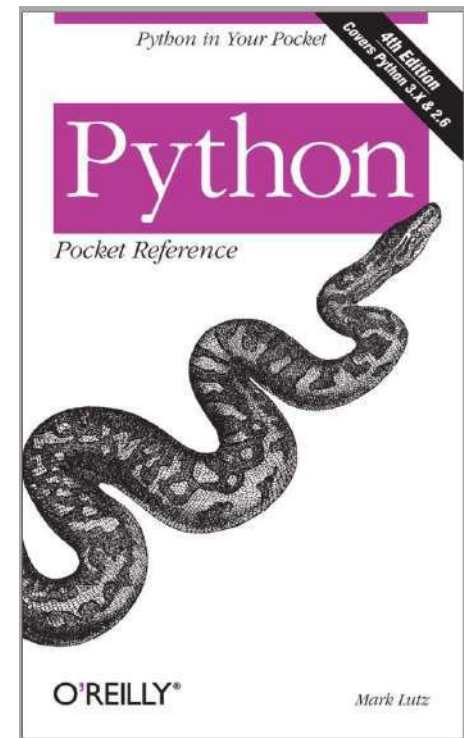
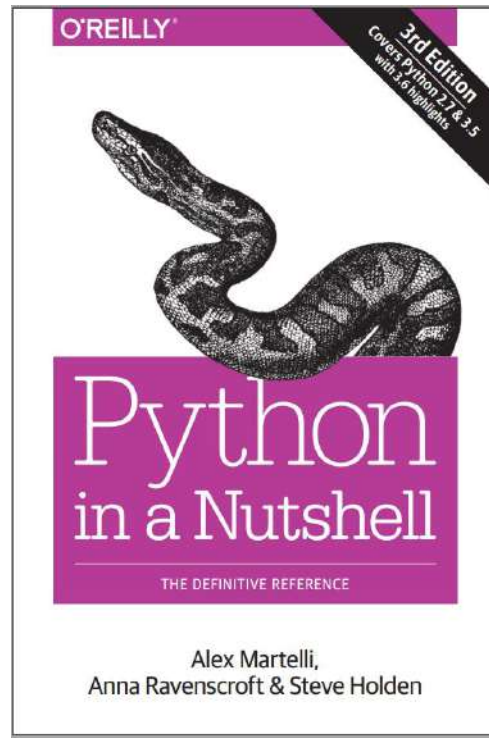
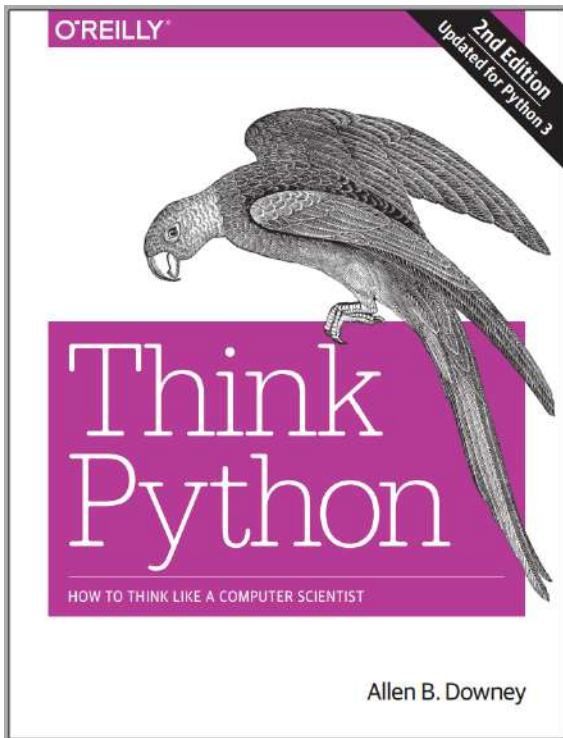


语言基础



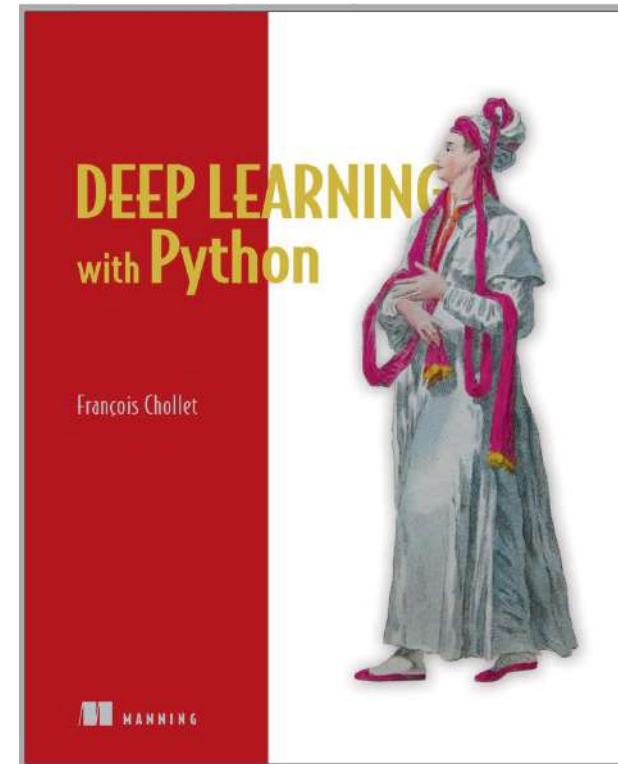
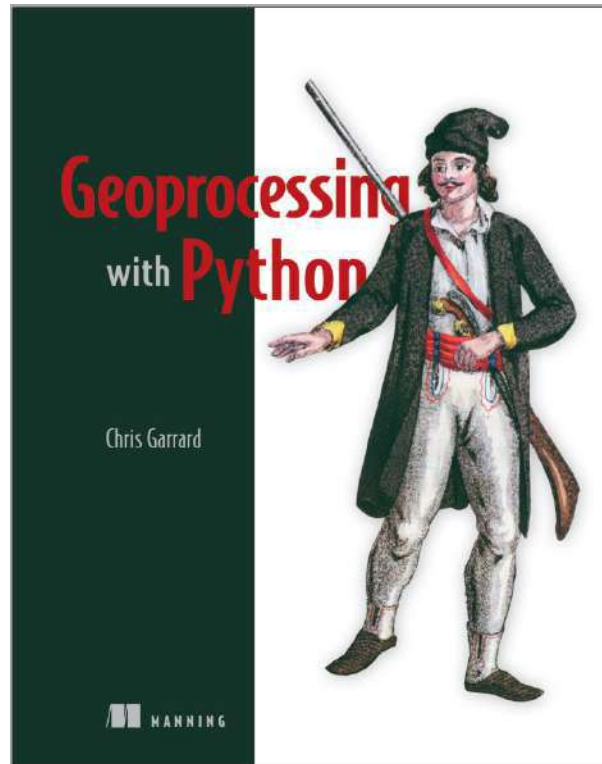
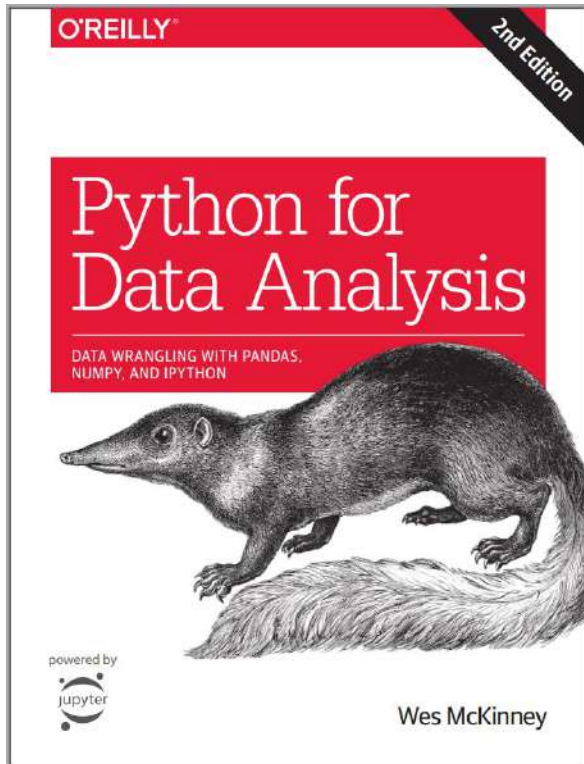


语言进阶



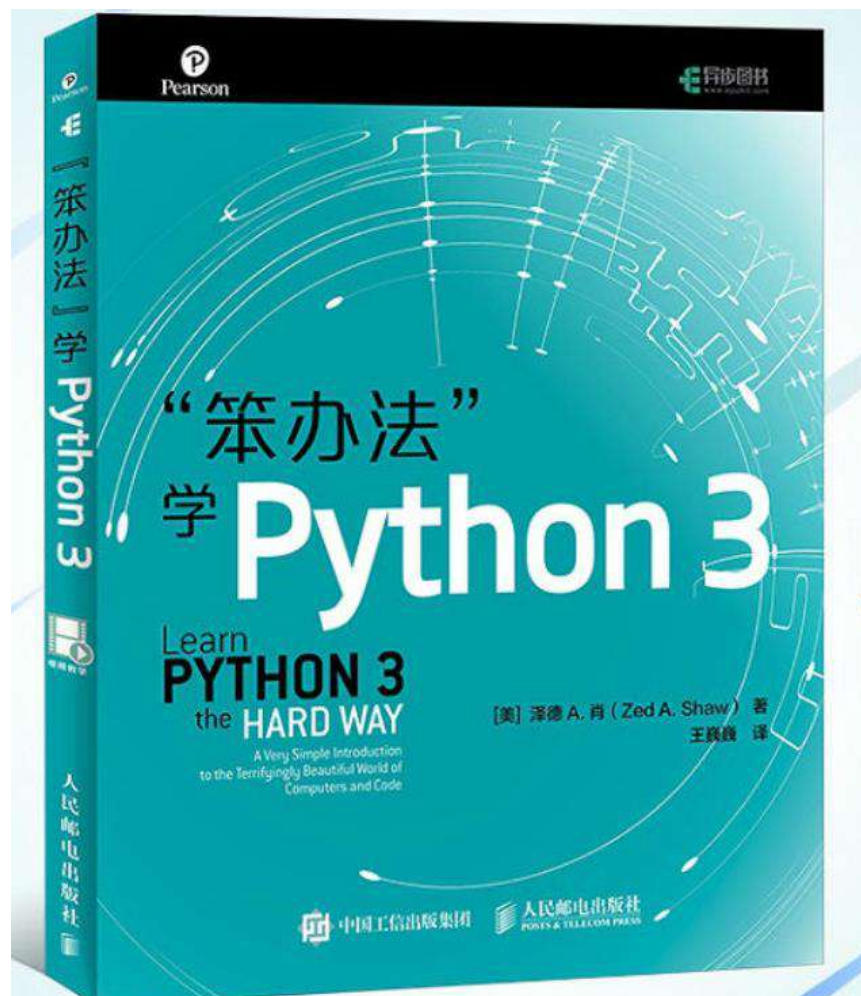
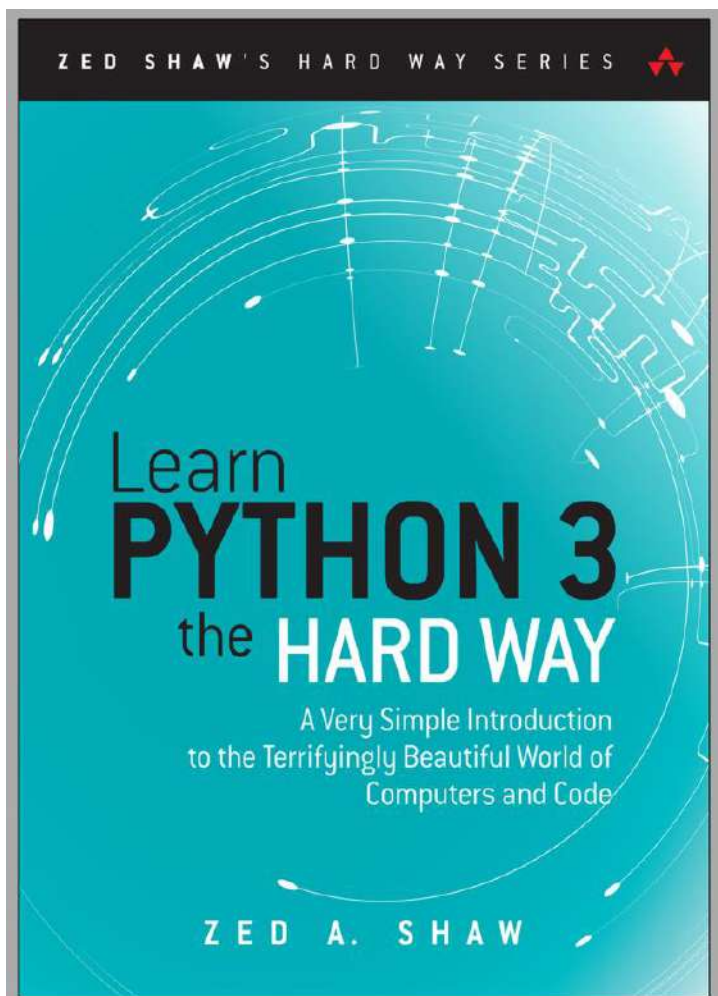


专业书籍



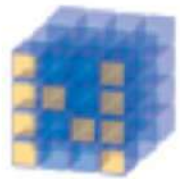


本实验课参考书





Python核心科学库



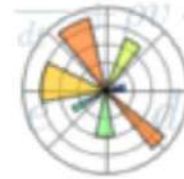
NumPy

Base N-dimensional
array package



SciPy library

Fundamental library
for scientific
computing



Matplotlib

Comprehensive 2D
Plotting

IP[y]:
IPython

IPython



Enhanced
Interactive Console



Sympy

Symbolic
mathematics



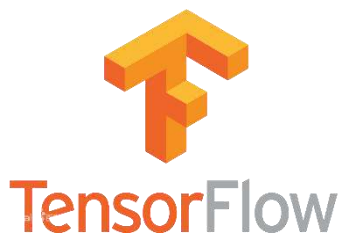
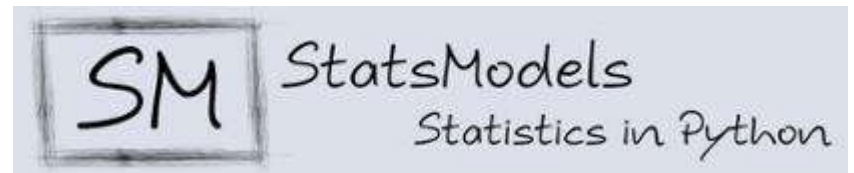
pandas

Data structures &
analysis



scikit-learn

machine learning in Python



Keras





Numpy:科学计算的基本包



它包含以下内容：

- 一个强大的N维数组对象： `ndarray`
- 复杂的函数功能
- 用于集成C / C ++和Fortran代码的工具
- 有用的线性代数， 傅里叶变换和随机数能力

NumPy也可以用作通用数据的高效多维容器。 可以定义任意数据类型。 这使得NumPy能够与各种数据库无缝并快速地整合。



SciPy:科学计算的基础库

用于解决科学计算中的许多不同领域标准问题的一组软件包

- `scipy.integrate` 数值积分、微分方程
- `scipy.linalg` 线性代数
- `scipy.optimize` 函数优化、根查找
- `scipy.signal` 信号处理工具
- `scipy.sparse` 稀疏矩阵、稀疏线性系统求解器
- `scipy.special` Fortran库，实现许多常用的数学函数
- `scipy.stats` 统计分析工具

NumPy和SciPy一起构成了一个相当完整和成熟的计算许多传统科学计算应用的基础。



- SymPy是符号数学的Python库。它旨在成为一个全功能的计算机代数系统(Computer algebra system, CAS), 同时保持代码尽可能简单, 以便于理解和扩展。
- SymPy完全是用Python编写的。

$$f''(x) - 2f'(x) + f(x) = \sin(x)$$

```
>>> diffeq = Eq(f(x).diff(x, x) - 2*f(x).diff(x) + f(x), sin(x))
```

```
>>> diffeq
```

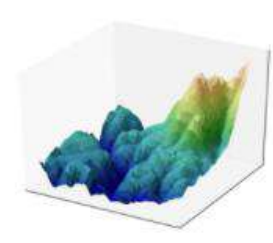
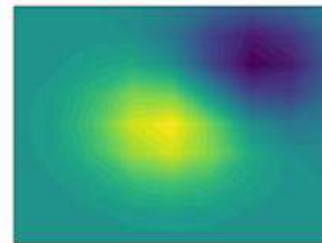
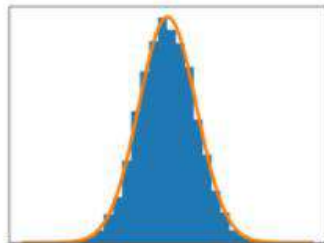
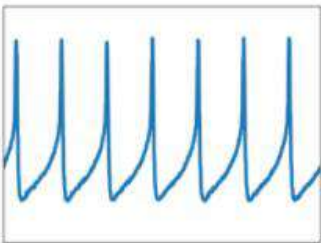
$$f(x) - 2 \cdot \frac{d}{dx}(f(x)) + \frac{d^2}{dx^2}(f(x)) = \sin(x)$$



matplotlib

Version 2.2.2

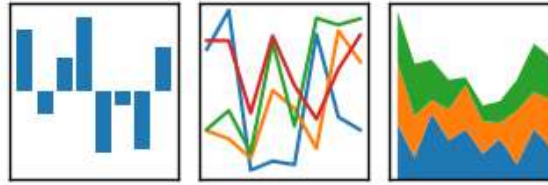
- matplotlib是用于生成绘图和其他二维数据可视化的最流行的Python库。
- 它最初由John D. Hunter创建，现在由一个大型开发团队维护。它旨在创建适合发布的绘图。
- 与生态系统的其他部分通常有良好的集成，默认的可可视化工具。





pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Python Data Analysis Library

Python数据分析库

- 带标记轴的数据结构支持自动或显式数据对齐
- 集成的时间序列功能
- 相同的数据结构时间序列数据和非时间序列数据
- 保存元数据的算术运算和简化
- 灵活处理丢失的数据
- 流行数据库中的合并和其他关系操作（如基于SQL的）

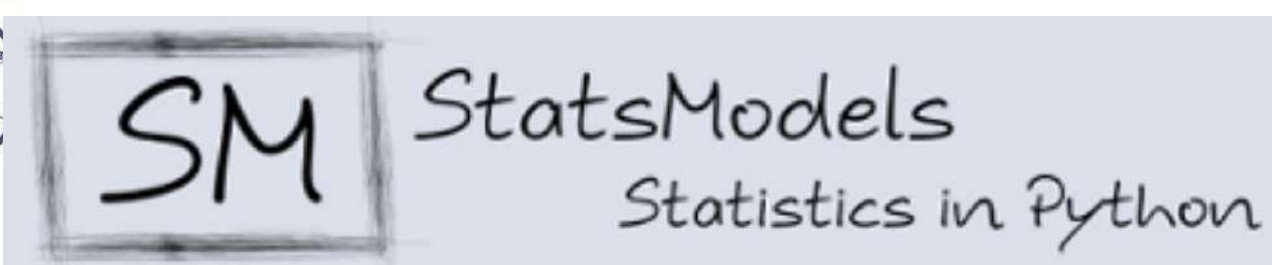


scikit-learn

Machine Learning in Python

简单高效的数据挖掘和数据分析工具，基于 NumPy, SciPy和matplotlib，开源

- 分类：支持向量机SVM，最近邻居，随机森林，逻辑回归等。
- 回归：套索，岭回归等。
- 聚类：k均值，谱聚类等。
- 降维：PCA，特征选择，矩阵分解等。
- 模型选择：网格搜索，交叉验证，度量
- 预处理：特征提取，归一化



- 回归模型：线性回归，广义线性模型，强健线性模型，线性混合效应模型等。
- 方差分析（ANOVA）
- 时间序列分析：AR，ARMA，ARIMA，VAR和其他模型
- 非参数方法：核密度估计，核回归
- 统计模型结果的可视化



IP[y]: IPython
Interactive Computing

 Jupyter

IPython提供了一个丰富的交互式计算架构:

- 强大的交互式外壳。
- Jupyter的内核。
- 支持交互式数据可视化和使用GUI工具包。
- 灵活的可嵌入式解释器可加载到您自己的项目中。
- 易于使用的高性能并行计算工具。

Jupyter Notebook是一款开放源代码的Web应用程序，允许您创建和共享包含实时代码，方程式，可视化和叙述文本的文档。

用途包括：数据清理和转换，数值模拟，统计建模，数据可视化，机器学习等等。



TensorFlow™

Install

Develop

API r1.7

Deploy

Extend

Community

Versions

TFRC

🔍 搜索

GITHUB

An open-source machine learning framework for everyone

开始使用

TensorFlow Dev Summit 2018

请预留 TensorFlow Dev Summit 2018 的时间！名额有限 - [点击订阅活动更新](#)。

快速

在构建和部署机器学习系统时，性能是至关重要的。因此，TensorFlow 中包含了 XLA，这是一款强大的线性代数编译器，可以帮助 TensorFlow 代码在嵌入式处理器、CPU、GPU、TPU 和其他硬件平台上尽可能快速地运行。

灵活

TensorFlow 既提供了高层级的 API 以便让您轻松构建和训练您的模型，也提供了低层级的控制功能以尽可能提高灵活性和性能。

支持生产环境

TensorFlow 可以灵活适应不同的使用规模，既支持探索性研究，也支持大规模生产用途。无论您是在开发新类型的模型还是在处理生产环境中的数以百万计的请求，所使用的 TensorFlow API 是不变的，因而您不会有陌生感。

最新动态

TensorFlow 简介

TensorFlow™ 是一个使用数据流图进行数值计算的开源软件库。图中的节点代表数学运算，而图中的边则代表在这些节点之间传递的多维数组（张量）。这种灵活的架构可让您使用一个 API 将计算工作部署到桌面设备、服务器或者移动设备中的一个或多个 CPU 或 GPU。TensorFlow 最初是由 Google 机器智能研究部门的 Google Brain 团队中的研究人员和工程师开发的，用于进行机器学习和深度学习研究，但它是一个非常基础的系统，因此也可以应用于众多其他领域。





Keras: The Python Deep Learning library



You have just found Keras.

Keras is a high-level neural networks API, written in Python and capable of running on top of [TensorFlow](#), [CNTK](#), or [Theano](#). It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.
- Runs seamlessly on CPU and GPU.

Read the documentation at [Keras.io](#).

Keras is compatible with: **Python 2.7-3.6**.

Guiding principles

- **User friendliness.** Keras is an API designed for human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.
- **Modularity.** A model is understood as a sequence or a graph of standalone, fully-configurable modules that can be plugged together with as little restrictions as possible. In particular, neural layers, cost functions, optimizers,



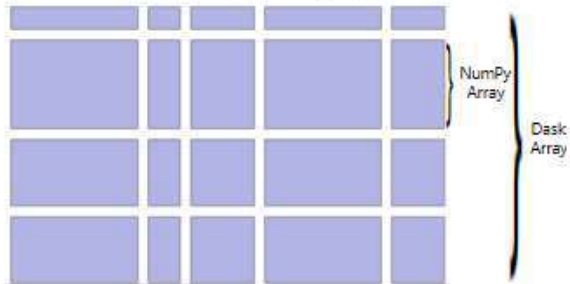
Dask: 综合数据分析

Integrates with existing projects

BUILT WITH THE BROADER COMMUNITY

Dask is open source and freely available. It is developed in coordination with other community projects like Numpy, Pandas, and Scikit-Learn.

Numpy

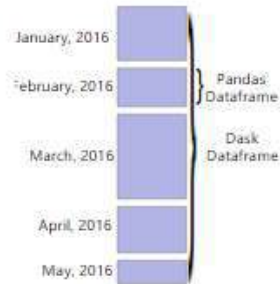


Dask arrays scale Numpy workflows, enabling multi-dimensional data analysis in earth science, satellite imagery, genomics, biomedical applications, and machine learning algorithms.

[Learn More »](#)

[Try Now »](#)

Pandas

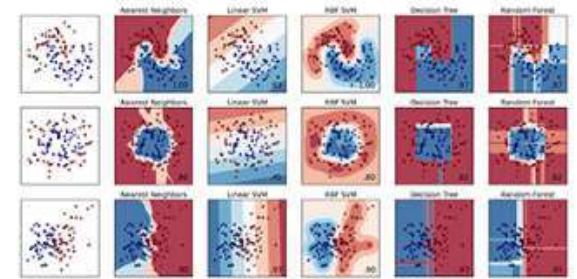


Dask dataframes scale Pandas workflows, enabling applications in time series, business intelligence, and general data munging on big data.

[Learn More »](#)

[Try Now »](#)

Scikit-Learn



Dask-ML scales machine learning APIs like Scikit-Learn and XGBoost to enable scalable training and prediction on large models and large datasets.

[Learn More »](#)

[Try Now »](#)

Python安装与设置



Python的下载

- 官方源 <https://www.python.org/downloads/>

Download the latest version for Windows

Download Python 3.7.3

- 官方包PyPI - the Python Package Index
- 包管理工具pip <https://pypi.python.org/pypi/pip>
- 课程推荐使用Anaconda
<https://www.anaconda.com/>





高性能分发

- 轻松安装**1500+**多种数据科学软件包

包管理

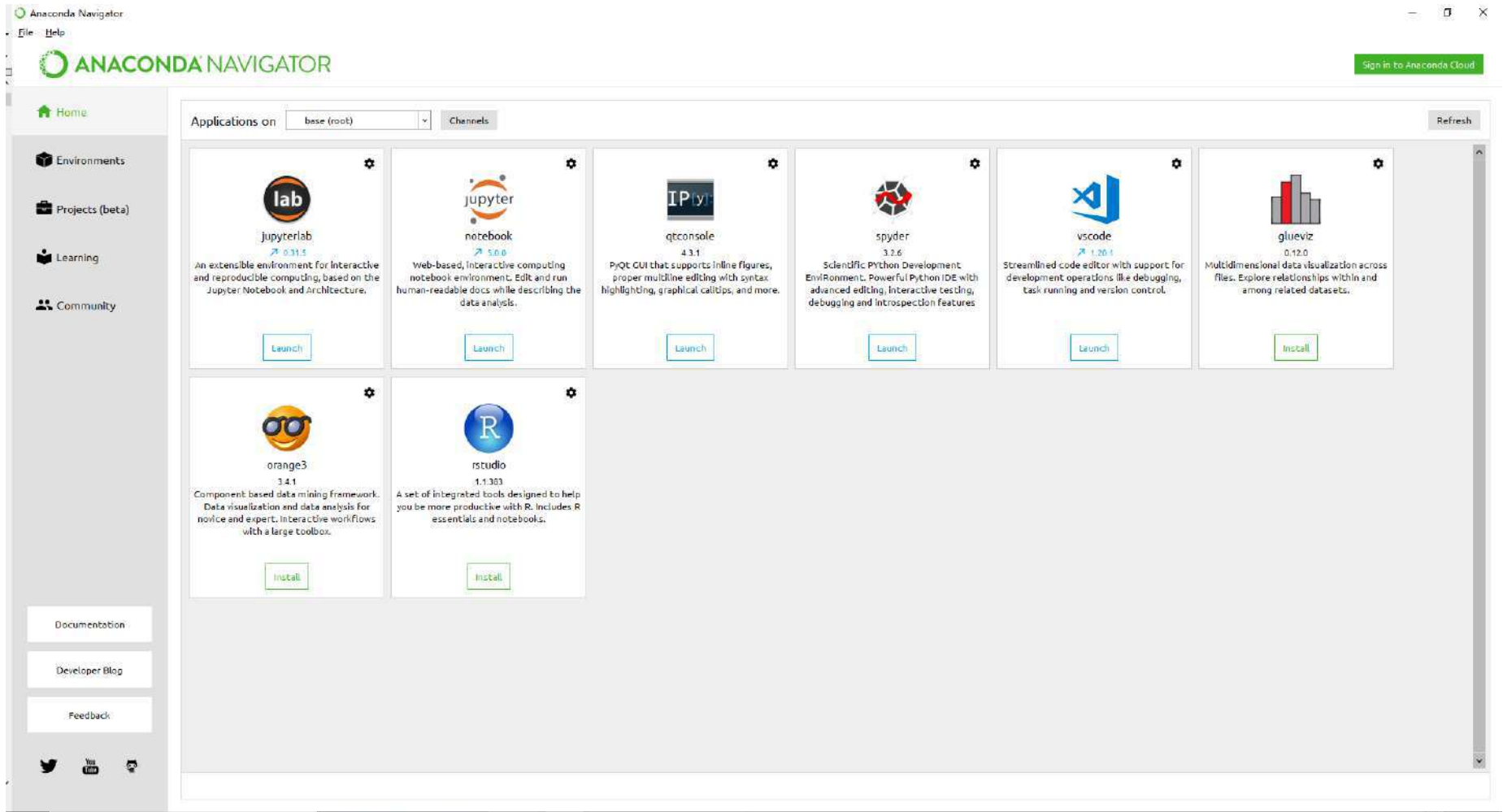
- 使用conda管理软件包，依赖项和环境，兼容pip

数据科学门户

- 发掘数据中的洞察力并创建交互式可视化



Anaconda界面





安装或者更新Python包

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Projects (beta)

Learning

Community

Documentation

Developer Blog

Feedback

Search Environments

base (root)

Installed Channels Update index... numpy X

Name	Description	Version
blaze	Numpy and pandas interface to big data	0.11.3
bottleneck	Fast numpy array functions written in cython.	1.2.1
numba	Numpy aware dynamic python compiler using llvm	0.35.0
numexpr	Fast numerical expression evaluator for numpy	2.6.2
numpy	Array processing for numbers, strings, records, and objects	1.13.3
numpydoc	Sphinx extension to support docstrings in numpy format	0.7.0

Create Clone Import Remove

6 packages available matching "numpy" 1 package selected

Apply Clear



Anaconda 软件源

- conda默认的软件源是官方的，访问不易，
- 更改为国内镜像源清华源(TUNA)中科大源(USTC)

添加USTC仓库镜像：

```
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkg/free/  
conda config --add channels https://mirrors.ustc.edu.cn/anaconda/pkg/main/  
conda config --set show_channel_urls yes
```

关于停止Anaconda镜像服务的通知

📅 2019-04-16 👤 TUNA Staff

根据 Anaconda [软件源](#)上的说明，Anaconda 和 Miniconda 是 Anaconda, Inc. 的商标，任何未经授权的公开镜像都是不允许的。去年我们曾尝试与公司有关人员联系，但未能取得授权。

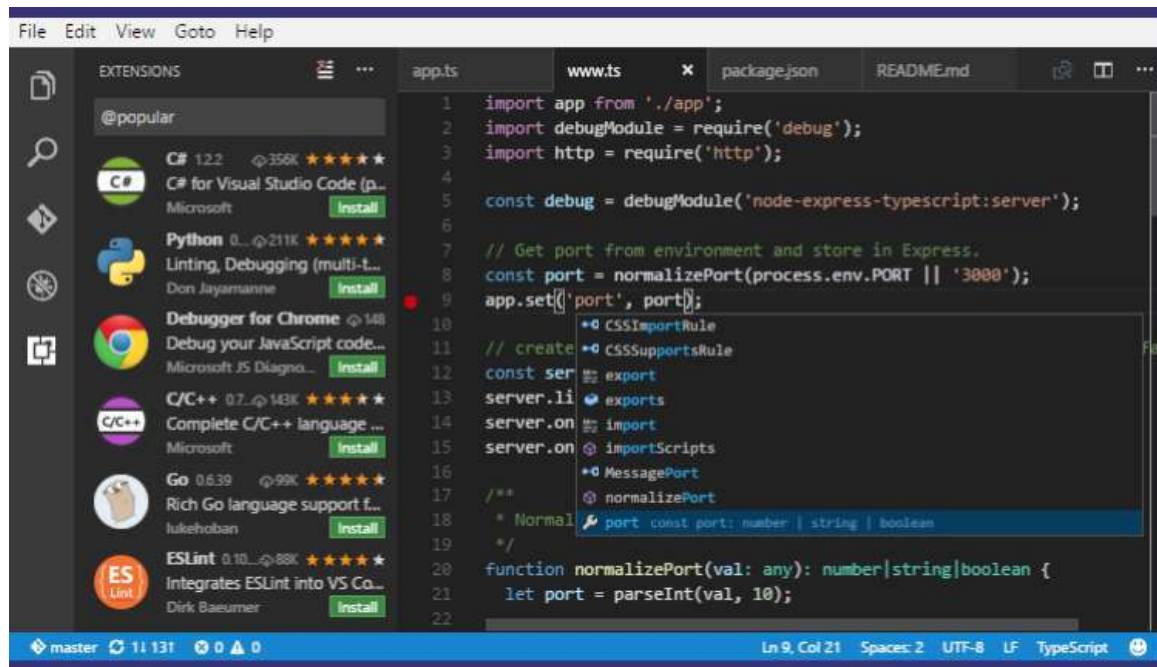
在没有上游授权的情况下，我们无法保证镜像的合法性与服务质量。因此我们决定，在取得授权之前无限期停止 Anaconda 镜像服务。即日起，我们将停止 Anaconda 的更新并隐藏镜像入口链接。一个月后，彻底关闭 Anaconda 镜像的文件下载。请现有用户尽快切换至官方下载地址，以免影响正常使用。

感谢您的理解与支持！



集成开发环境(IDEs) / 文本编辑器

- PyCharm JetBrains公司 (对于开源开发者免费)
- Python Tools for Visual Studio (VS插件)
- Spyder (免费), Anaconda自带
- Visual Studio Code





Python解释器

- Anaconda->Anaconda Prompt命令行输入python

```
Anaconda Prompt
(base) C:\Users\JUNY>python
Python 3.6.3 |Anaconda custom (64-bit)| (default, Oct 15 2017, 03:27:45) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> a = 5
>>> print(a)
5
>>> exit()

(base) C:\Users\JUNY>_
```

```
>>> ^Z
```

Ctrl-Z



运行py文件

- 在D盘目录下建立hello.py文件

A screenshot of a code editor window with a tab labeled 'hello.py'. The code inside the editor is a single line: `print('Hello world')`. The line number '1' is visible on the left margin.

```
1 print('Hello world')
```

A screenshot of an Anaconda Prompt terminal window. The title bar says 'Anaconda Prompt'. The prompt is `(base) C:\Users\JUNY>`. The command `python D:\hello.py` has been entered and executed, resulting in the output `Hello world`.

```
(base) C:\Users\JUNY>python D:\hello.py
Hello world
```



IPython

Anaconda Navigator

File Help



Home

Environ

Projects

Learning

Commu

Jupyter QtConsole

File Edit View Kernel Window Help

```
Jupyter QtConsole 4.3.1
Python 3.6.3 |Anaconda custom (64-bit)| (default, Oct 15 2017, 03:27:45) [MSC v.
1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.1.0 -- An enhanced Interactive Python. Type '?' for help.
```

```
In [1]: a = 5
```

```
In [2]: a
Out[2]: 5
```

```
In [3]: 2 ** 10
Out[3]: 1024
```

```
In [4]: print('Hello')
Hello
```

```
In [5]: print(
```

Tab键：代码补全、提示

多行输入：Ctrl+Enter

%run 运行文件

```
Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file:  a file-like object (stream); defaults to the current sys.stdout.
sep:   string inserted between values, default a space.
end:   string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
Type:      builtin_function_or_method
```



spyder

3.2.6

Python Development
Powerful Python IDE with
interactive testing
introspection feature

Launch

Documen



Jupyter Notebook

jupyter

Untitled1

Last Checkpoint: 2 minutes ago (autosaved)

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Code

In [1]:

```
print('Hello World!')
```

Hello World!

Shift-Enter

Links

Music

OneDrive

Pictures

PubMed

Saved Games

Searches

Videos

ch02.ipynb

Untitled.ipynb

Untitled1.ipynb

debug.log

25 days ago

4 months ago

25 days ago

2 minutes ago

8 days ago

24 days ago

25 days ago

5 months ago

9 days ago

12 days ago

15 hours ago

20 days ago

20 days ago

25 days ago

25 days ago

25 days ago

4 hours ago

2 months ago

Running seconds ago

2 months ago

Logout

Upload

New

Name

Last Modified

	25 days ago
	4 months ago
	25 days ago
	2 minutes ago
	8 days ago
	24 days ago
	25 days ago
	5 months ago
	9 days ago
	12 days ago
	15 hours ago
	20 days ago
	20 days ago
	25 days ago
	25 days ago
	25 days ago
	4 hours ago
	2 months ago
	Running seconds ago
	2 months ago