



南京信息工程大学  
Nanjing University of Information Science & Technology

大气科学学院  
SCHOOL OF ATMOSPHERIC SCIENCES

# Introduction to Python and Anaconda

NUIST

School of Atmospheric Science

崔叶孟 20201201010@nuist.edu.cn

2020-10-19

# Review

1. What is Python and Anaconda?
2. WHY?
3. Start Anaconda and Python
4. Package Install and Use
5. Choose IDE(Integrated Development Environment )
6. Best Practice

*Python is a programming language that lets you work quickly and integrate systems more effectively.*

--- <https://www.python.org/>

# 1 What is Python

- **High-level** easier than c, Fortran...
- **General-purpose** do more than grads, ncl, not only draw ...
- **Interpreted** no need compile like C, Fortran, Java...
- **Libraries** NumPy, Xarray, netCDF4, matplotlib...
- **Many users** find solution on Baidu, Bing or Stack Overflow...



# 1 What is Anaconda

- **Open source**
- **Conda packages**
- **Manage environment**

# 2 WHY

## The Zen of Python

```
Python 3.8.3 (default, Jul 2 2020, 11:26:31)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.16.1 -- An enhanced Interactive Python. Type '?' for help.
```

In [1]: `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!

In [2]:

# 2 WHY

From pip to conda

```
ModuleNotFoundError: No module named 'xxxxx'
```

# 3 Start Anaconda and Python

Hello world

1. anaconda官网, 下载安装



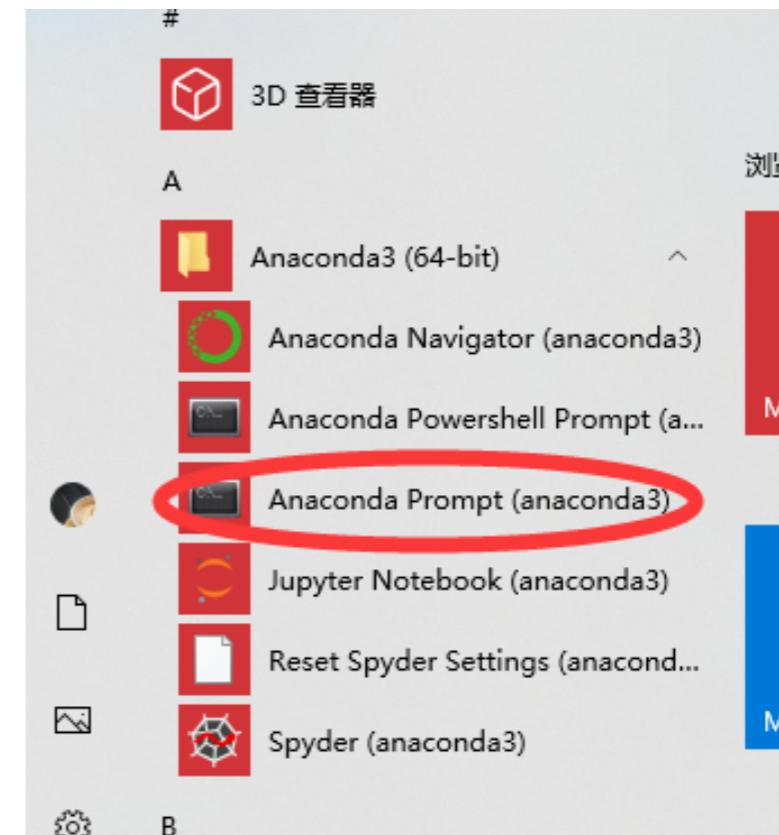
2. Anaconda Prompt(anaconda3)

3. python -V

4. python

5. >>> print("你好, 世界! ")

6. 你好, 世界!



```
cymeng — python — python — python — 80x24
[(base) promote:~:]% python
Python 3.8.3 (default, Jul 2 2020, 11:26:31)
[Clang 10.0.0 ] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("你好, 世界!")
你好, 世界!
>>>
```

安装指南: <https://docs.anaconda.com/anaconda/install/>

# 3 Start Anaconda and Python

修改 channel 加速下载

1. 清华大学镜像

2. conda config --set show\_channel\_urls yes

3. 修改用户目录下的配置文件 .condarc

4. 保存

5. conda clean -i



# 3 Start Anaconda and Python

## conda虚拟环境管理

- 创建名叫 myenv 环境， 版本3.7

```
conda create -n myenv python=3.7 可选：如果你的库只能在3.7版本下运行，甚至2.7...
```

- 查看所有环境

```
conda env list
```

- 激活 myenv 环境

```
conda activate myenv
```

- 查看当前python版本

```
python -V
```

- 退出环境

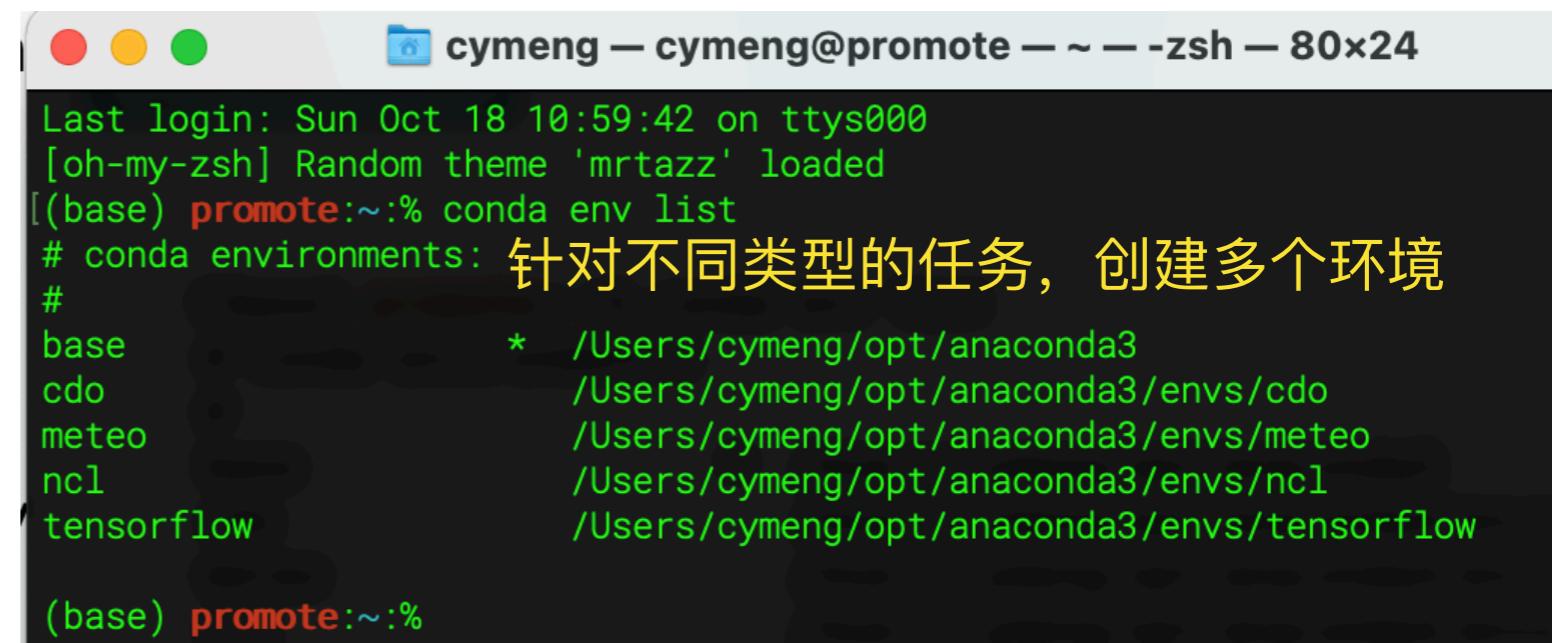
```
conda deactivate
```

- 再次查看版本

```
python -V
```

- 删除 myenv 环境

```
conda remove -n myenv --all
```



A terminal window titled "cymeng" showing the output of the command "conda env list". The output lists several environments: base, cdo, meteo, ncl, and tensorflow. The tensorflow environment is marked with an asterisk (\*) and its path is shown as "/Users/cymeng/opt/anaconda3/envs/tensorflow". The terminal prompt "(base) promote:~:%" is visible at the bottom.

```
Last login: Sun Oct 18 10:59:42 on ttys000
[oh-my-zsh] Random theme 'mrtazz' loaded
(base) promote:~:%
# conda environments: 针对不同类型的任务，创建多个环境
#
base
cdo
meteo
ncl
tensorflow
*
  /Users/cymeng/opt/anaconda3
  /Users/cymeng/opt/anaconda3/envs/cdo
  /Users/cymeng/opt/anaconda3/envs/meteo
  /Users/cymeng/opt/anaconda3/envs/ncl
  /Users/cymeng/opt/anaconda3/envs/tensorflow
(base) promote:~:%
```

如果你换了电脑，或者要重装系统

或者要在服务器上运行

- 导出环境

```
conda env export > environment.yml
```

- 恢复环境

```
conda env create -f environment.yml
```

# 3 Start Anaconda and Python

## 4. 一行代码安装 ncl, cdo

- CDO

```
$ conda create -n cdo -c  
conda-forge cdo  
$ conda activate cdo
```

```
(base) cyemeng@DESKTOP-680SRV5:~$ conda activate cdo  
(cdo) cyemeng@DESKTOP-680SRV5:~$ cdo  
No operator given!  
  
usage : cdo [Options] Operator1 [-Operator2 [-OperatorN]]  
  
Options:  
  -a           Generate an absolute time axis  
  -b <nbits>   Set the number of bits for the output preci  
                (I8/I16/I32/F32/F64 for nc1/nc2/nc4/nc4c/nc  
                b2/srv/ext/ieg; P1 - P24 for grb1/grb2)  
  -L           Add L or B to set the byteorder to Little o  
  -C           CHOP conform NetCDF output
```

- NCL

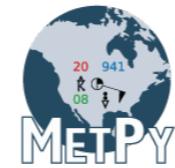
```
$ conda create -n ncl -c  
conda-forge ncl  
$ conda activate ncl
```

```
(123) cyemeng@DESKTOP-680SRV5:~$ conda activate ncl  
(ncl) cyemeng@DESKTOP-680SRV5:~$ ncl  
Copyright (C) 1995-2019 - All Rights Reserved  
University Corporation for Atmospheric Research  
NCAR Command Language Version 6.6.2  
The use of this software is governed by a License Agreement.  
See http://www.ncl.ucar.edu/ for more details.  
ncl 0>
```

! 仅限UNIX (Windows 推荐 WSL)

# 4 Package Install and Use

conda install <name>



# 4 Package Install and Use

## Numpy introduction

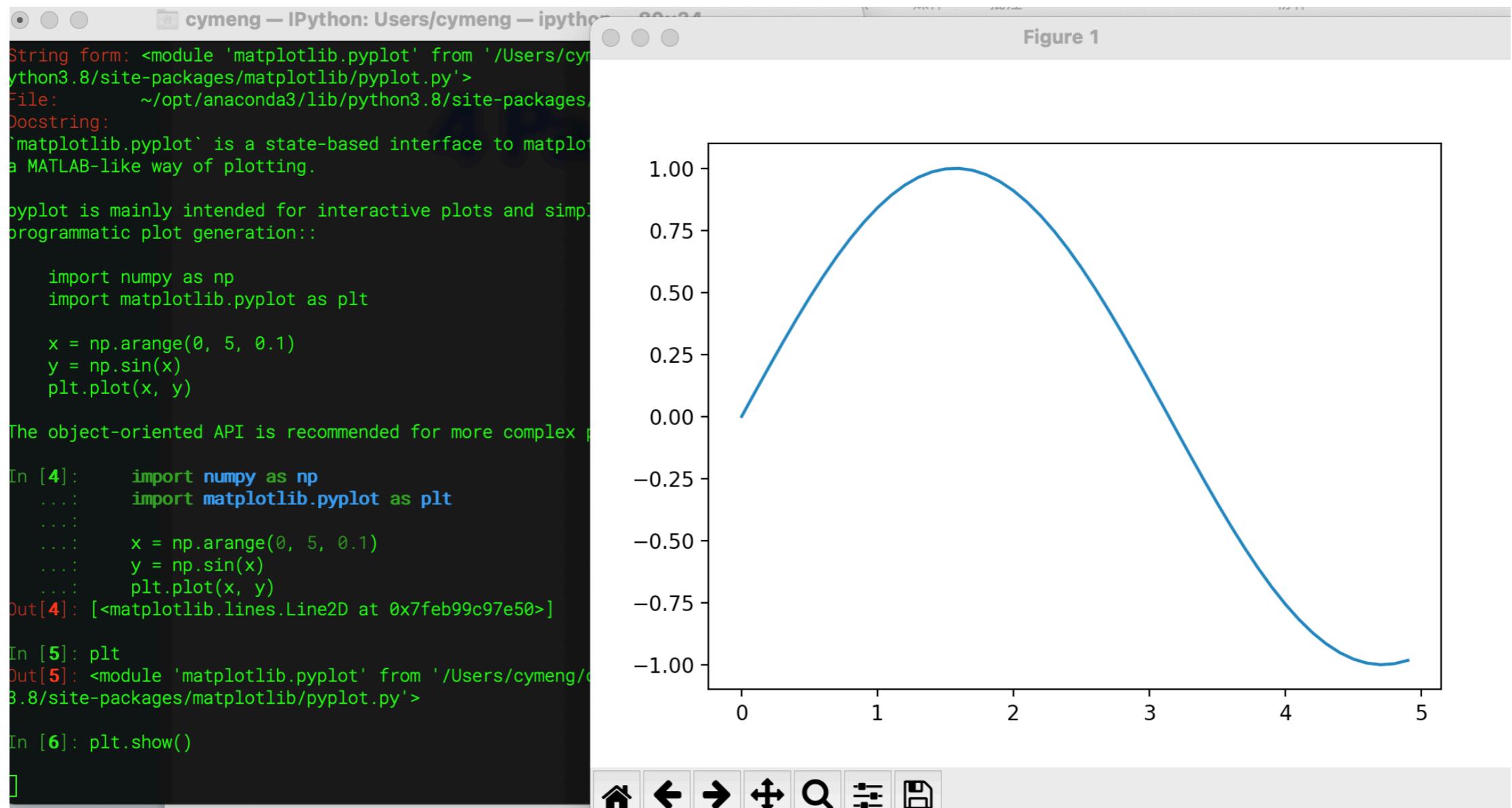
- 多维数组
- 对数组进行快速数学处理
- 线性代数，傅里叶变换，随机数生成器



Quantum Computing	Statistical Computing	Signal Processing	Image Processing	3-D Visualization	Symbolic Computing	Astronomy Processes	Cognitive Psychology
QuTiP PyQuil Qiskit	Pandas statsmodels Seaborn	SciPy PyWavelets	Scikit-image OpenCV	Mayavi Napari	SymPy	AstroPy SunPy SpacePy	PsychoPy
Bioinformatics	Bayesian Inference	Mathematical Analysis	Simulation Modeling	Multi-variate Analysis	Geographic Processing	Interactive Computing	
BioPython Scikit-Bio PyEnsembl	PyStan PyMC3	SciPy SymPy cvxpy	PyDSTool	PyChem	Shapely GeoPandas Folium	Jupyter IPython Binder	

# 4 Package Install and Use

## Matplotlib introduction



# 4 Package Install and Use

# Pandas introduction

- 结构化数据的处理，电子表格，例如站点观测资料

- 多种格式，比如 csv, xlsx, xml, html

- 数据整理, 清洗, 归纳, 统计

- [https://pandas.pydata.org/intro/tutorials/01\\_table\\_c](https://pandas.pydata.org/intro/tutorials/01_table_c)



## DataFrame

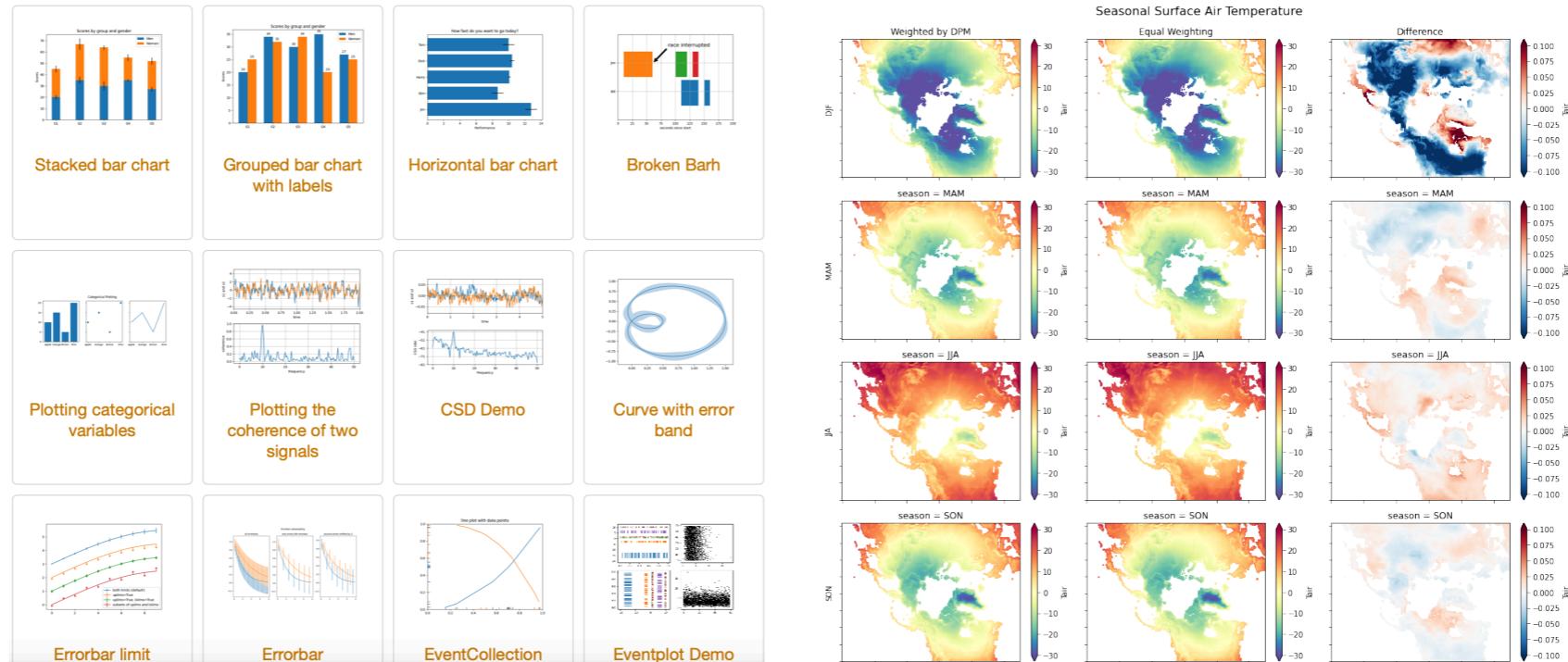
	温度	湿度	气压	...
50136				
50137				
50420				row

演示

# 4 Package Install and Use

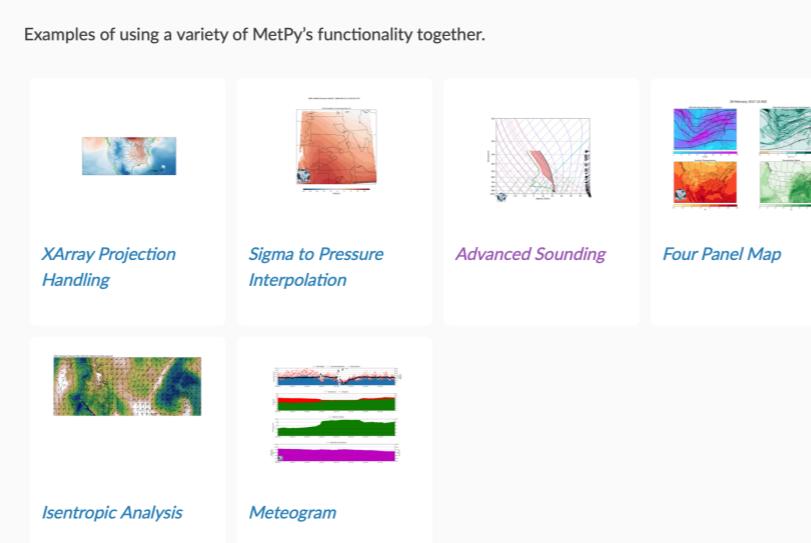
## Find Example

- User Guide
- Example
- Gallery
- Or Baidu



### General Examples [»](#)

Examples of using a variety of MetPy's functionality together.



# 5 Choose IDE

代码提示，语法高亮，检测错误

- Visual Studio Code



- Spyder 自带



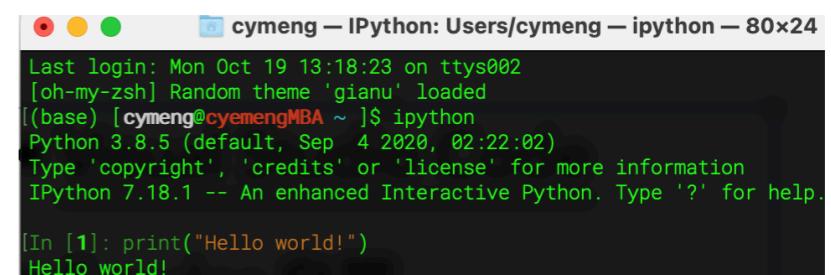
- PyCharm ✓



- Jupyter Notebook ✓



- IPython✓



A screenshot of a terminal window titled "cymeng — IPython: Users/cymeng — ipython — 80x24". The window shows the following text:

```
Last login: Mon Oct 19 13:18:23 on ttys002
[oh-my-zsh] Random theme 'gianu' loaded
[(base) [cymeng@cymengMBA ~ ]$ ipython
Python 3.8.5 (default, Sep  4 2020, 02:22:02)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.18.1 -- An enhanced Interactive Python. Type '?' for help.

[In [1]: print("Hello world!")
Hello world!
```

# 5 Choose IDE



# SPYDER

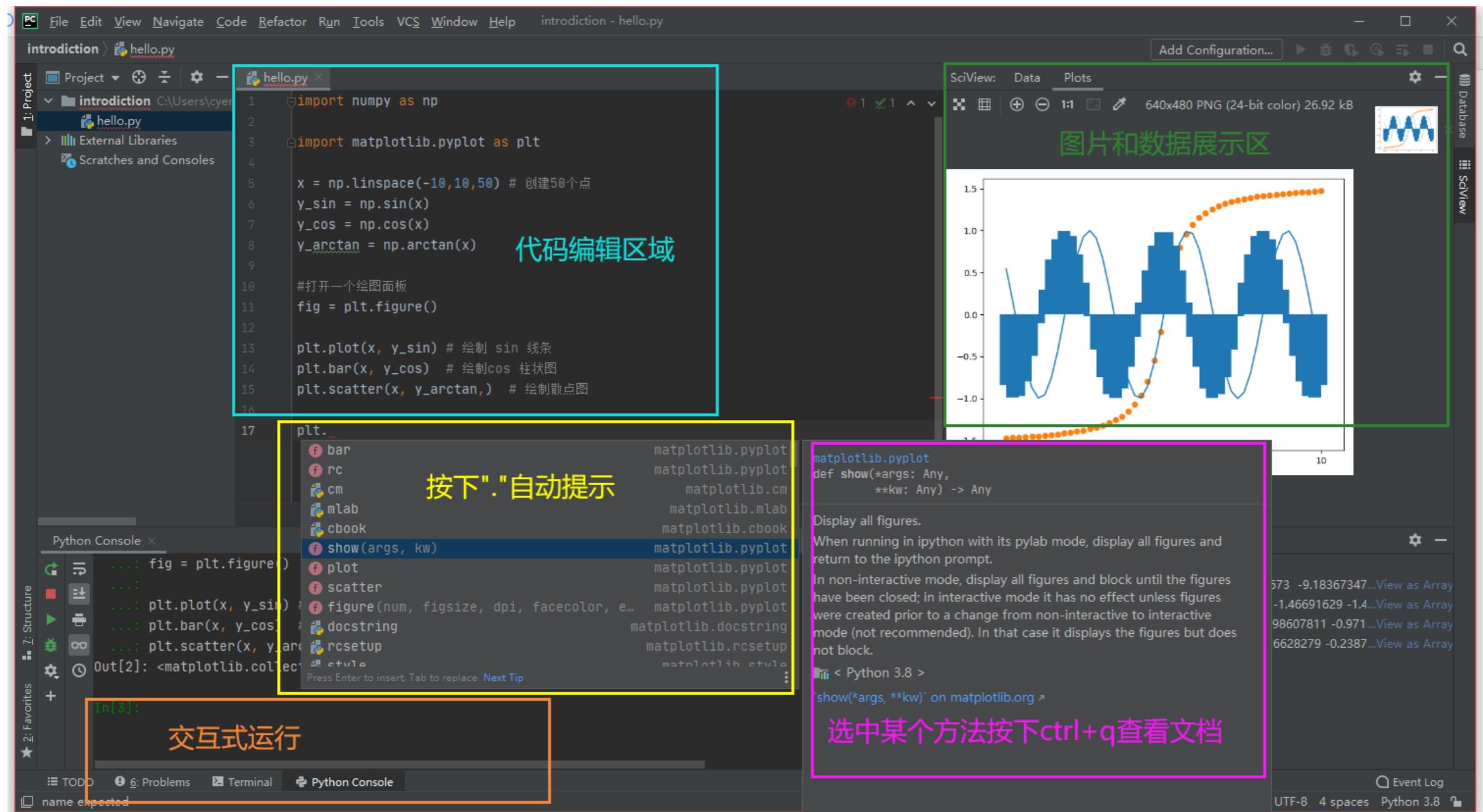
The Scientific Python Development Environment

The screenshot displays the Spyder IDE interface with the following components:

- Left Panel:** Shows a file tree with files like App.py, template.py, plot\_example.py (selected), plugin.py, and IPythonConsole.
- Code Editor:** Displays the content of plot\_example.py, which includes imports for numpy, matplotlib, and mpl\_toolkits, along with functions for generating polar and DEM plots.
- Variable Explorer:** A table showing variables: data (ndarray), df (DataFrame), filename (str), i (array), li (list), r (float), t (tuple), and tinylist (list).
- Plots:** Two plots are shown: a 3D surface plot of a terrain model and a polar plot.
- Bottom Status Bar:** Shows "LSP Python: restarting...", "conda: base (Python 3.7.4)", "Line 1, Col 1", "ASCII", "LF", "RW", and "Mem 50%".

# 5 Choose IDE

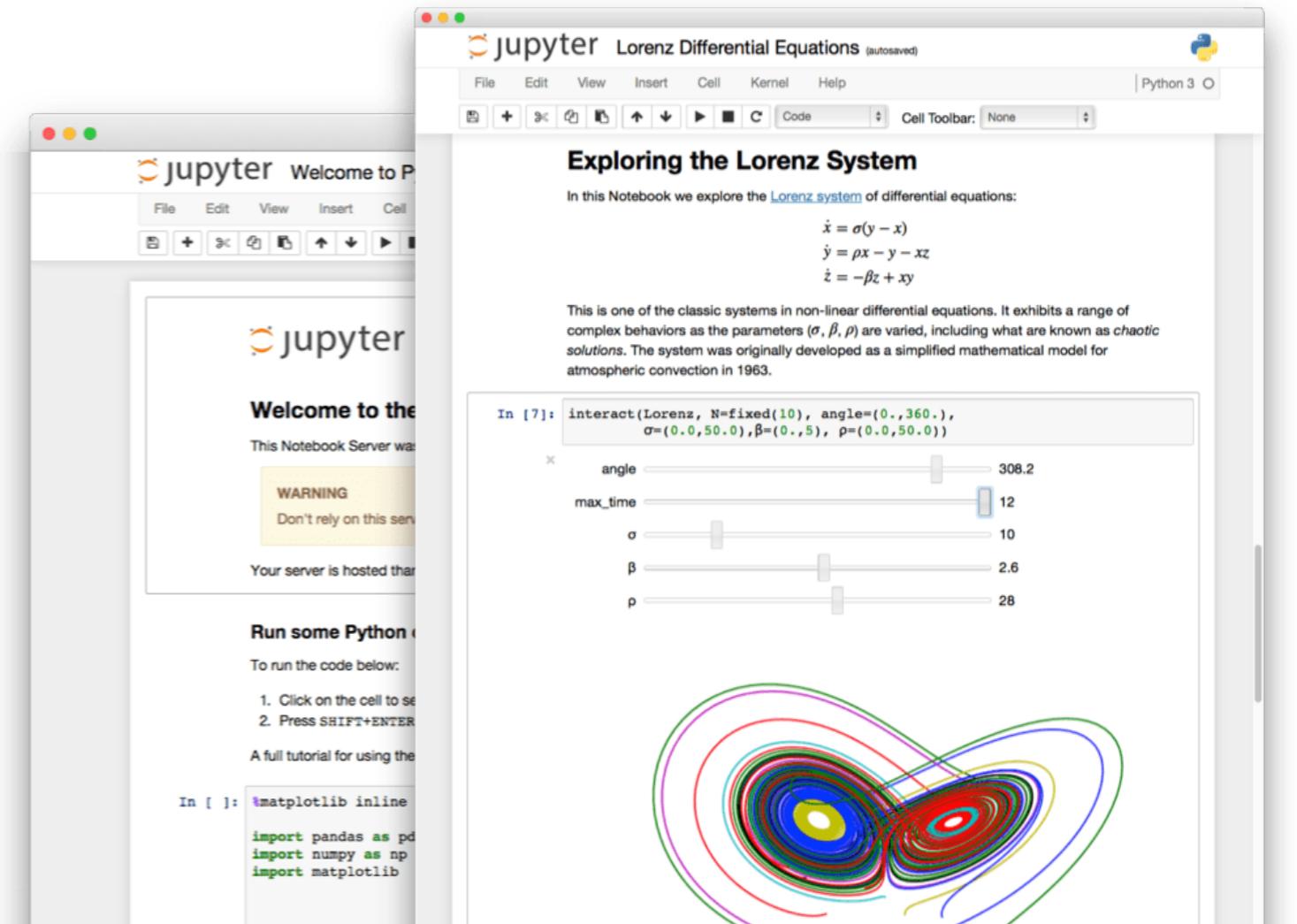
## PyCharm



# 5 Choose IDE

jupyter notebook--literate programming

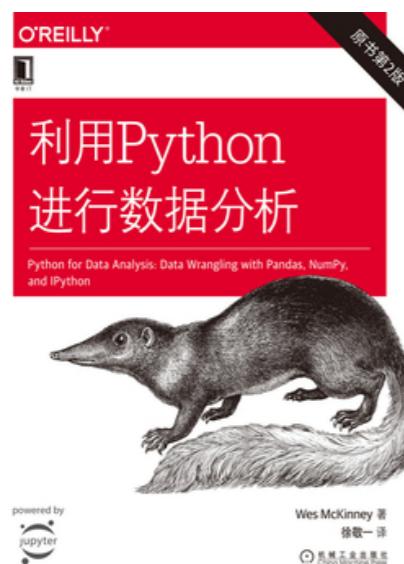
- \$ jupyter notebook
- http://localhost:8888/
- 新建>Python3



# 6 Best Practice

## Online Resource

- [Python 3 教程|菜鸟教程](#)
- [《利用Python进行数据分析》](#)
- [《Python数据可视化之美》](#)
- . . .



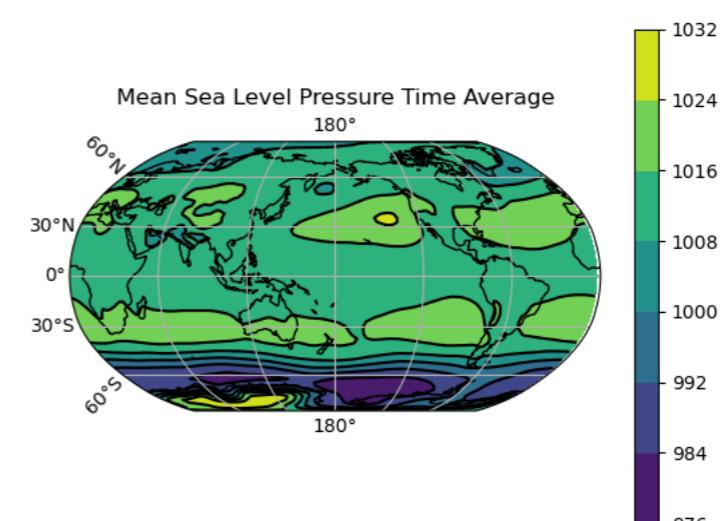
The screenshot shows a web browser displaying the RUNOOB.COM website. The URL bar shows 'runoob.com'. The main content area is titled 'Python 3 教程' (Python 3 Tutorial). On the left is a sidebar with a navigation menu for Python 3, including 'Python3 教程' (selected), 'Python3 简介', 'Python3 环境搭建', 'Python3 基础语法', 'Python3 基本数据类型', 'Python3 解释器', 'Python3 注释', 'Python3 运算符', 'Python3 数字 (Number)', 'Python3 字符串', 'Python3 列表', 'Python3 元组', 'Python3 字典', 'Python3 集合', 'Python3 编程第一步', 'Python3 条件控制', and 'Python3 循环语句'. The main content area contains text about Python 3.0, a Python logo, code examples like 'print("Hello, world!")', and a note about Python 2's end-of-life. A search bar at the top right says '搜索.....'.

# 6 Best Practice

## Batch Process

The screenshot illustrates the use of command-line arguments in a Python script. On the left, a code editor shows the script `Mean_Sea_Level_Pressure.py`. The line `filename = sys.argv[1]` is highlighted in red, indicating it is the first argument passed to the script. In the center, a file explorer window shows the directory `C:\Users\cyemeng\PycharmProjects\introduction` containing files like `Mean_Sea_Level_Pressure.py` and `mslp.2020.nc`. On the right, a terminal window titled "Anaconda Powershell Prompt (anaconda3)" shows the command `python Mean_Sea_Level_Pressure.py mslp.2020.nc` being run. Three arrows point from the text "python命令" (Python command), "要执行的脚本名称" (script name to be executed), and "第一个参数 sys.argv[1]" (first parameter `sys.argv[1]`) to the corresponding parts of the terminal command.

```
Mean_Sea_Level_Pressure.py
1 import cartopy.crs as ccrs
2 import xarray as xr
3 import matplotlib.pyplot as plt
4 import sys
5
6 # nc文件处理
7 filename = sys.argv[1] # 第一个参数为文件名称
8 ds = xr.open_dataset(filename) # 打开 nc 文件
9 mslp = ds.mslp # 提取气压场变量
10 mslp_avg = mslp.mean("time")/100 # 先做时间平均, 再除以100
11
12 # 打开画板
13 fig = plt.figure()
14 # 作为单个整数编码的子绘图网格参数
15 # "111"表示"1×1网格, 第一子图", "234"表示"2×3网格, 第三子图"
16 ax1 = fig.add_subplot(111, projection=ccrs.PlateCarree())
17 ax1.set_global()
18 ax1.coastlines() # 设置海岸线
19 ax1.gridlines(draw_labels=True) # 绘制网格
20
21 contourf = ax1.contourf(ds.lon, ds.lat,
22 filled = ax1.contour(ds.lon, ds.lat, ms
23 colors=['black'],
24 transform=ccrs.Plat
25
26 plt.colorbar(contourf, ax=ax1) # 设置色
27 plt.title("%s Time Average" % mslp.var_
28 plt.savefig("%s Time Average" % mslp.var_
```



# 6 Best Practice

4 P

- Practice
- Passion
- Patience
- Persistance

**Good Luck!**