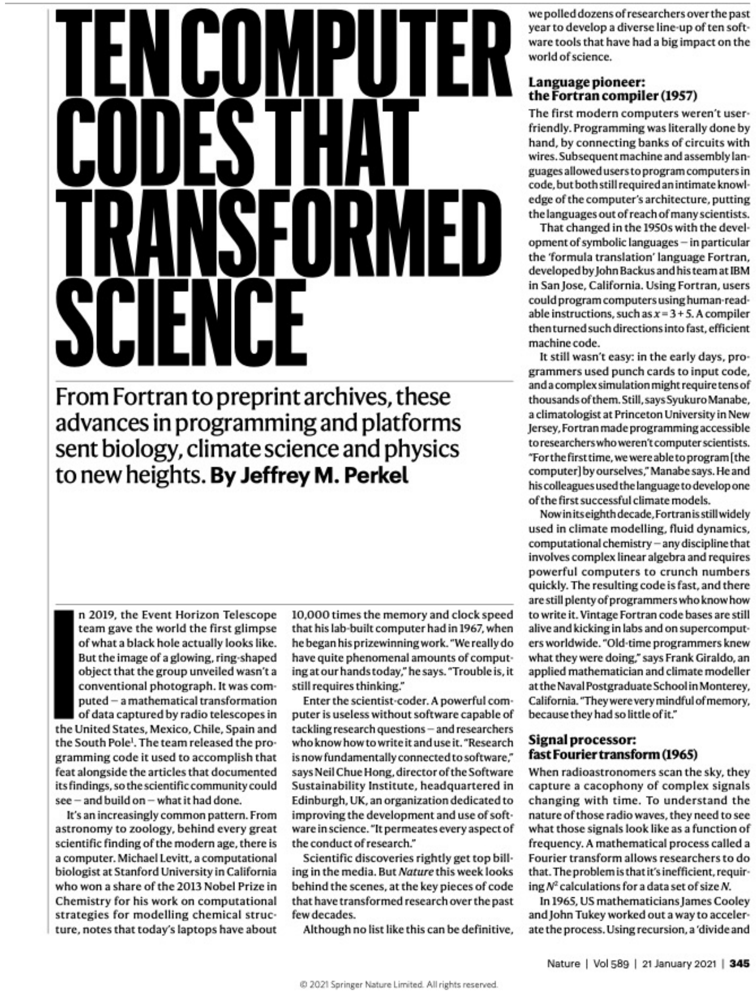
科研经验分享

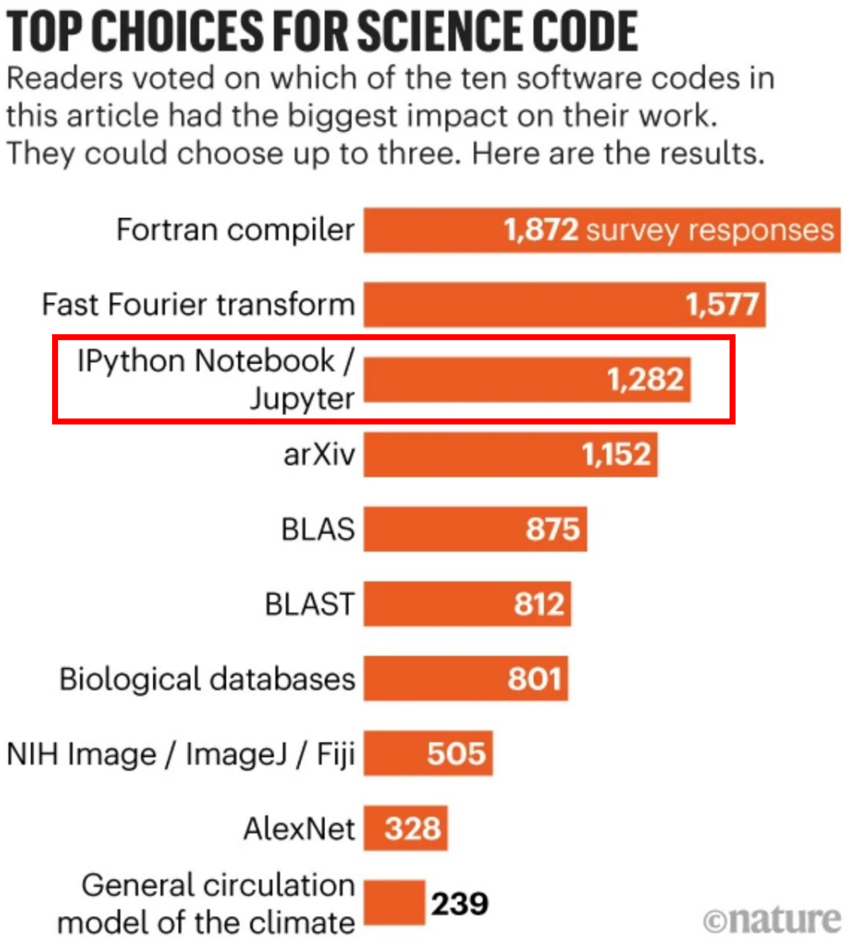
为交互式计算而生——Jupyter项目

赵相杰

2023年2月27日

In 2021, Nature named Jupyter as one of ten computing projects that transformed science





数据科学岗位任职要求

高级机器视觉工程师

任职要求：

1、硕士及以上学历，具有处理深度学习或工业项目的经验者优先;

2、掌握以下其中一种编程能力：

1）掌握C++或C#，有实际编码经验，熟练使用VS或QT等编译器；

2）掌握python，有实际编码经验，熟练使用jupyter notebook，PyCharm等编译器；

3、具备数字图像处理及模式识别相关的基础理论和算法知识优先；

4、使用任一种视觉软件库(Halcon/Opencv)，深度框架（如PyTorch）进行视觉系统开发，；

5、具备较强的自学能力和独立解决问题的能力，善于团队协作与沟通。

6、能接受出差工作安排。

数据分析师

职位描述

hive python 数据分析 运营

任职要求：

1、统招本科及以上学信网可查学历；

2、3年及以上数据分析相关经验，有运营经验优先；

3、熟练使用Python及jupyter、sklearn等工具及包，根据需求进行数据分析。

4、熟练编写HSQL，熟悉hive、hadoop等基本操作。

5、熟练编写shell和Python脚本。

6、会写hive+Python即可，会写map reduce任务加分项

数据分析师

任职要求：

学历及专业：

学历：本科及以上。

专业：计算机、软件工程、通信、数学等相关专业。

专业技能：

1. 熟练掌握Linux下的编程，掌握Python语言，熟练使用Jupyter等工具；

2. 熟练掌握数据分析、探索、BI等技术；

3. 熟悉数据库操作，SQL语言，熟悉大数据技术，包括Hive等；

4. 深入了解机器学习、人工智能相关算法，且有一定广度者优先；

5. 熟悉人工智能领域当前热点和前沿技术，对至少一个应用领域有系统的认识；

6. 熟悉业界通用的人工智能和机器学习方案和应用流程，具有根据需求制定算法方案并完成工程开发的能力；

7. 有系统学习和应用机器学习/人工智能技术经验者优先。

Free software, open standards, and web services for **interactive computing** across all programming languages

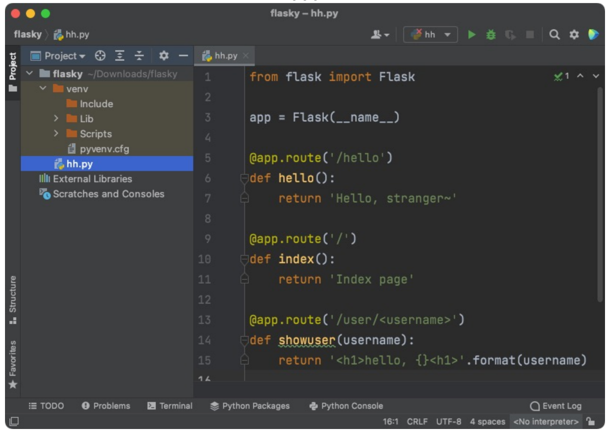
Python

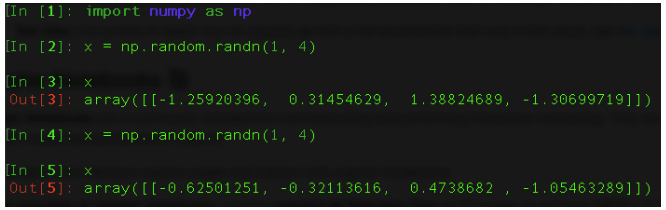
软件工程

数据科学

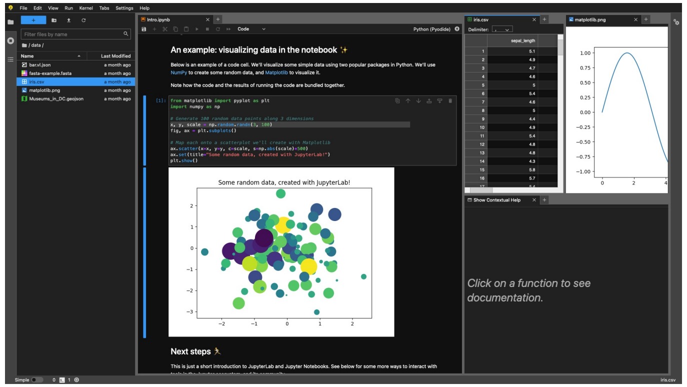
scripts (.py files)

Pycharm
Visual Studio Code
...





IPython



Jupyter

Python

软件工程

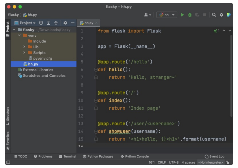
数据科学

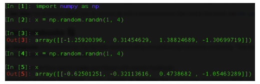
scripts (.py files)

Users > xiangjie > Documents > 备忘 > omics-pipeline-master > omxpipeline > bulkRNA-pipeline.py

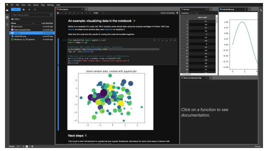
```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Created on Sun Jul 21 2019

@author: Xiangjie Zhao
@email: xjzhao@genetics.ac.cn

A pipeline for bulk RNA-seq analysis
"""

# --------------------------------------------------
# module, compatibility

from __future__ import (division, print_function, absolute_import, unicode_literals)
from os.path import join as join_path
from datetime import datetime

import os, subprocess
import argparse

try:
    from future_builtins import ascii, filter, hex, map, oct, zip
except:
    pass

if os.sys.version_info.major > 2:
    xrange = range

# --------------------------------------------------
# utils

# --------------------
# general

class OmicsException(Exception):

    """custom exception
    """
    pass
```

Pycharm
Visual Studio Code
...





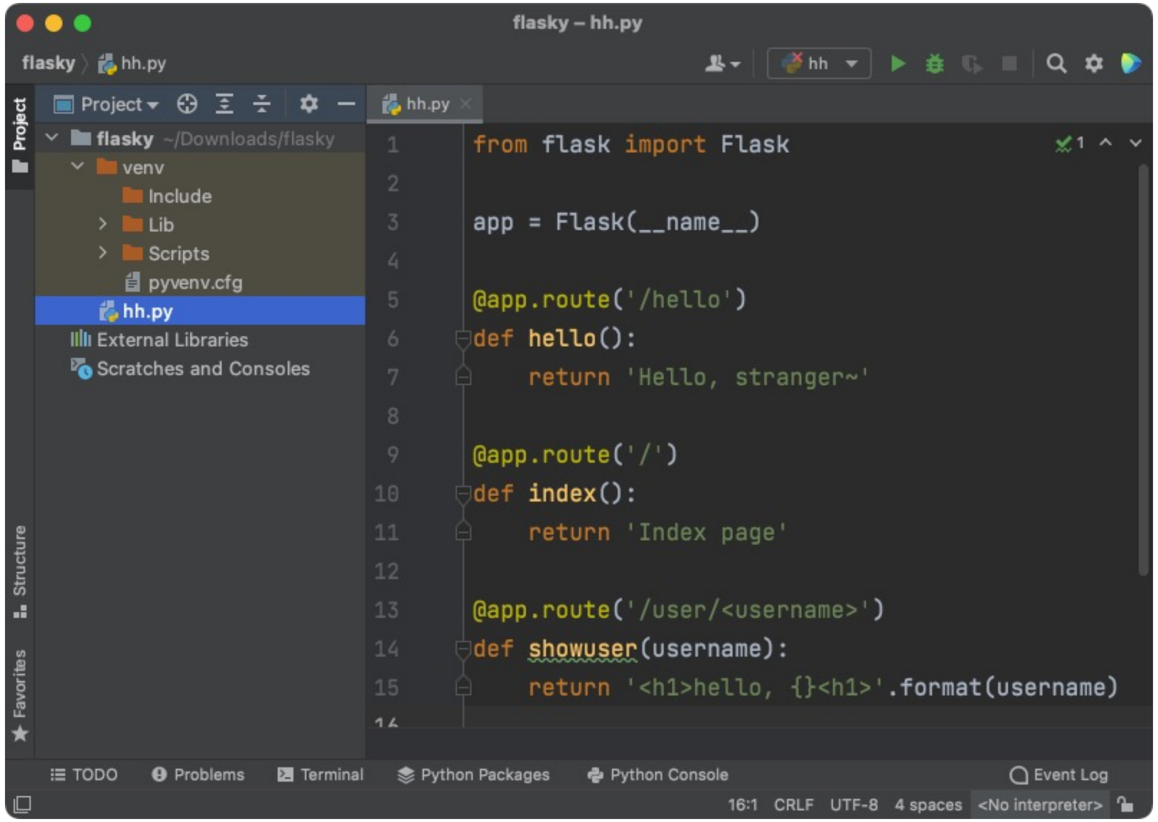
IPython



Jupyter

scripts (.py files)

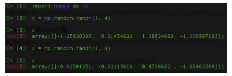
Pycharm
Visual Studio Code
...



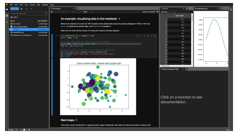
软件工程

Python

数据科学



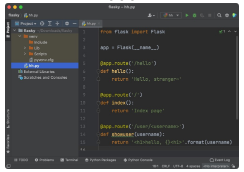
IPython



Jupyter

scripts (.py files)

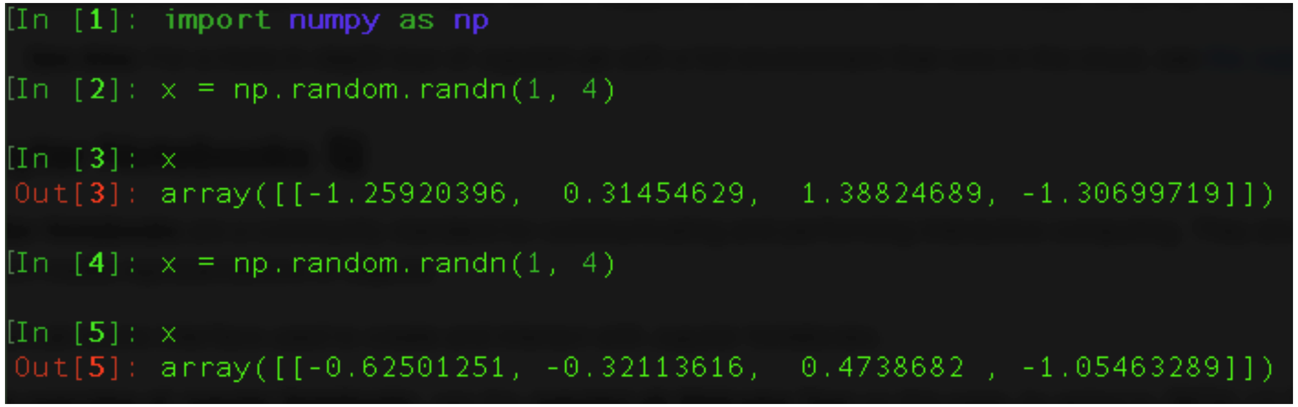
Pycharm
Visual Studio Code
...



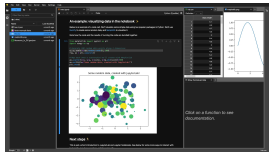
软件工程

Python

数据科学



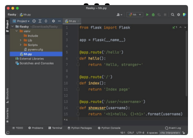
Ipython（2001年）



Jupyter

scripts (.py files)

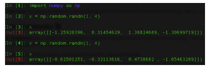
Pycharm
Visual Studio Code

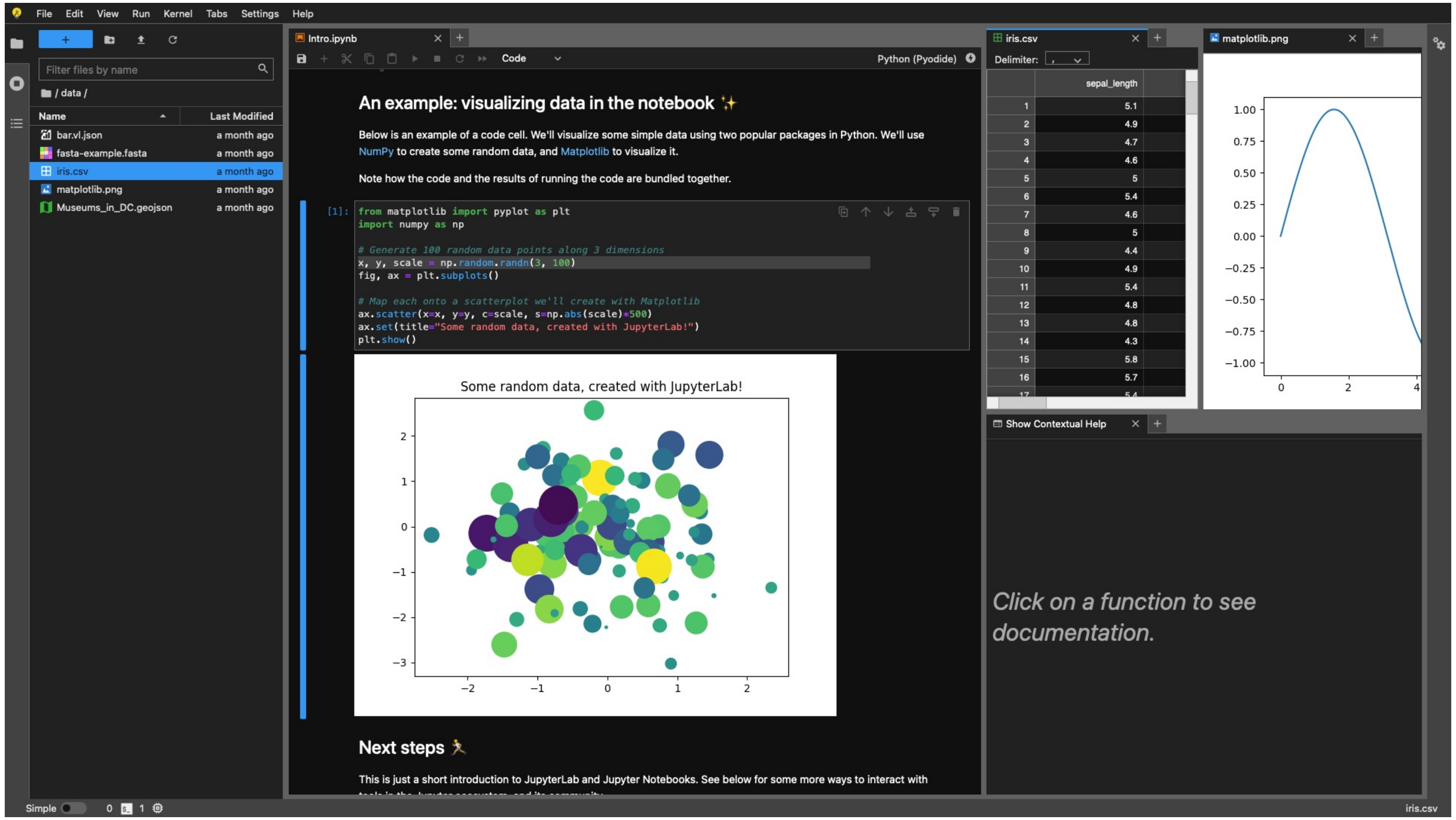


软件工程

Python

数据科学





IPython

Jupyter（2014年）

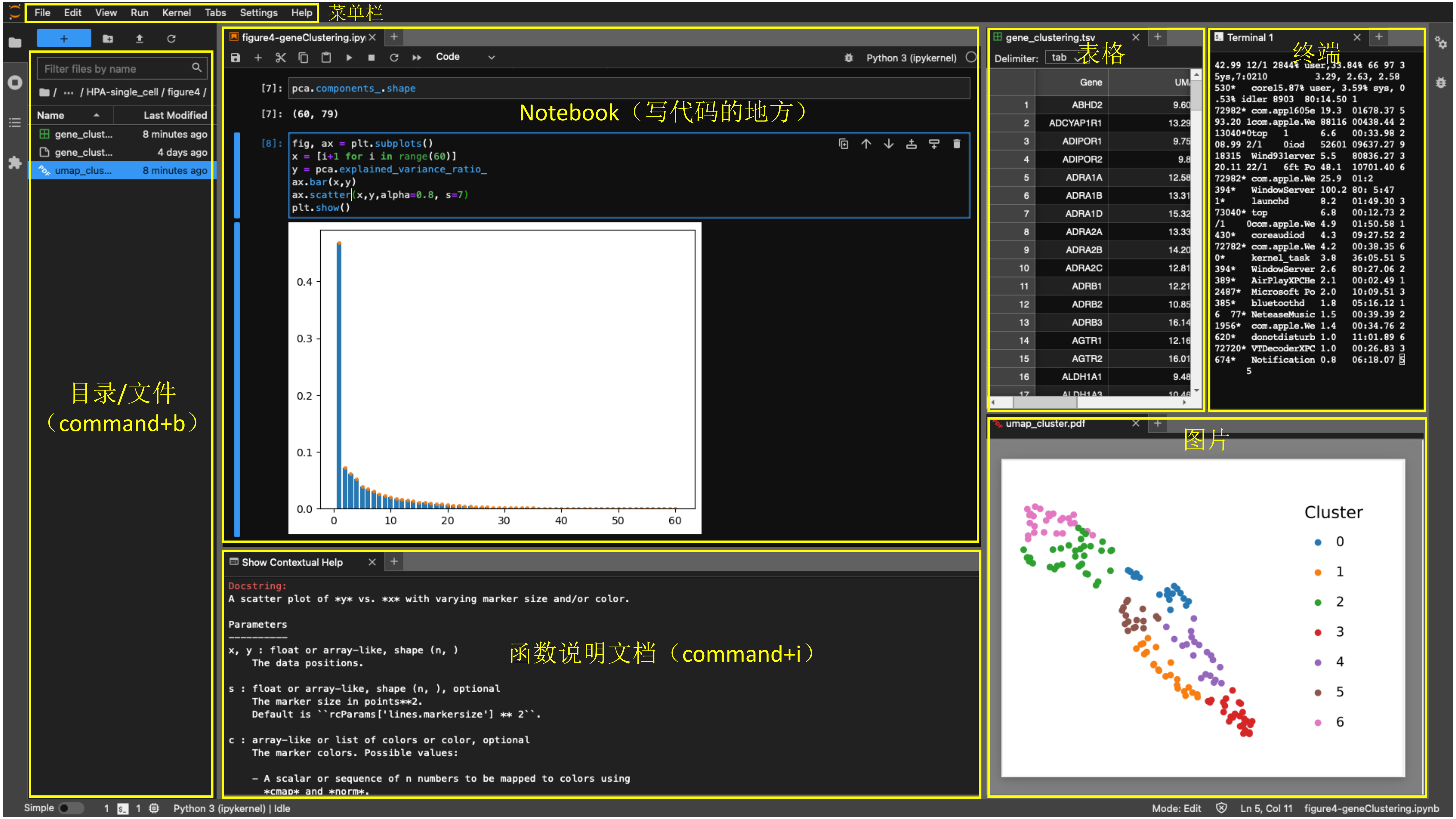
Jupyter名字的由来

Jupyter

Julia, Python and R

Jupiter（木星）

JupyterLab界面介绍

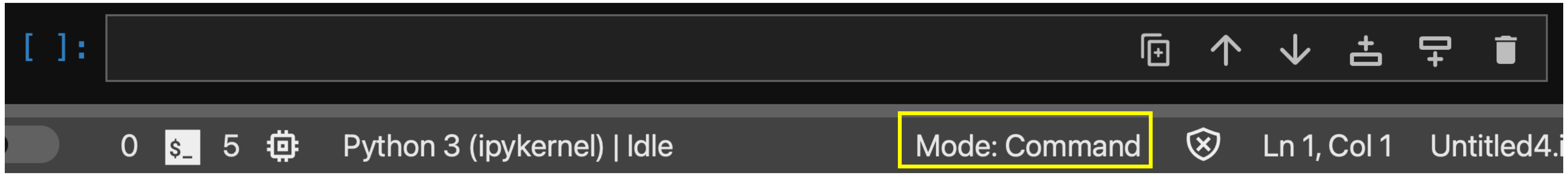


菜单栏
File Edit View Run Kernel Tabs Settings Help
目录/文件
（command+b）
Notebook（写代码的地方）
表格
终端
图片
函数说明文档（command+i）
figure4-geneClustering.ipynb
[7]: pca.components_.shape
[7]: (60, 79)
[8]: fig, ax = plt.subplots()
x = [i+1 for i in range(60)]
y = pca.explained_variance_ratio_
ax.bar(x,y)
ax.scatter(x,y,alpha=0.8, s=7)
plt.show()
gene_clustering.tsv
Terminal 1
umap_cluster.pdf
Cluster
Show Contextual Help
Docstring:
A scatter plot of *y* vs. *x* with varying marker size and/or color.
Parameters
x, y : float or array-like, shape (n,)
The data positions.
s : float or array-like, shape (n,), optional
The marker size in points**2.
Default is ``rcParams['lines.markersize'] ** 2``.
c : array-like or list of colors or color, optional
The marker colors. Possible values:
- A scalar or sequence of n numbers to be mapped to colors using
cmap and *norm*.

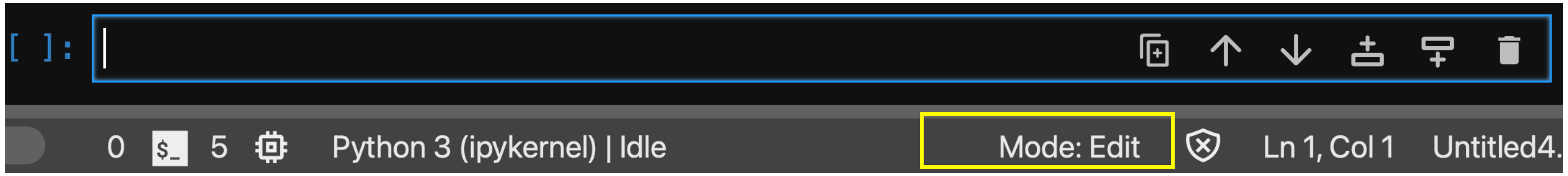

File Edit View Run Kernel Tabs Settings Help
显示和选择当前Notebook
的编程语言环境
Untitled1.ipynb
Code
Python 3 (ipykernel)
对cell进行添加、复制、粘贴、运行等操作（左），中断
和重启整个Notebook（中），显示当前cell的类型（右）
运行状态
[ ]:
[ ]:
[ ]:
这叫一个cell，或者说代码块，运行的结果会直接展示在下方
Simple
0
2
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
Untitled1.ipynb

用Jupyter的*Hello World*介绍Notebook的两种操作模式

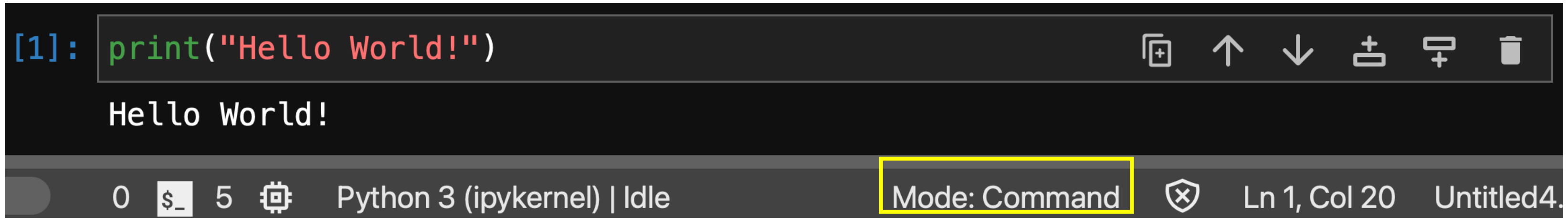
新建一个Notebook



默认是*命令模式*



按下ENTER进入*编辑模式*



*编辑模式*下输入代码并运行

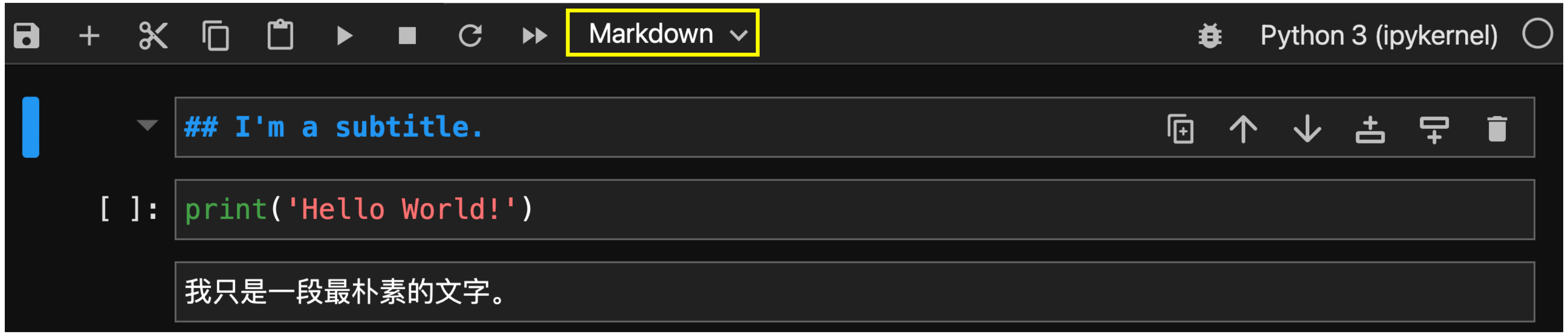
*命令模式*下（按下ESC进入）的输入视为各种命令，比如这里依次键入c和v的效果是复制并粘贴了一个cell

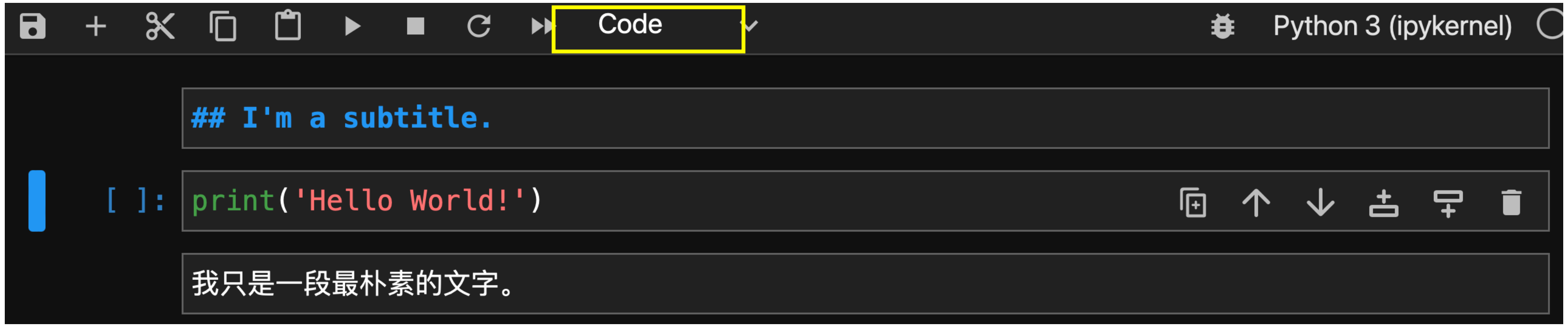
Notebook的两种操作模式

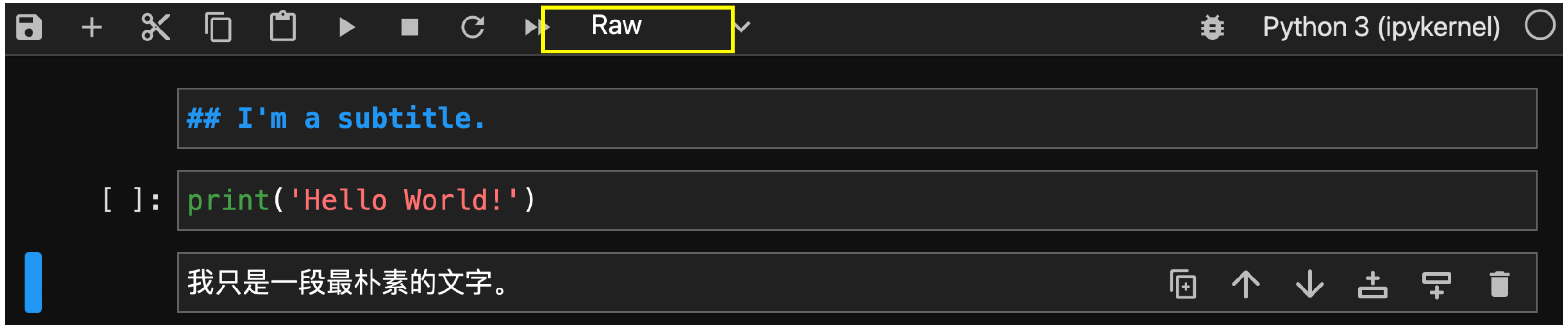
编辑模式（按下ENTER进入）：写代码、在一个cell内部进行编辑。

命令模式（按下ESC进入）：进入后无法写代码，而是对整个cell进行操作，比如复制、删除、改变cell类型等。

三种cell类型


Markdown
Python 3 (ipykernel)
I'm a subtitle.
[ ]: print('Hello World!')
我只是一段最朴素的文字。


Code
Python 3 (ipykernel)
I'm a subtitle.
[ ]: print('Hello World!')
我只是一段最朴素的文字。


Raw
Python 3 (ipykernel)
I'm a subtitle.
[ ]: print('Hello World!')
我只是一段最朴素的文字。

三种cell类型

```
## I'm a subtitle.
```
Markdown（命令模式下输入m）

```
[ ]: print('Hello World!')
```
Code（命令模式下输入y）

```
我只是一段最朴素的文字。
```
Raw（命令模式下输入r）

Run

I'm a subtitle.

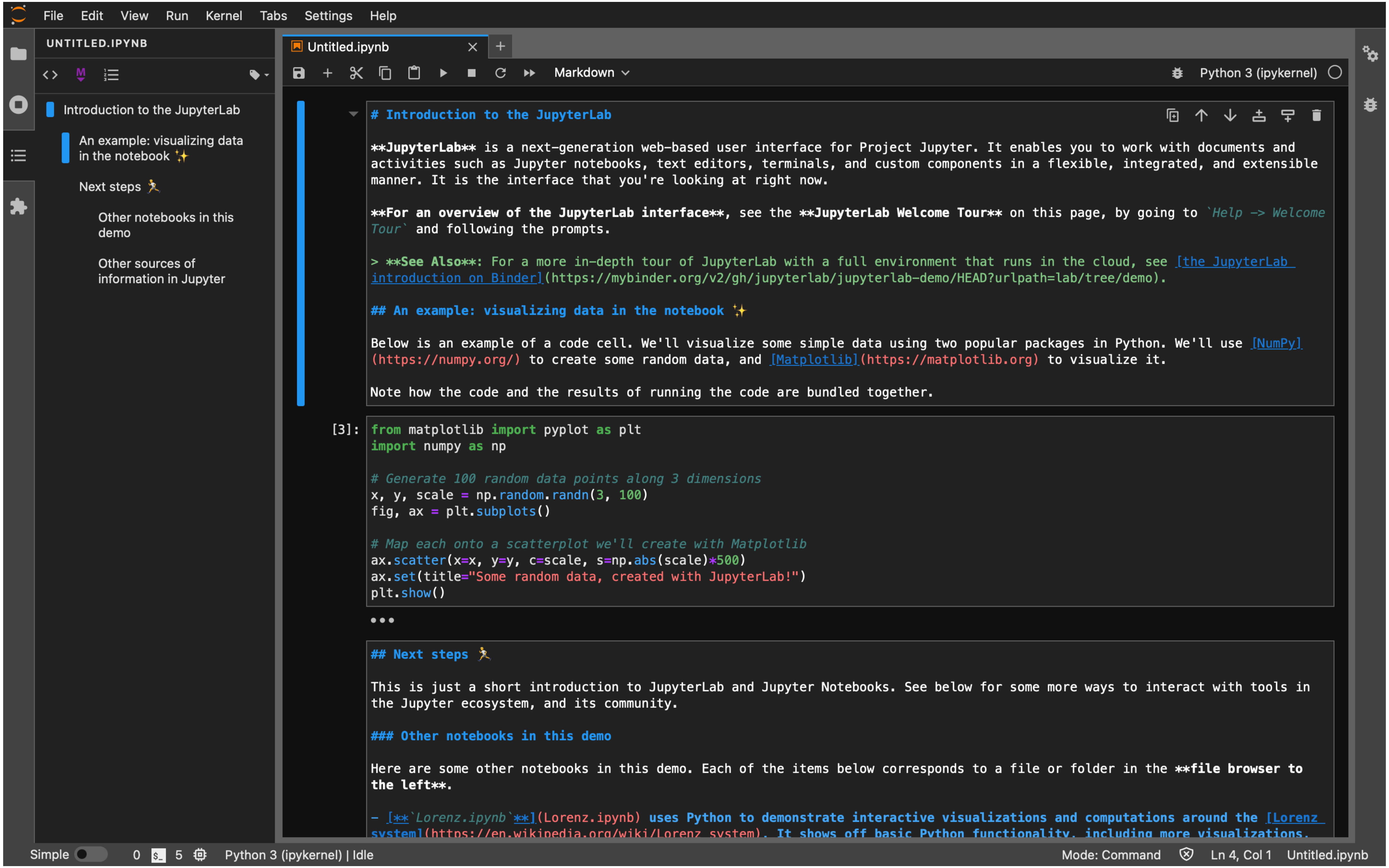
以Markdown的语法显示

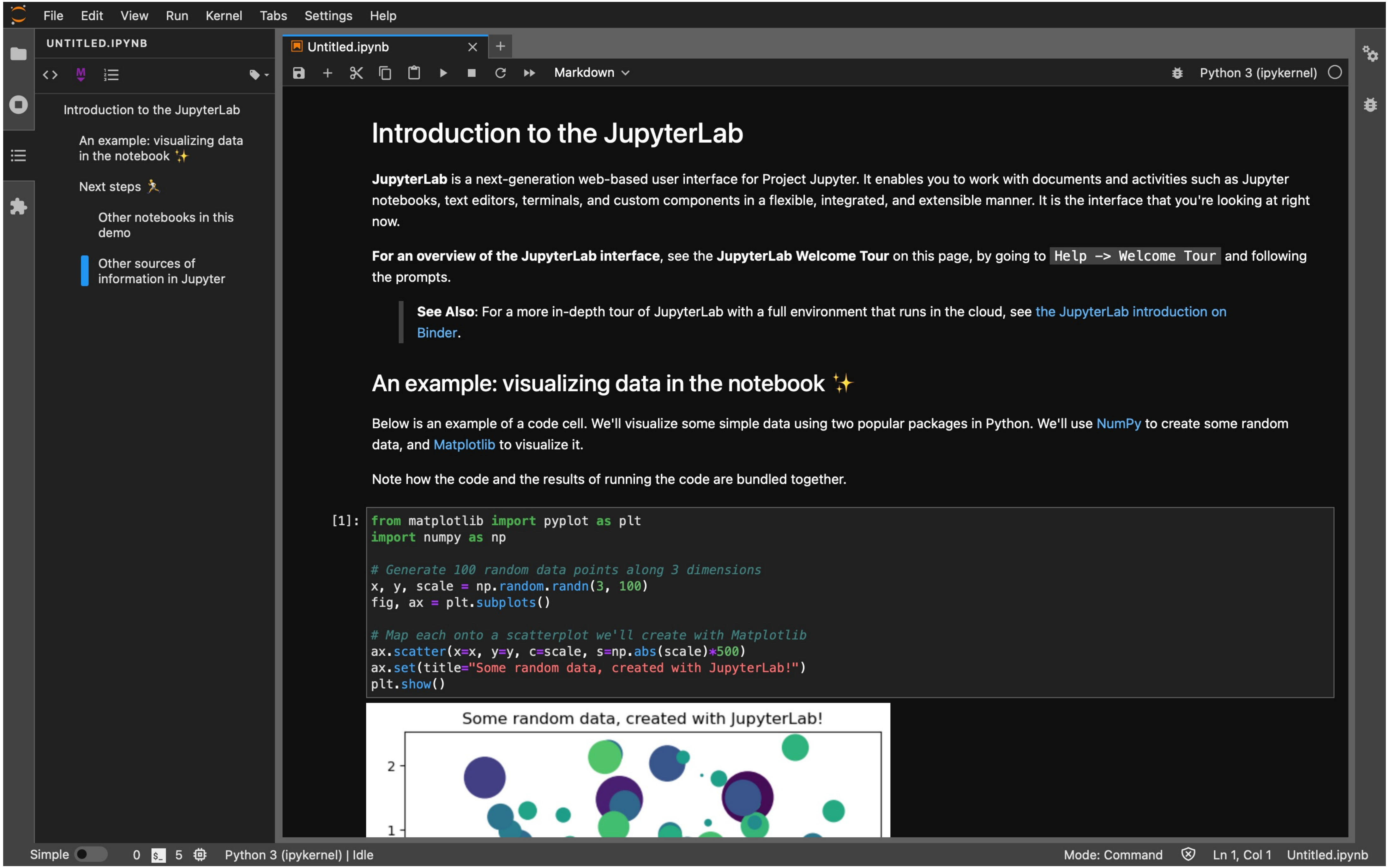
```
[3]: print('Hello World!')
```

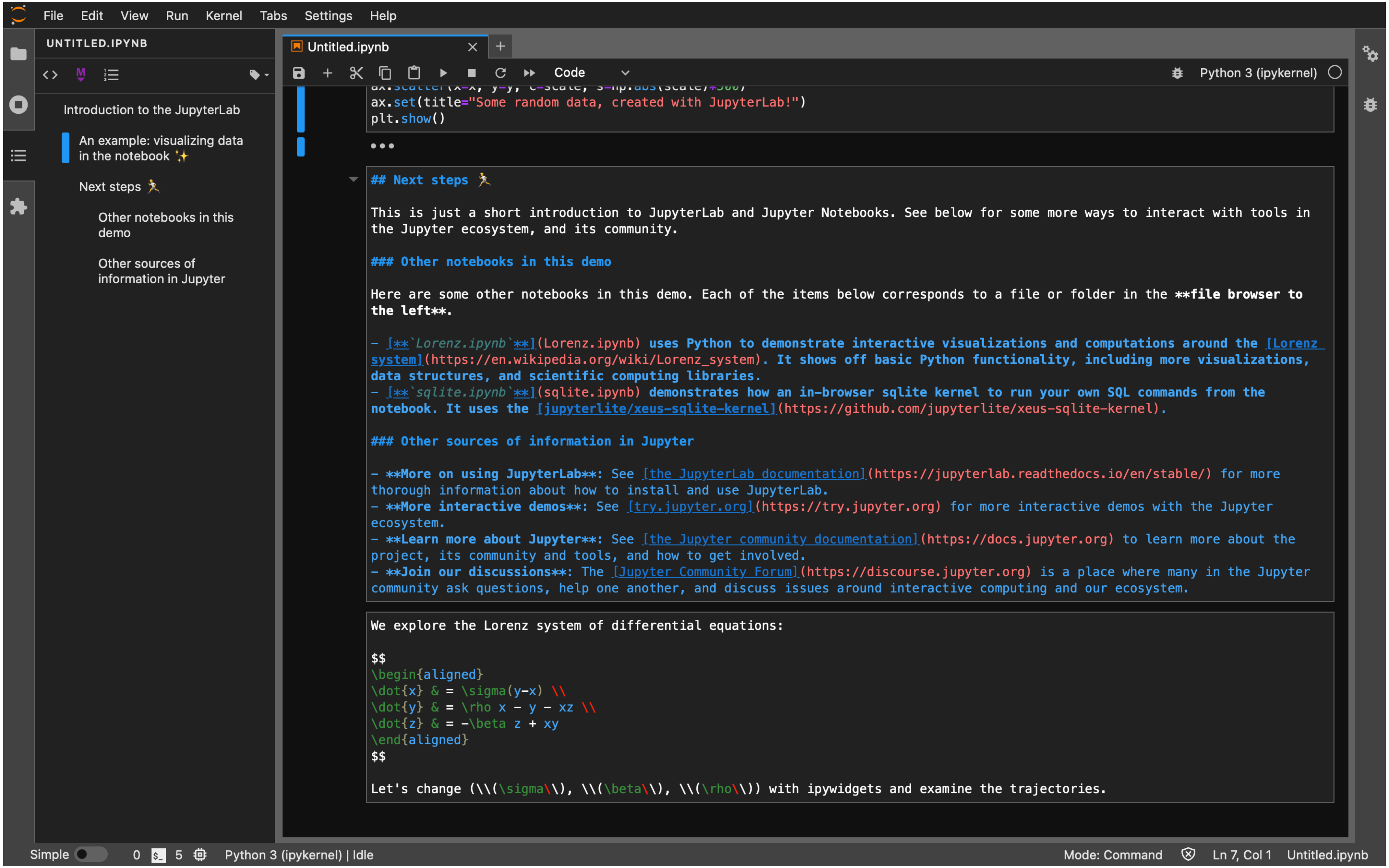
```
Hello World!
```

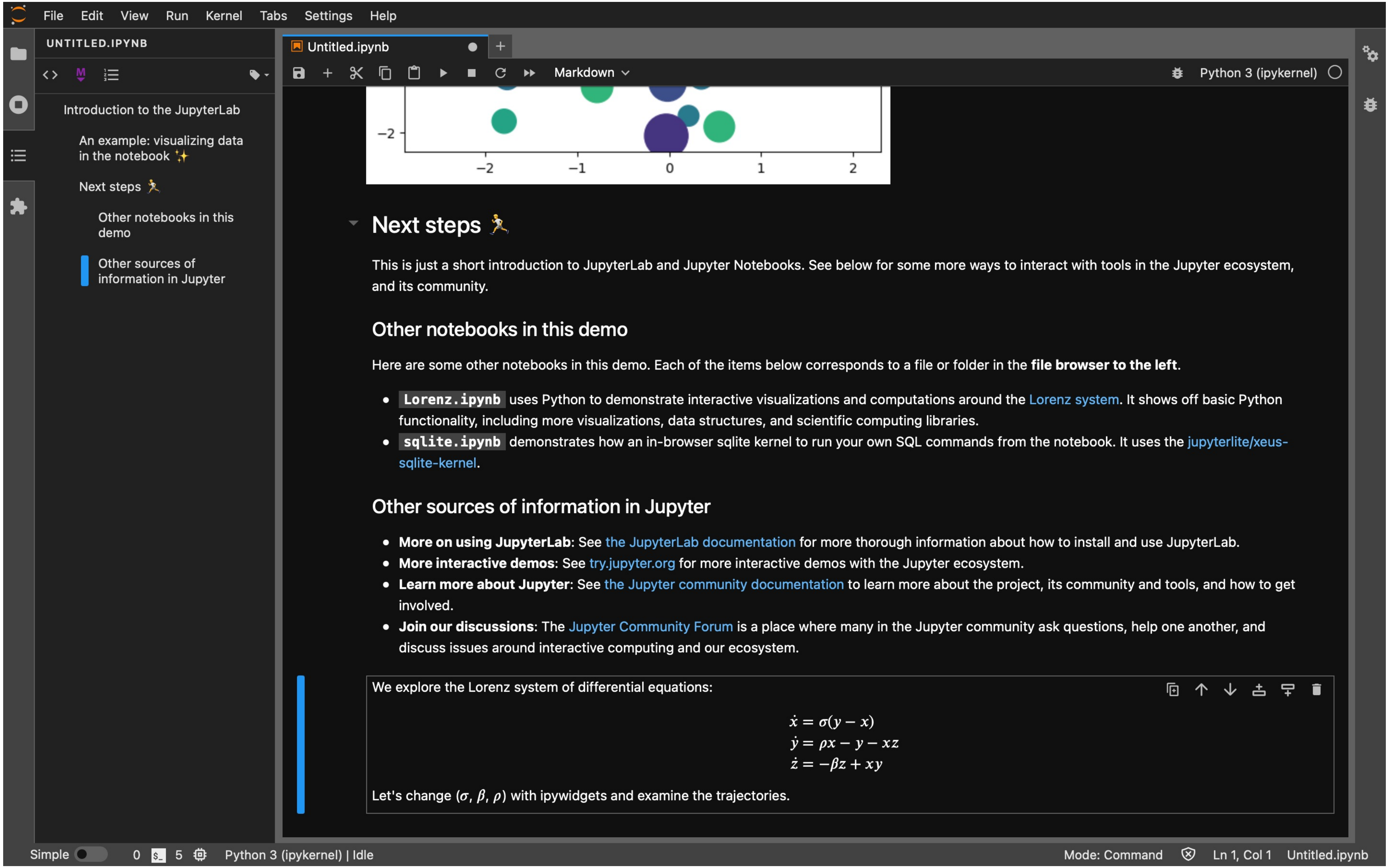
以当前选择的语言（Python）运行并将结果直接显示在下方

```
我只是一段最朴素的文字。
```
无变化

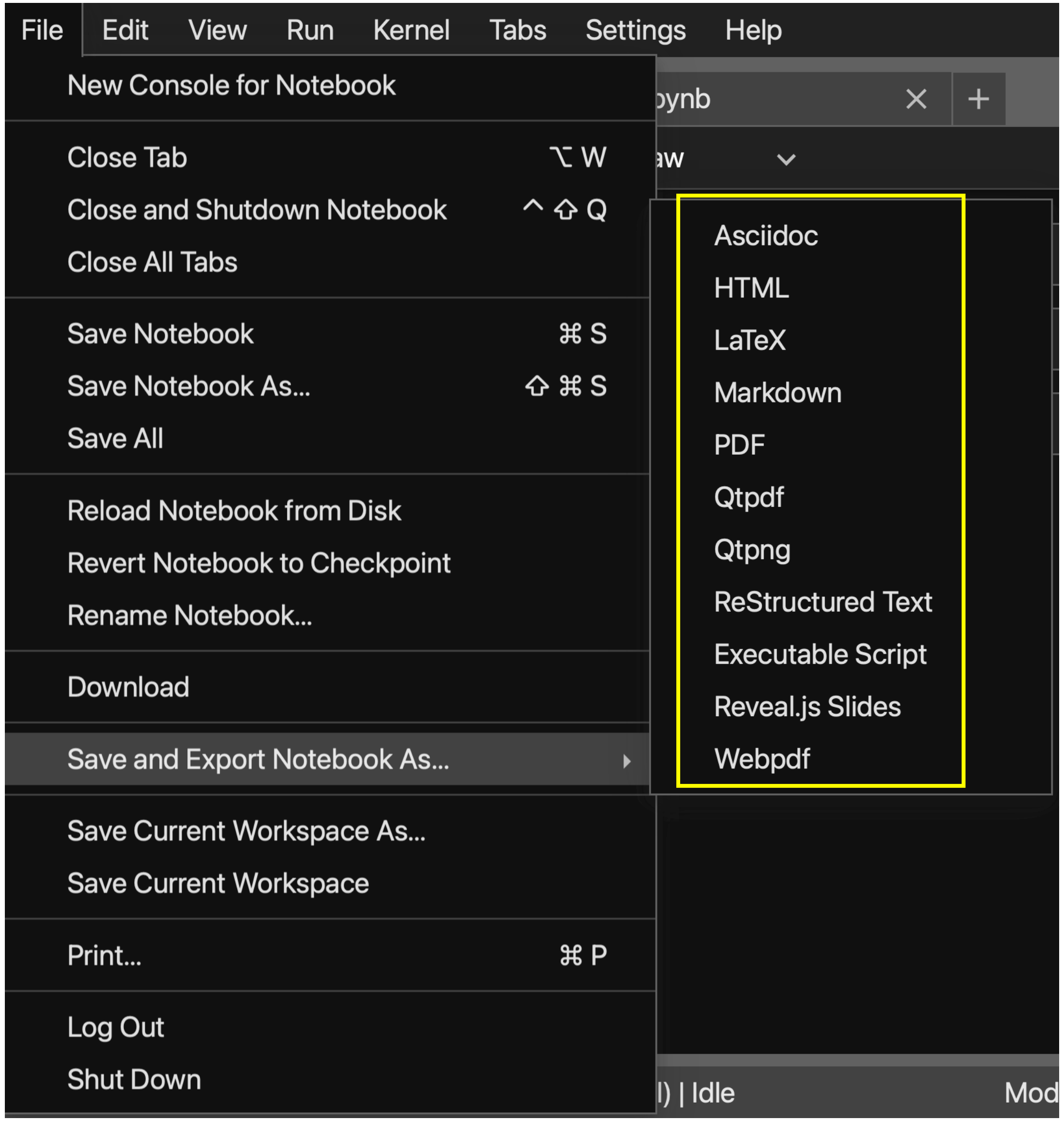

File Edit View Run Kernel Tabs Settings Help
UNTITLED.IPYNB
Introduction to the JupyterLab
An example: visualizing data in the notebook ✨
Next steps
Other notebooks in this demo
Other sources of information in Jupyter
Untitled.ipynb
Markdown
Python 3 (ipykernel)
Introduction to the JupyterLab
JupyterLab is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. It is the interface that you're looking at right now.
For an overview of the JupyterLab interface, see the **JupyterLab Welcome Tour** on this page, by going to `Help -> Welcome Tour` and following the prompts.
> **See Also**: For a more in-depth tour of JupyterLab with a full environment that runs in the cloud, see [the JupyterLab introduction on Binder](https://mybinder.org/v2/gh/jupyterlab/jupyterlab-demo/HEAD?urlpath=lab/tree/demo).
An example: visualizing data in the notebook ✨
Below is an example of a code cell. We'll visualize some simple data using two popular packages in Python. We'll use [NumPy](https://numpy.org/) to create some random data, and [Matplotlib](https://matplotlib.org) to visualize it.
Note how the code and the results of running the code are bundled together.
[3]:
from matplotlib import pyplot as plt
import numpy as np
# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()
# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
...
Next steps
This is just a short introduction to JupyterLab and Jupyter Notebooks. See below for some more ways to interact with tools in the Jupyter ecosystem, and its community.
Other notebooks in this demo
Here are some other notebooks in this demo. Each of the items below corresponds to a file or folder in the **file browser to the left**.
- [**`Lorenz.ipynb`**](Lorenz.ipynb) uses Python to demonstrate interactive visualizations and computations around the [Lorenz system](https://en.wikipedia.org/wiki/Lorenz_system). It shows off basic Python functionality, including more visualizations,
Simple 0 5 Python 3 (ipykernel) | Idle
Mode: Command Ln 4, Col 1 Untitled.ipynb


File Edit View Run Kernel Tabs Settings Help
UNTITLED.IPYNB
Untitled.ipynb
Markdown
Python 3 (ipykernel)
Introduction to the JupyterLab
An example: visualizing data in the notebook ✨
Next steps 🏃
Other notebooks in this demo
Other sources of information in Jupyter
Introduction to the JupyterLab
JupyterLab is a next-generation web-based user interface for Project Jupyter. It enables you to work with documents and activities such as Jupyter notebooks, text editors, terminals, and custom components in a flexible, integrated, and extensible manner. It is the interface that you're looking at right now.
For an overview of the JupyterLab interface, see the JupyterLab Welcome Tour on this page, by going to Help -> Welcome Tour and following the prompts.
See Also: For a more in-depth tour of JupyterLab with a full environment that runs in the cloud, see the JupyterLab introduction on Binder.
An example: visualizing data in the notebook ✨
Below is an example of a code cell. We'll visualize some simple data using two popular packages in Python. We'll use NumPy to create some random data, and Matplotlib to visualize it.
Note how the code and the results of running the code are bundled together.
[1]:
from matplotlib import pyplot as plt
import numpy as np
# Generate 100 random data points along 3 dimensions
x, y, scale = np.random.randn(3, 100)
fig, ax = plt.subplots()
# Map each onto a scatterplot we'll create with Matplotlib
ax.scatter(x=x, y=y, c=scale, s=np.abs(scale)*500)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
Some random data, created with JupyterLab!
2
1
Simple 0 5 Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
Untitled.ipynb


File Edit View Run Kernel Tabs Settings Help
UNTITLED.IPYNB
Introduction to the JupyterLab
An example: visualizing data in the notebook ✨
Next steps 🏃
Other notebooks in this demo
Other sources of information in Jupyter
Untitled.ipynb
Code
Python 3 (ipykernel)
ax.set(title="Some random data, created with JupyterLab!")
plt.show()
Next steps 🏃
This is just a short introduction to JupyterLab and Jupyter Notebooks. See below for some more ways to interact with tools in the Jupyter ecosystem, and its community.
Other notebooks in this demo
Here are some other notebooks in this demo. Each of the items below corresponds to a file or folder in the **file browser to the left**.
- [**`Lorenz.ipynb`**](Lorenz.ipynb) uses Python to demonstrate interactive visualizations and computations around the [Lorenz system](https://en.wikipedia.org/wiki/Lorenz_system). It shows off basic Python functionality, including more visualizations, data structures, and scientific computing libraries.
- [**`sqlite.ipynb`**](sqlite.ipynb) demonstrates how an in-browser sqlite kernel to run your own SQL commands from the notebook. It uses the [jupyterlite/xeus-sqlite-kernel](https://github.com/jupyterlite/xeus-sqlite-kernel).
Other sources of information in Jupyter
- **More on using JupyterLab**: See [the JupyterLab documentation](https://jupyterlab.readthedocs.io/en/stable/) for more thorough information about how to install and use JupyterLab.
- **More interactive demos**: See try.jupyter.org for more interactive demos with the Jupyter ecosystem.
- **Learn more about Jupyter**: See [the Jupyter community documentation](https://docs.jupyter.org) to learn more about the project, its community and tools, and how to get involved.
- **Join our discussions**: The [Jupyter Community Forum](https://discourse.jupyter.org) is a place where many in the Jupyter community ask questions, help one another, and discuss issues around interactive computing and our ecosystem.
We explore the Lorenz system of differential equations:
$$
\begin{aligned}
\dot{x} & = \sigma(y-x) \\
\dot{y} & = \rho x - y - xz \\
\dot{z} & = -\beta z + xy
\end{aligned}
$$
Let's change (\\(\sigma\\), \\(\beta\\), \\(\rho\\)) with ipywidgets and examine the trajectories.
Simple 0 5 Python 3 (ipykernel) | Idle
Mode: Command Ln 7, Col 1 Untitled.ipynb


File Edit View Run Kernel Tabs Settings Help
UNTITLED.IPYNB
Introduction to the JupyterLab
An example: visualizing data in the notebook ✨
Next steps 🏃
Other notebooks in this demo
Other sources of information in Jupyter
Untitled.ipynb
Markdown
Python 3 (ipykernel)
−2
−2
−1
0
1
2
Next steps 🏃
This is just a short introduction to JupyterLab and Jupyter Notebooks. See below for some more ways to interact with tools in the Jupyter ecosystem, and its community.
Other notebooks in this demo
Here are some other notebooks in this demo. Each of the items below corresponds to a file or folder in the file browser to the left.
Lorenz.ipynb uses Python to demonstrate interactive visualizations and computations around the Lorenz system. It shows off basic Python functionality, including more visualizations, data structures, and scientific computing libraries.
sqlite.ipynb demonstrates how an in-browser sqlite kernel to run your own SQL commands from the notebook. It uses the jupyterlite/xeus-sqlite-kernel.
Other sources of information in Jupyter
More on using JupyterLab: See the JupyterLab documentation for more thorough information about how to install and use JupyterLab.
More interactive demos: See try.jupyter.org for more interactive demos with the Jupyter ecosystem.
Learn more about Jupyter: See the Jupyter community documentation to learn more about the project, its community and tools, and how to get involved.
Join our discussions: The Jupyter Community Forum is a place where many in the Jupyter community ask questions, help one another, and discuss issues around interactive computing and our ecosystem.
We explore the Lorenz system of differential equations:
$\dot{x} = \sigma(y - x)$
$\dot{y} = \rho x - y - xz$
$\dot{z} = -\beta z + xy$
Let's change (σ, β, ρ) with ipywidgets and examine the trajectories.
Simple
0
5
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
Untitled.ipynb

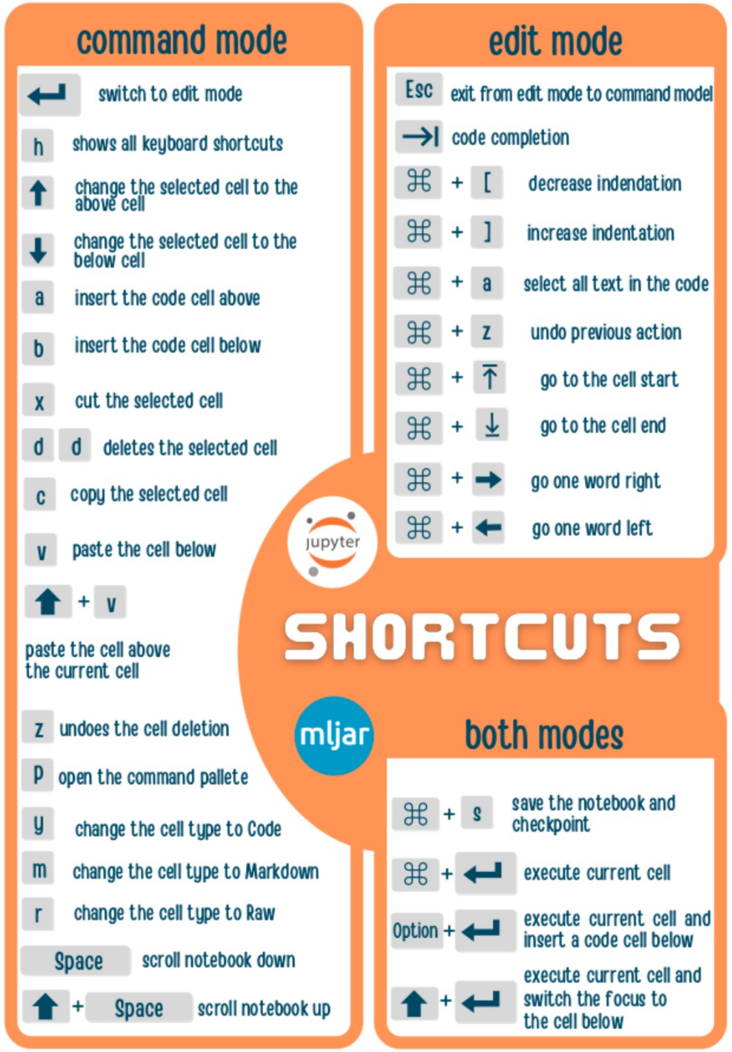
Jupyter可以导出为多种格式去分享



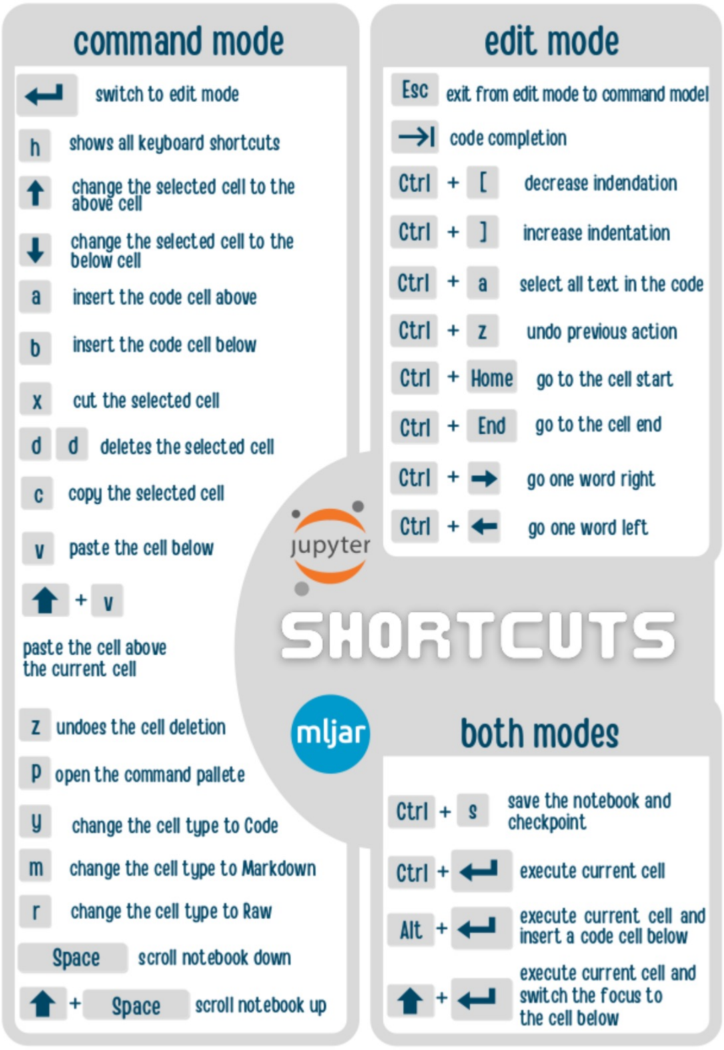
一些常用的快捷键

命令模式下一些常用的快捷键跟vim的类似

For Mac users



For Windows users



Running JupyterLab remotely

jupyter server

Users Operators Developers Contributors Other

Section Navigation

Operators

- Managing multiple extensions
- Configuring Extensions
- Migrating from Notebook Server
- Running a public Jupyter Server
- Security in the Jupyter Server
- Configuring Logging

Running a public Jupyter Server

The Jupyter Server uses a two-process kernel architecture based on ZeroMQ, as well as Tornado for serving HTTP requests.

> **Note**
>
> By default, Jupyter Server runs locally at 127.0.0.1:8888 and is accessible only from `localhost`. You may access the server from the browser using `http://127.0.0.1:8888`.

This document describes how you can secure a Jupyter server and how to run it on a public interface.

> **Important**
>
> **This is not the multi-user server you are looking for**. This document describes how you can run a public server with a single user. This should only be done by someone who wants remote access to their personal machine. Even so, doing this requires a thorough understanding of the set-ups limitations and security implications. If you allow multiple users to access a Jupyter server as it is described in this document, their commands may collide, clobber and overwrite each other.
>
> If you want a multi-user server, the official solution is JupyterHub. To use JupyterHub, you need a Unix server (typically Linux) running somewhere that is accessible to your users on a network. This may run over the public internet, but doing so introduces additional security concerns.

Securing a Jupyter server

You can protect your Jupyter server with a simple single password. As of notebook 5.0 this can be done automatically. To set up a password manually you can configure the `ServerApp.password` setting in `jupyter_server_config.py`.

On this page

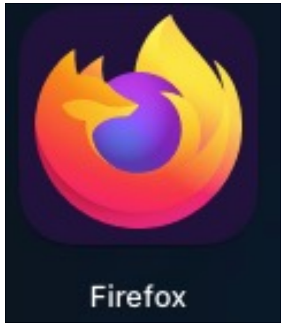
Show Source

https://jupyter-server.readthedocs.io/en/stable/operators/public-server.html

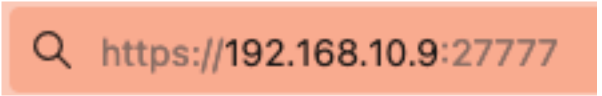
我的JupyterLab工作流的日常

1. 打开VPN

2. 打开浏览器



3. 输入Jupyter Server运行的地址和密码（上一张slide中你设置的）



4. 开始/继续工作

5. 保存Notebook

6. 关掉JupyterLab网页，结束工作

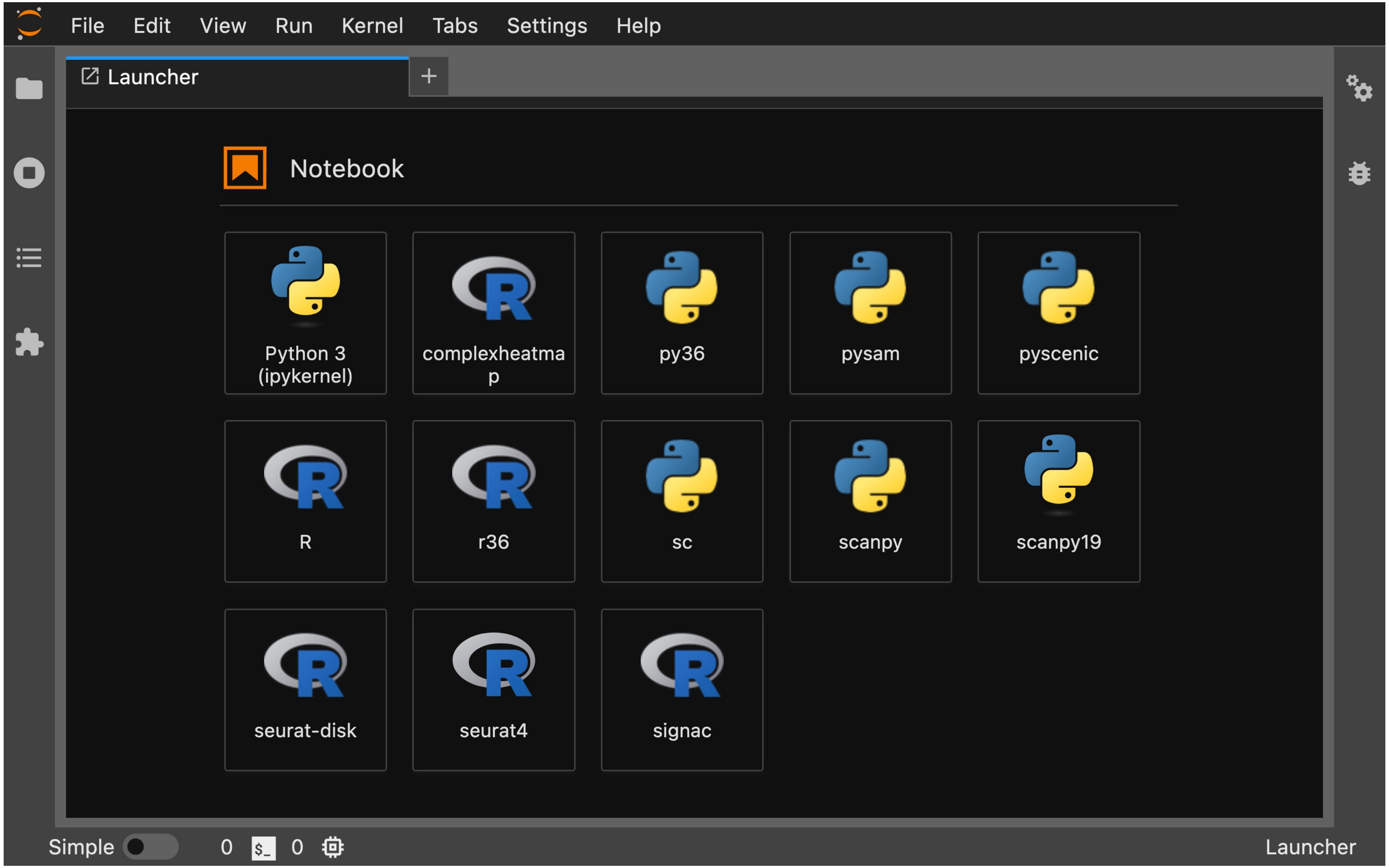
JupyterLab的多编程语言支持

1. 安装**，建议给每一种语言单独建立一个conda环境。

2. Google或Bing搜索“i**kernel”，然后点进第一个查看下载和安装方法。

3. **代表编程语言的名字，比如Julia、R、Java。

我用不同的conda环境来解决软件之间的冲突



Thanks for your attention!