

0

扫码签到，累积图书馆积分，获取定制纪念品



1

讲座预告——微信公众号“i学堂XMU”



2

WISER CLUB微信公众号





<https://lecture.xmu.edu.cn>

提示:

- 1.请先连接校外访问VPN
- 2.校外用户可通过图书馆主页获取
(主页-文档-讲座课件)



厦门大学讲座报名系统

[首页](#)
[思明校区](#)
[翔安校区](#)
[登录](#)

☰ 近期讲座

i学堂Excel可视化专题 (1) : Excel数据处理技巧 【思明校区】杨薇	2019-03-05(星期二) 19:30	点击报名
i学堂平面设计专题 (1) : 初识PS—基础上手指南 【翔安校区】王楷	2019-03-06(星期三) 19:30	点击报名
i学堂Excel可视化专题 (2) : Excel函数入门 【思明校区】杨薇	2019-03-07(星期四) 19:30	点击报名
i学堂平面设计专题 (2) : 设计元素的获取——PS之抠图大法 【翔安校区】王楷	2019-03-09(星期六) 19:30	点击报名

More...

☰ 讲座课件

i学堂翔安场: 详解文献管理利器 EndNote X9 (进阶篇) 【翔安校区】张妮妮	2018-12-18 19:30	i学堂翔安场: 详解文献管理软件 EndNote X9 (进阶篇).rar
i学堂本部场: MATLAB编程技巧与数据分析 【思明校区】许悦伊	2018-12-13 19:30	i学堂本部场: MATLAB编程技巧与数据分析.rar
i学堂翔安场: MATLAB编程技巧与数据分析 【翔安校区】许悦伊	2018-12-12 19:30	i学堂翔安场: MATLAB编程技巧与数据分析.rar
i学堂本部场: 详解文献管理利器 EndNote X9 (入门篇) 【思明校区】韩冬丽	2018-12-11 19:30	i学堂本部场: 详解文献管理软件 EndNote X9-韩冬丽.pptx

More...

☰ 联系我们

请使用“我的图书馆”的帐号(读者证号或学号、教工号)和密码登录本系统。

网站使用问题 <ul style="list-style-type: none"> • 联系老师: 魏老师 • 联系电话: 0592-2184973-810 • 联系邮箱: xywei@xmu.edu.cn 	讲座意见及建议(思明) <ul style="list-style-type: none"> • 联系老师: 麦老师 • 联系电话: 0592-2188693 • 联系邮箱: mailin@xmu.edu.cn 	讲座意见及建议(翔安) <ul style="list-style-type: none"> • 联系老师: 李老师 • 联系电话: 0592-2888315 • 联系邮箱: shining@xmu.edu.cn
--	--	---

☰ 用户登录

用户名*

密码*

• [重设密码](#)

☰ 导航

- i学堂
- 宣传月
- 新生培训
- 在线培训
- 往期讲座

☰ 讲座日历

<< 2019年3月 >>

日	一	二	三	四	五	六
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23



- 哔哩哔哩网站 (<https://www.bilibili.com>) 搜索“厦大图书馆”，可学习往期课程

共找到1个用户

默认排序

全部用户



厦大图书馆

LV2

+ 关注

稿件: 14 粉丝: 832

此用户没有个性签名啊啊啊

厦门大学图书馆 i学堂



i学堂-毕业论文WORD排版全攻略(2)-杨薇-202003
2020-03-17

厦门大学图书馆 i学堂



i学堂-如何运用LaTex排版论文-魏小燕-20200306
2020-03-10



i学堂-文献管理软件
EndNote X9使用入门-韩
2020-03-07

[全部14个稿件>](#)

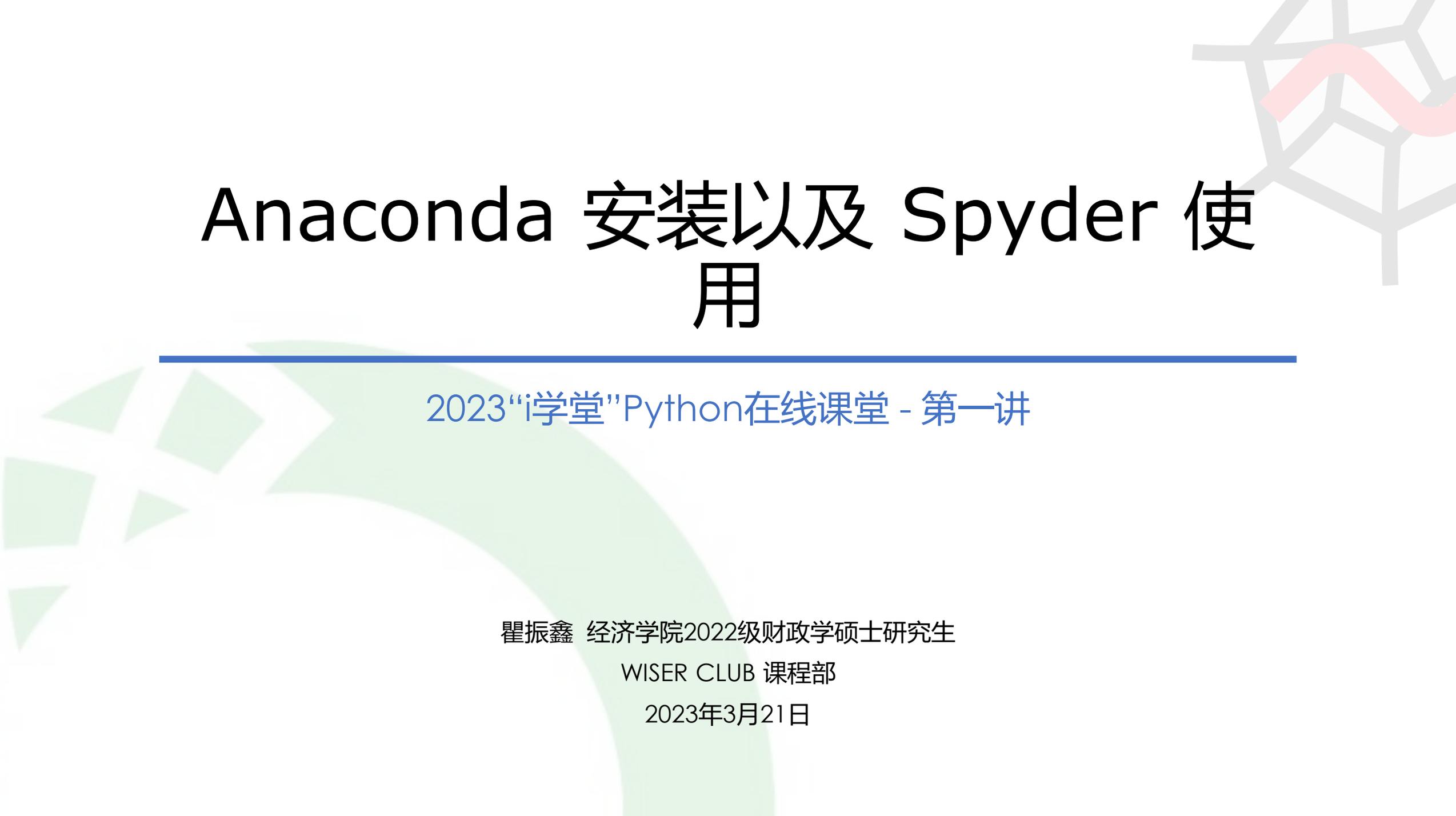
5

Python训练营QQ交流群 (884539407)



群名称:2021-2023 “i学堂” Pytho...
群 号:884539407

Anaconda 安装以及 Spyder 使用



2023“i学堂”Python在线课堂 - 第一讲

瞿振鑫 经济学院2022级财政学硕士研究生

WISER CLUB 课程部

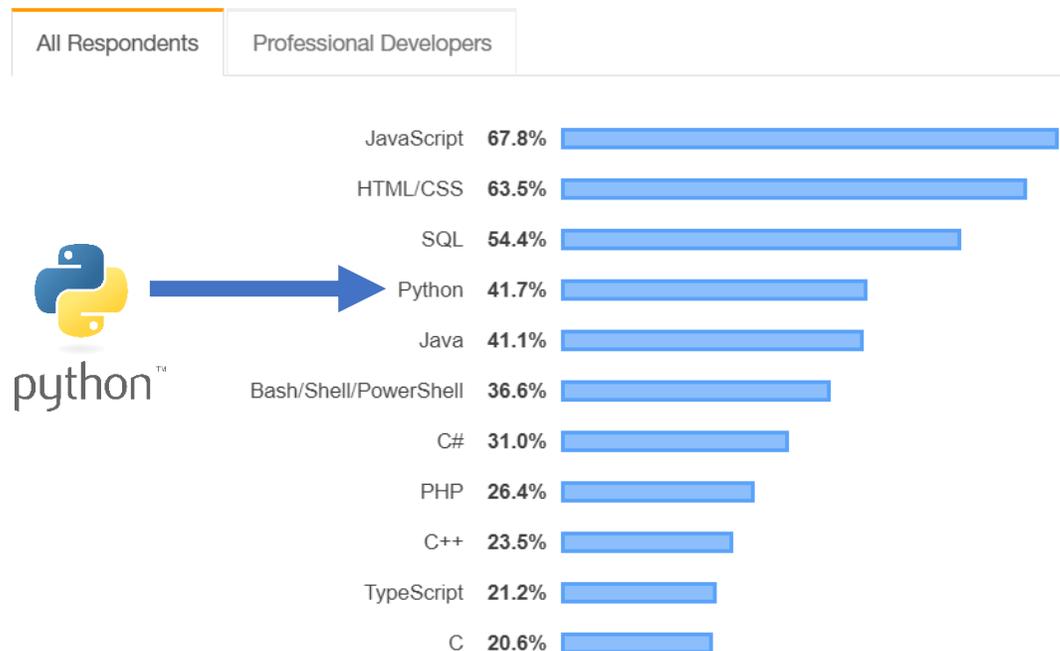
2023年3月21日

Developer Survey Results 2019



Most Popular Technologies

Programming, Scripting, and Markup Languages

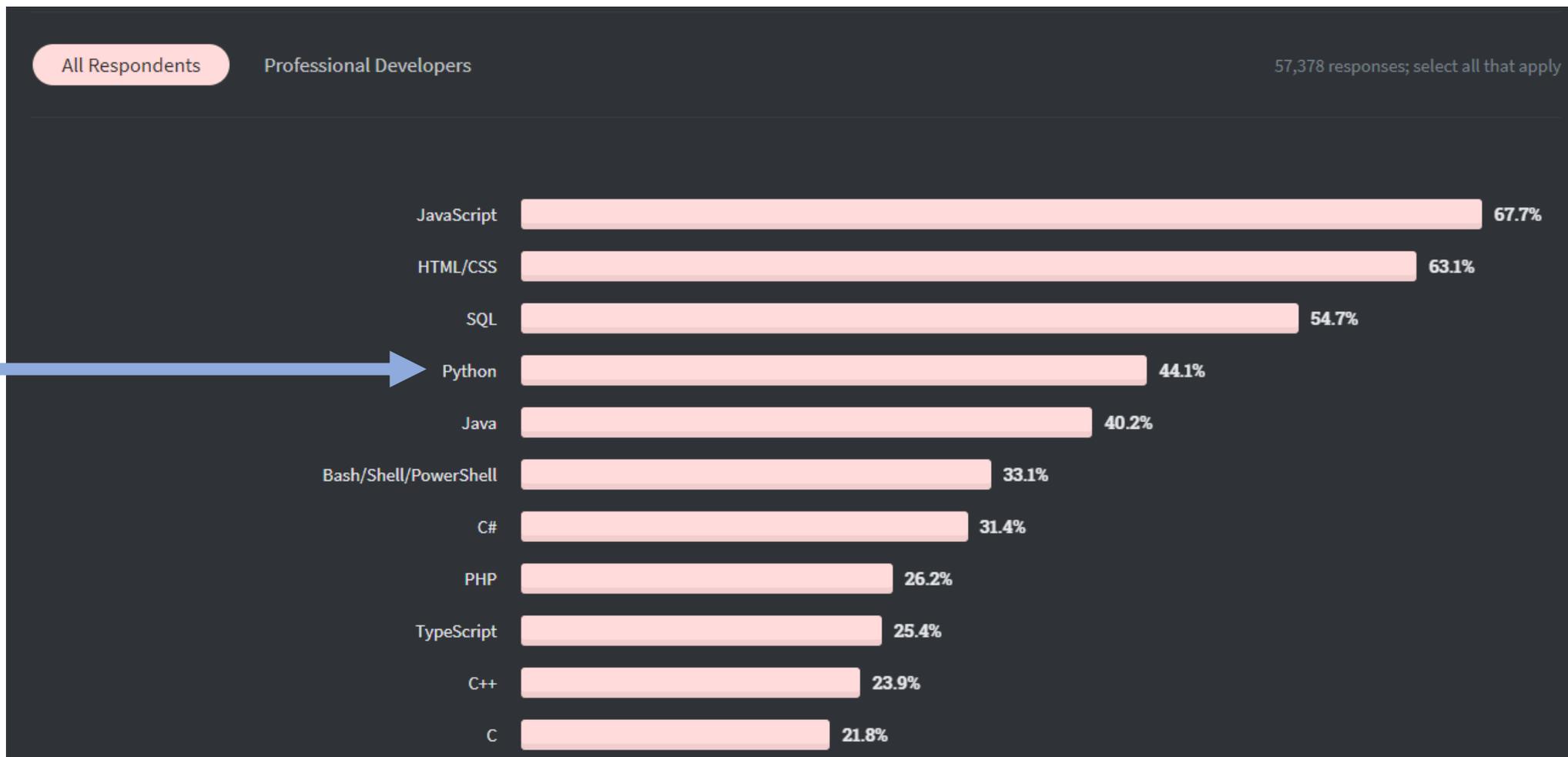


87,354 responses; select all that apply

For the seventh year in a row, JavaScript is the most commonly used programming language, but Python has *risen in the ranks again*. This year, Python just edged out Java in overall ranking, much like it surpassed C# last year and PHP the year before. Python is the *fastest-growing* major programming language today.



Developer Survey Results 2020

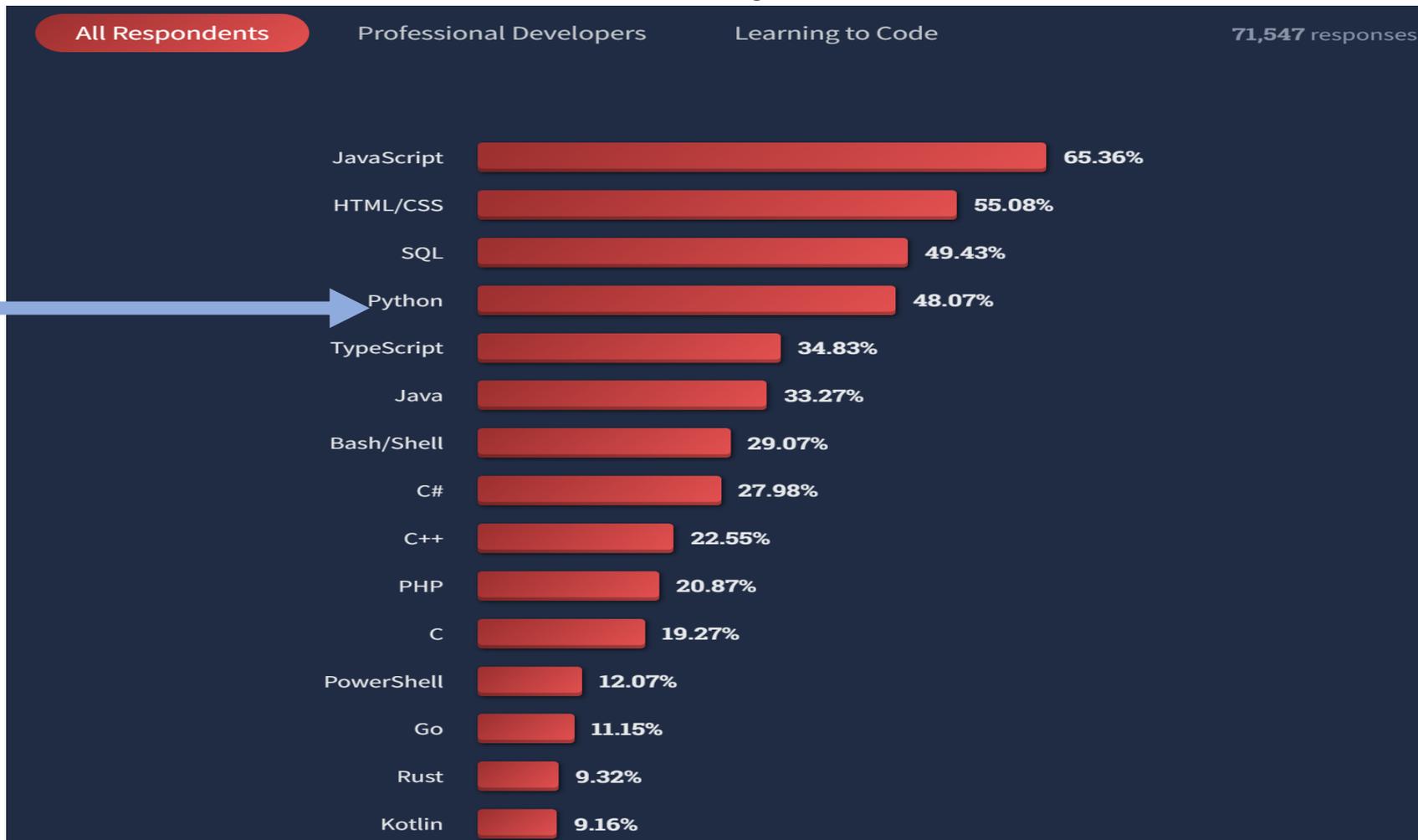


来源: [Stack Overflow Developer Survey Results 2020](https://survey.stackoverflow.co/2020/)

Developer Survey Results 2021



Developer Survey Results 2022





VS



学习难度大
程序设计复杂
适合**造轮子**

易学易用
快速开发和部署
适合**组装轮子**

主要内容



Python

为什么选择Python?

Python可以用来干什么

Anaconda

下载与安装Anaconda

使用pip安装Python包

使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook

Jupyter Lab

为什么选择Python?



python™

免费开源

- **容易上手**
无须花费过多精力
- **语法简洁直观，开发效率高**
可以专注于解决问题而不是语法和软硬件底层的细节
- **功能强大，可扩展性好**
各种库功能十分丰富，基本上你能想到的，实用的功能都有现成库可以使用
- **可移植性好**
代码可以不经修改运行于各种平台

主要内容



Python

为什么是Python?

Python可以用来干什么

Anaconda

下载与安装Anaconda

使用pip安装Python包

使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook

Jupyter Lab

豆瓣 **douban**

知乎

 **Dropbox**

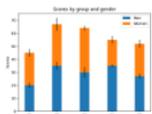
使用Python搭建的网站/应用

网站与互联网开发

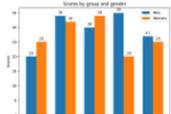
Web and Internet Development

- 成熟的框架：[Django](#)与[Flask](#)
- 标准库对互联网协议的支持：
 - [HTML与XML](#)
 - [JSON](#)
 - [E-mail处理](#)
 - 支持[FTP](#)，[IMAP](#)，以及其他[互联网协议](#)
- 还有很多有用的库，比如**爬虫**经常用到的：
 - [Requests](#)，强有力的HTTP库
 - [BeautifulSoup](#)，从HTML或XML文件中提取数据
 - ...

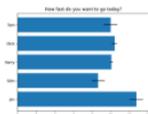
Lines, bars and markers



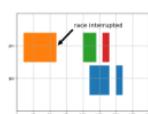
Stacked Bar Graph



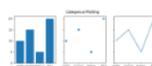
Grouped bar chart with labels



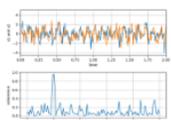
Horizontal bar chart



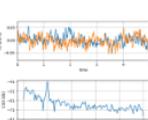
Broken Bar



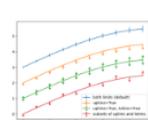
Plotting categorical variables



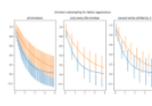
Plotting the coherence of two signals



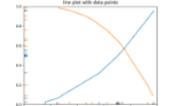
CSD Demo



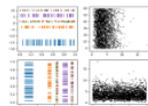
Errorbar limit selection



Errorbar Subsample



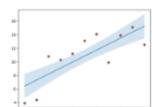
EventCollection Demo



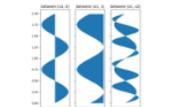
Eventplot Demo



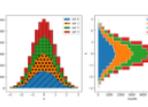
Filled polygon



Filling the area between lines



Fill Betweenx Demo



Hatch-filled histograms



Bar chart with gradients

科学与数值计算 Scientific and Numeric

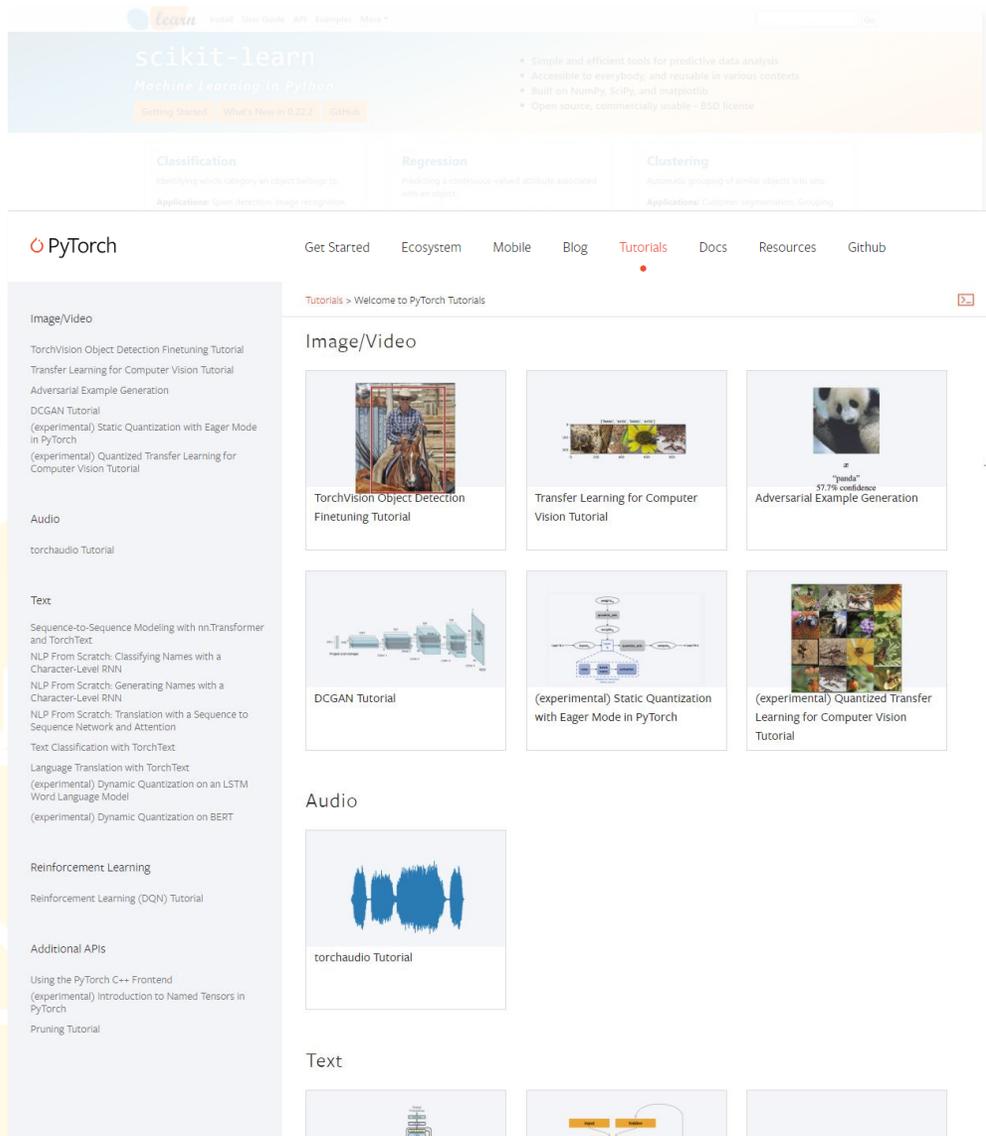
SciPy系列：用于数学、科学与工程的开源库集成

- [NumPy](#)，使用Python进行科学计算的基础软件包
- [Pandas](#)，以NumPy为基础，强大的分析结构化数据的工具集
- [Matplotlib](#)，Python的2D绘图库
- [IPython](#)，强大的交互式shell

MATLAB和R能做到的东西，Python基本上都能做！



The screenshot shows the NumPy website's navigation menu with links for '概述', '文章', '深度学习', '用户指南', '参考手册', '精选资源', '其他文档', '了解更多', and '选择语言'. A search bar is visible. The main heading is '与 Matlab 比较'. Below it, there is an '介绍' (Introduction) section with a paragraph of text. Further down, there is a section titled '一些关键的差异' (Some key differences) which includes a table with columns for 'MATLAB' and 'NumPy'.



机器学习与深度学习

Machine Learning and Deep Learning

机器学习: [scikit-learn](#), 简称sklearn

- 支持包括分类，回归，降维和聚类四大机器学习算法
- 还包括了特征提取，数据处理和模型评估三大模块

深度学习:

- [PyTorch](#)

更有利于研究人员、爱好者、小规模项目等快速搞出原型

- [TensorFlow](#)

更适合大规模部署，特别是需要跨平台和嵌入式部署时

- [Keras](#)，基于Tensorflow、Theano以及CNTK后端，

为支持快速实验而生，能够把你的idea迅速转换为结果

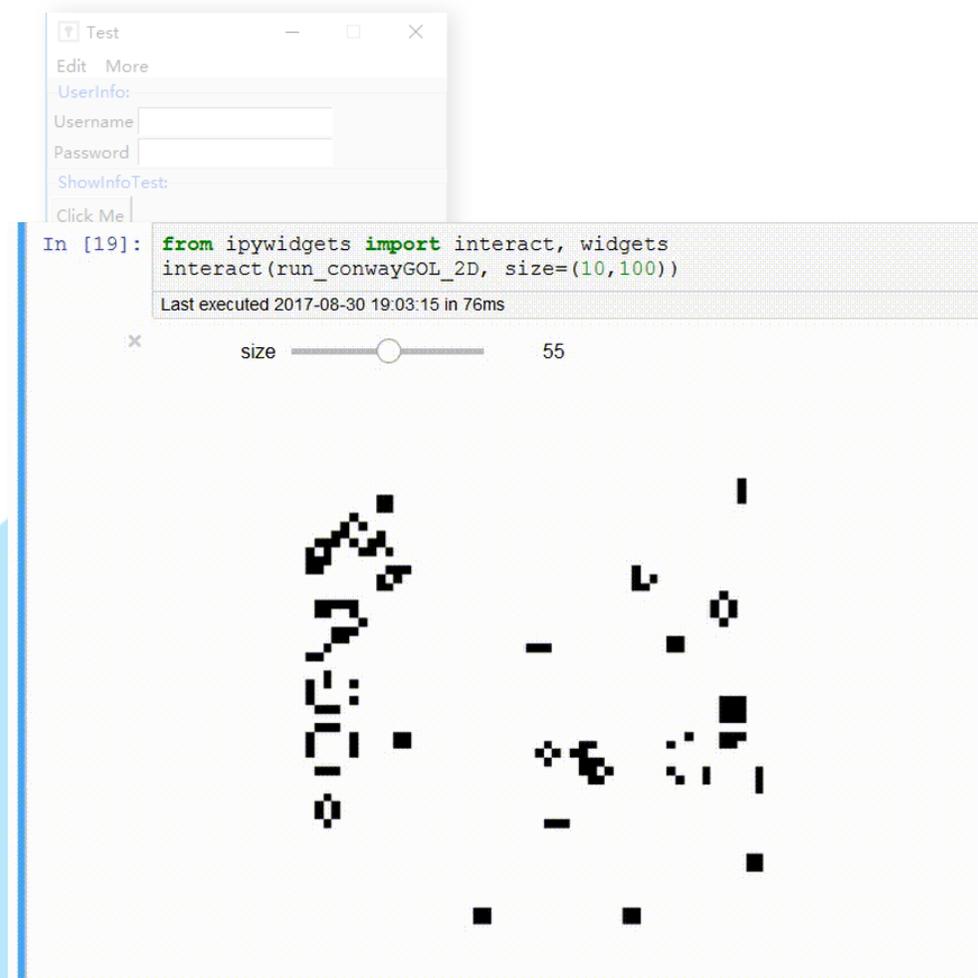
桌面GUI（图形用户界面）开发 Desktop GUIs

桌面程序/应用的GUI开发

- [Tkinter](#)，Python自带的GUI库
 - 与 Tk/Tcl 关系最密切的图形库，用户广泛
 - 跨平台运行，支持 windows/macOS/Linux
- [PyQt](#)
- [wxPython](#)

Jupyter Notebook上的交互界面设计

- [ipywidget](#)



• 为什么是Python?

- 容易上手
- 语法简洁直观，开发效率高
- 功能强大，可扩展性好
- 可移植性好
- 开源！开源！开源！

• Python可以用来干什么?

- 网站与互联网开发
- 科学与数值计算
- 机器学习与深度学习
- 桌面GUI (图形用户界面) 开发
- 其实还有很多.....





ANACONDA[®]



ANACONDA®



Python

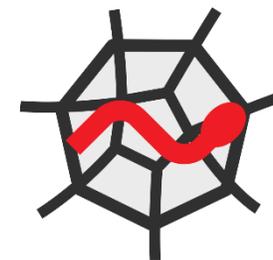
+



常用第三方包

如：SciPy系列下的NumPy,
Pandas, Matplotlib等

+

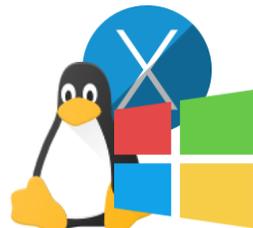


常用IDE

如：Spyder,
Jupyter Notebook,
Jupyter Lab



用户体验好



多平台支持



环境与
包管理工具

特性

主要内容



Python

为什么是Python?

Python可以用来干什么

Anaconda

下载与安装Anaconda

使用pip安装Python包

使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook

Jupyter Lab

有成功地安装了Anaconda吗?

选项	回答
A	安装好了
B	没装好

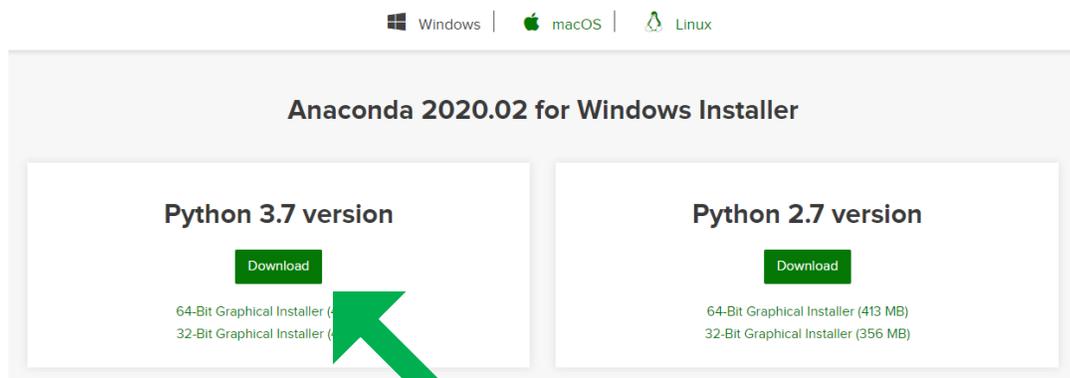
安装教程参考_windows系统: <https://blog.csdn.net/fan18317517352/article/details/123035625>

安装教程参考_macOS系统: <https://anaconda.org.cn/anaconda/install/mac-os/>

下载与安装Anaconda

首先，下载Anaconda安装包

可以到[Anaconda官方网站](https://www.anaconda.com/)下载



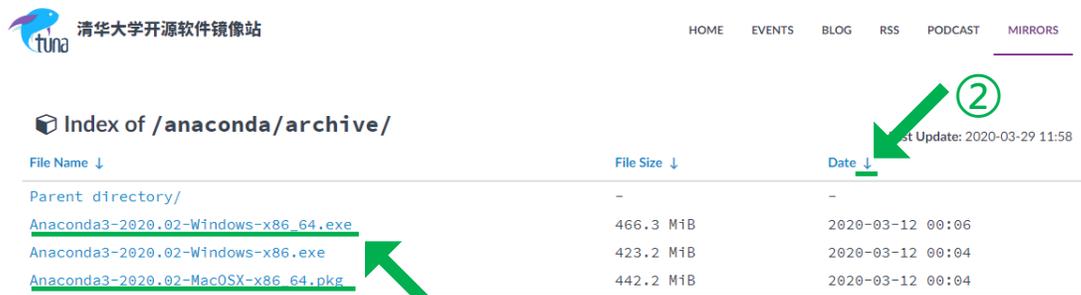
考虑到官网速度可能会很慢，

建议到[清华大学开源软件镜像站](https://mirrors.tuna.tsinghua.edu.cn/)下载

Anaconda 镜像使用帮助

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

Anaconda 安装包可以到 <https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/> 下载。



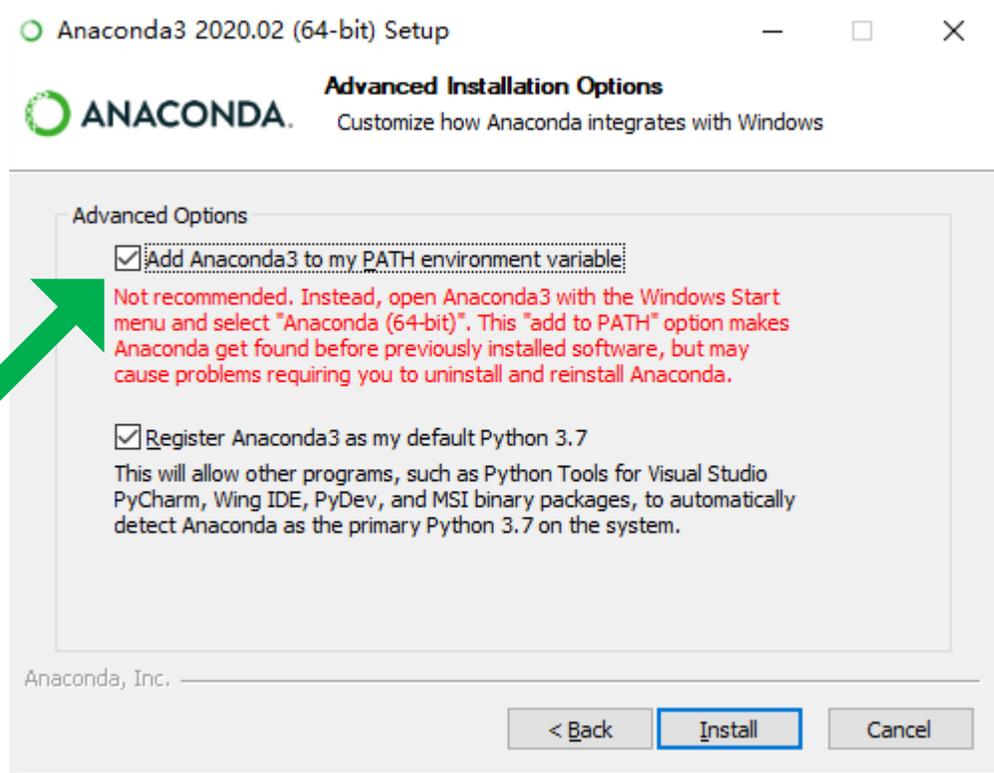
<https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/>

下载与安装Anaconda

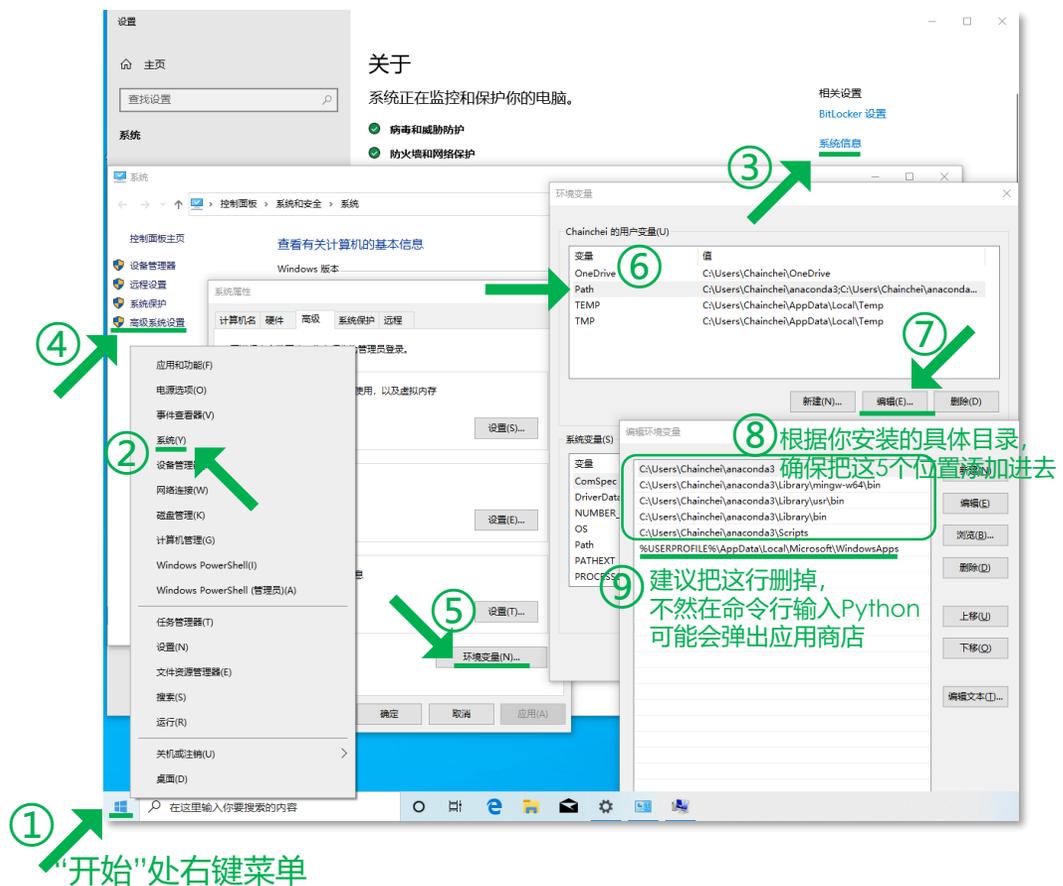
然后是安装

基本上一直next就行，安装目录可以自己选择

Windows平台需要注意：这里的Add PATH要勾上



如果忘了勾上，后面需要在系统设置里自行添加

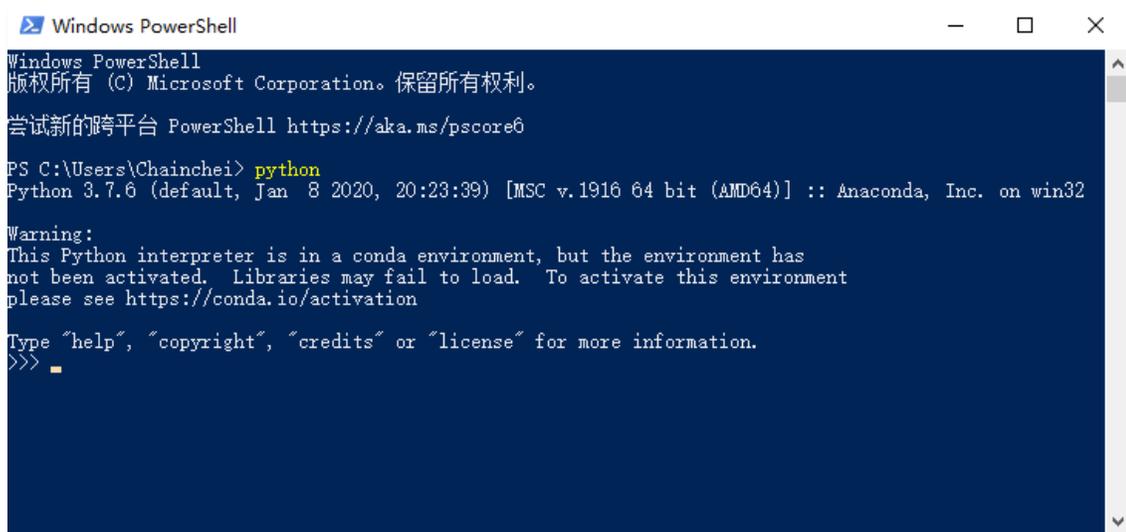


下载与安装Anaconda

安装好了，试试吧

在命令行里输入 `python`

Windows: 打开“命令提示符”或“PowerShell”



```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

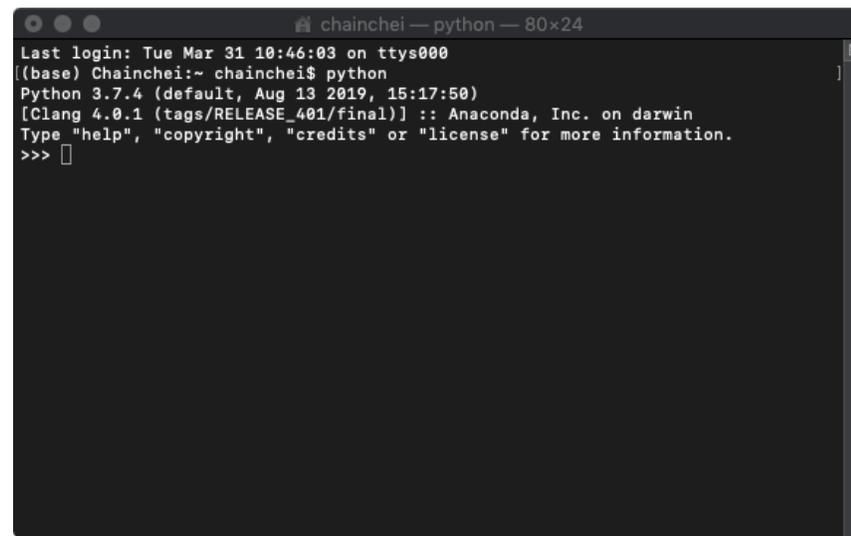
尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\Chainchei> python
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

MacOS: 打开“终端”



```
chainchei — python — 80x24
Last login: Tue Mar 31 10:46:03 on ttys000
(base) Chainchei:~ chainchei$ python
Python 3.7.4 (default, Aug 13 2019, 15:17:50)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

进入Python后，输入 `print("Hello World!")`

通过输入 `exit()` 从Python退出到命令行

主要内容



Python

为什么是Python?

Python可以用来干什么

Anaconda

下载与安装Anaconda

使用pip安装Python包

使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook

Jupyter Lab

变量与赋值

3

变量与赋值

a = 3
名字 引用 **→** **对象**

```
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: An
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> 3
3
>>> a = 3
>>> a
3
>>> ■
```

包与模块的概念



module_1.py

模块 (module)

- 有时候也被称作**库 (library)**
- 说白了，就是Python文件
- 其中通常包含 Python 对象定义和Python语句
- 提供了一定的功能供别人调用

包与模块的概念

模块 (module)

- 有时候也被称作**库 (library)**
- 说白了，就是Python文件
- 其中通常包含 Python 对象定义和Python语句
- 提供了一定的功能供别人调用



module_1.py

※超纲内容

- 通过import module_1来调用模块
- 然后就可以通过module_1.func_1()来调用module_1中的func_1函数

func_1()

func_2()

...

Class_1()

Class_2()

...

函数 (function)

- 组织好的，可重复使用的，用来实现单一，或相关联功能的代码段
- 在括号输入**参数 (parameters)**，它为你干活
- 比如 print(“Hello World!”)

类 (class)

- 简单理解：对象的一个模板代码段
- 可以通过这个模板去创建一类对象

对象 (object)

- 万物皆对象，函数和类也是对象
- 对于函数和类
 - 加括号表示调用，不加括号指代对象本身
- 比如：print可赋值给变量，也可作函数的参数

使用pip安装Python包

包与模块的概念

The image shows a screenshot of the Visual Studio Code editor with a Python file named `my_module_1.py` open. The code is as follows:

```
1 class Song(object):
2
3     def __init__(self, lyrics):
4         self.lyrics = lyrics
5
6     def sing_me_a_song(self):
7         for line in self.lyrics:
8             print(line)
9
10 def load_a_song(file):
11
12     with open(file, encoding="utf-8"):
13         lyrics = fp.read().splitlines()
14
15     return lyrics
16
17 info_module = "这就是一个模块"
18
```

Annotations are overlaid on the code:

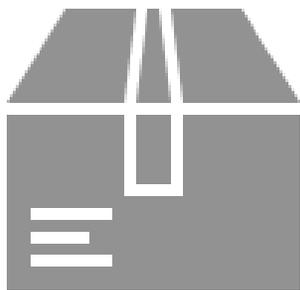
- A green box highlights the `class Song(object):` definition, with the Chinese character **类** (Class) and the English word **Class** next to it.
- A green box highlights the `def load_a_song(file):` function definition, with the Chinese character **函数** (Function) and the English word **Function** next to it.
- A green box highlights the `info_module = "这就是一个模块"` line, with the Chinese text **其他对象的定义** (Other Objects) and the English text **Other Objects** next to it.

The status bar at the bottom indicates the Python 3.7.4 64-bit environment, with the current cursor position at Line 18, Column 1.

包与模块的概念

包 (package)

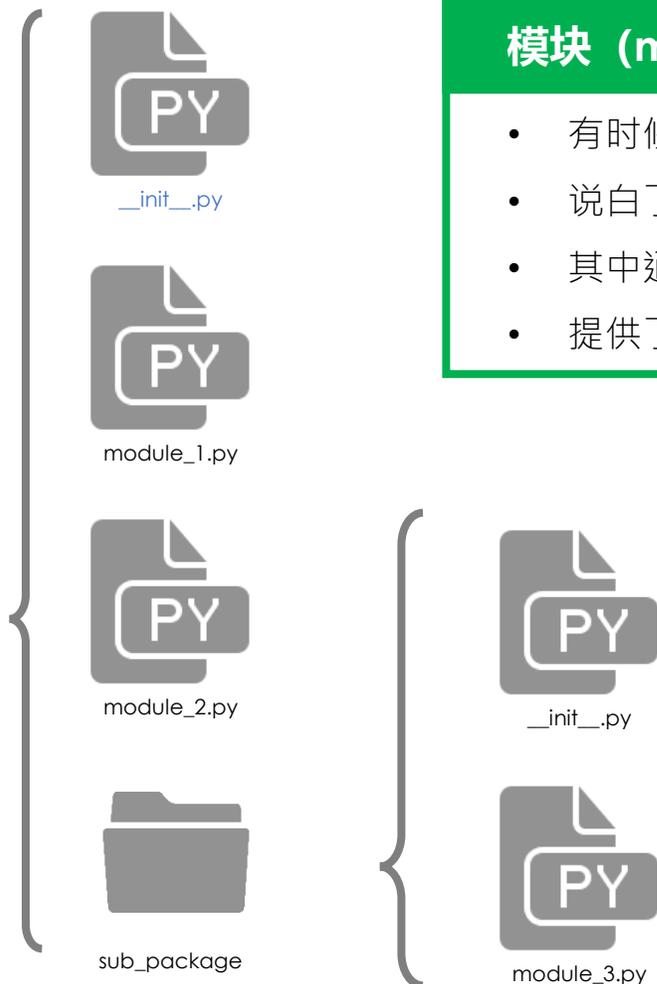
- 多个相关模块的组合，就像一个工具箱
- 把相关的模块（工具箱里的工具）组织在一起
- 本质上就是包含 `__init__.py` 文件的文件夹



package

※超纲内容

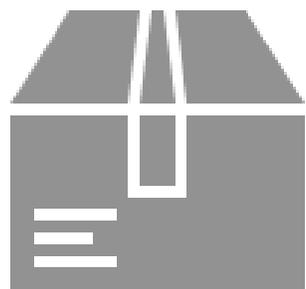
- 通过 `from package import module_1` 来调用模块
- 然后就可以通过 `module_1.func_1()` 来调用 `module_1` 中的 `func_1` 函数



模块 (module)

- 有时候也被称作 **库 (library)**
- 说白了，就是Python文件
- 其中通常包含 Python 对象定义和Python语句
- 提供了一定的功能供别人调用

包与模块的概念（太长不看版）



package

整合 ←



module_1.py

整合 ←



func_1()或者Class_1()

包 (package)

- 多个**相关**模块的组合·就像一个工具箱
- 把相关的模块(工具箱里的工具)组织在一起
- 本质上就是包含__init__.py文件的文件夹

模块 (module)

- 有时候也被称作**库 (library)**
- 说白了·就是Python文件
- 其中通常包含 Python 对象定义和Python语句
- 提供了一定的**功能**供别人调用

函数与类 (function and class)

- **函数**: 组织好的·可重复使用的·用来实现单一·或相关联功能的**代码段**
- **类**: 用户自定义对象的模板**代码段**

使用pip安装Python包

使用pip

pip 是一个现代的，通用的 Python 包管理工具。提供了对 Python 包的查找、下载、安装、卸载的功能。

在使用前，强烈建议将pip镜像源更换为清华pypi源。

```
C:\Users\Chain>pip install WeRoBot
Looking in indexes: http://pypi.douban.com/simple
Collecting WeRoBot
  Downloading http://pypi.doubanio.com/packages/c6/89/5469df61e4fd53403a9b830b6a5334d765a8f14ad773b487d4fe5003001/WeRoBot-1.12.0-py2.py3-none-any.whl (44kB)
    |#####| 51kB 544kB/s
Requirement already satisfied: requests in c:\users\chain\anaconda3\lib\site-packages (from WeRoBot) (2.22.0)
Collecting bottle (from WeRoBot)
  Downloading http://pypi.doubanio.com/packages/e9/39/2bf3a1fd963e749cde5036a184eda8c37d8af25d1297d94b8b7aeec17c4/bottle-0.12.18-py3-none-any.whl (89kB)
    |#####| 92kB 653kB/s
Requirement already satisfied: xmltodict in c:\users\chain\anaconda3\lib\site-packages (from WeRoBot) (0.12.0)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\chain\anaconda3\lib\site-packages (from requests->WeRoBot) (2.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\chain\anaconda3\lib\site-packages (from requests->WeRoBot) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in c:\users\chain\anaconda3\lib\site-packages (from requests->WeRoBot) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\chain\anaconda3\lib\site-packages (from requests->WeRoBot) (2019.11.28)
Installing collected packages: bottle, WeRoBot
Successfully installed WeRoBot-1.12.0 bottle-0.12.18
```

注意：pip命令是直接命令行下运行的，不要进入Python再运行。

例：想要安装WeRoBot包，只需要简单一行命令：

```
pip install WeRoBot
```

常用命令

- 安装 `pip install <包名>`
- 卸载 `pip uninstall <包名>`
- 查看已安装的包 `pip list`
- 查看pip命令的参数及其用法 `pip --help`

主要内容



Python

为什么是Python?

Python可以用来干什么

Anaconda

下载与安装Anaconda

使用pip安装Python包

使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook

Jupyter Lab

为什么需要管理环境?

情景一

包	版本
TensorFlow	1.4.0
NumPy	1.13.3

Run on Python 2.7

你接手了一个**前几年的**项目，其中所用到的TensorFlow还有其他一些包的版本都很老

情景二

包	版本
TensorFlow	2.2.0
NumPy	1.18.2

Run on Python 3.7

你**现在**新开了一个项目，用的是新版本的TensorFlow以及其他包

你不会希望每运作一个项目就重装一次Python或包



你需要一个“平行世界”
不同的“世界”配置着不同的**环境**，支持那个“世界”正常运转

Conda

Conda 是一个开源的软件包管理系统和环境管理系统，可以用于管理 Python 或者 R 语言版本以及依赖包。

在使用前，强烈建议将conda镜像源更换为清华Anaconda源。

环境管理命令

- 查看所有环境 `conda env list` `conda info -e`
 - 创建新环境 `conda create -n <环境名> python=<版本>`
 - 例子 `conda create -n wechatdev python=3.7`
 - 激活环境 `activate <环境名>`
 - 退出环境 `deactivate`
 - 删除环境 `conda remove -n <环境名> --all`
- 对于MacOS/Linux，需要使用
`source activate/deactivate`

包管理命令

- 安装 `conda install <包名>`
- 卸载 `conda remove <包名>`
- 查看当前环境已安装的包 `conda list`
- 查看conda命令的参数及其用法 `conda --help`

注意：conda命令是直接在命令行下运行的，不要进入Python再运行。

- 下载与安装**Anaconda**

- 在命令行下运行Python

- 使用**pip**安装**Python**包

- 包与模块的概念：万物皆对象的思想
- pip的使用

- 使用**conda**管理环境

- 环境是什么，为什么需要管理环境？
- conda的使用

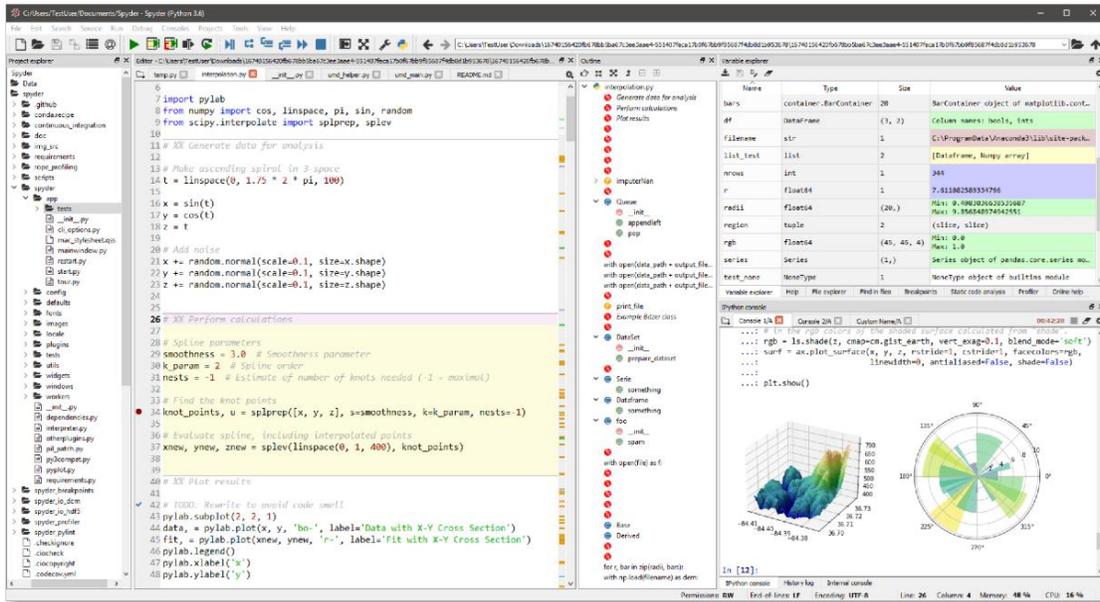


大约5分钟

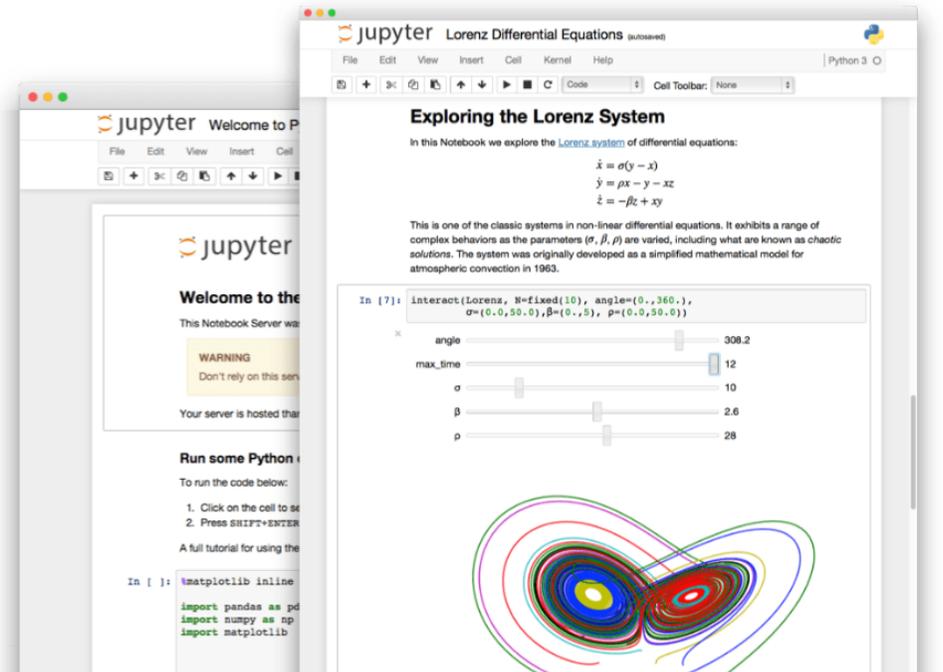
可以**连麦**上来问，也可以在**讨论区**以文字的形式问

集成开发环境 (IDE, Integrated Development Environment)

是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具，集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。

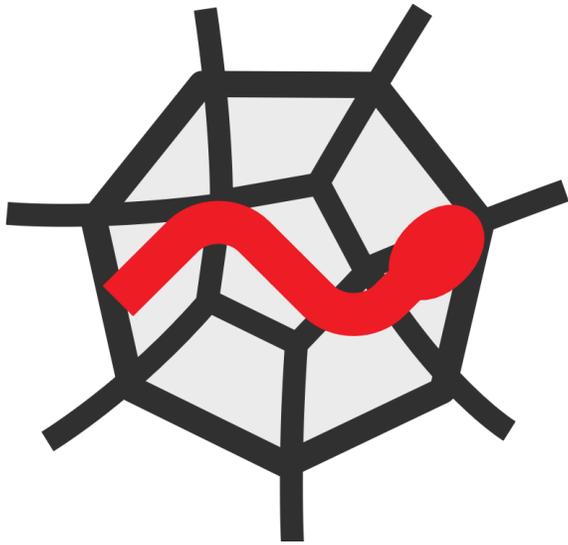


Spyder



Jupyter Notebook

主要内容



Python

为什么是Python?

Python可以用来干什么

Anaconda

下载与安装Anaconda

使用pip安装Python包

使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook

Jupyter Lab


```

1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4 This is a temporary script file.
5 """
6
7
8 a = 3
9
10 b = [1, 2, 3, 4]
11
12 ### Cell 1
13 print(">> Cell 1")
14 print(">> We are in Cell 1.")
15 ### Cell 2
16 print(">> Cell 2")
17 print(">> We are in Cell 2")
18

```

编辑器

Editor

Var	Type	Size	Value
a	int	1	3
b	list	4	[1, 2, 3, 4]

变量浏览器

Variable Explorer

```

Python 3.8.1 (default, Mar 2 2020, 13:06:26) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.12.0 -- An enhanced Interactive Python.

In [1]: a = 3

In [2]: b = [1, 2, 3, 4]

In [3]: runcell('Cell 1', 'C:/Users/Chain/Desktop/wiser 课件/example.py')
>> Cell 1
>> We are in Cell 1.

In [4]: runcell('Cell 2', 'C:/Users/Chain/Desktop/wiser 课件/example.py')
>> Cell 2
>> We are in Cell 2

In [5]:

```

IPython控制台 & 历史

IPython Console & History

Spyder常用操作

编辑器

- 运行文件 (F5)
- 运行当前单元格 (Ctrl + Enter) [通过 `#%%` 创建单元格]
- 运行当前单元格并跳到下一单元格 (Shift + Enter)
- 运行选区或者当前行 (F9)



```
12 #%% Cell 1
13  print(">> Cell 1")
14  print(">> We are in Cell 1.")
15
16 #%% Cell 2
17  print(">> Cell 2")
18  print(">> We are in Cell 2")
19
```

Ipython控制台

- 在编辑器运行代码，会显示到Ipython控制台
- 你也可以交互式地在控制台输入代码
- Ctrl + L 或者输入 `clear` 以清空控制台内容
- 输入 `del <变量名>` 清除某个变量
- 输入 `reset` 清除所有变量

```
Console 1/A
In [1]: a = 3
In [2]: b = [1, 2, 3, 4]
In [3]: runcell('Cell 1', 'C:/Users/Chain/Desktop/wiser 课件/example.py')
>> Cell 1
>> We are in Cell 1
In [4]: runcell('Cell 2', 'C:/Users/Chain/Desktop/wiser 课件/example.py')
>> Cell 2
>> We are in Cell 2
In [5]: print(a)
3
```

在编辑器中用F9运行的代码

在编辑器中用Ctrl+Enter运行的代码

在控制台交互式输入的代码

主要内容



Python

为什么是Python?
Python可以用来干什么

Anaconda

下载与安装Anaconda
使用pip安装Python包
使用conda管理环境

Spyder

Spyder使用

Jupyter

Jupyter Notebook
Jupyter Lab

Jupyter Notebook

运行Jupyter Notebook

Jupyter Notebook最大的亮点是，具有极强的交互性和展示性

在用Conda创建一个新环境后，在该环境下安装Anaconda库。

即先 `activate <环境名>`

`conda install anaconda`

这可能需要花费一段时间。



例：在icourse环境下安装好anaconda库后，开始菜单中出现了对应环境下应用的快捷方式

这样便在该环境下装好了Anaconda全家桶，其中也包含了Jupyter Notebook，正如你最初安装Anaconda一样。

既可以通过开始菜单快捷方式启动某个环境下的Jupyter Notebook，也可以通过在命令行激活环境后，输入

`jupyter notebook`

来启动

```
C:\Users\Chain>activate icourse
(icourse) C:\Users\Chain>jupyter notebook
[I 19:50:04.458 NotebookApp] The port 8888 is already in use, trying another port.
[I 19:50:04.582 NotebookApp] JupyterLab extension loaded from C:\Users\Chain\Anaconda3\envs\icourse\lib\site-packages\jupyterlab
[I 19:50:04.582 NotebookApp] JupyterLab application directory is C:\Users\Chain\Anaconda3\envs\icourse\share\jupyterlab
[I 19:50:04.606 NotebookApp] Serving notebooks from local directory: C:\Users\Chain
[I 19:50:04.606 NotebookApp] The Jupyter Notebook is running at:
[I 19:50:04.607 NotebookApp] http://localhost:8889/?token=62f0d2aacadb9319600813981246c5c234e4e6424def3210
[I 19:50:04.607 NotebookApp] or http://127.0.0.1:8889/?token=62f0d2aacadb9319600813981246c5c234e4e6424def3210
[I 19:50:04.607 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 19:50:04.673 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/Chain/AppData/Roaming/jupyter/runtime/nbsrvr-17172-open.html
Or copy and paste one of these URLs:
http://localhost:8889/?token=62f0d2aacadb9319600813981246c5c234e4e6424def3210
or http://127.0.0.1:8889/?token=62f0d2aacadb9319600813981246c5c234e4e6424def3210
```

注1：激活环境后，在行的前面会有括号提示当前所处环境。

注2：在命令行中启动应用，要保持该命令行窗口运行。否则若命令行窗口关闭，应用也将关闭



启动成功



在当前目录下
创建ipynb文件

注：启动后，会位于命令行所在的目录位置，建议学习命令行使用方法，或者了解[如何在某个目录下打开命令行](#)。

菜单

Menu

工具栏 Toolbar

笔记本区域与单元格

Notebook area and cells

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

Hello World!

- # 一级标题
- ## 二级标题
- 项目1
- 项目2
- 项目3

Jupyter Notebook 的一些概念

两个模式

- 命令模式 Command mode



- 左边沿为蓝色，处于命令模式，且选中该单元格
- 在命令模式下，键盘输入会被理解为快捷指令
- 按Enter进入所选中单元格的编辑模式

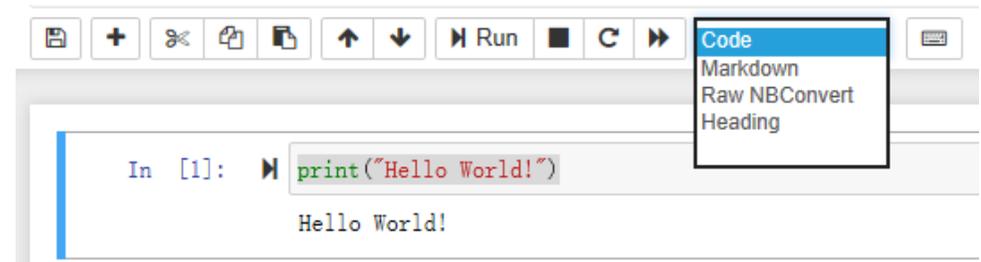
- 编辑模式 Edit mode



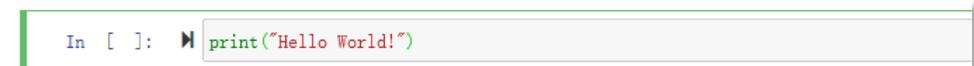
- 左边沿为绿色，处于该单元格的编辑模式
- 在编辑模式下，键盘输入用来编辑代码
- 按Esc以从编辑模式退出到命令模式

两种常用单元格类型

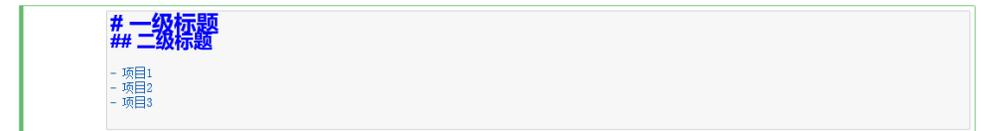
在工具栏中，可以调整单元格类型



- 代码单元格 Code，用于编辑与运行代码



- 笔记单元格 Markdown，使用[Markdown](#)展示笔记



Jupyter Notebook 常用操作

单元格操作

- 在当前单元格下方插入新单元格 (B) 
- 运行当前单元格并跳到下一格 (Shift + Enter) 

```
In [ ]: print("Hello World!")
```

```
# 一级标题  
## 二级标题  
- 项目1  
- 项目2  
- 项目3
```



```
In [2]: print("Hello World!")
```

```
Hello World!
```

```
一级标题  
二级标题  
• 项目1  
• 项目2  
• 项目3
```

- 将当前单元格切换为Code单元格 (Y)
- 将当前单元格切换为Markdown单元格 (M)

更多快捷键操作

用好快捷键才是精髓

Command Mode (press `Esc` to enable)

- `F`: find and replace
- `Ctrl-Shift-F`: open the command palette
- `Ctrl-Shift-P`: open the command palette
- `Enter`: enter edit mode
- `P`: open the command palette
- `Shift-Enter`: run cell, select below
- `Ctrl-Enter`: run selected cells
- `Alt-Enter`: run cell and insert below
- `Y`: change cell to code
- `M`: change cell to markdown
- `R`: change cell to raw
- `1`: change cell to heading 1
- `2`: change cell to heading 2
- `3`: change cell to heading 3
- `4`: change cell to heading 4
- `5`: change cell to heading 5
- `6`: change cell to heading 6
- `K`: select cell above
- `Up`: select cell above
- `Down`: select cell below
- `J`: select cell below
- `Shift-K`: extend selected cells above
- `Shift-Up`: extend selected cells above
- `Shift-Down`: extend selected cells below

Help

- User Interface Tour
- Keyboard Shortcuts
- Edit Keyboard Shortcuts

Edit Shortcuts

- `Shift-J`: extend selected cells below
- `Ctrl-A`: select all cells
- `A`: insert cell above
- `B`: insert cell below
- `X`: cut selected cells
- `C`: copy selected cells
- `Shift-V`: paste cells above
- `V`: paste cells below
- `Z`: undo cell deletion
- `D, D`: delete selected cells
- `Shift-M`: merge selected cells, or current cell with cell below if only one cell is selected
- `Ctrl-S`: Save and Checkpoint
- `S`: Save and Checkpoint
- `L`: toggle line numbers
- `O`: toggle output of selected cells
- `Shift-O`: toggle output scrolling of selected cells
- `H`: show keyboard shortcuts
- `I, I`: interrupt the kernel
- `0, 0`: restart the kernel (with dialog)
- `Esc`: close the pager
- `Q`: close the pager
- `Shift-L`: toggles line numbers in all cells, and persist the setting
- `Shift-Space`: scroll notebook up
- `Space`: scroll notebook down

主要内容



Python

为什么是Python?
Python可以用来干什么

Anaconda

下载与安装Anaconda
使用pip安装Python包
使用conda管理环境

Spyder

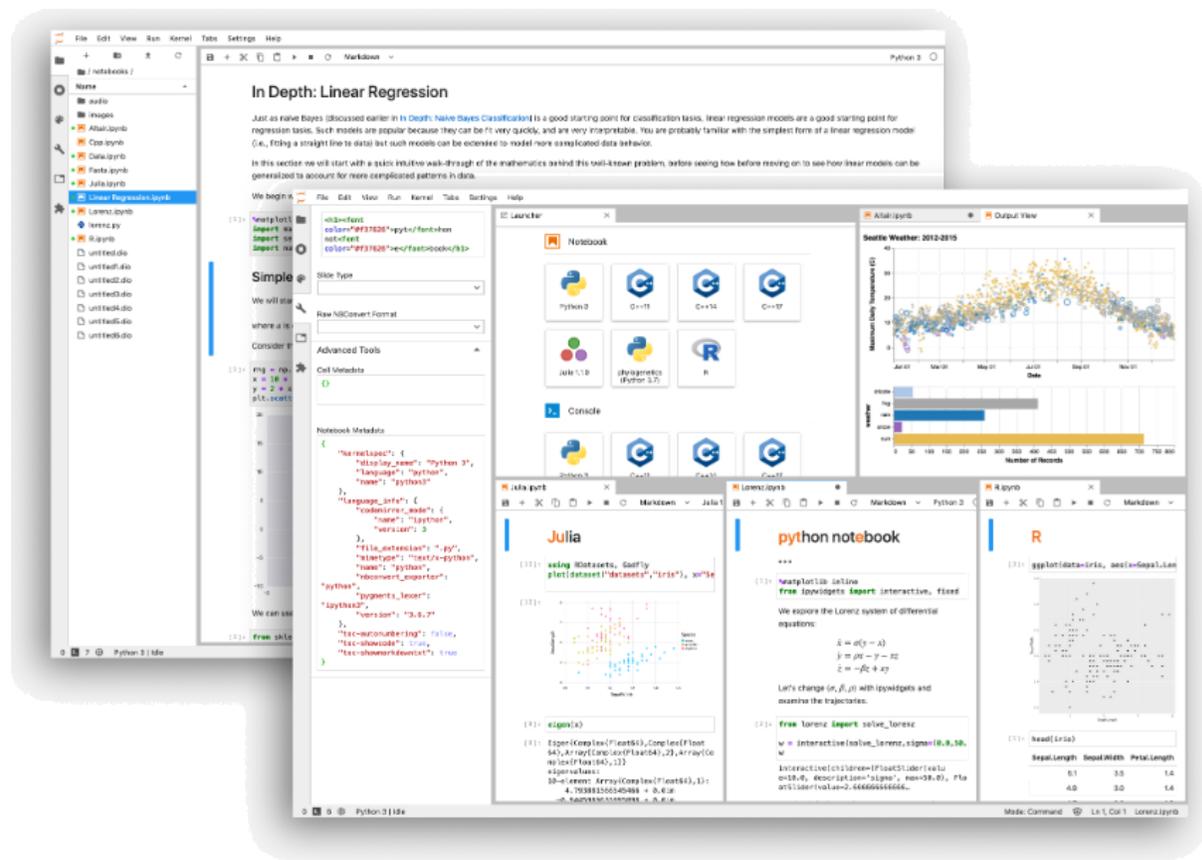
Spyder使用

Jupyter

Jupyter Notebook
Jupyter Lab

Jupyter Lab简介

Jupyter Lab可以看作是Jupyter Notebook的进化版，在外观和实用性上都很优秀



在启动和使用等各方面都基本和Jupyter Notebook一样，有兴趣可自行摸索

- **Spyder**

- 如何在某个环境下启动Spyder
- Spyder 入门

- **Jupyter Notebook / Lab**

- 如何在某个环境下启动Jupyter Notebook
- Jupyter Notebook 入门



jupyter

- *Learn Python 3 the Hard Way* - Zed A. Shaw

强烈推荐该书以及系列[教学视频](#)，对Python初学者乃至编程零基础的同学都相当友好，由浅入深。

- [Python官方文档](#)

官方文档永远是最好的教科书，有着系统而完整的学习路径，现在还有中文版，要什么自行车？

- **以及请继续关注我们的后续课程：**

- 第二讲 Python基础数据结构
- 第三讲 Python控制流与函数
- 第四讲 NumPy与Pandas
- 第五讲 Python数据可视化

6

扫码填写反馈问卷，有机会获取图书馆纪念品！



讲座签到



讲座反馈