

Chapter 1 – Python 概览

Frank Ziwei Zhang
School of Finance



上海對外經貿大學
SHANGHAI UNIVERSITY OF INTERNATIONAL BUSINESS AND ECONOMICS

Contents

Q1

Python 语言简介

Q2

Python 软件安装

Q3

Spyder 简介

Q4

金融数据获取

1.1 Python语言简介

1.1.3 Python相关

专业技能要求逐渐升高

资源性格要求逐渐降低

经纪业务

从业资格

- 广泛涉猎
- 多多益善

投资银行

CPA/司考/CFA

- 公司金融
- 投资学
- 财务会计
- 投资银行学
- 资本市场
- 公司分析
- 固定收益
- 金融工程

资产管理

CFA/CPA/司考

- 投资学
- 公司金融
- 风险管理
- 固定收益
- 金融工程
- 量化投资
- 财务会计
- python

金融科技

大厂实习/ACM
国赛/项目/CFA

- 金融学
- C++/python
- 数据库
- 数据分析
- 量化投资
- 人工智能
- 数据结构
- 计算方法

1.1.1 Python语言简介



The world's largest technical professional organization for the advancement of technology

Rank	Language	Type	Score
1	Python	🌐 🖥️ ⚙️	100.0
2	Java	🌐 📱 🖥️	95.4
3	C	📱 🖥️ ⚙️	94.7
4	C++	📱 🖥️ ⚙️	92.4
5	JavaScript	🌐	88.1
6	C#	🌐 📱 🖥️ ⚙️	82.4
7	R	🖥️	81.7
8	Go	🌐 🖥️	77.7
9	HTML	🌐	75.4
10	Swift	📱 🖥️	70.4

IEEE Spectrum 发布了 2021 年度编程语言排行榜，Python 在总榜单以及其他几个分榜单中依然牢牢占据第一名的位置。

IEEE 采用了 8 个信息源、按照 11 种指标去进行分析，这些来源包括：

Google、Twitter、GitHub、StackOverflow、Reddit、Hacker News、CareerBuilder 和 IEEE。

1.1.1 Python语言简介

起源于1989年，发行于1991年。

是一个免费、开源、跨平台、动态、面向对象的编程语言。

- **免费**：使用者不需要支付任何费用，没有版权问题。
- **开源**：将程序的源代码免费提供给开发者进行修改传播。
- **跨平台**：一次开发到处部署。

Linux、Windows、Unix (Mac OS)



吉多·范罗苏姆
Guido van Rossum
俗称“龟叔”

1.1.1 Python语言简介

“优雅”、“明确”、“简单”



Python 语言Hello World代码：

```
print("Hello World!")
```

C 语言Hello World 代码：

```
int main()  
{  
    printf("Hello, World!");  
    return 0;  
}
```

Java 语言Hello World 代码：

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

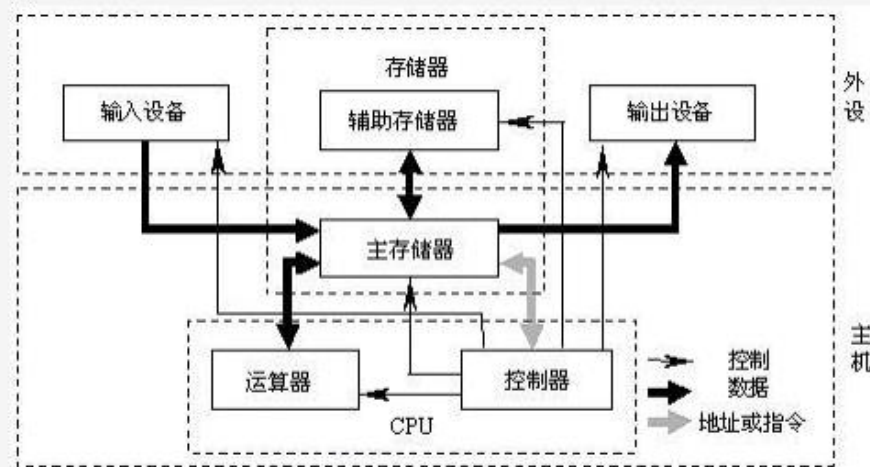

1.1.1 Python语言简介



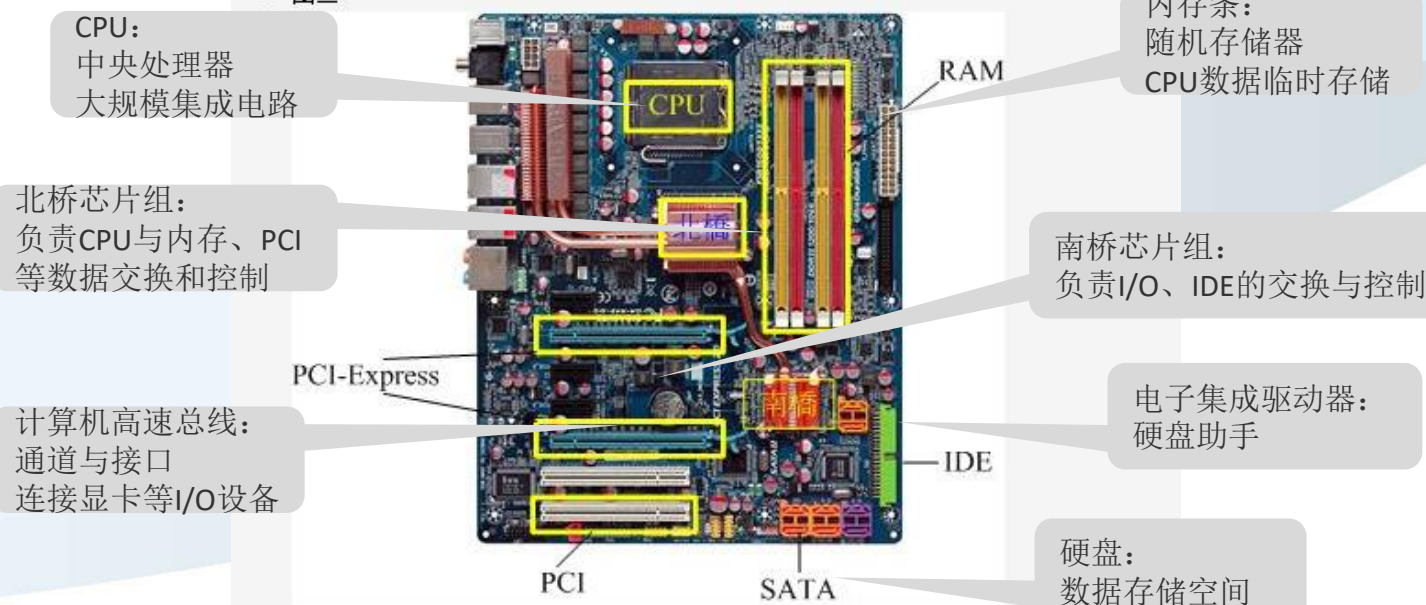
1.1.2 Python工作原理

组成的联系:

• 图一

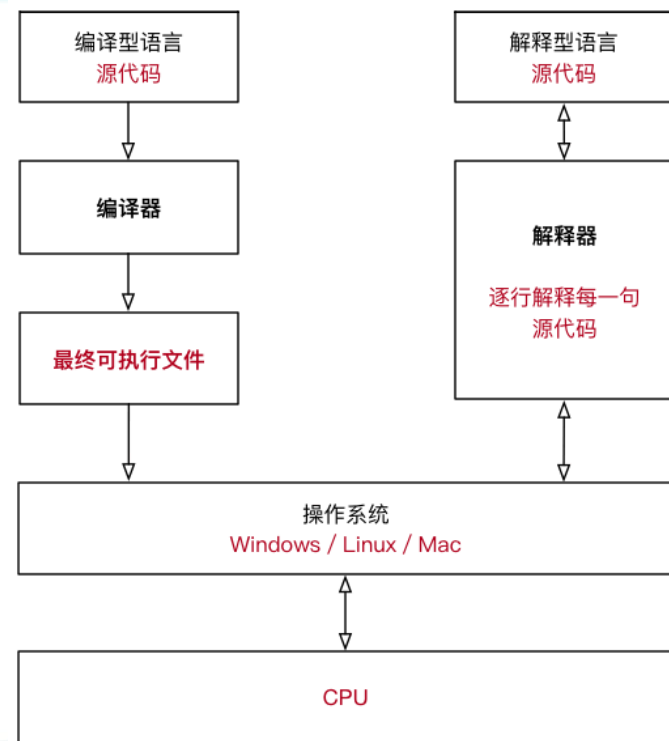
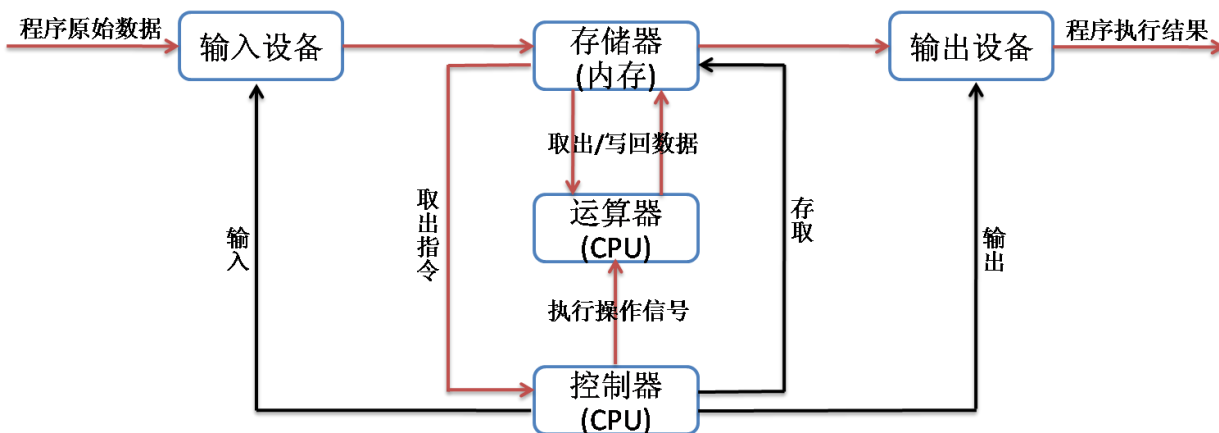


• 图二



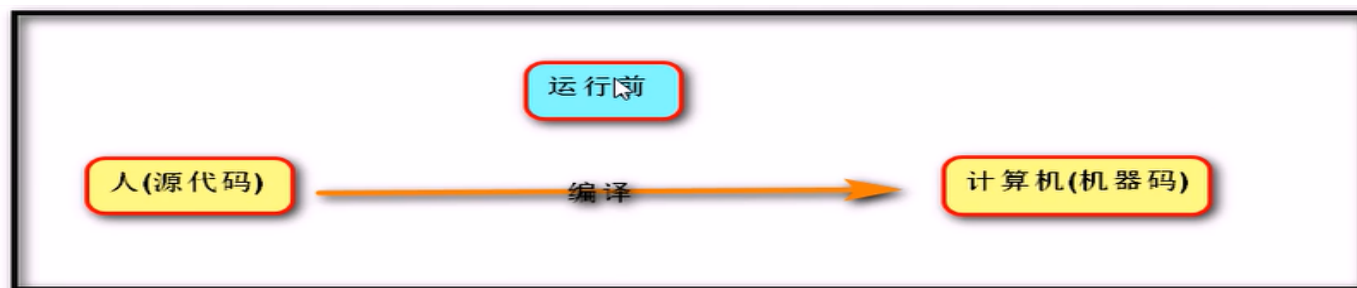
1.1.2 Python工作原理

冯·诺伊曼结构计算机

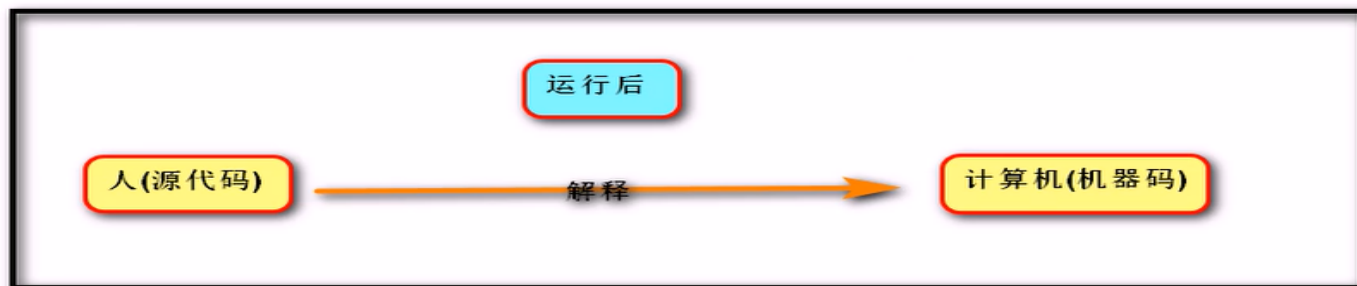


类型	原理	优点	缺点	代表
编译型语言	通过专门的编译器，将所有源代码一次性转换成特定平台（Windows、Linux 等）执行的机器码（以可执行文件的形式存在）。	编译一次后，脱离了编译器也可以运行，并且运行效率高。	可移植性差，不够灵活。	C语言、C++、汇编
解释型语言	由专门的解释器，根据需要将部分源代码临时转换成特定平台的机器码。	跨平台性好，通过不同的解释器，将相同的源代码解释成不同平台下的机器码。	一边执行一边转换，效率低。	JavaScript、PHP、MATLAB

1.1.2 Python工作原理

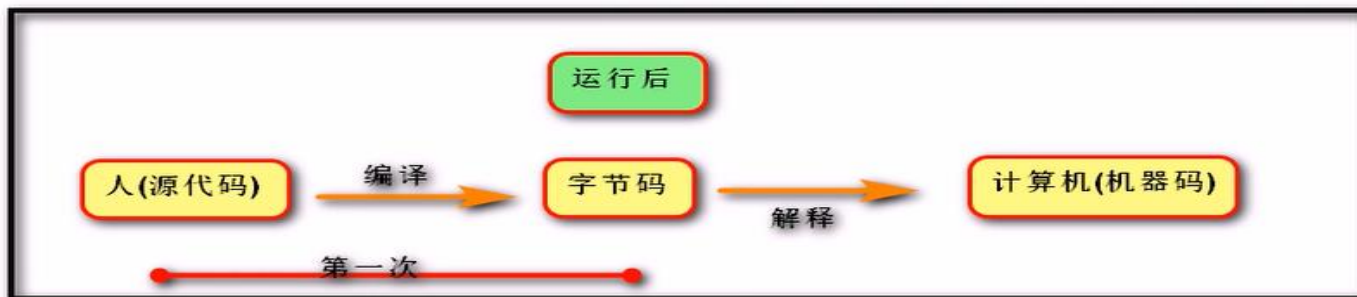


C语言



Javascript

.pyc文件是由.py文件经过编译后生成的字节码文件，其加载速度相对于之前的.py文件有所提高，而且还可以实现源码隐藏，以及一定程度上的反编译。因此，Python在运行原理上类似于Java，但又不基于虚拟机。



Java
Python

➤ 计算机软件开发主要课程:

- 1、核心编程：核心算法，面向对象，集成操作框架
- 2、高级软件开发：Linux，数据结构/计算方法，IO网络编程，并行计算，正则表达式，mySQL交互
- 3、web全栈开发：前端，后端
- 4、爬虫工程师：Redis，爬虫
- 5、人工智能：数学基础，数据分析与可视化，机器学习

➤ Python相关学习网站:

1、官方中文文档:

<https://docs.python.org/zh-cn/3/>

2、程序员教程:

<https://www.runoob.com/>

3、程序员交流平台:

<https://www.csdn.net/>

4、python中文开发者社区:

<https://www.pythontab.com/>

1.2 Python软件安装

1.2.1 Python, Anaconda, Spyder的关系

➤ 1、Python不方便

- ✓ 如果直接安装官网上下载的Python软件，它包含解释器和类似编辑器的IDLE，需要用pip来管理软件包（即管理module）。这样编程效率低下、不好调试，而且自带的python shell和带图形界面的IDLE只能一行一行的交互，也比较麻烦。
- ✓ 也就是说，为了解决python不方便直接使用的问题，有不少发行版的Python，比如WinPython、Anaconda等，这些发行版将python和许多常用包整合起来，方便我们直接使用。

➤ 2、Anaconda集成的Spyder效率高

- ✓ Anaconda是Python发行版，包含了Python解释器，集成开发环境（IDE）Spyder，还有常用的科学包（module）等。
- ✓ Anaconda利用工具/命令conda来进行package和environment的管理，并且已经包含了Python和相关的配套工具。Anaconda是一个打包的集合，里面预装好了conda、某个版本的python、众多packages、科学计算工具等等，所以也称为Python的一种发行版。
- ✓ 与上面几个概念完全不同的是，PyCharm、Jupyter和Spyder 都是目前比较主流的几种Python IDE（集成开发环境），相当于是 在一个搭建好的环境中进行程序开发，因此效率更高。

➤ 1、下载Anaconda安装包

✓ 下载地址：<https://www.anaconda.com>

➤ 2、管理员权限运行

✓ 安装的时候需要用到管理员权限，注意在安装anaconda软件的时候，右击安装软件→选择以管理员的身份运行即可。

✓ 新版本的anaconda安装非常简单，如果安装过程中出现问题，可能需要注意：

- 最好卸载本机已存在的其他python环境；
- 尽量避免安装目录中的中文、空格等生僻字符；

✓ 旧版本的anaconda安装完毕后需要添加环境变量：控制面板\系统和安全\系统\高级系统设置\环境变量\Path 中新增D:\Working\Anaconda\Scripts（示例安装路径），此目录记得更改为你安装anaconda的目录；

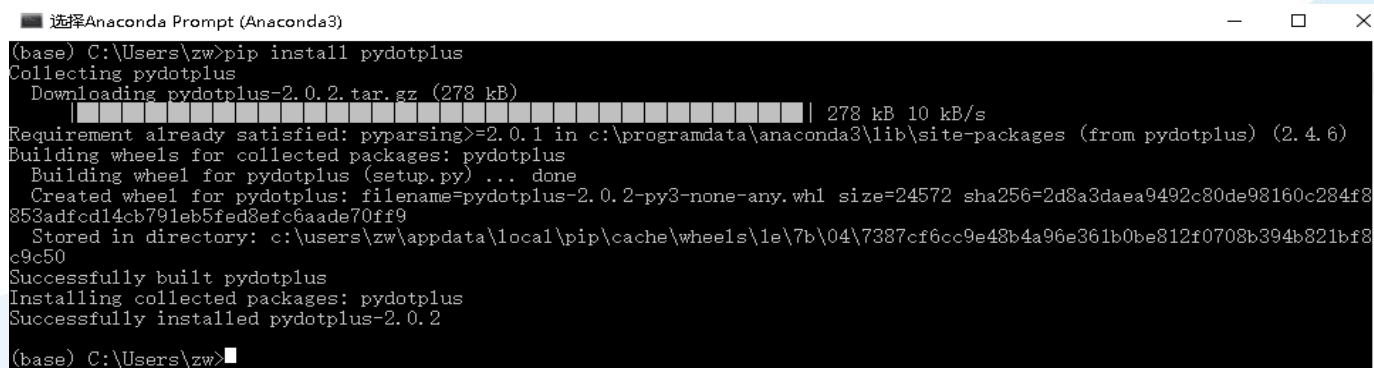
✓ anaconda集成了很多常用模块，但这些模块的版本不一定是最新的，为了避免可能发生的错误，我们可以在anaconda prompt命令中输入conda upgrade --all 先把所有工具包进行升级；

1.2.3 pip安装工具包

python能够这么流行，一个重要的原因是第三方库/工具包/模块，这些库提供了丰富的功能，如PyPI、github等开源代码库上提供了有非常多的python的第三方库，不仅提供了很多实用的功能，也是学习python项目编程的好材料。

- 1、点击开始—>anaconda—>anaconda prompt，右键管理员打开Anaconda Prompt命令窗口；或者需要用windows系统中的cmd命令安装；
- 2、pip install “包名”，enter后，进入下载安装界面；
- 3、安装成功会是Successfully状态；
- 4、可以在Spyder中进行import测试，看是否安装成功：

import pydotplus as pdp



```
选择Anaconda Prompt (Anaconda3)
(base) C:\Users\zw>pip install pydotplus
Collecting pydotplus
  Downloading pydotplus-2.0.2.tar.gz (278 kB)
    |#####| 278 kB 10 kB/s
Requirement already satisfied: pyparsing>=2.0.1 in c:\programdata\anaconda3\lib\site-packages (from pydotplus) (2.4.6)
Building wheels for collected packages: pydotplus
  Building wheel for pydotplus (setup.py) ... done
  Created wheel for pydotplus: filename=pydotplus-2.0.2-py3-none-any.whl size=24572 sha256=2d8a3daea9492c80de98160c284f8853adfc14cb791eb5fed8efc6aade70ff9
  Stored in directory: c:\users\zw\appdata\local\pip\cache\wheels\1e\7b\04\7387cf6cc9e48b4a96e361b0be812f0708b394b821bf8c9c50
Successfully built pydotplus
Installing collected packages: pydotplus
Successfully installed pydotplus-2.0.2
(base) C:\Users\zw>
```


1.2.3 pip安装一切

➤ 1、pip是什么？

✓ pip实际上是一个通用的 Python 包管理工具，提供了对Python 包的查找、下载、安装、卸载的功能（<https://pypi.org/project/pip>）。
[anaconda](#)已经集成了这个包，因此我们不用再独立安装。

✓ 两种途径进入：

- 1、点击开始 —> anaconda —> anaconda prompt，右键管理员打开 Anaconda Prompt 命令窗口；
- 2、或者需要用 windows 系统中的 cmd 命令安装；

✓ 功能查看：

- 在命令行中键入：pip+Enter，就会出现如下使用说明，也就是pip能够接受的命令：

```
(base) C:\Windows\system32>pip
Usage:
  pip <command> [options]

Commands:
  install           Install packages.
  download          Download packages.
  uninstall         Uninstall packages.
  freeze            Output installed packages in requirements format.
  list              List installed packages.
  show              Show information about installed packages.
  check             Verify installed packages have compatible dependencies.
  config            Manage local and global configuration.
  search            Search PyPI for packages.
  cache             Inspect and manage pip's wheel cache.
  index             Inspect information available from package indexes.
  wheel             Build wheels from your requirements.
  hash              Compute hashes of package archives.
  completion        A helper command used for command completion.
  debug            Show information useful for debugging.
  help              Show help for commands.

General Options:
  -h, --help            Show help.
  --isolated            Run pip in an isolated mode, ignoring environment variables and user configuration.
  -v, --verbose         Give more output. Option is additive, and can be used up to 3 times.
  -V, --version         Show version and exit.
  -q, --quiet           Give less output. Option is additive, and can be used up to 3 times (corresponding to
                        WARNING, ERROR, and CRITICAL logging levels).
  --log <path>         Path to a verbose appending log.
  --no-input            Disable prompting for input.
  --proxy <proxy>       Specify a proxy in the form [user:passwd@]proxy.server:port.
  --retries <retries>   Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>       Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
                        (a)bort.
  --trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any
                        HTTPS.
  --cert <path>         Path to PEM-encoded CA certificate bundle. If provided, overrides the default. See 'SSL
                        Certificate Verification' in pip documentation for more information.
  --client-cert <path> Path to SSL client certificate, a single file containing the private key and the
                        certificate in PEM format.
  --cache-dir <dir>     Store the cache data in <dir>.
  --no-cache-dir        Disable the cache.
  --disable-pip-version-check Don't periodically check PyPI to determine whether a new version of pip is available for
                        download. Implied with --no-index.
  --no-color            Suppress colored output.
  --no-python-version-warning Silence deprecation warnings for upcoming unsupported Python versions.
  --use-feature <feature> Enable new functionality, that may be backward incompatible.
  --use-deprecated <feature> Enable deprecated functionality, that will be removed in the future.
```

➤ 2、常用命令：

✓ 安装install

- 命令行直接键入**pip install** 包名，当然也可以指定安装的版本：
- **pip install SomePackage** # 最新版本
- **pip install SomePackage==1.0.4** # 指定版本

✓ 卸载uninstall:

- **pip uninstall** 包名

✓ 升级包

- **pip install -U** 包名
- **pip install** 包名 **--upgrade**

✓ 展示安装包

- **pip list**

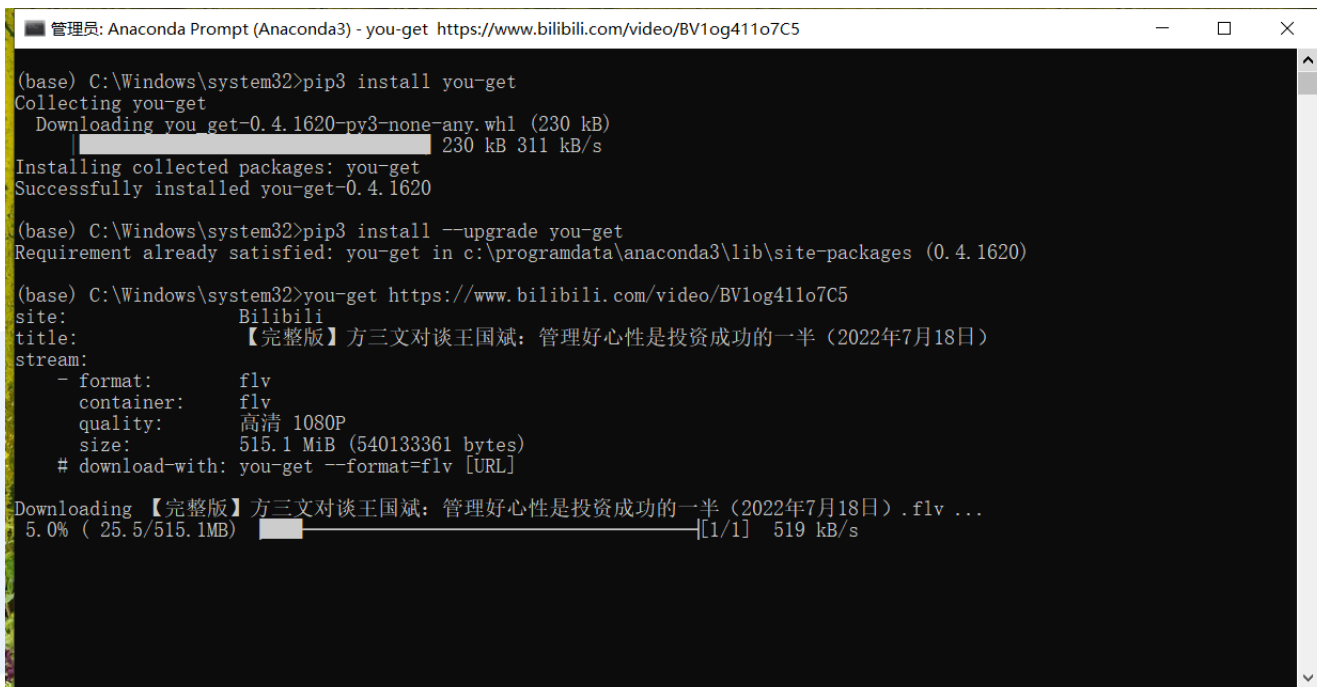
✓ 获取帮助

- **pip --help**

1.2.3 pip安装一切

➤ 3、you-get

- ✓ you-get是github上的一个开源库(<https://github.com/soimort/you-get>), 使用you-get可以轻松下载网页上的视频内容, 目前you-get所支持的网站包含国内外几十个网站(youtube、twitter、腾讯、爱奇艺、优酷、bilibili等)。
- ✓ 安装you-get: `pip install you-get`
- ✓ 主要参数:
 - -o: 后面跟下载目录
 - -i: 查看资源清晰度, 一般模式最高清晰度下载
 - -c: 后面跟cookies, 即下载会员资源需要会员的信息



```
管理员: Anaconda Prompt (Anaconda3) - you-get https://www.bilibili.com/video/BV1og411o7C5

(base) C:\Windows\system32>pip3 install you-get
Collecting you-get
  Downloading you_get-0.4.1620-py3-none-any.whl (230 kB)
    |#####| 230 kB 311 kB/s
Installing collected packages: you-get
Successfully installed you-get-0.4.1620

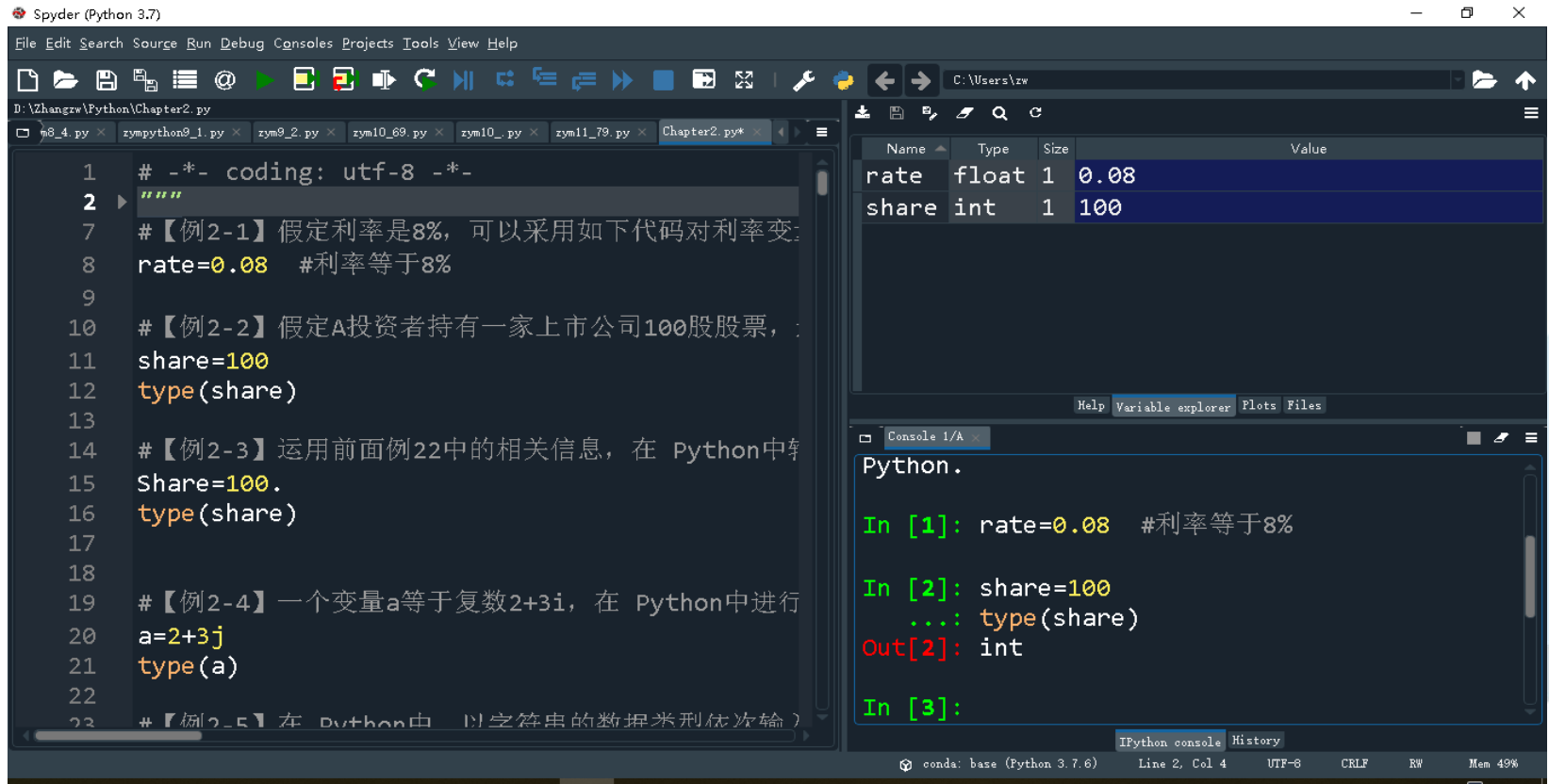
(base) C:\Windows\system32>pip3 install --upgrade you-get
Requirement already satisfied: you-get in c:\programdata\anaconda3\lib\site-packages (0.4.1620)

(base) C:\Windows\system32>you-get https://www.bilibili.com/video/BV1og411o7C5
site: Bilibili
title: 【完整版】方三文对谈王国斌：管理好心性是投资成功的一半（2022年7月18日）
stream:
  - format: flv
    container: flv
    quality: 高清 1080P
    size: 515.1 MiB (540133361 bytes)
    # download-with: you-get --format=flv [URL]

Downloading 【完整版】方三文对谈王国斌：管理好心性是投资成功的一半（2022年7月18日）.flv ...
5.0% ( 25.5/515.1MB) |#####| [1/1] 519 kB/s
```

1.3 Spyder简介

1.3.1 Console简介



Spyder是使用Python编程语言进行科学计算的集成开发环境。它有一个Editor（编辑器）用于编写代码，Console（控制台）可以评估代码并且在任何时候都可以看到运行结果，Variable Explorer（变量管理器）可以查看代码中定义的变量，还有其他的一些设备可以帮你高效地开发代码。

1、IPython Console

IPython Console比标准的Python Console的功能更多，在执行代码的过程，Python会逐行读取Editor中的代码，并且它会自动忽略注释（即以#符号开头的行）。

通过使用IPython Console，我们可以知道Spyder中Editor窗口中的Python代码是如何执行的。当Python遇到从最左边的列开始写的命令（除了def...和其他一些关键字），它将马上执行这些命令；当遇到def关键字，Python知道在这里定义了一个函数。在def之后的所有缩进行都是函数体。

1.3.1 Console

【例题1.1】定义一个函数，执行著名的python程序“hello world”；然后执行这段代码，从菜单中选择Run > Run（或者按F5键），可以定义这个函数。

```
def hello():  
    """Print "Hello World" and return None"""  
    print("Hello World")  
# main program starts here
```

你一旦执行了hello.py文件，函数对象hello就被定义了，并且IPython console也知道了它的存在。因此，我们可以用如下方式在console中调用函数：

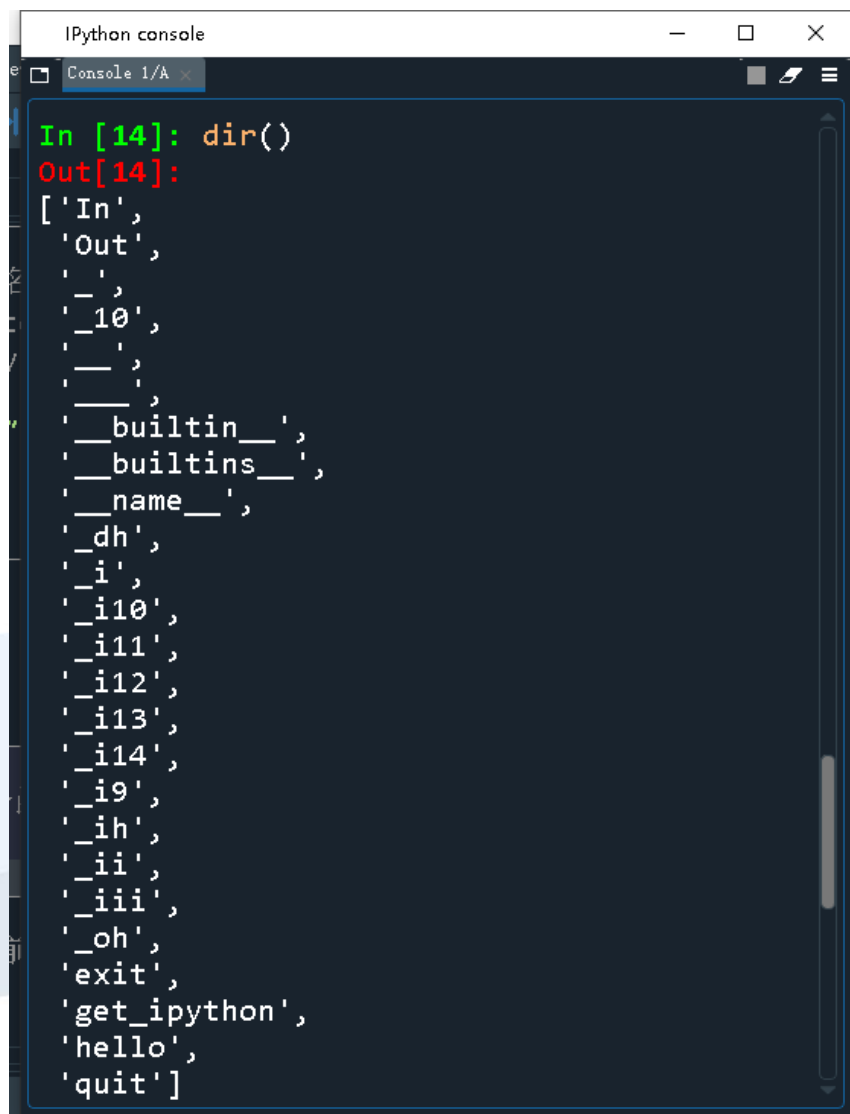
【例题1.2】执行hello()函数，在console中写入hello()，然后按下Enter键：

```
hello()  
Hello World
```

1.3.1 Console

【例题1.3】Python提供了一个函数，它可以显示console中当前控制台中有已知的对象。它就是`dir()`：当你在console中写入`dir()`，将得到一个对象列表，你能看见`hello`在列表中：

`dir()`

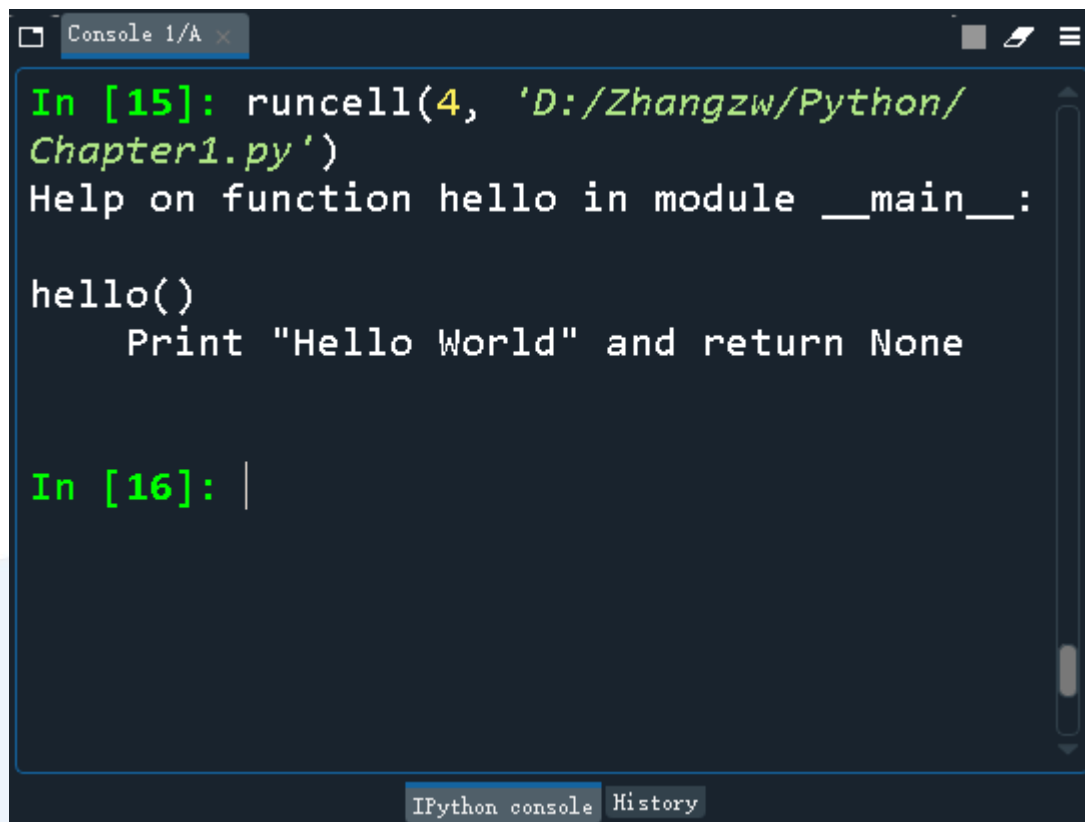
A screenshot of an IPython console window. The window has a title bar that says "IPython console" and standard window controls (minimize, maximize, close). Below the title bar, there's a tab labeled "Console 1/A". The main area of the window is a dark-themed text editor showing the execution of the `dir()` function. The prompt "In [14]: " is followed by `dir()`. The output "Out[14]: " is followed by a list of strings representing the objects in the current namespace. The list includes: `['In', 'Out', '_', '_10', '_', '__builtin__', '__builtins__', '__name__', '_dh', '_i', '_i10', '_i11', '_i12', '_i13', '_i14', '_i9', '_ih', '_ii', '_iii', '_oh', 'exit', 'get_ipython', 'hello', 'quit']`. The word "hello" is visible in the list, confirming its presence in the console's namespace.

```
IPython console
Console 1/A
In [14]: dir()
Out[14]:
['In',
 'Out',
 '_',
 '_10',
 '_',
 '__builtin__',
 '__builtins__',
 '__name__',
 '_dh',
 '_i',
 '_i10',
 '_i11',
 '_i12',
 '_i13',
 '_i14',
 '_i9',
 '_ih',
 '_ii',
 '_iii',
 '_oh',
 'exit',
 'get_ipython',
 'hello',
 'quit']
```

1.3.1 Console

【例题1.4】一旦一个对象在当前控制台中可见（例如本例中的hello），我们可以使用help函数了解这个对象：在console提示区写入help(hello)，你应该可以看到如下的输出：

help(hello)



```
Console 1/A x
In [15]: runcell(4, 'D:/Zhangzw/Python/Chapter1.py')
Help on function hello in module __main__:

hello()
    Print "Hello World" and return None

In [16]: |
```

IPython console History

1.3.1 Console

【例题1.5】控制台（在任何给定时间内在console中定义的对象集合）在IPython中可以使用%reset命令清除。输入%reset然后按下enter键，用y确认：
%reset

```
In [8]: runcell(5, 'D:/Zhangzw/Python/  
Chapter1.py')
```

```
WARNING: This is not valid Python code. If  
you want to use IPython magics, flexible  
indentation, and prompt removal, please  
save this file with the .ipy extension.  
This will be an error in a future version  
of Spyder.
```

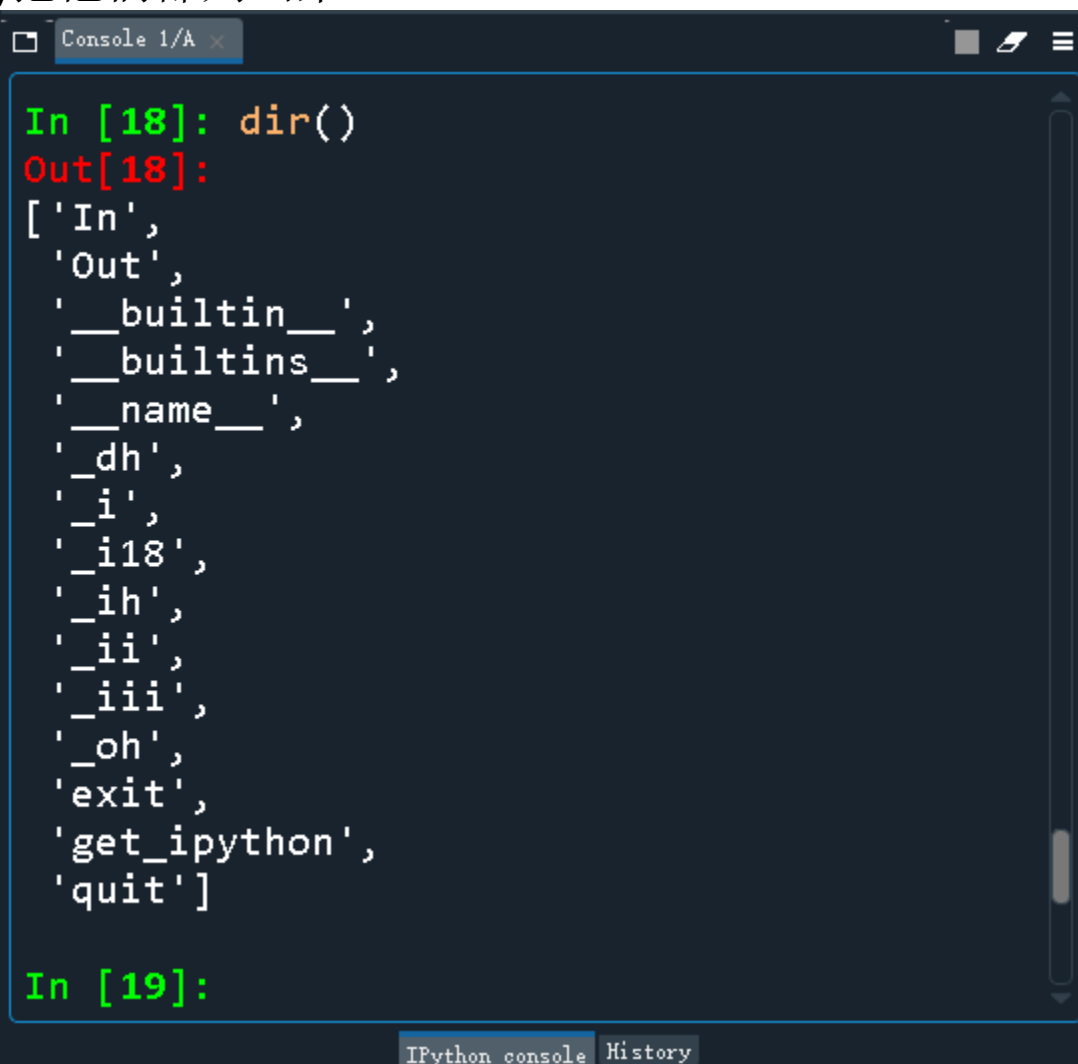
```
Once deleted, variables cannot be  
recovered. Proceed (y/[n])?
```

IPython console History

1.3.1 Console

【例题1.6】使用`%reset`命令之后，在当前会话中只有少部分对象在命名空间中。我们可以使用`dir()`把他们都列出来：

`dir()`



```
Console 1/A x
```

```
In [18]: dir()
Out[18]:
['In',
 'Out',
 '__builtin__',
 '__builtins__',
 '__name__',
 '_dh',
 '_i',
 '_i18',
 '_ih',
 '_ii',
 '_iii',
 '_oh',
 'exit',
 'get_ipython',
 'quit']

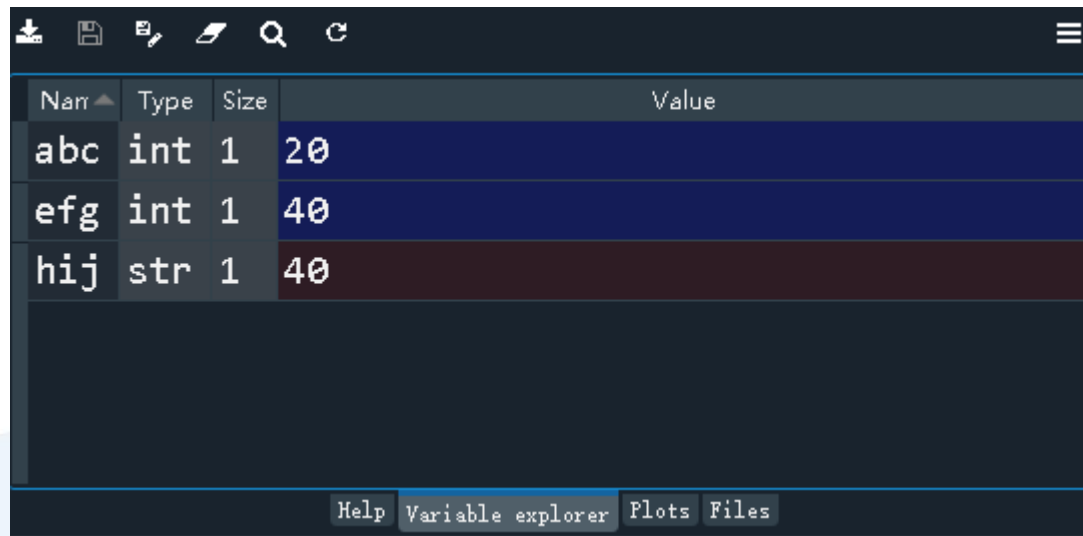
In [19]:
```

IPython console History

1.3.2 Variable Explorer

2、Variable Explorer中的对象

和其他的Python开发环境相比，它最大的优点就是模仿MATLAB的“工作空间”的功能，可以很方便地观察和修改数组的值。这是一个非常有用的版块，会显示当前执行代码中的变量（只显示变量的数值和数组类型，不会显示函数或者类对象）。



The screenshot shows the Variable Explorer window with a table of variables. The table has four columns: Name, Type, Size, and Value. The variables listed are abc (int, size 1, value 20), efg (int, size 1, value 40), and hij (str, size 1, value 40). The window has a toolbar at the top and a tabbed interface at the bottom with tabs for Help, Variable explorer, Plots, and Files.

Name	Type	Size	Value
abc	int	1	20
efg	int	1	40
hij	str	1	40

1.3.3 Editor

Tab键自动补全命令、函数名、变量名、Console（Python和IPython）和Editor中的方法名。这个特点也很有用，将来你可能会频繁地使用。

【例题1.7】定义一个变量：`mylongvariablename = 42`；假设我们需要写代码计算`mylongvariablename + 100`，我们只需要输入 `my`然后按Tab键。如果这个变量名是唯一的，在光标位置会自动补齐变量名，然后再继续输入 `+ 100`。如果以字母`my`开头的变量名不是唯一的，按下Tab键之后会列出所有以`my`开头的变量名以供选择，之后可以使用上下键选择，也可以输入更多的字母匹配。

Ctrl+Enter执行当前cell

Shift+Enter执行当前cell，并将光标移到下一个cell

Ctrl + +（Cmd + +在MacOS），将增大Editor窗口的字体

Ctrl + s（Cmd + s在MacOS上）保存当前Editor窗的文件；在IPython console窗将会把当前IPython会话以HTML文件保存，包括任何显示在命令行的参数。这样可以快速记录在会话中完成了什么。

Ctrl + i（Cmd + i在MacOS上），当光标在一个对象上时使用，将在help窗口显示这个对象的文档。


Ctrl +1，将Spyder中当前行作为注释行的快捷键（双次取消注释），类似于java等语言。


Ctrl+4，将Spyder中选中段落作为注释的快捷键


1.3.4 Debug

IPython console处于激活状态，通过Debug > Debug菜单选项或者Ctrl + F5则开启了IPython调试器。

之后，Editor窗口将会高亮将要执行的代码行，Variable Explorer窗口将显示当前执行代码中的变量。（它只显示变量的数值和数组类型，不会显示函数或者类对象）。

进入调试模式之后，使用调试工具栏中的Step按钮，你可以逐行执行代码，或者使用快捷方式Ctrl + F10；

你可以查看某一个具体的函数是如何执行的，通过Step into按钮，或者使用快捷方式Ctrl + F11。

从某一函数中跳出并执行下一行程序则使用Step return按钮，或者使用快捷方式Ctrl + F12。

1.3.6 Spyder更新

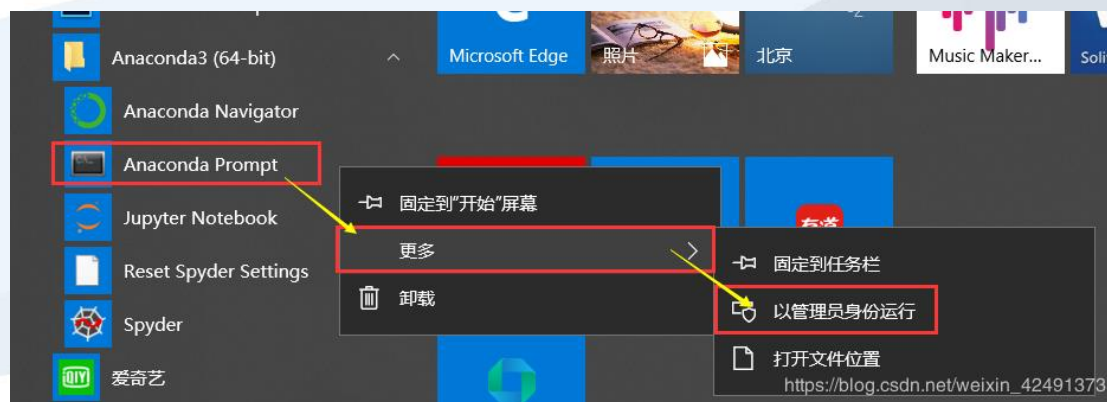
关于Spyder安装与更新，Python也给出了建议：

“如果您是Python或科学Python生态系统的新手，我们强烈建议您安装并使用Anaconda。它在一个简单易用的环境中提供了Spyder及其所有依赖项，以及最重要的Python科学库（即Numpy、Pandas、Matplotlib、IPython等）。

我们也支持pip，但请注意pip安装是为熟悉所有Spyder依赖项的高级用户而设计的。因此，您遇到的所有安装问题都希望由您解决，所以请不要向我们寻求帮助。

如果通过关闭Spyder并在系统终端（Windows上的Anaconda提示符、Linux上的xterm或macOS上的terminal.app）中运行以下命令使用Anaconda，则可以轻松安装此版本：

```
conda install spyder=4.0.1
```



1.4 金融数据获取

1.4.1 Wind数据

【例题 1.8】通过在参数表中增加 `usedf=True`，WindPy 支持直接输出 DataFrame 格式。此时，WindPy 函数返回错误码和提取的数据。其中数据的格式为 DataFrame。以下以 WSD 为例，展示如何使用 WindPy 函数以 DataFrame 格式提取数据：

```
from WindPy import w
w.start()
# 命令如何写可以用命令生成器来辅助完成
# 定义打印输出函数，用来展示数据使用
def printpy(outdata):
    if outdata.ErrorCode!=0:
        print('error code:'+str(outdata.ErrorCode)+'\n')
        return()
    for i in range(0,len(outdata.Data[0])):
        strTemp=""
        if len(outdata.Times)>1:
            strTemp=str(outdata.Times[i])+' '
        for k in range(0, len(outdata.Fields)):
            strTemp=strTemp+str(outdata.Data[k][i])+' '
    printpy(wsetdata)
```

1.4.2 Tushare数据

<http://tushare.org/index.html>

Tushare是一个免费、开源的python财经数据接口包。主要实现对股票等金融数据从数据采集、清洗加工 到 数据存储的过程，能够为金融分析人员提供快速、整洁、和多样的便于分析的数据，为他们在数据获取方面极大地减轻工作量，使他们更加专注于策略和模型的研究与实现上。考虑到Python pandas包在金融量化分析中体现出的优势，Tushare返回的绝大部分的数据格式都是pandas DataFrame类型，非常便于用pandas/NumPy/Matplotlib进行数据分析和可视化。

1.4.2 Tushare数据

【例1-9】仅以Tushare作为例子来演示如何通过API接口在 Python中导入浦发银行（600000.SH）2018年第一季度每个交易日的开盘点位、最高点位、最低点位以及收盘点位，具体的过程如下：

首先，下载安装：

```
pip install tushare
```

```
pip install pandas #下面两个是要用到的工具包，安装过的可以忽略
```

```
pip install lxml
```

如果是使用以前版本的，可以通过以下命令来进行升级：

```
pip install tushare --upgrade
```

其次，Tushare是一个免费开源的金融数据平台，新版本Tushare Pro更加稳定、流畅，而且将数据扩大到了股票、基金、期货、债券、外汇、行业大数据等区块链的数据。推荐大家注册一个账户，下面就是注册链接：
<https://tushare.pro/register?reg=397881>。

1.4.2 Tushare数据

再次，获取token密钥：在刚才注册过网站的右上角点击个人主页，在接口TOKEN中我们就可以复制到token：



最后，就是设置token了
`import tushare as ts`

#方式一

`ts.set_token('你刚才复制的token填在这里')`

#这种方式设置token我们会把token保存到本地，所以我们在使用的时候只需设置一次，失效之后，我们可以替换为新的token

#方式二

`pro = ts.pro_api()`

`pro = ts.pro_api('你刚才复制的token填在这里')`

这种在初始化接口的时候设置token

1.4.2 Tushare数据

```
PFYH_excel2 = pro.daily(ts_code='600000.SH', start_date='20180101',  
end_date='20180331')
```

PFYH_excel2 - DataFrame

Index	ts_code	trade_date	open	high	low	close
0	600000.SH	20180330	11.63	11.69	11.61	11.65
1	600000.SH	20180329	11.57	11.71	11.45	11.62
2	600000.SH	20180328	11.53	11.75	11.52	11.57
3	600000.SH	20180327	11.7	11.77	11.56	11.62
4	600000.SH	20180326	11.71	11.75	11.5	11.61
5	600000.SH	20180323	12	12.06	11.64	11.71
6	600000.SH	20180322	12.35	12.35	12.25	12.25
7	600000.SH	20180321	12.34	12.4	12.28	12.29
8	600000.SH	20180320	12.28	12.39	12.25	12.32
9	600000.SH	20180319	12.31	12.33	12.26	12.31

1.4.3 AKShare数据

[AKShare 项目概览 — AKShare 1.12.87 文档 \(akfamily.xyz\)](#)

AKShare 是基于 Python 的财经数据接口库, 目的是实现对股票、期货、期权、基金、外汇、债券、指数、加密货币等金融产品的基本面数据、实时和历史行情数据、衍生数据从数据采集、数据清洗到数据落地的一套工具, 主要用于学术研究目的.

AKShare 的特点是获取的是相对权威的财经数据网站公布的原始数据, 通过利用原始数据进行各数据源之间的交叉验证, 进而再加工, 从而得出科学的结论.

通用安装

```
pip install akshare --upgrade
```

注意: 程序运行时, 文件名、文件夹名不能是: akshare

国内安装-Python

```
pip install akshare -i http://mirrors.aliyun.com/pypi/simple/ --trusted-host=mirrors.aliyun.com --upgrade
```

升级 AKShare

P.S. 由于目前版本更新迭代频繁, 请在使用 AKShare 前先升级, 命令如下所示

```
pip install akshare --upgrade -i https://pypi.org/simple
```

1.4.3 AKShare数据

0、请首先定义要下载的股票数据的起止日期，格式如下请不要改动

```
startDate = '20210603' #起始日期  
endDate = '20210620' #终止日期
```

1、从本地文件夹读入股票代码数据

```
import numpy as np  
import pandas as pd  
Code_name = pd.read_csv('D:\Zhangzw\Python\Python金融数据分析  
\RawData\第1章\FundManagers.csv',encoding =  
'gb2312',header=0,index_col=0)
```

2、链接tushare数据库，读入每日要更新的股票数据

#以Tushare作为例子来演示如何通过API接口在 Python中导入沪深300指数
2018年每个交易日的开盘点位、最高点位、最低点位以及收盘点位，具体的
代码如下：

```
import akshare as ak
```

1.4.3 AKShare数据

```
Num = 0
for i in Code_name['通用代码']:
    Num=Num+1
    Price_new = ak.stock_zh_a_hist(symbol=i[0:6], start_date=startDate,
end_date=endDate, adjust="hfq")
    Price_new['stockCode'] = i[0:6]
    Price_new = Price_new.set_index('stockCode') #列转化为行索引
    #df = pro.daily_basic(ts_code=i, start_date=startDate,
end_date=endDate, fields='ts_code,trade_date,turnover_rate')
    if Num==1:
        Price=Price_new
    else:
        Price=pd.concat([Price,Price_new],axis=0)
    print('今日更新数据共计', len(Code_name), '个， 目前更新至第',Num, '个
...')

### 3、存入本地股票数据库
Price.to_csv('D:\Zhangzw\Python\Python金融数据分析
\Output\Price_FundManagers.csv')
print('数据更新完毕，请打开文件夹 D:\Zhangzw\Python\Python金融数据分
析\Output 查看！！！')
```

1.4.4 BaoStock数据

证券宝www.baostock.com是一个免费、开源的证券数据平台（无需注册）。

- 提供大量准确、完整的证券历史行情数据、上市公司财务数据等。
- 通过python API获取证券数据信息，满足量化交易投资者、数量金融爱好者、计量经济从业者数据需求。
- 返回的数据格式：
 - pandas DataFrame类型，以便于用pandas/NumPy/Matplotlib进行数据分析和可视化。
 - 同时支持通过BaoStock的数据存储功能，将数据全部保存到本地后进行分析。
- 下载安装：
`pip install baostock`
- 使用国内源安装：
`pip install baostock -i https://pypi.tuna.tsinghua.edu.cn/simple/ --trusted-host pypi.tuna.tsinghua.edu.cn`

1.4.4 BaoStock数据

0、请首先定义要下载的股票数据的起止日期，格式如下请不要改动

```
Start_date = '2021-2-3' #起始日期
```

```
End_date = '2021-2-4' #终止日期
```

1、从本地文件夹读入股票代码数据

```
import numpy as np
```

```
import pandas as pd
```

```
Code_name = pd.read_csv('D:\Zhangzw\Python\Python金融数据分析\RawData\第1章  
\FundManagers.csv',encoding = 'gb2312',header=0,index_col=0)
```

2、链接baostock数据库，读入每日要更新的股票数据

```
import baostock as bs
```

```
#### 登陆系统 ####
```

```
lg = bs.login()
```

1.4.4 BaoStock数据

```
Num = 0
for i in Code_name['Baostock代码']:
    Num=Num+1
    ##### 获取历史K线数据 #####
    rs = bs.query_history_k_data_plus(i,
        "code,date,preclose,open,high,low,close,volume,amount,pctChg,turn",
        start_date=Start_date, end_date=End_date,
        frequency="d", adjustflag="3") #frequency="d"取日k线, adjustflag="3"默认不复
    权, 1: 后复权; 2: 前复权
    ##### 打印结果集 #####
    data_list = []
    while (rs.error_code == '0') & rs.next():
        # 获取一条记录, 将记录合并在一起
        data_list.append(rs.get_row_data())
    Price_new = pd.DataFrame(data_list, columns=rs.fields)
    ##### 拼接成一个文件 #####
    if Num==1:
        Price=Price_new
    else:
        Price=pd.concat([Price,Price_new],axis=0)
    print('今日更新数据共计', len(Code_name), '个, 目前更新至第',Num, '个...')
```

1.4.4 BaoStock数据

登出系统

```
bs.logout()
```

#%% 3、存入本地股票数据库

```
Price.to_csv('D:\Zhangzw\Stock\TrueTrading ETF\RawData\Price ETFs_New_Baostock.csv')
```

```
print('数据更新完毕，请打开文件夹
```

```
D:\Zhangzw\Stock\TrueTrading ETF\RawData\Price ETFs_New_Baostock.csv 查看  
!!!')
```


**Your appreciation makes me a miracle.
Thank you!**

**Frank Ziwei Zhang
18117228563
frank8027@163.com**



上海對外經貿大學
SHANGHAI UNIVERSITY OF INTERNATIONAL BUSINESS AND ECONOMICS