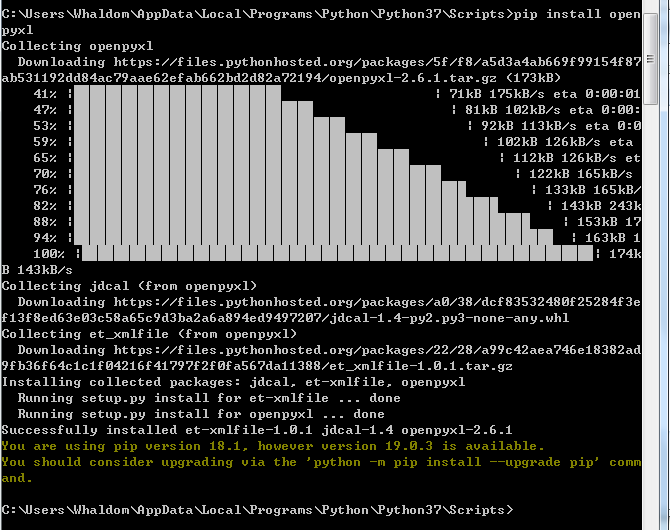
# 调试篇

## Openpyxl安装



## pip升级中遇到的问题

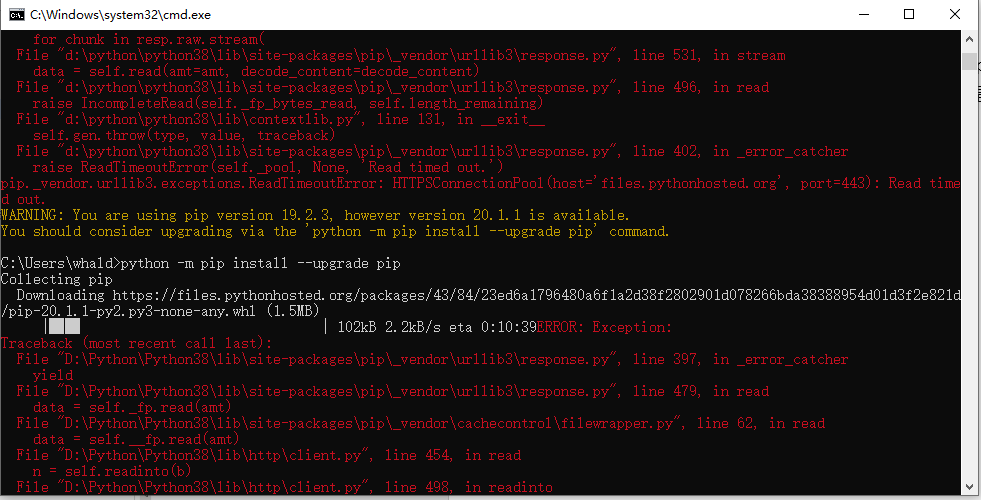
### pip组件升级

pip install时，提示pip版本过低，这时需要升级pip version。

使用此命令升级pip：

python -m pip install --upgrade pip

输入升级命令之后显示升级失败。



连接国外网站超时导致，尝试以下代码：

python -m pip install --upgrade pip -i https://pypi.tuna.tsinghua.edu.cn/simple/

安装成功。

出现这种情况实际是因为国外的网站访问限速，所以会一直报错timeout，可以用参数延长timeout也可以换源。

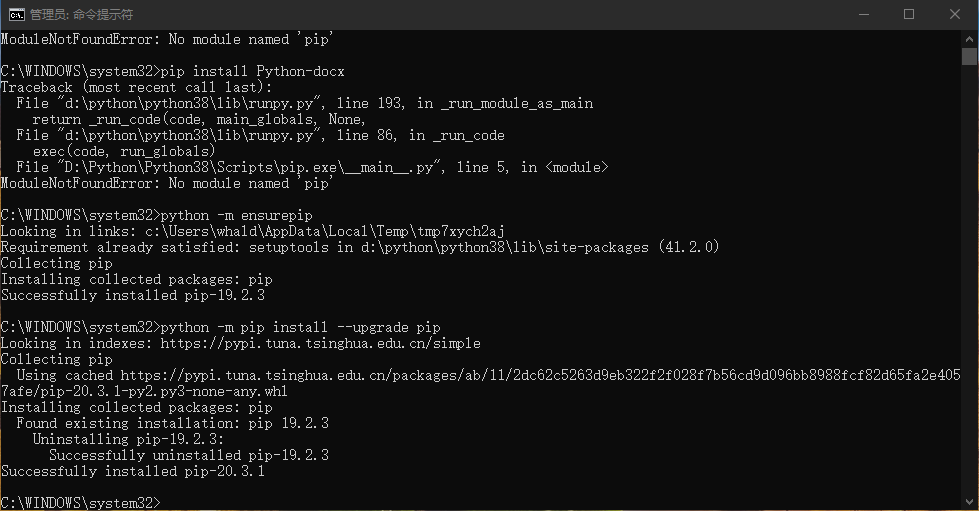
常见pip下载镜像还有：

清华：https://pypi.tuna.tsinghua.edu.cn/simple/ **# 推荐**

阿里：https://mirrors.aliyun.com/pypi/simple

中国科学技术大学：<https://pypi.mirrors.ustc.edu.cn/simple>

### No module named ‘pip’



python -m ensurepip

python -m pip install --upgrade pip

## Pip install

### 临时使用

可以在使用pip的时候加参数-i https://pypi.tuna.tsinghua.edu.cn/simple

例如：pip install -i https://pypi.tuna.tsinghua.edu.cn/simple gevent，这样就会从清华这边的镜像去安装gevent库。

pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pinyin

### 永久修改

**windows**

直接在user目录中创建一个pip目录，如：C:\Users\xx\pip，新建文件pip.ini，内容如下

[global]

index-url = https://pypi.tuna.tsinghua.edu.cn/simple

**Linux**

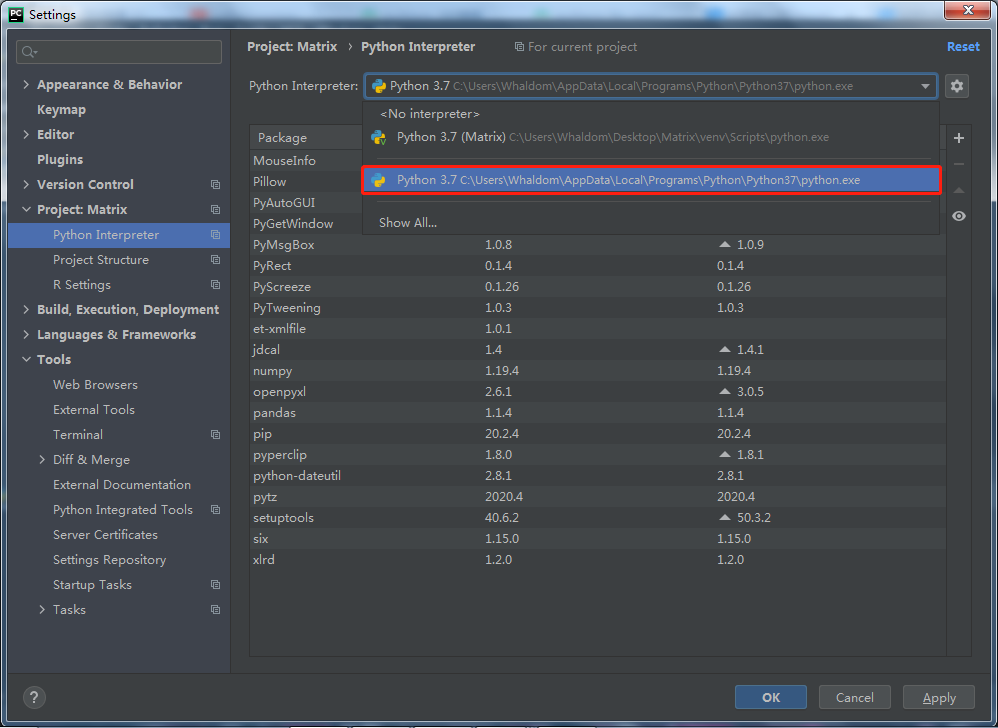
修改 ~/.pip/pip.conf (没有就创建一个)， 修改 index-url至tuna，内容如下：

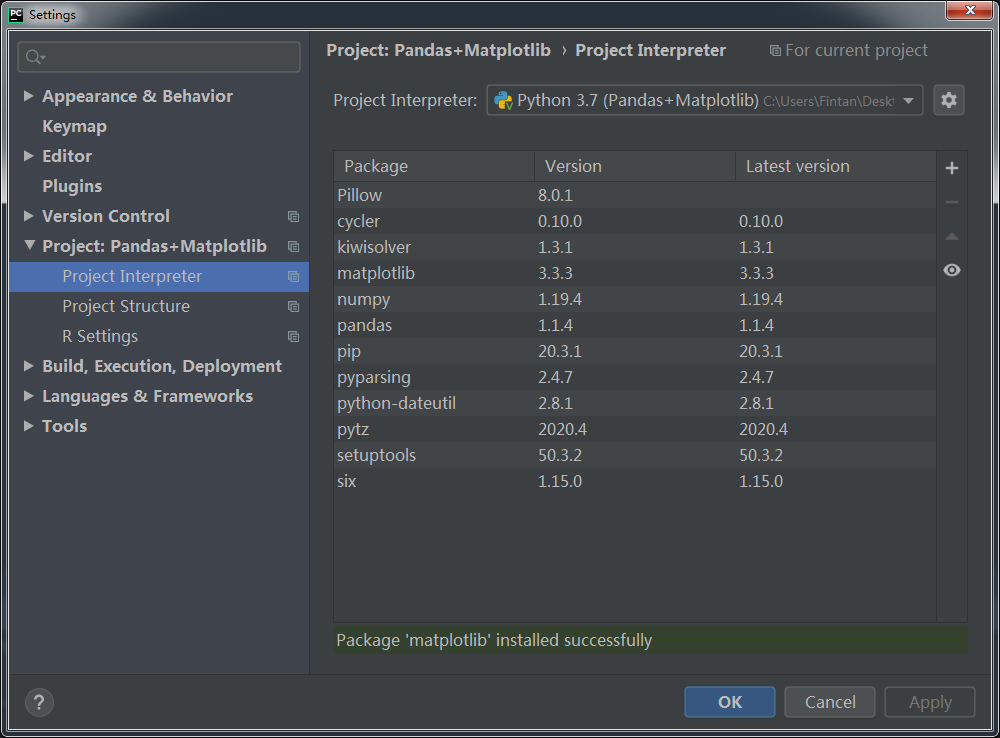
[global]

index-url = https://pypi.tuna.tsinghua.edu.cn/simple

## Pycharm章

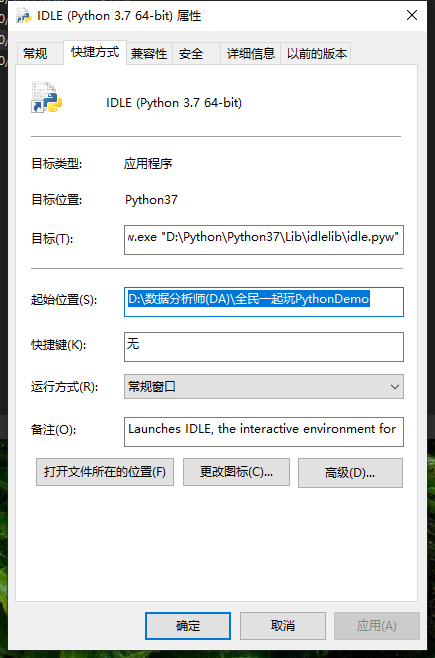
### 解释器/编译器





# 小技巧篇

## Demo保存默认路径修改



自动补全（联想），Tab键，选择使用blank，不是enter。

## 引用库（import）

无论是使用import还是from import，第一个要求是代码能够正常运行，其次，根据代码维护性，团队编码风格来确定选择哪一种方案。

如果我们只会使用到某个模块下面的一个函数（或者常量、类）并且名字不会产生混淆，可识别性高，那么from 模块名 import 函数名这没有什么问题。

如果我们会用到一个模块下面的多个函数，或者是我们将要使用的函数名、常量名、类名可能会让人产生混淆（例如 re.S、re.I），那么这种情况下，import 模块名然后再 模块名.xxx来调用会让代码更加清晰，更好维护。

但无论什么情况下，都禁止使用from xxx import \*这种写法，它会给你带来无穷无尽的噩梦。

from collections import deque # 少量使用mod中的变量、方法

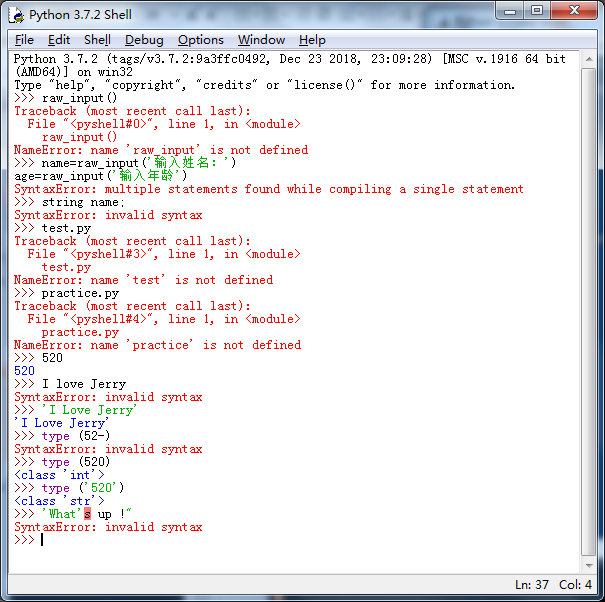
# import collections as **cls # 大量使用mod中的变量、方法**

# from collections import \* # 不建议：\*内容过多，变量、方法极易混淆

queue = deque(["YuanShuai", "MaoZedong", "Eric"])

# queue = **cls**.deque(["YuanShuai", "MaoZedong", "Eric"])

# 字符串



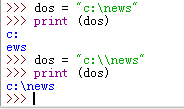
“What’s up !”

'What\'s up!' #反斜杠

## 字符串的快速入门

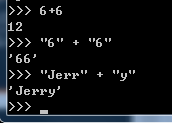
### 打印字符中需要转义的解决方案：

转义字符“\”



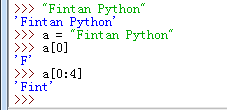
原始字符串r’’





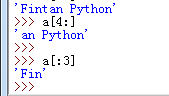
## 字符串的基本操作

### 索引

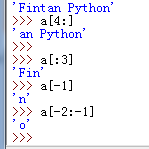


在Python中，按前包括，后不包括原则，即[0, 4)。

故索引为4的字符不会拿到。



### 切片



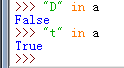
1. **Len()字符长度**



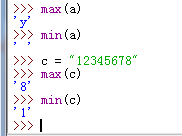
1. **打印分割线**

C:\Users\Whaldom\AppData\Local\Temp\1553497608(1).png

1. **判断某个字符是否在字符串**



1. **求最大值、最小值**



对应ASCII码大小

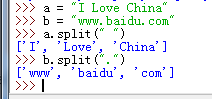


### 内建函数查询

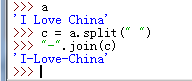


## 字符串常用方法

### Split()分割字符串

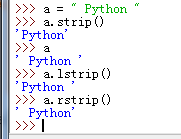


### Join()拼接字符串



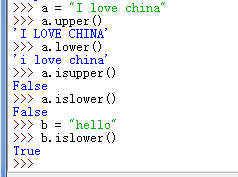
### 去空格

1. **Strip()去掉左右两边空格**
2. **Lstrip()去掉左边空格**
3. **Rstrip()去掉右边空格**



### 大小写转换

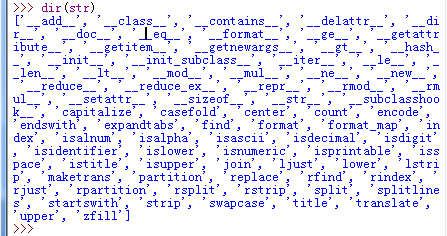
1. **Upper()大写**
2. **Lower()小写**
3. **Isupper()是否大写**
4. **Islower()是否小写**
5. **Istitle()是否是标题**



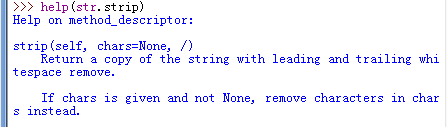


### 字符串的其他方法

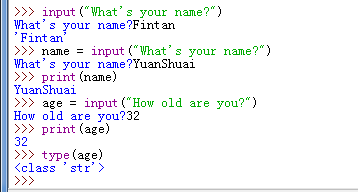
1. **Dir**



1. **Help**

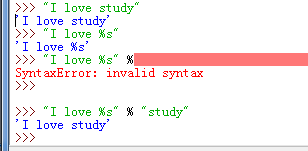


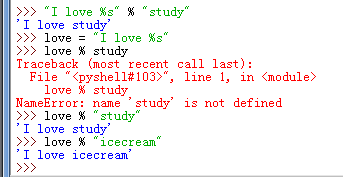
### 格式化输出



1. **占位符**

%s





%d



%f



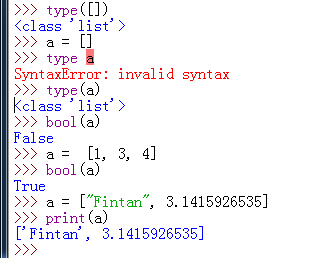
1. **提倡的格式化输出**

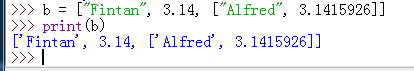
花括号 + 索引占位符



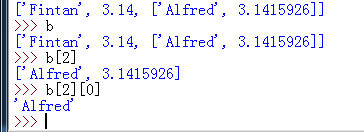
# 列表

## 概念

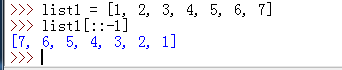


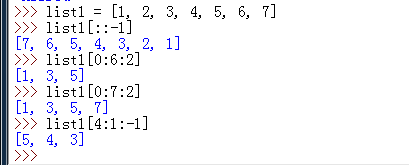


### 二位列表索引



### 倒叙遍历





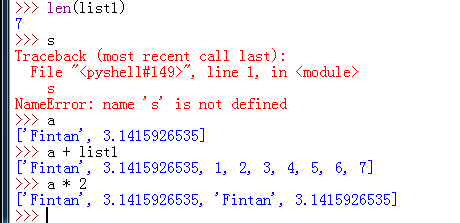
### 步长

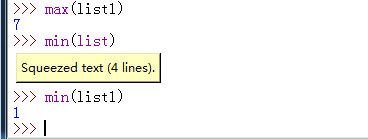




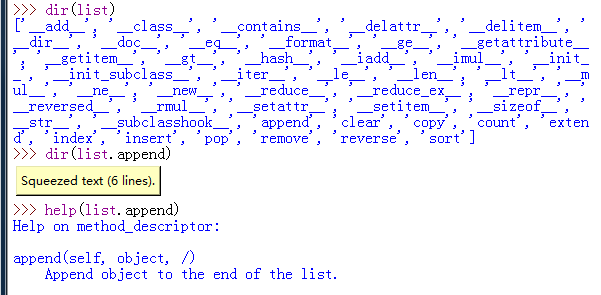
## 列表的基本操作

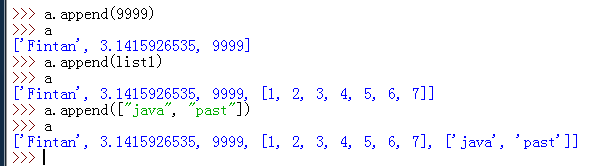
Len(), +, \*, max, min





### append





### pop

返回pop()的值，默认pop列表最后一个值，后进先出，堆栈。

## 列表的高级操作

### 列表当队列(QUEUE)使用

from collections import deque

queue = deque(["Eric", "John", "Michael", "YuanShuai", "MaoZedong"])

queue.append("Terry")

queue.append("Graham")

queue.popleft()

queue.popleft()

print(queue)

### Numpy数组

列表转为Numpy数组，方可进行运算。否则List只是容器，不能直接参与运算。

import numpy

x\_left = [[9.96, 2968.08], [9.17, 2388.64], [7.78, 1515.02]]

x\_leftNp = np.array(x\_left)

for moveI in range(0, len(x\_left)):

x\_leftNp[moveI][0] += aMove

### 列表中的列表，取值计算

此处x是一个数组，

a = 3

b = 5

x\_list = [[9.96, 2968.08], [9.17, 2388.64], [7.78, 1515.02]]

for x in x\_list:

print('记住x在for in循环中的**数据类型**，是**会变**的：' + str(type(x)))

# 此处x的数据类型是列表型，<class 'list'>

y01 = a \* x[0] + b

print(y01)

y02 = a \* x[1] + b

print(round(y02, 2))

### 矩阵

详见程序：matrixbase.py

# 数学

## 平方、开方

### 平方

**# 方法一：**

pow(144, 2) **# 144**

**# 方法二：**

12\*\*2 **# 144**

### 开方

**# 方法一：**

import math

math.sqrt(144) **# 12**

**# 方法二：**

pow(144, 0.5) **# 12**

**# 方法三：**

144\*\*0.5 **# 12**

## round()

注意：

在python2.7的doc中，round()的最后写着，"Values are rounded to the closest multiple of 10 to the power minus ndigits; if two multiples are equally close, rounding is done away from 0." 保留值将保留到离上一位更近的一端（四舍六入），如果距离两端一样远，则保留到离0远的一边。所以round(0.5)会近似到1，而round(-0.5)会近似到-1。

但是到了python3.5的doc中，文档变成了"values are rounded to the closest multiple of 10 to the power minus ndigits; if two multiples are equally close, rounding is done toward the even choice." 如果距离两边一样远，会保留到偶数的一边。比如round(0.5)和round(-0.5)都会保留到0，而round(1.5)会保留到2。

所以如果有项目是从py2迁移到py3的，可要注意一下round的地方（当然，还要注意/和//，还有print，还有一些比较另类的库）。

**# 方法一：**

round(x, 2) **# 为float型保留2位小数**

# 思想篇

## 师授之以渔，非鱼

形成自己的思想，自己去查工具书和文档。

## 勿在浮沙筑高台

研究容器

研究对象

研究字符串

继续深入学习

## Python数据分析三宝

数据分析有大宝：matplotlib, numpy, pandas.

## 开始coding前，先分析

做习题，要像开发项目的流程一样：

（需求 -> 需求分析 -> 设计 -> 编码 -> 测试 -> 交付等），

拿到一个习题，还没有进行分析，就匆匆忙忙敲代码，

这是不好的学习方法。

先要分析清楚题目（所谓的需求分析，何为需求分析，请自己搜索），

再设计，用那种数据类型（数据结构）

来组织或保存数据，用何种算法来计算效率最高，

用面向过程，还是面向对象的编程范式，还是用函数式编程等等。

设计后，再编写代码，最后写测试。

（也可以采用软件开发过程中的应用方法，测试驱动开发等）

# 代码

# 更新内容

## 0601

Pip升级中遇到的问题

Pip install

## 0819

pip升级报错

## 1125

文件名：跬步小流，改为：PythonExp

列表的高级操作

引用

## 1130

更新引用

解释器/编译器

## 1207

更新版式，重调章节