廖雪峰python教程总结

学习思路：看语法基础，跟着视频和书籍学，注意实操；看优秀的例子，总结算法精髓，试着自己编程，养成先森林后树木的习惯，一定先想好框架，再编程，再优化框架……可参考敏捷开发（先完成功能后优化代码）

廖雪峰python：<https://www.liaoxuefeng.com/wiki/1016959663602400> （有安装解释器步骤）

1. 自己总结

* 要注意GPU对应

1. Ubuntu

* 安装numpy等包：<https://blog.csdn.net/yato0514/article/details/78754551>
* Anaconda中python2.7和python3.7共存：<https://www.jianshu.com/p/d2e15200ee9b>

<https://blog.csdn.net/weixin_38569817/article/details/82466847>

* 安装anaconda：<https://blog.csdn.net/qq_15192373/article/details/81091098>
* conda安装特定版本tensorflow：<https://www.cnblogs.com/wxshi/p/6805120.html>

conda install --channel https://conda.anaconda.org/anaconda tensorflow=1.6.0

* 1

1. Window安装第三方库方法
2. 安装anaconda：可以从Anaconda官网下载GUI安装包，安装包有500~600M，所以需要耐心等待下载，下载后按默认安装即可。网速慢的同学请移步国内镜像。下载后直接安装，更改安装位置，否则会被隐藏？Anaconda会把系统Path中的python指向自己自带的Python，并且，Anacnda安装的第三方模块会安装在Anaconda自己的路径下，不影响系统已安装的Python目录。

安装完后，按如下链接配置pycharm环境。Location是新建的位置，自己新建一个文件夹，放在anaconda文件夹内比较方便。

在pycharm中导入anaconda的库\_qq\_44616044的博客-CSDN博客 <https://blog.csdn.net/qq_44616044/article/details/90142037>

其它相关链接

JetBrains PyCharm for Anaconda <https://www.jetbrains.com/pycharm/promo/anaconda/>

Getting started with Anaconda — Anaconda documentation <https://docs.anaconda.com/anaconda/user-guide/getting-started/>

<https://anaconda.org/>

1. 直接在pycharm，setting-解释器搜索安装
2. Pip安装第三方库步骤：cd C:\Program Files (x86)\Python27\Scripts（pip在此文件夹下，需先下载第三方包，从第三方库都会在Python官方的pypi.python.org（<https://pypi.org/> ）网站注册，注意按第三方库命名方式：<https://blog.csdn.net/lanchunhui/article/details/62417519>下载对应的包）；pip install –user <文件名.后缀>，先到网站下载python对应版本库到pip所在文件夹
3. Conda常用命令

* conda install 包名
* conda remove 包名
* conda deactivate
* conda activate 环境名

1. 自己总结

* anaconda=python+常用用第三方库，安装anaconda后可以不安装python，可以使用pycharm直接新建anaconda工程即可，Anaconda指的是一个开源的Python发行版本，其包含了conda、Python等180多个科学包及其依赖项。因为包含了大量的科学包，Anaconda 的下载文件比较大（约 531 MB），如果只需要某些包，或者需要节省带宽或存储空间，也可以使用Miniconda这个较小的发行版（仅包含conda和 Python）

使用方法：<https://www.cnblogs.com/yuxuefeng/p/9235431.html>

<https://www.cnblogs.com/aloiswei/p/6510355.html>

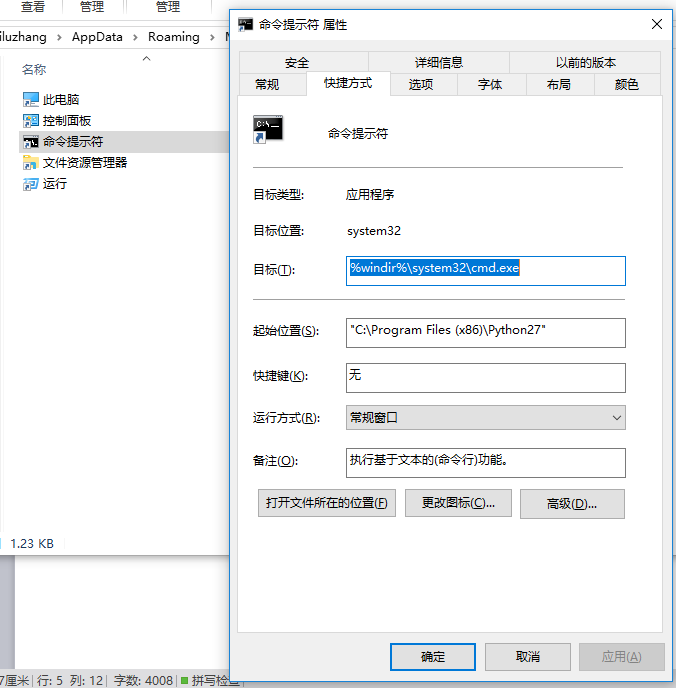
* 各种python.exe统称解释器（interpreter）
* 注意环境变量，添加环境变量后要重启软件，如anaconda未配置库的环境变量，导入库失败
* Interpreter：一个project可以建立自己环境，也可以使用常用的环境。解释器版本和已经默认安装好了。最好与python安装位置共用第三方库。

1. cmd运行命令：cd（切换目录，运行.py 文件时需先切换到其对应目录下，然后用python name.py打开，跨盘切需先输入“盘符：”，如G:,再用cd命令，cd\可直接进入根目录）；dir（查看目录）；more（查看文件内容）；exit()退出当前命令解释程序。
2. 问题记录
3. Win10下更改cmd路径步骤：

a.右击打开文件所在位置，



b.右击打开属性



c.将默认起始位置%HOMEDRIVE%%HOMEPATH%改为python.exe所在文件夹路径

还可通过更改环境变量的方式快速进入python.exe

1. Pip安装第三方包报错：is not a supported wheel on this platform

安装包时报错Python2.7进入shell查看pip支持命令，注意安装时需cd到pip所在文件夹python安装文件夹scripts文件夹，包也放在该文件夹

import pip.internal; print(pip.\_internal.pep425tags.get\_supported())

1. Pip安装第三方包报错：is not a supported wheel on this platform.

说明第三方包与python版本不匹配，用2）步查询下载对应包，注意看名字

1. Pip安装第三方包报错：Could not install packages due to an EnvironmentError: [Error 5]，Consider using the `--user` option or check the permissions.

权限问题，使用pip install –user 文件名.后缀 就可以了

1. pip安装matplotlib包报错：Could not find a version that satisfies the requirement kiwisolver

缺少kiwisolver包，在<https://pypi.org/>按名字搜索kiwisolver包，相似报错缺啥补啥，可以先尝试安装setuptools包；

另外可以直接从[Anaconda官网](https://www.anaconda.com/download/)下载GUI安装包，安装包有500~600M，包括Pillow，以及MySQL驱动程序，Web框架Flask，科学计算Numpy等。下载后直接安装，Anaconda会把系统Path中的python指向自己自带的Python，并且，Anaconda安装的第三方模块会安装在Anaconda自己的路径下，不影响系统已安装的Python目录。可以尝试直接import第三方模块。常用模块还有：functools；operator；

1. anaconda安装时未勾选配置环境变量，导入库失败

解决：在系统环境变量添加C:\Program\_Files1\anaconda3\Library\bin，

1. pycharm直接安装matplotlib报错：fatal error C1083: 无法打开包括文件: “ft2build.h”: No such file or directory

解决：关闭杀毒软件，升级了pip，把matplotlib包下好后再装了重启了一次又装好了。莫名其妙好了。注意pip安装时要转到对应环境目录

1. 安装pattern：<https://anaconda.org/asmeurer/pattern>；conda install -c asmeurer pattern
2. 常用内置函数（如min（））、方法（list.append()）及操作、语法

注意力：函数参数为形参，一般有返回值，方法参数为实参，一般无返回值，字符串、元组、

（<http://www.runoob.com/python/python-built-in-functions.html>）

* python中的字符数字之间的转换函数<https://www.cnblogs.com/junglefish/p/8409581.html>
* python的基本数学运算 - qiwsir - 博客园 <https://www.cnblogs.com/qiwsir/p/4209492.html>
* with as：<https://blog.csdn.net/u012609509/article/details/72911564>
* lambda函数：<https://blog.csdn.net/zjuxsl/article/details/79437563>
* padded\_batch<https://blog.csdn.net/yeqiustu/article/details/79795639>
* tensorflow常用深度学习函数：<https://blog.csdn.net/weixin_31767897/article/details/79365968>
* 常用内置函数：len()、min()、max()、int()、str()、del、list(‘zhang’)
* append()-添加一个参数，会将[]加进去，格式为a.append(),下面格式类似
* extend()-添加一个列表中的所有参数，不添加[]；
* insert()插入，
* pop()-删除，
* sort()-逆序，
* replace(‘a’，‘b’)-替换；
* range(start，stop，step)；list[start:stop:step]表示第start+1个-第stop个，为-时反过来；isinstance(a, Iterable)：判断a是否可迭代；
* enumerate函数可以把一个list变成索引-元素对；
* os.listdir可以列出文件和目录；
* str.lower将字符串str中所有字符变为小写，str.upper()；
* iter()函数可以把list、dict、str等Iterable变成Iterator，检查格式：isinstance(iter([]), Iterator)；map()函数接收两个参数，一个是函数，一个是Iterable，map将传入的函数依次作用到序列的每个元素，并把结果作为新的Iterator返回；
* reduce（）把一个函数作用在一个序列[x1, x2, x3, ...]上，这个函数必须接收两个参数，reduce把结果继续和序列的下一个元素做累积计算，其效果就是：reduce(f, [x1, x2, x3, x4]) = f(f(f(x1, x2), x3), x4)，from functools import reduce；
* filter()把传入的函数依次作用于每个元素，然后根据返回值是True还是False决定保留还是丢弃该元素，返回的也是一个Iterator，需要用list()函数获得所有结果并返回list；
* s.strip(rm)：删除s字符串中开头、结尾处，位于 rm删除序列的字符，s.lstrip(rm)：删除s字符串中开头处，位于 rm删除序列的字符，s.rstrip(rm) ：删除s字符串中结尾处，位于 rm删除序列的字符，当rm为空时，默认删除空白符（包括'\n', '\r', '\t', ' ')；
* sorted(['bob','about','Zoo','Credit'], key=str.lower, reverse=True)其中，key函数来实现自定义的排序， reverse=True表示进行反向排序；
* functools.wraps将decorator中wrapper.\_\_name\_\_ = func.\_\_name\_\_，定义wrapper()的前面加上@functools.wraps(func)；
* functools.partial就是帮助我们创建一个偏函数的，int2 =functools.partial(int, base=2)；
* b=\* //定义函数b，获取对象的第1个域的值;
* list.count(list[0])：统计list[0]在list中出现的次数
* del(list[n]):删除list中list[n]元素
* set():创建一个无序不重复元素集，可进行关系测试，删除重复数据，还可以计算交集、差集、并集等。
* pickle模块：用于对文件的操作，注意读写完后调用close（）函数，调用close（）才能完成缓存到磁盘的读写
* 十六进制：0x，八进制：0o；二进制：0b
* y=Input(“y：”)函数只会将键盘输入值赋给y；y=Input(“y：”)[3],y取值为输入字符串第2个字符
* Unicode码点和名称网站：http：//unicode-table.comb
* 序列相加及乘法：d=['nd']+4\*['th']得到d= ['nd', 'th', 'th', 'th', 'th']

再输入d=d+[2]为['nd', 'th', 'th', 'th', 'th', 2]，不能拼接不同类型的序列

* list[0:10:x]中x为步长，d[:5:2]=['nd', 'th', 'th']，x不能为0，可为负数，如list[10:0:x]从右往左提取元素，都按第一个索引包含在片内，第二个索引不包含片内的原则
* list(‘zhang’)=’z’’h’’a’…
* list[1:1]=[2,4,5]可给切片赋值，相当于替换了一个空切片
* 字符串所有元素赋值和切片赋值都是非法的
* 注意不同版本python的print函数打印格式不一样
* status = 'friend' if name.endswith('an') else 'stranger'：name以an结尾status=friend，否则status=stranger
* a=[x\*x for x in range(10)]= [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
* a=[x\*x for x in range(10) if x%3 == [0, 9, 36, 81]
* a=[(x,y) for x in range(2) for y in range(2)]= [(0, 0), (0, 1), (1, 0), (1, 1)]
* 链式比较：0<a<10
* 对于不再使用的值，python解释器会立即删除
* exec函数：将字符串作为代码执行
* eval函数：计算字符串表示的python表达式的值
* 变量名要避免和函数名和关键字同名
* Python也会优先使用后命令的同名变量和函数，如局部变量与全局变量同名时，全局变量会被遮住
* 注意列表和字典作为列表参数时，相当于实参，直接对外部列表、字典进行操作，另外，使用x=[]赋值语句是不行的
* 在没完善一个程序之前，我们不知道程序在哪里会出错，与其让它在运行最崩溃，不如在出现错误条件时就崩溃，这时候就需要assert断言的帮助。如：assert a==b，若a==b为真继续执行，若a==b为假，程序崩溃
* 注意递归命名空间，return返回数据只给上一层，函数有return，可能调用时不能返回数据
* 抽象：隐匿不必要的细节。比如用函数将步骤的具体细节打包，有需要再看细节
* 常用模块：（待完善）

技巧：help（）函数；\_\_all\_\_变量：模块公有接口（函数）；\_\_doc\_\_变量：函数简单描述；dir（）函数：查看对象所有属性

内置模块：（具体查阅《python基础》第十-十二章；）

abc：提供多个很有用的抽象类

sys：与解释器紧密相关的变量和函数

os：让你访问多个操作系统服务

fileinput、json和pickle：读写文件

shelve：用于创建永久映射，其内容存储在使用给定文件名的数据库中

time：时间模块

re：对正则表达式支持，字符串查找匹配表达式

sets（集合）、heapq（堆）和deque（双端队列）：提供三种数据结构，内置set也能实现集合

tkinter：是Python 的标准 Tk GUI 工具包的接口，

<http://www.runoob.com/python/python-gui-tkinter.html>

第三方模块：

* numpy：矩阵
* matplotlib：画图
* pandas：存储二维数据，如Excel，<https://www.cnblogs.com/chenice/p/7257237.html>
* pattern：<https://blog.csdn.net/artemisrj/article/details/41556449>

技巧总结：

1. 继承object类的作用：<https://blog.csdn.net/DeepOscar/article/details/80947155>

* 少使用if/else处理异常，尽量使用try/except/else/finally处理异常
* 注意使用继承，继承太多容易出错
* Python 3之前版本创建新类需要在文件头添加\_\_metaclass\_\_ = type
* 类定义中将self改为类名，所有对象共用一个属性
* Python目录用来放置模块文件夹：Python27\\lib\\site-packages
* 学习python最有用的文档是《python库参考手册》
* 注意软件默认换行符’\r’、’\n’或’\r\n’,在使用readline、readlines函数读取的list会有区别、
* 可能的话，首选for循环，另外避免相同的语句重复出现
* Json和pickle模块：json可以用于字符串或者字典等与python数据类型之间的序列化与反序列化操作

pickle可以用于python特有类型与python数据类型之间的序列化与反序列化操作。主要用到dumps、loads、dump、load函数。（https://www.cnblogs.com/renpingsheng/p/7184005.html）

dump可在文件中存储多个变量，load时一个一个load就可以了

* 数据库：MySQL，大家都在用，一般错不了；

PostgreSQL，学术气息有点重，其实挺不错，但知名度没MySQL高；

sqlite，嵌入式数据库，适合桌面和移动应用。

模块学习笔记

* numpy模块

NumPy 算术函数 | 菜鸟教程 <https://www.runoob.com/numpy/numpy-arithmetic-operations.html>

* mat[n,m]:n表示行，m表示列，可用list及其切片表示
* matplotlib模块

网址：

<https://blog.csdn.net/lilianforever/article/details/48786795>；

<https://www.cnblogs.com/kallan/p/7365472.html>；

<https://www.baidu.com/s?tn=80035161_2_dg&wd=%E6%80%8E%E4%B9%88%E5%AD%A6%E4%B9%A0matplotlib>；

3、C语言与python对比，可作为入门语言思路

|  |  |  |
| --- | --- | --- |
| C语言叫法 | C语言 | Python |
| 主函数 | void main() | if \_\_name\_\_==“\_\_main\_\_”: |
| 源文件 | .c文件 | 模块，.py文件 |
| 引用.h头文件 | #include “\*.h” | import 模块  import 模块.函数 as 自己命名  from 模块 import 函数1，函数2，函数3  from 模块 import \* 当且仅当要导入模块一切时才使用 |
| 库函数 | 芯片官方给的库 | Python内部模块或第三方模块 |
| 常用功能及快捷键 | 全局查找、当前文件查找、跳转到定义、function函数视图 |  |
| 调试方法 | 设置断点、查看变量 |  |
| 强制转换格式 | （float）a | float（a） |
| 变量初始化 | int a；可以赋值，初始值为0，且需要定义数据类型 | a=10，必须赋初值，因为python没有初始值，不需定义数据类型 |
|  |  |  |

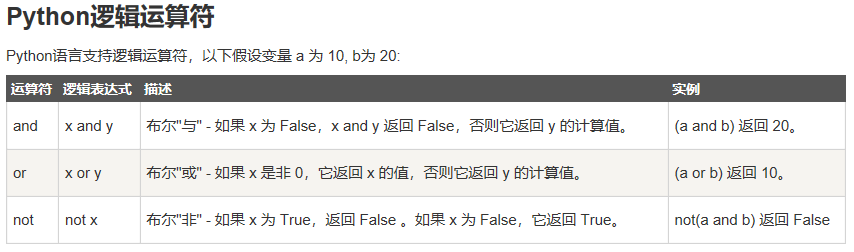
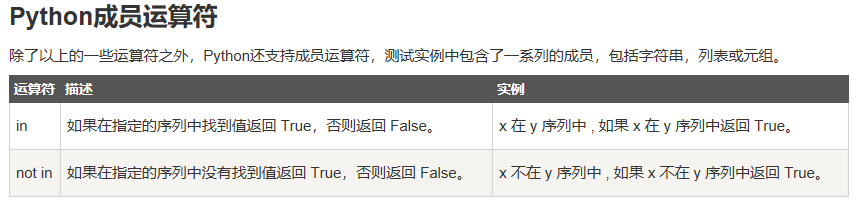
与C语言不同点：

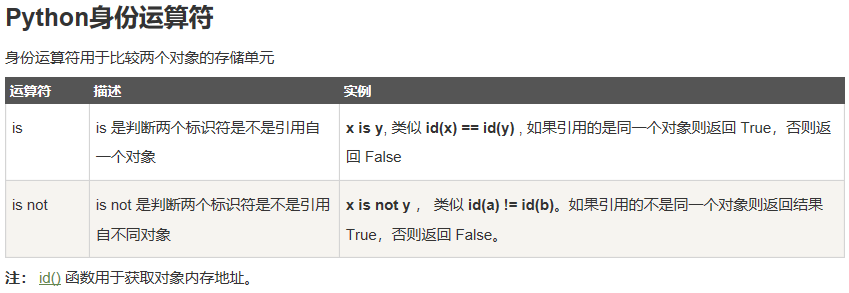
1、列表list参数作形参时是会被改变，因为两个变量是指向这个列表，要避免需创建副本，可通过切片n[:]；

1. 运算符（<http://www.runoob.com/python/python-operators.html>）









Python除了基础代码库：网络、文件、GUI、数据库、文本等大量内容，还有第三方库，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。

一般来说，第三方库都会在Python官方的pypi.python.org网站注册，要安装一个第三方库，必须先知道该库的名称，可以在官网或者pypi上搜索，比如Pillow的名称叫Pillow，因此，安装Pillow的命令就是。

在使用Python时，我们经常需要用到很多第三方库，例如，上面提到的Pillow，以及MySQL驱动程序，Web框架Flask，科学计算Numpy等。用pip一个一个安装费时费力，还需要考虑兼容性。我们推荐直接使用Anaconda，这是一个基于Python的数据处理和科学计算平台，它已经内置了许多非常有用的第三方库，我们装上Anaconda，就相当于把数十个第三方模块自动安装好了，非常简单易用。

可以从Anaconda官网下载GUI安装包，安装包有500~600M，所以需要耐心等待下载。网速慢的同学请移步国内镜像。下载后直接安装，Anaconda会把系统Path中的python指向自己自带的Python，并且，Anaconda安装的第三方模块会安装在Anaconda自己的路径下，不影响系统已安装的Python目录。

4、Python的哲学就是简单优雅，尽量写容易看明白的代码，尽量写少的代码。

5、应用：首选是网络应用，包括网站、后台服务等等；其次是许多日常需要的小工具，包括系统管理员需要的脚本任务等等；另外就是把其他语言开发的程序再包装起来，方便使用。

6、缺点：运行速度慢，开发一个下载MP3的网络应用程序，C程序的运行时间需要0.001秒，而Python程序的运行时间需要0.1秒，慢了100倍，但由于网络更慢，需要等待1秒，用户很难感觉到1.001秒和1.1秒的区别；代码不能加密，C语言可发布机器码，解释型的语言，则必须把源码发布出去。（解释型语言:程序不需要编译，程序在运行时才翻译成机器语言，每执 行一次都要翻译一次。因此效率比较低。源代码不是直接翻译成机器语言，而是先翻译成中间代码，再由解释器对中间代码进行解释运行。比如Python/JavaScript / Perl /Shell等都是解释型语言。）

7、官方解释器：Cpython，运用最广；ipython：基于cpython，交互式有所增强；pypyPyPy采用JIT技术，对Python代码进行动态编译（注意不是解释），所以可以显著提高Python代码的执行速度；绝大部分Python代码都可以在PyPy下运行，但是PyPy和CPython有一些是不同的，这就导致相同的Python代码在两种解释器下执行可能会有不同的结果。如果你的代码要放到PyPy下执行，就需要了解PyPy和CPython的不同点；Jython是运行在Java平台上的Python解释器，可以直接把Python代码编译成Java字节码执行；IronPython和Jython类似，只不过IronPython是运行在微软.Net平台上的Python解释器，可以直接把Python代码编译成.Net的字节码。

8、绝对不能用Word和Windows自带的记事本。Word保存的不是纯文本文件，而记事本会自作聪明地在文件开始的地方加上几个特殊字符（UTF-8 BOM），结果会导致程序运行出现莫名其妙的错误。推荐使用文本编译器：Sublime Text、Notepad++，文件名只能是英文字母、数字和下划线的组合。

9、用Python开发程序，完全可以一边在文本编辑器里写代码，一边开一个交互式命令窗口，在写代码的过程中，把部分代码粘到命令行去验证，事半功倍！前提是得有个27'的超大显示器。

10、整数和浮点数在计算机内部存储的方式是不同的，整数运算永远是精确的（除法难道也是精确的？是的！），而浮点数运算则可能会有四舍五入的误差；Python还允许用r''表示''内部的字符串默认不转义；允许用'''...'''的格式表示多行内容，三行前后需加三个引号；True和False首字母大写；//除法取整数，%求余；

11、Python解释器干了两件事情：在内存中创建了一个'ABC'的字符串；在内存中创建了一个名为a的变量，并把它指向'ABC'。注意程序是：单行执行。

12、ASCII编码是1个字节，而Unicode编码通常是2个字节，UTF-8编码把一个Unicode字符根据不同的数字大小编码成1-6个字节。

13、用记事本编辑的时候，从文件读取的UTF-8字符被转换为Unicode字符到内存里，编辑完成后，保存的时候再把Unicode转换为UTF-8保存到文件。

14、在操作字符串时，我们经常遇到str和bytes的互相转换。为了避免乱码问题，应当始终坚持使用UTF-8编码对str和bytes进行转换。

由于Python源代码也是一个文本文件，所以，当你的源代码中包含中文的时候，在保存源代码时，就需要务必指定保存为UTF-8编码。当Python解释器读取源代码时，为了让它按UTF-8编码读取，我们通常在文件开头写上这两行：

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

第一行注释是为了告诉Linux/OS X系统，这是一个Python可执行程序，Windows系统会忽略这个注释；第二行注释是为了告诉Python解释器，按照UTF-8编码读取源代码，否则，你在源代码中写的中文输出可能会有乱码。

15、对于数组，可以用-1做索引，直接获取最后一个元素，如a[-1]，以此类推，可以获取倒数第2个、倒数第3个。

16、tuple和list非常类似，但是tuple一旦初始化就不能修改，tuple用()，list用[]，1个元素的tuple定义时必须加一个逗号“，”，如t=（4，）按小括号进行计算，计算结果自然是1。

17、list[：2]表示第二位之前的所有值，list[2:]表示第二位之后的所有值，其中list为数组名，；dict用{},.dict.keys()是dict所有key值，dict.values是dict所有value值，dict.iter‘’s()键值对。

set用([]),如a = set([1,2,3]),添加用a.add(4)是无序的，元素不重复

18、input()返回的数据类型是str，str不能直接和整数比较，必须先把str转换成整数。Python提供了int()函数来完成这件事情。

19、Python提供一个range()函数，可以生成一个整数序列，再通过list()函数可以转换为list，如：list(range(5))。

20、dict内部存放的顺序和key放入的顺序是没有关系的。dict的key必须是不可变对象。

21、实际上pass可以用来作为占位符，比如现在还没想好怎么写函数的代码，就可以先放一个pass，让代码能运行起来。

22、常用内置函数：isinstance(x, (int, float))；

23、类似于c语言包含头文件，导入另一个文件用import （file’s name），用file’name.function'name调用对应函数。

24、在语法上，返回一个tuple可以省略括号，而多个变量可以同时接收一个tuple，按位置赋给对应的值，所以，Python的函数返回多值其实就是返回一个tuple，但写起来更方便。

25、设置默认参数时，有几点要注意：一是必选参数在前，默认参数在后，否则Python的解释器会报错（思考一下为什么默认参数不能放在必选参数前面）；二是如何设置默认参数。当函数有多个参数时，把变化大的参数放前面，变化小的参数放后面。变化小的参数就可以作为默认参数。

26、 定义默认参数牢记一点：默认参数必须指向不变对象！如：数字，字母，字符串，none。

27、可以用必选参数、默认参数、可变参数、关键字参数和命名关键字参数，这5种参数都可以组合使用。但是请注意，参数定义的顺序必须是：必选参数、默认参数、可变参数、命名关键字参数和关键字参数。最神奇的是通过一个tuple和dict，你也可以调用函数。\*args是可变参数，args接收的是一个tuple；\*\*kw是关键字参数，kw接收的是一个dict。\*nums表示把nums这个list的所有元素作为可变参数传进去。\*\*extra表示把extra这个dict的所有key-value用关键字参数传入到函数的\*\*kw参数，kw将获得一个dict，注意kw获得的dict是extra的一份拷贝，对kw的改动不会影响到函数外的extra。

28、解决递归调用栈溢出的方法是通过尾递归优化。遗憾的是，大多数编程语言没有针对尾递归做优化，Python解释器也没有做优化，也会导致栈溢出。

29、写操作系统，这个只能用C语言写；写手机应用，只能用Swift/Objective-C（针对iPhone）和Java（针对Android）；写3D游戏，最好用C或C++；python可以自动备份你的MP3；可以做网站，很多著名的网站包括YouTube就是Python写的；可以做网络游戏的后台，很多在线游戏的后台都是Python开发的。

30、我们来介绍Python中非常有用的高级特性，1行代码能实现的功能，决不写5行代码。请始终牢记，代码越少，开发效率越高。

31、如果要同时迭代key和value，可以用for k, v in d.items()；字符串也是可迭代对象

32、列表生成式则可以用一行语句代替循环生成上面的list：[x \* xforxinrange(1,11)]，还可以使用两层循环，可以生成全排列：[m + nformin'ABC'fornin'XYZ'],其中‘+’是将字符mn连接起来的意思

33、创建generator的方法：a、把一个列表生成式的[]改成()；使用yield关键字，在每次调用next()的时候执行，遇到yield语句返回，再次执行时从上次返回的yield语句处继续执行。可以被next()函数调用并不断返回下一个值的对象称为迭代器：Iterator。Python的for循环本质上就是通过不断调用next()函数实现的，遇StopIteration返回。

34、高阶函数除了可以接受函数作为参数外，还可以把函数作为结果值返回。

35、返回闭包时牢记一点：返回函数不要引用任何循环变量，或者后续会发生变化的变量。

36、返回闭包（函数）时牢记一点：返回函数不要引用任何循环变量，或者后续会发生变化的变量，否则会发生意想不到的结果。

37、lambda表示匿名函数，如：lambda x: x \* x，冒号前面的x表示函数参数，只能有一个表达式，不用写return，返回值就是该表达式的结果。可以把匿名函数赋值给一个变量，再利用变量来调用该函数： f = lambda x: x \* x；可以把匿名函数作为返回值返回：defbuild(x, y):returnlambda: x \* x + y \* y。

38、函数也是一个对象，而且函数对象可以被赋值给变量，所以，通过变量也能调用该函数。

39、在代码运行期间动态增加功能的方式，称之为“装饰器”（Decorator）。

40、正常的函数和变量名是公开的（public），如：a；特殊变量，可以被直接引用，但是有特殊用途，如：\_name\_；类似\_xxx和\_\_xxx这样的函数或变量就是非公开的（private）。private函数和变量“不应该”被直接引用，而不是“不能”被直接引用，是因为Python并没有一种方法可以完全限制访问private函数或变量，但是，从编程习惯上不应该引用private函数或变量。

41、结构：包（packet）-模块（.py文件）-函数。为了避免模块名冲突，Python又引入了按目录来组织模块的方法，称为包。类似的，可以有多级目录，组成多级层次的包结构。

注意：a、每一个包目录下面都会有一个\_\_init\_\_.py的文件，这个文件是必须存在的，否则，Python就把这个目录当成普通目录，而不是一个包。\_\_init\_\_.py可以是空文件，也可以有Python代码，因为\_\_init\_\_.py本身就是一个模块，而它的模块名就是mycompany。

b、自己创建模块时要注意命名，不能和Python自带的模块名称冲突。例如，系统自带了sys模块，自己的模块就不可命名为sys.py，否则将无法导入系统自带的sys模块。

42、为了编写可维护的代码，我们把很多函数分组，分别放到不同的文件里，这样，每个文件包含的代码就相对较少，很多编程语言都采用这种组织代码的方式。在Python中，一个.py文件就称之为一个模块（Module）。