教材：《Python学习手册第四版》

第一章没有看

Python的源码也可在PyCharm里面打，再run，也可以在PythonShell里面（小问题，调试，可以直接运行PythonShell看看）

通过PythonShell得到的反馈是搞清楚一段代码是做什么的的最快的方式。

文件里面要写print, Shell里面不用

Python不用分号

Python的缩进是有语法意义的，不能随便加一个空格之类的。

Shell里面要运行复合语句，请加一个空行（所以，文件里面就直接加吧……到时候复制过去也方便）

一个Python源代码也叫一个模块，也叫一个脚本。

注释用#开头

最后写一个input()当做getchar()把程序停下来。

第二章也没有仔细看

整个大程序由一个一个模块（文件）构成，模块里面是一个一个的语句，语句包含表达式，表达式建立并且处理对象。

数字运算：加减乘除，外加一个乘方\*\*

不用担心整数的溢出，Python自己处理（提升类型）

str(200)把200创建为一个字符串

len(‘a’)字符串的长度

文件包含的方式变成了import math(不加.py)

math.pi 显示pi

math.sqrt()调用函数

import random

random.random() 得到一个0到1之间的随机数

random.choice([元素的集合])从给定集合中产生一个随机数

字符串是包含单个字符的字符串的序列（Python没有字符了，只有字符串）

字符串支持下标操作，有效下标范围是[-len, len).会被转化为模len的正的余数。

字符串之间有加法和数乘（数乘表示重复）

字符串分片取出（对序列取出一段）

s[a:b]:先把ab转化为[0, len)之内的数，再取出范围[a, b)之内的元素。

ab的默认实参为0和len

一些特殊用法：s[1:]不指定末尾，就是默认end, 不指定前面，就是0.

字符串创建了就不能改变。但是Python会自动清理旧的对象

Python里面的多种类型共用的方法会以函数形式存在，但是私有方法会以成员函数的形式存在。

字符串的特有方法find(str a)返回完全匹配的下标，没找到就返回-1.（但是Python里面-1作为下标是有意义的，这就尴尬了）

find是返回下标，并不改变原字符串。

但是即使是那些会“改变”字符串的方法，其实也不过是新建了一个字符串用来保存结果，原来的字符串是不会被更改的。

replace(a, b)接受两个字符串参数，凡是出现a的地方用b代替，返回一个新的字符串。

dir(str)返回字符串的所有成员函数

help(str.replace)返回其帮助（有点像matlab）（）注意，replace不要加函数调用的括号。

字符串的格式化：

通用方法：

‘%s ’%(‘spam’)

新方法:

‘{0} {1}’.format(‘spam’, ‘2’)

列表是一个有序集合，字符串是一个列表。

列表集合里面的元素可以不同类型。

列表的增删：L.append()（尾后增加一个元素）L.pop(n)删除下标为n的元素

列表是可变的，不像字符串（字符串是不可变的）

M.sort():当列表只有一种类型的元素的时候可行（升序排列）。M.reverse()则没有要求。

列表支持M[2] = 3;(可以改变)

列表嵌套（同样没有类型要求，列表可以嵌套字典，字典可以嵌套列表……）

Python内置类型：字典。

{keyword:meaning, keyword:meaning , keyword:meaning }通过关键词映射到“释义”

把字典里面的按照键的升序打印出来：

Ks = list(D.keys())

Ks.sort()

>>> Ks

['a', 'b', 'c']

>>> for key in Ks:

print(key, '=>', D[key])

（实际上不是把字典变动了，而是把关键字拿出来做了一个列表，把列表排序，再按照列表去找值）。

元组：不可变的序列

多态是使用Python的关键思想。（不用声明类型）

第五章：

Python的基本数字：

整数

浮点数3.12, 3.1e10,

八进制0o12, 十六进制0x11, 二进制0b111

hex(), oct(), bin()进行进制之间的转换（结果是字符串模式）

复数a + bj(b不可省略)

三目运算符x if y else z: x?y:z

x in y返回bool x not in y

Python没有递增运算符x++

整数除法是// 3//5 == 0 3/5=0.6

x < y < z就是 x < y and y < z(Python没有&&，只有and or not)

除法：为了兼容性问题，对于截尾的除法，总是用//，对于真除法，总是把一个运算对象变成浮点数。

Python自动支持大整数。

int(‘1234’, 8)把前面的字符串以8进制读取成整数

也可以int(2.2)

集合操作交并补差 a | b(并) a & b(交) a – b （差集）

x > y:x包含y吗？

Python的变量类型是在运行时自动决定的（动态类型）

Python里面，变量名和对象是完全不同的概念，a = 3, 先创建一个对象来表示3， 然后，如果变量名a没有被创建过，就创建一个新的。然后，变量名a变成3这个对象的引用。a可以引用任意对象。

对象除了存储值，还要存储其类型（类型属于对象，而不属于变量），还有一个引用的计数器，当计数器是0时，就可以被回收了。

PythonShell里面≫>3,创建对象‘3’，但是引用计数器为0，意味着这块内存可以被回收了（事实上没有方法可以再访问这块内存了，所以回收。）

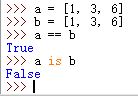
对变量赋值实际上是让变量引用另外一个对象。

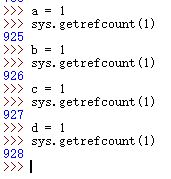
a = 3

a += 1;创建4， 使得a引用4,3被回收。

变量之间赋值是共享引用。

所以，为了实现按值拷贝，不能直接等于（Python里面的变量名就是一个引用，创建一个引用等于另一个引用，并不会开辟新的对象）

用is来比较两个变量名是在比较它们引用的是不是同样的对象，用==比较的是值是否相等。

但是小整数和小字符串等被Python系统缓存复用了，，对于1这样的常用的量，系统并不会创建多个“1”对象。

字符串：可以用单引号/双引号，但是必须配对（”2’是不行的）（有两种版本是为了可以直接输入引号（想输出单引号就用双引号括起来…………这优化真是奇葩了））

那我既要用单引号又要用双引号怎么办？那就用连续的三个单引号/双引号括起来吧，毕竟你的正常文本里面不会出现连续三个双引号（三重引号还可以跨行。）（三重引号还可以被用来在调试的时候注释掉某些代码）也可以Ctrl+/进行块注释（PyCharmIDE支持）

在输出的时候，Python会把非打印字符写成\0xaa的格式

转移序列只会转义一次， [\\n](file:///\\n)并不会被误认为是换行。

写文件名的时候常常会用到原生字符串r”C:\user\……”

分片操作的第三个可选参数：

X[i:j:k], k 默认为1， 是取样的步长。

k是正数的时候要求有效范围I < j(转化为[0, len)之后)

k是负数的时候要求I > j,得到的依然是[I, j), 不过这个区间I > j……

最常用的操作：x[∷-1]反转字符串， 而x[:-1]常用来丢弃最后的换行符。

单个字符字符串和ASCII码转换：chr(int) ord(‘a’)

由于字符串是不可改变的，经常要“修改”字符串会经常创建新的字符串。当字符串特别大的时候，这种工作效率很低。这个时候可以把字符串打散成列表，修改完再拼接成字符串。

S = ‘’.join(L),在L的每两个元素之间插上’’（也就是拼接起来）

拆分：S.split(默认为空白)， 把S中的（）当做分隔符，分割成一个列表。

格式化字符串：”%a%b……”%(a, b)支持大部分printf的格式化支持。

也可以s = “%a%b”, v = (a, b) print(s%v)

当%后面是字典的时候，前面的打印%(可以加关键字)

格式化方法并没有仔细看。

大体应该是’{name}{1}{3}{2}’.format(1, 2, 3, 4, name = ‘hello’)format里面的东西，先写被数字索引的，再写被名字索引的。