17 Aug 2022:

今天学习时间4为小时

继续学习了python基础课的8,9课程。完成了作业，笔记整理了一半。作业和笔记已经上传到了github。

其余相关工作：安装了实验室电脑的相关环境，重装了系统（ubuntu）

以下为学习内容摘要：

文件的读、写

r, w, x, a…

r+ w+ ...

文件的解析

utf-8, gbk

异常处理的方法： try except exception as else

模块的简介： import &from import as

列表的存储规则——存储地址

列表的别名、浅拷贝、深拷贝：所有内容重新拷贝至全新的地址，修改后对原始数据无任何影响

当存在一些可变类型如列表，字典等，用深拷贝更加安全

列表，元组，字典的区别

在列表中的元素：

列表、字典：地址可以不变，内容可以变换

而元组、数字、字符串：自身发生改变，地址也发生了改变

元组重新赋值的实质是：开启一个全新的内存地址负值元组并对其进行修改

字典查找速度极快

查看运行速度

import time

start = time.time()

代码

end = time.time()

print(round(end – start))

字典通过稀疏数组存储

通过内存空间换取速度，速度快但占用大量空间

原因：最大化存储空间，防止哈西值冲突

如果字典内部不存在该输入数据的散列值即没有该元素，会报错。

自动创建散列表，存储数据时查找散列值对应的散列表。

id()查找地址

列表通过地址查找存储数据

任意元素类型，位置空间，只保存地址

访问列表元素时会从头遍历寻找，因此耗时长效率低

18 Aug 2022:

学习时间4小时，具体内容如下：

完成了python第九章的作业，整理了笔记并上传至github

学习了第10章的time库：计时功能，暂停功能以及时间的格式化输出。

时间库：import time

当前时间：time.localtime()

time.time()返回纪元以来的秒数记录sleep

time.perf\_counter()选取随意一个时间点记录现在到此时间点的间隔秒数，记录sleep

time.process\_time()选取随意一个时间点记录现在到此时间点的间隔秒数，不记录sleep

perf\_counter()的精确度相比time.time 更高

记录时间差可以获取运行时间，一般第一种计时精度已经足够因此常用

第三种方法会去掉sleep的运行时间

格式化输出time.strftime(“%Y-%m-%d %H:%M:%S”, time.localtime())#年月日星期时分秒

暂停运行：time.sleep()

random库

用于生成随机数

通过random生成各种伪随机数，基本可以满足大多数的应用

需要给定一个随机种子，如果没有默认为当前系统时间

相同的种子会产生相同的随机数，但因为每时每刻的时间不同因此随机种子不一样所以生成的数字为随机的

seed()

random.random()

随机整数

randint (a,b)[a,b]之间的随机整数

randranget(a)[0,a)之间的随机整数 randrange(a,b,step)[a,b)之间以step为步长的随机整数

random()产生0到1之间的任意浮点数

uniform(a,b)——产生一个[a,b]之间的随机浮点数

序列用函数

choice(seq)——从序列类型中随机返回一个元素

choice(seq，weight，k)#序列，权重，个数，可以重复选择，选取后放回

shuffle(seq——经序列中的元素随机排列返回打乱后的序列)

sample(pop,k)——从pop类型中随机选取k个元素以列表返回,选取个数不能超过pop内元素

gauss(mean,std)——生产一个符合高斯分布的随机数均值

元组性质，索引;解包赋值

画图：

import matplotlib.pyplot as plt

res = [gauss(0,1), for i in range(10000)]

plt.hist(res, bin=1000)#正太分布直方图的绘制

plt.show()

例题：模拟扑克

random.shuffer()打乱顺序

random.choice()随机抽取

random.Counter()：计数器工具，实质是一个字典的子类，在一个列表中统计每个元素出现的频率

最常见的统计——. most\_common(n) 提供n个频率最高的元素以及计数

元素的展开——.elements()元素全部展开

list(cnt\_str.elements())

print(isinstance(random.Counter(),dict))

True

Counter的加减

c = Counter(a=3, b=1)

d = Counter(a=1, b=2)

c+d

Counter({‘a’:4, ’b’:3})

deque:双向队列，高效在两端进行删改

19 Aug 2022:

学习时间4小时，具体内容如下:

itertools库：排列组合

product()迪卡尔积——笛卡尔积

permutations——排列

combinations——组合

combinations\_with\_replacement——元素可重复组合

zip——拉链（按位置进行组合）

ziplongest——长拉链（确实位置用none代替）

count——计数

cycle——循环

repeat——重复

chain——锁链

enumerate——枚举

groupby——分组

numpy库，numpy数组：（高效）

由c语言编写，为编译型语言

numpy的广播运算方法：

可以实现多线程处理

