本周继续学习了python，python的基本的知识点到目前为止都已经过了一遍，能看懂一些经典的例题的代码，但是在手动自己编程方面的能力还是有待提高。

**特殊方法、属性和迭代器**

1.构造方法：这是面向对象语言所共有的，可能要为自己写的每个类实现构造方法，名为init的构造方法在对象被创建后即自动调用

2.重写：一个类可以通过实现方法来重写其超类中定义的这些方法和属性，这时可以调用超类的方法或者使用super函数

3.序列和映射：如果需要创建自己的序列或者映射，那么需要实现所有序列或者映射规则的方法，包括getitem和setitem的特殊方法

4.迭代器：迭代器是带有\_\_next\_\_方法的简单对象，能在一系列的值上进行迭代，如果没有值可供迭代则会引发StopIteration异常

5.可迭代对象：可迭代对象有一个返回迭代器的\_\_iter\_\_方法，能像序列一样在循环中使用；一般来说迭代器本身也是可迭代的

6.生成器：生成器是包含yield语句的函数（或方法），不需要明确定义\_\_iter\_\_和\_\_next\_\_方法

新学的函数：

iter(obj) 从可迭代对象得到迭代器

property(fget,fset,fdel,doc) 返回一个属性，所有参数可选

super(class, obj) 返回一个类的超类的绑定实例

**常用标准库**

1.模块：模块即为子程序，其主函数用于定义函数和类；若函数包含测试代码，则应将其放置在检查\_\_name\_\_==‘\_\_main\_\_’是否为真的if语句中；能够在PYTHONPATH中找到的模块都可以导入，语句import foo可以导入存储在foo.py中的所有模块

包：包是包含其它模块的模块，是作为包含\_\_init\_\_.py文件的目录来实现的

2.模块的细节：可以通过dir、检查\_\_all\_\_变量以及使用help函数来查看模块的细节

3.Python中包含了一些标准库，本章中涉及到的有

sys：访问多个和Python解释器联系紧密的变量和函数

dir(obj)：返回按字母顺序排序的属性名称列表

help([obj]) ：提供特定对象的交互式帮助信息

**文件和流**

1.类文件对象：类文件对象是支持read、readline、readlines、write和writelines方法的对象

2.打开和关闭文件：提供一个文件名，可以使用open函数打开一个文件；如果希望确保文件被正常关闭，则可以使用with语句

3.模式和文件类型：打开文件时可以提供模式信息，’r’代表读模式，’w’代表写模式，也可以将文件以二进制文件形式打开

4.读和写：使用read或是write方法可以对文件对象进行读写操作

5.读写行：使用readline和readlines方法可以从文件中读取行，使用writelines语句则可以写入一行或多行数据

6.迭代文件内容：有很多方式可以迭代文件的内容。如果是迭代文本中的行，可以通过迭代文件对象本身来完成

所学新函数：

open(file, mode='r', buffering=-1, encoding=None, errors=None, newline=None, closefd=True, opener=None)打开文件file并返回文件对象

**Python在数据分析中的应用**

1.用于数据分析的Python库简介：介绍了NumPy、pandas两个重要的数据分析Python库

2.pandas的重要数据结构：介绍了pandas的两个重要数据结构DataFrame。

3.连接两个DataFrame：merge函数可以将不同 DataFrame 中的行连接起来。pandas会根据列名的重叠情况推断出哪些列是连接键。

4.透视表：pivot\_table方法

5.查看分组大小：使用 GroupBy 对象（不论是 DataFrameGroupBy 还是 SeriesGroupBy）的.size() 方法查看分组大小