简短的回顾：

1. Python初探

#!/usr/bin/python：python的指向地址，表示以python的脚本运行。

uft-8：表示编码格式

if ….\_main\_:是python的入口

单一职责原则：defxx函数定义的代码块，通过return

1. 标准数据类型

Number: int float bool；运算符

String

List:一系列有序的值，对于列表的循环——for …in…

Tuple:不可变形式的List。

Dictionary: key是唯一的，通过key来查看value。

1. 代码运用：

异常处理

类的使用：将类似的一些具有相似属性的东西归在一个类里，避免代码重复。

1. 命令行以及工具等等

库的使用：使用其他人的代码（画乌龟的代码，日历，时间的代码）

（引入别人的模块，以及预设好的模块）

个人总结：

之所以选择这个课是因为自己是个拖延症晚期患者，而且一直有想学编程的心。而且价格的确很便宜，成年人什么事情都要靠自己完成，别人督促为辅。

因为没有脱产，而且平日的学习工作很忙，很多时候都是中午和晚上完成的作业，也有11点50几打卡的惊险时刻。对于一个强迫症，打卡中断是非常令人难过的事情。所幸，在大作业任务布置完成之前，除了两次不需要打卡的日子，基本每天都会在24点之前打卡。算是对自己的一个交代吧。

上大学的时候学过C++，但是学的很烂，还记得教材的名字叫面向对象的程序设计，这个书名一直被我们笑来着，也没有真正理解什么叫面向对象，而面向对象的相反地方是面向过程，那为什么面向对象更好了。学习python其实更看重的是一个思维过程吧，而不是具体的代码编写。作为一个非CS专业的学生，我对自己的要求很简单，一个是把最基础的操作搞懂能够顺利输出；另一个是复杂的代码能够知道输出的结果大概是什么养的。简而言之，就是会写简单的，会看复杂的。

特别想提的是大作业的写作过程！！！因为没有假期，所以没有太多时间搞这个。一直纠结于sorted排序这一块，后来pull request了大家的作业，发现很多人都写得非常非常好（只能自我安慰他们有基础），其中看了一位liluo同学的代码（真的是很高级，用了class还有Init代码什么的），受了些启发，终于照猫画虎地写出了排序。也反映了我的一个问题：一直对于代码块之间的变量传递非常不理解！所以导致了大作业的懵懵懂懂。不过还好，最后总算凭借自己误打误撞写出了排序，也是很开心。

目前来看，28天的S1我会给自己的课程收获打分70分吧，毕竟很多东西还是浮于表面的，如果没有更深刻的练习和实践，我想这20来天的内容会逐渐遗忘在时间中的。希望S2过程自己能够坚持下去。