# 第一周：熟悉Python

要积极探索Python的使用方法，尽可能多的完成下面这些任务

第一天：基本概念(4小时)：print，变量，输入，条件语句

第二天：基本概念(5小时)：列表，for循环，while循环，函数，导入模块

第三天：简单编程问题(5小时)：交换两个变量值，将摄氏温度转换为华氏温度，求数字中各位数之和，判断某数是否为素数，生成随机数，删除列表中重复项等等

第四天：中级编程问题(6小时)：反转一个字符串(回文检测)，计算最大公约数，合并两个有序数组，猜数字游戏，计算年龄等等

第五天：数据结构(6小时)：栈，队列，字典，元组，树，链表

第六天：面向对象编程(OOP)(6小时)：对象，类，方法和构造函数，面向对象编程之继承

第七天：算法(6小时)：搜索(线性和二分查找)、排序(冒泡排序、选择排序)、递归函数(阶乘、斐波那契数列)、时间复杂度(线性、二次和常量)

# 第二周：开始软件开发(构建项目)

第一天：熟悉一种IDE(5小时)：IDE是你在编写大型项目时的操作环境，所以你需要精通一个IDE。在软件开发的初期，我建议你在VS code中安装Python扩展或使用Jupyternotebook。

第二天：Gubhit(6小时)：探索Github，并创建一个代码仓库。尝试提交(Commit)、查看变更(Diff)和上推(Push)你的代码。另外，还要学习如何利用分支工作，如何合并(merge)不同分支以及如何在一个项目中创建拉取请求(pull request)。

第三天：第一个项目-简单计算器(4小时)：熟悉Tkinter，创建一个简单的计算器

第四、五、六天：个人项目(每天5小时)：选定一个项目并完成它

第七天：托管项目(5小时)：学习使用服务器和hosting服务来托管你的项目。创建一个Heroku设置并部署你构建的应用程序。

# 第三周：让自己成为一名程序员

第3周的目标是熟悉软件开发的整体过程。你不需要掌握所有的知识，但是你应该知道一些常识，因为它们会影响你的日常工作。

第一天：数据库基础(6小时)：基本SQL查询(创建表、选择、Where查询、更新)、SQL函数(Avg、Max、Count)、关系数据库(规范化)、内连接、外连接等

第二天：使用Python数据库(5小时)：利用一种数据库框架(SQLite或panda)，连接到一个数据库，在多个表中创建并插入数据，再从表中读取数据。

第三天：API(5小时)：如何调用API。学习JSON、微服务(micro-service)以及表现层应用程序转换应用程序接口(Rest API)。

第四天：Numpy(4小时)：熟悉Numpy并练习前30个Numpy习题。

第五、六天：作品集网站(一天5小时)：学习Django，使用Django构建一个作品集网站，也要了解Flask框架。

第七天：单元测试、日志、调试(5小时)：学习单元测试(PyTest)，如何设置和查看日志，以及使用断点测试。

# 第四周：认真考虑工作(实习)问题

第一天：准备简历(5小时)：制一份一页的简历。把你的技能总结放在最上面，必须在写项目的同时附上Github链接。

第二天：作品集网站(6小时)：写几个博客，将它们添加到你之前开发的作品集网站中。

第三天：LinkedIn简介(4小时)：创建一个LinkedIn个人简介，把简历上的所有内容都放到LinkedIn上。

第四天：面试准备(7小时)：准备一些谷歌常见的面试问题，练习白皮书中的10个面试编程问题。在Glassdoor、Careercup等网站中查看前人遇到的面试问题。

第五天：社交(-小时)：走出房门，开始参加聚会、招聘会，与其他开发人员和招聘人员见面。

第六天：工作申请(-小时)：搜索[Python Job]，查看LinkedIn Job和本地求职网站。选择3个工作岗位并发送工作申请。为每个工作指定你的简历。在每个工作要求中找出2到3件你不知道的事情，并在接下来的3-4天里学会它们。

第七天：在拒绝中学习(-小时)：每次你被拒绝的时候，找出两件为了获得这份工作你应该知道的事情，然后花4-5天的时间来掌握它们。这样，每次拒绝都会让你成为更好的开发人员。