

Python 编程基础教学计划

田雅夫

July 16, 2014

1 教学目标

1.1 知识性目标

1. 了解 Python
2. Python 基本开发环境的构建
3. 基础语法, 操作.
4. 列表推导和简单的 λ 演算
5. 字符串, 元组和字典
6. OOP 编程, 类, 对象和方法.
7. 文件操作
8. 调试的技巧
9. 简单实用的标准库操作
10. 简单的 GUI 构建
11. 串口库的使用
12. 建立自己的工程

1.2 能力性目标

1. 良好代码缩进的能力
2. 良好的命名风格
3. 良好的注释习惯
4. 代码抽象, 封装的能力
5. 根据需求编写代码的能力
6. 代码调试, 改正的能力

2 教学地点, 时数, 条件

2.1 地点

21B058 教室

2.2 学时数

大概有 10 个晚上, 每个晚上 2-3 个课时. 期望有 25 个课时.

2.3 教学条件

面向的成员以大一的居多, 兼有部分大二的学生. 大一新生因为刚考完 C 语言的因素对其还比较熟悉, 大二学生的程度则不好掌控. 同时经观察发现这些成员在变量命名, 代码缩进 (因为 Python 语言特性所以缩进特别重要), 代码封装方面的习惯尚有很大的改进空间.

2.4 需要的条件

1. 需要网络以保证学员可以即时参考资料
2. 一本教材/手册/学案 (我打算自己写一本)
3. 一套可供学员即时提交代码并展示的系统 (远程桌面?)

3 教学方法

用尽可能精简的方式讲解, 尽可能多的练习.
每次的授课预期分成下面几个方面:

3.1 讲解部分

每次的讲解时间不会长于一课时.(我争取...)

讲解的内容尽可能少的实操的原理, 而是力图讲清楚每一个命令是怎么用的, 是干什么用的.

对于教材的选择我认为应尽量精简, 只给他们可能用到的部分. 这种教材的作用与技术手册是一致的, 在后面操作环节中学员将依托手册中的内容编制程序.

3.2 操作部分

我会写足够的题目, 就以需求的方式给出. 确保每个学员拿到的需求都是不相同的, 根据这些需求让他们现场编制程序. 在这个过程中他们可以参考所有的资料 (包括网络). 但不可以相互交流. 提问是允许的.

对于能力较强的学员, 可以下发其他人的需求文档让其多编写一些程序.

3.3 讲解部分

让每一个人上台讲解其刚才编制的程序, 讲解时间尽量短, 尽量精炼. 在这个过程中其他学员对自己程序的修改是禁止的以保证其认真听讲解. 对完成任务的学员应有奖励措施 (完成的多的学员如果是男生可送一个月迅雷会员).

对没有完成需求的学员要求其讲明在那个位置出错或 bug 可能出现的位置 (讲了调试技巧以后要求定位出错位置). 如果学员成功定位出错位置则给予讲解 (鼓励其他学员上来 debug).

对于当天没有实现需求的学员要求在第二天上课前改正并检查.

3.4 作业

再给每个学员一个需求文档让其编制程序. 对于作业的检查可以采用互评的方法 (这里我也不大明白要怎么办, 总之让他们尽可能多的写代码, 读代码与 debug)

4 每个模块的教学任务

4.1 了解 Python

1. Python 与 C 的差别
2. 解释器语言的特点
3. 弱类型语言要注意的事项
4. 良好的代码风格

4.2 Python 基本开发环境的构建

本次教学拟使用 *windows+Python2.7+IDLE* 的工具链. 同时演示环境为 *OpenSUSE+Python2.7+Vim*.

之所以使用 Windows 而不是 Linux 是要贴合绝大多数成员的使用习惯. 同时应任昕昞强烈要求使用 OpenSUSE 这个发行版作为演示环境. 用 Vim 作为演示环境同样来自他的建议.

对于 IDE 的选择我认为 IDLE 就已经足够了. 首先它是 Python 自带的 IDE, 而且干净轻量实用, 不需要过多的配置. Ipython 可能在 Shell 命令的支持方面优于 Python 但是安装起来实在是太麻烦. Vim 由于学习成本的考量此次不采用.

4.3 基础语法和操作

本次教学拟讲解如下几个方面:

1. 交互模式与脚本模式
2. 格式输入与输出
3. 变量类型与类型转换
4. 操作符与表达式

5. 定义函数
6. 条件和递归
7. while 循环

4.4 列表推导和简单的 λ 演算

本次教学拟讲解如下几个方面:

1. 列表类型
2. 列表操作 (映射, 遍历, 拼接)
3. for 循环语句
4. 列表方法 (in,append, 切片,extend,sum,pop,del,remove)
5. 列表与字符串的转化
6. 简单的列表推导 (map,reduce,filter)

4.5 字符串, 元组和字典

这一块简单讲讲即可:

1. 可变类型与不可变类型
2. 字符串方法
3. 简单的正则表达式 (这块就是拓展...)
4. 字典的使用
5. 元组的使用

4.6 OOP 编程, 类, 对象和方法

讲完 Python 的四大结构以后 OOP 编程的思想就非常显然了. 因为列表, 串, 元组和字典都是类

1. 类和对象的简介
2. 定义类和构造对象
3. 使用内置方法
4. 自定义方法
5. 简单数据结构的 OOP 实现

4.7 文件的操作

这一块我认为不用讲的很深入, 但是应该辅以大量的练习. 因为对文件的存取操作会很频繁的出现以后的工作中.

要讲的部分:

1. 打开/关闭/创建文本文件
2. 读/写文本文件
3. `os.path` 模块解析路径/获取文件信息
4. `linecache` 模块按行读取文件
5. `tempfile` 模块建立临时文件
6. 其他文件操作 (删除, 重命名, 创建多级目录/目录树)
7. 遍历目录

4.8 调试的技巧

在这一阶段所有的学员应该都有一定的编码经验而且犯过一定量的错误了, 所以这个时候讲一点有关于调试的小技巧应该是非常适合的. 如果将这一模块提前可能学员不会有足够的编程经验来理解这些技巧. 不讲这些模块的话后面相对复杂的部分实现起来可能会有困难. 这一块的学习应该仍以时间为主, 也就是不断地定位错误及改错. 要讲的部分:

1. `print` 调试
2. `reload` 调试
3. `try/catch` 抛出异常
4. 设置断点

4.9 简单实用的标准库操作

在前面其实已经用到一部分标准库的内容了, 在这里介绍其他一部分可能用得到的:

1. 文本处理
2. 数组, 列表, 堆
3. 系统时间, 代码执行时间
4. 简单的网络访问
5. 其他需要的或者有意思而且简单的库函数

4.9.1 简单的 GUI 构建

这个部分拟简单的讲解一下 `Thinker` 的操作和使用. 不涉及自定义控件. 如果时间及其重组的话会介绍一下 `QT`(仅仅是介绍一下).

4.10 串口库的使用

这个部分主要介绍一下如何使用 serial 模块实现串口通信, 联系一下用电脑当做上位机控制单片机工作.

4.11 建立自己的工程

在这个部分拟设计一个比较复杂的情境, 让学员自己写一个程序控制单片机实现比较复杂的任务.

条件允许的话将该程序用 GUI 封装, 留出 GUI. 让学员自为自己的程序编写文档和说明.