Python语言学习

**学习方式主要是对比C/C++的语法快速学习并联想记忆**

1. 培养了写注释的习惯。
2. 用Python解决了两个动态规划题（目的是在实践的过程中学习语法）：
   1. [P1048 [NOIP2005 普及组] 采药 - 洛谷 | 计算机科学教育新生态](https://www.luogu.com.cn/problem/P1048)
   2. [P1455 搭配购买 - 洛谷 | 计算机科学教育新生态](https://www.luogu.com.cn/problem/P1455)

（其中第二题因为Python运行速度比较慢，没有通过所有样例）

（用C++同样思路写了个对比程序）

1. 通过这两个题学习到了dict数据结构，结合此次实践，我学习了Python 100 Days 中第7节（字符串和常用数据结构）

字符串（str）、

列表（list）、

元组（tuple）、

集合（set）、

字典（dict）。

我认为这分别对应C++标准库中的

字符串（string）、

向量（vector）、

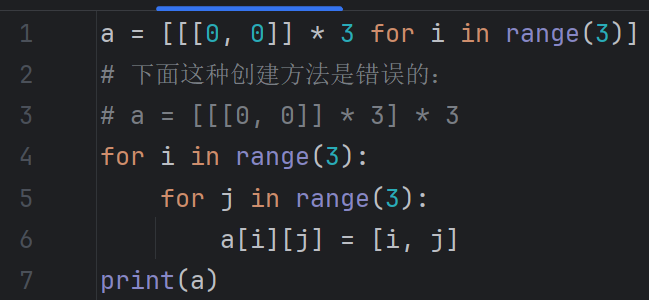
元组（pair, tuple）、

集合（set）、

映射（map）。

一些有趣的细节：

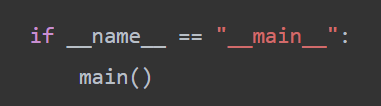
1. Python中的元组中的元素不能被改变，而C++的可以。不能因此认为元组用处不大。事实上，元组在创建时间和占用的空间上面都优于列表。并且在多线程环境中，不能被修改的元组可以减少许多不必要的程序错误。
2. 了解到for循环不止可以in range()这种方法使用，还可以直接遍历整个容器。
3. 可以使用in和not in来判断一个字符串是否包含另外一个字符串，还可以使用切片运算（这一点C++中是没有的）
4. 通过查询资料，了解到list是顺序存储，与C++中的list的链式存储不同。这表明查询和修改的时间复杂度都是O(1)，而插入和删除等都是O(n)。
5. Python中的set比C++中的set更为强大，支持 &、|、-、^等运算。
6. “is”与“==”其实是有区别的，is是单纯比较对象地址是否相同，而==就是比较对象的内容是否相同。而Python与Java相同，都有一个常量池（Python的常量池是[-5, 256]，Java是[-128, 127]）所以Python中 -6 is -6的值是False。
7. 当我们创建二维链表时，my\_list = [[0] \* 3] \* 3这种创建方法是错误的，它相当于把同一个引用复制了3次，我们本以为的3行其实都是引用的同一个[0] \* 3。



正确输出：[[[0, 0], [0, 1], [0, 2]], [[1, 0], [1, 1], [1, 2]], [[2, 0], [2, 1], [2, 2]]]

错误输出：[[[2, 0], [2, 1], [2, 2]], [[2, 0], [2, 1], [2, 2]], [[2, 0], [2, 1], [2, 2]]]

1. Python也是可以写main函数的，写完之后要通过以下代码实现。如果这个文件直接被运行，那么就可以走以下语句运行main()，反之不走main()。



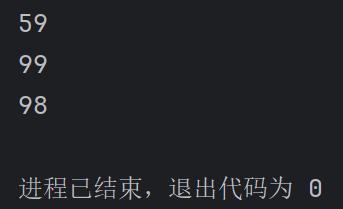
通过这一点，我了解到不同于C/C++，Python可以从任何地方开始执行。

1. 学习面向对象编程基础。了解了封装，继承与多态，跟随教程打了几个Demo试了一下Python的语法。不同于Java的public, protected, private，Python对于对象成员的访问权限只有两种，公开和私有，如果想要将对象的成员设置为私有，需要在给属性或者方法的名称前打“\_\_”（两个下划线）。（另外，以下划线结尾的成员并不是私有）

但是Python并没有从语法上严格保证私有成员的私密性。所以通过特殊的语法还是可以访问到“私有”的成员。例如下面的代码，虽然我在名称前打了两个下划线，但是仍然可以通过特殊语法访问成员（尽管PyCharm警告我访问不到）。



输出结果：



机器学习

1. 了解了机器学习的概念，认识了监督学习与非监督学习。
2. 监督学习需要人工手动打标签，来辅助机器学习出自己想要的模型。
3. 非监督学习是让机器自己探索数据中的特征，并进行分类（让机器自己寻找数据中的规律）。
4. 下载并安装了Jupyter Notebook。

未来

1. Python语言相关：学习面向对象进阶，异常处理，进程和线程
2. 机器学习：数学内容（线性回归模型，代价函数等）
3. CV：Pytorch，Numpy了解并探究如何使用