# **创意编程班第二周周末复盘总结**



## 复盘作业

复盘：1）整理目前学到的东西 2）整理学习所犯的错误以及当时的思维过程 3）整理对今后学习有指导意义的方法；

老规矩，我给一个模版

## 1.本周学习到的东西

每学到一个东西，就像你的工具集之中多了一件工具，以后你一定会使用到的，试着用自己的话说给组队的小伙伴听，你要如何让他听懂呢？

1. **字典**

**字典的元素是key:value的组合对。其中key类似于是一个标签，相当于list的默认编号一样，所以是不可以重复的，但优点是key的表示形式是可以自己定义的，比list灵活许多。然后每一个key都对应一个value，可以通过key来使用value。**

1. **Don’t Repeat Yourself.**

**说起这个Dry原则，看了蜀黍两天的代码对比真的是醍醐灌顶，前一天的那个真的是长而繁琐。乱。第二天改进之后真的是看起来炒鸡清爽~~~之前自己写的时候也觉得是很繁琐，但没有想着去做一些改进，看来以后写代码还是应该自己多想想，想一想能不能做到更简洁明快。**

1. **元组学习**

**元组和list一样都是列表，区别在于元组是【不可变】的，也就是说元组没有list的某些性质，比如说修改、删除、添加之类的会改变原来列表的操作。所以感觉元组可能主要用来提供一些数据，比如性别分为【男】【女】，放在一个元组里面，就不应该发生任何元素的改变，改变了就不合理了啊。**

1. **异常处理**

**异常处理我感觉理解不是很透彻。现在能有些印象的就是‘在程序有一些不当的操作时候，异常处理可以让程序运行下去’。比如昨天的作业里面，需要自己做的那个，我就是在一个元组的操作里面用到了异常处理，因为元组不能修改，一旦修改报错，程序就会终止，但是如果将修改的部分放在try里面，则程序可以正常运行。所以，我的问题就是，这个例子里面，我知道这个是错的，我才放try来试试。那么正常情况下，我怎么知道try应该用在哪些地方呢？**

[1,2,”3”,”aaa”]

{“aa”:”bb”, “cc”:”dd”}

(1,2,3)

以上这三个数据结构有何异同？什么场景时候使用，说出自己的理解

**见上（1）（3）**

try/except

这个编程语法存在的价值？说出你的理解

**见上（4）**

“Don’t repeat yourself” 原则是什么

**---在一个程序中，如果相同的东西出现了三次及以上，可以考虑修改了；重复次数多的字符，可以统一定义为一个字符变量；重复次数多的操作，可以定义为一个函数（方法），然后调用即可。**

## 2.本周编程中所犯的错误以及思维过程

1）我在写list的时候，想要把一些数字放入一个空list中，当时是我想直接赋值比如list[2] = 3，但实际上此时由于list的初始定义是空的，所以并不存在list[2]，所以说这个时候只能用list.append(2)方法来做。一个个加进去。List只能索引存在的序号。

2）我在写字典的时候，将key和value用=连接起来，实际上应该是用冒号连接，如key:value

3）在写代码的过程中，还是发生了一次缩进问题。但是现在好的就是一看那个报错提示，基本都知道问题了，也能改过来

4）有一次定义函数的时候，不知道被什么影响，写成了def output = func(argv)，结果很尴尬，Python并不需要写output的。对，是被MATLAB影响了。Python是只需要return一下就OK了，return的内容就是整个函数的output。

---

