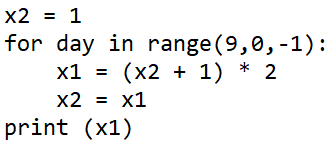
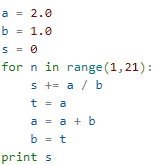
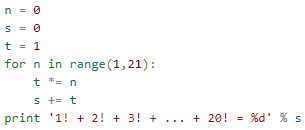
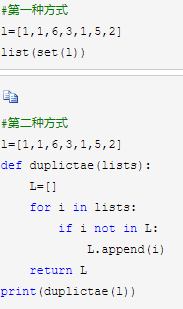
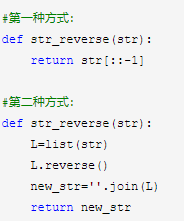
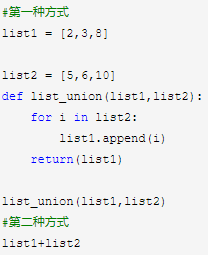
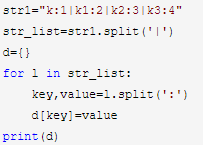
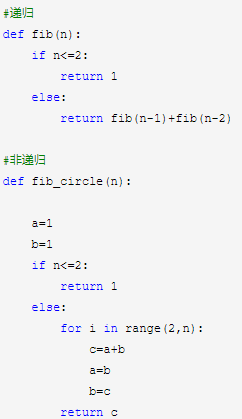
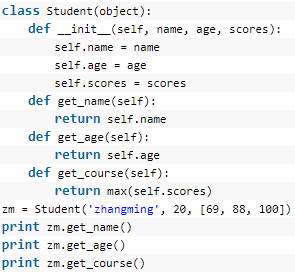
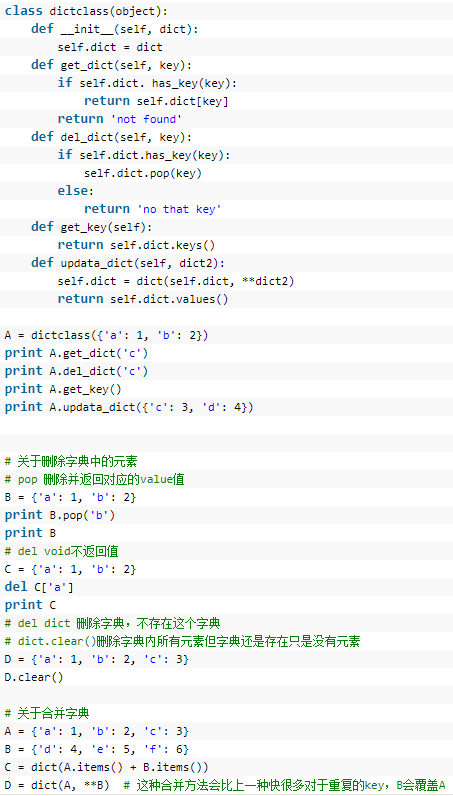
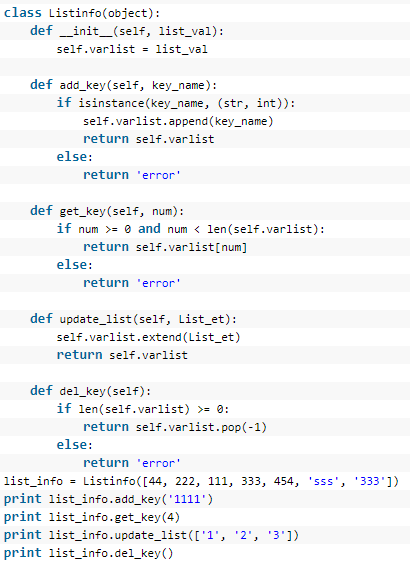
1. 猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。
2. 
3. 有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13...求出这个数列的前20项之和。
4. 
5. 求1+2!+3!+...+20!的和。
6. 
7. 删除列表中的重复元素
8. 
9. 定义函数实现字符串反转 例如：输入str="string"输出'gnirts'
10. 
11. 一行代码实现对列表a中的偶数位置的元素进行加3后求和
12. 
13. List = [-2, 1, 3, -6]，如何实现以绝对值大小从小到大将 List 中内容排序
14. 
15. 合并两个list
16. 
17. 什么是lambda函数？它有什么好处？另外python在函数编程方面提供了些什么函数和语法？
18. 
19. 详细说说tuple、list、dict的用法，它们的特点
20. 
21. list 对象 alist [{'name':'a','age':20},{'name':'b','age':30},{'name':'c','age':25}]， 请按 alist 中元素的age 由大到小排序；
22. 
23. 将字符串："k:1|k1:2|k2:3|k3:4"，处理成 python 字典：{'k':'1', 'k1':'2', 'k2':'3','k3':'4' }
24. 
25. 用代码体现斐波那契数列
26. 
27. 定义一个学生类。有下面的类属性：
    1. 1 姓名
    2. 2 年龄
    3. 3 成绩（语文，数学，英语)[每课成绩的类型为整数]
    4. 类方法：
    5. 1 获取学生的姓名：get\_name() 返回类型:str
    6. 2 获取学生的年龄：get\_age() 返回类型:int
    7. 3 返回3门科目中最高的分数。get\_course() 返回类型:int
    8. 写好类以后，可以定义2个同学测试下:
    9. zm = Student('zhangming',20,[69,88,100])
    10. 返回结果：
    11. zhangming
    12. 20
    13. 100



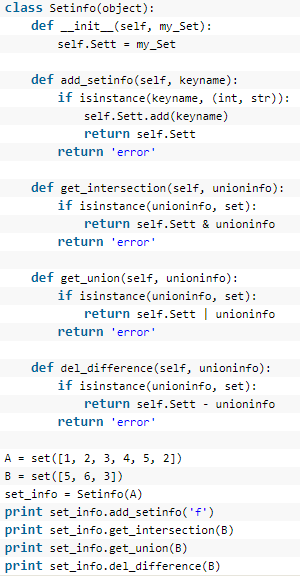
1. 定义一个字典类：dictclass。完成下面的功能：
   1. dict = dictclass({你需要操作的字典对象})
   2. 1 删除某个key
   3. del\_dict(key)
   4. 2 判断某个键是否在字典里，如果在返回键对应的值，不存在则返回"not found"
   5. get\_dict(key)
   6. 3 返回键组成的列表：返回类型;(list)
   7. get\_key()
   8. 4 合并字典，并且返回合并后字典的values组成的列表。返回类型:(list)
   9. update\_dict({要合并的字典})



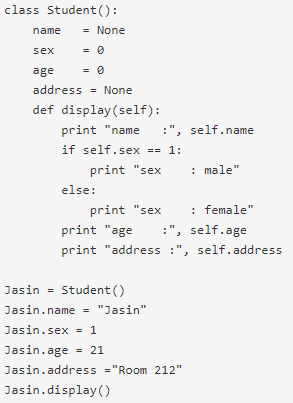
1. 定义一个列表的操作类：Listinfo
   1. 包括的方法:
   2. 1 列表元素添加: add\_key(keyname) [keyname:字符串或者整数类型]
   3. 2 列表元素取值：get\_key(num) [num:整数类型]
   4. 3 列表合并：update\_list(list) [list:列表类型]
   5. 4 删除并且返回最后一个元素：del\_key()
   6. list\_info = Listinfo([44,222,111,333,454,'sss','333'])



1. 定义一个集合的操作类：Setinfo
   1. 包括的方法:
   2. 1 集合元素添加: add\_setinfo(keyname) [keyname:字符串或者整数类型]
   3. 2 集合的交集：get\_intersection(unioninfo) [unioninfo :集合类型]
   4. 3 集合的并集： get\_union(unioninfo)[unioninfo :集合类型]
   5. 4 集合的差集：del\_difference(unioninfo) [unioninfo :集合类型]
   6. set\_info = Setinfo(你要操作的集合)



1. 将你自己的信息封装成一个类Student，包括姓名、性别、年龄、家庭地址。并在display()方法中显示这些信息。



1. 用代码体现继承的特点。