* **Python基础：**

**Python基础 26 自己的模块**

自己的模块m1:

def printdata(data):

print(‘I am m1’)

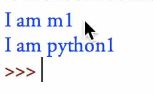
print(data)

调用自己的模块：

import m1

print(m1.printdata(‘I am python 1’))

运行结果：



* 注意：m1文件应当与调用文件放在同一目录下，如下：



* 另一个方式是放在python的site-packages目录下，同样可以调用

**Python 27 continue&break教学**

与C语言相似，作用分别为终止循环&跳出循环

**程序如下：**

a=True

while a:

b=input(‘type something’)

if b==’1’:

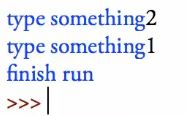
a=False

else:

pass

print(‘finish run’)

**运行结果：**



**#break终止循环#**程序如下：

while True:

b=input(‘type something’)

if b==’1’:

break

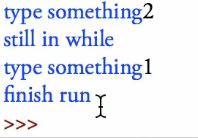
else:

pass

print(‘still in while’)

print(‘finish run’)

运行结果：



**#continue跳出循环#**程序如下：

while True:

b=input(‘type something’)

if b==’1’:

continue

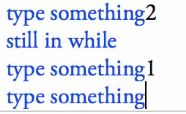
else:

pass

print(‘still in while’)

print(‘finish run’)

**运行结果：**



**Python 28 错误处理try**

**程序如下：**

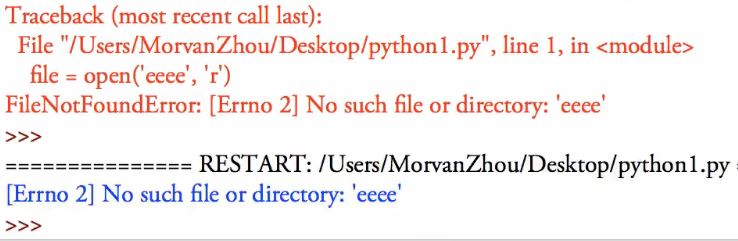
try:

file=open(‘eee’,’r’)%%%%%%打开名为‘eee’的文件

except Exception as e:

print(e)%%%%%输出错误原因

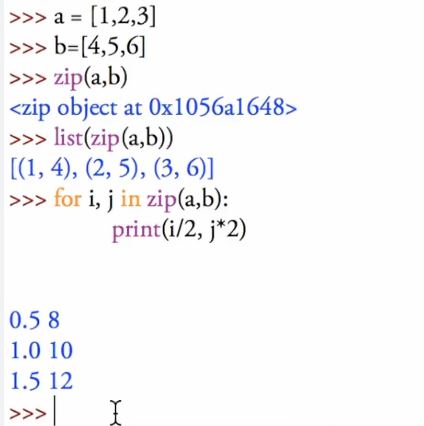
**运行结果：**



**Python 29 zip lambda map**

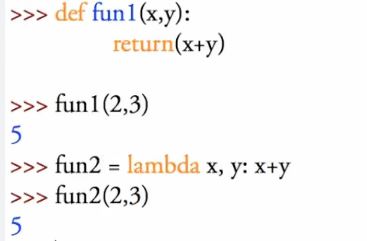
* map:把某个功能和参数一起运算
* lambda：map的简化版
* zip：迭代器

**zip**程序如下：



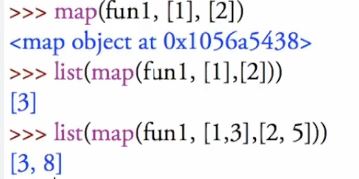
* 也可以zip三位及以上的变量：list（zip（a,a,b））

**lambda**程序如下：



* 所需写的代码减少了一些

**map**程序如下：

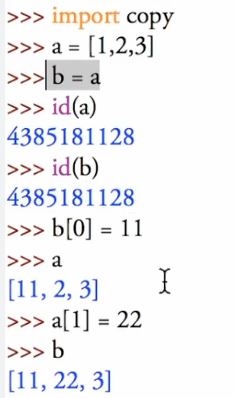


* 无法直接输出，需要用list输出

**Python 30 浅复制copy&深复制deepcopy**

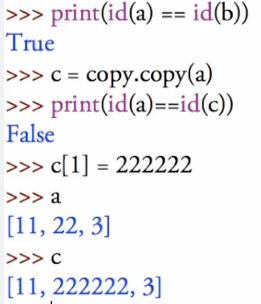
* 类似于共同体

**程序如下：**



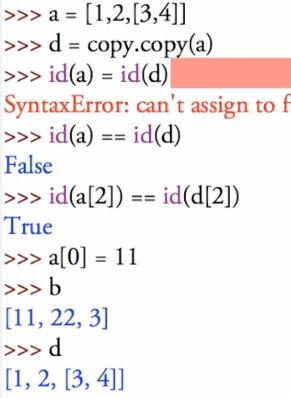
* 改变b值的同时，a的值也将随之改变

**浅复制copy**程序如下（接上）：



* 浅复制的c与a不是同一个变量，改变c的值时，a值并不会改变

**深复制deepcopy**程序如下：



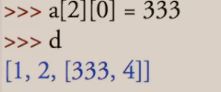
**a是a的一层列表**

**a[2]是a的二层列表**

**copy方式不一样**

**此时改变a的值，d内容并不能被改写**

* 但是如果改变a[2]中的内容：



**此时改变a的值，d内容被改写**

* 这就是copy.copy()的独特功能：第一层列表内容不会存储在同一空间，而第二层列表则会存储在同一空间
* 而copy.deepcopy()则不会重复引用到同一空间
* 而b==a则是共用同一空间
* **论文：《基于LSTM型RNN的CAPTCHA识别方法》**

1. CAPTCHA是一种可视化的人机交互证明，本质是用机器证明区分测试者是机器还是自然人。
2. 目的是防止人工智能的机器人滥用自然人力资源。
3. CAPTCHA的主要识别方式有**模板匹配法**、**BP神经网络法**、**SVM法**：
4. 模板匹配法与后两者的区别在于预处理和分割的方法
5. 它们有一个共同的阶段：图像分割。在目前的文字识别领域中，高级CAPTCHA已使用粘着严重的字符，而粘着字符的分割现在是一个仍待攻克的难题。
6. 基于LSTM型RNN的CAPTCHA识别方法：
7. 其优势在于不需要对CAPTCHA中的字符进行分割，直接使用滑动窗口在图片上提取特征值序列，然后与图片对应的字符序列构成训练样本，用其对RNN进行训练，避免了分割难题。
8. 特征提取:这段我看得云里雾里的，我读出来的意思就是滑动窗口提取出来的特征序列特征性不强，将灰度值作为特征值；过大的图像数据增加了计算的难度，需对其进行降维计算。
9. RNN的解码
10. 最优路径解码：此法认为由最大概率的路径形成的字符序列为最终输出，这也会使输出发生错误。
11. 领域解码算法：基于最优路径解码，求出最大概率解，将其作为初始解，再在初始解的邻域中寻找最优解。