

python文章

-----收集于迪艾姆公司python培训班主讲老师写的短文。

- 1、遍历list时，又修改list，这是初学者常犯错误。
- 2、python和递推算法解决一张大饼切N刀最多切多少块的问题
- 3、python初学者最常问的三个问题：
- 4、python子类调用父类（或者重载父类的）构造函数，有2个用法。
- 5、循环和递归实现二分法查找list，返回位置
- 6、python写excel文件，并设置单元格字体、字体大小、颜色等
- 7、在开发自用爬虫过程中如何处理汉字编码？
- 8、python用urllib2模拟post,get采集信息
- 9、python核心编程6-14习题的解题思路修改版
- 10、将python2中汉字会出现乱码的事一次性说清楚。
- 11、用一个例子演示python zip函数的应用
- 12、python的__call__()方法这样理解就对了
- 13、python学习之路（针对没有任何编程经验者或懂一点shell）
- 14、列表分组pythonic式的写法，代码来自[web.py](#) list(group([1,2,3,4], 2))
- 15、如果有个list=['a','b','c',.....'z'],怎么把这个list变成list=[['a','b','c'], ['d','e','f'].....]呢，这是以3个位一组，那如果是以4个一组呢，N个一组。
- 16、python多态示例代码
- 17、当你在自学python过程中，遇到问题，也没有帮助，怎么办，可以通过以下渠道获得帮助。
- 18、文本文件读写的3种方法。
- 19、[python用穷举法和逻辑表达式解题](#)

- 20、python安装模块的多种方法。
- 21、ubuntu server 12.04LTS+Nginx+uWSGI+[web.py](#) 配置备忘
- 22、django的forms类有2种实现方式。
- 23、python函数调用顺序,python函数互相调用顺序
- 24、 python核心编程第9章11题a的解题思路
- 25、帮一个网友写的代码，python用正则表达式抽出汉字和邮编到字典中
- 26、 php剪刀石头布编程思路
- 27、web开发，邮件激活思路
- 28、 7段python视频地址
- 29、python培训班简介

遍历list时，又修改list，这是初学者常犯错误。

```
#!/usr/bin/python
```

```
# -*- coding:utf-8 -*-
```

```
"""
```

黄老师写于2013-3-28日中午。

欢迎参加 “python培训-2013年python就业培训班 “

<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi170.html>

群内有2个同学犯这样的错误，遍历list，同时修改list。

```
list1 = [2,4,8,9,11,12,42,56,33,35]
```

```
for item in list1:
```

```
    if item % 2 != 0:
```

```
list1.remove(item)
```

```
print list1
```

以上代码是错误，要怎么才能输出正确的结果呢？

在遍历前加一个新的空list，遍历list时，逻辑处理后，将结果加入到新的list中

```
"""
```

```
list1 = [2,4,8,9,11,12,42,56,33,35]
```

```
list2 = []
```

```
for item in list1:
```

```
    if item % 2 != 0:
```

```
        list2.append(item)
```

```
print list2
```

#这是个示例，当然示例用下面的方法最简单。

```
print [item for item in list1 if item % 2 != 0]
```

python和递推算法解决一张大饼切N刀最多切多少块的问题

经常有家长抱怨小学出这么难的题目。

一张大饼，切三刀，最多能切多少块。其实这个问题是考的圆形直线相交的问题。

看"递推算法"是如何解决这一类问题的。

问题：

王小二自夸刀工不错，有人放一张大的煎饼在砧板上，问他："饼不许离开砧板，切100刀最多能分成多少块"。

解答：

切法是让2刀的直线都有交叉点，先找规律，令 $q(n)$ 为切 n 刀的块数，

$$q(1)=1+1=2$$

$$q(2)=1+1+2=4$$

$$q(3)=1+1+2+3=7$$

$$q(4)=1+1+2+3+4=11$$

归纳一下， $q(n)=q(n-1)+n$

$$q(0)=1 \text{ 一刀不切，还是一张大饼。}$$

下面用python代码计算。

```
#coding:utf-8
```

```
kshu_list = [0]*100
```

```
kshu_list[0] = 1
```

```
for i in xrange(1,100):
```

```
    kshu_list[i] = kshu_list[i-1]+i
```

```
print "100刀后最多切%d块" %kshu_list[99]
```

python初学者最常问的三个问题：

看啥书

python书有100多种，选择最适合入门的书最重要。

本人推荐几种，供大家参考。

A Byte of Python中文教程.pdf python简明教程中文.pdf---适合有c,C++,java,C#经验者，快速入门，快速用在工作上。

Introduction to Programming Using Python.pdf-----适合没有任何编程经验者，华人写的编程书。

Think Python How to Think Like a Computer Scientist---在线免费书，电子书地址<http://openbookproject.net/thinkcs/python/english2e/>

Learn Python The Hard Way 2nd Edition.pdf-----适合没有任何编程经验者，喜欢通过做题学习的同学。电子书地址<http://learnpythonthehardway.org/>

OReilly.Learning.Python.3rd.Edition.Oct.2007.pdf-----适合没有任何编程经验者，喜欢讲的详细的同学。

Python 核心编程 第二版-----适合有编程经验者，全面掌握python知识。

用啥ide

用这些ide

初学者用python自带的ide:idle就可以

PyCharm

PyDev(Eclipse)

Pyscripter

Sublime Text 2

TextMate UliPad

Vim

装python2还是python3

python目前有2个版本，python2和python3.

为啥现阶段还是学习python2.X,不学python3

基于以下理由：

python2.x还会流行多年。

python社区需要很多年才能将现有的模块移植到支持python3. django [web.py](#) flask等还不支持python3

掌握python2.x，过度到python3也很容易。

现在找工作，大部分用python2.x.

python子类调用父类（或者重载父类的）构造函数，有2个用法。

```
#coding:utf-8
```

```
"""
```

python子类调用父类（或者说是重载父类的）构造函数，有2个用法。

微博"python爱好者写于2012-12-12，北京下雪中"

```
"""
```

```
class Father_class(object):
```

```
    def __init__(self,x):
```

```
        self.x=x
```

```
    def print_x(self):
```

```
        print self.x
```

```
class Sun_class(Father_class):
```

```
    def __init__(self,x,y):
```

"""直接用父类的类名，加上父类的构造函数"""

```
Father_class.__init__(self,x)
```

```
self.y=y
```

```
def print_y(self):
```

```
    print self.y
```

```
newsun = Sun_class(6,32)
```

```
newsun.print_x()
```

```
newsun.print_y()
```

```
class Father_class(object):
```

```
    def __init__(self,x):
```

```
        self.x=x
```

```
    def print_x(self):
```

```
        print self.x
```

```
class Sun_class(Father_class):
```



```
def __init__(self,x,y):
```

```
    #Father_class.__init__(self,x)
```

#用这种方法重载父类的构造函数注意后面__init__不能带self,因为这里是调用父类,
self指子类的实例。

```
    super(Sun_class,self).__init__(x)
```

```
    self.y=y
```

```
def print_y(self):
```

```
    print self.y
```

```
newsun = Sun_class(6,32)
```

```
newsun.print_x()
```

```
newsun.print_y()
```

循环和递归实现二分法查找list，返回位置

```
#!/usr/bin/env python
```

```
# -*- coding:utf-8 -*-
```

```
"""
```

二分法查找的循环和递归实行代码

python爱好者写于2012年8-15日

微博：python爱好者

```
"""
```

#用循环实现二分法查找

```
def binary_serach_while(mylist,find):
```

```
    left = 0
```

```
    right = len(mylist)-1
```

```
    while left <= right :
```

```
        midpoint = (left + right)/2
```

```
        if find == mylist[midpoint]:
```

```
            return midpoint
```

```
        elif find < mylist[midpoint]:
```

```
            right = midpoint-1
```

```
        else :
```

```
            left = midpoint+1
```

```
test_list=[3,7,555,987,6,8,45]
```

```
test_list.sort()
```

```
#for item in test_list:
```

```
    # print item
```

```
print "用循环实现二分法查找:"
```

```
print binary_serach_while(test_list,6)
```

#用递归实现二分法查找

```
def binary_serach_recu(mylist,find,left=0,right=None):
```

```
    if right is None :
```

```
        right = len(mylist)-1
```

```
    if left == right :
```

```

    if find == mylist[right]:
        return right
    else :
        return -1
else :
    midpoint = (left + right)/2
    if find > mylist[midpoint]:
        return binary_serach_recu(mylist,find,midpoint+1,right)
    else :
        return binary_serach_recu(mylist,find,left,midpoint)
print "用递归实现二分法查找:"
print binary_serach_recu(test_list,66)

```

python写excel文件，并设置单元格字体、字体大小、颜色等

如何用python和xlwt写excel文件，并设置单元格字体、字体大小、颜色等

```
#coding=utf-8
```

```
import xlwt
```

```
#file = xlwt.Workbook(encoding = 'utf-8')
```

```
wbk = xlwt.Workbook(encoding = 'utf-8')
```

```
sheet = wbk.add_sheet('sheet 1', cell_overwrite_ok=True)
```

#这样表单就被创建了,写入数据也很简单:

indexing is zero based, row then column

```
#sheet2 = wbk.add_sheet('sheet 2', cell_overwrite_ok=True)
```

```
style = xlwt.XFStyle()
```

```
font = xlwt.Font()
```

```
font.name = 'SimSun' # 指定“宋体”
```

```
style.font = font
```

```
sheet.write(0,1,'test text')
```

```
#sheet.write(1,2,'excel读写测试',style)
```

```
sheet.write(1,2,'excel 读写测试',xlwt.easyxf('font: height 240, name Arial, colour_index  
black, bold off, italic on; align: wrap on, vert centre, horiz left;'))
```

```
)
```

#之后，就可以保存文件(这里不需要想打开文件一样需要close文件):

```
wbk.save('test4.xls')
```

```
## Startup Code Begin
```

```
import xlwt
```

```
from xlwt import *
```

```
import string
```

```
import os
```

```
from datetime import datetime
```

```
import time
```

```
from time import gmtime, strftime
```

```
current_date = strftime("%b %d %Y (%H:%M)")
```

```
## Startup Code End
```

```
## Modules Begin
```

```
def _size_col(sheet, col):
```

```
    return sheet.col_width(col)
```

```
def _size_row(sheet, row):
```

```
    return sheet.row_height(row)
```

```
## Modules End
```

```
## Style variable Begin
```

```
tittle_style = xlwt.easyxf('font: height 400, name Arial Black, colour_index blue, bold on;  
align: wrap on, vert centre, horiz center;'    "borders: top double, bottom double, left  
double, right double;")
```

```
subtittle_left_style = xlwt.easyxf('font: height 240, name Arial, colour_index brown, bold on,  
italic on; align: wrap on, vert centre, horiz left;'    "borders: top double, bottom double, left  
double;")
```

```
subtittle_right_style = xlwt.easyxf('font: height 240, name Arial, colour_index brown, bold  
on, italic on; align: wrap on, vert centre, horiz left;'    "borders: top double, bottom double,  
right double;")
```

```
subtittle_top_and_bottom_style = xlwt.easyxf('font: height 240, name Arial, colour_index  
black, bold off, italic on; align: wrap on, vert centre, horiz left;'    "borders: top double,  
bottom double;")
```

```
blank_style = xlwt.easyxf('font: height 650, name Arial, colour_index brown, bold off; align:  
wrap on, vert centre, horiz left;'    "borders: top double, bottom double, left double, right  
double;")
```

```
normal_style = xlwt.easyxf('font: height 240, name Arial, colour_index black, bold off; align:  
wrap on, vert centre, horiz left;'    "borders: top double, bottom double, left double, right  
double;")
```

```
## Style variable End
```

```
## Module Add Begin
```

```
def print_blank_line(A, B, C, D):
```

```
    ws.write_merge(A, B, C, D, "", xlwt.easyxf('font: height 240, name Arial, colour_index  
black, bold off, italic on; align: wrap on, vert centre, horiz left;'))
```

```
## Module Add End
```

```
## Variable Begin
```

```
Excel = xlwt.Workbook()
```

```
## Variable End
```

```
## Sheet Name Begin
```

```
ws = Excel.add_sheet('Test Sheet')
```

```
## Sheet Name End
```

```
## Column Width Determine Begin
```

```
for A in range(100):
```

```
    ws.col(A).width = 600
```

```
## Column Width Determine End
```

```
## Tittle Picture Begin
```

```
ws.write_merge(0, 0, 0, 42, "", xlwt.easyxf('font: height 700, name Arial, colour_index  
brown, bold off; align: wrap on, vert centre, horiz left;'))
```

```
##ws.insert_bitmap('test.bmp', 0, 0, 60, 5, 1.75, 1)
```

```
## Tittle Picture End
```

```
## Subtittle Write Begin
```

```
# Variable Begin
```

```
job_name = "Nicolas_Parametric"
```

```
fab_name = "Nicolas_Parametric"
```

```
printed_by = "Nic"
```

```
printed_date = current_date
```

```
# Variable End
```

```
border_type = ["dashed", "double", ""]
```

```
ws.write_merge(1, 1, 0, 6, "Job Name:", xlwt.easyxf('font: height 240, name Arial,  
colour_index black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top  
double, bottom double, left double;"))
```

```
ws.write_merge(1, 1, 7, 21, job_name, xlwt.easyxf('font: height 240, name Arial,  
colour_index black, bold off, italic on; align: wrap on, vert centre, horiz left;' "borders: top  
double, bottom double;"))
```

```
ws.write_merge(2, 2, 0, 6, "Fab Name:", xlwt.easyxf('font: height 240, name Arial,  
colour_index black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top  
double, bottom double, left double;"))
```



```
ws.write_merge(2, 2, 7, 21, fab_name, xlwt.easyxf('font: height 240, name Arial,
colour_index black, bold off, italic on; align: wrap on, vert centre, horiz left;' "borders: top
double, bottom double;"))
```

```
ws.write_merge(1, 1, 22, 27, "Printed By:", xlwt.easyxf('font: height 240, name Arial,
colour_index black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top
double, bottom double;"))
```

```
ws.write_merge(1, 1, 28, 42, printed_by, xlwt.easyxf('font: height 240, name Arial,
colour_index black, bold off, italic on; align: wrap on, vert centre, horiz left;' "borders: top
double, bottom double, right double;"))
```

```
ws.write_merge(2, 2, 22, 27, "Printed Date:", xlwt.easyxf('font: height 240, name Arial,
colour_index black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top
double, bottom double;"))
```

```
ws.write_merge(2, 2, 28, 42, printed_date, xlwt.easyxf('font: height 240, name Arial,
colour_index black, bold off, italic on; align: wrap on, vert centre, horiz left;' "borders: top
double, bottom double, right double;"))
```

```
print_blank_line(3, 3, 0, 42)
```

```
ws.write_merge(4, 4, 0, 42, "Report Name", xlwt.easyxf('font: height 400, name Arial,
colour_index red, bold on, italic on, underline on; align: wrap on, vert centre, horiz center;'))
```

```
print_blank_line(5, 5, 0, 42)
```

```
ws.write_merge(6, 6, 0, 1, "#", xlwt.easyxf('font: height 240, name Arial, colour_index
black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top double,
bottom double, left double, right double;"))
```

```
ws.write_merge(6, 6, 2, 7, "Piecemark", xlwt.easyxf('font: height 240, name Arial,
colour_index black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top
double, bottom double, left dashed, right double;"))
```

```
ws.write_merge(6, 6, 8, 10, "Qty", xlwt.easyxf('font: height 240, name Arial, colour_index
black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top double,
bottom double, left dashed, right double;"))
```

```
ws.write_merge(6, 6, 11, 20, "Size", xlwt.easyxf('font: height 240, name Arial, colour_index  
black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top double,  
bottom double, left dashed, right double;"))
```

```
ws.write_merge(6, 6, 21, 34, "Status", xlwt.easyxf('font: height 240, name Arial,  
colour_index black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top  
double, bottom double, left dashed, right double;"))
```

```
ws.write_merge(6, 6, 35, 42, "Date", xlwt.easyxf('font: height 240, name Arial, colour_index  
black, bold on, italic on; align: wrap on, vert centre, horiz left;' "borders: top double,  
bottom double, left dashed, right double;"))
```

```
## Subtittle Write End
```

```
normal_font = xlwt.easyxf('font: height 240, name Arial, colour_index black; align: wrap on,  
vert centre, horiz left;' "borders: top dashed, bottom dashed, left double, right double;"))
```

```
for A in range(98):
```

```
    ws.write_merge(7 + A, 7 + A, 0, 1, "", normal_font)
```

```
    ws.write_merge(7 + A, 7 + A, 2, 7, "", normal_font)
```

```
    ws.write_merge(7 + A, 7 + A, 8, 10, "", normal_font)
```

```
    ws.write_merge(7 + A, 7 + A, 11, 20, "", normal_font)
```

```
    ws.write_merge(7 + A, 7 + A, 21, 34, "", normal_font)
```

```
    ws.write_merge(7 + A, 7 + A, 35, 42, "", normal_font)
```

```
## File Save As Begin
```

```
Excel.save('Test.xls')
```

```
## File Save As End
```

```
#### Start File Begin

##os.startfile("C:/Documents and Settings/dev01/Desktop/Test.xls")

#### Start File End
```

在开发自用爬虫过程中如何处理汉字编码？

在开发自用爬虫过程中，有的网页是utf-8，有的是gb2312,有的是gbk，怎么办？

python从零基础到网页采集培训

下面所说的都是针对python2.7

如果不加处理，采集到的都是乱码，解决的方法是将html处理成统一的utf-8编码。

#chardet 需要下载安装

```
import chardet
```

```
#抓取网页html
```

```
html_1 = urllib2.urlopen(line,timeout=120).read()
```

```
#print html_1
```

```
mychar=chardet.detect(html_1)
```

```
#print mychar
```

```
bianma=mychar['encoding']
```

```
if bianma == 'utf-8' or bianma == 'UTF-8':
```

```
#html=html.decode('utf-8','ignore').encode('utf-8')
html=html_1
else :
    html =html_1.decode('gb2312','ignore').encode('utf-8')
有以上处理，整个html就不会是乱码。
```

python用urllib2模拟post,get采集信息

python用urllib2模拟post,get采集信息

来源于<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi43.html>

post,get用firefox的插件httpfox等拦截得到的。

```
import urllib2,re,time
```

```
file2=open('url.txt','w')
i=0
for i in range(1,300):
    i=i+1
    file2.write('编号'+str(i)+line+'\n')
    #按照具体网页href后的链接写正则，也可以用美丽的汤
    rc = '<a href=(.*)target="'+_blank" '>'
    #rc = '<a href=(.*)target="_blank" class="t" tips="t">'
    #×××××填写你在拦截post或get的信息。
    headers = {
        'GET': '*****',
        'User-Agent': 'Mozilla/5.0 (Windows NT 5.1; rv:5.0) Gecko/20100101 Firefox/5.0',
        'Host': '*****',
        'Referer': '*****',
        'Cookie': '*****'
    }
    req = urllib2.Request(
        url = '*****'+str(i)+'/',
        headers = headers
    )
    try :
```

```
result = urllib2.urlopen(req,timeout=30).read() #timeout=30设置很重要，防止urllib2死在那儿。
```

```
urls=re.findall(rc,result)
```

```
for url in urls:
```

```
    file2.write(url+'\n')
```

```
except :
```

```
    pass
```

```
time.sleep(8)
```

```
file2.close()
```

python核心编程6-14习题的解题思路修改版

```
#coding:utf-8
```

```
"""
```

python核心编程6-14习题的解题思路

设计一个"石头,剪子,布"游戏,有时又叫"Rochambeau",你小时候可能玩过,下面是规则.你和你的对手,在同一时间做出特定的手势,必须是下面一种手势:石头,剪子,布.胜利者从下面的规则中产生,这个规则本身是个悖论.

(a) 布包石头.

(b) 石头砸剪子,

(c) 剪子剪破布.在你的计算机版本中,用户输入她/他的选项,计算机找一个随机选项,然后由你的程序来决定一个胜利者或者平手.注意:最好的算法是尽量少的使用 if 语句.

写于2013-4-8下午, 5月23日修改, 在终端下运行python *.py

不同的平台可能会出现汉字编码问题。

在mac os 终端测试过。

python视频培训班

<http://www.gy7788.com.cn/shiyongxinxi/shiyongxinxi193.html>

```
"""
```

```
import random
```

```
guess_list = ["石头","剪刀","布"]
```

```
guize = [["布","石头"],["石头","剪刀"],["剪刀","布"]]
```

```

while True:
    computer = random.choice(guess_list)

    people = raw_input('请输入：石头,剪刀,布\n').strip()

    # people = people.encode("utf-8")

    if people not in guess_list:
        people = raw_input('重新请输入：石头,剪刀,布\n').strip()
        continue
    if computer == people:
        print "平手，再玩一次！"

    elif [computer,people] in guize :
        print "电脑获胜！"
    else:
        print "人获胜！"
        break

```

将python2中汉字会出现乱码的事一次性说清楚。

为了让初学者，不再为python2中汉字出现乱码的事烦恼！

python培训班暑假班和周末班

<http://www.010dm.com/xflml/3069.html>

python视频培训班

<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi193.html>

请看迪艾姆公司python培训上课老师黄哥细细道来：

1、写的代码模块需要指定编码

如果代码没有指定coding,python就默认所有的字符为ASCII码,

ASCII码只支持256个字符,ASCII码不支持中文,所以就报错。

所以要在代码前写上#coding:utf-8或#coding:gbk

但通用写上#coding:utf-8

2、python2内部所有编码统一为unicode
unicode可以处理世界上所有语言的字符。

utf-8为unicode的一种实现形式，所以需要在代码前写上#coding:utf-8

3、编码转换

牢记python2内部编码为unicode.

其它的编码decode()为unicode,再编码encode()为你指定的编码,就不会出现乱码。

4、网页采集时

代码指定#coding:utf-8

如果网页的编码为gbk

需要这样处理：

```
html = html.decode('gbk').encode('utf-8')
```

5、代码前也可以写#coding:gbk,但也要保证你的代码文件的保存格式为gbk.这个在windos下会出现这样的问题。

6、字典等key或值的汉字问题

```
#coding:utf-8
```

```
dict1 = {1:'python周末培训班',2:'咨询010-68165761 QQ: 1465376564'}
```

```
print dict1
```

```
# 这样输出的没有显示汉字，是显示汉字的其它编码
```

```
dict2 = {1:'python视频培训班',2:'咨询010-68165761 QQ: 1465376564'}
```

```
for key in dict2:
```

```
    print dict2[key]
```

7、unicode的汉字编码写到文本文件中

需要根据文本文件的编码进行转换

可以encode('utf-8')或encode('gbk')

总结：凡是报错信息中出现的错误包含“ASCII”，一般都是汉字编码的问题。

用一个例子演示python zip函数的应用

```
#coding:utf-8
"""
迪艾姆python培训老师写于6-26早
python培训班暑假班和周末班http://www.010dm.com/xflml/3069.html
"""

str_numnum='3.1415926'

number=['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.']
translate=['零', '壹', '贰', '叁', '肆', '伍', '陆', '柒', '捌', '玖', '点']
dict1 = dict(zip(number,translate))
mi = ""
for item in str_numnum:
    mi += dict1[item]
```

python的__call__()方法这样理解就对了

```
#coding:utf-8
class TestCall(object):
    """
    由迪艾姆公司python培训班黄哥所写。python培训班暑假班和周末班
    http://www.010dm.com/xflml/3069.html
    __call__说明，一看我这个，当即明白怎么回事
    这个__call__特殊的方法，作用是让类对象生成的实例对象可以函数一样调用。
    它重载了()功能，当实例对象带括号()调用时，被 __call__ 这个方法钩住（所谓的钩子），也就是说
    实例对象带括号()调用的是__call__这个方法。如果__call__()带参数，实例对象括号内必须也要带参数。
    """

    def __call__(self,num=0):
        print "__call__ 调用"
        print str(num)

foo = TestCall()
foo()
foo(5)
```


python学习之路（针对没有任何编程经验者或懂一点shell）

- 1、一些初学者认为python很简单，就以为编程很简单。
- 2、没有学好基础知识，听说django很火，就开始学django开发web。
- 3、建议没有任何编程经验者，象学C语言那样来训练学习python。
- 4、从语句、表达式、顺序结构、判断结构、循环结构学起，最后过度到函数抽象，类抽象。
- 5、先打好基础、再玩高级的应用：生成器、装饰器、类的一些属性的高级用法、正则、多线程、多进程等。
- 6、怎么克服只能看懂玩具代码，不会写代码，唯一的出路是多实践、多敲代码。

一个资深的系统运维工程师在北京公司7000到1万左右，如果会python编程，公司可以到达1.5万。

好多系统运维工程师对学习python有动力和需求，但需要注意学习的方法。

python视频培训班<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi193.html>愿意加快你的学习进度。

列表分组pythonic式的写法，代码来自[web.py](#)

```
list(group([1,2,3,4], 2))
```

```
def group(seq, size):
```

```
    """
```

```
    Returns an iterator over a series of lists of length size from iterable.
```

```
    >>> list(group([1,2,3,4], 2))
```

```
    [[1, 2], [3, 4]]
```

```
    >>> list(group([1,2,3,4,5], 2))
```

```
    [[1, 2], [3, 4], [5]]
```

```
    """
```

```
def take(seq, n):
```

```

    for i in xrange(n):
        yield seq.next()

if not hasattr(seq, 'next'):
    seq = iter(seq)
while True:
    x = list(take(seq, size))
    if x:
        yield x
    else:
        break

```

如果有个list=['a','b','c',.....'z'],怎么把这个list变成list=[['a','b','c'],['d','e','f'].....]呢，这是以3个位一组，那如果是4个一组呢，N个一组。

最简单的方法参考上面的文章 “列表分组pythonic式的写法，代码来自[web.py](#)
list(group([1,2,3,4], 2))”

```
#coding:utf-8
```

```
"""
```

本代码由黄老师所写

2013年python培训就业班

培训的第一目标是：通过学习python学会编程 价格最低，质量有保证

第一期周末班，已经上课，讲课方式得到学员的好评！

上课:2013-5-20就业班

限10名，名额有限，请速报名 咨询:qq:1465376564 电话：010-68165761

<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi181.html>

```
"""
```

```
import string
```

```

def return_list(list1,n):
    """
    这个函数的形式参数是list,返回值也是list
    """
    length = len(list1)
    list2 = []
    list3 = []
    if length % n == 0:
        for i in xrange(length):
            list2.append(list1[i])

            if (i+1) % n == 0 :
                list3.append(list2)
                list2= []
    #加这一段的目的是如果list的长度不是N的倍数，代码还可以优化
    #留提问者自己优化吧
    if length %n != 0 :
        for i in xrange(length/n*n):
            list2.append(list1[i])

            if (i+1) % n == 0 :
                list3.append(list2)
                list2= []
        list3.append(list1[(length/n*n):length])

    return list3

n =4

listp = ['a','b','c','d','e','f','g','h','i','k']
listletter = list(string.ascii_letters)
print return_list(listp,n)
print
print return_list(listletter,n)

```

python多态示例代码

```
#coding:utf-8
```

```
'''''
```

```
python多态演示  
2013年python培训就业班
```

培训的第一目标是：通过学习python学会编程 价格最低，质量有保证
第一期周末班，已经上课，讲课方式得到学员的好评！
上课:2013-5-20就业班

限10名，名额有限，请速报名 咨询:qq:1465376564 电话: 010-68165761
<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi181.html>

```
'''''
```

```
class People(object):
```

```
    def __init__(self,name):  
        self.name = name
```

```
    def printName(self):  
        print "In People类中,My name is %s",(self.name)
```

```
class Man(object):
```

```
    def __init__(self,name):  
        self.name = name
```

```
    def printName(self):  
        print "In Man类中,My name is %s",(self.name)
```

```
for p in [People("AAA"),Man("BBB")]:  
    p.printName()
```

当你在自学python过程中，遇到问题，也没有帮助，怎么办，可以通过以下渠道获得帮助。

There are a variety ways to get help for Python.

有这些途径可以得到关于python的帮助。

1、 Do a Google search, starting with the word "python", like "python list" or "python string lowercase". The first hit is often the answer. This technique seems to work better for Python than it does for other languages for some reason.

试着在google搜"python",象这样 "python list" 或"python string lowercase",几乎可以在第一页找到满意的答案。

2、 The official Python docs site -- docs.python.org -- has high quality docs. Nonetheless, I often find a Google search of a couple words to be quicker.

在python官方网站docs.python.org,可以找到很好的文档。

3、 There is also an official Tutor mailing list specifically designed for those who are new to Python and/or programming!

有许多官方的邮件列表供新python程序员使用。

4、Many questions (and answers) can be found on StackOverflow.

许多问题可以在这个StackOverflow网站找到答案

5、Use the help() [and dir()] functions described below.

可以在python shell下用help()或dir()方法显示模块的帮助或属性

文本文件读写的3种方法。

本文由迪艾姆公司python培训老师所写

python培训班暑假班和周末班<http://www.010dm.com/xflml/3069.html>

第一种方法：

```
file1 = open("test.txt")
file2 = open("output.txt","w")

while True:
    line = file1.readline()
    #这里可以进行逻辑处理
    file2.write(''+line[:s]+''+",")
    if not line:
        break
#记住文件处理完，关闭是个好习惯
file1.close()
file2.close()
```

读文件有3种方法：read()将文本文件所有行读到一个字符串中。

readline()是一行一行的读

readlines()是将文本文件中所有行读到一个list中，文本文件每一行是list的一个元素。

优点：readline()可以在读行过程中跳过特定行。

第二种方法：

文件迭代器，用for循环的方法

```
file2 = open("output.txt", "w")
for line in open("test.txt"):
    #这里可以进行逻辑处理
    file2.write('"' + line[:s] + '"' + ",")
```

第三种方法：

文件上下文管理器

```
with open('somefile.txt', 'r') as f:
    data = f.read()
```

```
# Iterate over the lines of the file
with open('somefile.txt', 'r') as f:
    for line in f:
        # process line
```

```
# Write chunks of text data
with open('somefile.txt', 'w') as f:
    f.write(text1)
    f.write(text2)
    ...
```

```
# Redirected print statement
with open('somefile.txt', 'w') as f:
    print(line1, file=f)
    print(line2, file=f)
```

python用穷举法和逻辑表达式解题

#coding:utf-8

"""

本代码由迪艾姆公司黄老师所写，思路大家自己看代码。

2013年python培训就业班（4月16日开班）

<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi181.html>

抓了a,b,c,d4名犯罪嫌疑人.其中有一名是小偷，审讯中：

a说我不是小偷

b说c是小偷

c说小偷肯定是d

d说c胡说！

其中有3个人说的是实话，一个人说的是假话，编程推断谁是小偷。

（用穷举法和逻辑表达式）

"""

```
for xiaotu in ['a','b','c','d']:
```

```
    sum = (xiaotu != 'a') + (xiaotu == 'c') + (xiaotu == 'd') + (xiaotu != 'd')
```

```
    if sum == 3:
```

```
        print "小偷是： %s " % xiaotu
```

```
-
```

python安装模块的多种方法。

1、自己写的模块，可以直接添加到路径下。这样就可以直接调用。

```
import sys
```

```
sys.path.append("/home/username/")
```

2、单文件模块

直接把文件拷贝到\$python_dir/lib

3、网上下载的第三方库，一般解压后，找setup.py文件

运行python setup.py install

4、egg文件

1) 下载ez_setup.py,运行python ez_setup

2) easy_install *.egg

5、pip安装方法

Pip 是安装python包的工具，提供了安装包，列出已经安装的包，升级包以及卸载包的功能。

Pip 是对easy_install的取代，提供了和easy_install相同的查找包的功能，因此可以使用easy_install安装的包也同样可以使用pip进行安装。

安装Pip

Pip的安装可以通过源代码包，easy_install或者脚本。

下面介绍一下各种安装方法：

源代码方式：

```
$ wget https://pypi.python.org/packages/source/p/pip/pip-1.2.1.tar.gz （替换为最新的包）
```

```
$ tar xzf pip-1.2.1.tar.gz
```

```
$ cd pip-1.2.1.tar.gz
```

```
$ python setup.py install
```

easy_install:

```
$ easy_install pip
```

get_pip.py 脚本：

```
$ curl -O https://raw.githubusercontent.com/pypa/pip/master/contrib/get-pip.py
```

```
$ sudo python get-pip.py
```

OK, 下面来看一下Pip的使用

安装package

```
$ pip install Markdown
```

列出安装的packages

```
$ pip freeze
```

安装特定版本的package

通过使用==, >=, <=, >, <来指定一个版本号。

```
$ pip install 'Markdown<2.0'
```

```
$ pip install 'Markdown>2.0,<2.0.3'
```

升级包

升级包到当前最新的版本, 可以使用-U 或者 --upgrade

```
$ pip install -U Markdown
```

卸载包

```
$ pip uninstall Markdown
```

查询包

```
pip search "Markdown"
```

6、特殊库的安装（一个科学计算库Enthought Python Distribution的安装）

Enthought Python Distribution在ubuntu下安装的测试, 应一个网友的请求, 问怎么安装这个库, 从官方网站上下载了一个文件

epd_free-7.3-2-rh5-x86.sh, 这个文件是sh 格式的文件, 用sudo bash epd_free-7.3-2-rh5-x86.sh 运行这个文件, 出现很多权限的阅读, 点回车, 不要连续点

需要点一次看一下，最后出现一个同一不同一它的版权声明，输入"yes",再下来是让选择安装的目录，点回车，就安装在当前目录下。

这样就安装好了。

当前目录下输入

```
>>> ipython --pylab
```

就出来

```
Python 2.7.3 (default, Apr 20 2012, 22:44:07)
```

```
Type "copyright", "credits" or "license" for more information.
```

```
IPython 0.12.1 -- An enhanced Interactive Python.
```

```
?      -> Introduction and overview of IPython's features.
```

```
%quickref -> Quick reference.
```

```
help    -> Python's own help system.
```

```
object? -> Details about 'object', use 'object??' for extra details.
```

```
Welcome to pylab, a matplotlib-based Python environment [backend: TkAgg].
```

```
For more information, type 'help(pylab)'.
```

再这样调用。这个库由于我工作上用不上，所以也不需要进一步的研究。

```
In [1]: from scipy import *
```

```
In [2]: a=zeros(1000)
```

```
In [3]: a[:100]=1
```

```
In [4]: b=fft(a)
```

```
In [5]: plot(abs(b))
```

ubuntu server 12.04 LTS+Nginx+uWSGI+[web.py](#) 配置备忘

以下是参考网友的文章和自己摸索的内容。安装完ubuntu server 12.04LTS（已经自带python2.7.3），再安装[web.py](#),mysql等。

-、先安装nginx官方最新版本1.2.4

nginx官方最新版本更新到1.2.4 ubuntu默认的版本是1.1.9

1、 打开文件：

```
sudo nano /etc/apt/sources.list
```

2 在上述文件中增加内容：

对于使用 Ubuntu 12.04 Precise 增加下列两行：

```
deb http://nginx.org/packages/ubuntu/ precise nginx
```

```
deb-src http://nginx.org/packages/ubuntu/ precise nginx
```

3. 更新资源库：

```
apt-get update
```

你会看到如下警告：

```
W: GPG error: http://nginx.org lucid Release: The following signatures  
couldn't be verified because the public key is not available: NO_PUBKEY
```

```
ABF5BD827BD9BF62
```

First of all this is only warning and you can ignore it, if you know what are you doing and in case you prefer to add public key, used for signing packages and repository, just run:

解决方法有两种，如下：

```
gpg -a --export 7BD9BF62 | sudo apt-key add -
```

或者

```
wget http://nginx.org/packages/keys/nginx\_signing.key
```

```
cat nginx_signing.key | sudo apt-key add -
```

本博建议第二种方法，因为本博用的第二种。

再次运行 apt-get update

ok 此次运行无警告提示

4. 安装 nginx 最新版本：

```
apt-get install nginx
```

如果之前安装过系统默认版本nginx，会提示以下错误：

```
dpkg: error processing /var/cache/apt/archives/nginx_1.2.2-1~precise_amd64.deb (-  
unpack):
```

```
trying to overwrite '/etc/logrotate.d/nginx', which is also in package nginx-common  
1.1.19-1
```

```
dpkg-deb: error: subprocess paste was killed by signal (Broken pipe)
```

```
Errors were encountered while processing:
```

```
/var/cache/apt/archives/nginx_1.2.2-1~precise_amd64.deb
```

请先删除nginx-common:

```
apt-get remove nginx-common
```

再次安装即可

二、安装uWSGI

```
wget http://projects.unbit.it/downloads/uwsgi-1.2.6.tar.gz
```

注意wget下载文件到当前目录下

```
tar xzvf uwsgi-1.2.6.tar.gz
```

```
cd uwsgi-1.2.6/
```

```
python uwsgiconfig.py --build
```

我操作这一步提示了类似这个错误 “Python.h: No such file or directory” 只要安装下python-dev 就可以了

```
sudo apt-get install python-dev
```

这样uwsgi就安装完了

三、配置nginx

[web.py](#)开发的代码放在 /var/www/qym/下。

需要sudo chmod -R 777 /var/www/qym/ 开发该目录的权限

打开配置文件 sudo nano /etc/nginx/nginx.conf

插入下面的配置信息

```
server {  
    listen    8080;  
    server_name localhost; #上线时再设置成其他的  
  
    access_log /var/log/nginx_access.log;  
    error_log /var/log/nginx_error.log;  
    location /static {  
        root    /var/www/qym/; #静态文件存放的目录  
    }  
  
    location / {  
        include uwsgi_params;
```

```
uwsgi_pass 127.0.0.1:9001;
```

```
uwsgi_param UWSGI_CHDIR /var/www/qymf;#代码存放的目录
```

```
uwsgi_param UWSGI_SCRIPT index; #关键点是指web.py开发的web应用程序的入口代码，不要带后缀.py
```

```
uwsgi_read_timeout 1800;
```

```
}
```

```
}
```

四、设置随机器启动

打开该文件sudo nano /etc/rc.local

加入以下一行uwsgi启动

```
uwsgi -s 127.0.0.1:9001 -w index
```

最后在浏览器中输入192.168.0.*:8080 就可以看到首页的内容

django的forms类有2种实现方式。

1、由models生表的表字段名，直接生成表单from的name

请看代码

```
class QymfForm(ModelForm):
```

```
    class Meta:
```

```
        model=Qymftable
```

用这种方式简单，问题是form的html表现形式不好控制。

2、自定义from,请看代码，是测试，代码写得很简陋。

```
class QymfForm2(forms.Form):
```

```
    dwmc=forms.CharField(max_length=100 )
```

```
    dz=forms.CharField(max_length=100)
```

```
    fzz=forms.CharField(max_length=60)
```

```

tel=forms.CharField(max_length=100)
fax=forms.CharField(max_length=100)
post=forms.CharField(max_length=6)
renshi=forms.CharField(max_length=100)
email=forms.CharField(max_length=100)
www=forms.CharField(max_length=100)
zczj=forms.CharField(max_length=50)
zgrs=forms.CharField(max_length=50)
zcrq=forms.CharField(max_length=50)
tzf=forms.CharField(max_length=50)
jjlx=forms.CharField(max_length=50)
yyfw=forms.CharField(widget=forms.Textarea(attrs={'width':'900px','height':'1450px'}))
hy=forms.CharField(max_length=50)
gx=forms.CharField(max_length=50)

```

```
def save(self):
```

```

    return Qymtable.objects.create(dwmc=self.cleaned_data['dwmc'],
        dz=self.cleaned_data['dz'],
        fzs=self.cleaned_data['fzs'],
        tel=self.cleaned_data['tel'],
        fax=self.cleaned_data['fax'],
        post=self.cleaned_data['post'],
        renshi=self.cleaned_data['renshi'],
        email=self.cleaned_data['email'],
        www=self.cleaned_data['www'],
        zczj=self.cleaned_data['zczj'],
        zgrs=self.cleaned_data['zgrs'],
        zcrq=self.cleaned_data['zcrq'],
        tzf=self.cleaned_data['tzf'],
        jjlx=self.cleaned_data['jjlx'],
        yyfw=self.cleaned_data['yyfw'],
        hy=self.cleaned_data['hy'],
        gx=self.cleaned_data['gx'])

```

自定义时，问题来了，

```

def update(request,id):
a = Qymtable.objects.get(pk=id)

```

```

form = QymlForm2(instance=a)
if form.is_valid():
    a=form.save()
    a.save()

```

```

return HttpResponseRedirect("/")
instance=a在moldes生成的表单时，可以这样用。

```

但是自定义form时，先用找到的特定记录对象，自动填充form时，instance=a失效了，报错，不能用。

以下是新的解决方法：

```

#xiugai
@csrf_protect
def edit(request, id):
    #incident = Qymltable.objects.get(pk=id)
    #a = get_object_or_404(Qymltable, id)
    a = Qymltable.objects.get(pk=id)

    if request.POST:
        #form = QymlForm2(request.POST, instance=a)
        form = QymlForm2(request.POST)

        #用post传过来的值更新特定记录对象的相应值
        a.dwmc=request.POST['dwmc']
        a.dz=request.POST['dz']
        a.fzr=request.POST['fzr']
        a.tel=request.POST['tel']
        a.fax=request.POST['fax']
        a.post=request.POST['post']
        a.renshi=request.POST['renshi']
        a.email=request.POST['email']
        a.www=request.POST['www']
        a.zczj=request.POST['zczj']
        a.zgrs=request.POST['zgrs']
        a.zcrq=request.POST['zcrq']
        a.tzf=request.POST['tzf']
        a.jjlx=request.POST['jjlx']

```



```

a.yyfw=request.POST['yyfw']
a.hy=request.POST['hy']
a.gx=request.POST['gx']
#a=form.save()
a.save()
return HttpResponseRedirect('/')
else:

```

#根据查询的特定记录对象，填充或初始花form

```

form = QymIForm2(initial={'dwmc': a.dwmc,
    'dz':a.dz,
    'fzr':a.fzr,
    'tel':a.tel,
    'fax': a.fax,
    'post':a.post,
    'renshi':a.renshi,
    'email': a.email,
    'www': a.www,
    'zczj': a.zczj,
    'zgrs':a.zgrs,
    'zcrq':a.zcrq,
    'tzf':a.tzf,
    'jjlx':a.jjlx,
    'yyfw':a.yyfw,
    'hy':a.hy,
    'gx':a.gx})
return render_to_response('contact_form2.html',
{'form':form},context_instance=RequestContext(request))

```

python函数调用顺序,python函数互相调用顺序

```

#-*- coding=utf-8 -*-

```

```

"""

```

程序头注释开始

```

*****

```

程序名: 函数顺序

作者:python爱好者

创建时间: 03-06-2012

目的:演示python 函数调用顺序。python也象C语言一样可以在一个函数中调用其他自定义函数。

函数最好是将被调用函数方在调用函数前面, 但是放在后面有时也是可以。对初学者来说, 还是遵循顺序执行的

顺序, 先定义, 再调用。

总结: 定义函数时, 不管内部被调用的函数, 在这个本函数调用前面, 还是后面。在实际调用函数时(非定义函数时)比喻下面实例, 一定要保证所有

被调用函数在它的前面(比喻这个语句"repeat_print()")就OK

程序头注释完

"""

```
#def print_str():
```

```
    #print "python very good"
```

```
    #print "3-1习题"
```

```
    #print "调用我, 只要我在调用语句的前面就OK,其他定义函数内部调用不属此限制"
```

```
def repeat_print():
```

```
    """调用print_str()函数 这里指定义该函数, 不需在定义时, 内部调用的函数一定要写在该函数定义之前, 但最好写
```

```
    在前面, 以使程序代码清晰好读。
```

```
    """
```

```
    print_str()
```

```
    print '*****'
```

```
    print_str()
```

```
#repeat_print() 放在这里不行, 为啥, 因为这行在被调用函数print_str()前面
```

```
def print_str():
```

```
    print "python very good"
```

```
    print "3-1习题"
```

```
    print "调用我, 只要我在调用语句的前面就OK,其他定义函数内部调用不属此限制"
```

```
repeat_print()
```

python核心编程第9章11题a的解题思路

```
#coding:utf-8
```

```
"""
```

python核心编程第9章11题a的解题思路

python核心编程第9章11题a的答案

本答案由python视频培训班授课老师-黄哥所写。

做这样的习题关键得有思路，参加python视频培训班的一个学员，对这种题目无从下手。

黄哥简单说一下思路：用list或dict来组织数据，文件记录数据，无穷循环+break来

控制文本菜单。由于本人时间有限，代码有进一步完善的必要，请感兴趣者，进一步完善。

也可以用字典来组织数据。

9-11.

Web site Addresses.

Write a URL bookmark manager. Create a text-driven menu-based application that allows the user to add, update, or delete entries.

Entries include a site name, Web site URL address,

and perhaps a one-line description (optional).

Allow search functionality so that a search "word" looks

through both names and URLs for possible matches.

Store the data to a disk file when the user quits the application,

and load up the data when the user restarts.

python视频培训班

形式：自己看视频+作业+答疑，利用业余时间学习。视频50讲，视频中详细的讲解解题思路，课后大量习题

训练学员自己写代码，通过训练达到不光能看懂代码，而且能自己写代码的目的。

目标：

第一目标是：通过学习python学会编程。

第二目标：会写爬虫的基础部分，网页采集。

第三目标：学会web开发

学习周期：三个月到6个月

咨询:qq:1465376564 电话: 010-68165761

<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi193.html>

```
"""
```

```
import re
```

```
import cPickle
```

```
from os.path import exists
```

```
def addUrl(urllist,new):
```

```
    urllist.append(new)
```

```
def deleteUrl(urllist,new):
```

```
    """删除"""
```

```
    urllist.remove(new)
```

```
def findUrl(urllist,webname="",url=""):
```

```
    """查询"""
```

```
    f = open("url.dat")
```

```
    storelist = cPickle.load(f)
```

```
    f.close()
```

```
    # print storelist
```

```
    for item in list(storelist):
```

```
        if webname in item[0] and url in item[1]:
```

```
            return item
```

```
def writeUrl(urllist):
```

```
    """写文件"""
```

```
    f = open(r"url.dat","w")
```

```
    cPickle.dump(urllist,f)
```

```
    f.close()
```

```
def readUrl():
```

```
    """读文件，先检测文件存在不？"""
```

```
    if exists('url.dat'):
```

```
        f = open("url.dat")
```

```
        storelist = cPickle.load(f)
```

```
        f.close()
```

```
        # print storelist
```

```

        for item in storelist:
            for urlitem in item:
                print urlitem

```

```

urllist = []
#将文件内部的原始信息读到urllist内
if exists('url.dat'):
    f = open("url.dat")
    storelist = cPickle.load(f)
    f.close()
    urllist .extend(storelist)

```

```

print (30 * '-')
print ("  M A I N - M E N U")
print (30 * '-')
print ("1. 增加")
print ("2. 删除")
print ("3. 查询")
print ("4. 写文件")
print ("5. 读文件")
print ("6. 退出")
print (30 * '-')

```

```

while True:
    try :
        choice = int ( raw_input('Enter your choice [1-6] : ') )

    except ValueError, e :
        print ("'%s' is not a valid integer." % e.args[0].split(": ")[1])

    # choice = int ( raw_input('Enter your choice [1-3] : ') )
    if choice == 1:
        print ("Starting 增加")
        webname = raw_input("请输入站名: ")
        url = raw_input("请输入网址: ")
        content = raw_input("请输入描述: ")

```

```

        # urllist.append([webname,url,context])
        addUrl(urllist,[webname,url,content])

elif choice == 2:
    print ("删除...")
    webname = raw_input("请输入站名: ")
    url = raw_input("请输入网址: ")
    content = raw_input("请输入描述: ")
    deleteUrl(urllist,[webname,url,content])

elif choice == 3:
    print ("查询...")
    webname =raw_input("请输入站名: ")
    url = raw_input("请输入网址: ")
    # print str(webname)
    chaxun = findUrl(urllist,webname,url)
    if chaxun:
        print "查询结果为: "
        for item in chaxun :
            print item
    else:
        print "没有此记录! "

elif choice == 4:
    print ("写文件...")
    writeUrl(urllist)

elif choice == 5:
    print ("读文件...")
    readUrl()

elif choice == 6:
    print ("退出...")
    break

else:
    print ("非法内容. Try again...")

```

帮一个网友写的代码，python用正则表达式抽出汉字和邮编到字典中

```
#coding:utf-8
import re
"""
```

本代码由黄老师(python爱好者,qq: 1465376564)所写.

怎么把下面这段字符串转化为字典格式(例如: '北京':'101010100', '朝阳':'101010300')?

```
“北 京101010100朝阳101010300顺义101010400怀柔101010500通州101010600昌平
101010700延庆 101010800丰台101010900石景山101011000大兴101011100房山
101011200密云101011300门头沟 101011400平谷101011500/\达岭101011600佛爷顶
101011700汤河口101011800密云上甸子101011900斋堂 101012000霞云岭101012100北京
城区101012200”
```

```
"""
```

```
str1 ="北京101010100朝阳101010300顺义101010400怀柔101010500通州101010600昌平
101010700延庆 101010800丰台101010900石景山101011000大兴101011100房山
101011200密云101011300门头沟 101011400平谷101011500/\达岭101011600佛爷顶
101011700汤河口101011800密云上甸子101011900斋堂 101012000霞云岭101012100北京
城区101012200"
```

```
tuples = re.findall(ur"([\u4e00-\u9fa5]+)(\d+)",str1.decode('utf8'))
```

```
dic1 = dict(tuples)
```

```
for key in dic1:
```

```
    print key.encode('utf-8')+':'+dic1[key].encode('utf-8')
```

php剪刀石头布编程思路

```
<?php
```

```
/*
```

本代码由python视频培训班黄哥所写。

python核心编程6-14习题,用php写一遍。

提供python培训和php培训

咨询:qq:1465376564 电话: 010-68165761

在linux下终端运行 php test.php

本代码在mac下测试运行无误。

总计: 这个代码是根据本人所写python代码修改过来的

学会一种编程语言, 再学第二种, 就很容易, 为啥?

编程思路是一样的。

```
*/
```

```
$my_array = array("石头","剪刀","布");
```

```
$guize = array(array("石头","剪刀"),array("剪刀","布"),array("布","石头"));
```

```
//上面2个变量定义一个需要输入的数组, 和一个获胜规则的二维数组
```

```
// var_dump($guize);
```

```
$rand_keys = array_rand($my_array);
```

```
$computer = $my_array[$rand_keys];
```

```
//取数组中随机值
```

```
echo $computer . "\n";
```

```
// echo $person;
```

```
while (True)
```

```
{
```

```
    echo "请输入: 石头 剪刀 布\n";
```

```
    $person = trim(fgets(STDIN)) ;
```

```
    $input = array($computer,$person);
```

```
    //将输入的$person和电脑随机产生的值构造一个数组
```

```
    //再判断在不在获胜规则数组中
```

```
    if (!(in_array($person,$my_array)))
```

```
    {
```

```
        echo "只能输入'剪刀、石头、布, 请重新输入'";
```

```
        continue;
```

```
    }
```

```
    if ($computer == $person )
```

```
    {
```

```
        echo "平手\n";
```



```

}
else if (in_array($input,$guize)) {
    echo "电脑胜\n";

}
else
{
    echo "人获胜\n";
    break;
}
}
?>

```

web开发，邮件激活思路

web开发，邮件激活思路

用[web.py](#),dajngo,tornado,flask等开发web，经常要用到邮件激活功能。

思路是这样的，可能还有其它好的方法，只提供参考！

- 1、数据库中用户表设计一个字段，原始值为0，表示没有激活。
- 2、用户注册代码中，构造一个链接，发到用户邮箱，链接要加密。
- 3、用这个链接来update，用户表的一个那个表示激活的字段。更新为1，表示激活。
- 4、要设置过期时间。

[web.py](#) 中写一个激活的类，类中定义一个GET方法。

当用户点击这个激活链接时，就调用这个类中的GET方法。

tornado和[web.py](#)思路一样。

7段python视频地址

既讲语法，又讲编程思路的视频。

python编程思路1 <http://you.video.sina.com.cn/b/105491176-1631501663.html>

python编程思路2 <http://you.video.sina.com.cn/b/105496654-1631501663.html>

python学习方法 <http://you.video.sina.com.cn/b/104818436-1631501663.html>
python的while循环 <http://you.video.sina.com.cn/b/100047526-1631501663.html>
python函数抽象 <http://you.video.sina.com.cn/b/100050769-1631501663.html>
python类 <http://you.video.sina.com.cn/b/100065337-1631501663.html>

嵌套函数

<http://video.sina.com.cn/v/b/112299484-1631501663.html>

python视频培训班

<http://www.qy7788.com.cn/shiyongxinxi/shiyongxinxi193.html>

特别提示：不喜欢做题，不喜欢动脑，不喜欢敲代码。请不要参加这个培训。

形式：自己看视频+作业+答疑，利用业余时间学习。视频50讲，视频中详细的讲解解题思路，课后大量习题

训练学员自己写代码，通过训练达到不光能看懂代码，而且能自己写代码的目的。

目标：

第一目标是：通过学习python学会编程。

第二目标：会写爬虫的基础部分，网页采集。

第三目标：学会web开发

学习周期：三个月到6个月

咨询:qq:1465376564 电话：010-68165761 黄老师

目标：通过学习python来到达学会编程的目的。

面向过程编程训练

1、变量，表达式，语句。

2、常用的数据结构：list,tuple,set,字典。

3、顺序结构、判断结构、循环结构(if判断 for while 循环)

4、文本文件读写，利用if语句和while循环将采集的有一定规律的文本处理成有规律文本，导入到mysql

以上重点是：逻辑思维的训练，多重循环在文本处理中的应用。

5、python和数据库编程

函数抽象和类抽象训练

6、函数,异常,类(展开来讲类和实例, 类属性和实例属性, 构造函数__new__和__init__方法, 析构函数__del__(), 调用绑定方法和 非绑定方法, 静态方法和类方法, 类封装性(数据隐藏), 类继承,重载, 运算符重载等, 最后怎么样用面向对象的思想来编写代码。), 进程、多线程, 递归等。

迭代器, 生成器精讲

闭包、装饰器精讲、类的属性精讲(@property, __getattr__, __setattr__, __getattr__, __slots__等特殊应用)

7、网络编程, urllib2的使用, 正则表达式, 字符处理, utf-8和gb2312互换。群发邮件如何实现。

8、爬虫(采集网页)实战:

模拟post,get.汉字乱码的处理。

网页自动翻页, 自动点击。

(可以用在测试自动化上。))

以上重点: 是正则表达式的运用。还有是网页规律查找, 利用工具拦截post,get的参数。

9、模块和包

10、数据结构和算法初步训练: 穷举法、递推法, 二分法查找, 排序等训练。

11、web开发:

先从简单的[web.py](#)开发为例, 分页, 表单提交, 修改, 删除。用户注册, 认证, 验证码, 注册用户邮件激活, 文章评论, 后台管理等。

django开发。

12、python图形界面编程。

13、python-PIL图像编程

14、python-pygame游戏编程简介。

15、python调试和测试。

python调试

python测试: doctest和UnitTest

16、python和运维。

Paramiko和fabric等的应用。

案例：

- 1、群发邮件系统的实现。
- 2、网页采集实战。
- 3、在线黄页查询实现。（[web.py](#)实现完整网站）
- 4、django,web开发实战（分页，查询，用户注册，登陆，退出，文章评论，后台管理等）。

教材：

- 1、自编的讲义。
- 2、参考书目：

OReilly.Learning.Python.3rd.Edition.Oct.2007.pdf

Python 核心编程 第二版
Data Structures and Algorithms Using Python

提供培训形式：
视频+作业+答疑。

咨询:qq:1465376564 电话：010-68165761 黄老师