

python基础:

1. 让用户输入姓名, 年龄, 性别, 然后打印一句话"我叫: , 今年: , 性别: "

```
name = input("名字: ")
age = input("年龄: ")
sex = input("性别: ")
msg = "我叫: " + name + ", 今年: " + age + ", 性别: " + sex      #字符串拼接成一句话
# '我叫: '等文字需要用引号引起来, 变成字符串, 才可以和name等字符串拼接
print(msg)
```

2. 输出1-100之间所有的数字

```
代码1:
count = 1
flag = True
while flag:
    print(count)
    count = count+1
    if count == 101:
        flag = False

代码2:
count = 1
while count<=100 :
    print(count)
    count += 1
```

3. 1+2+3+.....+100=?

```
n = 0
count = 1
while count<101:
    n = n + count
    count += 1
print(n)
```

4. 打印1-100之间的偶数?

```
代码1:
count = 1
while count<=100 :
    if count % 2 == 0 :
        print(count)
    count += 1

代码2:
count = 2
while True :
    print(count)
    count += 2
    if count == 102 :
        break      #循环中遇到break直接退出
```

5. 用print打印出下面内容:文能提笔安天下，武能上马定乾坤。心存谋略何人胜，古今英雄唯是君。

```
msg = '''文能提笔安天下，
        武能上马定乾坤。
        心存谋略何人胜，
        古今英雄唯是君。''' #多行用三引号
print(msg)
```

6. 利用if语句写出猜大小的游戏：设定一个理想数字 比如: 66, 让用户输入数字，如果比66大，则显示猜测的结果大了;如果比66小，则显示猜测的结果小了;只有等于66,显示猜测结果正确。

```
n = 66
guess = int(input("请输入你的猜测: "))
if guess > n :
    print("猜大了")
elif guess < n :
    print("猜小了")
else:
    print("猜对了")
```

7. 在上面第6题的基础上进行升级：给用户三次猜测机会，如果三次之内猜测对了，则显示猜测正确，退出循环，如果三次之内没有猜测正确，则自动退出循环，并显示‘太笨了你....’。

```
n = 66
count = 0
while count < 3:
    guess = int(input("请输入你的猜测: "))
    if guess > n :
        print("猜大了")
    elif guess < n :
        print("猜小了")
    else:
        print("猜对了")
        break
    count += 1
else:
    print("你太笨了")
```

8. 使用while循环输出 1 2 3 4 5 6 8 9 10

```
count = 1
while count < 11 :
    if count == 7:
        pass
    else:
        print(count)
    count += 1

count = 0
while count < 10:
    count += 1
    if count == 7:
        continue
    print(count)
```

```
count = 1
while count < 11:
    if count == 7:
        count += 1
    print(count)
    count += 1
```

9. 求1-2+3-4+5 ... 99的所有数的和?

```
count = 1
s = 0
while count < 100:
    if count % 2 == 0:
        s -= count
    else:
        s += count
    count += 1
print(s)
```

10. 用户登录（三次输错机会）且每次输错误时显示剩余错误次数?（提示：使用字符串格式化）

```
username = "aticc"
password = 123456
count = 1
while count < 4:
    name = input("请输入用户名: ")
    passwd = int(input("请输入密码: "))
    if name == username and passwd == password :
        print("欢迎%s登录"%name) #字符串格式化，注意两者之间没有逗号隔开
    else:
        print("用户名或者密码错误，还剩%s次机会"%(3-count))
    count += 1
```

11. 使用while和for循环分别打印字符串s="asdfer"中每个元素。

```
s = "asdfer"
for i in s:
    print(i)

s = "asdfer"
count = 0
while count < len(s):
    print(s[count])
    count += 1

#使用for循环对s="asdfer"进行循环，但是每次打印的内容都是"asdfer"。
s = "asdfer"
for i in s:
    print(s) #直接打印s

# 使用for循环对s="abcdefg"进行循环，每次打印的内容是每个字符加上sb， 例如： asb, bsb,
csb,...gsb
s="abcdefg"
for i in s:
    print(i + 'sb')
```

12. 使用for循环对s="321"进行循环, 打印的内容依次是: "倒计时3秒", "倒计时2秒", "倒计时1秒", "出发! "。

```
s = '321'
for i in s:
    print('倒计时{}秒'.format(i))
print('出发! ')
```

13. 实现一个整数加法计算器(两个数相加): 如: content = input("请输入内容:") 用户输入: 5+9或5+9或5 + 9, 然后进行分割再进行计算。

```
content = input("请输入内容:")
l1 = content.split('+')
#print(l1)
result = int(l1[0].strip()) + int(l1[1].strip()) #其实strip可以去掉, 因为字符串转换成数字, 最终得到的就是一个数字
print(result)
```

#选做题: 实现一个整数加法计算器(多个数相加): 如: content = input("请输入内容:") 用户输入: 5+9+6 +12+ 13, 然后进行分割再进行计算。

```
content = input("请输入内容:")
l1 = content.split('+')
#print(l1)
result = 0
for i in l1:
    result = result + int(i)
print(result)
```

14. 计算用户输入的内容中有几个整数(以个位数为单位), 如fhda1234slfh9876+=9fjdl有几个整数?

```
content = input("请输入内容: ")
result = 0
for i in content:
    #print(i, type(i))
    if i.isdecimal(): #字符串只由十进制组成
        result += 1
print(result)
```

15. 写代码: 计算 1 - 2 + 3 ... + 99 中除了88以外所有数的总和?

```
count = 0
n = 0
while count < 99:
    count += 1
    if count == 88:
        continue
    if count % 2 == 0:
        n -= count
    else:
        n += count
print(n)
```

16. 选做题: 判断一句话是否是回文。回文: 正着念和反着念是一样的。例如, 上海自来水来自海上

```

content = input('请输入: ')
if content == content[::-1]:    #content[::-1], 字符串的切片、倒序
    print('它是回文')
else:
    print('它不是回文')

```

17. 利用下划线将列表的每一个元素拼接成字符串"alex_wusir_taibai", l1 = ["alex", "wusir", "taibai"]

```

l1 = ["alex", "wusir", "taibai"]
代码一:
s = '_'.join(l1)
print(s)          #alex_wusir_taibai
代码二:
s = ''            #先定义一个空字符串
for i in l1:
    s = s + '_' + i
print(s)          #_alex_wusir_taibai
print(s[1:])      #alex_wusir_taibai

```

18. 查找列表li中的元素, li = ["TaiBai ", "alexC", "AbC ", "egon", " riTiAn", "WuSir", " aqc"], 移除每个(想到用for循环)元素的空格(想到用strip), 并找出以"A"或者"a"开头(想到用startswith或者其它), 并以"c"结尾的所有元素(想到用endswith或者其它), 并添加到一个新列表(想到创建一个new_l = [])中,最后循环(想到用for循环)打印这个新列表。

```

li = ["TaiBai ", "alexC", "AbC ", "egon", " riTiAn", "WuSir", " aqc"]
new_list = []
for i in li:
    new_i = i.strip()    # i.strip()操作是形成一个新的字符串, 所以需要重新赋值
    if (new_i.startswith('A') or new_i.startswith('a')) and
new_i.endswith('c'):    #注意and前面那个括号
    #或者if new_i.upper().startswith('A') and new_i.endswith('c'):
    #或者if new_i[0].upper() == 'A' and new_i[-1] == 'c':
        new_list.append(new_i)
for i in new_list:
    print(i)

```

19. 利用for循环和range, 将1-30的数字一次添加到一个列表中, 并循环这个列表, 将能被3整除的数改成*。

```

l1 = []
count = 0
for i in range(1,31):
    l1.append(i)
for x in l1:
    if x % 3 == 0:
        l1[count] = '*'
        count += 1
print(l1)

```

20. (难题) 开发敏感词语过滤程序, 提示用户输入评论内容, 如果用户输入的内容中包含特殊的字符: 敏感词列表 li = ["苍老师", "东京热", "武藤兰", "波多野结衣"], 则将用户输入的内容中的敏感词汇替换成同等长度的*, 并添加到一个列表中; 如果用户输入的内容没有敏感词汇, 则直接添加到上述的列表中。

```

li = ["苍老师", "东京热", "武藤兰", "波多野结衣"]
comment_list = []
comment = input('请输入您的评论: ') # 输入这个评论测试: 武藤兰hd苍老师skafh苍老师sdf苍老师fdslkafjl东京热波多野结衣。
for word in li:
    if word in comment: #这句话可以省略不写, 见最下面分析, 最好不要省。
        comment = comment.replace(word, '*' * len(word)) #字符串 '*' 可以与数字相乘, 乘以自己长度就行; 还要注意这个重新赋值, 只能重新赋值给自己comment, 如果重新赋值给新变量会出错, 见下面分析。
comment_list.append(comment) #放在for循环外面。
print(comment_list) #['***hd***skafh***sdf***fdslkafjl*****']

#出错的分析:
li = ["苍老师", "东京热", "武藤兰", "波多野结衣"]
comment_list = []
comment = input('请输入您的评论: ') # 输入这个评论测试: 武藤兰hd苍老师skafh苍老师sdf苍老师fdslkafjl东京热波多野结衣。
for word in li:
    if word in comment:
        c = comment.replace(word, '*' * len(word))
        # 第一次循环: c = '武藤兰hd***skafh***sdf***fdslkafjl东京热波多野结衣。'
        # 第二次循环: c = '武藤兰hd苍老师skafh苍老师sdf苍老师fdslkafjl***波多野结衣。'
        # .....
        # 最后一次循环: c = '武藤兰hd苍老师skafh苍老师sdf苍老师fdslkafjl东京热*****。'
comment_list.append(c)
print(comment_list) #['武藤兰hd苍老师skafh苍老师sdf苍老师fdslkafjl东京热*****。']

#if 那句话不写分析:
s1 = 'fdsaaalex'
print(s1.replace('太白', 'barry')) #结果: fdsaaalex, '太白' 不在里面也不会报错。

```

21. 有如下列表 (选做题) , li = [1, 3, 4, "alex", [3, 7, 8, "TaiBai"], 5, "RiTian"]
循环打印列表中的每个元素, 遇到列表则再循环打印出它里面的元素。

我想要的结果是:

```

1
3
4
alex
3
7
8
TaiBai
5
RiTian

```

```

li = [1, 3, 4, "alex", [3, 7, 8, "TaiBai"], 5, "RiTian"]
for i in li:
    if type(i) == list:
        for j in i:
            print(j)
    else:
        print(i)

```

22. 请将列表中的每个元素通过 "_" 链接起来。users = ['李少奇','李启航','渣渣辉']

```
users = ['李少奇', '李启航', '渣渣辉']
s = '_'.join(users)
print(s)    #李少奇_李启航_渣渣辉

#变形题：请将列表中的每个元素通过 "_" 链接起来。users = ['李少奇', '李启航', 666, '渣渣辉']
方法一：（适合列表里整数比较少）
users = ['李少奇', '李启航', 666, '渣渣辉']
users[2] = str(users[2])    #或者users[-2] = '666'
d = '_'.join(users)
print(d)    #李少奇_李启航_666_渣渣辉

方法二：（适合列表里整数很多时，不过打印结果要切片）
users = ['李少奇', '李启航', 666, '渣渣辉']
s = ''    #先定义一个空字符串
for i in users:
    s = s + '_' + str(i)
print(s)    #_李少奇_李启航_666_渣渣辉
print(s[1:])    #李少奇_李启航_666_渣渣辉

方法三：（不用切片打印，这种方法最好）
users = ['李少奇', '李启航', 666, '渣渣辉']
s = ''
for index in range(len(users)):
    if index == 0:
        s = s + str(users[index])
    else:
        s = s + '_' + str(users[index])
print(s)    #李少奇_李启航_666_渣渣辉
```

23. 请将元组 v1 = (11,22,33,44,55,66,77,88,99) 中的所有偶数索引位置的元素 追加到列表 v2 = [44,55,66] 中。

```
方法一：老师讲解的
v1 = (11,22,33,44,55,66,77,88,99)
v2 = [44,55,66]
print(v1[::2])    #(11, 33, 55, 77, 99)
v2.extend(v1[::2])
print(v2)    #[44, 55, 66, 11, 33, 55, 77, 99]

方法二：我写的
v1 = (11,22,33,44,55,66,77,88,99)
v2 = [44,55,66]
for i in range(len(v1)):
    if i % 2 == 0:
        # print(v1[i])
        v2.append(v1[i])
print(v2)
```

24. 将字典的键和值分别追加到 key_list 和 value_list 两个列表中，如：key_list = [], value_list = [], info = {'k1':'v1','k2':'v2','k3':'v3'}。

```
方法一：老师的
key_list = []
```

```

value_list = []
info = {'k1': 'v1', 'k2': 'v2', 'k3': 'v3'}
key_list.extend(info.keys())
value_list.extend(info.values())
print(key_list)      #['k1', 'k2', 'k3']
print(value_list)    #['v1', 'v2', 'v3']

```

方法二：我的

```

key_list = []
value_list = []
info = {'k1': 'v1', 'k2': 'v2', 'k3': 'v3'}
for key,value in info.items():
    key_list.append(key)
    value_list.append(value)
print(key_list)
print(value_list)

```

25. 有字符串"k: 1|k1:2|k2:3 |k3 :4" 处理成字典 {'k':1,'k1':2,...} (注意字典中的值是int类型)

```

msg = "k: 1|k1:2|k2:3 |k3 :4"
'''
msg 按照 | 分隔: split ['k: 1','k1:2','k2:3 ', 'k3 :4']
对每个元素进行操作想到用 for循环 ...
'''

l1 = msg.strip().split('|')
dic = {}
#print(l1)          #['k: 1', 'k1:2', 'k2:3 ', 'k3 :4']
for i in l1:
    key,value = i.split(":") # ['k', ' 1']
    dic[key.strip()] = int(value)
print(dic)           #{'k': 1, 'k1': 2, 'k2': 3, 'k3': 4}

```

26. 输出商品列表，用户输入序号，显示用户选中的商品

```

"""
商品列表：
goods = [
    {"name": "电脑", "price": 1999},
    {"name": "鼠标", "price": 10},
    {"name": "游艇", "price": 20},
    {"name": "美女", "price": 998}
]
要求：
1: 页面显示 序号 + 商品名称 + 商品价格，如：
    1 电脑 1999
    2 鼠标 10
    ...
2: 用户输入选择的商品序号，然后打印商品名称及商品价格
3: 如果用户输入的商品序号有误，则提示输入有误，并重新输入。
4: 用户输入Q或者q，退出程序。
"""

```

```

'''
想显示每个商品，用for循环
输入选择的商品序号，用input

```


输入序号，打印的商品名称以及价格
输入有误， 重新输入 用while循环
Q或者q，用upper()，break
'''

```
goods = [  
    {"name": "电脑", "price": 1999},  
    {"name": "鼠标", "price": 10},  
    {"name": "游艇", "price": 20},  
    {"name": "美女", "price": 998}  
]
```

#方式一：打印序号、商品名称以及价格

```
count = 1  
for i in goods:  
    print(count,i['name'],i['price'])  
    count += 1
```

#方式二：打印序号、商品名称以及价格

```
for index in range(len(goods)):  
    print(index+1,goods[index]['name'],goods[index]['price'])
```

'''

补充知识点：

```
l = ['a','b','c']  
for i in enumerate(l):  
    print(i)    #(0, 'a') (1, 'b') (2, 'c')  
for j in enumerate(l,start=100):  
    print(j)    #(100, 'a') (101, 'b') (102, 'c')
```

```
for i in enumerate(goods):  
    print(i)
```

```
for num,dic in enumerate(goods):  
    print(num,dic)
```

#方式三：打印序号、商品名称以及价格

```
for num,dic in enumerate(goods):  
    print('{}\t{}\t{}'.format(num+1,dic['name'],dic['price']))  
'''
```

#正式代码：

```
while 1 :  
    for num,dic in enumerate(goods):  
        print('{}\t{}\t{}'.format(num+1,dic['name'],dic['price']))  
    choice_num = input("请输入商品序号:").strip()    #用input，一般都加strip，把前  
    后空格去掉  
    if choice_num.isdecimal():    #判断是正整数  
        choice_num = int(choice_num)  
        if 0< choice_num <= len(goods):  
            print("您选择的商品名为{},价格为{}元".format(goods[choice_num-1]  
['name'],goods[choice_num-1]['price']))  
        else:  
            print("您输入的超出了范围，请重新输入")  
    elif choice_num.upper() == 'Q':  
        break  
    else:  
        print("您输入了非数字元素，输入有误")
```

27. 看代码写结果，并解释每一步的流程。(重要，注意这道题的思想：如果字典中没有某个键值对，先创建一个键值对，然后再对键值对进行操作)

```
v1 = [1,2,3,4,5,6,7,8,9]
v2 = {}

for item in v1:
    if item < 6:
        continue
    if 'k1' in v2:
        v2['k1'].append(item)
    else:
        v2['k1'] = [item ]
print(v2)
```

28. 看代码写结果 (深浅copy知识理解)

```
import copy

v1 = [1,2,3,{"name":'太白',"numbers":[7,77,88]},4,5]

v2 = copy.deepcopy(v1)

print(v1 is v2)

print(v1[0] is v2[0])
print(v1[3] is v2[3])

print(v1[3]['name'] is v2[3]['name'])
print(v1[3]['numbers'] is v2[3]['numbers'])
print(v1[3]['numbers'][1] is v2[3]['numbers'][1])
```

29. 文件a.txt内容：每一行内容分别为商品名字，价钱，个数。（注意：每一行是规律的，相隔一个空格）

```
apple 10 3
tesla 100000 1
mac 3000 2
lenovo 30000 3
chicken 10 3
```

通过代码，将其构建成这种数据类型：[{'name':'apple','price':10,'amount':3}, {'name':'tesla','price':100000,'amount':1},.....] 并计算出总价钱。

```
#方法一：
#先自己创建a.txt文件
l1 = []
with open('a.txt',encoding='utf-8') as f1:
    for line in f1:
        dic = {} #注意空字典创建在for循环里面
        #print(line,type(line))
```

```

    #print(line.strip())      #不确定的话自己先打印出来看看
    line = line.strip()      #先去除空格、换行符等  'apple 10 3'
    line_list = line.split() #默认按照空格分隔成列表 ['apple', '10', '3']
    # print(line_list)
    dic['name'] = line_list[0]
    dic['price'] = int(line_list[1])
    dic['amount'] = int(line_list[2])
    l1.append(dic)
print(l1)

#方法二：（没有实质性改变，只是简化了）
#先自己创建a.txt文件
l1 = []
with open('a.txt',encoding='utf-8') as f1:
    for line in f1:
        line = line.strip() # 'apple 10 3'
        line_list = line.split() # ['apple', '10', '3']
        dic = {'name':line_list[0],'price':int(line_list[1]),'amount':
int(line_list[2])}
        l1.append(dic)
print(l1)

# 上面两个代码：如果数据的列数增加了，就得给字典手动增加一个相应的键值对，麻烦，下面代码可
解决，列数增加，只需在列表name_list加一下键就行。
l1 = []
name_list = ['name', 'price', 'amount']
with open('a.txt',encoding='utf-8') as f1:
    for line in f1:
        line_list = line.strip().split()
        # print(line_list) # ['apple', '10', '3']
        dic = {}
        for index in range(len(name_list)):
            dic[name_list[index]] = line_list[index]
        l1.append(dic)
print(l1)

```

30. 利用python代码构建一个这样的列表(升级题):

```

[['_', '_', '_'], ['_', '_', '_'], ['_', '_', '_']]

```

```

l = []
for i in range(3):
    l.append(['_']*3) #列表可以和数字相乘, ['_']*3 == ['_', '_', '_']
print(l)

```

31. 用户输入一个数字，判断一个数是否是水仙花数。水仙花数是一个三位数, 三位数的每一位的三次方的和还等于这个数. 那这个数就是一个水仙花数,例如: $153 = 1^3 + 5^3 + 3^3$ 。

```

num = input("请输入三位数: ")
count = 0
if num.isdecimal():
    for i in num:
        count += int(i)**3
    if count == int(num):
        print("是水仙花数")
    else:
        print("不是水仙花数")
else:
    print("请输入纯数字")

```

32. 把列表中所有姓周的人的信息删掉(此题有坑, 请慎重): lst = ['周老二', '周星星', '麻花藤', '周扒皮'], 结果: lst = ['麻花藤']。

```

#错误代码（入坑了）：
lst = ['周老二', '周星星', '麻花藤', '周扒皮']
for i in lst:
    if '周' in i:
        lst.remove(i)
print(lst)    #['周星星', '麻花藤']
#错误原因：循环一个列表的时，最好不要改变列表的大小（增加值，或者删除值），这样会影响你的最终的结果。

#正确代码：倒序法删除
lst = ['周老二', '周星星', '麻花藤', '周扒皮']
for i in range(len(lst)-1,-1,-1):
    if lst[i].strip()[0] == "周":
        lst.pop(i)
print(lst)    #['麻花藤']

```

33. 车牌区域划分, 现给出以下车牌. 根据车牌的信息, 分析出各省的车牌持有量. (选做题)

```

cars = ['鲁A32444','鲁B12333','京B8989M','黑C49678','黑C46555','沪 B25041'.....]
locals = {'沪':'上海', '黑':'黑龙江', '鲁':'山东', '鄂':'湖北', '湘':'湖南'.....}
结果: {'黑龙江':2, '山东': 1, '北京': 1}

```

```

#方法一：
cars = ['鲁A32444', '鲁B12333', '京B8989M', '黑C49678', '黑C46555', '沪B25041']
locals = {'沪':'上海', '黑':'黑龙江', '鲁':'山东', '鄂':'湖北', '湘':'湖南', '京':
'北京'}
dic = {}
for i in cars:
    if locals[i[0]] not in dic:
        dic[locals[i[0]]] = 1    #第一次 {'山东': 1}
    else:
        dic[locals[i[0]]] = dic[locals[i[0]]] + 1
print(dic)    #{'山东': 2, '北京': 1, '黑龙江': 2, '上海': 1}

#本题重要思想：1）在循环中给字典添加键值对 2）没有键，就创建一个键，有此键就+1

#方法二：（方法二只适合本题，方法一是一种思想，以后会遇到很多题都要用到方法一）
cars = ['鲁A32444', '鲁B12333', '京B8989M', '黑C49678', '黑C46555', '沪B25041']
locals = {'沪':'上海', '黑':'黑龙江', '鲁':'山东', '鄂':'湖北', '湘':'湖南', '京':
'北京'}

```

```
dic = {}
for i in cars:
    dic[locals[i[0]]] = dic.get(locals[i[0]],0) + 1
    '''
    第一次循环:
        i ----> '鲁A32444'
        i[0] ----> '鲁'
        locals[i[0]] == locals['鲁'] == '山东'
        dic.get('山东',0) + 1 == 1
        dic[locals[i[0]]] == dic['山东']
        dic['山东'] = 1
    第二次循环:
        i ----> '鲁B12333'
        i[0] ----> '鲁'
        locals[i[0]] == locals['鲁'] == '山东'
        dic.get('山东',0) + 1 == 1 + 1 == 2
        dic[locals[i[0]]] == dic['山东']
        dic['山东'] = 2
    '''
print(dic)  #{'山东': 2, '北京': 1, '黑龙江': 2, '上海': 1}
```