

# 副本 山脚关课后题答案

## 第0关

### 练习-打印皮卡丘-参考

请你使用\*\*print()函数\*\*和\*\*""\*\*将下面的皮卡丘打印出来。皮卡丘字符画可以直接复制步骤1里的。

```
...
    ^    /|
  /\7    7_/_/
  /  |   /  /
  |  Z_,<  /  /\
  |    \  / }
  Y    \  / /
4●    、    ●  <=> { /
() ^    |    \ {
>- _  イ    |  //
/ ^    /  /<| \ \
\_/ ( _/    |  //
7      | /
>-r-----_ _
...
```

### 题目讲解

1.复制题目要求打印的“皮卡丘”（直接复制下一步中的字符画即可）

2. 用print(“” ”)打印

**\*\*注意：标点符号都要用英文\*\***

### 【参考】

```
1 print(''
2     ^    /|
3   /\7    7_/_/
4   /  |   /  /
5   |  Z_,<  /  /\
6   |    \  / }
7   Y    \  / /
8   4●    、    ●  <=> { /
9   () ^    |    \ {
10  >- _  イ    |  //
11  / ^    /  /<| \ \
12  \_/ ( _/    |  //
13  7      | /
```

```
14 >-r-----`-- _
15 '''
```

## 练习-定个小目标-参考

### 【题目】

利用print()函数，打印出自己的学习目标和学习计划，以及对应的奖惩措施，即“flag学习法”

### 【提示】

“flag学习法”：规定时间——>目标——>惩罚和奖励措施。建议完成代码点击右上角分享到社交平台：如微信/朋友圈/QQ空间，让更多朋友见证你的进步。

### 【参考】

```
1 print('目标：用一个月的时间入门python，写出一个简单的项目!')
2
3 print('计划：每天一个小时学习Pyhton小课；每周复盘知识点；边学边练!')
4
5 print('惩罚：做不到，就给朋友圈所有点赞的亲友，每人发个一元红包!')
6
7 print('奖励：完成目标，奖励自己一款索尼蓝牙耳机!')
```

## 练习-听见无脸男的声音-参考

### 题目要求：

在千寻进入魔法世界的期间，有一位“无脸男”一直跟在她的身边。一开始，千寻对无脸男的跟踪感到害怕，直到她第一次听见了无脸男对她说的话后，才理解了他，并和他成为了朋友。

### 【讲解】

下面是无脸男对千寻说的话，请你把这段话打印出来，并且使用\*\*转义字符\*\*，把无脸男说的话按照正确格式打印：

千寻你好，人们叫我‘无脸男’  
这个世界的人都选择无视我  
只有你看到了我并和我打招呼  
我感到很孤单，很孤单  
你愿意和我成为朋友吗？

### 题目讲解

1.复制下面这段无脸男对千寻说的话：

千寻你好，人们叫我‘无脸男’ 这个世界的人都选择无视我 只有你看到了我并和我打招呼 我感到很孤单，很孤单 你愿意和我成为朋友吗？

2.提示：文本中有引号的地方需要用到转义字符 \’ ；需要换行的地方用 \n

### 【参考】

```
1 print("千寻你好，人们叫我'无脸男'\n这个世界的人都选择无视我\n只有你看到了我并和我打\n招呼\n我感到很孤单，很孤单\n你愿意和我成为朋友吗？")
```

## 第1关

### 练习-大话西游-参考

#### 【题目】

下面，是一串程序的运行结果。请根据运行结果来反推出它的代码应该怎么写。运行成功后，请你给代码加注释：

输入一个你错过的人：紫霞仙子

输入你最想对ta说的话：我爱你

输入一个时长(比如三分钟)：一万年

曾经有一份真挚的爱情放在我面前，那个人的名字是紫霞仙子

可我没有珍惜，如果上天能再给我一次机会，我会对ta说一句话：我爱你

如果非要给这句话加上一个期限，我希望是一万年

#### 【提示】

1. 利用input()函数获取用户输入的数据，并把这些数据放在一个变量中

示例：A = input('输入一个你错过的人：')

2. 在print()函数中，利用加号+实现字符串的拼接<br>

示例：print('曾经有一份真挚的爱情放在我面前，ta的名字是'+ A)

3. 代码添加注释，操作：#(文本内容)

#### 【示例】

A = input('输入一个你错过的人：') #用input()函数获取用户数据，并放到变量A里面

print('曾经有一份真挚的爱情放在我面前，ta的名字是'+A)

#通过加号，把获取到的数据A和字符串拼接起来，并用print()函数打印出来

运行结果：(输入 紫霞仙子)

输入一个你错过的人：紫霞仙子

曾经有一份真挚的爱情放在我面前，ta的名字是紫霞仙子

#### 【参考】

```
1 A = input('输入一个你错过的人：')      #用input()函数获取用户数据，并放到变量A里面
2 B = input('输入你最想对ta说的话：')    #用input()函数获取用户数据，并放到变量B里面
3 C = input('输入一个时长(比如三分钟)：')#用input()函数获取用户数据，并放到变量C里
4 面
5 print('曾经有一份真挚的爱情放在我面前，ta的名字是' + A)
6 #通过加号，把获取到的数据A和字符串拼接起来，并用print()函数打印出来
7 print('可我没有珍惜，如果上天能再给我一次机会，我会对他说三个字'+ B)
8 #通过加号，把获取到的数据B和字符串拼接起来，并用print()函数打印出来
9 print('如果非要给这句话加上一个期限，我希望是'+ C)
10 #通过加号，把获取到的数据C和字符串拼接起来，并用print()函数打印出来
```

### 练习-程序员的一人饮酒醉-参考

## 题目要求

在Python的魔法世界，最常用的数据类型有三种：字符串(str)、整数(int)和浮点数(float)。  
在\*\*数据拼接\*\*中，为了将不同的信息进行整合，可以使用\*\*拼接符号\*\*`+`。但是，如果数据非字符串类型，则无法进行拼接。  
此时，我们可以使用数据转换函数\*\*`str()`\*\*，将数据转换为字符串类型后，再进行拼接。

### 【讲解】

请运用下方变量，使用`str()`函数打印两句话。

第一句话：1人我编程累碎掉的节操满地堆

第二句话：2眼是bug相随我只求今日能早归

可使用变量：

```
number1 = 1, number2 = 2
```

```
unit1 = '人', unit2 = '眼'
```

```
line1 = '我编程累', line2 = '是bug相随'
```

```
sentence1 = '碎掉的节操满地堆', sentence2 = '我只求今日能早归'
```

## 题目讲解

- 1.将非字符串型变量转换为字符串类型
- 2.将字符串类变量拼接在一起

### 【讲解】

- 1.使用`str()`函数
- 2.使用拼接符号\*\*`+`
- 3.使用`print()`函数

### 【参考】

```
1 number1 = 1
2 number2 = 2
3 unit1 = '人'
4 unit2 = '眼'
5 line1 = '我编程累'
6 line2 = '是bug相随'
7 sentence1 = '碎掉的节操满地堆'
8 sentence2 = '我只求今日能早归'
9 first = str(number1) + unit1 + line1 + sentence1
10 second = str(number2) + unit2 + line2 + sentence2
11 print(first)
12 print(second)
```

## 练习-非酋的吐槽-参考

### 题目要求

在\*\*数据拼接\*\*中，为了将不同的信息进行整合，可以使用\*\*拼接符号\*\*`+`。但是，如果数据非字符串类型，则无法进行拼接。此时，我们可以使用数据转换函数\*\*`str()`\*\*，将数据转换为字符串类型后，再进行拼接。但是有的时候，针对不同类型的数据，我们需要经历多次转换，最后才能转为字符串类型。

### 【讲解】

请使用\*\*转换函数\*\*str()、int()、float()，运用下方变量，使用数据转换及数据拼接，输出结果：脸黑怪我咯7张蓝票一个SSR都没有

可使用变量：

slogan = '脸黑怪我咯'， number = '7.8'， unit = '张'， sentence = '蓝票一个SSR都没有'

### 题目讲解

- 1.由于变量中与小数形式的字符串，所以需要使用转换函数将此变量转换为浮点型
- 2.转换后使用int()函数取整
- 3.由于数据拼接需要变量为字符串形式，所以需要将变量转化为字符串类型
- 4.最后数据拼接，得出结果

#### 【讲解】

- 使用float()函数将小数形式的字符串变量转换
- 使用int()函数取整
- 使用str()函数将数据转换为字符串形式
- 使用\*\*+\*\*进行数据拼接

#### 【参考】

```
1 slogan = '脸黑怪我咯'
2 number = '7.8'
3 unit = '张'
4 sentence = '蓝票一个SSR都没有'
5 word = slogan + str(int(float(number))) + sentence
6 print(word)
```

## 第2关

### 练习-红豆-参考

#### 【题目】

请让这首诗实现换行的功能

```
print('《红豆》王维 红豆生南国，春来发几枝。 愿君多采撷，此物最相思。')
```

换行后，终端的运行效果如下：

```
《红豆》王维
红豆生南国，春来发几枝。
愿君多采撷，此物最相思。
```

#### 【提示】

可以使用转义字符 【\n】

#### 【参考】

```
1 print('《红豆》王维\n红豆生南国，春来发几枝。\n愿君多采撷，此物最相思。')
```

## 练习-手机啊手机-参考

### 【题目】

请输入每天花在手机上的娱乐时间 a，再把时间 a 乘以7，再打印--“您一周花在手机上的娱乐时间总共是xx小时”。

### 【提示】

1. 用input()函数获取用户数据
2. 使用int()函数把数据转换为整数，再乘以7
3. 使用str()函数把一周花在手机上的娱乐时间转换为字符串，用print() 函数输出。

### 【参考】

```
1 a = input('请输入你每天花在手机上的娱乐时间：')
2 b = int(a) * 7
3 print('您一周花在手机上的娱乐时间总共是'+str(b)+'小时')
```

## 练习-偷回宝石-参考

### 题目要求

在灭霸打了一个响指，宇宙一半生物都灰飞烟灭后，剩下的复仇者联盟成员们在巨大的创伤之中，依旧没有放弃反击灭霸的机会，他们决定利用最后一次行动机会，去把灭霸手里的宝石偷回来。最终，他们因为实力相差太大，1颗宝石都没有偷回来。

### 【讲解】

请你写出一段代码：

如果偷回的宝石数是4颗及以上，输出结果“获得了打败灭霸的力量，反杀稳了”

如果偷回的宝石数是1-3颗，输出结果“可以全员出动，殊死一搏”

如果偷回的宝石数是0颗，输出结果“没办法了，只能尝试呼叫惊奇队长”

注意：标点符号都为英文格式

### 题目讲解

- 1.为变量【赋值】，写出“一个宝石都没偷回来”的情况
- 2.用【if】写第一个条件：如果偷回的宝石数是4颗及以上，输出结果：获得了打败灭霸的力量，反杀稳了
- 3.用【elif】写第二个条件：如果偷回的宝石数是1-3颗，输出结果：可以全员出动，殊死一搏
- 4.用【else】写第三个条件：如果偷回的宝石数是0颗，输出结果：没办法了，只能尝试呼叫惊奇队长

### 【参考】

```
1 #以下为答案示例(stonenumber=0/1/2/3/4/5/6都是对的)
2 stonenumber=5
3 #偷回宝石数量赋值
4 if stonenumber >= 4:
5     #条件：如果你拥有的宝石数量大于等于4个
6     print('获取打败灭霸的力量')
7     #结果：显示‘获取打败灭霸的力量’的结果
```

```

8 elif 1 < stonenumber <= 3:
9     #条件: 如果想让宝石数量停留在4至5个
10    print('可以全员出动, 殊死一搏')
11    #结果: 显示‘可以全员出动, 殊死一搏’的结果
12 else:
13    #条件: 当赋值不满足if和elif条件时, 执行else下的命令, 宝石数量在3个以下
14    print('没办法了, 只能尝试呼叫惊奇队长')
15    #结果: 显示‘没办法了, 只能尝试呼叫惊奇队长’的结果

```

## 练习-美国队长的工资-参考

### 题目要求

复仇者联盟的成员也是有工资的, 然而, 由于美国队长一直被冰封在北极, 错过了多次调薪机会, 所以美国队长的工资一直是每月80美元。光荣挺进史塔克穷人榜前三名, 并获封“美元队长”称号。

### 【讲解】

请你写出一段代码, 判断美国队长的工资水平, 代码需要满足如下条件:

如果月工资小于等于500美元, 显示“欢迎进入史塔克穷人帮前三名”

如果月工资在100-500美元之间, 显示“请找弗瑞队长加薪”

如果月工资小于等于100美元, 显示“恭喜您荣获“美元队长”称号!”

如果月工资在500-1000美元之间(含1000美元), 打印“祝贺您至少可以温饱了。”

其他情况下, 如果工资大于1000美元, 打印“经济危机都难不倒您!”

如果工资在1000-20000美元(含20000美元)之间, 打印“您快比钢铁侠有钱了!”

如果月工资大于20000美元, 打印“您是不是来自于瓦坎达国?”

不管赋值改变后输出结果如何, 都需固定打印“程序结束”

### 题目讲解

1.根据题目, 我们知道这里涉及嵌套条件, 所以可以用扒洋葱法梳理代码逻辑

2.根据要求, 先写最外层的\*\*if .....elif .....else.....\*\*条件

3.根据题目的从属关系分析, 最外层的\*\*if\*\*条件和\*\*else\*\*条件都有额外条件

4.依次在外层基础条件下添加额外条件

### 【讲解】

1.使用if条件嵌套

2.嵌套条件需要缩进

3.注意单双引号同时存在时的互相转换

4.通过钱数输出对应的结果

### 【参考】

```

1 #以下为答案示例(salary为多少都是对的)
2 salary = 200
3 #对美国队长的工资赋值
4 if salary <= 500:
5     #条件: 如果工资小于等于500
6     print('欢迎进入史塔克穷人帮前三名')
7     #结果: 显示‘欢迎进入史塔克穷人帮前三名’的结果
8     if salary <= 100:

```

```

9         #条件: 如果工资小于等于100
10        print('恭喜您荣获“美元队长”称号! ')
11        #结果: 显示‘恭喜您荣获“美元队长”称号!’的结果
12    else:
13        #条件: 当条件满足第4行if, 不满足第8行if, 执行这个命令
14        print('请找弗瑞队长加薪')
15        #结果: 显示‘请找弗瑞队长加薪’的结果
16 elif 500 < salary <= 1000:
17     #条件: 如果工资大于500, 小于等于1000
18     print('祝贺您至少可以温饱了。')
19     #结果: 显示‘祝贺您至少可以温饱了。’的结果
20 elif salary > 1000:
21     #条件: 如果工资大于1000
22     print('经济危机都难不倒您! ')
23     #结果: 显示‘经济危机都难不倒您!’的结果
24     if 1000 < salary <= 20000:
25         #条件: 如果工资大于1000, 小于等于20000
26         print('您快比钢铁侠有钱了! ')
27         #结果: 显示‘您快比钢铁侠有钱了!’的结果
28     else:
29         #条件: 满足第20行elif条件, 不满足第24行if条件, 则执行这个命令
30         print('您是不是来自于瓦坎达国? ')
31         #结果: 显示‘您是不是来自于瓦坎达国?’的结果
32 print('程序结束')

```

## 第3关

### 练习-我要控制我自己-参考

#### 【背景】

最近, 派大星想要减肥, 他决定控制自己的饮食, 少吃一点蟹黄堡。海绵宝宝为了帮助好朋友派大星, 和派大星一起制定了一个饮食游戏。

游戏规则如下: 如果派大星一天吃超过40个蟹黄堡, 就要给海绵宝宝100块; 如果派大星一天吃小于等于40个的蟹黄堡, 海绵宝宝就给派大星100块。

#### 【题目】

请你写出一段代码: 当输入派大星一天吃的蟹黄堡数量时, 可以打印出这天该派大星给海绵宝宝钱, 还是海绵宝宝给派大星钱。

#### 【提示】

1. 使用input()函数获取获得派大星吃的蟹黄堡的数量。
2. 使用int() 函数强制类型转换, 将字符串转换为整数。
3. 使用if...else... 条件判断语句。

#### 【参考】

```

1 t = int(input('派大星今天吃了多少个蟹黄堡'))
2 if t > 40:

```



```
3     print('派大星给海绵宝宝100块')
4 else:
5     print('海绵宝宝给派大星100块')
```

## 练习-恋爱告白决策-参考

### 【题目】

使用Python代码来演示恋爱过程：

首先判断：两人的关系是不是达到了【朋友之上，恋人未满】？

如果不是，那么输出：进度条还不够，再等等。程序终结。

如果是，那么再判断：你是不是想和对方有进一步的发展？

如果不是，那么输出：还是做朋友吧。程序终结。

如果是，那么再判断：对方是不是想有进一步的发展？

如果不是，那么输出：恭喜获得“好人卡”。

如果是，那么输出：恭喜你们，有情人终成眷属！

### 【提示】

1. 条件判断的使用：if...elif...elif...else
2. 使用input()函数，不需要赋值也可以直接进行条件判断。比如：if input('吃了吗?')== '吃了':

### 【参考】

```
1 now = input('你们之前的感情有进入暧昧期了吗？回答有/没有：')
2
3 if now == '没有':
4     print('进度条还不太够，再等等吧。')
5 else:
6     p1 = input('你希望你们的关系再往前一步吗？回答想/不想：')
7     if p1 == '不想':
8         print('那还是做朋友吧。')
9     else:
10        p2 = input('那个人呢，ta想吗？回答想/不想：')
11        if p2 == '不想':
12            print('恭喜你！获得好人卡*1')
13        else:
14            print('恭喜你们！在一起！')
15
16 now = input('你们之前的感情有进入暧昧期了吗？回答有/没有：')
17 p1 = input('你希望你们的关系再往前一步吗？回答想/不想：')
18 p2 = input('那个人呢，ta想吗？回答想/不想：')
19
20 if now == '没有':
21     print('进度条还不太够，再等等吧。')
22 elif p1 == '不想':
23     print('那还是做朋友吧。')
24 elif p2 == '不想':
25     print('恭喜你！获得好人卡*1')
```

```

26 else:
27     print('恭喜你们! 在一起! ')
28
29 #提示是为了让大家更好写出代码, 其实更优雅和符合逻辑的代码是这样的:
30
31 if input('你们之前的感情有进入暧昧期了吗? 回答有/没有: ') == '没有':
32     print('进度条还不太够, 再等等吧。')
33 elif input('你希望你们的关系再往前一步吗? 回答想/不想: ') == '不想':
34     print('那还是做朋友吧。')
35 elif input('那个人呢, ta想吗? 回答想/不想: ') == '不想':
36     print('恭喜你! 获得好人卡*1')
37 else:
38     print('恭喜你们! 在一起! ')
39
40 #我们是可以在写出代码后再进行优化的。
41
42 now = input('你们之前的感情有进入暧昧期了吗? 回答有/没有: ')
43 if now == '没有':
44     exit('进度条还不太够, 再等等吧。')
45 p1 = input('你希望你们的关系再往前一步吗? 回答想/不想: ')
46 if p1 == '不想':
47     exit('那还是做朋友吧。')
48 p2 = input('那个人呢, ta想吗? 回答想/不想: ')
49 if p2 == '不想':
50     exit('恭喜你! 获得好人卡*1')
51 else:
52     print('恭喜你们! 在一起! ')
53 #第三种答案
54
55 now = input('你们之前的感情有进入暧昧期了吗? 回答有/没有: ')
56
57 if now == '没有':
58     print('进度条还不太够, 再等等吧。')
59 else:
60     p1 = input('你希望你们的关系再往前一步吗? 回答想/不想: ')
61     if p1 == '不想':
62         print('那还是做朋友吧。')
63     else:
64         p2 = input('那个人呢, ta想吗? 回答想/不想: ')
65         if p2 == '不想':
66             print('恭喜你! 获得好人卡*1')
67         else:
68             print('恭喜你们! 在一起! ')
69 #第四种答案

```

## 练习-测测你健康吗? -参考

### 【题目】

输入你的体重和身高，根据BMI指数的计算公式，输出不同BMI指数对应的身体状况（过轻、正常、过重、健康）。

### 【背景】

BMI指数是一个国际上常用的衡量人体胖瘦程度，以及是否健康的指数。

它的计算方式如下：  $BMI = \text{体重} \div (\text{身高} \times \text{身高})$  比如一个体重50kg，身高160cm的人，  $BMI = 50 \div (1.6 \times 1.6) = 19.5$

对照BMI的表格：

过轻	$BMI < 18.5$
正常	$18.5 \leq BMI < 24.0$
过重	$24.0 \leq BMI < 28.0$
肥胖	$28.0 \leq BMI$

by Python小课

### 【提示】

1. 使用条件判断语句if...elif...else的语句来写代码。
2. 使用input()函数获取输入的身高和体重数据。
3. 使用float()函数将字符串数据转换为浮点数。

### 【参考】

```
1 a = float(input('请输入你的体重(kg): '))
2 b = float(input('请输入你的身高(m): '))
3 bmi = a/(b*b)
4 if bmi >= 28.0:
5     print('肥胖')
6 elif bmi >= 24.0:
7     print('过重')
8 elif bmi >= 18.5:
9     print('正常')
10 else:
11     print('过轻')
12
13 #这并不是唯一的答案。
```

## 练习-哈利波特的宠物-参考

### 题目要求

哈利·波特在对角巷购买宠物时犯难了，他不知道该给自己的猫头鹰起什么名字。现在请你帮助哈利·波特，为他的猫头鹰起一个名字。

### 【讲解】

请你写一段代码：

为哈利·波特的猫头鹰起个名字，并打印出结果【哈利·波特的猫头鹰叫做XX】（XX是你起的名字）。

### 题目讲解

- 1.搜集猫头鹰的名字（知识点：input(),赋值）
- 2.打印结果【哈利·波特的猫头鹰叫做XX】（XX是你取的名字）（知识点：数据拼接）

### 【参考】

```
1 name = input('为哈利·波特的猫头鹰起个名字：')
2 print('哈利·波特的猫头鹰叫做' + name)
```

## 练习2-罗恩的减肥计划-参考

### 题目要求

为了三强争霸赛的舞会，罗恩想要减肥，所以他决定控制自己的饮食，少吃一点巧克力蛙。

哈利为了帮助好朋友罗恩，和他一起制定了一个饮食游戏。

游戏规则如下：

如果罗恩一天吃超过10个巧克力蛙，罗恩要给哈利100块；

如果罗恩一天吃小于等于10个的巧克力蛙，哈利就给罗恩100块。

### 【讲解】

请你写出一段代码：

当随机输入罗恩一天吃的巧克力蛙数量时，可以判断出这天，是罗恩给哈利钱，还是哈利给罗恩钱，并打印出来。

### 题目讲解

- 1.搜集罗恩吃的巧克力数量（知识点：input()、数据转换）
- 2.写第一个条件：如果罗恩一天吃超过10个巧克力蛙，罗恩要给哈利100块；
- 3.写第二个条件：如果罗恩一天吃小于等于10个的巧克力蛙，哈利就给罗恩100块。  
（知识点：if...else...）

### 【参考】

```
1 number = int(input('请输入罗恩吃的巧克力数量：'))
2 #int将其转换为整数，之后好进行大小的比较
3 if number > 10:
4     print('罗恩要给哈利100块')
5 else:
6     print('哈利就给罗恩100块')
7
```

## 练习3-古灵阁金币兑换-参考

### 题目要求

接受了霍格沃茨来信的你，带着欣喜与好奇，跟着魔咒学老师吴枫教授，来到了巫师世界的对角巷。在这个相当于人类集市的对角巷，你立刻被魔法世界稀奇古怪的东西吸引，想掏手机刷花呗买买买。但是吴枫教授说，麻瓜（没有魔法的人）货币在魔法世界不流通，但是你可以去巫师世界的银行——古灵阁兑换货币。你立刻跟着吴枫老师跑到了古灵阁。当你进到巫师世界的银行时，就有银行的小精灵职员问好：

**\*\*注意：以下背景与题目要求相关\*\***

小精灵：您好，欢迎古灵阁，请问您需要帮助吗？需要or不需要？

你：需要

小精灵：请问您需要什么帮助呢？1 存取款；2 货币兑换；3 咨询

你：2

小精灵：金加隆和人民币的兑换率为1:51.3，即一金加隆=51.3人民币

小精灵：请问您需要兑换多少金加隆呢？

（你说了一个数字）

小精灵：好的，我知道了，您需要兑换（你刚说的数字）金加隆。

小精灵：那么，您需要付给我（将你刚说的数字转换为金加隆的数字）人民币。

注1：如果选择不需要帮助，小精灵会礼貌地说‘好的，再见。’

注2：如果选择帮助【1 存取款】，小精灵会推荐你去存取款窗口；如果选择帮助【3 咨询】，小精灵会推荐你去咨询窗口。

### 题目讲解

和小精灵的对话可以被这么拆分：

- 1.最基础的条件判断：需要帮助；不需要帮助（if...else...）
- 2.在需要帮助的情况下，又分为1存取款；2货币兑换；3咨询；三种情况。（if...elif...else...）
- 3.在2-货币兑换的情况下，需要再提一个问题，收集答案，打印出小精灵回答的结果（input()、数据拼接、数据转换）。

### 【参考】

```
1 chioce = input('您好，欢迎古灵阁，请问需要帮助吗？需要or不需要？')
2 if chioce == '需要':
3     number = input('请问您需要什么帮助呢？1 存取款；2 货币兑换；3 咨询')
4     if number == '2':
5         print('金加隆和人民币的兑换率为1:51.3，即一金加隆=51.3人民币')
6         print('请问您需要兑换多少金加隆呢？')
7         money = input('请输入你需要兑换的金加隆')
8         print('好的，我知道了，您需要兑换' + money + '金加隆。')
9         print('那么，您需要付给我'+str(int(float(money)*51.3))+ '人民币。')
10
11     elif number == '1':
12         print('请到存取款窗口办理')
13
14     elif number == '3':
15         print('请到咨询窗口咨询')
```

```

16
17     else:
18         print('输入错误，没有你需要的服务')
19
20 elif chioce == '不需要':
21     print('好的，再见')
22
23 else:
24     print('输入错误')

```

## 第4关

### 练习-君子爱‘数’取之有道-参考

#### 题目讲解

第一步：取出列表中的第三个元素（list1[2]），字典{'爱':'love'}

第二步：取出list1[2]中键‘爱’所对应的值，即‘love’（list1[2]['爱']）

#### 【参考】

```

1 list1 = [{'嫉妒':'envy'},{'恨':'hatred'},{'爱':'love'}]
2 print(list1[2]['爱'])

```

#### 题目讲解

第一步：取出字典中键为3对应的值（dict1[3]），即['love','hatred','envy']

第二步：再取出列表['love','hatred','envy']中的第一个元素（dict1[3][0]）

#### 【参考】

```

1 dict1 = {1:['cake','scone','puff'],2:['London','Bristol','Bath'],3:
2         ['love','hatred','envy']}
3 print(dict1[3][0])

```

#### 题目讲解

第一步：元组和列表都是序列,提取的方式也是偏移量，如 tuple1[1]、tuple1[1:]。

第二步：先取出列表中的第二个元素中的（list2[1]）

第三步：然后在第二个元素的基础上在去取值，即list2[1][1]

注：（元组和列表的取值方式是相同的）

#### 【参考】

```

1 tuple1 = ('A','B')
2 list2 = [('A','B'),('C','D'),('E','F')]
3
4 print(tuple1[0])
5 print(list2[1][1])

```

## 练习-找到那只狼-参考

### 【讲解】

明确目标很重要（所以重复三遍）。

做到后面的步骤，可再点开左侧的“练习介绍”查看。<br>

#### 练习目标：

在层层嵌套的各种数据类型中，准确地提取出你需要的数据。<br>

#### 练习要求：

在未来世界里，一个新建的童话镇吸引了不少人入住。

不过，在人群里隐藏着一只狼，会威胁大家的安全。

童话镇的镇长希望你能找到它，并揭发其身份。

用程序语言就是说：列表中有个字符串是“狼”，将其打印出来吧。

### 【解答】

```
1 townee = [  
2     {'海底王国':['小美人鱼','海之王','小美人鱼的祖母','五位姐姐'],'上层世界':['王  
   子','邻国公主']},  
3     '丑小鸭','坚定的锡兵','睡美人','青蛙王子',  
4     [{'主角':'小红帽','配角1':'外婆','配角2':'猎人'},{'反面角色':'狼'}]  
5 ]  
6  
7 print(townee[5][1]['反面角色'])  
8
```

## 第5关

## 练习-数数字-参考

### 题目讲解

小美想知道你是怎么用 for循环 的，在下方写下你的代码吧~

小美还想知道你是怎么用 while循环 的，在下方写下你的代码吧~

### 【参考】

```
1 # while 循环  
2 n = 0  
3 while n < 7:  
4     n = n+1  
5     if n != 4: # 当num != 4，执行打印语句；等于4时不打印。  
6         print(n)
```

```
1 # for 循环
```

```
2 for num in range(1,8): # 为同时能运行两个循环，新取参数 num。
3     if num != 4: # 当num != 4，执行打印语句；等于4时不打印。
4         print(num)
```

## 练习-轮流坐前排-参考

### 【讲解】

明确目标很重要（所以重复三遍）。

做到后面的步骤，可再点开左侧的“练习介绍”查看。

### 练习目标：

通过这个练习，你会尝试用循环来解决生活中的问题，并了解一种新的列表方法。

### 练习要求：

小明、小红、小刚是同班同学，且坐在同一排，分别坐在第一位、第二位、第三位。

由于他们的身高都差不多，所以，老师计划让他们三个轮流坐在第一位。

每次换座位的时候，第一位变第三位，后面两位都往前一位。

方法1：append()函数

### 【提示】

这里需要用到列表的切片和索引。

### 【解答】

```
1 students = ['小明','小红','小刚']
2 for i in range(3):
3     student1 = students[0]
4     students = students[1:]
5     students.append(student1)
6     print(students)
```

参考代码中用的是 for 循环，当然，while 循环 也是可以的。

感兴趣的同学，可以两者都试试，然后对比一二。

### 方法2：pop()函数

我们先介绍一下列表中的pop()函数，用于移除列表中的一个元素（默认最后一个元素），并且返回该元素的值。

可以将其理解为提取和删除的融合：①提取：取到元素，对列表没有影响；②删除：删除列表的元素。

而移除，则是同时做到取到元素，并且删除列表中的元素。

### 【提示】

你可以先运行一下右侧的头几行代码感受一下，然后再运用pop()函数来满足“轮流坐前排”这个需求。

### 【解答】

```
1 students = ['小明','小红','小刚']
```



```
2 for i in range(3):
3     student1 = students.pop(0)
4     students.append(student1)
5     print(students)
6
```

相信你已经用了新学的方法完成了代码。可以发现：代码从6行减为5行，虽然仅仅1行代码的差别，还不能很好地体现“简化”代码的思想，但这是一种必备的编程思维。

当你发现要实现某功能要用较多代码实现，最先做的不是埋头打代码，而是找找是否有更好、更简洁的解决方法。

## 第6关

### 练习-囚徒困境-参考

#### 【讲解】

1.开启循环，两人分别选择

首先，我们需要知道两个囚徒各自的选择（input\*2）。

2.循环当中，有判断和跳出

两个囚徒的选择，会有四种不同的结果。所以，可以用'if ...elif ...else'。

#### 判决提示：

若两人都认罪，则两人各判10年；

若一个认罪一个抵赖，则认罪的人判1年，抵赖的人判20年；

若两人都抵赖，则各判3年——这种情况下跳出循环。

#### 【参考】

```
1 while True:
2     a = input('A, 你认罪吗？请回答认罪或者不认')
3     b = input('B, 你认罪吗？请回答认罪或者不认')
```

```
1 while True:
2     a = input('A, 你认罪吗？请回答认罪或者不认')
3     b = input('B, 你认罪吗？请回答认罪或者不认')
4     if a == '认罪' and b == '认罪':
5         print('两人都得判10年，唉')
6     elif a == '不认' and b == '认罪':
7         print('A判20年，B判1年，唉')
8     elif a == '认罪' and b == '不认':
9         print('A判1年，B判20年')
10    else:
11        print('都判3年，太棒了')
12        break # 当满足开头提到的条件时，跳出循环。
```

## 练习-记录困境中的选择-参考

### 练习要求

上一个练习，我们将“囚徒困境”写成了代码，让程序收集两名囚犯的认罪情况，进而决定他们的判决：

两人都认罪，则各判10年；一个认罪一个抵赖，则前者判1年，后者判20年；两人都抵赖，各判3年。只有两人都都不认罪，程序才会停止。

现在有一个社会学家，在不同的人群中做这个实验，一旦遇到都不认罪的情况，就停止该人群中的实验。

同时，他希望程序能记录每一对实验者的选择，以及记录第几对实验者都选择不认罪。请你帮帮他吧。

### 【讲解】

明确目标很重要（所以重复三遍）。

做到后面的步骤，可再点开左侧的“练习介绍”查看。

### 练习目标：

这个作业会建立在上一个练习的基础上，完善代码的功能。

### 几个提示：

为了记录每一对实验者的选择，需要用一个可拓展的“容器”来存放这些数据；

为了记录是第几对实验者做了最优选择，需要用一个变量来计数；

为了将每一对实验者的选择再打印出来，需要写一个循环。

在打印的那个循环里，需要留意列表是从0开始计数的。

### 【解答】

```
1  n = 0
2  list_answer = []
3
4
5  while True:
6      n += 1
7      a = input('A, 你认罪吗？请回答认罪或者不认：')
8      b = input('B, 你认罪吗？请回答认罪或者不认：')
9      list_answer.append([a,b])
10     if a == '认罪' and b == '认罪':
11         print('两人都得判10年，唉')
12     elif a == '不认' and b == '认罪':
13         print('A判20年，B判1年，唉')
14     elif a == '认罪' and b == '不认':
15         print('A判1年，B判20年')
16     else:
17         print('都判3年，太棒了')
18         break
```

```
19
20
21 print('第' + str(n) + '对实验者选了最优解。')
22
23
24 for i in range(n):
25     print('第' + str(i+1) + '对实验者的选择是: ' + str(list_answer[i]))
26
27
```

## 练习-演员的作品-参考

### 练习要求

我很喜欢看电影，我回忆了一下，这两年我觉得还不错的国产电影。

下面，会将电影的影片名和主演放在字典里，如`movie = {'妖猫传':['黄轩','染谷将太']}`。

需要你补充一些代码，让其他人只要输入演员名，就打印出：xx出演了电影xx。

### 【讲解】

明确目标很重要（所以重复三遍）。

做到后面的步骤，可再点开左侧的“练习介绍”查看。

### 练习目标：

通过这个练习，你会学会综合使用循环、数据提取和条件判断。

注：这个练习的提示比较完整，建议先多尝试几次独立完成。

### 【提示】

对该练习完整的提示：

- 1.先用for...in...遍历字典，把键（即电影名）取出来；
- 2.再用字典的键把字典里的值（即包含主演名字列表）取出来；
- 3.最后用if A in B 作为条件判断A在不在B中，即输入的演员名在不在这个列表中。如果在这个列表中，就打印出xx出演了电影xx。

### 【解答】

```
1 movies = {
2     '妖猫传':['黄轩','染谷将太'],
3     '无问西东':['章子怡','王力宏','祖峰'],
4     '超时空同居':['雷佳音','佟丽娅'],
5 }
6
7
8 actor = input('你想查询哪个演员? ')
9 for movie in movies:
10     actors = movies[movie]
11     if actor in actors:
12         print(actor + '出演了电影' + movie)
```

## 第7关

### 练习-再来一盘-参考

#### 【讲解】

要达成目标，有两种方案：

#### 1.while True+break:

开启一个无限循环，设定跳出条件。

当得到肯定回复时，继续运行；当得到否定回复时，运行break，停止循环，退出游戏。

#### 2.while 变量名+变量名的布尔值判断:

在开头设某变量的布尔值为True，input后开启判断变量的布尔值是否改变。

当得到肯定回复时，while True，继续运行；当得到否定回复时，while False，停止循环，退出游戏。下面，我们先用较简单的代码来体会一下这两种方案。

#### 方案1：while True+break

开启一个无限循环，设定跳出条件。

当得到肯定回复时，继续运行；当得到否定回复时，运行break，停止循环，退出游戏。

#### 方案2：while 变量名+变量名的布尔值判断

在开头设某变量的布尔值为True，input后开启判断变量的布尔值是否改变。

当得到肯定回复时，while True，继续运行；当得到否定回复时，while False，停止循环，退出游戏。

```
1 # 方案1
2 while True:
3     a1 = input('要继续游戏吗，请输入n退出，输入其他继续: ')
4     if a1 == 'n':
5         break
6
7 # 方案2
8 again = True
9 while again:
10    a2 = input('要继续游戏吗，请输入y继续，输入其他退出: ')
11    if a2 == 'y':
12        again = True
13    else:
14        again = False
15
```

#### 【参考】

```
1 #for循环
2 import time
```

```

3 import random
4
5
6
7 for i in range(1,4):
8     player_victory = 0
9     enemy_victory = 0
10    time.sleep(1.5)
11    print(' \n—————现在是第 %s 局—————' % i)
12    player_life = random.randint(100,150)
13    player_attack = random.randint(30,50)
14    enemy_life = random.randint(100,150)
15    enemy_attack = random.randint(30,50)
16
17    print('【玩家】\n血量: %s\n攻击: %s' % (player_life,player_attack))
18    print('-----')
19    time.sleep(1)
20    print('【敌人】\n血量: %s\n攻击: %s' % (enemy_life,enemy_attack))
21    print('-----')
22    time.sleep(1)
23
24    while player_life > 0 and enemy_life > 0:
25        player_life = player_life - enemy_attack
26        enemy_life = enemy_life - player_attack
27        print('你发起了攻击, 【玩家】剩余血量%s' % player_life)
28        print('敌人向你发起了攻击, 【敌人】的血量剩余%s' % enemy_life)
29        print('-----')
30        time.sleep(1.2)
31
32    if player_life > 0 and enemy_life <= 0:
33        player_victory += 1
34        print('敌人死翘翘了, 你赢了! ')
35    elif player_life <= 0 and enemy_life > 0:
36        enemy_victory += 1
37        print('悲催, 敌人把你干掉了! ')
38    else:
39        print('哎呀, 你和敌人同归于尽了! ')
40
41    if player_victory > enemy_victory :
42        time.sleep(1)
43        print('\n【最终结果: 你赢了! 】')
44    elif enemy_victory > player_victory:
45        print('\n【最终结果: 你输了! 】')
46    else:
47        print('\n【最终结果: 平局! 】')
48
49
50 #while 循环
51 import time
52 import random
53

```

```

54
55 while True:
56     player_victory = 0
57     enemy_victory = 0
58     for i in range(1,4):
59         time.sleep(1.5)
60         print(' \n—————现在是第 %s 局—————' % i)
61         player_life = random.randint(100,150)
62         player_attack = random.randint(30,50)
63         enemy_life = random.randint(100,150)
64         enemy_attack = random.randint(30,50)
65
66         print('【玩家】\n血量: %s\n攻击: %s' % (player_life,player_attack))
67         print('-----')
68         time.sleep(1)
69         print('【敌人】\n血量: %s\n攻击: %s' % (enemy_life,enemy_attack))
70         print('-----')
71         time.sleep(1)
72
73         while player_life > 0 and enemy_life > 0:
74             player_life = player_life - enemy_attack
75             enemy_life = enemy_life - player_attack
76             print('你发起了攻击,【玩家】剩余血量%s' % player_life)
77             print('敌人向你发起了攻击,【敌人】的血量剩余%s' % enemy_life)
78             print('-----')
79             time.sleep(1.2)
80
81             if player_life > 0 and enemy_life <= 0:
82                 player_victory += 1
83                 print('敌人死翘翘了,你赢了! ')
84             elif player_life <= 0 and enemy_life > 0:
85                 enemy_victory += 1
86                 print('悲催,敌人把你干掉了! ')
87             else:
88                 print('哎呀,你和敌人同归于尽了! ')
89
90         if player_victory > enemy_victory :
91             time.sleep(1)
92             print('\n【最终结果: 你赢了! 】')
93         elif enemy_victory > player_victory:
94             print('\n【最终结果: 你输了! 】')
95         else:
96             print('\n【最终结果: 平局! 】')
97
98     a1 = input('要继续游戏吗,请输入n退出,输入其他继续: ')
99     if a1 == 'n':
100         break

```

## 练习-再来一种-参考

### 练习目标

在这个练习，我们会学会一种新的“格式化字符串”的方法：format()函数。

### 练习要求

在项目1的末尾，我们学会了一种简化代码的方式：格式化字符串。

不过，还有一种更强大的方法，下面我们会先学习，然后再练习。

### 【讲解】

明确目标很重要（所以重复三遍）。

做到后面的步骤，可再点开左侧的“练习介绍”查看。

### 学习format()函数

format()函数是从 Python2.6 起新增的一种格式化字符串的函数，功能比课堂上提到的方式更强大。

format()函数用来占位的是大括号{}，不用区分类型码（%+类型码）。

具体的语法是：'str.format()'，而不是课堂上提到的'str % ()'。

而且，它对后面数据的引用更灵活，不限次数，也可指定对应关系。

看完左侧的代码、结果和注释，你就懂上面几句话的意思了。

### 运用format()函数

将代码中字符串格式化的代码改成format()函数的方法，改完后运行一下，检验是否正确。

### 【提示】

一个小提示：一共有5行代码需要改，找到它们，然后改下。

### 【解答】

```
1  import time
2  import random
3
4
5  player_victory = 0
6  enemy_victory = 0
7
8
9  while True:
10     for i in range(1,4):
11         time.sleep(1.5)
12         print(' \n—————现在是第 {} 局—————'.format(i))
13         player_life = random.randint(100,150)
14         player_attack = random.randint(30,50)
15         enemy_life = random.randint(100,150)
16         enemy_attack = random.randint(30,50)
17
18
```

```

19     print('【玩家】\n血量: {}\n攻击:
    {}'.format(player_life,player_attack))
20     print('-----')
21     time.sleep(1)
22     print('【敌人】\n血量: {}\n攻击:
    {}'.format(enemy_life,enemy_attack))
23     print('-----')
24     time.sleep(1)
25
26
27     while player_life > 0 and enemy_life > 0:
28         player_life = player_life - enemy_attack
29         enemy_life = enemy_life - player_attack
30         print('敌人发起了攻击, 【玩家】剩余血量{}'.format(player_life))
31         print('你发起了攻击, 【敌人】的血量剩余{}'.format(enemy_life))
32         print('-----')
33         time.sleep(1.2)
34
35
36         if player_life > 0 and enemy_life <= 0:
37             player_victory += 1
38             print('敌人死翘翘了, 你赢了! ')
39         elif player_life <= 0 and enemy_life > 0:
40             enemy_victory += 1
41             print('悲催, 敌人把你干掉了! ')
42         else:
43             print('哎呀, 你和敌人同归于尽了! ')
44
45
46     if player_victory > enemy_victory :
47         time.sleep(1)
48         print('\n【最终结果: 你赢了! 】')
49     elif enemy_victory > player_victory:
50         print('\n【最终结果: 你输了! 】')
51     else:
52         print('\n【最终结果: 平局! 】')
53
54
55     a1 = input('要继续游戏吗, 请输入n退出, 输入其他继续: ')
56     if a1 == 'n':
57         break
58
59

```

## 第8关

练习-老师我帮你-参考



## 1. 合并列表-1

分析问题，明确结果

我们的问题是：将两个列表合并起来，并按照从低到高的顺序排序，要得到问题的结果，我们还需要借助打印函数看看我们合并的是否正确。

### 思考要用到的知识&思考切入点

增加列表的内容与合并列表的含义相同，所以我们可以使用append作为解题的切入点，请你试试！

**提示：**假设要用已学过的知识来完成这道作业，需要用到列表的新增函数和循环。

```
1 list1 = [91, 95, 97, 99]
2 list2 = [92, 93, 96, 98]
3 for i in list2:
4     list1.append(i)
5 print(list1)
```

## 2.合并列表-2

上网搜索新知识&找到新的切入点

好了。你已经完成了第一个需求：合并列表。不过，有没有发现，这个代码还是比较冗余的。有没有更简单的方法呢？请你自己上网搜索一下

`python 合并两个列表`，看看是否有更简单的方法（**提示：**以字母e开头），学会后再回来继续做作业吧。

请你根据新学到的知识，简化代码。

注（如果你直接将 list2 合并到 list1 上，那就无法做到只看A组的成绩，所以，最好再建一个列表来合并两组的成绩）

```
1 list1 = [91, 95, 97, 99]
2 list2 = [92, 93, 96, 98]
3 # 把 A 组成绩赋值给一个新列表，用来存合并的成绩——这个细节要注意！
4 list3 = list1
5 list3.extend(list2)
6 print(list3)
```

## 3.列表排序

主动搜索，掌握新知

老师鼓励你自己去探寻解决问题的方法，上网搜索一下列表的排序方法吧，是不是发现了一个以字母s开头的方法？

```
1 list3= [91, 95, 97, 99, 92, 93, 96, 98]
2 list3.sort()
3 print(list3)
```

**练习-老师我又来了-参考**

## 练习目标

这个练习，是建立在上一个练习之上，用代码帮老师完成更多的成绩处理工作。

## 练习要求

上一个练习中，我们完成了两组成绩的合并和排序。

不过，老师有了新的需求：想知道两组的平均分，以及把低于平均分的成绩也打印出来。

所以，在这个练习中，我们会帮老师计算出两组的平均分，并挑出那些在平均分之下的成绩。

## 【讲解】

明确目标很重要（所以重复三遍）。

做到后面的步骤，可再点开左侧的“练习介绍”查看。

## 【提示】

解题三连击：

- 1.目前我们想要的结果是：求平均值和判断比较；
  - 2.我们可以用到的知识有：循环叠加成绩，然后除以总人数，即可求平均值。判断比较用小于号和空列表即可；
  - 3.我们的切入点：由于学生的成绩已经被我们集中到一个列表里，所以可以用for循环遍历这个列表来取出小于平均值的成绩。
- 经过三连击，相信你已经可以帮老师自动化地达成目标了。

## 【解答】

```
1 scores1 = [91, 95, 97, 99, 92, 93, 96, 98]
2 sum = 0
3 scores2 = []
4
5 for score in scores1:
6     sum = sum + score
7     average = sum/len(scores1)
8 print('平均成绩是：{}'.format(average))
9
10
11 for score in scores1:
12     if score < average:
13         scores2.append(score)
14         continue
15 print(' 低于平均成绩的有：{}'.format(scores2))
```

## 第二种 解题三连击：

- 1.目前我们想要的结果是：求平均值和判断比较；
- 2.我们可以去找的新知识有：Python 求平均值；
- 3.我们的切入点：请你通过搜索，找到更简单的求平均值的方法，来改造代码。

## 【提示】

这个方法的单词首字母是“n”，性质是“拓展程序库”。在改造代码前，可以先读懂网上的案例。

## 【解答】

```

1 import numpy as np
2
3
4 scores1 = [91, 95, 97, 99, 92, 93, 96, 98]
5 scores2 = []
6
7
8 average = np.mean(scores1) # 一行解决。
9 print('平均成绩是: {}'.format(average))
10
11
12 for score in scores1:
13     if score < average:
14         scores2.append(score)
15         continue
16 print(' 低于平均成绩的有: {}'.format(scores2))
17
18

```

## 第9关

### 练习-Hellokitty抽奖器-参考

练习目标：

我们会通过今天的项目练习，学习函数的封装和调用。

练习要求：

我们已经有一个hellokitty抽奖器，现在，请你把这个程序封装成一个新的函数。

### 运行抽奖器的代码参考

```

1 import random
2 import time
3
4 # 用random函数在列表中随机抽奖，列表中只有3位候选者。
5 luckylist = ['海绵宝宝','派大星','章鱼哥']
6 # random模块中有个随机选取一个元素的方法：random.choice()。
7 a = random.choice(luckylist) # 从3个人中随机选取1个人。
8 print('开奖倒计时',3)
9 time.sleep(1) # 调用time模块，控制打印内容出现的时间
10 print('开奖倒计时',2)
11 time.sleep(1)
12 print('开奖倒计时',1)
13 time.sleep(1)

```

```

14 # 使用三引号打印hellokitty的头像
15 image = '''
16 /\_)o<
17 |      \\\
18 | 0 . 0|
19 \_____/
20 '''
21 print(image) # .....
22 print('恭喜'+a+'中奖! ') # 使用print函数打印幸运者名单
23 import random
24 import time

```

运行代码，熟悉了解抽奖器抽奖的原理，请仔细查看注释讲解。

### 分析代码结构，完成函数封装

```

1 # 提示：将以下部分封装进函数
2 luckylist = ['海绵宝宝','派大星','章鱼哥']
3 a = random.choice(luckylist)
4 print('开奖倒计时',3)
5 time.sleep(1)
6 print('开奖倒计时',2)
7 time.sleep(1)
8 print('开奖倒计时',1)
9 time.sleep(1)
10 image = '''
11 /\_)o<
12 |      \\\
13 | 0 . 0|
14 \_____/
15 '''
16 print(image)
17 print('恭喜'+a+'中奖! ')
18 # 查看注释，运行代码。
19 import random
20 import time

```

### 抽奖程序分为两部分：

我们需要将第二部分用函数封装起来，并调用函数。提示：3个参与者即函数有3个参数。

```

1 import random
2 import time
3
4 # 将抽奖程序封装成函数
5 def choujiang(q,w,e): # 定义一个抽奖函数，带有3个参数，也就是3位候选人
6     luckylist = [q,w,e] # 定义一个中奖名单的列表
7     a = random.choice(luckylist) # 在中奖名单里面随机选择
8     print('开奖倒计时',3)
9     time.sleep(1)
10    print('开奖倒计时',2)
11    time.sleep(1)
12    print('开奖倒计时',1)

```

```
13     time.sleep(1)
14     image = '''
15     /\_)o<
16     |      \\  
17     | 0 . 0|
18     \_____/
19     '''
20     print(image)
21     print('恭喜'+a+'中奖! ')
22     choujiang('虚竹','萧峰','段誉') # 调用函数
```

## 练习-年终奖有多少

### 【解答】

```
1  def bonus(month):
2      if month < 6:
3          money = 500
4          return money
5      elif 6 <= month <= 12:
6          money = 120 * month
7          return money
8      else:
9          money = 180 * month
10         return money
11
12
13  def info(name, month):
14      gain = bonus(month)
15      print('%s来了%s个月, 获得奖金%s元' % (name, month, gain))
16
17
18  info('大聪',14)
```