

# 学习小宝典

---

## 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:
  ['love','hatred','envy']}
2 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 print('第' + str(n) + '对实验者选了最优解。')
21
22 for i in range(n):
23     print('第' + str(i+1) + '对实验者的选择是: ' + str(list_answer[i]))
```

## 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:
20         {}'.format(player_life,player_attack))
21         print('-----')
22         time.sleep(1)
23         print('【敌人】\n血量: {}\n攻击:
24         {}'.format(enemy_life,enemy_attack))
25         print('-----')
26         time.sleep(1)
27
28         while player_life > 0 and enemy_life > 0:
29             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-生成一副扑克牌-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，你可以通过Python优雅地生成一副扑克牌。

### 练习要求：

我们将通过这个练习，简单地复习一下return的用法。

另外，这个练习也要求你能够快速的学习新知识并将其运用出来。

新知识有3个：一种新的列表生成方式、extend 的新用法和列表生成式。

### 三个新知识

请仔细查看右侧的代码和注释。重要的几行代码：

```
1 num2 = list(range(1,6))
2
3 num2.extend('ABCDE')
4
5 list2 = [m+n for m in ['天字', '地字'] for n in '一二']
6
7 list3 = [n*n for n in range(1,11) if n % 3 == 0]
```

请根据学到的新知识，补全函数cards()。

每张扑克牌的展现形式是一个元组：(花色，大小)。

### 【解答】

```
1 def cards():
2     color = ['红心', '方块', '梅花', '黑桃']
3     num = list(range(2, 11))
4     num.extend('JQKA')
5     return [(x, y) for x in color for y in num ]
6
7
8 print(cards())
9
10
```

## 第9关（新版本）

### 练习目标

通过这个练习，掌握函数定义和调用的基本用法



**练习要求：**

眼看要过年了，深夜食堂经营的不错，你打算给员工发奖金犒劳一下。请你定义函数，当输入员工姓名和工作时长两个参数，即可打印出该员工获得多少年终奖。