

# 自动组卷系统(二)





## 1. 探索新知

### 1.1

### 打乱顺序

```
landmarks = {"中国": "长城", "古埃及": "金字塔", "法国": "埃菲尔铁塔",  
             "印度": "泰姬陵", "英国": "大本钟", "美国": "自由女神像",  
             "意大利": "比萨斜塔", "澳大利亚": "悉尼歌剧院",  
             "日本": "富士山", "希腊": "帕特农神庙", "南非": "好望角",  
             "丹麦": "小美人鱼铜像", "巴西": "基督像",  
             "泰国": "大皇宫", "加拿大": "CN塔",  
             "荷兰": "桑斯安斯风车村", "俄罗斯": "红场",  
             "捷克": "布拉格城堡", "沙特阿拉伯": "麦加大清真寺",  
             "西班牙": "圣家族大教堂"}  
}
```

字典中有20个国家及其地标性建筑，咱们要选择10个国家进行组卷，这里采用的是打乱顺序后选择前10个国家的方式来实现



首先，我们如何提取出字典中20个国家，放在一个列表中呢？

字典中的键是国家的名称，字典中的值是对应的地标建筑，想要获取所有的国家名称，我们可以使用`landmarks.keys()`来获取

```
states = list(landmarks.keys())
```

【state】：  
国家

## 1.1

### 打乱顺序

拓展：

`list()`函数是Python内置的一个函数，用于将其他可迭代对象（iterable）转换为列表（list）



此时states列表中存储着20个国家，如何实现随机打乱的效果呢？

【shuffle】：  
洗牌，变换位置

`random.shuffle()`

`random`模块中的`shuffle()`函数可以实现随机打乱的效果，可以随机地打乱一个可变序列（如列表）中元素的顺序。

举个例子帮助大家理解`shuffle()`函数：

```
import random
my_list = [1,2,3,4,5]
random.shuffle(my_list)
print(my_list)
```

输出（不唯一）：

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

打乱国家顺序的代码如下：

```
random.shuffle(states)
```

别忘了导入  
`random`库呀！

## 1.1

### 打乱顺序

任务一新增代码如下：

```
# 导入随机库
import random
# <已完成：匹配的国家地标>
# 生成10个测验文档
for i in range(10):
    # <已完成：新建带编号的测试卷和答案文件>
    # <已完成：写测试卷的抬头>
    # 打乱顺序
    states = list(landmarks.keys())
    random.shuffle(states)
```

## 1.2

### 生成10道题目和答案

```
# 生成10道题目和答案
for j in range(10):
    # <待做：1. 设置问题并写入 >
    # <待做：2. 获取选项 >
    # <待做：3. 打乱选项 >
    # <待做：4. 将选项写入测验文件>
    # <待做：5. 将答案写入答案文件>
```



我们先来完成【问题的设置及写入】

## 1.2

## 生成10道题目和答案

### 1. 设置问题并写入

1. 中国 的地标建筑是?

- A. 金字塔
- B. 悉尼歌剧院
- C. 长城
- D. 大本钟

2. 俄罗斯 的地标建筑是?

- A. 比萨斜塔
- B. 红场
- C. 好望角
- D. 基督像

拆解题目，找到固定的格式：

题目编号. 国家 的地标建筑是?



$j=0$ 时，题目编号为1;

$j=1$ 时，题目编号为2;

.....

所以， $questionNum = j + 1$

`print(states)`，我们可以看到列表中`states`列表中存储着20个随机打乱顺序的国家

控制台

清空 缩小

```
['中国', '俄罗斯', '日本', '西班牙', '意大利', '加拿大', '南非', '泰国', '古埃及', '法国', '英国', '印度', '澳大利亚', '美国', '巴西', '希腊', '丹麦', '捷克', '沙特阿拉伯', '荷兰']
```

在【`for j in range(10):`】这个循环中，我们将按顺序选择前10个国家出题。

从`states[0]`到`states[9]`，所以选择的国家 `state = states[j]`

已知：

问题格式：

题目编号. 国家 的地标建筑是?

题目编号、国家都有了，接下来选择自己喜欢的方法，将题目进行格式化吧！

```
# 题目编号
questionNum = j + 1
# 国家
state = states[j]
```



这里给出两种示例：

方法一：%占位符

```
'%s. %s 的地标建筑是?\n' % (questionNum, state)
```

方法二：f-string

```
f'{questionNum}. {state} 的地标建筑是?\n'
```



接着把格式化后的题目写入测试卷文件，轻轻松松~

```
testFile.write(f'{questionNum}. {state} 的地标建筑是?\n')
```

【1. 设置问题并写入】新增代码如下：

```
# 生成10道题目和答案
for j in range(10):
    # 设置问题并写入
    # 题目编号
    questionNum = j + 1
    # 国家
    state = states[j]
    # 写入文件
    testFile.write(f'{questionNum}. {state} 的地标建筑是?\n')
```

## 1.2

## 生成10道题目和答案

### 2. 获取选项

1. 中国的地标建筑是?

- A. 金字塔
- B. 悉尼歌剧院
- C. 长城
- D. 大本钟

观察左侧的题目可以发现，每道题目有4个选项，其中有1个正确选项，3个错误选项



正确的答案很容易得到，它作为一个值保存在 landmarks 字典中。



只要知道正确答案的键，就可以得到正确答案，那正确答案的键又是什么呢？

国家: `state`

地标: `landmarks[ state ]`

```
correctAnswer = landmarks[state]
```

【correctAnswer】：  
正确答案

获取错误地标列表需要一点小技巧。你有没有好的方法呢？

我们可以：

① 从landmarks字典中获取所有的值

```
wrongAnswers = list(landmarks.values())
```

【wrongAnswer】：  
错误答案

## 1.2

### 生成10道题目和答案

② 再删掉正确的值，剩下的值组成的就是【错误地标列表】

```
wrongAnswers.remove(correctAnswer)
```

新增代码如下：

```
correctAnswer = landmarks[state] # 正确选项  
wrongAnswers = list(landmarks.values())  
wrongAnswers.remove(correctAnswer) # 错误地标总列表
```

③ 接着我们从【错误地标列表】中随机选择3个值作为题目的三个错误选项即可



随机生成一个整数我倒是学过。  
从列表中随机选择3个值，这该怎么实现???  
大聪明们，你们有办法吗？



这个没有学过，咱们笔记上没有。  
我不会(; '⌣`)

我觉得可以使用shuffle()函数，先把  
wrongAnswers列表打乱，再用循环取出前三  
个元素就可以了！跟前面随机取10个国家是一  
样的。



的确用`shuffle()`函数可以实现效果。这里再教大家一种更简单的方法——`sample()`函数

【sample】：  
取样，抽样

`random.sample(可迭代序列, 元素个数)`

Python中`random`模块提供的一个函数。用于从指定的序列中随机选择多个唯一的元素。函数返回一个列表，包含这些随机选择的元素。

举个例子帮助大家理解`sample()`函数：

```
import random
a = [1, 2, 3, 4, 5]
b = random.sample(a, 3)
print(b)
```

输出（不唯一）：

```
[3, 1, 5]
```

随机选出三个错误地标的代码如下：

```
wrongAnswers = random.sample(wrongAnswers, 3)
```

1个正确地标和3个错误地标都已经准备好了，把这四个地标放到一个【答案选项】列表中。

【answerOption】：  
答案选项

```
# 获取answerOptions列表
answerOptions = wrongAnswers + [correctAnswer]
```

【2. 获取选项】新增代码如下：

```
# 获取选项
# 获取正确地标
correctAnswer = landmarks[state]
# 获取错误地标列表
wrongAnswers = list(landmarks.values())
wrongAnswers.remove(correctAnswer)
wrongAnswers = random.sample(wrongAnswers, 3)
# 获取【答案选项】列表
answerOptions = wrongAnswers + [correctAnswer]
```

### 3. 打乱顺序

金字塔	}	wrongAnswers
悉尼歌剧院		
大本钟		
长城	}	correctAnswer

按照上面的方法，得到的【answerOptions】，正确选项总是最后一个，为了防止部分学生投机取巧，我们需要把4个选项打乱顺序，这样才能考查大家的真实水平。



随机打乱顺序的方法咱们上个任务已经学过了，自己试着打乱【answerOptions】中选项的顺序吧！

```
# 打乱答案选项
random.shuffle(answerOptions)
```

## 1.2

### 生成10道题目和答案

【3. 打乱顺序】新增代码如下：

```
# 打乱【答案选项】  
random.shuffle(answerOptions)
```

#### 4. 将选项写入测验文件

1. 法国 的地标建筑是？

- A. 布拉格城堡
- B. 埃菲尔铁塔
- C. 小美人鱼铜像
- D. CN塔

2. 巴西 的地标建筑是？

- A. 长城
- B. 大本钟
- C. 基督像
- D. CN塔

拆解选项，找到固定的格式：

■ ‘ABCD’ 中的一个字符。 ■ 答案选项之一

那么此时问题又拆解为，  
如何按顺序得到A、B、C、  
D四个字符？



试试看，下面这个方法，能按顺序得到A、B、C、D四个字符吗？

```
for i in range(4):  
    print('ABCD'[i])
```

输出：

A  
B  
C  
D

接着试着通过循环4次，把格式化后的4个选项写入测试卷文件吧！

```
for k in range(4):  
    testFile.write(f' {"ABCD"[k]}. {answerOptions[k]}\n')
```

## 1.2

### 生成10道题目和答案

选项写入完成后，光标在下一行的开头，准备写入，我们可以再换一行，让两道题目之间空一行，试卷更加美观

```
testFile.write('\n')
```

1. 法国 的地标建筑是?

- A. 布拉格城堡
- B. 埃菲尔铁塔
- C. 小美人鱼铜像
- D. CN塔

2. 巴西 的地标建筑是?

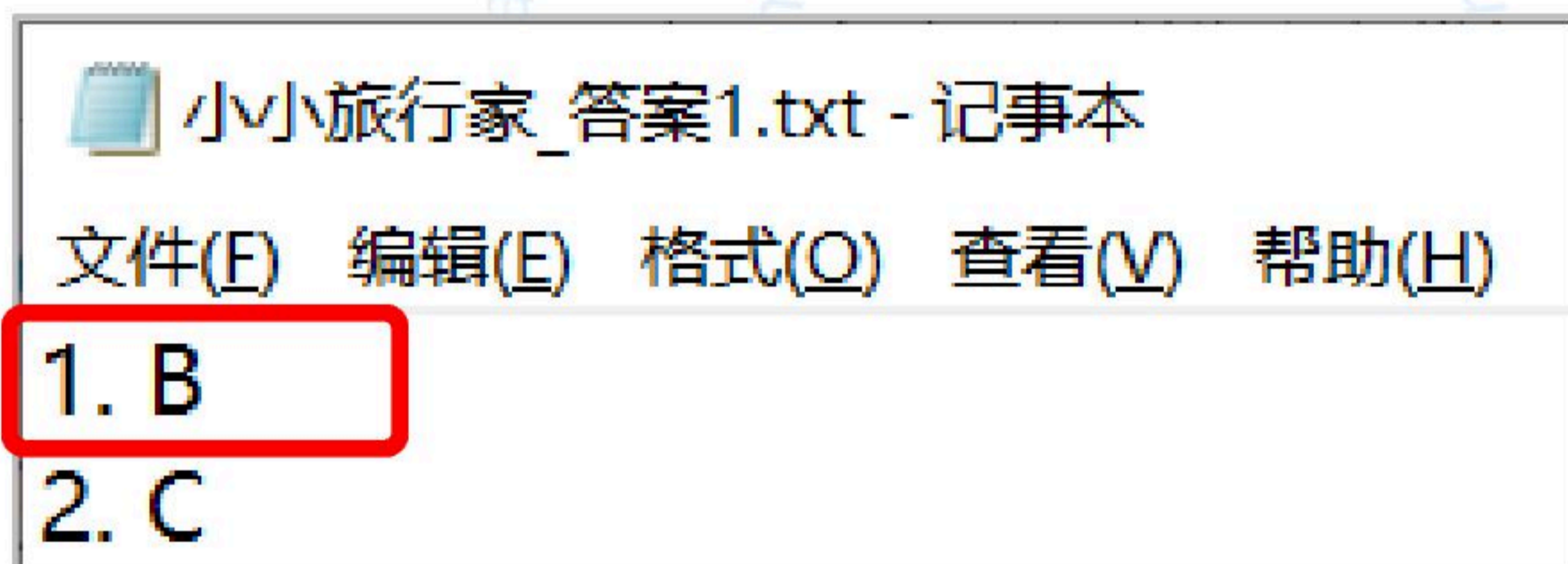
- A. 长城
- B. 大本钟
- C. 基督像
- D. CN塔

【4. 将选项写入测验文件】新增代码如下：

```
# 将选项写入测验文件
for k in range(4):
    testFile.write(f'{"ABCD"[k]}. {answerOptions[k]}\n')
testFile.write('\n')
```

#### 5. 将答案写入答案文件

啊！差点忘记在答案文件中，写入每道题目的正确答案。



拆解选项，找到固定的格式：

题目编号. 正确答案对应的字母编号

questionNum

【index】：  
索引

```
'ABCD'[answerOptions.index[correctAnswer]]
```

拓展：列表.index(元素) 用于返回指定元素在列表中第一次出现的索引位置

【5. 将答案写入答案文件】新增代码如下：

```
# 将答案写入答案文件
```

```
correctOption = 'ABCD'[answerOptions.index(correctAnswer)]  
answerFile.write(f'{questionNum}. {correctOption}\n')
```

任务二新增全部代码如下：

```
# 生成10道题目和答案
```

```
for j in range(10):
```

```
    # 1. 设置问题并写入
```

```
    questionNum = j + 1
```

```
    state = states[j]
```

```
    testFile.write(f'{questionNum}. {state} 的地标建筑是?\n')
```

```
    # 2. 获取选项
```

```
    correctAnswer = landmarks[state]
```

```
    wrongAnswers = list(landmarks.values())
```

```
    wrongAnswers.remove(correctAnswer)
```

```
    wrongAnswers = random.sample(wrongAnswers, 3)
```

```
    answerOptions = wrongAnswers + [correctAnswer]
```

```
    # 3. 打乱选项
```

```
    random.shuffle(answerOptions)
```

```
    # 4. 将选项写入测验文件
```

```
    for k in range(4):
```

```
        testFile.write(f'{"ABCD"[k]}. {answerOptions[k]}\n')
```

```
    testFile.write('\n')
```

```
    # 5. 将答案写入答案文件
```

```
    correctOption = 'ABCD'[answerOptions.index(correctAnswer)]
```

```
    answerFile.write(f'{questionNum}. {correctOption}\n')
```



文件操作完成后，不要忘记关闭文件。自己试着去完成吧！

注意：每套试卷编写完后都得关闭文件，代码要写在哪层循环中呢？

任务三新增全部代码如下：

```
# 生成10套测验
for i in range(10):
    # <新建带编号的测试卷和答案文件>
    # <写测试卷的抬头>
    # <打乱顺序>
    # <生成10道题目和答案>
    # 关闭测验文件、答案文件
    testFile.close()
    answerFile.close()
```

## 完整代码

```
import random
# 匹配的国家地标
landmarks = {"中国": "长城", "古埃及": "金字塔", "法国": "埃菲尔铁塔",
             "印度": "泰姬陵", "英国": "大本钟", "美国": "自由女神像",
             "意大利": "比萨斜塔", "澳大利亚": "悉尼歌剧院",
             "希腊": "帕特农神庙", "日本": "富士山", "南非": "好望角",
             "丹麦": "小美人鱼铜像", "巴西": "基督像", "泰国": "大皇宫",
             "加拿大": "CN塔", "荷兰": "桑斯安斯风车村", "俄罗斯": "红场",
             "捷克": "布拉格城堡", "沙特阿拉伯": "麦加大清真寺",
             "西班牙": "圣家族大教堂"}
# 生成10套测验
for i in range(10):
    # 新建带编号的测试卷和答案文件
    testNum = i + 1
    testFile = open('小小旅行家%s.txt'%(testNum), 'w')
    answerFile = open('小小旅行家_答案%s.txt'%(testNum), 'w')
    # 写测试卷的抬头
    testFile.write('姓名:\n\n班级:\n\n日期:\n\n')
    testFile.write((' '*20)+'小小旅行家地标挑战赛 (Form %s)'% (testNum))
    testFile.write('\n\n')
    # 打乱顺序
    states = list(landmarks.keys())
    random.shuffle(states)
    # 生成10道题目和答案
    for j in range(10):
        # 设置问题并写入
        questionNum = j + 1
        state = states[j]
        testFile.write(f'{questionNum}. {state} 的地标建筑是?\n')
```

## 完整代码

```
# 获取选项
correctAnswer = landmarks[state]
wrongAnswers = list(landmarks.values())
wrongAnswers.remove(correctAnswer)
wrongAnswers = random.sample(wrongAnswers, 3)
answerOptions = wrongAnswers + [correctAnswer]
# 打乱【答案选项】
random.shuffle(answerOptions)
# 将选项写入测验文件
for k in range(4):
    testFile.write(f'{"ABCD"[k]}. {answerOptions[k]}\n')
testFile.write('\n')
# 将答案写入答案文件
correctOption = 'ABCD'[answerOptions.index(correctAnswer)]
answerFile.write(f'{questionNum}. {correctOption}\n')
# 关闭测验文件、答案文件
testFile.close()
answerFile.close()
```



## 2. 强化练习

1. `wrongAnswers = list(landmarks.values())` 这行代码的作用是什么？  
( )

- A. 获取所有国家的名称
- B. 获取所有地标建筑的名称，并存储在 `wrongAnswers` 列表中
- C. 删除正确的答案选项
- D. 打乱地标建筑的顺序

2. 在生成每道题目的答案选项时，代码中使用了哪个函数来确保答案选项的顺序是随机的？ ( )

- A. `random.sample()`
- B. `random.shuffle()`
- C. `list.sort()`
- D. `list.reverse()`

3. 在生成每个问题的答案选项时，为什么要对 `answerOptions` 列表进行清洗？ ( )

- A. 为了确保正确答案总是在最后一个
- B. 为了确保正确答案总是在第一个
- C. 为了确保正确答案随机出现在四个选项中的任意一个
- D. 以上说法都不对



## 2. 强化练习

4. `correctOption = 'ABCD'[answerOptions.index(correctAnswer)]`  
这行代码是如何确定正确答案的字母标识的？（ ）

- A. 通过查找 `correctAnswer` 在 `answerOptions` 中的位置，并用该位置对应的字母表示
- B. 通过比较 `correctAnswer` 和所有答案选项的文本内容
- C. 随机选择一个字母作为正确答案的标识
- D. 这样代码不能确定正确答案的字母标识

5. 下面哪个函数，可以实现从列表中随机选择三个元素的效果？（ ）

- A. `random.shuffle(my_list)`
- B. `random.randint(0,3)`
- C. `random.sample(my_list, 3)`
- D. `random.choice(my_list)`

### 3. 术语箱

shuffle 洗牌，变换位置

sample 取样，抽样

index 索引

state 书写

correct 正确的

wrong 错误的

option 选项

### 4. 课后挑战

#### 来拆盲盒吧

超市最近新推出了商品盲盒的活动，顾客支付30元可以获得一次开盲盒的机会。顾客们都跃跃欲试，想看看自己的手气。

盲盒中会随机出现三种商品，你可能会买到价值远高于30的商品，你也可能得到刚好30或者低于30的商品。快来试试自己的运气吧！

#### 提示如下：

1. 已知参与盲盒活动的商品及价格
2. 从中随机抽取三种商品并输出
3. 计算抽出的三种商品总价值
4. 输出“你的盲盒价值××元”

