

实验题目：实验 1 抓狐狸小游戏，查找 Word 中红色文本和加粗文本

实验环境：Python3.7、PyCharm 等

---

## 一、实验目的

### (1) 抓狐狸小游戏

1. 培养分析问题并对问题进行建模的能力。
2. 熟练使用列表解决实际问题。
3. 熟练运用选择结构和循环结构解决实际问题。
5. 理解带 else 子句的循环结构的执行流程。
5. 理解使用异常处理结构约束用户输入的用法。

### (2) 查找 Word 中的红色文本和加粗文本

1. 了解扩展库 python-docx 的安装与使用。
2. 理解 Word 文档结构和内容组织形式。
3. 理解 Word 文档中 run 的概念。
4. 熟练运用列表、字典、集合等结构解决实际问题

## 二、实验内容

### 1. 抓狐狸小游戏

编写程序，模拟抓狐狸小游戏。假设一共有一排 5 个洞口，狐狸最开始的时候在其中一个洞口，然后玩家随机打开一个洞口，如果里面有狐狸就抓到了，如果洞口里没有狐狸就第二天再来抓，但是第二天狐狸会在玩家来抓之前跳到隔壁洞口里。如果在规定的次数内抓到了狐狸就提前结束游戏并提示成功；如果规定的次数用完还没有抓到狐狸，就结束游戏并提示失败。

相关知识：

(1) random.randrange([start], stop[, step]) 从指定范围内，按指定基数递增的集合中获取一个随机数。

(2) random.choice(sequence) 从序列中获取一个随机元素，参数 sequence 表示一个有序类型。sequence 在 python 不是一种特定的类型，而是泛指一系列的类型。list、tuple、str 都属于 sequence。

(3) 使用 try...except...语句捕获处理输入异常。

### 2. 查找 word 中的红色文本和加粗文本

编写程序，读取 Word 文件中的所有段落文本，然后输出其中所有红色的文本和加粗的文本以及同时具有这两种属性的文本。具体步骤如下：

(1) 在命令提示符环境使用 pip install python-docx 命令安装扩展库 python-docx。

(2) 创建测试用的 Word 文档 test.docx，写入测试内容，并根据需要设置红色文本和加粗文本。

(3) 编写程序查找并输出 Word 文档 test.docx 中的红色文本和加粗文本。

## 三、实验步骤及结果

## 1. 抓狐狸小游戏

```

import random
from random import randrange

"""
模拟抓小狐狸，一共有n个洞口，MaxTime次没有抓到算作失败
每次抓捕失败小狐狸会跳到隔壁洞口
"""

def catchFox():
    n = 5;          # 有n个洞口
    MaxTimes = 10;   # 最多抓MaxTime次
    position = [0] * (n);
    # 随机生成小狐狸初始所在洞口编号
    OldPos = randrange(0, n)
    position[OldPos] = 1

    # 进行抓捕
    flag = False
    for i in range(0, MaxTimes):
        # print(OldPos)
        # 输入判断，输入格式错误不耗费抓捕次数
        while True:
            try :
                pos = input("请输入洞口编号：")
                pos = int(pos)
                assert 0<=pos<n
                # 当输入为数字且范围合理时为合法输入
                break
            except:
                print("没有这个洞口，再试一次吧！")

        if(position[pos] == 1) :
            print("抓到了！")
            flag = True;
            break;
        else:
            print("没有抓到哦！")

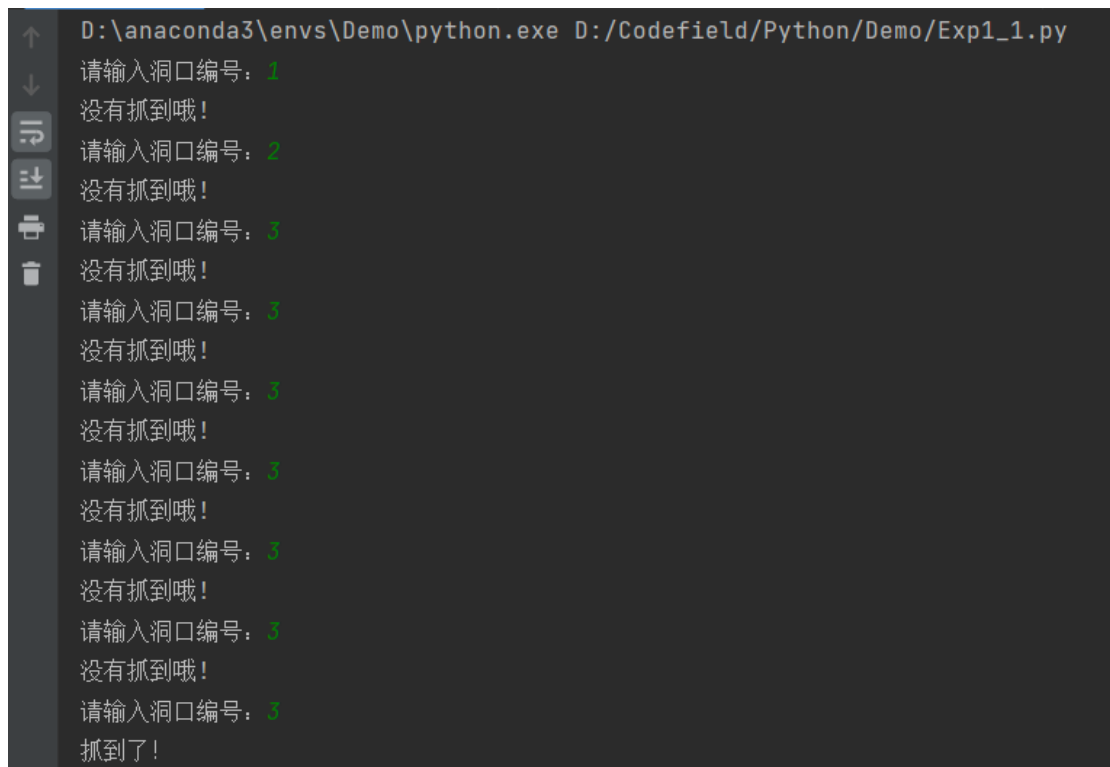
```

```
NewPos = (OldPos + random.choice([-1, 1])) % n
position[NewPos], position[OldPos] = 1, 0
OldPos = NewPos

if flag == False:
    print("次数用完了喔，你没有机会啦！")

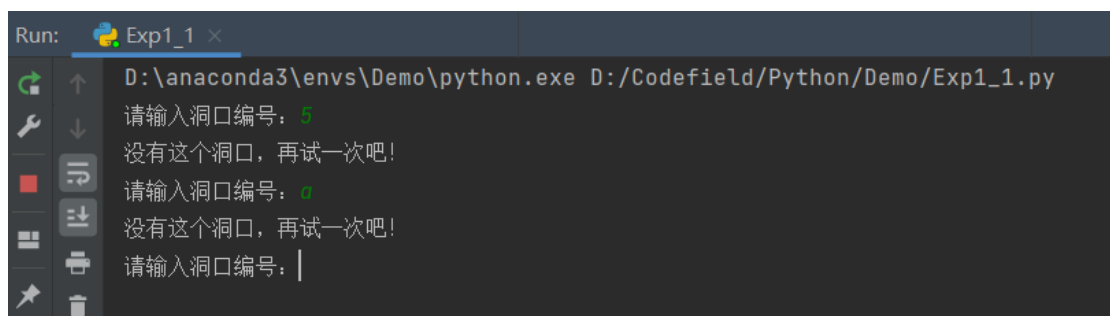
catchFox()
```

运行结果如图 1，2，3：



A terminal window showing the execution of a Python script. The command prompt is 'D:\anaconda3\envs\Demo\python.exe D:/Codefield/Python/Demo/Exp1\_1.py'. The program prompts the user to enter a cave number ('请输入洞口编号:'). The user enters '1', '2', '3', '3', '3', '3', '3', '3', '3', and '3'. For the first nine inputs, the program outputs '没有抓到哦!' (Didn't catch it!). On the tenth input, the program outputs '抓到了!' (Caught it!).

图 1 在次数用完之前抓到狐狸



A terminal window showing the execution of the same Python script. The command prompt is 'D:\anaconda3\envs\Demo\python.exe D:/Codefield/Python/Demo/Exp1\_1.py'. The program prompts the user to enter a cave number ('请输入洞口编号:'). The user enters '5' and '0'. For both inputs, the program outputs '没有这个洞口，再试一次吧!' (No such cave, try again!). The prompt is shown again at the end of the screenshot.

图 2 输入不合法

```
Run: Exp1_1 x
请输入洞口编号: 2
没有抓到哦!
请输入洞口编号: 1
没有抓到哦!
请输入洞口编号: 4
没有抓到哦!
请输入洞口编号: 0
没有抓到哦!
请输入洞口编号: 3
没有抓到哦!
请输入洞口编号: 1
没有抓到哦!
请输入洞口编号: 4
没有抓到哦!
请输入洞口编号: 3
没有抓到哦!
请输入洞口编号: 2
没有抓到哦!
次数用完了喔, 你没有机会啦!

Process finished with exit code 0
```

图 3 全部次数用尽之后还没有抓到

## 2. 查找 Word 中的红色文本和加粗文本

```
from docx import Document
from docx.shared import RGBColor

Bold = []
Red = []
Text = Document('test.docx')
for p in Text.paragraphs:
    for r in p.runs:
        if r.bold:
            Bold.append(r.text)
        if r.font.color.rgb == RGBColor(255, 0, 0):
            Red.append(r.text)

print(f"{10*'}'}Red Text{10*'}'}")
for word in Red:
    print(word)

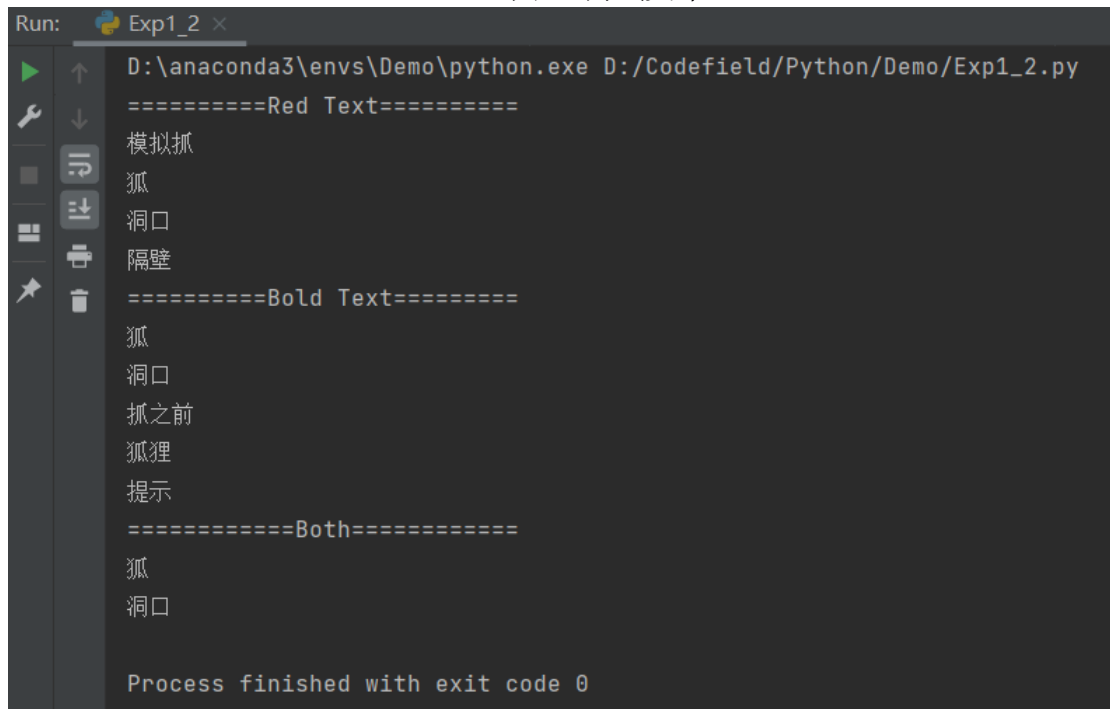
print(f"{10*'}'}Bold Text{9*'}'}")
for word in Bold:
    print(word)

Both = set(Red) & set(Bold)
print(f"{12*'}'}Both{12*'}'}")
for word in Both:
    print(word)
```

运行结果：

编写程序，模拟抓狐狸小游戏。假设一共有一排 5 个洞口，狐狸最开始的时候在其中一个洞口，然后玩家随机打开一个洞口，如果里面有狐狸就抓到了，如果洞口里没有狐狸就第二天再来抓，但是第二天狐狸会在玩家来抓之前跳到隔壁洞口里。如果在规定的次数内抓到了狐狸就提前结束游戏并提示成功；如果规定的次数用完还没有抓到狐狸，就结束游戏并提示失败。↵

图 1 测试文本



```
Run: Exp1_2 x
D:\anaconda3\envs\Demo\python.exe D:/Codefield/Python/Demo/Exp1_2.py
=====Red Text=====
模拟抓
狐
洞口
隔壁
=====Bold Text=====
狐
洞口
抓之前
狐狸
提示
=====Both=====
狐
洞口
Process finished with exit code 0
```

图 2 运行结果

## 四、实验分析

实验收获：

实现实验中要求的功能时，需要使用一些目前理论课还没有涉及到的知识，根据实验任务书中的提示对需要使用到的方法、包等上网查询，学到了新的用法。

1. random.choice([-1, 1])

从序列 sequence 中随机选取一个元素，sequence 泛指列表 list、元组 tuple、字典 dict、字符串 str 等数据序列。

2. assert 0<=pos<n

断言，当表达式正确的时候直接通过，如果表达式为 False 会引发 AssertionError 错误。常用于调试检测代码问题。

3. Python 对 Word 文档的操作需要安装模块 python-docx，只能读取 docx 文件不能读取 doc 文件。对 Word 文档进行编辑之前需要导入 Document 模块。

4. docx 文件的结构可以分为三层。Document 对象表示整个文档；其中包含 Paragraph 对象的列表，表示文档中的一个段落；Paragraph 中包含 Run 对象的列表，表示样式相同的一段文本。

5. set (Red) & set(Bold)

将 Reg 列表和 Bold 列表转化为集合，做并集运算。