



上海大学

SHANGHAI UNIVERSITY

Python 计算实验报告

组 号 第 2 组

实 验 序 号 实验一

学 号 21120860

姓 名 刘虚谷

日 期 2023 年 4 月 10 日

Python 计算实验报告撰写提纲

一、实习目的与要求

1. 熟悉 Python 的开发调试环境；
2. 熟悉 Python 外部库的调用；
3. 掌握 Python 语言基本语法；
4. 熟悉 Python 的数据结构。

二、实习环境

1. 操作系统不限；
2. Python IDLE、PyCharm 等开发环境不限。

三、实习内容

1. Python 代码理解 `polygon.py`：（1）运行和阅读代码；（2）理解代码功能；（3）修改代码，练习调用文件中其他几个图形函数。
2. 输入输出：编写脚本文件，设计友好的用户输入输出提示，用户输入一个时间（24 小时制，包含时、分、秒），输出 1 秒后的时间。
3. 反序对：如果一个单词是另一个单词的反向序列，则称这两个单词为“反向对”。编写代码输出 `word.txt` 中词汇表包含的反向对。
4. 文本分析算法设计：
 - （1）设计 Python 程序读入一个英文单词组成的文本文件，统计该文本文件中各个单词出现的次数。设计测试用例验证代码的正确性。
 - （2）设计 Python 程序读入一个英文单词组成的文本文件，统计其中包含的某给定关键词列表中各个单词出现的频率。设计测试用例验证代码的正确性。

四、实习内容的设计与实现

1. `polygon.py`

```
def polyline(t, n, length, angle):
```

```
    """Draws n line segments.
```

```
    t: Turtle object
```

```
    n: number of line segments
```

length: length of each segment
angle: degrees between segments

"""

for i in range(n):

 fd(t, length)

 lt(t, angle)

该函数具有绘制多条线段的功能，其利用了 Turtleworld 模块创造了 Turtle 对象，并利用了循环结构控制 Turtle 对象绘制想要的线段。

2.time.py

x=input('请输入时间(时/分/秒): ')

a,b,c=map(int,x.split('/'))

c+=1

y=c/60

c%=60

b+=y

z=b/60

b%=60

a+=z

a%=24

a=str(int(a))+ '时'

b=str(int(b))+ '分'

c=str(int(c))+ '秒'

```
print(a,b,c,sep='/')
```

利用了 map 内置函数简化了输入功能的代码，体现了 python 的简便性，之后运用了除模运算算出了+1 后的时间。

3.反序对

```
with open('words.txt','r') as f:
```

```
    wordlist = set([word.strip() for word in f.readlines()])
```

```
    #strip()去除字符串前后空格和换行符
```

```
# print(wordlist)
```

```
reverse = set()
```

```
#无序集合，不重复
```

```
for word in wordlist:
```

```
    if word[::-1] in wordlist and word != word[::-1]: #[::-1]返回逆序字符串
```

```
        # x[start:stop:step] -1--> stop to start
```

```
        reverse.add((word,word[::-1]))
```

```
for i in reverse:
```

```
    print(i[0], i[1])
```

wordlist = set([word.strip() for word in f.readlines()])利用了列表推导式去掉字符串前后的空格和和换行， reverse.add((word,word[::-1]))利用了集合 add 函数，这样可以避免集合出现重复元素。

4.文本分析算法设计

4.1.py:

```
import collections
```

```

import re

with open('words.txt','r') as words:

    word=words.read()

    word=re.split('\n',word)

    word=collections.Counter(word)

    with open('output.txt','w') as output:

        output.writelines(str(word))

```

利用了 re 模块的 split 对象,将存储 word.txt 内容的 word 对象中的字符串以换行为标志符分割,然后利用了 collections 的 Counter 对象直接统计了 word 对象中的各个不同字符串出现的次数。

4.2.py:

```

import re

import collections

word=input('请输入要查找的单词:')

word=list(map(str,word.split(' ')))

with open('words.txt','r') as words:

    lines=words.read()

    lines=re.split('\n',lines)

    lines=collections.Counter(lines)

    for i in range(len(word)):

        print(str(word[i]),lines[str(word[i])],sep=':',end='\n')

```

前面部分的代码与 1.4.1.py 基本一样,而在最后我们利用了一个 for 类型的结构,

以及 collections 模块中 Counter 对象返回字典的性质来得到我们指定单词的频率。

五、测试用例

Words.txt

六、收获与体会

在本次实验后，以前没学过 python 的我大致了解了 python 语言的简便性，可读性，以及其他与 c/c++ 语言的不同，同时我变得更加在意代码的编写规范，并且在我用 python 绘制出我自己的图形时，我也认知到 Python 能做非常多有趣的东西，我也有了很大的动力学习下去。