

实验二 PYTHON 数据结构与流程控制

一、目的和要求

1. 熟悉 Python 的流程控制;
2. 熟悉 Python 的数据结构;
3. 掌握 Python 语言基本语法;

二、实验环境

1. Win 7 操作系统;
2. Python 2.7.X, IDLE、PyCharm 等开发环境;

三、实验内容

(一) 验证实验 (每个同学完成)

1. Python 流程控制

(1) 运行调试第三章课后练习的程序设计题, 检查运行结果是否正确, 记录实验结果。

(2) 生成含有 20 个 1 到 100 之间随机数的、所有元素不相同的列表。

(3) 安装 Swampy 扩展库, 阅读和运行 koch.py, 理解 Koch 曲线绘制的原理。

2. Python 序列对象

(1) 阅读和运行 birthday.py, 理解代码的组织结构, 理解列表的使用方法。

(2) 用嵌套循环编写 Python 源码生成 100 以内的所有素数, 并且将这段源码修改为一条列表推导式;

(3) 运行下面代码, 自学 Counter 对象的使用

```
from collections import Counter
```

```
cnt = Counter()
```

```
for word in ['red', 'blue', 'red', 'green', 'blue', 'blue']:
```

```
    cnt[word] += 1
```

```
cnt
```

（二）设计实验（小组完成，代码提交，算法设计和测试写入实验报告）

3. Python 流程控制（二选一）

（1）编写循环控制代码用下面公式逼近圆周率(精确到小数点后 15 位)，并且和 `math.pi` 的值做比较

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{k!^4 (396^{4k})}.$$

（2）阅读 https://en.wikipedia.org/wiki/Koch_snowflake，通过修改 `koch.py` 绘制其中一种泛化的 Koch 曲线。

4. 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。

（三）综合实验（小组完成，代码提交，算法设计和测试写入实验报告）

5. 文本分析算法设计（二选一）

（1）参照验证实验 2.(3)中的例示，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。