

# 实验一 PYTHON 基础与数据结构

## 一、目的和要求

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、实验环境

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

## 三、实验内容

### (一) 验证实验 (每个同学完成, 不写入报告)

#### Part 1: 熟悉 Python 的开发环境

1. 参照课本 1.1 节: 使用 IDLE 和 Python (Command Line) 两种界面, 用 P4 代码查看实验用机安装的 Python 版本。

2. 参照课本 1.2 节: (1) 安装 Anaconda/Miniconda 和 Python3.x; (2) 在 IDLE 下使用命令行和创建脚本两种方式创建 Hello World。

3. 装扩展库 Swampy。

4. 熟悉 PyCharm:

(1) 用桌面图标启动 PyCharm, 浏览 PyCharm 工作环境, 尝试修改 PyCharm 界面风格等, 可参考 <http://www.jetbrains.com/pycharm/documentation/>。

(2) 用 PyCharm 创建一个新工程, 在工程中添加源文件, 填写 Hellow World 代码, 调试和运行。

5. 熟悉 Python 调试工具:

(1) 参照课本 8.6 和 8.7 节, 尝试代码调试功能。

(2) 参照课本 1.5 节最末段 (P23 页), 尝试学习 Python 的代码风格工具。

## Part 2:熟悉 Python 基础编程

### 1. 参照课本

- (1) 验证 Python 动态类型语言的特性;
- (2) 用 `dir()` 查看 Python 关键字;

### 2. 把 Python 作为计算器:

- (1) 求下式计算结果, 并且计算该结果各位数字之和:

$$\underbrace{1515 \dots 15}_{10 \uparrow 15} \times \underbrace{333 \dots 33}_{20 \uparrow 3}$$

- (2) 判断下式计算结果十位数字的值

$$\underbrace{2016 \times 2016 \times 2016 \times \dots \times 2016}_{2016 \uparrow 2016}$$

### 3. 输入以下表达式并且查看结果

```
23+3、 23>3、 '23'+ '3'、 23/3、 23//3、 23%3、 23* *3  
23+24.5、 23+ '3'、 23+ int('3')、 'hello ' + str("123")、 int(23/3)、 round(23/3,2)、 round(23/3)、  
0<23<100
```

- 4. 文本对象: 声明字符串 `s1` 和 `s2`, 分别初始化为 `'programming'` 和 `'language'`, 观察以下表达式的计算结果

```
s1[1],s1[:4],s1[0]+s2[1:3],s1.capitalize()+ ' '+s2.upper(),s1.count('r')+s1.find('r')+  
s1.rfind('r'),s3=s2.join('-- '),s4='- '.join(s2),L1=s4.split(),3*(s2[:2]+' '), "Python"  
+s2.rjust(10)).
```

### 5. 内置函数的使用

- (1) 用 `dir()` 查看 Python 内置对象, 用 `help()` 分别查看任意 5 个关键字和 5 个内置对象的使用帮助, 并且尝试使用;

- (2) 比较函数 `ord()` 和 `str()` 的差异;

### 6. 模块导入

- (1) 导入 `math` 库: 查看所有函数、比较对数运算函数、平方根计算、幂运算、比较整函数 `ceil()` 和 `floor()` 等。

- (2) 导入 `random` 库: 生成 10 个 `[1, 100]` 间的随机整数, 计算最大值、最小值、和、平均 (参照课本 P17)。

### Part 3: Python 应用编程

1. 编写实现课本例程。
2. 附件 mypolygon.py: (1) 运行和阅读代码; (2) 理解代码功能。

#### (二) 设计实验 (小组验收, 算法设计和测试写入实验报告: 代码作为附件)

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。