苏州大学实验报告∗

# 2019 年 10 月 29 日

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 院、系 | 计算机学院 | 年级专业 | 计算机科学 | 姓名 | 周泓余 | 学号 | 1927405082 |
| 课程名称 | Python 程序设计 | | | | | 成绩 |  |
| 指导老师 |  | | 同组实验者 | 无 | 实验日期 | 10 月 28 号 | |

表 1: 实验报告相关信息

# 实验目的

实验名称：实验一 **Python** 语言基础

通过本次实验要达到如下目的：

* + 1. 掌握 Python 开发环境的使用
    2. 掌握变量的使用方法
    3. 了解数据的输入和输出方法
    4. 了解并学会选择数据类型
    5. 掌握算术运算符的使用
    6. 掌握 math 模块中常用函数的使用

# 实验内容

* + 1. 从键盘输入两个正整数 a 和 b ，计算并输出 a 除以 b 的商和余数。
    2. 从键盘输入四个整数，并输出其中最大的数。
    3. 编写程序让用户输入自己姓名，输出该姓名字符串的长度。
    4. 一只大象口渴了，要喝 20 升水才能解渴，但现在只有一个深 h 厘米，底面半径为 r 厘米的小圆桶 (h 和 r 都是整数)。问大象至少要喝多少桶水才会解渴。编写程序输入半径和高度，输出需要的桶数（一定是整数）。
    5. 编写程序让用户输入两个平面上点的坐标，计算该两点间的距离。

∗ 注：本文档在 peterlits.com/download/doc/python\_test/report/001/index.html 处

1

* + 1. 产生一个随机 3 位正整数，并将该整数的数字首尾互换输出，例如：157 互换后为

751。

# 实验步骤和结果

* + 1. 实验之前

在给出先关代码前，先定义 print\_cm 来定义输入容错和输入输出格式。

import re

class print\_cm(object):

def init (self, \*, indent:int = 4, ps:str = '>>> ', newline:bool = True): self.ps1 = ' '\*indent

self.ps2 = self.ps1\*2 + ps

self.ps3 = self.ps1\*2 + '{:2}' + ps[2:] self.newline = newline

if self.newline == True: print()

def del (self):

if self.newline == True: print()

def format\_out(self, type, addition:object or [object, '...']=''): if type == 'invalid format':

self.info('you have enter in wrong format, please try a again:') elif type == 'in format':

self.info('Please Enter in format like:' \

' (enter only one {} one time):'.format(addition)) elif type == 'enter':

self.info('Please Enter {} {}:'.format(addition[0], addition[1]))

def format\_re\_in(self, ma\_str, re\_str, res\_f, times = 1) \

-> [tuple(str and '...') or 'else... (by res\_f)']: """to input a format data

ma\_str(match re string): to match the target string

re\_str(re string): input the re string to get data in format str or str tuple

: matched str -> tuple of str

res\_f(result geting function): return the result by this result\_func

: tuple('str', ...) -> the object you want

times: the len of the result list, or the times of loop """

res = []

for i in range(times): while True:

input\_ = self.enter(info=i+1)

# type of result: list of ‘str’ or tuple(‘str, ...)

if input\_ == '':

# if here has no input at all continue

elif re.fullmatch(ma\_str, input\_):

res\_tuple = tuple(re.findall(re\_str, input\_)) res.append( res\_f(res\_tuple) )

break else:

self. format\_out('invalid format') self.info(f'What you had input is: {res}')

return res

def enter(self, type:str = None, times:int = 1, \*, info = None) \

-> [object, '...']:

re\_float = r'(\d+(?:\.\d\*)?|\.\d+)' if type == 'int':

self. format\_out('enter', [times, 'int num'])

return self. format\_re\_in(r' \*[\+|-]? \*\d+ \*', r'\d+', \ lambda x: int(\*x), times)

elif type == 'name':

self. format\_out('enter', [times, 'str']) self.info('Please to know about it:') self.info("iuput: ' Peterlits Zo '(ugly space) ->" \

" output:'Peterlits Zo'", indent=2)

result\_func = lambda res: ' '.join([i for i in res])

return self. format\_re\_in(r'\s\*(\w+\s\*)+', r'\w+', result\_func, times) elif type == 'pair':

self. format\_out('enter', [times, 'pair(2-d vector)']) self. format\_out('in format', 'pair')

self.info("(4, 8) or (9.0, 2) or (.4, 0.2)", indent=2) result\_func = lambda res: tuple(float(i) for i in res)

re\_str = r'\s\*\(\s\*{ f }\s\*,\s\*{ f }\s\*\)\s\*'.format( f = re\_float) return self. format\_re\_in(re\_str, re\_float, result\_func, times)

else:

if info == None:

return input(self.ps2) else:

return input(self.ps3.format(str(info)))

def info(self, \*info:str, indent = 1, with\_ps = False): for i in info:

if with\_ps == True:

print(self.ps1\*(indent-1) + self.ps2, i, sep='') else:

print(self.ps1\*indent, i, sep='')

通过print\_cm对象实例的方法enter和info来进行输入输出。其中enter支持pair，int，name格式，提供人性化的可交互的输入输出。在使用该类的情况下：（所有代码在python3.8.0环境运行)

* + 1. 实验之中

第一题程序如下：

pc = print\_cm()

pc.info('请输入数字以求取商和余数：')

a, b = pc.enter('int', 2)

pc.info(f'{a} 除以 {b} 的商：{a//b}', f'{a} 除以 {b} 的余数：{a b}')

运行结果如下：

请输入数字以求取商和余数：

Please Enter 2 int num: 1 > 324df

you have enter in wrong format, please try a again: What you had input is: []

1 > 23

2 > 243tf

you have enter in wrong format, please try a again: What you had input is: [23]

2 > 12

23 除以 12 的商：1

23 除以 12 的余数：11

第二题程序如下：

pc = print\_cm()

pc.info('请输入数字以求取最大数')

a\_list = pc.enter('int', 4)

pc.info(f'最大的数为{max(a\_list)}')

运行结果如下：

请输入数字以求取最大数

Please Enter 4 int num: 1 > 342

2 > 4653

3 > dfs

you have enter in wrong format, please try a again: What you had input is: [342, 4653]

3 > 3547

4 > 234

最大的数为4653

第三题程序如下：

pc = print\_cm()

pc.info('请输入姓名以求取姓名的长度：')

a, = pc.enter('name')

pc.info(f'格式化后输入的字符串为{repr(a)}，总长度为{len(a)}')

运行结果如下：

请输入姓名以求取姓名的长度：

Please Enter 1 str: Please to know about it:

iuput: ' Peterlits Zo '(ugly space) -> output:'Peterlits Zo'

1. > Pfesjf4 reganj

格式化后输入的字符串为'Pfesjf4 reganj'，总长度为14

第四题程序如下：

import math

pc = print\_cm()

pc.info('请分别输入水桶的深和底面半径，以求出一个特定的大象的喝水桶数：(cm)')

h, r = pc.enter('int', 2)

res = 20 \* 1000 / (math.pi \* r \*\* 2)\*h

pc.info(f'需要的桶数为{math.floor(res)+1}')

运行结果如下：

请分别输入水桶的深和底面半径，以求出一个特定的大象的喝水桶数：(cm) Please Enter 2 int num:

|  |  |  |
| --- | --- | --- |
| 1 | > |  |
| 1 | > |
| 1 | > | 1 |
| 2 | > | 7 |

需要的桶数为130

第五题程序如下：

pc = print\_cm()

pc.info('请输入平面上的两个点：')

p1, p2 = pc.enter('pair', 2)

get\_len = lambda p1, p2: ((p1[0]-p2[0])\*\*2 + (p1[1]+p2[1])\*\*2)\*\*.5

pc.info(f'两点之间的距离为{get\_len(p1, p2)}')

运行结果如下：

请输入平面上的两个点：

Please Enter 2 pair(2-d vector):

Please Enter in format like: (enter only one pair one time): (4, 8) or (9.0, 2) or (.4, 0.2)

1 > (7,6)

2 > --

you have enter in wrong format, please try a again: What you had input is: [(7.0, 6.0)]

1. > yi

you have enter in wrong format, please try a again: What you had input is: [(7.0, 6.0)]

2 > (8,0)

两点之间的距离为6.082762530298219

第六题程序如下：

import random

rand\_int = random.randint(100, 999) pc = print\_cm()

pc.info(f'产生的随机数字为{rand\_int}，互换后为{str(rand\_int)[::-1]}')

运行结果如下：

产生的随机数字为191，互换后为191

**0.4** 实验总结

通过本次实验，我学会了 vs code 开发环境的使用；了解了程序开发的过程，加深理解程序运行的流程。学习基本操作符的使用。