**长沙学院计算机科学与工程学院**

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 课程名称 | Python程序开发基础 | 班级 | 20软件01 | 实验日期 | 2023.2.27 |
| 姓名 | 胡锦华 | 学号 | B20200304125 | 指导教师 | 周景 |
| 成绩 |  | | | | |

### 一. 实验名称

Python基础实验

### 二. 实验目的及要求

掌握Python编程环境的安装配置和使用，以及第三方库文件的安装。掌握Python基本语法、变量和简单数据类型、注释的使用、以及列表的基本用法。

### 三. 实验环境

Python 3.6+

### 四. 实验准备和练习

4.1 安装Python开发环境，并设置好环境变量，在命令行使用python --version可以查看python安装的版本来验证python是否安装正确并设置好了环境变量，

4.2在命令行终端（Windows下使用cmd命令），利用pip install命令安装python软件包，至少要求安装numpy、matplotlib、pygame三个软件包，使用pip list命令可以查看已经安装好的软件包。

4.3 下载并安装集成开发环境（三选一：PyCharm, Visual Studio Code, Sublime）, 使用集成开发环境创建新的工程，并编写HelloWorld程序运行。熟练使用集成开发环境的快捷键，并在集成开发环境安装和使用常用的插件提高开发效率。

4.4 程序练习：

练习2-5：名言 　找一句你钦佩的名人说的名言，将其姓名和名言打印出来。输出应类似于下面这样（包括引号）。

Albert Einstein once said, “A person who never made a mistake never tried anything new.”

练习2-6：名言2 　重复练习2-5，但用变量famous\_person 表示名人的姓名，再创建要显示的消息并将其赋给变量message ，然后打印这条消息。

练习2-7：剔除人名中的空白 　用变量表示一个人的名字，并在其开头和末尾都包含一些空白字符。务必至少使用字符组合"\t" 和"\n" 各一次。

练习3-4：嘉宾名单 　如果你可以邀请任何人一起共进晚餐（无论是在世的还是故去的），你会邀请哪些人？请创建一个列表，其中包含至少三个你想邀请的人，然后使用这个列表打印消息，邀请这些人来与你共进晚餐。

练习3-5：修改嘉宾名单 　你刚得知有位嘉宾无法赴约，因此需要另外邀请一位嘉宾。以完成练习3-4时编写的程序为基础，在程序末尾添加一条print 语句，指出哪位嘉宾无法赴约。修改嘉宾名单，将无法赴约的嘉宾的姓名替换为新邀请的嘉宾的姓名。再次打印一系列消息，向名单中的每位嘉宾发出邀请。

练习3-6：添加嘉宾 　你刚找到了一个更大的餐桌，可容纳更多的嘉宾。请想想你还想邀请哪三位嘉宾。以完成练习3-4或练习3-5时编写的程序为基础，在程序末尾添加一条print 语句，指出你找到了一个更大的餐桌。

使用insert() 将一位新嘉宾添加到名单开头。  
使用insert() 将另一位新嘉宾添加到名单中间。  
使用append() 将最后一位新嘉宾添加到名单末尾。  
打印一系列消息，向名单中的每位嘉宾发出邀请。

练习3-7：缩减名单 你刚得知新购买的餐桌无法及时送达，因此只能邀请两位嘉宾。以完成练习3-6时编写的程序为基础，在程序末尾添加一行代码，打印一条你只能邀请两位嘉宾共进晚餐的消息。  
使用pop() 不断地删除名单中的嘉宾，直到只有两位嘉宾为止。每次从名单中弹出一位嘉宾时，都打印一条消息，让该嘉宾知悉你很抱歉，无法邀请他来共进晚餐。对于余下两位嘉宾中的每一位，都打印一条消息，指出他依然在受邀人之列。  
使用del 将最后两位嘉宾从名单中删除，让名单变成空的。打印该名单，核实程序结束时名单确实是空的。

练习3-8：放眼世界 　想出至少5个你渴望去旅游的地方。

将这些地方存储在一个列表中，并确保其中的元素不是按字母顺序排列的。按原始排列顺序打印该列表。不要考虑输出是否整洁的问题，只管打印原始Python列表。

* 使用sorted() 按字母顺序打印这个列表，同时不要修改它。
* 再次打印该列表，核实排列顺序未变。
* 使用sorted() 按与字母顺序相反的顺序打印这个列表，同时不要修改它。
* 再次打印该列表，核实排列顺序未变。
* 使用reverse() 修改列表元素的排列顺序。打印该列表，核实排列顺序确实变了。
* 使用reverse() 再次修改列表元素的排列顺序。打印该列表，核实已恢复到原来的排列顺序。
* 使用sort() 修改该列表，使其元素按字母顺序排列。打印该列表，核实排列顺序确实变了。
* 使用sort() 修改该列表，使其元素按与字母顺序相反的顺序排列。打印该列表，核实排列顺序确实变了。

练习3-9：晚餐嘉宾 　在完成练习3-4~练习3-7时编写的程序之一中，使用len() 打印一条消息，指出你邀请了多少位嘉宾来共进晚餐。

### 五. 实验内容

前往[codewars.com](http://www.codewars.com)注册账号，完成下面的习题并且在题目的页面提交，通过题目的测试用例来获得分数，并将完成情况在表格中登记。

20软件1班：【腾讯文档】20软件1班Codewars登记

[20软件1班Codewars账号登记](https://docs.qq.com/sheet/DWXhEV2lNZVVhVUFP?tab=BB08J2)

20软件2班：【腾讯文档】20软件2班Codewars登记

[20软件2班Codewars账号登记](https://docs.qq.com/sheet/DWVJVV3Z2YXlnWWxR?tab=BB08J2)

1. 求离整数n最近的平方数

你的任务是找到一个正整数n的最近的平方数

例如，如果n=111，那么nearest\_sq(n)（nearestSq(n)）等于121，因为111比100（10的平方）更接近121（11的平方）。

如果n已经是完全平方（例如n=144，n=81，等等），你需要直接返回n。

代码提交地址：

<https://www.codewars.com/kata/5a805d8cafa10f8b930005ba>

2. 弹跳的球

一个孩子在一栋高楼的第N层玩球。这层楼离地面的高度h是已知的。他把球从窗口扔出去。球弹了起来（例如），弹到其高度的三分之二（弹力为0.66）。他的母亲从离地面1.5米的窗户向外看,母亲会看到球在她的窗前经过多少次（包括球下落和反弹的时候）？

一个有效的实验必须满足三个条件。

* 参数 "h"（米）必须大于0
* 参数 "bounce "必须大于0且小于1
* 参数 “window "必须小于h。

如果以上三个条件都满足，返回一个正整数，否则返回-1。

注意:只有当反弹球的高度严格大于窗口参数时，才能看到球。

代码提交地址：

<https://www.codewars.com/kata/5544c7a5cb454edb3c000047/train/python>

3. 括号匹配

写一个函数，接收一串括号，并确定括号的顺序是否有效。如果字符串是有效的，它应该返回True，如果是无效的，它应该返回False。

例如：

"(){}[]" => True

"([{}])" => True

"(}" => False

"[(])" => False

"[({})](]" => False

代码提交地址：

<https://www.codewars.com/kata/5277c8a221e209d3f6000b56>

提示：

python中没有内置堆栈数据结构，可以直接使用list来作为堆栈，其中append方法用于入栈，pop方法可以出栈。

4. 从两个列表来创建字典

有两个列表，可能有不同的长度。第一个由 keys 组成，第二个由 values 组成。写一个函数 createDict(keys, values) ，返回由 keys 和 values 创建的 dictionary。如果没有足够的值，其余的键应该有一个 None 值。如果没有足够的键，就忽略其余的值。

Example 1:

keys = ['a', 'b', 'c', 'd']

values = [1, 2, 3]

createDict(keys, values) # returns {'a': 1, 'b': 2, 'c': 3, 'd': None}

Example 2:

keys = ['a', 'b', 'c']

values = [1, 2, 3, 4]

createDict(keys, values) # returns {'a': 1, 'b': 2, 'c': 3}

代码提交地址：

<https://www.codewars.com/kata/5533c2a50c4fea6832000101/python>

5. 偶数或者奇数

创建一个函数接收一个整数作为参数，当整数为偶数时返回”Even”当整数位奇数时返回”Odd”。

代码提交地址：

<https://www.codewars.com/kata/53da3dbb4a5168369a0000fe>

### 六. 实验过程和结果

将完成的代码文本（包括代码的注释）和测试结果复制到下面，注意不要使用截图。

1. 求离整数n最近的平方数

代码：

def nearest\_sq(n):

    return round(n \*\* 0.5) \*\* 2

输出：

print(nearest\_sq(111))

121

2. 弹跳的球

代码：

def bouncing\_ball(h, bounce, window):

    if h > 0 and bounce > 0 and bounce < 1 and window < h:

        times = 0

        while window < h:

            h = h \* bounce

            if window < h:

                times+=2

            else:

                times+=1

                break

        return times

    else:

        return -1

输出：

print(bouncing\_ball(30, 0.75, 1.5))

21

3. 括号匹配

代码：

def valid\_braces(string):

    x = len(string)

    if x % 2 != 0:

        return False

    l = []

    for j in string:

        if j in ['(', '[', '{']:

            l.append(j)

        else:

            if len(l) != 0:

                if l[-1] == '(' and j == ')':

                    del l[-1]

                elif l[-1] == '[' and j == ']':

                    del l[-1]

                elif l[-1] == '{' and j == '}':

                    del l[-1]

                else:

                    return False

            else:

                return False

    return len(l) == 0

输出：

print(valid\_braces("(){[]}()"))

True

4. 从两个列表来创建字典

代码：

def create\_dict(keys, values):

    d = {}

    if not values:

        for x in keys:

            d[x] = None

    for i,x in enumerate(keys):

        for j,y in enumerate(values):

            if (i == j):

                d[x] = y

                continue

            elif (i > j):

                d[x] = None

                continue

            else:

                break

    return d

输出：

print(create\_dict(['a', 'b', 'c', 'd'], [1, 2, 3]))

{'a': 1, 'b': 2, 'c': 3, 'd': None}

5. 偶数或者奇数

代码：

def even\_or\_odd(number):

    if number % 2 == 0:

        return 'Even'

    else:

        return 'Odd'

输出：

print(even\_or\_odd(22))

Even

### 七. 实验小结

Python对代码缩进要求十分严格，四格缩进代表一个新的代码块，并且一些条件语句后面都需要接一个冒号。然后就是Python的类型很强大，不需要复杂的定义就可以直接输入输出以及进行各种操作。还有Python的列表理解，真的简洁优雅，一行代码就可以完成很复杂的功能。