

实验二 Python变量、简单数据类型 班级： 21计科1班

学号： B20210302112

姓名： 张栢棋

Github地址： <https://github.com/2580wasd/pykc2.git>

实验目的 使用VSCode编写和运行Python程序 学习Python变量和简单数据类型 实验环境 Git Python 3.10
VSCode VSCode插件 实验内容和步骤 第一部分 实验环境的安装

安装Python，从Python官网下载Python 3.10安装包，下载后直接点击可以安装：Python官网地址 为了在VSCode集成环境下编写和运行Python程序，安装下列VScode插件 Python Python Environment Manager Python Indent Python Extended Python Docstring Generator Jupyter indent-rainbow Jinja 第二部分 Python变量、简单数据类型和列表简介

完成教材《Python编程从入门到实践》下列章节的练习：

第2章 变量和简单数据类型 第三部分 在Codewars网站注册账号，完成下列Kata挑战：

第1题：求离整数n最近的平方数（Find Nearest square number） 难度：8kyu

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：

```
import math n = int(input("请输入一个整数: ")) square = round(math.sqrt(n)) ** 2 print("离整数", n, "最近的平方数是: ", square) return n
```

第2题：弹跳的球 (Bouncing Balls)

难度：6kyu

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w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>

第3题：元音统计(Vowel Count)

难度：7kyu

返回给定字符串中元音的数量（计数）。对于这个Kata，我们将考虑a、e、i、o、u作为元音（但不包括y）。输入的字符串将只由小写字母和/或空格组成。

代码提交地址：<https://www.codewars.com/kata/54ff3102c1bad923760001f3>

第4题：偶数或者奇数 (Even or Odd)

难度：8kyu

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”当整数为奇数时返回“Odd”。代码提交地址：<https://www.codewars.com/kata/53da3dbb4a5168369a0000fe>


第四部分

使用Mermaid绘制程序流程图

安装Mermaid的VSCode插件：

- Markdown Preview Mermaid Support
- Mermaid Markdown Syntax Highlighting

使用Markdown语法绘制你的程序绘制程序流程图（至少一个），Markdown代码如下：

 程序流程图

显示效果如下：

```
flowchart LR
    A[Start] --> B{Is it?}
    B -->|Yes| C[OK]
    C --> D[Rethink]
    D --> B
    B -.->|No| E[End]
```

查看Mermaid流程图语法-->[点击这里](#)

使用Markdown编辑器（例如VScode）编写本次实验的实验报告，包括[实验过程与结果](#)、[实验考查](#)和[实验总结](#)，并将其导出为 **PDF格式** 来提交。

实验过程与结果

请将实验过程与结果放在这里，包括：

- [第二部分 Python变量、简单数据类型和列表简介](#)
- [第三部分 Codewars Kata挑战](#)
- [第四部分 使用Mermaid绘制程序流程图](#)

注意代码需要使用markdown的代码块格式化，例如Git命令行语句应该使用下面的格式：

 Git命令

显示效果如下：

```
git init
git add .
git status
git commit -m "first commit"
```

如果是Python代码，应该使用下面代码块格式，例如：

 Python代码

显示效果如下：

```
def add_binary(a,b):
    return bin(a+b)[2:]
```

第三部分的解答

第1题：求离整数n最近的平方数（Find Nearest square number）解决方法

代码：

```
def nearest_sq(n):
    # pass
    return round(n**0.5)**2
```

流程图：

```
flowchart LR
    A(开始) --> B[n开平方]
    B --> C[四舍五入]
    C --> D[n平方]
    D --> E(End)
```

第2题：弹跳的球（Bouncing Balls）解决方法

代码：

```
def bouncing_ball(h, bounce, window):
    # your code
    from math import log
    if h<=0 or bounce>=1 or bounce<=0 or window>=h:
```

```

    return -1
t=0 # t为看到球的次数
b=window/h # b为球初始下落高度与窗高的比值
b=log(b,bounce) # 计算以bounce为基的b的对数, 即球能弹回到窗户上方的次数
if b>int(b): #若b>int(b),则球有弹回到窗户之上
    t=int(b)*2 #每弹回的窗户之上, 可以看到2次
return t+1 #加上初始下落时看到的那一次

```

流程图:

```

flowchart TD
    a(开始)-->b{?输入的数据是否符合要求}
    b-->|NO| c[return -1]
    c-->d(结束)
    b-->|YES| e[t=0,b=window/h]
    e-->f["b=log(b,bounce)"]
    f-->g{"?if b>int(b)"}
    g-->|YES| h["t=int(b)*2"]
    g-->|NO| i[return t+1]
    h-->i
    i-->j(结束)

```

第3题: 元音统计(Vowel Count)解决方法

代码:

```

def get_count(sentence):
    return sentence.count('a') + sentence.count('e') + sentence.count('i') +
    sentence.count('o') + sentence.count('u')

```

流程图:

```

flowchart TD
    a(开始)-->b[计算sentence中元音的数目之和]
    b-->c[return得到的结果]
    c-->d(结束)

```

注意: 不要使用截图, Markdown文档转换为Pdf格式后, 截图可能会无法显示。

实验考查

实验考查 请使用自己的语言并使用尽量简短代码示例回答下面的问题, 这些问题将在实验检查时用于提问和答辩以及实际的操作。

Python中的简单数据类型有那些？我们可以对这些数据类型做哪些操作？ Python的六个标准数据类型分类方式：按数据可变性 不可变数据（3个） Number（数字）、String（字符串）、Tuple（元组） 可变数据（3个） List（列表）、Dictionary（字典）、Set（集合） 按照类型区分 基本数据类型（2个） Number（数字）、String（字符串） 复合数据类型（4个） List（列表）、Tuple（元组）、Set（集合）、Dictionary（字典） 为什么说Python中的变量都是标签？

因为对变量的常见解释是存储值的容器，对于C语言来说这个解释是合理的，但是对python来说不算精确，python中的变量不是容器，而是指向python对象的标签，对象位于解释器的命名空间中，但是任意数量的标签可以指向同一个对象，当对象发生变化时，所有指向它的变量的值都会改变。

有哪些方法可以提高Python代码的可读性？ 1：使用有变量的注释名 2：添加注释

实验总结

在这次实验中，我学会了基础的一些python代码，并可以实现简单的一些功能，当然也会更了解Python的语法规则与结构，我会更加努力地学习python代码来实现更多比较复杂的功能。