

实验二 Python变量、简单数据类型

班级： 21计科2班

学号： B20210202314

姓名： 朱华畅

Github地址： https://github.com/Lakzhu/python_project

CodeWars地址： <https://www.codewars.com/users/Lak朱>

实验目的

1. 使用VSCode编写和运行Python程序
2. 学习Python变量和简单数据类型

实验环境

1. Git
2. Python 3.10
3. VSCode
4. VSCode插件

实验内容和步骤

第一部分

实验环境的安装

1. 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](#)
 2. 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](https://www.codewars.com/)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。代码提交地址 <https://www.codewars.com/kata/5a805d8cafa10f8b930005ba>

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：

```
graph 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](#)

第一题 Find Nearest square number

```
def nearest_sq(n):
    f = pow(n, 0.5)
    a = int(f//1)
    if a*a == n:
        return n
    b = (a+1)//1
    if n - a*a < b*b - n:
        return a*a
    elif n - a*a > b*b - n:
        return b*b
```

```
graph LR
    A[f等于n开平方] --> B[a等于取整]
```

```

B --> C{a*a是否等于n? }
C -->|是| D[返回n]
C -->|否| E[b等于a+1除以1]
E --> F{n-a*a是否小于b*b-n? }
F -->|是| G[返回a*a]
F -->|否| H[返回b*b]

```

8 kyu Find Nearest square number ✓

☆ 276 82 90% of 2,208 7,108 of 19,199 riyakayal

Instructions Output Past Solutions

Time: 563ms Passed: 1105 Failed: 0

Test Results:

- Fixed Tests
 - Basic Test Cases (5 of 5 Assertions)
 - Completed in 0.13ms
- Random Tests
 - Small Random Tests (100 of 100 Assertions)
 - Big Random Tests (1000 of 1000 Assertions)
 - Completed in 22.45ms

You have passed all of the tests! :)

Solution

```

1 def nearest_sq(n):
2     f = pow(n, 0.5)
3     a = int(f//1)
4     if a*a == n:
5         return n
6     b = (a+1)//1
7     a = a//1
8     if n - a*a < b*b - n:
9         return a*a
10    elif n - a*a > b*b - n:
11        return b*b
12
13

```

Outstanding! You may take your time to refactor/comment your solution. Submit when ready. ✕

Sample Tests

```

3
4 @test.describe("Fixed Tests")
5 def fixed_tests():
6     @test.it('Basic Test Cases')
7     def basic_test_cases():
8         test.assert_equals(nearest_sq(1), 1)
9         test.assert_equals(nearest_sq(2), 1)
10        test.assert_equals(nearest_sq(10), 9)
11        test.assert_equals(nearest_sq(111), 121)
12        test.assert_equals(nearest_sq(9999), 10000)

```

SKIP VIEW SOLUTIONS DISCUSS (43) RESET TEST SUBMIT

第二题 Bouncing Balls

```

def bouncing_ball(h, bounce, window):
    sum = 0;
    if h > window and 0 < bounce < 1 and h > 0:
        while(h>window):
            sum+=1
            h = h * bounce
            if h > window:
                sum+=1
    else:
        return -1
    return sum

```

6 kyu

Bouncing Balls

☆ 2238

👤 373

83% of 7,497

🏆 24,920 of 78,632

👤 g964

🚩 1 Issue Reported

Instructions

Output

Time: 516ms Passed: 61 Failed: 0

Test Results:

Tests

Fixed Tests (11 of 11 Assertions)

Random tests (50 of 50 Assertions)

Completed in 3.13ms

You have passed all of the tests! :)

Solution

Search:| (Use /re/ syntax for regexp search)

```

3     if h > window and 0 < bounce < 1 and h > 0:
4         while(h>window):
5             sum+=1
6             h = h * bounce
7             if h > window:
8                 sum+=1
9         else:
10            return -1
11    return sum

```

Sample Tests

```

3     def testing(n, bounce, window, exp):
4         actual = bouncing_ball(h, bounce, window)
5         test.assertEqual(actual, exp)
6
7     @test.it('Fixed Tests')
8     def tests():
9         testing(2, 0.5, 1, 1)
10        testing(3, 0.66, 1.5, 3)
11        testing(30, 0.66, 1.5, 15)
12        testing(30, 0.75, 1.5, 21)

```

SKIP

DISCUSS (371)

RESET

TEST

ATTEMPT

第三题 Vowel Count

```

def get_count(sentence):
    sum = 0
    for i in sentence:
        if i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u':
            sum+=1
    return sum

```

7 kyu

Vowel Count

☆ 2623

👤 658

📈 89% of 29,756

🏆 114,678 of 355,856

👤 jayeshcp

🚩 1 Issue Reported

Instructions

Output

Time: 524ms

Passed: 105

Failed: 0

Test Results:

Sample tests

Should count all vowels

Should not count "y"

Should return 0 when no vowels

Should return 0 for empty string

Should return 5 for "abracadabra"

Completed in 0.13ms

Random tests

Random tests (100 of 100 Assertions)

Completed in 7.50ms

You have passed all of the tests! :)

Solution

```
1 def get_count(sentence):
2     sum = 0
3     for i in sentence:
4         if i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u':
5             sum+=1
6     return sum
```

Sample Tests

```
7 @test.it("Should count all vowels")
8 def all_vowels():
9     test.assert_equals(get_count("aeiou"), 5, f"Incorrect answer for \"aeiou\"")
10
11 @test.it("Should not count \"y\"")
12 def only_y():
13     test.assert_equals(get_count("y"), 0, f"Incorrect answer for \"y\"")
14
15 @test.it("Should return 0 when no vowels")
```

SKIP

UNLOCK SOLUTIONS

DISCUSS (206)

RESET

TEST

ATTEMPT

第四题 Even or Odd

```
def even_or_odd(number):
    if number%2 == 0:
        return 'Even'
    else:
        return 'Odd'
```

8 kyu

Even or Odd

☆ 2072

👤 430

📈 91% of 30,898

🏆 117,748 of 418,781

👤 suuuzi

🚩 1 Issue Reported

Instructions

Output

Time: 572ms

Passed: 111

Failed: 0

Test Results:

Fixed Tests

Basic Test Cases (11 of 11 Assertions)

Completed in 0.12ms

Random Tests

Testing for even_or_odd(2381642)

Testing for even_or_odd(7478014)

Testing for even_or_odd(6172409)

Testing for even_or_odd(2048796)

Testing for even_or_odd(3608760)

Testing for even_or_odd(7208736)

Testing for even_or_odd(1735546)

Testing for even_or_odd(3360375)

Testing for even_or_odd(5714137)

Solution

```
1 def even_or_odd(number):
2     if number%2 == 0:
3         return 'Even'
4     else:
5         return 'Odd'
```

Correct! You may take your time to refactor/comment your solution. Submit when ready.

Sample Tests

```
1 import codewars_test as test
2 from solution import even_or_odd
3
4 @test.describe("Fixed Tests")
5 def fixed_tests():
6     @test.it('Basic Test Cases')
7     def basic_test_cases():
8         test.assert_equals(even_or_odd(2), "Even")
9         test.assert_equals(even_or_odd(1), "Odd")
10         test.assert_equals(even_or_odd(0), "Even")
```

SKIP

UNLOCK SOLUTIONS

DISCUSS (206)

RESET

TEST

ATTEMPT

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

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。

1. 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？

字符串；改变大小写字符串比对
整数；基本算数运算；取余；取整；比较大小；
浮点数；基本算数运算；取余取整比较大小；
布尔值；当做true和false使用；

2. 为什么说Python中的变量都是标签？

因为python的变量创建都是创建在内存中的，然后变量名只是用与指向那个内存中的，你可用a和b同时等于一个数，然后查看他们的地址会发现他们是一样的，所以说是标签

3. 有哪些方法可以提高Python代码的可读性？

遵循python之禅，多做注释。

实验总结

总结一下这次实验你学习和使用到的知识，例如：编程工具的使用、数据结构、程序语言的语法、算法、编程技巧、编程思想。

这次实验让我熟悉了基本的python语言，练习了python语法的使用，如while循环，for循环，if条件语句.....熟悉了如何绘制基本的流程图。