

# 实验二 Python变量、简单数据类型

---

班级： 21计科3班

学号： B20210302321

姓名： 向钟源

Github地址： <[Ch1rs \(github.com\)](#)>

CodeWars地址： <[Ch1rs X | Codewars](#)>

---

## 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—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

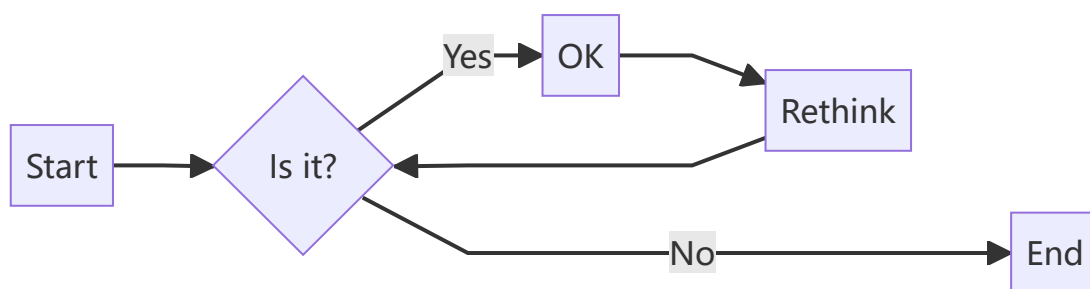
安装Mermaid的VSCode插件：

- 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：

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

显示效果如下：



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](#)

2.1 简单消息 将一条消息赋给变量，并将其打印出来。

```
message = "i am a teapot"
print(message)
```

2.2 多条简单消息 将一条消息赋给变量，并将其打印出来；再将变量的值修改为一条新消息，并将其打印出来。

```
message = "hhh"
print(message)

message = "i don't like sleep"
print(message)
```

2.3 个性化消息 用变量表示一个人的名字，并向其显示一条消息。显示的消息非常简单，如下所示。

Hello Eric,would you like to learn some Python today?

```
message = "Eric"
print(f"Hello {message},would you like to learn some Python today?")
```

2.4 调整名字的大小写 用变量表示一个人的名字，再分别以全小写、全大写和首字母大写的方式显示这个人名。

```
name = "kris wu"
print(name.lower())
print(name.upper())
print(name.title())
```

2.5 名言 1 找到你钦佩的名人说的一句名言，将这个人的姓名和名言打印出。类似于：

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

```
print("蔡徐坤 once said, '一个真正的man'")
```

2.6 名言2 重复2.5，但用famous\_person表示名人的姓名，再创建要显示的消息并将其赋给message,然后打印

```
name = "蔡徐坤"
saying = "一个真正的man"
print(f'{name} once said,"{saying}").')
```

2.7 删除人名中的空白 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\n和\t各一次

打印人名后，显示其开头和末尾的空白。然后，分别使用函数lstrip()、rstrip()、strip()对人名进行出力，并将结果打印

```
name = " \nkris wu \t"
print(name)
print(name.lstrip())
print(name.rstrip())
print(name.strip())
```

2.8 请将值'python\_notes.txt'赋给变量filename,再使用 `removesuffix()` 方法来显示不包括扩展名的文件名

```
filename = 'python_notes.txt'
print(filename.removesuffix('.txt'))
```

2.9 数字8 编写四个表达式，分别使用加减乘除运算，但结果都是8

```
print(5 + 3)
print(10 - 2)
print(2 * 4)
print(16 * 2)
```

2.10 最喜欢的数 用一个变量表示最喜欢的数，再使用这个变量创建一条消息，指出你最喜欢的数是什么，再打印出来

```
favorite_num = 77
print("what is your favorite number?")
print(f"My favorite number is {favorite_num}.")
```

2.11/2.12 pass

- [第三部分 Codewars 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

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

### 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。**

```
def bouncing_ball(h, bounce, window):
    # your code
    if h <= 0 or not (0 < bounce < 1) or window >= h:
        return -1
    count = 1

    while h * bounce > window:
        h *= bounce
        count += 2
    return count
```

### 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。

```
def get_count(sentence):
    count = 0
    sentence_length = len(sentence)
    list1 = list(sentence)
    for value in list1[:sentence_length]:
        if value == 'a' or value == 'e' or value == 'i' or value == 'o' or value == 'u':
            count += 1
    return count
pass
```

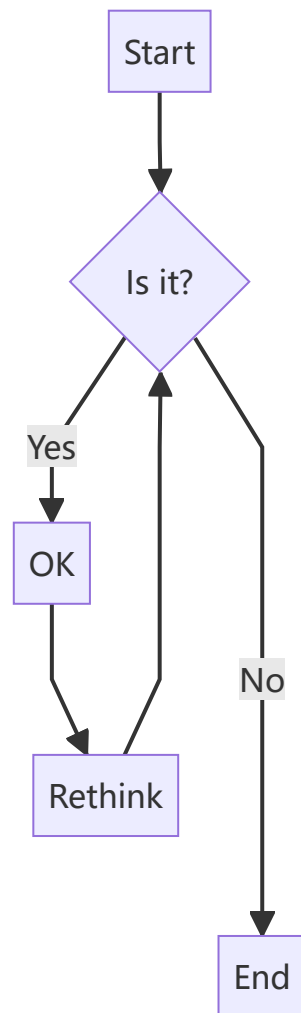
### 第4题：偶数或者奇数 (Even or Odd)

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”。

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

- [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：

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

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

显示效果如下：

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

代码运行结果的文本可以直接粘贴在这里。

**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？

**答：1、数字型：整型、浮点型、布尔型、复数类型(没学) 2、字符串 3、list 列表 4、tuple 元组 5、集合(好像暂时也没学) 6、字典；**

**我们可以做出如下三个操作：**

- 整数和浮点数可以进行基本的数学运算，如加法、减法、乘法、除法等。
- 布尔值可以进行逻辑运算，如与 (and)、或 (or)、非 (not) 等。
- 字符串可以进行拼接（用加号连接）、重复、索引（获取指定位置的字符）等操作。

2. 为什么说Python中的变量都是标签？

**答：比如我们创建了一个变量，名字叫做message，`message = 99`，**

**这个时候在内存当中创建了一个空间用来存放99，当我们让 `message = 88` 后，99的空间并没有消失，python只是重新创建了一个存放88的空间，让后将贴在99的标签message撕下来，贴在了88上**

3. 有哪些方法可以提高Python代码的可读性？

**答：目前来说，我觉得可以从以下几个方面来提高python代码可读性：**

- 使用有意义的变量名：使用描述性的变量名可以使代码更易于理解和维护。避免使用单个字母或无意义的变量名。
- 缩进和格式化：使用一致的缩进和代码格式化风格，如使用4个空格进行缩进，适当地使用空行和垂直对齐等，以使代码结构清晰。
- 添加注释：在代码中添加注释可以解释代码的目的、实现细节或特定算法。注释应该清晰、简洁，并与代码保持同步
- 避免过长的代码行：将过长的代码行分成多行，以提高可读性。可以使用括号或续行符号 (\) 来实现。



# 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。

1、体会到了再python中，变量不是盒子而是一个标签。

2、体会到了全英文在线OJ的乐趣

3、学会了mermaid的基础语法