

# 实验二 Python变量、简单数据类型

---

班级： 21计科03

学号： B20210302322

姓名： 董一浩

Github地址： <https://github.com/ByL1eng/xuexi>

CodeWars地址： <https://www.codewars.com/users/ByL1eng>

---

## 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>

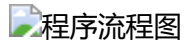
## 第四部分

## 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：



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](#)

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：



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：



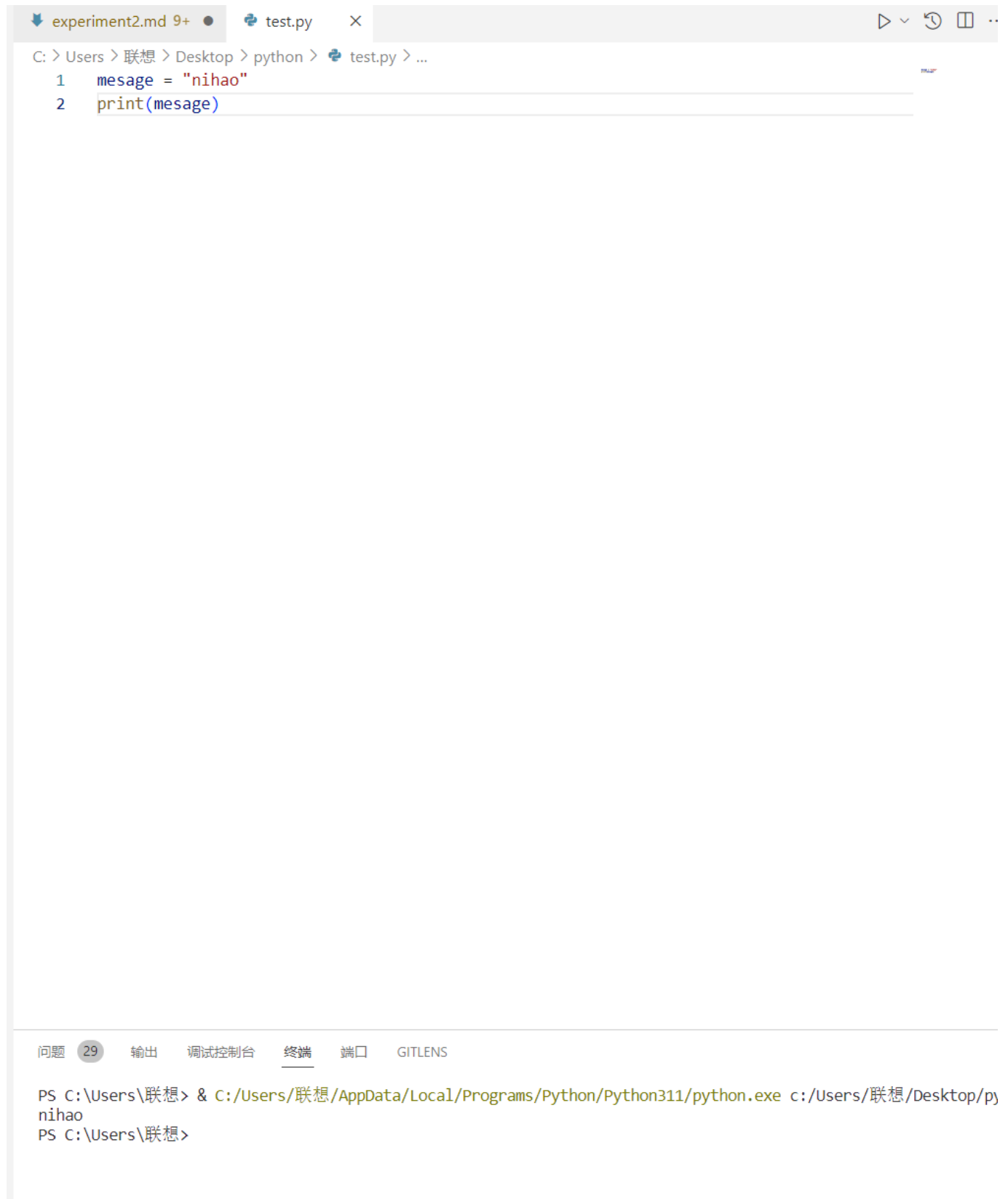
显示效果如下：

```
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。

## 第二部分

### 练习 2.1

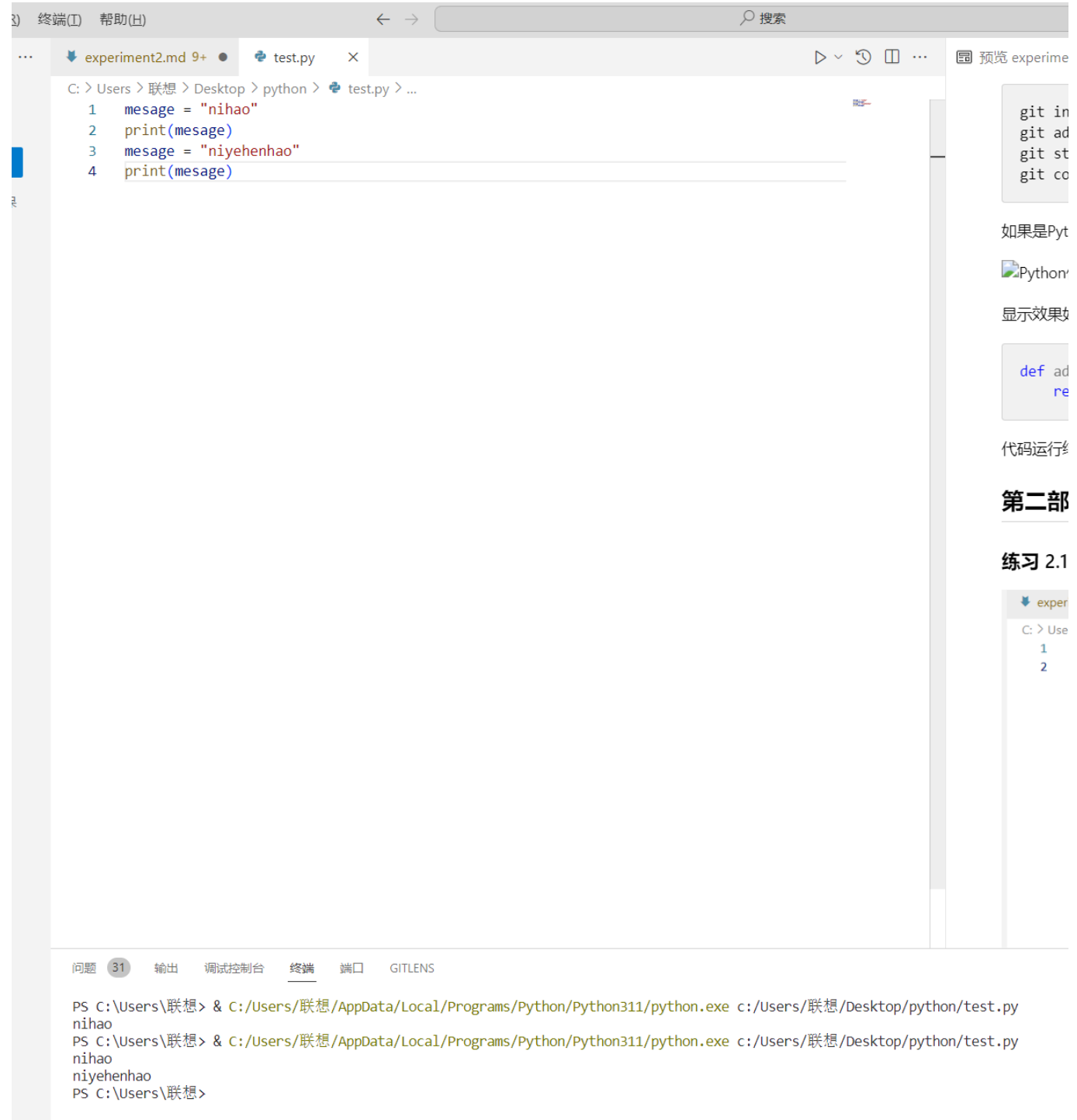


The image shows a Visual Studio Code editor window with two tabs: 'experiment2.md 9+' and 'test.py'. The 'test.py' tab is active, displaying a Python script with two lines: `1 message = "nihao"` and `2 print(message)`. The file explorer on the left shows the file path: `C:\Users\联想\Desktop\python\test.py`. At the bottom, the 'Terminal' panel is open, showing the command prompt output: `PS C:\Users\联想> & C:/Users/联想/AppData/Local/Programs/Python/Python311/python.exe c:/Users/联想/Desktop/py` followed by the output `nihao` on a new line.

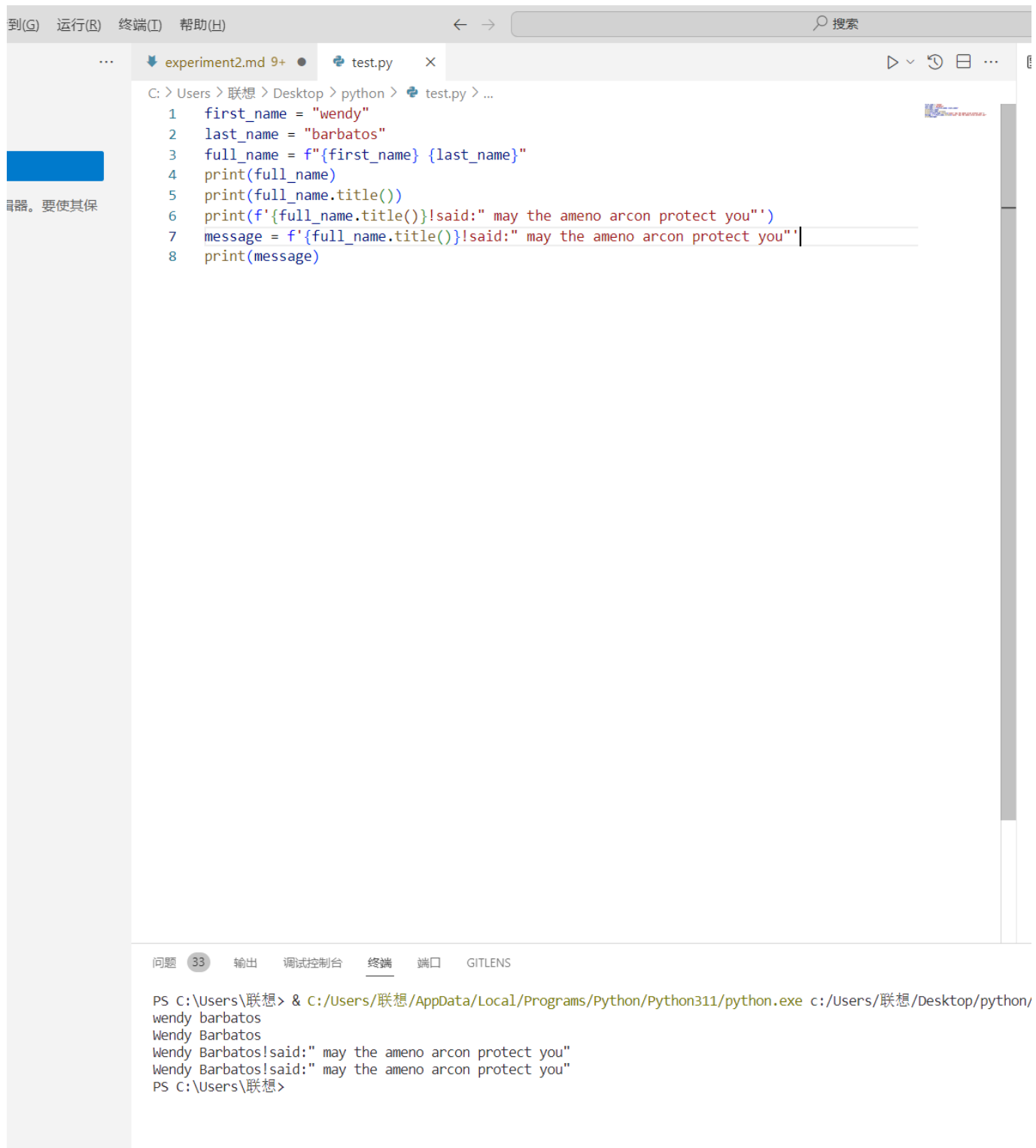
```
experiment2.md 9+ test.py X
C: > Users > 联想 > Desktop > python > test.py > ...
1  message = "nihao"
2  print(message)

问题 29 输出 调试控制台 终端 端口 GITLENS
PS C:\Users\联想> & C:/Users/联想/AppData/Local/Programs/Python/Python311/python.exe c:/Users/联想/Desktop/py
nihao
PS C:\Users\联想>
```

## 练习 2.2



练习 2.3-2.6



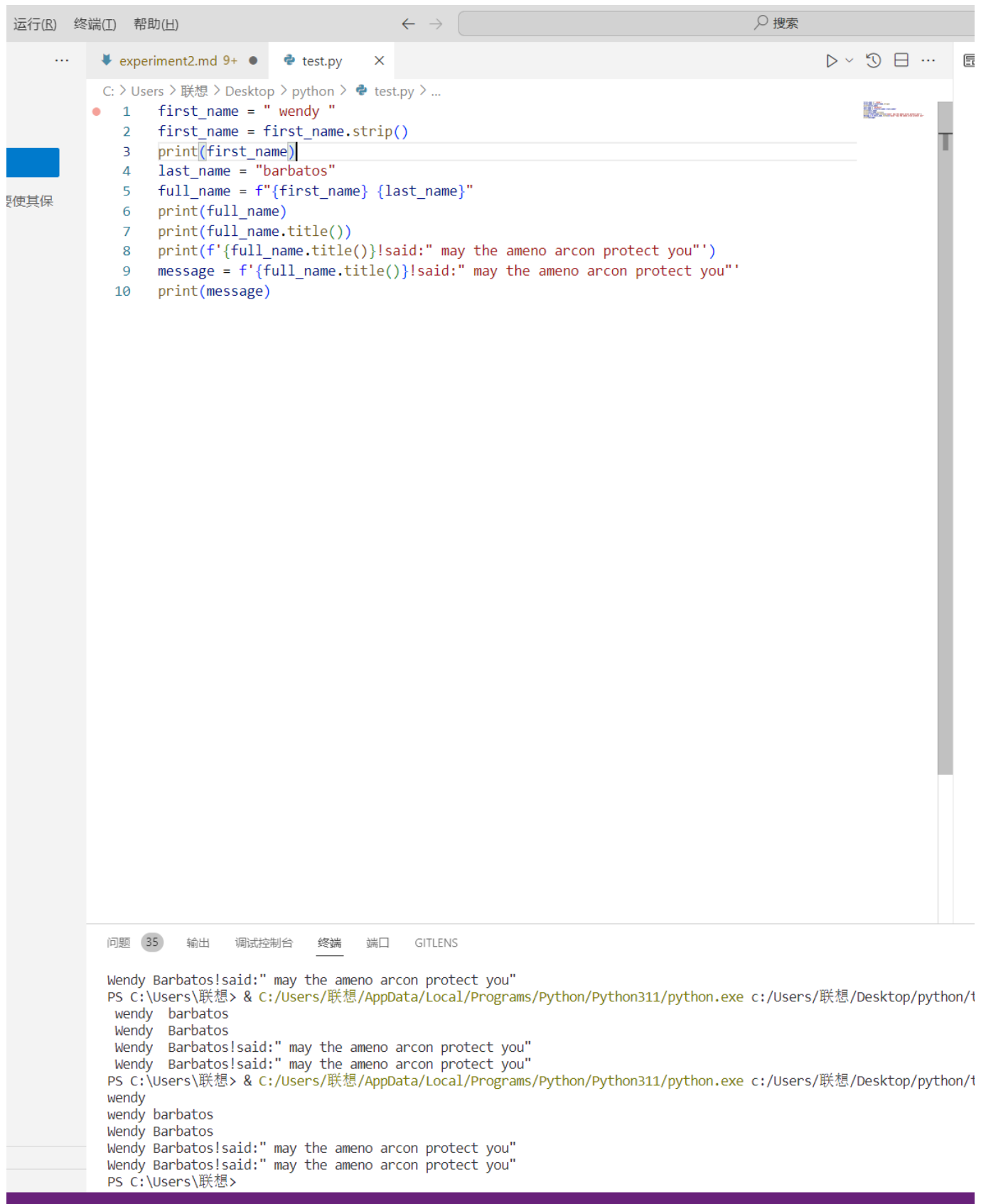
The screenshot shows a code editor with a file named `test.py` open. The code defines variables for first and last names, concatenates them into a full name, and prints it in title case. It then constructs a message string using the full name and prints it. Below the code editor, a terminal window shows the command to run the script and the resulting output.

```
C: > Users > 联想 > Desktop > python > test.py > ...
1 first_name = "wendy"
2 last_name = "barbatos"
3 full_name = f'{first_name} {last_name}'
4 print(full_name)
5 print(full_name.title())
6 print(f'{full_name.title()}!said:" may the ameno arcon protect you"')
7 message = f'{full_name.title()}!said:" may the ameno arcon protect you"'
8 print(message)
```

问题 33 输出 调试控制台 终端 端口 GITLENS

```
PS C:\Users\联想> & C:/Users/联想/AppData/Local/Programs/Python/Python311/python.exe c:/Users/联想/Desktop/python/
wendy barbatos
Wendy Barbatos
Wendy Barbatos!said:" may the ameno arcon protect you"
Wendy Barbatos!said:" may the ameno arcon protect you"
PS C:\Users\联想>
```

## 练习 2.7



The screenshot shows a code editor with a file named `test.py` open. The code defines variables for a first name, last name, and full name, then prints them and a message. The terminal output shows the execution of the script, displaying the first name, last name, full name, and the message.

```
C: > Users > 联想 > Desktop > python > test.py > ...
1 first_name = "wendy"
2 first_name = first_name.strip()
3 print(first_name)
4 last_name = "barbatos"
5 full_name = f"{first_name} {last_name}"
6 print(full_name)
7 print(full_name.title())
8 print(f'{full_name.title()}!said:" may the ameno arcon protect you"')
9 message = f'{full_name.title()}!said:" may the ameno arcon protect you"'
10 print(message)
```

Wendy Barbatos!said:" may the ameno arcon protect you"

PS C:\Users\联想> & C:/Users/联想/AppData/Local/Programs/Python/Python311/python.exe c:/Users/联想/Desktop/python/t  
wendy barbatos  
Wendy Barbatos  
Wendy Barbatos!said:" may the ameno arcon protect you"  
Wendy Barbatos!said:" may the ameno arcon protect you"

PS C:\Users\联想> & C:/Users/联想/AppData/Local/Programs/Python/Python311/python.exe c:/Users/联想/Desktop/python/t  
wendy  
wendy barbatos  
Wendy Barbatos  
Wendy Barbatos!said:" may the ameno arcon protect you"  
Wendy Barbatos!said:" may the ameno arcon protect you"

PS C:\Users\联想>

## 练习 2.11





```

C: > Users > 联想 > Desktop > python > test.py > ...
1  first_name = " wendy "
2  first_name = first_name.strip()
3  print(first_name)
4  last_name = "barbato"
5  full_name = f"{first_name} {last_name}"
6  print(full_name)
7  print(full_name.title())
8  print(f'{full_name.title()}!said:" may the ameno arcon protect you"'#我是温迪巴巴托斯
9  message = f'{full_name.title()}!said:" may the ameno arcon protect you"'
10 print(message)

```

### 第三部分

#### 第1题：求离整数 $n$ 最近的平方数 (Find Nearest square number)

```

def nearest_sq(n):
    sqrt_n = int(math.sqrt(n))
    sq1 = sqrt_n ** 2
    sq2 = (sqrt_n+1) ** 2

    if abs(n-sq1) <= abs(n-sq2):
        return sq1
    else:
        return sq2

```

#### 第2题：弹跳的球 (Bouncing Balls)

```

def bouncing_ball(h, bounce, window):
    if h <= 0 or h <= window or bounce <= 0 or bounce>=1:
        return -1
    count=0
    while h > window:
        h *= bounce
        count += 1
    return count * 2 - 1
\

```

#### 第3题：求一个数的质因数 (Prime Factors)

```
def get_count(sentence):  
    count = 0  
    for letters in sentence:  
        if letters == "a":  
            count+=1  
        if letters == "e":  
            count+=1  
        if letters == "i":  
            count+=1  
        if letters == "o":  
            count+=1  
        if letters == "u":  
            count+=1  
    return count
```

#### 第4题：求一个数的质因数 (Prime Factors)

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

### 第四部分

```
flowchart LR  
    number-->|"%2"| result  
  
    result-->|"!0| odd;  
  
    result-->| "= 0| even;
```

**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：Python中的简单数据类型有：整数、浮点数、字符串、布尔值。

## 整数

整数可以被表示为任意长度，但通常只使用十进制，整数可以表示为正数或负数，整数可以表示为任何长度，但通常只使用十进制。

## 浮点数

浮点数是带小数部分的数字。浮点数可以表示为正数或负数。浮点数可以表示为任何长度，但通常只使用十进制。

## 字符串

字符串是字符的序列。字符串可以表示为正数或负数。字符串可以表示为任何长度，但通常只使用十进制。

## 布尔值

布尔值是True或False。布尔值可以表示为正数或负数。布尔值可以表示为任何长度，但通常只使用十进制。

## 2. 为什么说Python中的变量都是标签？

答：Python中的变量都是标签，Python中的变量是内存中值的引用，变量本身没有值，变量只是提供了一种更方便

## 3有哪些方法可以提高Python代码的可读性？

答：1. 变量命名规范： 2. 缩进规范， 3. 注释规范， 4. 空行规范， 5. 代码排版规范，

## 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。