

G4-18 课后练习 —— 综合项目二

姓名： ____ 日期： ____

面向7-10岁少儿 | G4 编程高手 | 函数与模块综合

一、填空题（回顾概念）

题目1： 用「def」 定义函数时， 函数名后面的括号里可以放 ____ 。 「return」 关键字的作用是 ____ 并 ____ 值。

题目2： 模块导入方式中， 「import math」 后要用 ____ 来调用 sqrt 函数； 「from math import sqrt」 后可以 ____ 调用 sqrt。

二、代码补全（补写代码）

题目3： 编写几个数学工具函数， 并组合使用。 补全代码。

```
import math

def circle_area(radius):
    """计算圆的面积"""
    return ____ .pi * radius ____ 2

def distance(x1, y1, x2, y2):
    """计算两点之间的距离"""
    return math.sqrt((x2 - x1) ____ 2 + (y2 - y1) ____ 2)

def factorial(n):
    """计算阶乘"""
    if n == 0 ____ n == 1:
        return ____
    else:
        return n * factorial(____)

# 测试
print(f"半径5的圆面积: {circle_area(5):.2f}")
print(f"点(0,0)到(3,4)的距离: {distance(0, 0, 3, 4)}")
print(f"5的阶乘: {factorial(5)}")
```

题目4：编写一个"计时器"装饰器，计算函数执行时间。补全代码。

```
import ____
```

```
def timer(func):
    """计算函数执行时间的装饰器"""
    def wrapper(*args, **kwargs):
        start = time.__()
        result = func(*args, **kwargs)
        end = time.__()
        print(f"{func.____} 执行了 {end - start:.4f} 秒")
        return ____
    return wrapper
```

```
@timer
def slow_sum(n):
    total = 0
    for i in range(n):
        total += i
    return total
```

```
print(f"结果: {slow_sum(1000000)}")
```

三、创作题（动手编程）

题目5：综合项目：编写"猜单词游戏"。

- 定义一个单词库（列表）：「words = ["python", "turtle", "computer", "science", "program"]」
- 编写以下函数：
 1. 「choose_word(words)」：随机选一个单词，返回单词和对应的带空格的提示字符串（如 "p y t h o n" 每个字母间有空格）
 2. 「display_word(word, guessed)」：显示当前猜词进度，已猜对的字母显示，未猜对的显示 "_"
 3. 「hangman()」：主游戏函数
- 玩家有6次错误机会
- 每轮显示当前进度、已猜字母、剩余次数
- 玩家输入一个字母进行猜测
- 用 try/except 处理输入无效的情况
- 猜中所有字母或次数用完则游戏结束

请写出完整代码。

<details>

<summary>参考答案</summary>

题目1： 参数（形式参数）， 结束函数， 返回

题目2： 「math.sqrt」， 直接

题目3：

```
import math
```

```
def circle_area(radius):  
    return math.pi * radius ** 2
```

```
def distance(x1, y1, x2, y2):  
    return math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
```

```
def factorial(n):  
    if n == 0 or n == 1:  
        return 1  
    else:  
        return n * factorial(n - 1)
```

```
print(f"半径5的圆面积: {circle_area(5):.2f}")  
print(f"点(0,0)到(3,4)的距离: {distance(0, 0, 3, 4)}")  
print(f"5的阶乘: {factorial(5)}")
```

题目4：

```
import time
```

```
def timer(func):  
    def wrapper(*args, **kwargs):  
        start = time.time()  
        result = func(*args, **kwargs)  
        end = time.time()  
        print(f"{func.__name__} 执行了 {end - start:.4f} 秒")  
        return result  
    return wrapper
```

```
@timer  
def slow_sum(n):  
    total = 0  
    for i in range(n):  
        total += i  
    return total
```

```
print(f"结果: {slow_sum(1000000)}")
```

题目5：

```
import random
```

```
words = ["python", "turtle", "computer", "science", "program"]
```

```
def choose_word(words):
```

```
word = random.choice(words)
return word
```

```
def display_word(word, guessed):
    display = ""
    for letter in word:
        if letter in guessed:
            display += letter + " "
        else:
            display += "_ "
    return display.strip()
```

```
def hangman():
    word = choose_word(words)
    guessed = set()
    attempts_left = 6
    word_letters = set(word)
```

```
print("==== 猜单词游戏 =====")
print(f"单词有{len(word)}个字母，你有6次错误机会！")
```

```
while attempts_left > 0 and word_letters != guessed:
    print(f"\n当前进度: {display_word(word, guessed)}")
    print(f"已猜字母: {' '.join(sorted(guessed)) if guessed else '无'}")
    print(f"剩余机会: {attempts_left}")
```

```
guess = input("请输入一个字母: ").lower()
```

```
# 输入验证
if len(guess) != 1 or not guess.isalpha():
    print("请输入单个英文字母！")
    continue
```

```
if guess in guessed:
    print("这个字母已经猜过了！")
    continue
```

```
guessed.add(guess)
```

```
if guess in word_letters:
    print("猜对了！")
else:
    attempts_left -= 1
    print(f"猜错了！还剩{attempts_left}次机会")
```

```
# 游戏结果
print(f"\n答案: {word}")
if word_letters == guessed:
    print("恭喜你猜对了！")
else:
    print("机会用完了，下次加油！")
```

```
# 开始游戏
```

hangman()

</details>

自我评价: ☐ 全都会了 ☐ 大部分会了 ☐ 还要再练练