

G4-04 课后练习 —— 作用域与值引用

姓名： ____ 日期： ____

今天学了局部变量和全局变量——局部变量 = 房间里的东西，全局变量 = 客厅里的东西。

>

记忆钩子：

- "局部变量 = 房间里的东西，全局变量 = 客厅里的东西"
- "房间里定义的东西，只能在房间里用；客厅里的东西，房间里也能看到"
- "想改客厅的东西？用 global 喊一声！"

一、填空题（每空 2 分，共 20 分）

- 在函数内部用「=」定义的变量叫做 ____，在函数外部定义的变量叫做 ____。
- 局部变量在函数 ____ 时创建，函数 ____ 时销毁。
- 函数外无法访问函数内的局部变量，否则会报 ____ 错误。
- 在函数内部修改全局变量，需要在赋值前写上 ____ 声明。
- 在函数嵌套中， ____ 层函数只能在 ____ 层函数内部被调用。
- 「x = 10」写在所有函数之外，「x」是 ____ 变量；「y = 20」写在「def f():」里面，「y」是 ____ 变量。
- LEGB 规则中，L 代表 ____，G 代表 ____。

二、选择题（每题 3 分，共 24 分）

第1题 下面代码运行后，输出结果是什么？

```
def test():  
    a = 100  
test()  
print(a)
```

- A. 100
- B. None
- C. 报错 NameError
- D. 不确定

第2题 下面代码运行后，输出结果是什么？

```
x = 10
def show():
    print(x)
show()
```

- A. 10
- B. 报错 NameError
- C. None
- D. 不确定

第3题 下面代码运行后，输出结果是什么？

```
num = 5
def change():
    num = 10
change()
print(num)
```

- A. 5
- B. 10
- C. 报错
- D. 15

第4题 哪段代码能正确修改全局变量「score」？

A.

```
score = 60
def add():
    score = score + 10
```

B.

```
score = 60
def add():
    global score
    score += 10
```

C.

```
score = 60
def add():
    global score
    score = score + 10
```

D.

```
score = 60
def add():
    print(score)
```

第5题 关于函数嵌套，下列说法正确的是？

- A. 内层函数可以在任何地方被调用
- B. 内层函数只能在外层函数内部被调用
- C. 内层函数不能访问外层函数的变量
- D. 嵌套函数最多只能定义两层

第6题 下面代码运行后，输出结果是什么？

```
count = 0
def increment():
    global count
    count += 1
increment()
increment()
print(count)
```

- A. 0
- B. 1
- C. 2
- D. 报错

第7题 下面代码中，「msg」是什么变量？

```
msg = "你好"
def say():
    print(msg)
```

- A. 局部变量
- B. 全局变量
- C. 既是局部又是全局
- D. 不确定

第8题 下面代码的运行结果是什么？

```
def outer():
```

```
x = 1
def inner():
    print(x)
    inner()
outer()
```

- A. 1
 - B. 报错 NameError
 - C. None
 - D. 不确定
-

三、代码阅读题（每题 4 分，共 24 分）

第9题 写出下面代码的输出结果：

```
a = 10
def func1():
    a = 20
func1()
print(a)
```

输出： ____ ____ ____ ____

第10题 写出下面代码的输出结果：

```
b = 5
def func2():
    global b
    b = 15
func2()
print(b)
```

输出： ____ ____ ____ ____

第11题 写出下面代码的输出结果：

```
total = 0
def add_num(n):
    global total
    total += n
```

```
add_num(3)
add_num(7)
print(total)
```

输出： ____ ____ ____ ____

第12题 写出下面代码的输出结果：

```
def outer():
    print("A")
    def inner():
        print("B")
    print("C")
    inner()
```

```
outer()
```

输出：

- ① ____ ____ ____ ____
- ② ____ ____ ____ ____
- ③ ____ ____ ____ ____

第13题 写出下面代码的输出结果：

```
x = 100
def demo():
    x = 200
    print(x)
```

```
demo()
print(x)
```

输出：

- ① ____ ____ ____ ____
- ② ____ ____ ____ ____

第14题 写出下面代码的输出结果：

```
value = 1
def change():
    value = 2
    print("函数内:", value)
```

```
change()
print("函数外:", value)
```

输出：

- ① ____ ____ ____ ____
- ② ____ ____ ____ ____

四、编程题（共 32 分）

第15题：判断变量类型（8分）

阅读下面的代码，在横线上填写"局部变量"或"全局变量"。

```
level = 1      # ① _____
```

```
def update():  
    exp = 100   # ② _____  
    global level  
    level += 1  
    print(exp) # exp 是 ③ _____
```

```
update()  
# level 是 ④ _____
```

① _____

② _____

③ _____

④ _____

第16题：补全「global」声明（8分）

补全代码，使程序能正确累加总金额。

```
total = 0      # 全局变量：总金额
```

```
def add_money(amount):  
    # 请补全代码  
    # _____  
    # _____ += _____  
    print(f"已添加 {amount} 元")
```

```
add_money(50)  
add_money(30)  
print(total) # 期望输出：80
```

第17题：写嵌套函数画复杂图形（16分）

使用函数嵌套，用 turtle 画一个"小房子+太阳"的场景。

要求：

- 定义一个外层函数「draw_scene()」
- 在外层函数内部定义三个内层函数：
 - 「draw_sun()」—— 画太阳（黄色圆，半径 25）

- 「draw_house_body()」 —— 画房子主体（红色正方形，边长 60）
- 「draw_roof()」 —— 画屋顶（棕色三角形）
- 在外层函数中依次调用这三个内层函数

提示代码框架：

```
import turtle
t = turtle.Turtle()
t.speed(3)
t.pensize(2)
```

```
def draw_scene():
    """画场景：包含太阳、房子主体和屋顶"""
```

```
def draw_sun():
    """画太阳：黄色圆"""
    # 请补全代码
    pass
```

```
def draw_house_body():
    """画房子主体：红色正方形"""
    # 请补全代码
    pass
```

```
def draw_roof():
    """画屋顶：棕色三角形"""
    # 请补全代码
    pass
```

```
# 补全：按顺序调用三个内层函数
# _____
# _____
# _____
```

```
draw_scene()
turtle.done()
```

请写出完整代码：

五、挑战题（附加 10 分）

第18题：利用 global 实现游戏计分器

设计一个简单的游戏计分器，使用全局变量「score」记录分数。

要求：

- 定义函数「add_score(points)」：用「global」修改「score」，加上「points」分
- 定义函数「reset_score()」：用「global」将「score」重置为 0
- 定义函数「show_score()」：打印当前分数
- 模拟游戏过程：
 - 击败怪物 +10 分
 - 找到宝藏 +50 分
 - 掉入陷阱 -20 分
 - 显示最终分数
- 定义一个「game()」函数，用函数嵌套组织上面的步骤

score = 0

请在此处写代码

提交说明

- 提交方式：下节课前交给老师
- 编程题需在电脑上运行通过
- 选择题和填空题直接写在题目旁

教师批阅栏

题型	满分	得分
填空题（1-7）	20	
选择题（1-8）	24	
代码阅读（9-14）	24	
编程题（15-17）	32	
挑战题（18）	10	
总分	**110**	

评语： _____

补充题：浅拷贝与nonlocal

- 浅拷贝陷阱：「a = [1, 2, [3, 4]]; b = a[:]; b[2][0] = 99」 后 a 的值是 ____。
- 如何完全复制一个嵌套列表？用 ____ . ____ ()。
- 补全代码：

```
count = 0
def outer():
    ____ count # 用global还是nonlocal?
    count = 10
outer()
print(count) # 希望输出10
```

自我评价： ☐ 全都会了 ☐ 大部分会了 ☐ 还要再练练