

G2-17 课后练习 —— 函数参数入门

姓名: ____ 日期: ____

GESP 考点 #31 (参数化函数)、#29 (形参与实参)

姓名: ____ 日期: ____

一、选择题 (每题10分, 共40分)

1. 以下哪个是形参 (parameter) 的定义?

- A. 调用函数时传入的具体数值
- B. 定义函数时括号里的变量名
- C. 函数体里的变量
- D. 函数的返回值

2. 以下代码中, 实参是哪个?

```
def draw_polygon(sides, size):  
    angle = 360 / sides  
    for _ in range(sides):  
        t.forward(size)  
        t.right(angle)
```

```
draw_polygon(5, 80)
```

- A. 「sides」 和 「size」
- B. 「angle」
- C. 「5」 和 「80」
- D. 「t」

3. 以下代码运行后会画出什么?

```
def draw_polygon(sides, size):  
    angle = 360 / sides  
    for _ in range(sides):  
        t.forward(size)  
        t.right(angle)
```

```
draw_polygon(4, 100)
```

- A. 三角形 (边长100)
- B. 正方形 (边长100)
- C. 五边形 (边长100)
- D. 六边形 (边长100)

4. 参数化函数最大的好处是什么？

- A. 让代码看起来更复杂
 - B. 同一个函数传不同参数，得到不同结果
 - C. 不需要 import turtle
 - D. 可以少写几个函数名
-

二、填空题 (每题5分, 共30分)

1. 定义函数时写在括号里的变量名叫做 ____ 参数。

2. 调用函数时传入的具体数值叫做 ____ 参数。

3. 以下划线处填上适当的代码：

```
def draw_square(____): # 定义形参 size
    for _ in range(4):
        t.forward(____) # 用形参控制边长
        t.right(90)
```

4. 调用「draw_polygon(6, 70)」时，形参「sides」获得的值是 ____ ，形参「size」获得的值是 ____ 。

5. 以下代码画出的正多边形有 ____ 条边，每条边长 ____ 。

```
def draw_polygon(sides, size):
    angle = 360 / sides
    for _ in range(sides):
        t.forward(size)
        t.right(angle)
```

```
draw_polygon(8, 45)
```

6. 如果我想画一个三角形 (边长120) ，应该这样调用：「draw_polygon(____ , ____)」

三、写出输出 (每题5分, 共20分)

1.

```
def draw_square(size):  
    for _ in range(4):  
        t.forward(size)  
        t.right(90)
```

```
draw_square(120)
```

画出的图形： ____ ____ ____ ____ ____ （边长是多少？）

2.

```
def draw_polygon(sides, size):  
    angle = 360 / sides  
    for _ in range(sides):  
        t.forward(size)  
        t.right(angle)
```

```
draw_polygon(5, 60)
```

画出的图形： ____ ____ ____ ____ ____ （边数和边长各是多少？）

3.

```
def draw_line(length):  
    t.forward(length)
```

```
draw_line(50)  
draw_line(100)  
draw_line(150)
```

画出的是： ____ ____ ____ ____ ____ （描述三条线的特点）

4.

```
def draw_polygon(sides, size):  
    angle = 360 / sides  
    for _ in range(sides):  
        t.forward(size)  
        t.right(angle)
```

```
draw_polygon(10, 30)
```

画出的图形： ____ ____ ____ ____ ____ （边数？ 边长？ 每次转多少度？）

四、简答题（10分）

形参和实参有什么区别？请用自己的话解释，并举一个例子。

五、动手编程挑战（附加题，20分）

在电脑上运行以下代码，补全函数体和调用代码。

挑战①：画一排不同大小的正方形

```
import turtle
```

```
def draw_square(size):  
    """画一个边长为 size 的正方形"""  
    for _ in range(4):  
        t.forward(size)  
        t.right(90)
```

```
t = turtle.Turtle()  
t.speed(3)
```

```
# 任务：用循环画一排正方形，边长从 20 到 100，每次增加 20  
# 提示：for size in range(20, 101, 20):  
t.penup()  
t.goto(-200, 0)  
t.pendown()
```

```
# 请补全循环代码  
for size in range(20, 101, 20):  
    _____ # 画正方形  
    t.penup()  
    t.forward(_____) # 前进（边长+20）作为间距  
    t.pendown()
```

```
turtle.done()
```

预期结果：画出5个正方形，边长依次为 20、40、60、80、100，水平排列。

挑战②：画一个"多边形渐变"图案

```
import turtle
```

```
def draw_polygon(sides, size):  
    """画边数为 sides、边长为 size 的正多边形"""  
    angle = 360 / sides  
    for _ in range(sides):  
        t.forward(size)  
        t.right(angle)
```

```
t = turtle.Turtle()  
t.speed(3)
```

```
# 任务：画 3 边形到 10 边形，边长递减  
# 3边形边长100，4边形边长90，5边形边长80，.....10边形边长30  
# 提示：用 for 循环  
t.penup()  
t.goto(-200, 50)  
t.pendown()
```

```
# 请补全循环代码  
for sides in range(3, 11):      # 3 到 10 边形  
    size = _____          # 边长计算公式：130 - sides * 10  
    draw_polygon(sides, size)  
    t.penup()  
    t.forward(size + 20)        # 前进到下一个位置  
    t.pendown()
```

```
turtle.done()
```

预期结果： 画出8个正多边形（3~10边形），边长依次递减。

挑战③：自定义"画房子"函数

```
import turtle
```

```
def draw_square(size):  
    """画正方形（房子的主体）"""  
    for _ in range(4):  
        t.forward(size)  
        t.right(90)
```

```
def draw_triangle(size):  
    """画等边三角形（房子的屋顶）"""  
    for _ in range(3):  
        t.forward(size)
```

```
t.right(120)
```

```
def draw_house(size):  
    """画房子：下面正方形 + 上面三角形屋顶"""  
    # 先画正方形  
    draw_square(____) # 正方形边长 = size  
    # 移动到屋顶起点  
    t.penup()  
    t.goto(t.xcor(), t.ycor() + size) # 移到正方形顶部  
    t.pendown()  
    # 画三角形屋顶  
    draw_triangle(____) # 三角形边长 = size (和正方形一样)
```

```
t = turtle.Turtle()  
t.speed(3)
```

```
# 画三栋大小不同的房子  
t.penup()  
t.goto(-200, 0)  
t.pendown()  
draw_house(____) # 大房子 (size=80)
```

```
t.penup()  
t.goto(-50, 0)  
t.pendown()  
draw_house(____) # 中房子 (size=50)
```

```
t.penup()  
t.goto(70, 0)  
t.pendown()  
draw_house(____) # 小房子 (size=30)
```

```
turtle.done()
```

预期结果： 画出三栋房子，大、中、小，每栋由正方形+三角形组成。

完成情况表

请家长/学生勾选完成情况：

第1题	选择题		/	
第2题	填空题		/	
第3题	写出输出		/	
第4题	简答题		/	
挑战①	编程挑战		/ (选做)	
挑战②	编程挑战		/ (选做)	

挑战③	编程挑战		/ (选做)	
-----	------	--	--------	--

家长辅导小贴士

- 形参 vs 实参的核心区别
 - 形参 (parameter) = 定义函数时括号里的"占位符变量名"
 - 实参 (argument) = 调用函数时传入的"具体值"
 - 记忆钩子: 形参是"遥控器按钮", 实参是"按下的数字"
- 参数传递过程
 - 调用 [draw_square(100)] 时, Python 内部执行了一句隐式的赋值: [size = 100]
 - 然后进入函数体, [size] 就是 100
 - 所以 [t.forward(size)] 等价于 [t.forward(100)]
- 多个参数的一一对应
 - [def draw_polygon(sides, size):] 定义了两个形参
 - [draw_polygon(3, 100)] 调用时, 3sides, 100size
 - 顺序很重要! [draw_polygon(100, 3)] 会把 100 赋给 sides, 意思全反了
- 常见错误
 - 形参名写错: 定义时叫 [size], 调用时写 [draw_square(side=100)] 会报错
 - 忘记传参: [draw_square()] 没有传实参会报 [TypeError]
 - 参数个数不匹配: 定义了两个参数却只传一个, 也会报错
 - 函数体内没用形参: 定义了 [size] 但是函数体内用了固定值 [100]
- 检查代码的要点
 - 函数定义: [def 函数名(形参名):] —— 有没有冒号?
 - 函数体内: 有没有用到形参? (而不是硬编码固定值)
 - 函数调用: 传了几个实参? 顺序对不对?

自我评价: ☐ 全都会了 ☐ 大部分会了 ☐ 还要再练练