

多彩图形争霸赛 (二)





1. 探索新知

1.1

函数的参数

假设你开了一家深夜食堂，由于前期人手不足，所以每位顾客只能点一碟小菜和一份主食。如果用无参函数表示，如下：

第一位顾客：

```
def menu1():  
    print('一份小菜： 酸辣海带')  
    print('一份主食： 兰州拉面')  
  
menu1()
```

第二位顾客：

```
def menu2():  
    print('一份小菜： 老味花生')  
    print('一份主食： 三鲜水饺')  
  
menu2()
```

可以发现两位顾客点单的函数，只有小菜和主食的菜名在变化



是否能够升级一下我们的函数，
达到以一抵百的效果呢~

1.1

函数的参数

今天我们要学的叫做有参函数，由三个部分组成：函数名、参数、函数体

定义函数的语法

```
def 函数名(参数):  
    函数体
```

```
def menu():  
    print('一份小菜: 酸辣海带')  
    print('一份主食: 兰州拉面')
```

```
def menu(a, b):  
    print('一份小菜: ' + a)  
    print('一份主食: ' + b)
```



以一抵百的函数已经升级完成，刚好有位顾客点了小菜“凉拌三丝”和主食“蛋炒饭”，该怎么调用函数呢？



我们怎么把“凉拌三丝”传递到参数 a 中，“蛋炒饭”传递到参数 b 中？

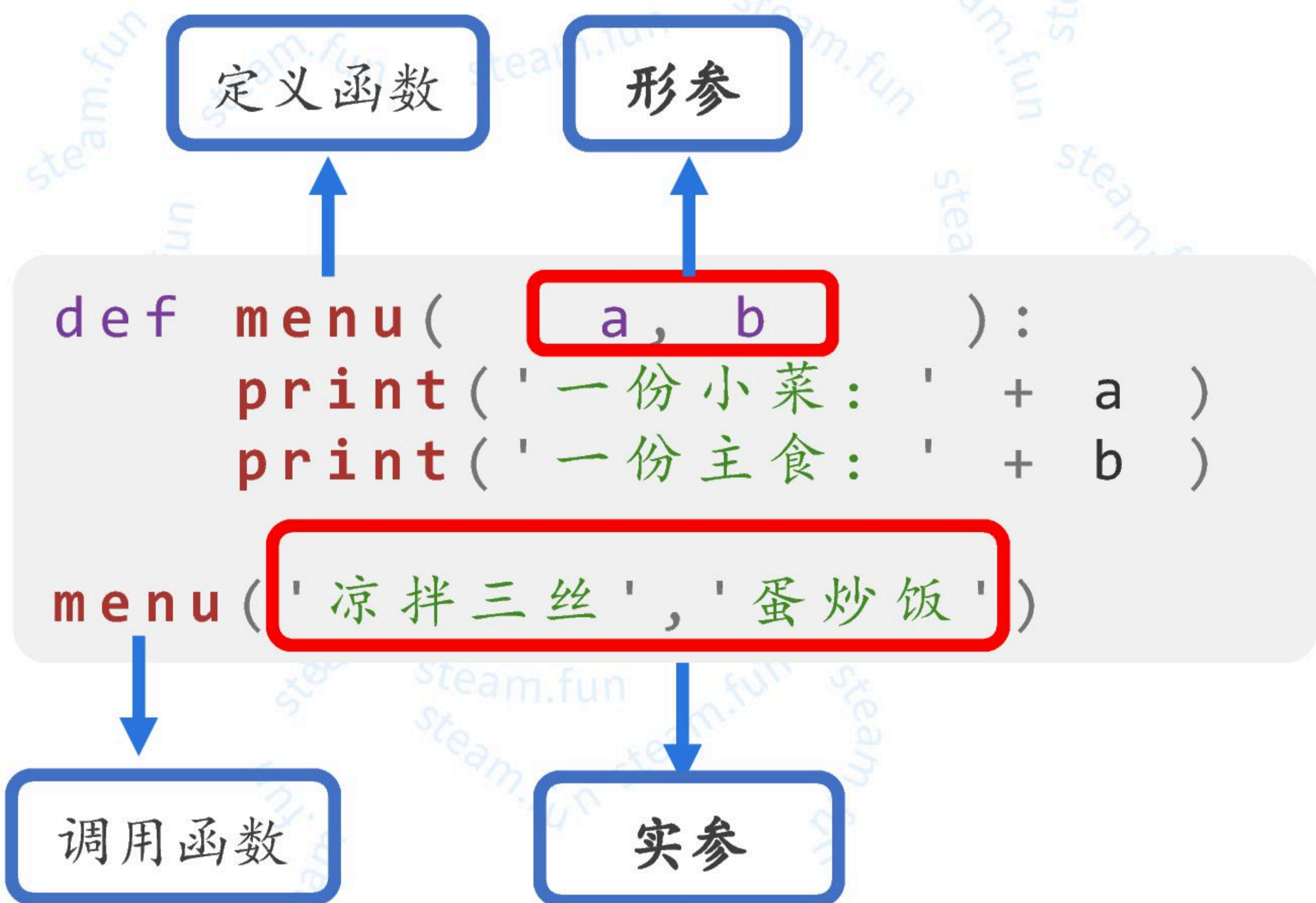
函数中有一条神奇的通道可以传送数据，
实参传递到形参，函数体内再对形参进行处理

1.1

函数的参数



以刚才的menu()函数为例，给大家介绍下形参和实参



形参：形式上的参数，没有实际的值，可以想象成占位用的，也可以想象成变量，调用函数时写入实参，也就相当于给变量赋值

实参：实际意义上的参数，实际存在的参数，可以想象成要赋给变量的值，实参的值单向传递给形参

➤ 无参函数和有参函数对比

```
# 定义无参函数say_hello()  
def say_hello():  
    print("Hello!")  
# 调用无参函数  
say_hello()
```

输出：_____

1.1

函数的参数

```
# 定义有参函数greet
def greet(name,say_hello):
    print("Hello",name,say_hello)
# 调用有参函数
greet('小明','很高兴见到你')
```

输出：_____

无参函数：没有参数，可以直接执行函数内特定的操作

有参函数：选择“喂”给函数一个或几个数据（参数），它就能内部消化，给你“吐”出你想要的东西，多个参数间用逗号分隔

1.2

return返回值

return:
返回

return返回值：当咱们给函数传入参数，函数就会返回一个值给你。每个函数都有返回值。

```
def shifoujige(score):
    if score < 60:
        return '不及格!'
    else:
        return '及格!' # 指定返回值
word = shifoujige(90) # 调用函数，传入参数90，返回“及格”
print(word) # 打印函数的返回值
```



通过几个例子再来感受一下return返回值，猜一猜并且自己编写代码运行试试吧

1.2

return返回值

add:
增加

```
def my_add(a,b):
    c = a + b
    return c
print(my_add(1,2))      # 输出 _____
```

```
def my_add(a,b):
    return a + b
print(my_add(1,2))     # 输出 _____
```

```
def my_fun(a,b):
    c = a + b
    d = a - b
    return c,d
print(my_fun(5,4))    # 输出 _____
```

结论：返回值可以是单个值，也可以是多个值，用逗号隔开

```
def my_add(a,b):
    c = a + b
    return c
    print('计算完毕')
print(my_add(1,2))    # 输出 _____
```

会打印“计算完毕”吗？

return语句的作用：

1. 结束函数调用，停止执行
2. 返回指定的返回值

1.2

return返回值



return返回值也不难嘛！但是我有个疑问，我见过好多函数中没写return语句，他们的返回值是什么呢？自己试试吧！

```
def say_hello():  
    print("Hello!")  
word = say_hello()  
print(word)
```

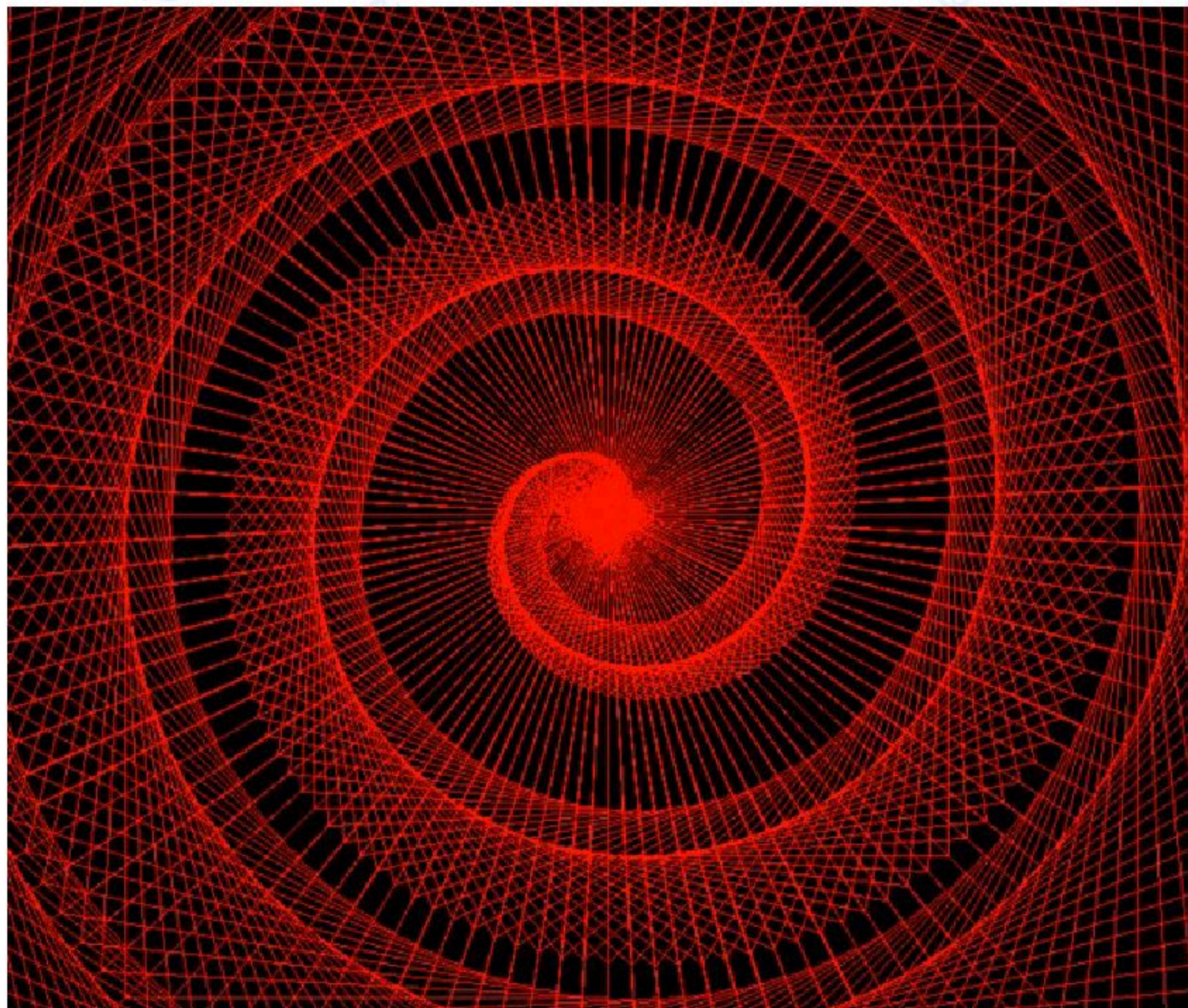
输出：_____

结论：没有return语句的函数会默认返回None

1.3

视觉漩涡

你能看出哪些有效信息呢？



有效信息：

1. 定义画正方形的函数



画正方形的函数难不倒我们，那怎么定义画边长变化的正方形的函数呢？

length:
长度

```
def side4(length):
    for i in range(4):
        t.forward(length)
        t.left(90)
```

不要忘记导入 turtle 库

2. 初始化设置

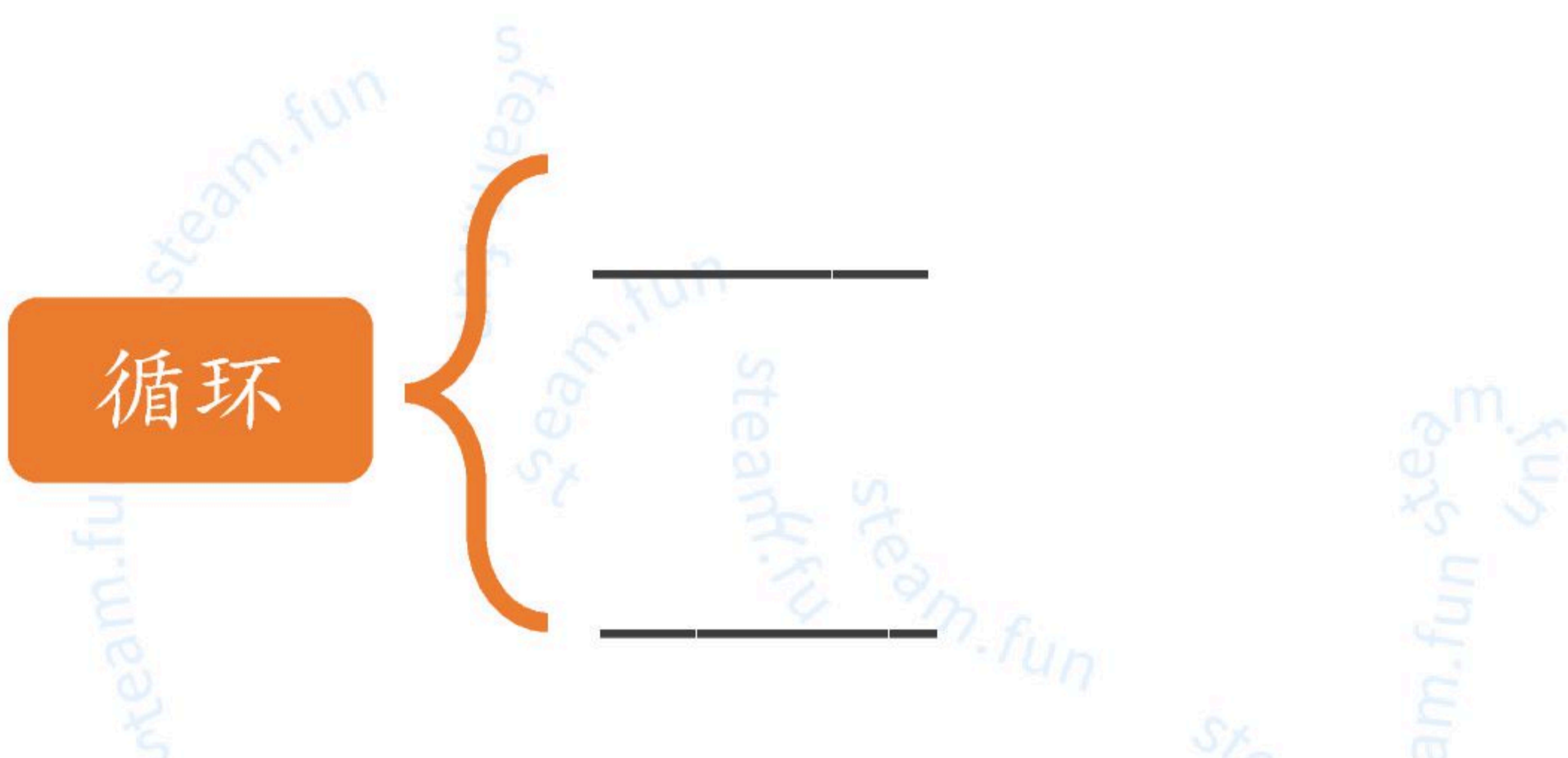
```
_____ # 初始化画笔颜色
_____ # 初始化背景尺寸、颜色
_____ # 初始化第一个正方形的边长为5
```

3. 启动漩涡



漩涡是由多个边长不等的正方形组成的，如何绘制多个正方形？

咱们学过哪些实现循环方法？有什么区别？



以for循环为例讲解，现在来一步步分析绘制过程吧！



自由尝试、选择：

1. 向右转可以吗？
2. 旋转角度多少合适？
3. 边长增量多少合适？
4. 循环多少次合适？

4. 停止画笔绘制，绘图窗口不关闭

```
t.done()
```

如果想要快些看到绘制的图案，该如何让绘图加速呢？自己试试吧！

完整代码

```
import turtle as t

def side4(length):
    for i in range(4):
        t.forward(length)
        t.left(90)

t.speed(0)
t.pencolor('red') # 初始化画笔颜色
t.screensize(400,300,'black') # 初始化背景尺寸、颜色
length = 5 # 初始化第一个正方形的边长为5
for i in range(300):
    side4(length)
    t.left(3)
    length += 1

t.done()
```



2. 强化练习

1. 下列代码输出的结果是？（ ）

A. 3

B. 6

C. 0

D. 9

```
def py(b):  
    b+=3  
    return(b)  
print(py(3))
```

2. 通常，定义一个函数，最多包含多少个参数？（ ）

A. 3

B. 4

C. 5

D. 任意多

3. 关于函数，正确的描述是？（ ）

A. 规范的函数设计都应该带有返回值

B. 函数体中，只要遇到 break 关键字，函数就结束

C. 函数体中，只要遇到 return 关键字，函数就结束

D. 函数的返回值只能是整型、浮点型、布尔型和字符串

4. 以上程序运行的结果是？（ ）

```
def ThankYou(name):  
    print("感谢%s对公司的贡献！" %name)  
ThankYou('张三')  
ThankYou()
```



2. 强化练习

A. 打印输出：感谢张三对公司的贡献！

然后报错：TypeError: ThankYou() missing 1 required positional argument: 'name'

B. 打印输出：

感谢张三对公司的贡献！

感谢对公司的贡献！

C. 无法输出

D. 需要增加main函数才能正确运行

5. 执行如下 Python 代码，输出结果是？（ ）

A. 11

B. 10

C. 1

D. 运行错误

```
def hs(num):  
    num += 1  
    return num  
  
n = 10  
s = hs(n)  
print(s)
```

3. 术语箱

return	返回	level	等级
none	没有	word	单词; 话语
add	增加	score	分数
greet	欢迎		

4. 课后挑战

加法计算器

制作一个加法计算器。将加数1、加数2作为参数，返回计算结果。

编写一段代码，要求如下：

- (1) 定义有参函数
- (2) 返回计算结果
- (3) 输出函数的返回值

