

CH5 Python函数

- 01 函数介绍

- 定义：组织好的，可重复使用的，用来实现特定功能的代码段。

- 演示

```
"""
演示：快速体验函数的开发与应用
"""

# 需求：统计字符串的长度，使用内置函数len()
str1 = "itheima"
str2 = "itcast"
str3 = "python"

# 定义一个计数变量
count = 0
for i in str1:
    count += 1
print(f"字符串{str1}的长度是{count}")

count = 0
for i in str2:
    count += 1
print(f"字符串{str2}的长度是{count}")

count = 0
for i in str3:
    count += 1
print(f"字符串{str3}的长度是{count}")
```

```
# 可以使用函数，来优化这个过程
def my_len(data): 3用法
    count = 0
    for i in data:
        count += 1
    print(f"字符串{data}的长度是{count}")

my_len(str1)
my_len(str2)
my_len(str3)
```

字符串itheima的长度是7
字符串itcast的长度是6
字符串python的长度是6
字符串itheima的长度是7
字符串itcast的长度是6
字符串python的长度是6

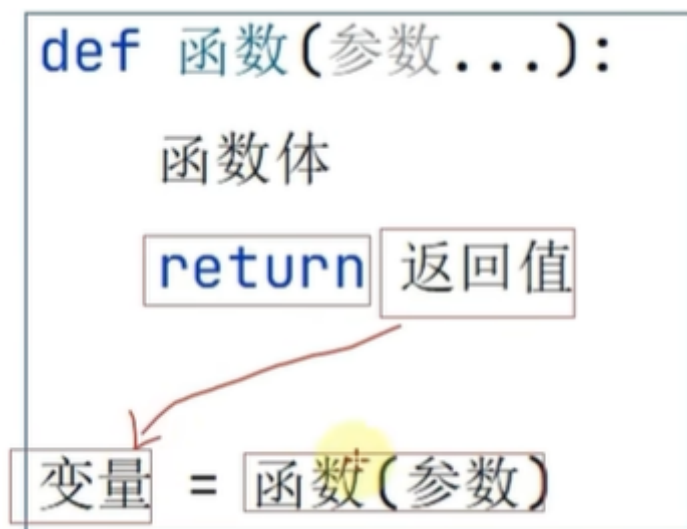
- 好处：
 - 1) 将功能封装在函数内，可供随时随地重复利用
 - 2) 提高代码的复用性，减少重复代码

• 02 函数的定义

• 03 函数的参数

• 04 函数的返回值

- 定义：程序中函数完成事情后，最后给调用者的结果
- 语法格式：如图，变量就能接收到函数的返回值



- 注意：函数体在遇到return后就结束了，所以写在return后的代码不会执行
- None类型
 - 是什么：Python中有一个特殊的字面量：None，其类型是：<class'NoneType'>，用于表示：空的、无意义的
 - 函数如何返回None：
 - 不使用return语句即返回None
 - 主动return None
 - 应用场景：
 - 用在函数无返回值上
 - 用在if判断上：

- 在if判断中，None等同于False
- 一般用于在函数中主动返回None，配合if判断做相关处理
- 用于声明无内容的变量上
 - 定义变量，但暂时不需要变量有具体值，可以用None来代替

```
# 暂不赋予变量具体值
name = None
```

• 05 函数说明文档

- 语法：

```
def func(x, y):
    """
    函数说明
    :param x: 形参x的说明
    :param y: 形参y的说明
    :return: 返回值的说明
    """
    函数体
    return 返回值
```

- 通过多行注释的格式，对函数进行说明解释
- 内容应写在函数体之前

• 06 函数的嵌套调用

- 定义：一个函数里面又调用了另外一个函数

```
def func_b():
    print("---2---")

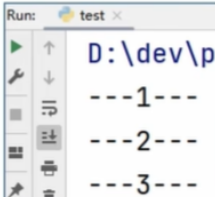
def func_a():
    print("---1---")

    func_b()

    print("---3---")

# 调用函数func_a
func_a()
```

执行效果：



```
Run: test x
D:\dev\p
---1---
---2---
---3---
```

- 注意：函数A中执行到调用函数B的语句，会将函数B全部执行完成后，才继续执行函数A的剩余内容