

Day11-函数的定义与使用

一、学习目标

- 理解什么是函数及其作用
- 掌握函数的定义语法：def 关键字、参数、返回值
- 学会调用函数并理解变量作用域
- 能够定义简单函数完成计算任务（如两数求和）
- 了解函数编程的基本思想：封装、复用、可读性

二、知识点讲解

1. 什么是函数

在编程中（函数就像是一个“小工厂”或“工具箱里的工具”，你给它一些原料（输入），它经过处理后返回一个结果（输出）。

为什么需要函数呢？

- 避免重复写代码（代码复用）
- 让程序结构更加清晰（模块化）
- 提高代码的可读性和维护性

例子：

```
# 不使用函数

# 第一次计算
a1, b1 = 3, 5
result1 = a1 + b1
print(f"{a1} + {b1} = {result1}")#3 + 5 = 8

# 第二次计算
a2, b2 = 10, 20
```

```

result2 = a2 + b2

print(f"{a2} + {b2} = {result2}")#10 + 20 = 30

# 第三次计算

a3, b3 = 7, 8

result3 = a3 + b3

print(f"{a3} + {b3} = {result3}")#7 + 8 = 15

```

使用函数

```

# 使用函数（定义一次，调用多次）

def add(a, b):

    """计算两个数的和并返回结果"""

    return a + b

# 调用函数多次

print(f"3 + 5 = {add(3, 5)}")    #3 + 5 = 8

print(f"10 + 20 = {add(10, 20)}")#print(f"10 + 20 = {add(10, 20)}")

print(f"7 + 8 = {add(7, 8)}")    #print(f"7 + 8 = {add(7, 8)}")

```

2. 函数的定义：def 关键字

在 python 中，使用 def 关键字来定义一个函数

- 函数代码块以 def 关键词开头，后接函数标识符名称和圆括号()。
- 任何传入参数和自变量必须放在圆括号中间。圆括号之间可以用于定义参数。
- 函数的第一行语句可以选择性地使用文档字符串—用于存放函数说明。
- 函数内容以冒号起始，并且 Tab 缩进。
- return [表达式] 结束函数，选择性地返回一个值给调用方。不带表达式的 return 相当于返回 None。

语法格式：

```

def 函数名(参数 1, 参数 2, ...):

    # 函数体：要执行的操作

    结果 = 参数 1 + 参数 2

    return 结果

```

| | |
|--------|------------------------------------|
| def | 定义函数的关键字，必须小写 |
| 函数名 | 遵循变量命名规则，建议有意义，如 add,calculate_sum |
| 参数列表 | 放在括号内，多个参数用逗号分隔，可以为空 |
| : | 函数头结束必须加冒号 |
| 函数体 | 缩进的代码块，表示函数要执行的操作 |
| return | 可选，用于返回结果。没有 return 默认返回 None |

示例：

```
def hello() :
    print("Hello World!")
hello() #Hello World!
```

3.函数的调用

定义完函数之后，我们需要“调用”它，才能执行。

#示例2，定义一个函数，计算两个数字的和

定义函数

```
def add_numbers(a, b):#add_numbers 是函数名
    result = a + b    #a 和 b 是形参（形式参数），代表输入的两个输
    return result     #return result 是表示吧计算结果返回去是返回值
```

调用函数

```
sum1 = add_numbers(3, 5)#3 和 5 是实参（实际参数），传给 a 和 b
sum2 = add_numbers(10, 20)

print(sum1) # 输出：8
print(sum2) # 输出：30
```

4. 函数的返回值

`return` 是函数的“出口”，用来把结果传递回调用处。

| 情况 | 示例 | 返回值 |
|-------------------------|---------------------------|------------------------|
| 有 <code>return</code> 值 | <code>return a + b</code> | 返回具体值 |
| 只有 <code>return</code> | <code>return</code> | 返回 <code>None</code> |
| 没有 <code>return</code> | 函数结束无返回 | 默认返回 <code>None</code> |

```
def func1():
    return 100

def func2():
    print("Hello")

print(func1()) # 输出: 100
print(func2()) # 输出: Hello \n None
```

5.变量的作用域 (Scope)

变量不是在哪儿都能用的！Python 有作用域的概念。

局部变量 (Local)：定义在函数内部的变量，只能在函数内访问

全局变量 (Global)：定义在函数外部的变量，所有函数都可以读取（但修改需用 `global` 关键字）

```
x = 10 # 全局变量

def my_function():
    y = 5 # 局部变量
    print(x + y) # 可以访问全局变量 x

my_function() # 输出: 15

# print(y) # 错误! y 是局部变量，外部无法访问
```

三、作业安排

定义一个函数，计算两个数字的和，并返回结果。（使用 ipynb 格式的文件）