

Lesson4 函数

一. 函数概念和基本语法

函数是组织在一起以执行特定任务的代码段。函数可以提高代码的重用性和可读性。

函数在数学中的概念： $f(x) = y = ax + b$

$$1 + x = 3$$

$$y + x = 3$$

$$f(x) = 3 - x$$

```
In [2]: # 函数基本语法
def function_name(parameters):
    # function body
    return value # 可选
```

```
In [1]: # 简单的函数定义
# 定义一个简单的函数
def greet():
    print("Hello, world!")

# 调用函数
greet() # 输出: Hello, world!
```

Hello, world!

```
In [2]: # 带有参数的函数定义
# 定义带参数的函数
def greet(name):
    print(f"Hello, {name}!")

# 调用函数
greet("Jesse") # 输出: Hello, name!
```

Hello, Jesse!

二. 参数

--实参和形参

形参（形式参数）：在函数定义中出现的参数，用于接收函数调用时传入的值。形参是占位符，代表传入函数的实际数据。

实参（实际参数）：在函数调用时传递给函数的实际值。实参用于替换函数定义中的形参。

```
In [4]: # 定义函数时使用形参
def add(a, b):
    return a + b # a 和 b 是形参，在定义 add 函数时使用。

# 调用函数时使用实参
```

```
result = add(3, 5) # 3 和 5 是实参，在调用 add 函数时传递给形参 a 和 b。
print(result)
```

8

--位置参数

按位置顺序传递的参数

```
In [7]: # 定义一个打印学生分数的函数
def print_scores(student1, student2, student3):
    print(f"Student 1 Score: {student1}")
    print(f"Student 2 Score: {student2}")
    print(f"Student 3 Score: {student3}")

# 使用位置参数调用函数
print_scores(95, 80, 78)
```

```
Student 1 Score: 95
Student 2 Score: 80
Student 3 Score: 78
```

--关键词参数

关键字参数通过参数名传递，可以使函数调用更清晰。

```
In [9]: # 定义一个函数，打印名字和年龄
def print_info(name, age):
    print(f"Name: {name}, Age: {age}")

# 使用关键字参数调用函数
print_info(name="Jesse", age=13)
print_info(age=14, name="Andrew")
```

```
Name: Jesse, Age: 13
Name: Andrew, Age: 14
```

--默认参数

在函数定义中为形参提供默认值，如果函数调用时未传递该参数，则使用默认值。

```
In [11]: # 定义带默认参数的函数
def greet(name="Guest"):
    print(f"Hello, {name}!")

# 调用函数时可以使用或不使用实参
greet()
greet("Jesse")
```

```
Hello, Guest!
Hello, Jesse!
```

```
In [ ]: # 定义带两个形参的函数
def describe_pet(pet_name, animal_type="dog"):
    print(f"I have a {animal_type} named {pet_name}.")

# 调用函数时使用关键字参数
describe_pet(pet_name="Rex", animal_type="cat") # 输出: I have a cat named Budd
describe_pet(animal_type="hamster", pet_name="Nibbles") # 输出: I have a hamste
describe_pet(pet_name="Rex") # 输出: I have a dog named Rex.
```

I have a cat named Rex.
I have a hamster named Nibbles.
I have a dog named Rex.

三. 返回值

函数可以返回值，使用 return 语句。

```
In [12]: # 示例
# 定义一个返回两个数中较大值的函数
def get_max(a, b):
    if a > b:
        return a
    else:
        return b

# 调用函数并接收返回值
max_value = get_max(10, 20)
print("Max value:", max_value) # 输出: Max value: 20
```

Max value: 20

```
In [13]: # 定义一个函数，返回两个数的和与差
def add_subtract(a, b):
    return a + b, a - b

# 调用函数并接收返回值
sum_value, diff_value = add_subtract(10, 5)
print("Sum:", sum_value) # 输出: Sum: 15
print("Difference:", diff_value) # 输出: Difference: 5
```

Sum: 15

Difference: 5

四. 综合举例

```
In [1]: # 综合运用函数
# 输入两个数字，我要分别得到他的加减乘除的结果
# 获取用户输入的函数
def get_input():
    a = float(input("Enter the first number: "))
    b = float(input("Enter the second number: "))
    return a, b

# 计算和的函数
def add(a, b):
    return a + b

# 计算差的函数
def subtract(a, b):
    return a - b

# 计算积的函数
def multiply(a, b):
    return a * b

# 计算商的函数
def divide(a, b):
```

```

    if b != 0:
        return a / b
    else:
        return "Error! Division by zero."

# 主程序
def main():
    a, b = get_input()
    print(f"The sum of {a} and {b} is: {add(a, b)}")
    print(f"The difference between {a} and {b} is: {subtract(a, b)}")
    print(f"The product of {a} and {b} is: {multiply(a, b)}")
    print(f"The quotient of {a} and {b} is: {divide(a, b)}")

# 调用主程序
main()

```

```

In [ ]: # 定义一个函数，用于记录和打印学生的信息
def student_info(name, age=18, *courses, **additional_info):
    print(f"Name: {name}")
    print(f"Age: {age}")

    # 打印课程信息
    if courses:
        print("Courses:", end=" ")
        for course in courses:
            print(course, end=", ")
        print()

    # 打印其他额外信息
    for key, value in additional_info.items():
        print(f"{key}: {value}")

# 调用函数，传入不同类型的参数

# 仅传入位置参数和默认参数
student_info("Alice")
# 输出:
# Name: Alice
# Age: 18

# 传入位置参数、默认参数和不定长位置参数
student_info("Bob", 20, "Math", "Science", "History")
# 输出:
# Name: Bob
# Age: 20
# Courses: Math, Science, History,

# 传入位置参数、默认参数、不定长位置参数和不定长关键字参数
student_info("Charlie", 22, "Art", "Music", hobby="painting", city="New York")
# 输出:
# Name: Charlie
# Age: 22
# Courses: Art, Music,
# hobby: painting
# city: New York

# 位置参数: name, 例如 "Alice"、"Bob"、"Charlie"。
# 默认参数: age, 默认为 18, 可以在调用时指定不同的值, 例如 20、22。

```

```
# 不定长位置参数: *courses, 接收任意数量的课程名称, 例如 "Math"、"Science"、"Hist  
# 不定长关键字参数: **additional_info, 接收任意数量的额外信息, 例如 hobby="painti
```

作者: Yming

邮箱: yuemingn@student.unimelb.edu.au

版权声明: 本代码仅用于个人学习目的, 未经许可, 不得用于商业用途。