



函数的调用和返回

礼 欣

北京理工大学



6.3 函数调用

- 函数调用执行的四个步骤：
 - 调用程序在调用处暂停执行
 - 函数的形参在调用时被赋值为实参
 - 执行函数体
 - 函数被调用结束，给出返回值



6.3 函数调用

- ```
sing("Mike")
print()
sing("Lily")
```

- ```
def main():
    sing("Mike")
    print()
    sing("Lily")

def sing(person):
    happy()
    happy()
    print("Happy birthday, dear", person + ".")
    happy()
```



6.3 函数调用

■ happy()完整调用过程图：

```
def main():  
    sing("Mike")  
    print()  
    sing("Lily")  
    Person:Mike  
def sing(person):  
    happy()  
    happy()  
    print("Happy birthday, dear", person + ".")  
    happy()  
def happy():  
    print("happy Birthday to you")
```

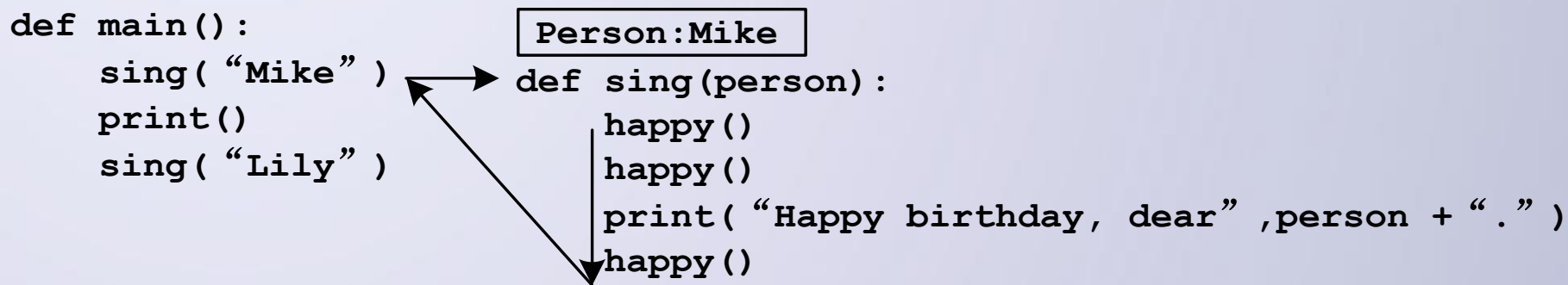
Diagram illustrating the function call process for `happy()`:

- The `main()` function calls `sing("Mike")`.
- The `sing(person)` function calls `happy()`.
- The `happy()` function calls `print("happy Birthday to you")`.
- The `happy()` function calls `print("Happy birthday, dear", person + ".")`.
- The `happy()` function calls `happy()` again.



6.3 函数调用

■ sing()完整调用过程图：





6.4 函数的返回值

- `return`语句：程序退出该函数，并返回到函数被调用的地方
- `return`语句返回的值传递给调用程序
- Python函数的返回值有两种形式：



- 返回一个值
- 返回多个值



6.4 函数的返回值

- 无返回值的return语句等价于return None。
- None是表示没有任何东西的特殊类型。

```
def happy():  
    print("Happy birthday to you!")
```

- 等价于：

```
def happy():  
    print("Happy birthday to you!")  
    return None
```





6.4 函数的返回值

- 返回值可以是一个变量，也可以是一个表达式。

```
def square(x):  
    y = x * x  
    return y
```

- 等价于：

```
def square1(x):  
    return x * x
```





6.4 函数的返回值

- square()函数调用举例：

```
>>> square(2)
4
>>> print(square(5))
25
```



6.4 函数的返回值

- 调用square()函数实例：

```
>>> x = 3
>>> y = square(x)
>>> print(y)
9
>>> print(square(x) + square(4))
25
```



6.4 函数的返回值

- 例：应用square()函数编写程序以计算两点之间的距离。
- 原理：给定两点坐标（ x1,y1 ）和(x2,y2) ， 根据勾股定理， 两点间距离公式为



$$\sqrt{(x1-x2)^2 + (y1-y2)^2}$$



6.4 函数的返回值

- 计算两点距离的函数代码：

```
def distance(x1, y1, x2, y2):  
    dist = math.sqrt(square(x1 - x2)  
                      + square(y1 - y2))  
    return dist
```



6.4 函数的返回值

- 例子：应用distance()编写程序计算三角形周长
- 代码：

```
# triangle.py
import math

def square(x):
    return x * x

def distance(x1, y1, x2, y2):
    dist = math.sqrt(square(x1 - x2) + square(y1 - y2))
    return dist

def isTriangle(x1, y1, x2, y2, x3, y3):
    flag = ((x1 - x2) * (y3 - y2) - (x3 - x2) * (y1 - y2)) != 0
    return flag
```



6.4 函数的返回值

```
def main():  
    print("Please enter (x,y) of three points in turn: ")  
    # 获取用户输入的三个坐标点  
    x1, y1 = eval(input("Point1: (x, y) = "))  
    x2, y2 = eval(input("Point2: (x, y) = "))  
    x3, y3 = eval(input("Point3: (x, y) = "))  
    # 判断三个点是否构成三角形  
    if(isTriangle(x1, y1, x2, y2, x3, y3)):  
        # 计算三角形周长  
        perim = distance(x1, y1, x2, y2)  
                + distance(x2, y2, x3, y3)  
                + distance(x1, y1, x3, y3)  
        print("The perimeter of the triangle is: {0:0.2f}".format(perim))  
    else:  
        print("kidding me? This is not a triangle!")
```



6.4 函数的返回值

■ 运行结果：

```
>>> main()
Please enter (x,y) of three points in turn:
Point1: (x, y) = 1, 1
Point2: (x, y) = 2, 2
Point3: (x, y) = 3, 3
kidding me? This is not a triangle!
```



6.4 函数的返回值

■ 运行结果：

```
>>> main()  
Please enter (x,y) of three points in turn:  
Point1: (x, y) = 0, 0  
Point2: (x, y) = 1, 0  
Point3: (x, y) = 0, 1  
The perimeter of the triangle is: 3.41
```




6.4 函数的返回值

- 程序同一行语句中`distance()`被调用了三次，用来计算三角形的周长
- 使用函数解决了代码的复用问题



6.4 函数的返回值

- 使用return语句返回多个值
- 举例：计算两个数的加法和减法
- 计算函数：

```
def sumDiff(x,y):  
    sum = x + y  
    diff = x - y  
    return sum, diff
```

```
num1, num2 = eval(input("Please enter two numbers (num1, num2)"))  
s, d = sumDiff(num1, num2)  
print("The sum is", s, "and the difference is", d)
```



6.4 函数的返回值

- 对于多返回值的函数，根据变量的位置来赋值
 - s将获得return的第一个返回值sum
 - d将获得第二个返回值diff