第八课: Python 自定义函数

魏守科 博士, 教授

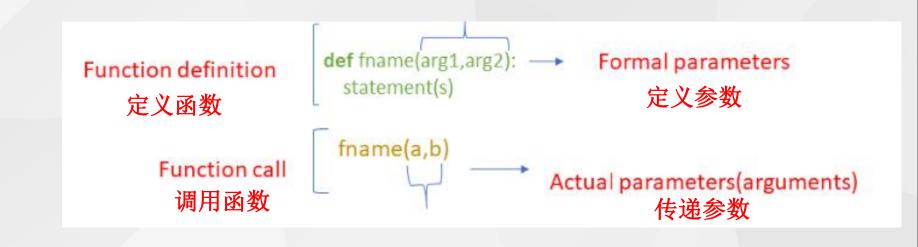
目录页 CONTENTS PAGE

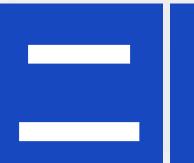
- 定义函数语法
- 定义函数
- 函数变量
- 四 匿名函数

定义函数语法

• 创建函数规则

定义函数语法





定义函数

- 定义函数
- 定义参数
- 返回值

定义函数

```
# 无参数
def welcome():
    print("欢迎参加我的Python课!")
```

调用函数

welcome()

```
#1个参数
def welcome(name):
   print(f"{name}, 欢迎参见我的Python课!")
# 调用函数
name = "老李"
welcome(name)
```

```
# 多个参数
def sum_caculator(x,y,z):
  sum = x+y+z
  print("计算的和:", sum)
#调用函数
sum caculator(8,22,38)
```

不定参数

```
# 任意参数,或可变参数

def sum_caculator(x,y,z):
    sum = x+y+z
    print("计算的和:", sum)
```

调用函数 sum caculator(8,22,38,50) #错误

任意参数:

*args 和 **kwargs

```
def sum_caculator(*args):
  sum = 0
  for n in args:
    sum + = n
  print("计算的和:",sum)
# 调用函数
sum caculator(8,22)
sum caculator(8,22,38,50)
```

```
def info(**kwargs):
  print("Data type of argument:", type(kwargs))
  for key, value in kwargs.items():
    print(f"{key}价格为{value}.")
#调用函数
info("苹果"=5.2)
info("苹果"=5.2, "香蕉"=2.2)
```

```
# 默认参数

def sum_caculator(x=8,y,z):
    sum = x+y+z
    print("计算的和:", sum)
```

调用函数 sum_caculator(22,38)

错误X

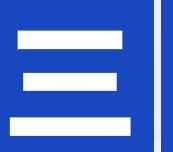
```
# 默认参数

def sum_caculator(y,z,x=8):
    sum = x+y+z
    print("计算的和:", sum)
```

调用函数 sum_caculator(22,38)

返回值

```
#返回值
def sum_caculator(x=8,y,z):
  sum = x+y+z
  print("计算的和:", sum)
  return sum*y
# 调用函数
sum caculator(22,38)
```



函数变量

- 局部变量
- 全局变量

函数变量

- 局部变量 (local variables): 所有在函数内部定义的 变量
- 全局变量 (global variables): 所有在函数外部定义的变量

```
#局部变量
                                 q = 10
def subtractor():
  q = 10
                                 def subtractor():
                                    global q #调用,已与局部变量区分
  p = 5
  print(q-p)
                                    p = 5
                                    print(q-p)
subtractor()
print(q)
                                 subtractor()
                                 print(q)
```



匿名函数

• 匿名函数

匿名函数

lambda x: x * x

关键字lambda表示匿名函数,冒号前面的x表示函数参数。

匿名函数有个限制,就是只能有一个表达式,不用写return,返回值就是该表达式的结果。

应用实例

匿名函数的好处: (1) 函数没有名字,不必担心函数名冲突; (2) 可以把匿名函数赋值给一个变量,再利用变量来调用该函数:

```
f = lambda x,y,z: x+y+z
f
f(10,15,30)
```

这个匿名函数等价于:

```
def f(x,y,z):
return x+y+z
```

总结



具体讲解了如何定义一个函数,包括定义参数(无参数、一个、多个、不定长、以及默认参数)和返回值

