第6讲 函数1

王肖燕

内容回忆

一．组合数据类型

- ✓ 字符串(str)：“abcdef”
- ✓ 元组(tuple): (1, 2, 3), (1, 2, 3, ’a’)
- ✓ 列表(list): [‘Li’, ’Wang’, ’Zhang’]
- ✓ 集合(set): {1, 2, 3, 4}
- ✓ 字典(dict): {‘a’:’China’, ‘b’:’Japan’, ‘c’:’USA’}

二．for循环结构

- ✓ for语句遍历上述组合结构
- ✓ else扩展模式

内容介绍

一．函数定义和调用

二．参数传递

三．变量作用域

四．函数内定义函数

五．lambda函数

六．Range函数

函数

- 数学中的函数
  - 从自变量到因变量之间的一种映射或对应关系
- 软件开发中的函数
  - 相似的含义
  - 把输入经过一定的变换和处理后得到预定的输出

编程中遇到的问题

- 操作完全相同，只是处理数据不同
  - 相同的代码被复制到多处
  - 缺点：
    - 增加了代码量
    - 增加了维护的难度
- 有效解决方法
  - 设计函数和类

基本语法

- def 函数名(n):
  函数体
- 解释：Python中使用def关键字来定义函数，然后是一个空格和函数名称，接下来是一对圆括号，在圆括号内是形式参数列表，如果有多个参数则使用逗号分隔开，圆括号之后是一个冒号和换行，最后是函数体代码。

注意的问题

① 函数形参不需要声明其类型，也不需要指定函数返回值类型；

② 即使该函数不需要接收任何参数，也必须保留一对空的圆括号；

③ 括号后面的冒号必不可少；

④ 函数体相对于def关键字必做保持一定的空格缩进。

BMI计算

```
def calBMI(w,h):
    BMI = w/(h**2)
    if BMI<18.5:
        print("偏瘦")
    elif 18.5<=BMI<24:
        print("正常")
    elif 24<=BMI<28:
        print("偏胖")
    else:
        print("肥胖")

while True:
    weight = float(input("请输入您的体重："))
    height = float(input("请输入您的身高："))
    calBMI(weight,height)
```

函数参数

- 函数定义时圆括号是使用逗号分隔开的形参列表（parameters）。
- 一个函数可以没有参数，但是定义和调用时一对圆括号必须要有，表示这是一个函数并且不接收参数。
- 函数调用时向其传递实参（arguments），根据不同的参数类型，将实参的值和引用传递给形参。
- 实参与对应的形参，个数要一样

不可变参数 & 可变参数

- 传值
  - 以整型、浮点数、字符串、布尔型等基本类型为参数
- 传址
  - 以列表、元组、字典、集合为参数

函数参数——默认值参数

- 定义函数时为形参设置默认值
- 在调用函数时是否为默认值参数传递实参是可选的

```
def demo(v=6):
    print(v)
i=7
demo(i)
```

```
def demo(s, v=6):
    print(s+v)
demo(5, 8)
```

```
def demo(v=6):
    print(v)
i=7
demo()
```

```
def demo(s, v=6):
    print(s+v)
demo(5)
```

函数参数——关键字参数

- 普通参数是位置参数，关键字参数是在传入实参时指定形参的变量名——适用于参数多的函数

```
>>> def demo(a, b, c=5):
        print(a, b, c)


>>> demo(3, 7)
3 7 5
>>> demo(a=7, b=3, c=6)
7 3 6
>>> demo(c=8, a=9, b=0)
9 0 8
```

```
>>> demo(7, c=8, b=1)
7 1 8
```

```
>>> demo(c=8, 7, 8)
SyntaxError: positional argument follows keyword argument
```

函数参数——可变长度参数

- 可变长度参数在定义函数时主要有两种形式：
  - *parameter
    - 接收任意多个实参并将其放在一个**元组**中
  - **parameter
    - 接收类似关键参数一样赋值形式的多个实参并将其放入**字典**中

```
>>> def demo(*p):
        print(p)


>>> demo(1,2,3)
(1, 2, 3)
>>> demo([1,2,3])
([1, 2, 3],)
```

```
>>> def demo(**p):
        for item in p.items():
            print(item)


>>> demo(x=1,y=2,z=3)
('x', 1)
('z', 3)
('y', 2)
```

函数参数——传递参数时的序列解包

- 调用含有多个参数的函数时，可以使用Python列表、元组、集合、字典以及其他可迭代对象作为实参，并在实参名称前加一个星号，Python解释器将自动进行解包，然后传递给多个单变量形参。

```
>>> def demo(a, b, c):
        print(a+b+c)
```

```
>>> demo(*d.values())
abc
```

```
>>> s=[1, 2, 3]
>>> demo(*s)
6
>>> t=(1, 2, 3)
>>> demo(*t)
6
>>> d={1:'a', 2:'b', 3:'c'}
>>> demo(*d)
6
>>> set1={1, 2, 3, 1}
>>> demo(*set1)
6
```

变量作用域

- 变量起作用的代码范围称为**变量的作用域**，不同作用域内同名变量之间互不影响
  - 就像不同文件夹里同名文件之间互不影响一样
- 一个变量在函数外部定义和在函数内部定义，其作用域是不同的
  - 函数内部定义的变量一般为**局部变量**
  - 函数外部定义的变量为**全局变量**

局部变量

```
>>> def demo():
        x=3
        print(x)


>>> x=5
>>> x
5
>>> demo()
3
>>> x
5
```

```
>>> def demo():
        y=4
        print(y)


>>> y
Traceback (most recent call last):
  File "<pyshell#47>", line 1, in
    y
NameError: name 'y' is not defined
```

全局变量

```
>>> def demo1(m):
        print(m)


>>> def demo2(n):
        print(n)


>>> x=5
>>> demo1(x)
5
>>> demo2(x)
5
```

```
>>> def demo():
        global x
        x=3
        print(x)


>>> demo()
3
>>> x=5
>>> demo()
3
>>> x
3
```

一个变量已在函数外定义，如果在函数内需要修改这个变量的值，并将这个赋值结果反映到函数之外，可以在函数内用**global**明确声明要使用已定义的同名全局变量。

全局变量

```
>>> def demo():
        global x
        x=4
        print(x)


>>> x
Traceback (most recent call last):
  File "<pyshell#72>", line 1, in <module>
    x
NameError: name 'x' is not defined
>>> demo()
4
>>> x
4
```

在函数内部直接使用global关键字将一个变量声明为全局变量。如果在函数外没有定义该全局变量，在**调用这个函数之后**，会自动增加新的全局变量。

lambda表达式

- 常用来声明匿名函数，即没有函数名字的临时使用的小函数
- 表达式的计算结果相当于函数的返回值

```
>>> f=lambda x,y,z:x+y+z
>>> print(f(1,2,3))
6
>>> g=lambda x,y=2,z=3:x+y+z
>>> print(g(1))
6
>>> print(g(2,z=4,y=5))
11
>>> L=[(lambda x:x**2),(lambda x:x**3)]
>>> print(L[0](2),L[1](2))
4 8
```

range函数

- 定义
  - range(start, end, step)
- >>> a=range(1,10)
- >>> type(a)
- >>> list(a)
- 一般用于for循环：for e in range(1,100)
  - e取得1到99中的每一个数

map函数

- 把函数映射到可变组合数据类型上的每个元素
- 时常接合无名函数使用
- 结果会被转换成列表使用
- 例如:
  - list(map(lambda x:x+2, list(range(1,10))))
  - list(map(lambda x:x+2,set(range(1,10))))
  - list(map(lambda x:x+2,{1:2,2:3,3:4}))

应用举例——素数问题

1. 输入一个数，判断是不是素数——函数IsPrime
2. 复用IsPrime，求1~200中的素数
3. 复用IsPrime，求前50个素数
4. 可逆素数：指将一个素数的各位数字顺序倒过来构成的反序数仍然是素数。求所有的四位素数。
5. 回文素数：当从左向右的读法与从右向左的读法是相同且为素数时则称该数为回文素数。求1000之内的所有回文素数。

应用举例2

- 编写根据输入长度产生随机验证码的函数codeGenerator(length)。
  - 参数length表示验证码的长度
  - 验证码的字符包括：0-9，a-z，A-Z

应用举例3

- 编写函数，实现比赛分数计算功能。
  - 可以输入任意个评委的成绩（百分制）
  - 输出评委的个数，以及最终成绩
  - 计算规则
    - 去掉一个最高，去掉一个最低分

应用举例4

- 把加密方法编写成一个函数。
  - 输入任意长度的整数
  - 各位加5对9求余后的数替换此数
  - 并把第一位和最后一位交换
  - 输出加密后的整数

应用举例5

- 如果一个渔夫从2000年1月1日开始每三天打一次鱼两天晒一次网，编程实现当输入2000年1月1日以后的任意一天，输出该渔夫是打渔还是晒网？
- **输入样式：** >>> 2017 4 27
- **输出样式：** >>> 2017-4-27打鱼

Thank You!