第5章 函数和代码复用

目录

CONTENTS

5.5 函数的递归

目录

CONTENTS

5.5.1 递归的定义

函数作为一种代码封装，可以被其他程序调用，当然，也可以被函数内部代码调用。这种函数定义中调用函数自身的方式称为递归。就像一个人站在装满镜子的房间中，看到的影像就是递归的结果。递归在数学和计算机应用上非常强大，能够非常简洁的解决重要问题。

数学上有个经典的递归例子叫阶乘，阶乘通常定义为：

$$n! = n(n-1)(n-2)...(1)$$

这个关系给出了另一种表达阶乘的方式：

$$n! = \begin{cases} 1 & n = 0 \\ n(n-1)!otherwise \end{cases}$$

阶乘的例子揭示了递归的2个关键特征：

（1）存在一个或多个基例，基例不需要再次递归，它是确定的表达式；

（2）所有递归链要以一个或多个基例结尾。

5.5.2 递归的使用方法

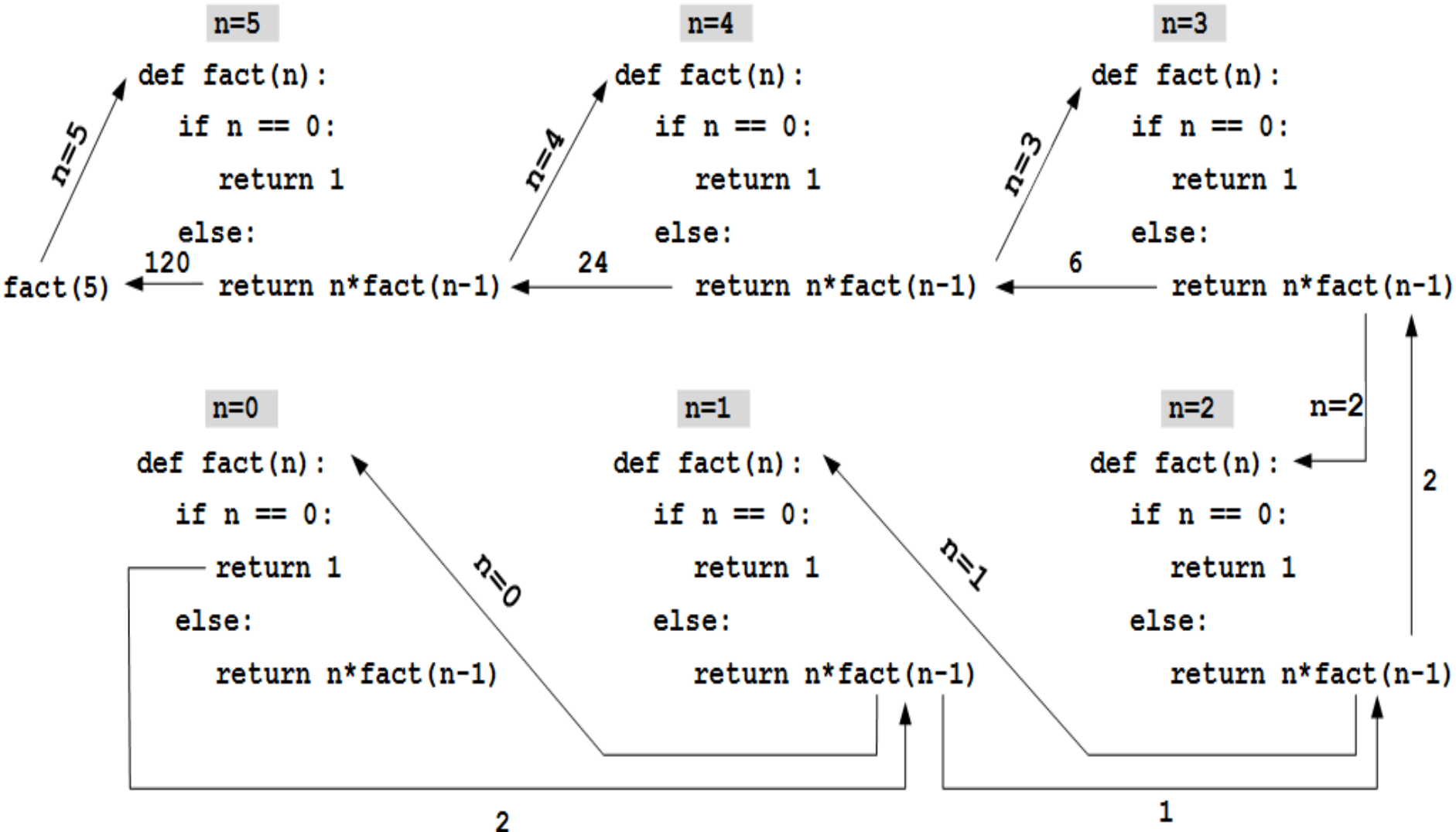
实例5.2：阶乘的计算。

根据用户输入的整数n，计算并输出n的阶乘值。

$$n! = \begin{cases} 1 & n = 0 \\ n(n-1)! & otherwise \end{cases}$$

实例5.2　　5.2CalFactorial.py

```
1  def fact(n):
2      if n == 0:
3          return 1
4      else:
5          return n * fact(n-1)
6  num = eval(input("请输入一个整数: "))
7  print(fact(abs(int(num))))
```


n=5
def fact(n):
if n == 0:
return 1
else:
return n*fact(n-1)
n=5
fact(5)
120
n=4
def fact(n):
if n == 0:
return 1
else:
return n*fact(n-1)
n=4
24
n=3
def fact(n):
if n == 0:
return 1
else:
return n*fact(n-1)
n=3
6
n=2
def fact(n):
if n == 0:
return 1
else:
return n*fact(n-1)
n=2
2
n=1
def fact(n):
if n == 0:
return 1
else:
return n*fact(n-1)
n=1
1
n=0
def fact(n):
if n == 0:
return 1
else:
return n*fact(n-1)
n=0
2

实例5.3：字符串反转。

对于用户输入的字符串s，输出反转后的字符串。

解决这个问题的基本思路是把字符串看作一个递归对象。

```
1  def reverse(s):
2      return reverse(s[1:]) + s[0]
```

观察这个函数的工作过程。s[0]是首字符，s[1:]是剩余字符串，将它们反向连接，可以得到反转字符串。执行这个程序，结果如下：

```
>>>def reverse(s):
    return reverse(s[1:]) + s[0]
>>> reverse("ABC")
...
RecursionError: maximum recursion
depth exceeded
```

reverse（）函数没有基例，递归层数超过了系统允许的最大递归深度。

默认情况下当递归调用到1000层，python解释器将终止程序。递归深度是为了防止无限递归错误而设计的，当用户编写的正确递归程序需要超过1000层时，可以通过如下代码设定。

```
>>>import sys
>>>sys.setrecursionlimit(2000)  #2000是新的递归层数
```

```
def reverse(s):
    if s==“”:
        return s
    else:
        return reverse(s[1:]) + s[0]
str=input(“请输入一个字符串：”)
print(reverse(str) )
```

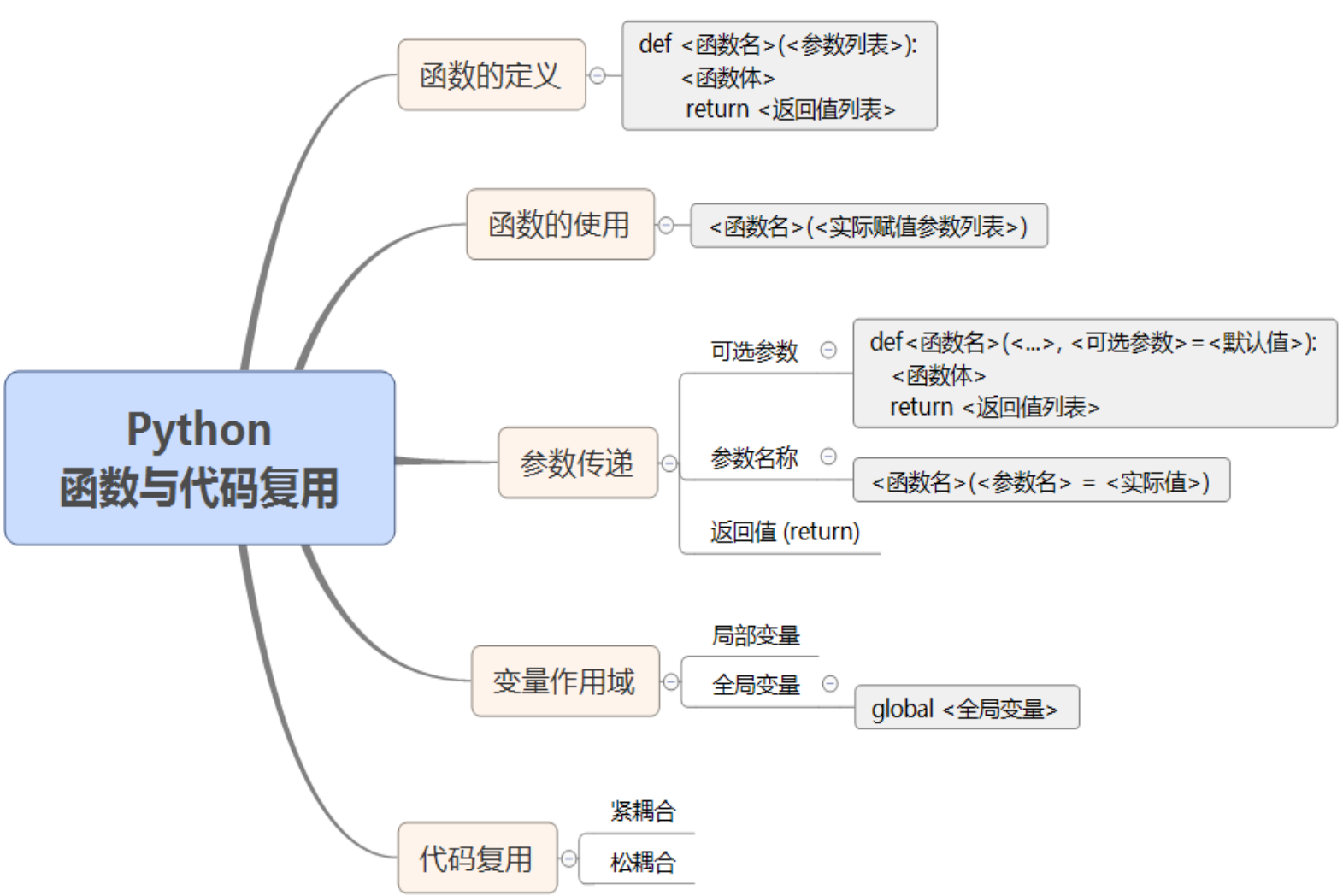
此例中把基例设置为字符串的最短形式，即空字符串。

总结与拓展

做什么？

01 本章总结

- 函数的定义、调用、lambda函数的使用
- 函数的参数传递
- 局部变量和全局变量
- 代码复用和模块化设计
- 函数的递归


Python
函数与代码复用
函数的定义
def <函数名>(<参数列表>):
<函数体>
return <返回值列表>
函数的使用
<函数名>(<实际赋值参数列表>)
参数传递
可选参数
def<函数名>(<...>, <可选参数>=<默认值>):
<函数体>
return <返回值列表>
参数名称
<函数名>(<参数名> = <实际值>)
返回值 (return)
变量作用域
局部变量
全局变量
global <全局变量>
代码复用
紧耦合
松耦合

总结与拓展

做什么？

02 拓展作业

学生管理系统主要负责编辑学生信息，适时地更新学生资料。例如：新生入校要在学生管理系统中录入刚入校的学生信息。编写一个学生管理系统，要求如下：

a.使用自定义函数，完成对程序的模块化；

b.学生信息至少包含姓名、性别及手机号；

c.该系统具有的功能：添加、删除、修改、显示学生信息、退出系统。

THANKS