

# 函数进阶

# 1

## 函数进阶

### ——变量作用域

# 全局变量

- 标识符的作用域是定义为其声明在程序里的可应用范围，也就是变量的可见性
- 在一个模块中最高级别的变量有全局作用域
- 全局变量的一个特征是除非被删除掉，否则它们的存活到脚本运行结束，且对于所有的函数，它们的值都是可以访问的

# 局部变量

- 局部变量只时暂时地存在，仅仅只依赖于定义它们的函数现阶段是否处于活动
- 当一个函数调用出现时，其局部变量就进入声明它们的作用域。在那一刻，一个新的局部变量名为那个对象创建了
- 一旦函数完成，框架被释放，变量将会离开作用域

## 局部变量（续1）

- 如果局部与全局有相同名称的变量，那么函数运行时，局部变量的名称将会把全局变量名称遮盖住

```
>>> x = 4
>>> def foo():
...     x = 10
...     print('in foo, x =', x)
...
>>> foo()
in foo, x = 10
>>> print('in main, x =', x)
in main, x = 4
```

# global语句

- 因为全局变量的名字能被局部变量给遮盖掉，所以为了明确地引用一个已命名的全局变量，必须使用global语句

```
>>> x = 4
>>> def foo():
...     global x
...     x = 10
...     print('in foo, x =', x)
...
>>> foo()
in foo, x = 10
>>> print('in main, x =', x)
in main, x = 10
```

# 名字空间

- 任何时候，总有一个到三个活动的作用域（内建、全局和局部）
- 标识符的搜索顺序依次是局部、全局和内建
- 提到名字空间，可以想像是否有这个标识符
- 提到变量作用域，可以想像是否可以“看见”这个标识符

# 2

函数进阶

——函数式编程



# 偏函数

- 偏函数的概念是将函数式编程的概念和默认参数以及可变参数结合在一起
- 一个带有多个参数的函数，如果其中某些参数基本上固定的，那么就可以通过偏函数为这些参数赋默认值

```
>>> from operator import add
>>> from functools import partial
>>> add10 = partial(add, 10)
>>> print(add10(25))
35
```

- 编写一个窗口程序，要求如下
  1. 窗口程序提供三个按钮
  2. 其中两个按钮的前景色均为白色，背景色为蓝色
  3. 第三个按钮前景色为红色，背景色为红色
  4. 按下第三个按钮后，程序退出

# 递归函数

- 如果函数包含了对其自身的调用，该函数就是递归的
- 在操作系统中，查看某一目录内所有文件、修改权限等都是递归的应用

```
>>> def func(num):  
...     if num == 1:  
...         return 1  
...     else:  
...         return num * func(num - 1)  
...  
>>> print(func(5))  
120  
>>> print(func(10))  
3628800
```

- 实现快速排序
  1. 随机生成10个数字
  2. 利用递归，实现快速排序



更多精彩...



<http://bj.linux.tedu.cn/>  
企业QQ：86198501