在Python中，通过def去定义函数。def语句是实时执行的代码：函数并不存在，直到Python运行了def后，其会生成一个新的函数对象并赋值给函数名这个变量。同之前学习的变量与对象之间的关系类似。

例：程序function\_detail/function\_detail1

def pets(pet):  
 print("pet is ", pet)  
  
  
# 函数名pets是一个变量，  
# 其引用了一个函数对象  
# 将其赋值给pets\_1后，  
# 可以通过pets\_1调用函数  
pets\_1 = pets  
pets\_1('cat')

输出为：

pet is cat

Python函数中，return语句返回的是一个结果对象。

Python中的多态：依赖类型的行为。

例：function\_detail/function\_detail2

def times(x, y):  
 return x \* y  
  
  
print(times(2, 4)) # 8  
print(times('Ni', 4)) # NiNiNiNi

根据输入对象类型的不同，执行不同的操作，这是Python中的多态。如果输入的对象无法执行定义的操作，Python就会抛出错误。

例：function\_detail/function\_detail2

def intersect(seq1, seq2):  
 res = []  
 for x in seq1:  
 if x in seq2:  
 res.append(x)  
 return res  
  
  
s1 = 'SPAM'  
s2 = 'SCAM'  
print(intersect(s1, s2))  
  
# 列表中对象的类型有可能不相同  
list1 = [1, 2, 3, 'a']  
list2 = (1, 'a')  
print(intersect(list1, list2))

输出为：

['S', 'A', 'M']

[1, 'a']