

# 匿名函数

用lambda关键词能创建小型匿名函数。这种函数得名于省略了用def声明函数的标准步骤。

lambda函数的语法只包含一个语句，如下：

```
lambda 参数列表: 运算表达式
```

如下实例：

```
sum = lambda arg1, arg2: arg1 + arg2

# 调用sum函数
print("Value of total : %d" % sum( 10, 20 ))
print("Value of total : %d" % sum( 20, 20 ))
```

以上实例输出结果：

```
Value of total : 30
Value of total : 40
```

Lambda函数能接收任何数量的参数但只能返回一个表达式的值

匿名函数可以执行任意表达式(甚至print函数)，但是一般认为表达式应该有一个计算结果供返回使用。

python在编写一些执行脚本的时候可以使用lambda，这样可以接受定义函数的过程，比如写一个简单的脚本管理服务器。

## 应用场合

### 函数作为参数传递

```
>>> def fun(a, b, opt):
...     print("a = " % a)
...     print("b = " % b)
...     print("result =" % opt(a, b))
...
>>> add = lambda x,y:x+y
>>> fun(1, 2, add) # 把 add 作为实参传递
a = 1
b = 2
result = 3
```

练习:

有一个列表

```
students = [
    {'name': 'zhangsan', 'age': 18, 'score': 92},
    {'name': 'lisi', 'age': 20, 'score': 90},
    {'name': 'wangwu', 'age': 19, 'score': 95},
    {'name': 'jerry', 'age': 21, 'score': 98},
    {'name': 'chris', 'age': 17, 'score': 100},
]
```

要求，对上述列表里的数据按照score进行升序排序。

Python中使用函数作为参数的内置函数和类:

函数名 或类名	功能	参数描述
sorted 函数	用来将一个无序列表进行排序	函数参数的返回值规定按照元素的哪个属性进行排序
filter类	用来过滤一个列表里符合规定的所有元素，得到的结果是一个迭代器	函数参数的返回值指定元素满足的过滤条件
map类	将列表里的每一项数据都执行相同的操作，得到的结果是一个迭代器	函数参数用来指定列表里元素所执行的操作
reduce 函数	对一个序列进行压缩运算，得到一个值。python3以后，这个方法被移到了functools模块	函数参数用来指定元素按照哪种方式合并