

# 高阶函数

在Python中，函数其实也是一种数据类型。

```
def test():  
    return 'hello world'  
print(type(test)) # <class 'function'>
```

函数对应的数据类型是 `function`，可以把它当做是一种复杂的数据类型。

既然同样都是一种数据类型，我们就可以把它当做数字或者字符串来处理。

## 定义一个变量指向函数

在Python中，我们还可以定义一个变量，让它来指向一个函数，相当于给函数起了一个别名。

```
def test():  
    return 'hello wrold'  
  
fun = test # 定义了一个变量fun,让它指向了 test 这个函数  
print(fun()) # 使用fun()可以直接调用test这个函数  
  
print(id(fun)) # 1819677672040  
print(id(test)) # 1819677672040
```

注意：在定义一个变量表示一个函数时，函数后面不能加括号！加括号表示的是调用这个函数。

```
def test():  
    return 'hello world'  
  
result = test() # 这种写法是调用test函数，并把函数的返回值赋值给result变量  
print(result()) # 这里会报错 TypeError: 'str' object is not callable  
  
fun = test # 这种写法是给test函数起了一个别名，注意，这里的test后面不能加()  
fun() # 可以使用别名调用这个函数
```

## 高阶函数

既然变量可以指向函数，函数的参数能接收变量，那么一个函数就可以接收另一个函数作为参数，同样，我们还可以把一个函数当做另一个函数的返回值。这种函数的使用方式我们称之为高阶函数。

## 函数做为另一个函数的参数

```
def test(age,action):
    if age < 18:
        print('您还没满十八岁，请退出')
    action()    # 把参数action直接当做一个函数来调用

def smoke():
    print('我已经年满十八岁了，我想抽烟')

my_action = smoke # 定义一个变量my_action, 让它指向smoke函数
test(21, my_action) # 将my_action传给 test 函数作为它的参数

test(21,smoke) # 还可以不再定义一个新的变量，直接传入函数名
```

## 函数作为另一个函数的返回值

```
def test():
    print('我是test函数里输入的内容')

def demo():
    print('我是demo里输入的内容')
    return test # test 函数作为demo函数的返回值

result = demo() # 我是demo里输入的内容 调用 demo 函数，把demo函数的返回值赋值给 result
print(type(result)) # <class 'function'> result 的类型是一个函数

result() # 我是demo里输入的内容    我是test函数里输入的内容    既然result是一个函数，那么就可以直接使用() 调用这个函数

demo()( ) # 我是demo里输入的内容    我是test函数里输入的内容
```