day10：函数嵌套

笔记本：	Python基础		
创建时间：	2018/6/15 9:18	**更新时间：**	2018/7/4 19:52
作者：	liuchang_0412@163.com		

1.可变/不可变类型 实参的参数传递类型：

*问题：函数只能通过返回值传回数据吗?

示例1（可变类型：列表）：

```
L=[ ]
def fn(x):
    x.append(10)
fn(L)
print(L)    # [10]
```

示例2（可变类型：字典）：

```
D={ }
def fn(x):
    x['name']='tarena'
fn(D)
print(D)    # {name:tarena}
```

示例3（不可变类型：元组）：

```
t=(1,2,3)
def fn(x):
    x[1] = 2.2
print(t)
fn(t)           # 对元组赋值出错
print(t)
```

区别：不可变的类型的数据作为函数参数传入时，函数内部不会改变变量的原数据值，是安全的。可变类型的数据作为参数传入时，函数内部可以改变原数据，多用来返回更多数据结果。

#练习：写一个函数，在函数内部读取学生姓名，并存入列表中，通过两种方式返回学生姓名数据并打印出来。方式一：返回值返回数据。方式二：通过参数返回数据。

2.函数嵌套：函数嵌套是指一个函数里用def语句来创建其他函数的情况

①函数变量：函数名是变量，它在创建函数时绑定一个函数

示例：

```
def fn( ) :
    print("hello world!)
```

```
    f1 = fn
    f1( )       #等同于调用函数fn( )
```

②嵌套示例：

```
    def fn_outer( ) :       #外部函数
        print("外部函数被调用")
        def fn_inner( ) :   #内部函数
            print("内部函数被调用")
        fn_inner( )
        fn_inner( )
        print("外部函数调用结束")

    fn_outter( )
    #fn_inner( )       # 错误，内嵌函数只存在于函数内部
```

3.函数作为函数的返回值：

示例：

```
    def get_fn( ) :
        def print_hello( ) :
            print("hello")
        return print_hello

    fn = getfn( )
    fn( )
```

#练习：写一个函数，此函数有一个参数op,如下：

```
def get_op(op):
    ...
```

此函数在传入字符串"+"时，返回加操作函数myadd(x,y);
此函数在传入字符串"-"时，返回加操作函数mysub(x,y);
此函数在传入字符串"*"时，返回加操作函数mymul(x,y);
此函数在传入字符串"/"时，返回加操作函数mydiv(x,y);

主函数：

```
a=int(input("请输入第一个数：")
b=int(input("请输入第二个数：")
operator = input("请输入操作方式：")
fn=get_op(operator)
print("结果是：",fn(a,b))
```

测试用例：

1,2,+ -->3
3,2,* -->6

4.函数作为函数的参数传递：
示例：

```
def table(x,y):
    return "|" + x.center(13) + "|" + y.center(13) + "|"
def string(x,y):
    return "姓名：" + x + "年龄：" + y
def my_print(fx,x,y):
    print(fx(x,y))

myprint(table,"tarena","15")
myprint(table,"小张","18")
```

5.全局变量和局部变量：
①局部变量：定义在函数内部的变量（包含函数参数）
②全局变量：定义在函数外部，模块内部的变量

示例：

```
v=100
def fn( ):
    print(v)
fn( )          # --->100
```

③Python作用域：作用域，也叫名字空间，是变量访问的时候查找变量名的范围空间
*局部作用域（函数内,local, L）：
*外部嵌套作用域（Enclosing function locals, E）：
*函数定义所在模块（文件）的作用域（Global, G）：
*Python内置模块的作用域（Builtin(Python), B）：
④变量名的查找规则：在访问变量时，先查找本地变量，然后是包裹此函数的外部函数的函数内部的变量，之后是全局变量，最后是内置变量。 L --> E --> G --> B
⑤在默认情况下，变量名赋值会创建或修改本地变量

示例见：namespace.py

6.global语句：告诉解释器，global语句生命的一个或多个变量，这些变量的作用域为模块级的作用域，也称作全局变量。
语法： global 变量名1,变量名2
示例：见global.py
说明：

1.全局变量如果要在函数内部被赋值，则必须经过全局声明，否则被认为是局部变量
2.全局变量在函数内部不经过声明就可以直接访问（前提是变量已经存在）
3.不能先声明局部变量，再用global声明为全局变量，此做法不符合语法规则
4.global变量列表里的变量名不能出现在此作用域的参数列表里，for循环控制目标，类定义，函数定义及import导入名字中

7.nonlocal语句：告诉解释器，nonlocal声明的变量不是局部变量，也不是全局变量，而是上一层外部嵌套函数内的变量。
语法：nonlocal 变量名1,变量名2
示例：见nonlocal.py
说明：
1.nonlocal语句只能在被嵌套函数的内部进行使用
2.访问nonlocal变量将对外部嵌套函数的作用域内的变量进行操作
3.当有两层或两层以上函数嵌套时，访问nonlocal变量只对最近一层的变量进行操作
4.nonlocal语句的变量列表里的变量名，不能出现在此作用域的参数列表中，for循环控制目标，类定义，函数定义及import导入名字中

#练习：
1.给出一个数n，写一个函数来计算1+2+3+..+n的和，要求用函数来做（如print(mysun(100)）
2.给出一个数n，写一个函数来计算n!(n的阶乘)，n!=1*2*3*...*n。（如print(myfac(5)）
3.给出一个数n，写一个函数来计算1+2**2+3**3+...+n**n的和，注意n给个小数点的数。
4.写函数打印n层杨辉三角