

一.循环嵌套

1.while循环嵌套

python之禅：扁平胜于嵌套

print()的参数：end="\n", 如果不想以\n换行的话，则使用end = ''

```
while True:
    num = int(input("请输入数字: "))
    n = 0
    while n <= num:
        print(n)
        n += 1
```

2.for 循环嵌套

```
for i in range(1, 10):
    for j in range(1, i + 1):
        print("{}*{}={:<4}".format(i, j, i * j), end="")
    print("\n")
```

二.函数

1.功能：可以用对重复使用的功能的代码进行封装，提高代码的复用性，提高工作效率，降低代码的耦合性

2.定义：通过关键字def定义

def 函数的名字():

 函数的功能代码

#定义好的函数，不会执行运行，需要调用才会执行函数内部的代码

#函数的参数：定义函数的时候，可以在后面的括号中定义参数

函数名命名规范：

- 1、变量只能使用数字、字母、下划线组成
- 2、变量不能使用数字开头
- 3、不能使用python中的关键字(内置函数，官方库，第三库的名字统统不要用)

命名风格：推荐使用下划线命名法

3.函数的参数类型：

1.形参：定义函数中的参数为形参,三种定义形式：

必需参数：定义了就一定要传，个数不能多也不能少

默认参数：定义的时候可以设置默认值，调用的时候可以传，也可以不传

不定长参数（动态\可变参数）：可以接收0个或者多个参数

```
def fun1(*args):          #只能接收位置传参，以元组的形式保存
    print(args)
fun1()
fun1(11)
```

```
def fun2(**kwargs):       #只能接收关键字传参，以字典的形式保存
    print(kwargs)
fun2(a=11,b=22,c=33)
fun2(a=11)
```

```
def fun3(*args,**kwargs):
    print(args,kwargs)
fun3(11,22,33,a=11,b=22).    #调用先写位置参数，再写关键字参数
```

2.实参：调用函数中的参数为实参有两种传参形式：

位置参数：按照参数定义的位置进行传递

关键字参数：通过参数名字指定，

注意：如果需要混合传参，先写位置参数，后写关键字参数（不然就会报错）

3.函数的参数拆包：

*： *加在形参前面（定义函数参数的时候使用），接收不定长的位置参数，转换为元组

 *加在实参前面（调用函数参数的时候使用），可以将元组（列表）拆包成多个位置参数

**： 加在形参前面（定义函数参数的时候使用），接收不定长的关键字参数，从转换为字典

 **加在实参前面（调用函数参数的时候使用），可以将字典（列表）拆包成多个关键字参数

```
tu= (11,22,33)
def fun(a,b,c):
    • print(a,b,c)
fun(*tu)
dic ={a=11,b=22,c=33}
def fun2(**kwargs):
    • print(kwargs)
fun2(**dic)
```

4.已经学过的内置函数：

print():输出

input():输入

id():获取数据的内存地址

len(): 获取数据的长度

range(n): 获取一个0到n-1的整数序列

dict(): 生成一个字典

type(): 获取数据的类型

```
def add(a,b):      #a,b 为形参,
    • print(a+b)
add(11,22) # 11, 22为实参. 位置传参
add(a=11,b=22)     #关键字传参
def fun3(*args,**kwargs):
    print(args,kwargs)
fun3(11,22,33,a=11,b=22). 先写位置参数, 再写关键字参数
```